

Instituto Politécnico de Leiria
13 e 14 de novembro

EP



**21º Encontro Português de
Computação Gráfica**

Livro de Atas



IPL

instituto politécnico de leiria



GPCG

Grupo Português de Computação Gráfica

Livro de Atas
EPCG - 2014
21º Encontro Português de Computação Gráfica

Nov 13-14, 2014

Leiria, Portugal

Atas do 21º Encontro Português de Computação Gráfica - EPCG 2014

Escola Superior de Tecnologia e Gestão
Instituto Politécnico de Leiria
Leiria, Portugal

Alexandrino Gonçalves, António Ramires Fernandes, Nuno Rodrigues (editores)

Publicado pelo
Instituto Politécnico de Leiria
Rua General Norton de Matos
Apartado 4133
2411 - 901 Leiria

ISBN: 978-972-8793-71-5
ebook ISBN: 978-972-8793-72-2
CD-ROM ISBN: 978-972-8793-73-9
Depósito Legal: 383157/14

Organização, Parcerias e Apoios

Coorganização

Departamento de Engenharia Informática
Escola Superior de Tecnologia e Gestão
Centro de Investigação em Informática e Comunicações
Instituto Politécnico de Leiria (<http://www.ipleiria.pt>)

Grupo Português de Computação Gráfica (<http://www.gpcg.pt>)

Parceria

Núcleo de Engenharia Informática da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria (<http://nei-estg.com>)

Apoios Empresariais

NVIDIA (<http://www.nvidia.com>)
Microsoft (<http://www.microsoft.com>)

Apoios Institucionais

Serviços de Ação Social do Instituto Politécnico de Leiria (<http://www.ipleiria.pt/sas/>)
Câmara Municipal de Leiria (<http://www.cm-leiria.pt>)
Santa Casa da Misericórdia de Leiria (<http://www.misericordia.deleiria.pt>)

L^AT_EXclass and proceedings files by Giampiero Salvi
(<https://github.com/giampierosalvi/LaTeXProceedings>)

Prefácio

Bem vindos ao 21º Encontro Português de Computação Gráfica, EPCG 2014, um encontro promovido pelo Grupo Português de Computação Gráfica. Organizado pelo Departamento de Engenharia Informática da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria, este encontro afirma-se mais uma vez como o ponto de encontro para um grande número de investigadores, docentes, estudantes e profissionais da área da Computação Gráfica, permitindo, no que toca à investigação, divulgar o que de melhor se faz em Portugal, potenciando, desse modo, sinergias para colaborações futuras.

A diversidade da investigação realizada nesta área no nosso país, está bem patente no leque de temas cobertos pelos artigos aceites. De um total de 31 artigos submetidos foram selecionados 19 artigos longos e 5 curtos, abrangendo as seguintes áreas: Jogos e Entretenimento, Interação, Síntese e Visualização, Modelação e Reconstrução, Aplicações, Visão por Computador e Realidade Aumentada.

Os artigos foram revistos no regime de blind reviewing. Gostariamos de agradecer a todos aqueles que contribuíram com a sua dedicação e conhecimento para o processo de revisão. As atas desta edição contam com a participação de 72 autores, distribuídos por um total de 24 artigos divididos pelas áreas apresentadas.

Neste Encontro contamos ainda com dois oradores convidados. Diego Gutierrez (Universidad de Zaragoza, Espanha) é o convidado estrangeiro desta edição. Com uma presença marcante na Siggraph 2014, com um total de 4 artigos, será certamente uma mais valia para o Encontro. A sua comunicação tem como título “Computational Imaging: how to see the invisible”. Joaquim Jorge, um dos fundadores do GPCG e que certamente dispensa apresentações, com a palestra intitulada “Touching more than three dimensions: having fun with surfaces”.

No programa do Encontro consta ainda a entrega dos prémios de melhor artigo, melhor apresentação, e a outorga do “Prémio Professor José Luís Encarnação 2014”, promovido pelo GPCG, para o melhor artigo publicado até 30 de Setembro de 2014, num fórum internacional de prestígio comprovado.

Gostariamos de agradecer aos autores e oradores convidados as preciosas contribuições para este Encontro, à Comissão Científica pelo importante trabalho desenvolvido no processo de revisão, à Comissão Organizadora por possibilitar as condições necessárias para a realização da conferência na Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria. Queremos igualmente agradecer ao Grupo Português de Computação Gráfica o convite que nos foi dirigido e o apoio na organização da 21ª edição deste Encontro que em muito tem contribuído para o panorama nacional da Computação Gráfica. A todas as entidades institucionais, empresariais e parcerias, um agradecimento especial pelo vosso apoio essencial à realização deste evento.

Esperamos sinceramente que este Encontro seja do vosso agrado e aguardamos com expectativa as próximas atividades do Grupo Português de Computação Gráfica.

Cordiais Saudações,

António Ramires Fernandes, UM
Nuno Rodrigues, ESTG/CIIC-IPLeiria
(Copresidentes da Comissão Científica)

Alexandrino Gonçalves, ESTG/CIIC-IPLeiria
(Presidente da Comissão Organizadora)

Leiria, novembro de 2014

Comissão Organizadora

Alexandrino Gonçalves (Presidente)

Nuno Rodrigues

Comissão Científica

Co-presidentes

António Ramires Fernandes, UM

Nuno Rodrigues, ESTG/CIIC-IPLeiria

Membros

Abel Gomes, UBI

Adérito Marcos, UAberta

Alberto Proença, UM

Alexandrino Gonçalves, ESTG/CIIC-IPLeiria

Ana Paiva, FE/UP

Ana Maria Mendonça, FE/UP

Ana Paula Cláudio, FC/UL

António Augusto Sousa, FE/UP

António Cardoso Costa, ISEP/IPP

António Coelho, FE/UP

António Nestor Ribeiro, UM

Beatriz Sousa Santos, DET/UA

Bruno Oliveira, FC/UP

Carlos Urbano, ESTG/CIIC-IPLeiria

Daniel Gonçalves, INESC-ID

Elisabeth Carvalho, UAtlântica

Emanuel Peres, UTAD

Fernando Birra, FCT/UNL

Fernando Pereira, IST/IT

Francisco Pereira, UTAD

Frutuoso da Silva, UBI

Hélder Araújo, ISR/UC

João Cunha, LNEC

João Madeiras Pereira, INESC

João Manuel R. S. Tavares, INEGI/FEUP

João Paulo Moura, UTAD

João Pinto, IST/UTL

Joaquim Jorge, IST/UTL/INESC-ID

Joaquim Madeira, DET/UA

Jorge S. Marques, IST/ISR

José Carlos Teixeira, FCT/UC

José Creissac Campos, UM

José Francisco Morgado, ESTV/IPV

José Pereira, IPSetúbal

Luís Gonzaga Magalhães, UTAD

Luís Marcelino, ESTG/IPLeiria

Luis Paulo Reis, UM

Luís Paulo Santos, UM

Luís Romero, IPVC

Manuel João Ferreira, ENERMETER

Manuel João Fonseca, LaSIGE/FCUL

Maria Beatriz Carmo, UL

Mário Rui Gomes, IST/UTL

Mauro Figueiredo, UAlg

Maximino Bessa, ECT/UTAD/INESC-TEC

Miguel Leitão, ISEP/IPP

Miguel Sales Dias, Microsoft, ADETTI/ISCTE

Miguel Velhote Correia, FE/UP

Nuno Correia, FCT/UNL

Óscar Mealha DeCA/UA

Paulo Dias, DETI/IEETA/UA

Pedro Branco UM

Pedro Castro, ESTG/IPVC

Pedro Faria Lopes, ISCTE/ADETTI

Pedro Miguel Moreira, ESTG/IPVC/LIACC

Pedro Rangel Henriques UM

Rui Rodrigues, FE/UP

Teresa Chambel, FC/UL

Teresa Romão, FCT/UNL

Verónica Orvalho, FC/UP

Índice

Prefácio	iii
Sessão - Jogos e Entretenimento	1
Gestual Life - Jogo Educativo de Língua Gestual Portuguesa <i>Ricardo Almeida, Pedro Sousa, Luisa Brito, Marco Capela, Pedro Faria Lopes, Isabel Alexandre</i>	3
Personagens Virtuais Expressivas <i>Renato Teixeira, Ana Paula Cláudio, Maria Beatriz Carmo and Augusta Gaspar</i>	11
Fluxo-Um jogo pedagógico sobre o Sistema Circulatório (artigo curto) <i>Inês Silva, Pedro Lopes, Filipe Baptista, Jorge Calado, Isabel Alexandre</i>	19
Sessão - Interação	23
EduVis: Visualização Interativa de Dados Educacionais <i>Sandra Gama, Vilma Jordão, Daniel Gonçalves</i>	25
IN2TUNNEL - Terapia de Exposição para Fobia de Atravessar Túneis <i>Inês Laureano Gomes, Ana Paula Cláudio, Maria Beatriz Carmo and Augusta Gaspar</i>	33
LS3D: Pesquisa de Blocos LEGO em Ambientes Imersivos <i>Pedro Pascoal, Daniel Mendes, Diogo Henriques, Isabel Trancoso, Alfredo Ferreira</i>	41
Older Adult Performance Using Body Gesture Interaction <i>Micael Carreira, Daniel Gonçalves and Karine Ting</i>	49
Studying Color Blending for Visualizing Social Artifacts (artigo curto) <i>Sandra Gama, Daniel Gonçalves</i>	57
Sessão - Síntese e Visualização	61
Atmospheric Scattering - State of the Art <i>Diogo Lopes, António Ramires Fernandes</i>	63
Ray Reordering Techniques for GPU Ray-Cast Ambient Occlusion <i>Vasco Costa, João Madeiras Pereira and Joaquim Jorge</i>	71
Visualização Dinâmica de Nuvens através de Diagramas Termodinâmicos <i>Rui Duarte, José Francisco Morgado, Abel Gomes</i>	79
Sessão - Modelação e Reconstrução	87
Geração Procedimental de Traçados Rodoviários para Simulação de Condução <i>Carlos Campos, Miguel Leitão, António Coelho</i>	89
Modelação Expedita de Terrenos e Subsolo via Gestos no Ar com Superfícies Interactivas <i>Daniel Lopes, Daniel Simão, Fernando Fonseca, Daniel Mendes, Joaquim Jorge</i>	97
Segmentação de Malhas Triangulares baseada em Convexidade Relaxada <i>Rui Sérgio Rodrigues, José Francisco Morgado, Abel Gomes</i>	103
Sessão - Aplicações	111
Galactica, a Digital Planetarium that explores the Solar System and the Milky Way <i>Jorge D'Alpuim and Miguel Sales Dias</i>	113
Medidas e Detecção de Estruturas Morfológicas para Craniometria usando Modelos 3D <i>Luís Neves, Paulo Dias, Catarina Coelho, Maria Teresa Ferreira, Hélder Santos, Beatriz Sousa Santos</i>	121
Open Source Debuggers and Integration with a 3D Engine <i>André Liu, António Ramires Fernandes</i>	129
Visualização de Música à Distância de um Gesto <i>João Tiago Gomes, Maria Beatriz Carmo, Ana Paula Cláudio</i>	137

Artefactos e Exposições 3D Virtuais no Contexto duma Rede de Cultura e Arte Digital (artigo curto) <i>Luis Romero, Pedro Miguel Moreira, Carla Dias, Luís Carrilho</i>	145
Sessão - Visão por Computador e Realidade Aumentada	149
Sistema de Realidade Aumentada para Apoio ao Projeto de Arquitetura <i>Mariana Lopes, Jorge Silva, Miguel Sales Dias, Sara Eloy, Filipe Gaspar, Ricardo Miguel, Nuno Mendonça</i>	151
3D Reconstruction and Visualization of Liver and Vascular Networks from CT data using VTK and IGSTK <i>João Oliveira, José Luis Cuevas, José Blas Pagador, Hugo Capote and Francisco Margallo</i>	159
Indoor Image Geocoding using Synthetic Views <i>Nuno Amorim, Jorge Gustavo Rocha</i>	167
Anotações Multimédia Estruturadas para Ambientes Virtuais Colaborativos (artigo curto) <i>João Guerreiro, Daniel Medeiros, Alberto Raposo, Ismael Santos, Joaquim Jorge</i> . . .	175
Eery Proxemics: Proximidade à Distância usando Múltiplas Superfícies (artigo curto) <i>Mauricio Sousa, Daniel Mendes, Alfredo Ferreira, João Madeiras Pereira, Joaquim Jorge</i>	179
Índice de Autores	183

Sessão - Jogos e Entretenimento

Gestual Life

Jogo Educativo de Língua Gestual Portuguesa

Ricardo Almeida
DCTI – ISCTE-IUL
rfvaa@iscte-
iul.pt

Maria Luísa Brito
DCTI – ISCTE-IUL
mlcst@iscte-
iul.pt

Pedro Sousa
DCTI – ISCTE-IUL
pdcsa@iscte-
iul.pt

Marco Capela
DCTI - ISCTE-IUL
a30535@iscte-
iul.pt

Pedro Faria Lopes
DCTI - ISCTE-IUL, ADETTI-IUL, ISTAR-IUL
pedro.lopes@iscte.pt

Isabel Alexandre
DCTI - ISCTE-IUL, IT-IUL
isabel.alexandre@iscte.pt

Resumo

Jogos de computador educativos têm vindo a ser investigados ao longo das últimas décadas como um meio de ensino e abrangem áreas como a Matemática, Gestão, Biologia, Química e Medicina.

Para a realização do nosso projecto, decidimos investigar uma área que ainda não tivesse sido muito explorada. Desenvolvemos um jogo educativo que ensina Língua Gestual Portuguesa a pessoas ouvintes, visto que a demografia surda já possui esta língua como língua materna. Desenvolvemos todo o trabalho apresentado, desde scripts a modelos 3D.

De modo a conseguir proporcionar aos utilizadores uma experiência de total imersão, desenvolvemos um ambiente amigável com jogabilidade imersiva, interação com NPCs (Non-Player Characters ou personagens controladas pelo sistema) e desafios.

Palavras Chave

Agente, inteligente, gesto, língua, educacional, jogo

1. INTRODUÇÃO

O objetivo é criar um jogo educativo, numa área ainda pouco explorada: a Língua Gestual, de modo a estimular as pessoas para a sua aprendizagem. Cada país possui a sua própria Língua Gestual, dependendo da sua língua oficial, portanto focámo-nos na Língua Gestual Portuguesa.

O foco principal foi ensinar a Língua de forma introdutória para, através do jogo, estimular o desejo dos utilizadores em aprendê-la. Este é um método complexo de comunicação (gestos complexos com cinco parâmetros cada), por isso, ensiná-la por completo é uma tarefa para um professor certificado.

O nosso jogo coloca o jogador no papel de uma pessoa surda: os sons encontram-se todos distorcidos e existe apenas um meio de comunicação, os gestos.

O ato de ensinar no nosso jogo é feito através de vídeos contendo pessoas reais a executar os gestos, enquanto que o jogo em si se passa num mundo modelado tridimensionalmente onde o utilizador joga numa perspetiva de primeira pessoa. O utilizador pode interagir com objetos, falar com os NPC, resolver desafios e receber ajuda da sua amiga L.Y.S. (acrónimo para *Learn Your Signs* ou "Aprende os Teus Gestos"). Sempre que o jogador interage com um objeto, a L.Y.S. mostrar-lhe-á um vídeo

sobre como se representa esse objeto em língua gestual; esse vídeo será depois adicionado ao inventário do jogador onde este poderá aceder sempre que se queira recordar de alguma palavra.

2. MOTIVAÇÃO

Começámos por conceptualizar um jogo que, para além de preencher todos os requisitos de um bom jogo, conseguisse ensinar algo aos utilizadores. Esperávamos conseguir que os utilizadores, para além de se sentirem imersos no jogo, aprendessem bases de língua gestual e ainda proporcionar um incentivo à inclusão social dos utilizadores na comunidade de surdos.

Para que o jogador se sentisse imerso no jogo, decidimos colocá-lo no lugar de um surdo, onde ele não consegue ouvir e apenas pode comunicar através de gestos. De maneira a enfatizar este ponto, a única companheira que percebe completamente o jogador é a L.Y.S. Acreditamos que assim quem quer que experimente o nosso jogo se sentirá de alguma maneira mais ligado à comunidade de surdos, pois desta forma o jogador experimentaria o modo como estes se sentem de alguma forma isolados da sociedade.

O nosso jogo tem também um grande potencial para providenciar um ambiente mais acessível para estimular a

aprendizagem de língua gestual. Enquanto o jogador está entretido a jogar, está a aprender língua gestual, sem que a aprendizagem e a jogabilidade colidam.

3. TRABALHO RELACIONADO

Os jogos educativos são bastante vastos no que toca a temas, e durante a nossa investigação sobre esta área em específico dos videojogos, deparámo-nos com bastantes jogos sérios que pretendiam ensinar todo o tipo de assuntos, desde biologia a matemática [Kebritchi13] [DeCoito14]. No entanto, como escolhemos criar um jogo para motivar a aprendizagem da língua gestual, o nosso foco de pesquisa alterou-se um pouco e encontrámos uma multitude de jogos e sistemas relacionados com este assunto. Encontrámos um jogo "CopyCat" [Lee09] que tenta ajudar crianças surdas a fazer a transição de gestos simples para gestos mais complexos para que possam dizer frases completas ao invés de palavras individuais. Este projeto usa uma câmara para capturar os gestos que as crianças executam e tal como no nosso projeto foram utilizados vídeos para ensinar o utilizador, tendo também um ambiente animado representado no ecrã [Lee09]. Encontrámos outros projetos que tal como o "CopyCat" utilizam uma câmara para reconhecer os gestos realizados [Lisztes12].

A maior parte dos projetos que encontramos relacionados com o nosso tema, pretende ensinar a língua gestual Inglesa e o seu público-alvo consiste em crianças surdas [Zafrulla15][Parton16].

Em termos de design do jogo, a nossa investigação indica que os jogos sérios têm bastante potencial pois estes têm a vertente de ensino e ao mesmo tempo entretenimento sem entediar o utilizador. No entanto, é importante referir que o objetivo deste tipo de jogos não é ser utilizado como uma ferramenta completa de ensino mas sim, adequado para motivar principiantes ou testar a aquisição de conhecimentos. Nós adotamos estes conceitos no nosso jogo, e tivemos também em conta a estratégia de descoberta seguida de questão. Desta forma o utilizador captura os vídeos que representam os gestos através da exploração do mundo, e tem de aplicar o conhecimento adquirido resolvendo desafios [Bellotti07][Cheng08].

Como alguns trabalhos passados provaram, diferenciar conhecimento (comparando através de observação) e motivação através da jogabilidade é uma excelente maneira de aprender [Hwang01]. Outra maneira para o jogador realmente aprender através do jogo é imergir o jogador de maneira a que a jogabilidade e a aprendizagem não colidam [Wei02].

A investigação feita sobre design de jogos, avaliação da aprendizagem feita pelo utilizador e em relação à criação de uma experiência ao invés de um simples jogo [Hunicke04] foi fulcral no sucesso da criação de imersão, e da tomada de decisões acerca do design presente no nosso jogo. Relativamente à validação tanto da base de conhecimento como da aquisição de conhecimento por parte do jogador durante o jogo, nós desenvolvemos o nosso próprio método mas a publicação científica [Bellotti03] ofereceu a base para o seu desenvolvimento assim como o

envolvimento da APS (Associação Portuguesa de Surdos).

A relação entre características de design, fatores de imersão e desempenho de aprendizagem foram estudadas e demonstra que dar ênfase ao design de qualidade torna os utilizadores mais imersos, o que por sua vez é bom quando se joga um jogo e para obtenção de presença virtual. No entanto, estar presente num ambiente virtual não parece ser suficiente para obter resultados mais complexos de aprendizagem [Schrader05] pois utiliza demasiados recursos cognitivos. O nosso jogo foca-se na aprendizagem de informação complexa, uma nova língua. Portanto, o design não poderia ser demasiado imersivo mas também não poderia denegrir a experiência, o que foi conseguido através de mecânicas de jogabilidade e narrativa. O ensino da língua gestual tem de ser feito passo a passo, e quando tentamos integrar aprendizagem no contexto de jogo pode tornar-se enganador ou disperso, portanto o jogador tem de ser guiado ao longo do jogo.

Além disso, está provado que missões ou "quests" oferecem ao utilizador mais responsabilidade, motivando-o e realçando o fator de persistência através de compromissos sociais, aprendizagem passiva através de participação activa [Chen06].

Outros trabalhos que utilizam agentes inteligentes relacionados com a língua gestual têm tendência a focar-se na criação de avatares para comunicar com o utilizador [Lee09][Kipp10][Andrei11][Bellotti03][Hunicke04], ou sistemas com câmaras que interpretam os gestos [Lee09][Kipp10]. Reunimos com uma Professora da Associação Portuguesa de Surdos (APS) que nos mostrou um avatar que tentava comunicar utilizando língua gestual porém, apenas conseguia passar uma sensação de estranheza ao utilizador e a conversa não parecia natural, pelo que decidimos não utilizar câmaras nem avatares. Adicionalmente, a APS proporcionou-nos algum material com vídeos de palavras em língua gestual que eram utilizados com os estudantes portanto, decidimos utilizá-los diretamente no nosso jogo, visto que estes ensinariam o conteúdo exato pretendido sem ser necessário criarmos algo que o replicaria num estado menos conseguido.

4. ACERCA DO JOGO

O nosso público-alvo começou por ser composto por adolescentes no ensino secundário que gostariam de aprender LGP. Após reunirmos com a APS, o nosso público-alvo expandiu-se definitivamente; a Secretária Geral da APS mostrou-nos que seria rentável também incluir surdos na nossa demografia, independentemente da idade (desde que estes saibam ler), utilizando o nosso jogo como material de estudo. Assim, depois de alguma discussão sobre o assunto, concluímos que o nosso jogo tinha potencial para ser jogado por pessoas de todas as idades, desde que o seu objetivo fosse aprender língua gestual de uma forma mais interativa.

O intuito do nosso jogo, como já foi referido acima, é ensinar língua gestual de maneira que o utilizador não se sinta aborrecido com o processo de aprendizagem e que queira aprender o máximo possível. Temos um mecanismo de acompanhamento do utilizador para que este tenha

uma aprendizagem progressiva, utilizando missões que este tem de seguir e completar antes de poder prosseguir para uma nova, criando assim o ambiente de aprendizagem passo a passo. Por outras palavras, o utilizador apenas avança no jogo e explora novos lugares se completar os desafios propostos pelos NPC que vai encontrando pelo caminho.

No que diz respeito à inteligência artificial do jogo, os nossos objetivos foram o de criar um agente que consiga ajudar o utilizador a alcançar os objetivos propostos e de certa maneira criar uma ligação com ele. Dito isto, a interação entre o agente e o jogador sofre alterações à medida que o jogo avança, tendo em conta o desempenho deste.



Figura 1 - Exemplo de um desafio. O jogador escolhe a opção correta tendo em conta o desafio proposto pelo NPC.

5. FERRAMENTAS

À medida que começámos a conceptualizar o jogo, apercebemo-nos que precisaríamos de um motor de jogo eficiente e fácil de utilizar, assim como um modelador 3D. Quisemos desenvolver tudo de raiz, modelação e scripting. Para o scripting das mecânicas, estruturação de eventos, construção dos níveis e implementação da inteligência artificial optámos por utilizar Unity 3D[Unity17]. Para a modelação 3D optámos por utilizar Blender[Blender18].

6. DESIGN CONCEPTUAL DO GESTUAL LIFE

Um dos nossos principais objetivos era imergir o jogador completamente no jogo, portanto decidimos criar um jogo em 3D utilizando a visão em primeira pessoa. Criámos um mundo onde o jogador pudesse de facto encaixar sem se sentir constrangido ou deslocado da sua realidade, pelo que um grande conjunto de características foi adicionado ao ambiente de forma a torná-lo o mais credível possível aos olhos do jogador. Algumas das características do nosso jogo são:

- NPCs interativos capazes de comunicar com o jogador, tendo em conta o seu progresso;
- Uma grande oferta de desafios;
- Um mundo com objetos e locais para descobrir (ex: uma escola onde o jogador pode aprender vários âmbitos da língua gestual, a sua casa, casas na vizinhança).

A parte relativa ao ensino do material em questão está dividida em duas metades: o segmento da manhã e o segmento da tarde.

No segmento da manhã o jogador é conduzido para onde tem de se deslocar a seguir, o que terá de fazer e com quem terá de falar de modo a progredir e aprender como jogar. O jogador começa na sua casa onde tem de aprender alguns gestos básicos, resolver o desafio proposto pela sua mãe e depois proceder à exploração do bairro, não apenas resolvendo desafios e colecionando vídeos, mas também falar e conhecer cada um dos NPC presentes.

O segmento da tarde trata-se de uma parte de exploração com mundo aberto. Nesta altura o jogador já estará familiarizado com o mundo do Gestual Life e como interagir com este. É-lhe permitido vaguear livremente, resolver mais desafios, acumular mais conhecimentos ou simplesmente interagir com tudo o que o mundo tem para oferecer. Por esta altura a imersão pretendida foi conseguida e é esperado que o jogador procure conhecimento por si mesmo.

Quando o jogador participa num desafio ele só o pode terminar obtendo a resposta correta. Esta escolha de design foi feita de modo a que, independentemente do desempenho, o jogador saia do desafio sempre com a informação correta.

Era importante imergir o jogador no nosso jogo de modo a que ele possa (tanto quanto possível) sentir-se como uma pessoa surda num ambiente social. Como uma pessoa surda possui um sistema auditivo debilitado, não removemos o som do jogo por completo, limitámo-nos a abafar todos os sons. O jogador reconhece que de facto consegue ouvir algo mas não consegue perceber o quê. Isto será estranho para o jogador, proporcionando assim uma nova realidade, a realidade de uma pessoa surda.

Para ajudar o jogador necessitávamos de alguém, ou algo, para comunicar por ele, assim foi criada a L.Y.S.

L.Y.S. é a ajudante do jogador que se parece com um antigo monitor CRT(*Cathode Ray Tube*). É um ser inteligente que fala e ouve pelo jogador, guardando também todos os vídeos com os quais se dá uma interação, servindo como um inventário que o jogador pode aceder sempre que desejar.

Para criar os diferentes níveis de dificuldade baseámos os desafios nos cinco parâmetros da língua gestual:

- Configuração dos dedos, mãos e braços;
- Localização espacial do gesto;
- Orientação da mão;
- Movimentos;
- Expressão facial.



Figura 2 - Pré-visualização do bairro e da casa do protagonista

7. L.Y.S., A COMPANHEIRA

7.1 Design

Optámos por criar uma personagem 3D de seu nome L.Y.S. num formato de uma televisão para que, para além de apresentar expressões faciais simples relativamente ao desempenho do jogador, pudesse reproduzir os vídeos das palavras representadas em língua gestual. O nosso objetivo foi o de a tornar uma personagem caricaturada, de modo a criar uma ligação entre ela e o utilizador. Quanto ao mundo, também o criámos sem demasiado realismo de maneira a que o utilizador também pudesse estabelecer uma conexão.

A L.Y.S. apresenta diferentes cores consoante o seu estado de espírito sendo elas: verde, quando o utilizador se está a sair bem nos desafios, vermelho, quando o utilizador tem alguma dificuldade em responder corretamente às questões e azul, que é a cor normal da L.Y.S. para quando está a dar sugestões ou simplesmente a acompanhar o jogador.

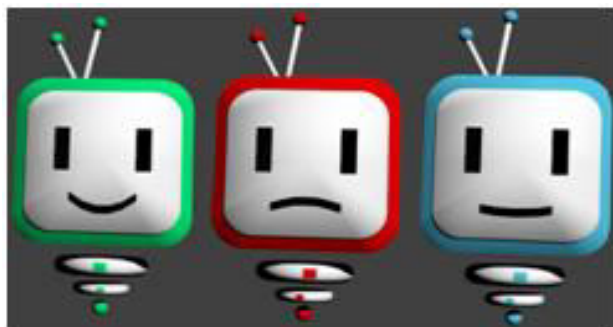


Figura 3 - Expressões faciais da L.Y.S.

7.2 Implementação da L.Y.S.

A inteligência e a interatividade da L.Y.S. são baseadas em dois pontos fulcrais: discurso interativo e um sistema de pontuação que altera a dificuldade dos desafios seguintes, ambos baseados no desempenho do jogador.

O primeiro ponto foi implementado utilizando pontos-chave após cada desafio. Quando a ajuda da L.Y.S. é requisitada, esta diz frases ao jogador relacionadas com o próximo desafio ou destino. Se o jogador demorar muito tempo a percorrer o mundo do jogo, a L.Y.S. intervém e diz-lhe explicitamente para onde este se deverá dirigir a seguir, de maneira condescendente, dando-lhe alguma personalidade. É bastante importante que o jogador sinta que a L.Y.S. é uma amiga ou companheira, ao invés de

uma personagem maçadora que lhe diga sempre o que fazer.

A L.Y.S. altera a pontuação do jogador (aumentando-a e diminuindo-a) baseando-se nas respostas aos desafios. Contudo, o jogador não tem acesso a esta pontuação, apenas o feedback relativo ao seu desempenho. O propósito disto encontra-se em adaptar a dificuldade dos desafios para que o jogador não se sinta demasiado frustrado. A pontuação varia numa escala entre zero e trezentos sendo que o intervalo de zero a cento e cinquenta pontos é o modo Fácil, cento e cinquenta a duzentos e cinquenta é o modo Médio e os restantes cinquenta pontos são para o modo Difícil. Os desafios são dados por cada NPC e as opções para os resolver são predeterminadas dado a complexidade da língua gestual.

No modo Fácil, as opções são completamente diferentes, tendo em conta os parâmetros da língua gestual. No modo Médio, as opções são similares, mas diferem em pelo menos dois dos parâmetros. No modo Difícil, apenas um parâmetro é diferente em cada uma das opções.

De maneira a calcular a pontuação, criámos uma função simples que tem em consideração seis variáveis: a pontuação atual (que será alterada), os pontos máximos que cada desafio pode oferecer, dois multiplicadores X e Y que diferem em cada modo (tiram mais pontos se o utilizador falha mais vezes), o tempo demorado e o número de tentativas utilizados para resolver o desafio.

$$\text{score} = \text{score} + (\text{MaximumChallengePoints} - (X * \text{TimeTaken} + Y * \text{NumberOfTries}))$$

Figura 4 - Função que calcula a pontuação do jogador

8. TESTES

De maneira a testar o *input* dos utilizadores e a jogabilidade, considerámos 80 utilizadores humanos, cujas idades diferiam entre os 15 e os 46 anos de idade, 7 surdos e 73 ouvintes. A esta população foi-lhes pedido que fizessem um conjunto de tarefas incluindo aprendizagem de gestos, resolução de desafios e exploração do mundo de jogo.

Devido ao calendário de desenvolvimento, para estes testes utilizámos uma demonstração do nosso jogo em que apenas estava incluído o segmento da manhã. Embora já partes do segmento da tarde se encontrassem desenvolvidas como o mundo e edifícios, o segmento da Manhã apresentava um ambiente mais controlado e importante para o resto do desenvolvimento do jogo.

Na primeira fase de testes, testámos 26 pessoas estudantes e professores da APS. Os seus resultados foram fundamentais para reparar alguns erros e adicionar novas funcionalidades para pessoas que não costumam ter muita interação com jogos.

Na segunda fase de testes, 34 pessoas foram testadas, sendo estes estudantes universitários de áreas variadas. Com o seu feedback adicionámos ainda mais funcionalidades, fazendo o jogo mais rápido em algumas partes em que estava a evoluir mais lentamente.

Na fase final de testes, testámos o nosso protótipo final com as restantes 20 pessoas da nossa bateria de teste também estes alunos da nossa instituição.

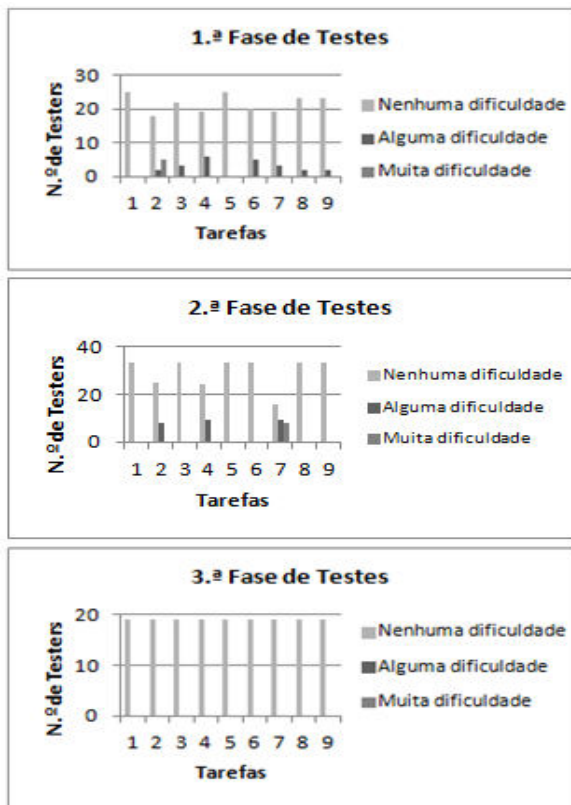


Figura 5 - Comparação de resultados da interação com as mecânicas do jogo entre as diferentes fases de testes

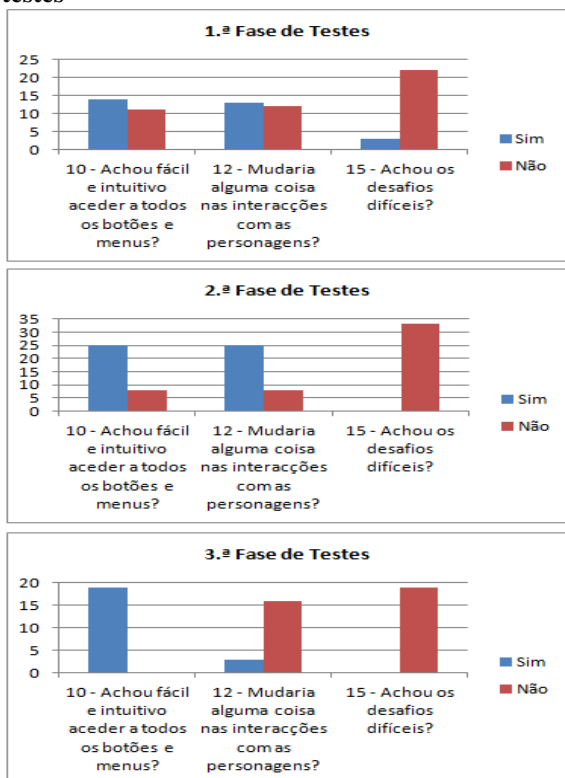


Figura 6 - Comparação de resultados da jogabilidade entre as diferentes fases de testes

9. RESULTADOS

Após testarmos o nosso jogo com 80 pessoas, apercebemo-nos que este, no que respeita ao segmento da manhã, está perto de estar finalizado. Os utilizadores gostaram de jogar o nosso jogo e interagiram muito bem com ele. Resolveram facilmente cada desafio após colecionarem os vídeos, falaram com cada NPC deslindando as suas personalidades e tentaram despoletar ações que temos planeadas para trabalho futuro, ou seja, os utilizadores de modo intuitivo procuraram utilizar mecânicas que ainda não se encontram implementadas, apenas definidas e conceptualizadas.

Em suma, o feedback dos utilizadores foi muito melhor que o esperado, sendo que os únicos problemas relevantes foram alguns erros que apareceram com utilizadores menos ambientados com jogos de computador como os professores na APS.

Os utilizadores tentavam imediatamente reproduzir os gestos que tinham acabado de aprender sem que nós o requerêssemos.

Nenhuns sinais de frustração foram registados enquanto os utilizadores interagiam com o mundo, tanto nos desafios como na parte de exploração.

Como é possível verificar nas figuras 5 e 6 na secção de Testes, os resultados sofreram alterações ao longo das fases de teste, de acordo com as modificações que o jogo foi sofrendo tendo em conta o feedback recebido por parte dos utilizadores e os resultados observados durante a interação com o jogo.

Na terceira fase de testes deu-se uma melhoria significativa nos níveis de dificuldade observados pelos utilizadores na resolução dos desafios.

Relativamente às figuras, na primeira fase de testes verificou-se que o jogo ainda não estaria intuitivo o suficiente em termos de acesso a menus e interações.

Devido à dificuldade inicial estabelecida, poucas pessoas acharam os desafios em si difíceis tendo todas elas suplantado os desafios propostos, de notar que nesta fase alguns dos utilizadores referiram ter de facto adquirido conhecimento novo com o jogo.

Após a primeira fase de testes, a principal preocupação foi tornar a jogabilidade mais intuitiva e acessível com base no feedback obtido, uma vez que os resultados demonstravam que a dificuldade do jogo em si estaria adequada, embora a jogabilidade tenha melhorado na segunda fase, algumas interações com NPCs ainda sofriam de problemas que foram corrigidos na terceira fase, sendo esta uma fase de otimização.

10. TRABALHO FUTURO

Reconhecemos que uma maior variedade de expressões faciais para a L.Y.S. providencia-a com uma maior credibilidade da sua inteligência e uma maior capacidade de se ligar com o utilizador, fazendo com que este fique mais atento às suas interações e dando-lhe a sensação que ela possui uma “alma”. Os níveis de dificuldade podem ser mais vastos e com mais complexidade sendo que só foram implementados os níveis fácil, médio e difícil.

Também temos de aumentar o espectro de frases para a interação entre o utilizador e a L.Y.S. para que as “conversas” tenham menos repetições e sejam mais naturais a ajudar.

O mundo disponível para o utilizador explorar pode ser alargado, dando-lhe mais escolha onde ir e aprender novos gestos, criando também mais NPCs para interagir.

O segmento da tarde também pode ser mais desenvolvido, facultando ao jogador mais interações e ainda mais desafios. A escola à qual o utilizador tem acesso pode também ser melhorada, tanto na questão de design como em arquitetura.

Finalmente, bugs precisam de ser corrigidos como conflitos entre mecânicas controladas pelo jogador e mecânicas automatizadas (ex: dicas da LYS e inventário) e algumas destas melhoradas, como interações com NPCs, movimento afinado e triggers de eventos mais precisos.

11. CONCLUSÕES

Desenhar, desenvolver e testar este jogo foi uma experiência tanto desafiante como gratificante. Os resultados demonstram que fomos bem-sucedidos quer nos nossos objetivos, quer na jogabilidade e na aprendizagem. Os utilizadores conseguiram criar uma ligação com a personagem emocionalmente e aprenderam o que a L.Y.S. lhes tinha para ensinar, já que requeriam muitas vezes as suas interações. O nosso método de associar objetos e palavras a vídeos provou ser frutífero e demonstrou grandes resultados no que toca à retenção de novo conhecimento que o utilizador adquiriu.

12. AGRADECIMENTOS

Gostaríamos de agradecer à Associação Portuguesa de Surdos, especialmente à sua Secretária Geral Mariana Martins, pelo seu apoio neste projeto e por todos os materiais providenciados. Após ver a nossa implementação e partilhar a sua experiência a ensinar língua gestual, aprovou o nosso método e também o conteúdo.

13. REFERÊNCIAS

[Andrei11] Stefan Andrei, Lawrence Osborne, Zanthia Smith. Designing an American Sign Language Avatar for Learning Computer Science Concepts for Deaf or Hard-of-Hearing Students and Deaf Interpreters. *Journal of Educational Multimedia*, Number 3, July 2013.

[Bellotti07] Francesco Bellotti, Michela Ott, Sylvester Arnab, Riccardo Berta, Sara de Freitas, Kristian Kiili and Alessandro De Gloria (January 2011). Designing Serious Games for Education: From Pedagogical Principles to Game Mechanisms. *Proceedings of the 5th European Conference on Games Based Learning*, 2011.

[Bellotti03] Francesco Bellotti, Bill Kapralos, Kiju Lee, Pablo Moreno-Ger, and Riccardo Berta(2013). Assessment in and of Serious Games: An Overview, *Advances in Human-Computer Interaction Volume 2013*, Article ID 136864, 11 pages.

[Blender18] Blender(Version 2.70a) [Computer software]. Retrieved January 12, 2014. Available from <http://www.blender.org/>

[Chen06] Chen, Z.-H., Liao, C. C. Y., Cheng, H. N. H., Yeh, C. Y. C., & Chan, T.-W. (2012). Influence of Game Quests on Pupils' Enjoyment and Goal-pursuing in Math Learning. *Educational Technology & Society*, 15 (2), 317–327.

[Cheng08] Irene Cheng, Saul Rodriguez and Anup Basu. Multimedia and Games incorporating Student Modeling for Education. *International Workshop on Technology for Education*, 2009.

[DeCoito14] DeCoito, I., DiGiuseppe, M., & Friedberg, J. The History of Biology: Examining a Digital Game for Improving Students' Nature of Science Conceptions and Promoting Student Engagement in Biology. *Proceedings of The Future of Education International Conference 2012 (Vol. 1)*, June 7-8, Florence, Italy. Simonelli Publisher University Press.

[Hunicke04] Robin Hunicke, Marc LeBlanc, Robert Zubek (2007). MDA: A Formal Approach to Game Design and Game Research, *Discovery Magazine* 2007.

[Hwang01] Gwo-Jen Hwang, Han-Yu Sung, Chun-Ming Hung, Li-Hsueh Yang and Iwen Huang (2013). A knowledge engineering approach to developing educational computer games for improving students' differentiating knowledge. *British Journal of Educational Technology* Vol 44 No 2, 183–196, 2013.

[Kebritchi13] Mansureh Kebritchi, Ph.D., Atsusi Hirumi, Ph.D. and Haiyan Bai, Ph.D. The Effects of Modern Math Computer Games on Learners' Math Achievement and Math Course Motivation in a Public High School Setting. *Journal Computers & Education*, Volume 55 Issue 2, Pages 427-443 Elsevier Science Ltd. Oxford, UK, September, 2010

[Kipp10] Michael Kipp, Quan Nguyen, Alexis Heloir Silke Matthes. Assessing the Deaf User Perspective on Sign Language Avatars ASSETS '11 *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility* 107-114 ACM New York, 2011.

[Lee09] Seungyon Lee, Valerie Henderson, Helene Bra-shear, Thad Starner, Steven Hamilton and Harley Hamilton. User-Centered Development of a Gesture-Based American Sign Language Game. *Instructional Technology and Education of the Deaf Symposium*. National Technical Institute for the Deaf Rochester, NY 14623 USA June 27-30, 2005.

[Lisztes12] Ákos Lisztes, Ákos Antal, Andor Gaudia and Péter Korondi. Sign Language in the Intelligent Sensory Environment. *Acta Polytechnica Hungarica* Vol. 2, No. 1, 2005.

[Parton16] Sue Parton, Becky. Sign Language Recognition and Translation: A Multidisciplinary Approach From the Field of Artificial Intelligence. *Journal of Deaf Studies & Deaf Education*; Winter 2006, Vol. 11 Issue 1, p94. January 2006

[Schrader05] Claudia Schrader, Theo J. Bastiaens (2012). The influence of virtual presence: Effects on experienced cognitive load and learning outcomes in educa-

tional computer games. *Computers in Human Behavior* 28, 648–658, 2012.

[Unity17] Unity 3D (Version4.5.1f3) [Computer software] Retrieved February 20, 2014. Available from <http://unity3d.com/>

[Weiß02] Sebastian A. Weiß, Wolfgang Müller (2008). The Potential of Interactive Digital Storytelling for the Creation of Educational Computer Games. Technologies

for E-Learning and Digital Entertainment, *Third International Conference, Edutainment 2008*, Nanjing, China, June 25-27, 2008, Proceedings Source: DBLP

[Zafrulla15] Zafrulla, Z., Brashear, H. ; Pei Yin ; Presti, P. ; Starner, T. ; Hamilton, H. American Sign Language Phrase Verification in an Educational Game for Deaf Children. *20th International Conference on Pattern Recognition (ICPR)*, Istanbul, 2010

Personagens Virtuais Expressivas

Renato Teixeira

Ana Paula Cláudio

Maria Beatriz Carmo

LabMAg, Faculdade de Ciências, Universidade de Lisboa

1749-016 Lisboa, Portugal

fc37057@alunos.fc.ul.pt, apclaudio@fc.ul.pt, mbcarmo@fc.ul.pt

Augusta Gaspar

CIS- Centro de Investigação e Intervenção Social, ISCTE Instituto Universitário de Lisboa,

1649-026 Lisboa, Portugal

augusta.gaspar@iscte.pt

Resumo

A Ansiedade Social afeta um número significativo de pessoas, limitando-as na vida pessoal e social, e apresentando um nível de comorbilidade elevado com a depressão. Neste artigo descrevemos um projeto que aplica Realidade Virtual, interativa, para a terapia de exposição da Ansiedade Social, recorrendo a Personagens Virtuais exibindo combinações de expressões faciais e corporais. A aplicação descrita neste artigo atualiza e melhora significativamente uma versão anterior, sobretudo no que respeita à qualidade gráfica dos personagens virtuais; é executada em computadores vulgares, sem equipamentos dispendiosos, sendo facilmente utilizada em contextos de aconselhamento e investigação.

A possibilidade de controlar de forma minuciosa as expressões faciais permite o seu uso em contextos de investigação do comportamento não-verbal. Embora se tenha recorrido apenas a modelos gratuitos ou de muito baixo custo, adotaram-se estratégias que permitiram obter a qualidade adequada, facto que conseguimos validar através de testes com um conjunto significativo de observadores.

Keywords

Personagens Virtuais; Fobia Social; Expressões Faciais; Terapia de Exposição; Comunicação não-verbal.

1. INTRODUÇÃO

A fobia social ou Transtorno de Ansiedade Social (TAS) é uma condição caracterizada por intensa ansiedade quando o indivíduo é confrontado com uma situação de exposição pública ou mesmo a antecipação de uma tal situação [APA00]. Esta condição limita extraordinariamente a vida pessoal e social, apresentando-se com uma comorbilidade alta com a depressão [Stein00]. As pessoas que sofrem de TAS temem julgamentos negativos, e acreditam ser julgadas deste modo; a sua perceção de sinais emocionais é semelhante à de outras pessoas quando se tratam de sinais neutros ou característicos de afetos positivos. Mas, são detetores significativamente mais rápidos e eficazes em relação a sinais que podem veicular conteúdo ameaçador ou negativo [Douilliez12] evidenciando assim a sua hipervigilância para a ameaça social.

As terapias usadas para tratar TAS incluem geralmente medicamentos (principalmente anti-depressivos), terapia cognitivo-comportamental, psicoterapia e técnicas de relaxamento. De entre estas, a terapia cognitivo-comportamental é a que mostrou os resultados mais eficientes e duradouros, especialmente quando é aplicada na forma de terapia de exposição [Beidel07], que consiste em expor o paciente à situação que lhe causa a fobia.

A Realidade Virtual (RV) tem vindo a ser usada na terapia de exposição desde o início dos anos 90, designando-se por Realidade Virtual na Terapia de Exposição (RVTE), tendo alguns estudos concluído que produz resultados semelhantes aos da exposição tradicional [Klinger04; Herbelin05]. A RVTE permite um controlo preciso do processo de habituação (e extinção) de medo a um objeto fóbico e, portanto, oferece muitas vantagens adicionais sobre a terapia de exposição clássica, que envolve imagens e contacto posterior com situações da vida real: a) permite a parametrização de cenários e interações para atender às necessidades e progresso de diferentes pacientes, e às de cada paciente durante todo o período de tratamento; b) permite uma otimização da preparação do paciente antes de enfrentar a realidade, evitando assim o risco de uma exposição prematura a uma situação real; c) reduz o risco de retrocessos e reações excessivas e proporciona um ambiente mais estável e progressivo, garantindo um resultado previsível e consolidado; d) e finalmente, protege a privacidade dos pacientes. As principais desvantagens típicas da RVTE são o custo financeiro de aquisição de equipamentos virtuais imersivos (*Head-Mounted Displays*, CAVE) e os efeitos secundários reportados por alguns utilizadores (*cybersickness*) [LaViola00].

Este artigo descreve uma abordagem ao uso da RV no tratamento da Ansiedade Social, em particular, em situações de medo de falar em público na frente de um júri numa situação de avaliação. O paciente enfrenta um júri composto por um a três Personagens Virtuais que exibem expressões faciais e corporais controláveis em tempo real pelo terapeuta. Deste modo, o terapeuta controla o nível de *stress* induzido ao paciente durante a sessão de terapia. O controlo é obtido através de uma interface que (entre outras funções) desencadeia nos personagens um conjunto de movimentos faciais e posturas do tronco que permitem uma vasta gama de composições que podem transmitir conteúdo emocional neutro, positivo ou negativo, bem como simular atenção ou falta de interesse. Antes de uma sessão, o terapeuta configura o cenário: escolhe o aspeto da sala em que encontra o júri, as personagens e a respetiva posição na mesa, as suas roupas e o penteado.

Além de se destinar ao tipo de terapia descrita, a aplicação desenvolvida é uma ferramenta útil na investigação do comportamento não-verbal: o efeito da comunicação através do comportamento facial e de outros elementos ou atitudes corporais não é totalmente conhecido, apesar de ser comumente apresentado como tal e é objeto atualmente de intenso debate e diversos estudos [Gaspar12; Gaspar14].

A aplicação permite visualizar individualmente o rosto de cada personagem e compor diferentes constelações de unidades de comportamento facial e corporal, controlando o efeito da mudança de uma única ação individualizada no rosto, enquanto outras ações são mantidas constantes. Avaliar o impacto destas combinações sobre os observadores pode clarificar o papel comunicativo das ações unitárias ou compostas, e auxiliar na definição de parâmetros objetivos para o aconselhamento em tratamentos de ansiedade causada por exposição social, bem como para o avanço da investigação em comportamento não-verbal.

A imagem do cenário deve ser projetada numa tela ou parede, de modo que permita simular o tamanho real das personagens, criando assim um efeito imersivo.

O trabalho aqui descrito é uma evolução de trabalho anterior da mesma equipa. Uma primeira versão da aplicação recria um auditório de personagens virtuais com comportamentos controláveis mas com pouca expressão facial [Cláudio13]; uma segunda versão implementa já uma simulação de um júri com um conjunto de personagens virtuais com expressões faciais e corporais controláveis pelo terapeuta em tempo-real [Cláudio14]. O desenvolvimento de todas as versões tem sido sempre acompanhado de perto por uma psicóloga, que tem tido um papel fundamental na identificação das expressões dos personagens virtuais que são potencialmente indutores de retorno positivo, negativo ou neutro sobre os observadores.

Este artigo debruça-se sobre uma etapa em que se procedeu a melhoramentos na interface do terapeuta, se

investiu na utilização de modelos virtuais de melhor qualidade gráfica, simulando idades variadas e visuais distintos, e em que se efetuou um estudo de validação preliminar destes modelos e das suas expressões faciais, confirmando os pressupostos tidos em conta pela psicóloga da equipa na elaboração do seu menu de ações faciais.

A aplicação recorre a ferramentas de software gratuitas ou a licenças temporárias de produtos comerciais e a modelos virtuais gratuitos ou de baixo custo. Pode ser executada em computadores vulgares, e não recorre a equipamentos de RV imersivos, razões pelas quais pode ser facilmente utilizada no aconselhamento e na investigação.

Este artigo está organizado da seguinte forma: a próxima secção apresenta alguns dos trabalhos relacionados mais relevantes; na secção 3 descrevemos a nossa aplicação; a secção 4 é dedicada à validação do comportamento expressivo da face dos personagens virtuais e finalmente, a secção 5 é dedicada às conclusões e trabalho futuro.

2. TRABALHO RELACIONADO

A primeira aplicação de Realidade Virtual (RV) para tratar o medo de falar em público foi descrita em [North98]. Esta aplicação incluía um cenário de um auditório com até 100 personagens e durante uma sessão de terapia, o terapeuta podia variar o número de personagens e as suas atitudes, utilizando sequências de vídeo pré-gravadas. O paciente utilizava um HMD, ouvindo o eco de sua própria voz. Slater et al. criaram uma sala de seminários virtual com 8 personagens exibindo comportamentos autónomos aleatórios, como balançar a cabeça e piscar os olhos [Slater99]. O estudo inicial realizado com 10 alunos com diferentes níveis de dificuldade de falar em público foi alargado a um conjunto mais amplo de pessoas, incluindo indivíduos fóbicos e não-fóbicos [Pertaub01; Pertaub02; Slater06].

James et al. propuseram um cenário duplo: uma carruagem de metro na qual os personagens expressam comportamentos neutros, o que se configura um ambiente não exigente do ponto de vista de interação social e uma situação com nível de exigência diametralmente oposto, um bar em que os personagens pareciam muito desinteressados [James03]. Os comportamentos dos personagens incluíam movimentação do olhar e frases pré-gravadas.

Klinger et al. conduziram um estudo em que se analisou as mudanças no medo de falar em público em 36 participantes ao longo de 12 sessões [Klinger04]. Para recriar os personagens virtuais usaram imagens de pessoas reais em situações quotidianas. Os participantes foram divididos em dois grupos: um tratado com terapia cognitiva comportamental tradicional e outro com RVTE, tendo-se verificado uma maior redução da ansiedade social no grupo tratado com RVTE. Herbelin publicou um estudo de validação com 200 pacientes, confirmando que a sua plataforma de RV preenchia os requisitos da exposição terapêutica para a fobia social [Herbelin05].

Além disso, provou que é possível melhorar a avaliação clínica com ferramentas de monitorização integradas, como o seguimento do olhar (*eye-tracking*).

Todas as abordagens referidas recorrem a equipamentos do tipo HMD; no estudo descrito por Pertaub et al. metade dos pacientes experimentaram um dos ambientes virtuais usando um HMD enquanto o restante grupo usou um computador de secretária [Pertaub02]. Herbelin e Grillon, além de recorrerem a HMD e a um ecrã de computador simples também usaram uma grande superfície de retro-projeção [Herbelin05; Grillon09].

Haworth e outros autores implementaram cenários virtuais para serem visualizados simultaneamente por pacientes e pelo terapeuta em ecrãs de computador, possivelmente em diferentes locais físicos e comunicação via internet [Haworth12]. Os cenários destinam-se a pacientes com acrofobia (medo das alturas) ou aracnofobia (medo de aranhas). A Kinect¹ é usada para controlar os movimentos do corpo do paciente. Os ainda poucos resultados deste estudo parecem provar que este tipo de solução de baixo custo é eficaz para estas situações fóbicas particulares.

3. VIRTUAL SPECTATORS

O trabalho descrito neste artigo é relativo à aplicação Virtual Spectators que tem vindo a ser desenvolvida pela nossa equipa que inclui uma psicóloga. A situação fóbica escolhida é o medo de falar em público, recorrendo-se à Realidade Virtual no uso da terapia de exposição para o tratamento desta situação fóbica.

No Virtual Spectators, o cenário virtual é, durante uma sessão de terapia, o palco de uma simulação controlada pelo terapeuta e observada pelo paciente- indivíduo que apresenta elevados níveis de ansiedade quando confrontado com uma situação em que tem de falar perante outras pessoas. O paciente, enquanto efetua o seu discurso perante os personagens virtuais, recebe os estímulos provenientes da simulação que observa; o terapeuta, atento ao comportamento e às respostas do paciente aos estímulos, interage com a aplicação de modo a afetar o decurso da simulação em conformidade, quer alterando o comportamento dos personagens, quer desencadeando eventos específicos no cenário.

A instalação adotada para usar a aplicação é composta por um computador, duas colunas de som, um projetor e uma tela ou uma parede como superfície de projeção. A aplicação cria duas janelas disjuntas: a da simulação, que deve ser projetada, e a da interface, no computador do terapeuta. As colunas de som devem estar junto à projeção para aumentar o grau de realismo da simulação.

Assim, temos as vantagens de o equipamento ser relativamente barato e fácil de instalar, de o tamanho da imagem projetada na tela ser de grandes dimensões, o que facilita a sensação de imersão sentida pelo paciente, e de

haver a possibilidade de várias pessoas assistirem em simultâneo a uma sessão de terapia, o que pode ser valioso, por exemplo, em ambientes de ensino. Além disto, eliminam-se os eventuais efeitos secundários desagradáveis que podem advir da utilização de equipamentos de RV, como vertigens e náuseas.

A versão aqui apresentada segue-se a duas outras: a situação simulada na primeira versão é uma audiência com humanos virtuais [Cláudio13] e, na segunda versão, um júri composto por um a três personagens virtuais [Cláudio14].

3.1 Virtual Spectators - versão atual

O principal objectivo desta etapa de desenvolvimento do Virtual Spectator era o de dotar as personagens de maior realismo e consequentemente de maior credibilidade. Tal objectivo teria que ser possível sem comprometer o desempenho da aplicação, a qual deveria continuar a suportar uma resposta em tempo real à interação do terapeuta. Foi assim necessário encontrar um equilíbrio entre o aspeto final das personagens e os aspetos críticos para a obtenção da resposta em tempo-real, nomeadamente, o detalhe da malha poligonal, o nível de resolução das texturas e os métodos de iluminação utilizados.

Numa primeira instância, constatou-se que não seria possível melhorar quaisquer das personagens das versões anteriores pelo se decidiu recorrer a novas personagens. Rapidamente se concluiu, depois de um conjunto de testes, que os modelos obtidos por recurso a ferramentas gratuitas, por exemplo, MakeHuman² ou Mixamo Fuse³, não satisfaziam o nível de qualidade pretendido. Levou-se então a cabo uma tarefa de combinação de modelos ou partes de modelos obtidos no software Poser⁴ e em repositórios *online*. Deste modo produziram-se duas personagens base do género masculino designadas por John e Carl, e uma personagem feminina, Jessi. Os três personagens aparentam idades distintas.

Procedeu-se à animação destas personagens tendo em conta as características/funcionalidades das principais ferramentas de software envolvidas na concretização da aplicação: o Blender⁵ e o Unity 3D⁶ na sua versão gratuita. Na ferramenta Unity as animações são conseguidas à custa de *rigging animation*, ou seja, pela animação de um esqueleto, uma estrutura hierárquica de ossos ligados entre si. A animação compreende ainda o processo de *skinning*: a associação de um osso a um determinado conjunto de vértices vizinhos. Quando o osso se movimenta, os vértices que lhe estão associados deslocam-se; os vértices mais próximos do osso sofrem maior deslocamento do que os vértices mais afastados.

² <http://www.makehuman.org> (último acesso set 2014)

³ <http://www.mixamo.com/fuse> (último acesso set 2014)

⁴ <http://poser.smithmicro.com/> (último acesso set 2014)

⁵ <http://www.blender.org/> (último acesso set 2014)

⁶ <http://Unity.com/> (último acesso set 2014)

¹ www.microsoft.com/en-us/kinectforwindows/ (último acesso set 2014)

No Blender a associação osso-vértices e os pesos que definem a amplitude do deslocamento é realizada usando a funcionalidade *weight painting*.

De modo a aproveitar tanto quanto possível as animações utilizadas na versão anterior, procedeu-se à integração do esqueleto das antigas personagens usadas pela aplicação nas novas personagens. Contudo, a posição base destes esqueletos não coincide com a posição base da malha poligonal das novas personagens, o que obrigou a realizar um ajustamento manual. Este ajustamento leva à modificação da posição base do esqueleto que resulta no desfaseamento das animações que dependem do(s) osso(s) afectados pelo que cada animação teve também de ser reajustada. Relativamente ao *skinning*, este é realizado de raiz, tanto para o rosto como para o corpo. Dado que as três personagens são anatomicamente diferentes entre si, este processo exaustivo é repetido para cada uma das personagens.

3.2 Expressões e Animações Faciais

A qualidade do modelo da face e as respetivas expressões são elementos fulcrais para a credibilidade da expressão das emoções. Com o propósito de obter estas características foram tomadas as seguintes opções:

- eliminar os ossos da cara não relevantes para as animações pretendidas (por exemplo, ossos associados ao nariz), ao mesmo tempo que se adicionaram outros em pontos específicos a fim de obter efeitos desejados em certas animações (por exemplo, franzir os cantos do olho);
- reduzir o número de polígonos do corpo, roupa e cabelo de cada personagem para poder aumentar, sem perda de desempenho, o número de polígonos da face;
- recorrer a um *shader* adquirido no próprio *site* do Unity para obter melhores resultados gráficos nos modelos.

Como resultado final, verificou-se uma melhoria na aparência das personagens virtuais, tornando possível e interessante o estudo de validação das expressões faciais que realizámos subsequentemente e que se encontra descrito na secção 4. A figura 1 exhibe o personagem John com duas expressões faciais distintas.

3.3 Interface de Configuração

Nesta etapa foram também incorporadas novas funcionalidades na aplicação, o que teve reflexos na interface de configuração do cenário.

As novas funcionalidades implementadas são:

- A possibilidade de alternar entre diferentes cenários. A aplicação está preparada para facilmente incluir outros cenários que se considerem relevantes.
- Um sistema de escolha de personagens em que as personagens estão todas visíveis para o utilizador. O processo de escolha (ou troca) de personagens usa uma interação simples do tipo *drag-and-drop*: o utilizador seleciona a fotografia que representa a personagem escolhida e arrasta-a para a posição que esta ocupará durante a simulação na mesa do cenário virtual; a fotografia passa a ser representada em tons de cinzento

como indicação de que aquela personagem não se encontra mais disponível para escolha;

- Escolher óculos: selecionada uma personagem que figure no cenário virtual é dada a opção de poder utilizar óculos tradicionais ou óculos de sol.
- Escolher um estilo de roupa formal ou informal.
- Escolher um tipo de penteado formal ou informal (exceto na personagem Carl que é calvo). A escolha de cabelo é independente da escolha de roupa, o que permite diversas possibilidades de conjugação. A figura 2 mostra a personagem Jessi com duas aparências distintas.
- Pré-visualizar a personagem: o terapeuta tem a oportunidade de visualizar a personagem selecionada e personalizada com as escolhas realizadas. A aplicação permite visualizar um *close-up* da sua face.



Figura 1: Personagem John com duas expressões faciais distintas.



Figura 2: Personagem Jessi com dois tipos distintos de aparência graças à alteração do cabelo e da roupa.

3.4 A Interface das Expressões Faciais

A figura 3 mostra os três personagens exibindo expressões faciais variadas. A interface que permite ao terapeuta controlar as expressões faciais (figura 4) contém botões que correspondem a unidades de ação faciais ou Action Units (AU) do Facial Action Coding System (FACS) [Ekman02]; em dois AUs (AU4, AU12) existe uma opção para dois níveis de intensidade. Tem havido grande controvérsia em torno da suposição, sem o apoio de estudos de expressão facial espontânea, de que certas expressões faciais transmitem determinadas emoções (ver, por exemplo, [Russell97]). Por este motivo, as AU escolhidas são as mais validadas em estudos de comportamento e perceção [Gaspar12]. Estas AU podem ser combinadas e a sua inserção simultânea no

rosto do humano virtual é o que cria a sua expressão facial.



Figura 3 : Três personagens base exibindo expressões faciais variadas (da esquerda para a direita, John, Carl e Jessi).

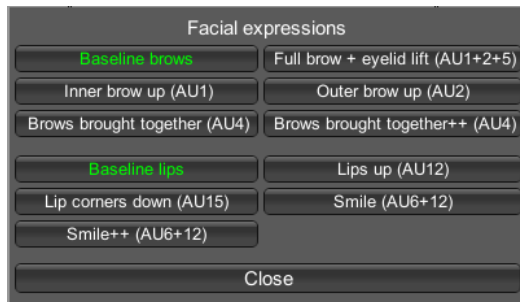


Figure 4. Interface de Controlo das Expressões Faciais. Os botões ficam verdes quando a respetiva expressão está ativa na personagem virtual.

Além disso, estas composições faciais podem também ser associadas com várias opções de postura corporal. As posturas e orientações foram escolhidas de acordo com resultados de conteúdos relevantes na comunicação não-verbal humana [Eibl- Eibbesfeldt89].

O painel de controlo da expressão facial (figura 4) inclui ações para transmitir afeto positivo: AU6+12, com dois níveis de intensidade, ativadas pelos botões "Smile (AU6+12)" e "Smile ++ (AU6+12)"; e só AU12, ativado por "Lips up (AU12)". Incluem-se igualmente ações muitas vezes associadas pelos observadores a retorno negativo, e associadas pelos próprios como atenção (AU4, AU5, AU2, AU1+2) ou afeto negativo (AU4, AU1+4 e AU15). O franzir das sobrancelhas formando rugas entre estas (AU4), produzido em humanos pela contração do corrugador, é particularmente relevante e, portanto, inseriram-se dois níveis de intensidade com os botões "Brows brought together (AU4)" e "Brows brought together (AU4)++". O menu também inclui dois botões para as posições neutras das sobrancelhas e da boca, que permitem obter um personagem quase inexpressivo.

A figura 5 mostra o mesmo modelo com diferentes combinações de AU. O rosto neutro, sem AU ativado (canto superior esquerdo) e uma expressão severa com cenho franzido, AU4 (canto superior direito). Na parte inferior direita, vemos uma combinação de AU4 e AU15,

ambos associados com retorno negativo. No canto inferior esquerdo, vemos uma combinação de AU4 e AU6+12 (o botão de "smile"). A combinação AU6+12 é a característica que define o "*Duchenne smile*", que é associado a afeto positivo, de forma mais honesta do que qualquer outro sorriso. O efeito ameaçador associado ao cenho franzido é mitigado e, possivelmente, este é associado a outro contexto frequente, a atenção.

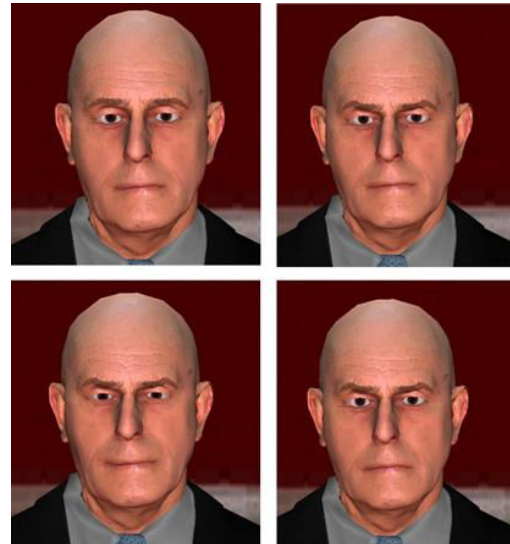


Figure 5- A partir do canto superior esquerdo e no sentido dos ponteiros do relógio: Carl com uma expressão neutra, com as sobrancelhas franzidas (AU4), mostrando uma combinação de AU4 e AU15, mostrando uma combinação de AU4 e AU6+12 (o botão "smile").

4. VALIDAÇÃO DO COMPORTAMENTO EXPRESSIVO DA FACE DOS HUMANOS VIRTUAIS

4.1 Participantes, materiais e procedimentos

Um estudo preliminar de validação dos pressupostos do comportamento facial dos Personagens Virtuais foi efetuado com uma amostra de 38 estudantes universitários (31F;7M), com idades compreendidas entre os 18 e os 25 anos, e que participaram voluntariamente como juízes das imagens após uma breve apresentação do projeto.

Para as experiências, os participantes foram reunidos em grupos de aproximadamente 10, instruídos sobre a tarefa, tendo-lhes sido fornecidas folhas de resposta, bem como formulário de consentimento informado. As experiências decorreram em sala semi-obscura, com a apresentação dos estímulos numa tela de projeção, encontrando-se os participantes em duas filas e a 2-3m de distância da tela, consoante a fila. Os participantes fizeram 3 ensaios iniciais das tarefas e esclareceram todas as dúvidas antes da experiência começar.



Figura 6. Tipos de ecrãs na experiência.

A experiência consistiu na apresentação de 28 *clips* animados de 3 segundos, contendo cada um grande plano de um dos avatares apresentando uma animação a partir da expressão neutra e evoluindo até uma outra expressão que corresponda a uma Unidade de Ação Facial (AU) ou a uma combinação de AUs. A lista de 14 AUs e combinações testadas foi a seguinte:

- Sobrancelhas em Baseline + Lábios em Baseline
- Aproximação das sobrancelhas AU4
- Aproximação (intensa) das sobrancelhas ++ (AU4)
- Levantamento das sobrancelhas + Levantamento da palpebral superior (AU1+2+5)
- Levantamento dos cantos internos das sobrancelhas (AU1)
- Levantamento dos cantos externos das sobrancelhas (AU2)
- Levantamento dos cantos dos lábios (AU12)
- Sorriso (AU6+12)
- Sorriso intenso ++(AU6+12)
- Abaixamento dos cantos dos lábios (AU15)
- Aproximação (intensa) das sobrancelhas ++ (AU4) + Levantamento dos cantos dos lábios (AU12)
- Aproximação (intensa) das sobrancelhas ++ (AU4)+ Abaixamento dos cantos dos lábios (AU15)
- Levantamento dos cantos internos das sobrancelhas (AU1)+ Levantamento dos cantos dos lábios (AU12)
- Levantamento dos cantos internos das sobrancelhas (AU1)+ Abaixamento dos cantos dos lábios (AU15)

Os *clips* foram apresentados por ordem aleatória. A apresentação de cada *clip* foi precedida por um ecrã de fixação com 1 segundo de duração e seguida da apresentação de escalas pictóricas de ativação, valência e dominância - o SAM (Self Assessment Manikin) [Bradley94] que estão internacional e amplamente validadas para avaliarem a percepção que os participantes têm relativamente ao modo como os estímulos os fazem sentir em termos de cada uma das dimensões inativo-ativo, emoção negativa-positiva e submissão – dominância. Os participantes dispuseram de 15 segundos para esta tarefa. Finalmente surgia um ecrã contendo rótulos com nomes de possíveis estados emocionais (contente, zangado, surpreendido, com medo, outro positivo, outro negativo ou neutro) expressados na face do personagem (Figura 6). Esta tarefa repetiu-se 28 vezes, pois cada combinação de AUs era replicada em 2 personagens virtuais. Os participantes foram encorajados a sugerir novos rótulos sempre que escolhessem as opções “Outro positivo” ou “Outro negativo”.

4.2 Resultados

No que concerne ao conteúdo atribuído pelos participantes às diferentes composições de unidades de ação facial, uma análise de X^2 de Independência mostrou existir associação altamente significativa entre as imagens e as frequências de conteúdos atribuídos ($X^2=2088,07$; $p<=.001$; $N=978$), sendo que 12 dos 28 *clips* obtiveram níveis de convergência acima dos 75%, todos eles representando associações significativas entre a imagem e o conteúdo ($p<=0,05$). As composições faciais com níveis maiores de concordância entre os vários observadores participantes foram as que o quadro 1, em anexo, ilustra. Este quadro mostra as imagens que obtiveram níveis de concordância mais elevados (acima dos 75%) na atribuição de conteúdo. As percentagens indicam a percentagem de participantes que atribuiu aquele conteúdo à imagem.

5. CONCLUSÕES E TRABALHO FUTURO

Os resultados observados na validação preliminar e situada exclusivamente na interpretação das expressões faciais são extremamente encorajadores: por um lado, as composições com elevadas concordâncias vão ao encontro dos pressupostos baseados nas observações de humanos, segundo as quais a interpretação universal tende a restringir-se a sinais de ameaça (como a aproximação de sobrancelhas) ou de disposição amigável/brincalhona (como o sorriso) e, por outro lado, os valores desta concordância são elevados relativamente aos reportados em estudos com humanos [Gaspar13; Gaspar14] o que aponta para as vantagens do controlo de todas as variáveis (fisionómicas, dimensão, luz, vestuário) com os personagens virtuais. É também de salientar como elemento importante nesta validação que os indivíduos identificaram corretamente a face neutra, um controlo importante dado que as imagens que não obtiveram elevadas concordâncias não o terão sido por outras razões que não o facto daquilo que comunicam depender de variáveis do próprio observador, sendo portanto ideais para comparar populações com características diferentes em termos de ansiedade social.

Pretende-se prosseguir com o desenvolvimento da aplicação com vista a um produto mais completo. Nomeadamente, através da concretização de um módulo de comportamentos baseados em personalidade com vista a facilitar o papel desempenhado pelo terapeuta de controlar o comportamento de cada personagem individualmente. Deste modo, este optaria por definir um tipo de personalidade para cada personagem que, por sua vez, adotaria durante a simulação, uma atitude

correspondente à escolha efetuada. Por exemplo, se uma personagem fosse escolhida como sendo “pouco amistosa”, passaria a adotar, automaticamente, comportamentos característicos (por exemplo, franzir sobrancelhas, postura corporal agressiva). Esta funcionalidade permitirá aumentar o número de personagens da cena sem sobrecarregar demasiado o terapeuta, contudo, sem nunca esquecer os aspetos relativos ao desempenho.

De momento ultimamos um módulo para prover as personagens de fala com o intuito de enriquecer a interação entre estas e o paciente.

Por fim, e acima de tudo, a validação da aplicação deverá compreender um processo de tratamento concreto de fobia social a fim de aferir quais as mais-valias da utilização desta aplicação quando comparada aos métodos tradicionais. Dado o interesse acentuado observado pelos terapeutas justificado, em parte, pelas limitações da tipicamente utilizada terapia de exposição *in imagino*, expectam-se resultados positivos e animadores e que justificam *de per se* a continuidade do projeto.













6. AGRADECIMENTOS

Agradecemos à Fundação para a Ciência e Tecnologia (FCT) e ao LabMAg o apoio financeiro ao abrigo do projeto PEst-OE/EEI/UI0434/2014.

7. REFERÊNCIAS

- [APA00] APA- American Psychiatric Association, 2000. DSM-IV-TR. American Psychiatric Publishing, Inc..
- [Beidel07] Beidel, D. C., Turner, S., 2007. *Shy Children, Phobic Adults: Nature and Treatment of Social Anxiety Disorder*. Washington, DC: APA, 2nd edition.
- [Bradley94] Bradley, M.M. and LANG, P.J., 1994. Measuring Emotion: the Self-Assessment Manikin and the Semantic Differential. *J. Behav. Therap. & Exp. Psvchyatry*, 25, 1, 49-59.
- [Cláudio13] Cláudio, A.P., Carmo, M.B., Pinheiro, T., Esteves, F. (2013), A Virtual Reality Solution to Handle Social Anxiety. *Int. J. of Creative Interfaces and Computer Graphics*, 4(2):57-72.
- [Cláudio14] Cláudio, A.P., Gaspar, A., Lopes, E., Carmo, M.B. (2014), Virtual Characters with Affective Facial Behavior. In Proc. GRAPP 2014, pp 348-355, SciTe-Press.
- [Douilliez12] Douilliez, C., et al., 2012. Social anxiety biases the evaluation of facial displays: Evidence from single face and multi-facial stimuli. *Cog & Emo*, 26(6): 1107-1115.
- [Eibl-Eibesfeldt89] Eibl-Eibesfeldt, I., 1989. *Human Ethology*. NY: Aldine de Gruyter.
- [Ekman02] Ekman, P, Friesen, WV., Hager, J C., 2002. *Facial action coding system*. Salt Lake City, UT: Research Nexus.
- [Gaspar12] Gaspar, A., Esteves, F., 2012. Preschoolers faces in spontaneous emotional contexts – how well do they match adult facial expression prototypes? *Int J Behav Dev*. 36(5), 348–357.
- [Gaspar14] Gaspar, A., Esteves, F., Arriaga, P. (2014) On prototypical facial expressions vs variation in facial behavior: lessons learned on the “visibility” of emotions from measuring facial actions in humans and apes. In M. Pina and N. Gontier (Eds), *The Evol. of Social Comm. in Primates: A Multidisciplinary Approach, Interdisciplinary Evolution Research*. NY: Springer.
- [Grillon09] Grillon, H., 2009. Simulating interactions with virtual characters for the treatment of social phobia. Doctoral dissertation, EPFL.
- [Haworth12] Haworth, M. B., Baljko, M., Faloutsos, P., 2012. PhoVR: a virtual reality system to treat phobias. In Proc. of the 11th ACM SIGGRAPH Int. Conf. VR Continuum and Applic in Industry, 171-174.
- [Herbelin05] Herbelin, B., 2005. Virtual reality exposure therapy for social phobia. Doctoral dissertation, EPFL.
- [James03] James, L. K., et al., 2003. Social anxiety in virtual environments: Results of a pilot study. *Cyberpsychology & behavior*, 6(3), 237-243.
- [Klinger05] Klinger, E., et al., 2005. Virtual reality therapy versus cognitive behavior therapy for social phobia: A preliminary controlled study. *Cyberpsychology & behavior*, 8(1), 76-88.
- [Klinger04] Klinger, E., et al., 2004. Virtual reality exposure in the treatment of social phobia. *Studies in health tech. and informatics*, 99, 91.
- [LaViola00] LaViola Jr, J. J., 2000. A discussion of cybersickness in virtual environments. *ACM SIGCHI Bulletin*, 32(1), 47-56.
- [North98] North, M. M., et al., 1998. Virtual reality therapy: an effective treatment for the fear of public speaking. *The Int. Journal of Virtual Reality* 3, 1-6.
- [Pertaub01] Pertaub, D. P., Slater, M., Barker, C., 2001. An experiment on fear of public speaking in virtual reality. *Studies in health tech. and inf.*, 372-378.
- [Pertaub02] Pertaub, D. P., Slater, M., Barker, C., 2002. An experiment on public speaking anxiety in response to three different types of virtual audience. *Presence: Teleoperators & Virtual Environments*, 11(1), 68-78.
- [Russell97] Russell, J. A., Fernandez-Dols, J. M., 1997. What does a facial expression mean? In *The psychology of facial expression*, pp. 3–30. New York, NY: Cambridge University Press.
- [Slater06] Slater, M., et al., 2006. A virtual reprise of the Stanley Milgram obedience experiments. *PloS one*, 1(1), e39
- [Slater99] Slater, M., Pertaub, D. P., Steed, A., 1999. Public speaking in virtual reality: Facing an audience of avatars. *IEEE CG&A*, 19(2), 6-9.
- [Stein00] Stein, M.B. and Kean, Y.M., 2000. Disability and Quality of Life in Social Phobia: Epidemiologic Findings. *Am J Psychiatry* 157: 1606–1613.

Quadro 1

		Branco (<i>não sei</i>)	Com Medo	Conten- te	Neutro	Outro	outro negativo: pena/tristeza	Outro +	Surpreen- dido	Zanga- do
	10 Jessi Sorriso intenso (AU6+12)++	5,7%	0,0%	91,4%	0,0%	0,0%	0,0%	2,9%	0,0%	0,0%
	12 Jessi Aproximação (intensa) das sobrancelhas (AU4) ++	11,4%	0,0%	0,0%	0,0%	5,7%	0,0%	0,0%	2,9%	80,0%
	16 Jessi Levant das sobrancelhas e abertura da pálpebra AU1+2+5	5,7%	11%	0,0%	0,0%	2,9%	0,0%	2,9%	77,1%	0,0%
	19 Jessi Aproxim. das sobrancelhas (AU4)	8,6%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	2,9%	88,6%
	22 Jessi sorriso (AU6+12)	5,7%	0,0%	80,0%	0,0%	5,7%	0,0%	5,7%	0,0%	2,9%
	24 Carl aproxim.intensa das sobrancelhas (AU4) ++ e abaixamentos dos cantos dos lábios (AU15)	5,7%	0,0%	0,0%	2,9%	5,7%	0,0%	0,0%	5,7%	80,0%
	28 Jessi Levantamento dos cantos dos lábios (AU12)	5,7%	0,0%	80,0%	0,0%	0,0%	0,0%	14,3%	0,0%	0,0%
	4 Jessi (AU4)	11,4%	0,0%	0,0%	2,9%	0,0%	0,0%	0,0%	2,9%	82,9%
	6 Jessi Baseline (sem expressão)	5,7%	2,9%	0,0%	85,7%	0,0%	0,0%	2,9%	2,9%	0,0%
	7 Carl Sorriso (AU6+12)	8,8%	0,0%	76,5%	5,9%	0,0%	0,0%	8,8%	0,0%	0,0%
	8 Jessi aproxim.intensa das sobrancelhas (AU4 ++) e abaixamentos dos cantos dos lábios (AU15)	8,6%	2,9%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	88,6%
	9 Carl Aproximação das sobrancelhas (AU4)	5,7%	0,0%	0,0%	11,4%	5,7%	0,0%	0,0%	0,0%	77,1%

Fluxo, um jogo pedagógico sobre o Sistema Circulatório

Inês Silva
ISCTE-IUL
Av. Forças Armadas Lisboa
ivsas@iscte-iul.pt

Filipe Baptista
ISCTE-IUL
Av. Forças Armadas Lisboa
fmsba@iscte-iul.pt

Jorge Calado
ISCTE-IUL
Av. Forças Armadas Lisboa
jmpsc@iscte-iul.pt

Pedro Faria Lopes
ISCTE-IUL, ADETTI-IUL, ISTAR-IUL
Av. Forças Armadas Lisboa
Pedro.Lopes@iscte.pt

Isabel Alexandre
ISCTE-IUL, IT-IUL
Av. Forças Armadas Lisboa
isabel.alexandre@iscte.pt

Resumo

Este artigo trata o desenvolvimento de um jogo educativo sobre o sistema circulatório. São abordados os temas-chave para a preparação do jogo, nomeadamente o que são jogos pedagógicos, o que são jogos side-scroller e os motivos que nos levaram a tomar determinadas decisões, tais como: a escolha do tema, a escolha das personagens, o design dos níveis, as mecânicas de jogo, etc. Será abordado também o uso de um agente inteligente que guia o jogador ao longo dos níveis e explica as mecânicas, as regras e a teoria relacionada com o tema. Pretende-se que os jogadores ganhem conhecimentos relativos ao percurso do sangue ao longo do corpo humano e que se aprenda quais são as funções dos vários componentes do sangue.

Palavras-chave

Jogos Pedagógicos, Sistema Circulatório, Biologia, Game Design, Side-scroller, Jogos 2D

1. INTRODUÇÃO

O Fluxo é um jogo pedagógico sobre o sistema circulatório do corpo humano. Ao longo do jogo, o jogador poderá ficar a conhecer alguns dos órgãos que fazem parte deste sistema, bem como outros componentes (e.g.: Hemácias). O objetivo é auxiliar no processo de aprendizagem do sistema circulatório e tem como público-alvo crianças com idades compreendidas entre os 8 e os 12 anos. Foi criado então, um jogo side-scroller 2D em que o jogador controla os elementos figurativos do sangue: glóbulos vermelhos (hemácias), glóbulos brancos (leucócitos) e plaquetas. O objetivo é fazer o percurso do sistema circulatório e, através das mecânicas de cada personagem jogável, perceber a função de cada elemento do sangue, bem como perceber a sua função nos diversos locais do corpo humano.

Nas secções seguintes analisaremos os pontos fulcrais por detrás da conceção do nosso jogo, bem como algumas decisões criativas relativamente ao aspeto visual das personagens, de todos os componentes do jogo e a definição das mecânicas de jogo.

2. TRABALHO RELACIONADO

Uma das principais fontes de inspiração para o desenvolvimento do jogo foi a série animada francesa “Era uma vez... O Corpo Humano” [Barillé87], criada por Albert Barillé. Nesta série, a ação de cada episódio aborda a funcionalidade de uma parte específica do corpo humano e os seus intervenientes. Os cenários e as personagens têm um estilo caricatural ao invés de serem foto-realistas, estando assim a série mais adaptada ao público infantil. Para além da série animada, existem outros materiais educativos que pretendem auxiliar a consolidação de conhecimentos, não só em relação ao corpo humano, como na área da Biologia em geral. Estes

materiais encontram-se em diversos tipos de média mas o mais interessante de analisar são os videojogos, os ditos jogos sérios (serious games).

Tomámos especial atenção à inteligência do agente no sistema. O artigo “Increasing believability in pedagogical agents” dos autores James C. Lester e Brian a. Stone sublinha que um agente credível condiciona a aprendizagem dos utilizadores [Lester97]. Quando as ações do agente são evidentemente automáticas e com condições simples, o jogador poderá rapidamente prever a ação seguinte, isto leva a que a credibilidade do agente inteligente diminua substancialmente. O artigo mostra uma solução para este problema através de um algoritmo de sequência de comportamentos. Decidimos, então, desenvolver um sistema semelhante ao dos autores para dinamizar as falas do agente inteligente ao longo do jogo. Tivemos em consideração o trabalho de Andy Nealen, Adam Saltsman, e Eddy Boxerman que explicam a importância do minimalismo em jogos no seu artigo [Nealen11]. Os autores mostram que podemos e devemos minimizar o conteúdo visual e mecânico de maneira a que o jogador se foque apenas no conteúdo essencial.

3. JOGOS PEDAGÓGICOS

Os jogos pedagógicos são jogos sérios e educativos que são explicitamente concebidos com um propósito de educar e passar conhecimentos a uma determinada área. Estes jogos podem ser jogos de tabuleiro, jogos de cartas ou videojogos, e é nestes últimos que nos focamos. Pretende-se que, ao jogar o “Fluxo”, o utilizador ganhe alguns conhecimentos sobre o funcionamento do corpo humano com a ajuda de um agente inteligente. Este irá guiá-lo ao longo dos níveis e ajudá-lo assim que for necessário. São focadas, ao longo do jogo, as várias funções de cada componente do sangue e mostra-se

também o percurso cíclico do sangue ao passar sistematicamente por locais importantes como o coração e os pulmões. Para a melhor compreensão de alguns eventos, introduzimos vídeos explicativos em são demonstradas as trocas gasosas dos órgãos, explicadas as funções dos componentes e atualizadas as posições no corpo humano.

4. DESIGN DO JOGO FLUXO

O Jogo Fluxo é um jogo de plataformas. Plataformas são superfícies em que as personagens podem andar ou correr. Nos jogos de plataformas 2D o jogador anda da esquerda para a direita e vice-versa, de forma a progredir no jogo. A personagem navega ao longo do sistema circulatório, pelas veias, artérias, capilares, etc., e sendo que o que delimita estes canais são células, estas são as plataformas do jogo. Escolhemos fazer um jogo deste género, pois as regras dos jogos side-scroller são facilmente compreensíveis, o ideal para crianças com idades compreendidas entre 8 e 12 anos. Para construir os níveis do jogo focámo-nos nos seguintes componentes: *background*, plataformas, obstáculos e colecionáveis.

Inspirados pelos livros e filmes da coleção “Era uma vez... O Corpo Humano”[Barillé87], focámo-nos em representar o tema de uma forma caricatural, onde os elementos do corpo humano têm personalidade e outras características humanoides, como por exemplo pernas e olhos. É importante um bom design de jogo para que se possa manter as crianças cativadas. Tentámos também, aumentar ao máximo o elemento surpresa, com várias personagens jogáveis ao longo do jogo, que introduzem novas mecânicas devido às novas funções destas personagens. Por outro lado foram construídos vários níveis com diferentes ambientes e cores para manter a diversidade e entusiasmo no jogo. Para que o jogo seja instrutivo decidimos eliminar tudo o que fosse acessório e mantivemos apenas o que é essencial para a informação que queremos passar. Para isso optámos por um design minimalista, onde apenas alguns elementos essenciais são representados. Por exemplo na Figura 1 podemos observar a comparação entre uma ilustração de uma célula de um livro escolar e uma célula do jogo Fluxo.

Um elemento muito importante no jogo é a cor. Todas as cores ao longo do jogo são alusivas ao órgão ou local do corpo que representam mas, por outro lado, estas tonalidades têm que combinar entre si, para evitar a incompatibilidade de estilo e design. No final obtivemos um resultado coeso, com cores relativamente claras, onde as personagens e alguns elementos do jogo têm algum destaque relativamente ao *background*.

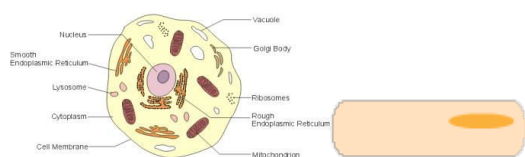


Figura 1. Comparação entre uma célula real e a sua representação no jogo

Para que o jogador se foque nas capacidades de cada componente do sangue decidimos dividir o jogo em três

ciclos com sequências de níveis diferentes, sendo que em cada ciclo é introduzida uma nova personagem e as suas funções no organismo. Os ciclos têm a seguinte sequência de níveis.

- Ciclo 1- Coração, Pulmões, Rins, Coração
- Ciclo 2-Coração, Pulmões, Capilares, Coração
- Ciclo 3-Coração, Pulmões, Intestino, Fígado, Coração

Em cada um destes ciclos é introduzido uma personagem jogável diferente.

5. PERSONAGENS

Foram criados dois tipos de personagens:

- Jogáveis: Glóbulo Vermelho (Hemácia), Glóbulo Branco (Leucócito) e Plaquetas (Figura 2)

- Não-Jogável: Agente Inteligente

Foram criadas estas personagens para que o jogador compreenda melhor a função de cada componente do sangue e estas podem ser trocadas usando a tecla “T”(troca). Deste modo, podem ser ultrapassados os desafios propostos ao longo dos níveis.

A principal função dos glóbulos vermelhos no sangue é transportar o oxigénio ou dióxido de carbono dos pulmões para as células e vice-versa o que nos obrigou a fazer uma representação destas hematoses, tanto no nível dos pulmões como nos capilares, através de vídeos explicativos. O glóbulo branco tem a principal função de proteger o organismo contra corpos estranhos, nomeadamente vírus e bactérias que entrem na corrente sanguínea. Para proteger o organismo, os glóbulos brancos utilizam um processo chamado “fagocitose” em que a bactéria é envolvida num vacúolo digestivo e eliminada. As plaquetas têm a função de coagular o sangue que é activada a partir da tecla “espaço”.

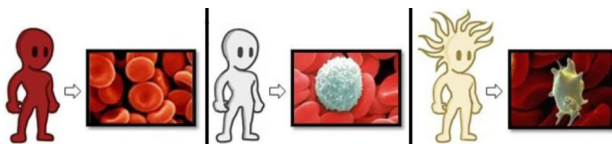


Figura 2. Comparação entre a representação visual dos elementos figurativos do sangue no jogo e na realidade (G. Vermelhos, G. Brancos, Plaquetas)

6. AGENTE INTELIGENTE

Devido ao facto da personagem principal ser um glóbulo vermelho, optámos pela representação do agente inteligente através da caracterização de um glóbulo vermelho mais envelhecido para que seja passado o conceito de sabedoria. Este agente foi o nosso maior foco quanto ao desenvolvimento da componente “inteligente” do jogo. Para que este agente inteligente fosse credível e demonstrasse alguma inteligência e personalidade focámo-nos na alteração do seu estado emocional, que é notório na forma como este se dirige à personagem controlada pelo jogador, face às suas ações. Para a implementação da personalidade do agente foi escolhida a abordagem de um algoritmo de fuzzy logic, uma das técnicas mais conhecidas na área da Inteligência Artificial. A lógica difusa é uma extensão da lógica booleana mas que rejeita o conceito de binário. Neste tipo de lógica lidamos com valores que não são unicamente falsos (e.g., 0) ou verdadeiros (e.g., 1),

havendo espaço para a incerteza (e.g., 0.5). Isto quer dizer que qualquer valor usado neste tipo de lógica difusa é um valor qualquer entre 0 e 1. A personalidade do agente é então definida a partir da leitura de um número entre 0.0 e 1.0 mas que é afetado constantemente ao longo do jogo pelas ações do utilizador que têm um determinado peso dependente da importância dessa ação e por um fator de uma soma ou subtração aleatória que torna o agente menos previsível. Quando o jogo se inicia, o agente tem uma personalidade neutra com o valor de 0.5. Os valores de personalidade assumidos estes subdividem-se em 5 estados emotivos possíveis que são distribuídos da seguinte maneira: As ações que podem afetar (positivamente ou negativamente) a personalidade do agente são:

- Cair várias vezes nas gorduras presentes nas artérias (Negativamente);
- Dirigir-se para o sentido contrário à corrente sanguínea (Negativamente);
- Ser várias vezes apanhado por vírus presentes na corrente sanguínea (Negativamente);
- Dirigir-se para a próxima área do corpo para onde o sangue deve ir (Positivamente);
- Oxigenar-se nos Pulmões (depende do número de bolhas de oxigénio colecionadas) (Positivamente).

Isto introduz alguma credibilidade e realismo, uma mesma ação poder afetar mais ou menos outra pessoa ao longo do tempo, bem como outros fatores que induzem alguma imprevisibilidade. A mudança de estado “emocional” do agente é notória cada vez que o jogador interagir com ele e despoletar um evento de diálogo, pois o seu discurso será mais ou menos elogioso, mais ou menos crítico, mais ou menos incisivo, conforme o seu estado emocional naquele instante. Os eventos de interação com o agente inteligente são transmitidos ao agente através da implementação de uma lógica de Triggers. Desta forma, apresentemos uma personagem não-jogável (NPC) que guia o jogador ao longo do jogo, explicando as diversas funções de cada parte do organismo, bem como os objetivos do jogador enquanto constituinte do sangue, mas com uma conotação mais ou menos positiva e de uma forma mais ou menos incisiva, conforme o desempenho do jogador.

7. TESTES E RESULTADOS

Para testar a eficácia do nosso sistema, perspetivámos testar o Fluxo com pessoas dentro e fora do nosso público-alvo. Devido ao processo de desenvolvimento ter terminado numa fase em que as escolas e outros locais onde poderíamos ter acesso a crianças com idades compreendidas entre os 8 e os 12 anos se encontrarem encerradas, foi-nos muito difícil encontrar um público para testes que correspondesse às características do público-alvo do nosso sistema. Deste modo, testámos o sistema com 15 pessoas (9 do sexo masculinos e 6 do sexo feminino) com idades entre 6 e 54 anos, com maior incidência entre 17 e 28 anos. Pedimos a cada pessoa que jogasse um dos ciclos do jogo Fluxo. Assim que terminaram, foi-lhes pedido que respondessem anonimamente a cinco perguntas sobre a sua experiência,

duas qualitativas e três quantitativas. As perguntas propostas foram as seguintes:

- Avaliação Qualitativa
 - 1) O que é que gostou mais no jogo?
 - 2) O que acha que podia ser melhorado?
- Avaliação Quantitativa
 - 3) Qualidade dos gráficos (1-10)
 - 4) Capacidade de transmissão de conceitos (1-10)
 - 5) Qualidade de controlo das personagens (1 - 10)

Relativamente às perguntas qualitativas, concluímos que os tópicos mais abordados relativamente aos elementos que mais agradaram são o aspeto gráfico, as animações e o sistema de troca de personagens. Podemos ver no Gráfico 1 a percentagem de pessoas que referiu cada um destes tópicos, e outros, como os seus aspetos preferidos da aplicação.

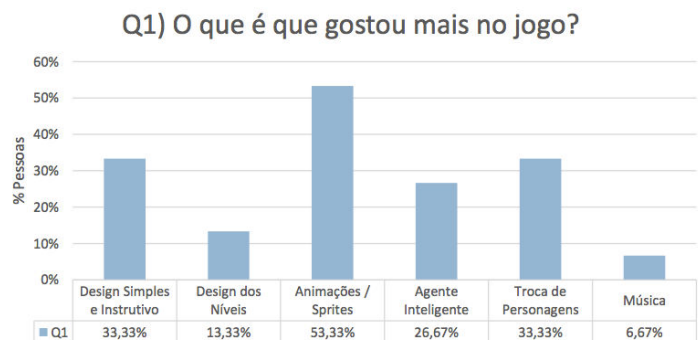


Gráfico 1. Percentagens de pessoas que referenciaram determinados tópicos relativamente à 1ª pergunta do questionário de testes.

Relativamente aos aspetos a melhorar (Gráfico 2), o tópico mais abordado foi a mecânica de salto na parede (wall jump) devido a algumas limitações existentes no seu controlo e à dificuldade do jogador realizar os saltos tal como desejaria. O segundo tópico mais abordado foi a inexistência de alguma ajuda quanto à explicação dos controlos da personagens. Pudemos concluir também que, para além destes tópicos abordados pela maioria, foram abordados outros aspetos como por exemplo a inexistência de checkpoints em níveis muito longos (e.g., as artérias) o que faz com que o jogador tenha de repetir todo o nível caso seja apanhado por um inimigo ou caia numa gordura ou numa ferida. Houve um igual número de pessoas a sugerirem extensões quanto ao papel do agente inteligente, nomeadamente a possibilidade de este intervir autonomamente sem que o jogador se dirija a ele ou mesmo a tecla de socorro referida anteriormente, para chamar o agente caso este não se encontre por perto.

Quanto às perguntas de avaliação quantitativas, podemos analisar através dos resultados apresentados no Gráfico 3 que o aspeto que teve melhor avaliação nos testes foi o aspeto visual do jogo, i.e., a qualidade dos gráficos 2D. Um dos aspetos que não ficou tão bem classificado foi o sistema de controlo das personagens e, através de uma análise das perguntas qualitativas, nomeadamente a segunda pergunta, deduzimos que este facto provavelmente se deve a problemas no controlo das

personagens que são notórios acima de tudo nos saltos da parede.

Q2) O que acha que podia ser melhorado?

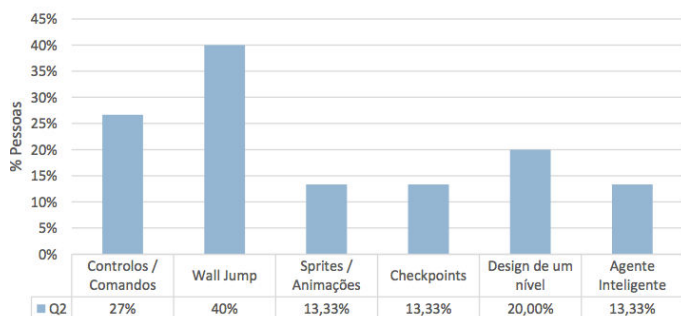


Gráfico 2. Percentagens de pessoas que referenciaram determinados tópicos relativamente à 2ª pergunta do questionário de testes.

Podemos concluir, através do Gráfico 3, que o aspeto mais bem classificado pelos utilizadores foi o aspeto visual do jogo, compreendendo tudo o que tem a ver com o desenho das personagens, cenários e elementos, respetivas animações e a sua disposição e organização no ecrã. A média de classificação para este aspeto foi de 9 em 10. As outras duas perguntas, referentes tiveram a mesma média, 8 em 10. Já a última pergunta, apesar de um resultado igualmente satisfatório, é a que apresenta maior desvio padrão, sendo a questão que mais classificações diversificadas obteve por parte dos jogadores. Isto pode querer dizer que este é ainda um aspeto frágil do nosso sistema, pois pode conter falhas ou limitações que podem comprometer de alguma forma a jogabilidade ou conter aspetos que cativam o jogador e que o fazem gostar ainda mais da experiência.

Respostas das Perguntas Quantitativas

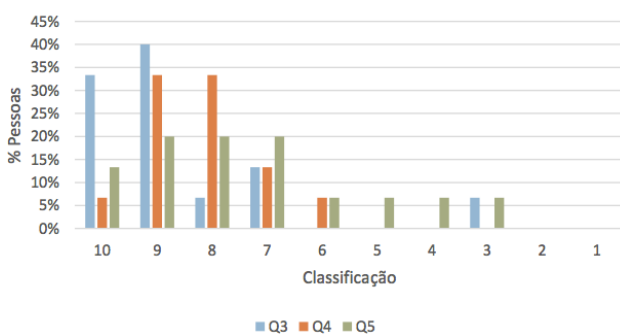


Gráfico 3. Percentagens de pessoas que atribuíram uma determinada classificação (de 1 a 10) a cada pergunta quantitativa do questionário de testes.

8. TRABALHO FUTURO

Como trabalho futuro propomos a melhoria de alguns aspetos de jogabilidade, bem como a introdução de novos elementos. Devemos referir a importância de testar o jogo com o público-alvo, a fim de provar a componente lúdica e pedagógica do jogo. Deve-se também questionar vários professores de biologia a fim de se poder fazer melhorias no sistema de acordo com o programa da disciplina.

Uma melhoria quanto à jogabilidade seria a introdução da opção de multiplayer. Assim, se o jogo fosse jogado

nas escolas, os computadores necessários seriam reduzidos a 1/3 visto que o jogo poderia ser jogado com 3 pessoas de cada vez. Esta opção poderia levar à cooperação entre os estudantes.

9. UNITY

Para a implementação do jogo foi utilizado o motor Unity [Unity12]. O Unity oferece uma IDE e uma framework que permitem desenvolver jogos para múltiplas plataformas (consolas, PC, dispositivos móveis, etc...) Todo o código do jogo foi desenvolvido na linguagem de programação C#.

10. CONCLUSÃO

Este projeto foi desenvolvido com o objetivo de auxiliar o processo de aprendizagem do funcionamento do sistema circulatório e dos órgãos com que interage. Devido à natureza educacional do jogo houve a necessidade de fazer um balanceamento entre a componente de diversão e a componente pedagógica, pelo que foi necessário encontrar um balanço entre os elementos de diversão que o jogo apresenta (e.g.: controlo dos glóbulos, níveis de eventos) e os elementos que passam informação sobre o sistema circulatório (e.g.: diálogos, vídeos). Este balanço foi alcançado através da passagem de informação de uma forma pontual e sempre em pequenas quantidades. Nas últimas iterações o material começou a ser validado por especialistas da área, nomeadamente por uma professora e por uma bióloga. Por fim, obtivemos um resultado satisfatório que conseguiu ter um impacto positivo nos seus utilizadores e que conseguiu entusiasamá-los e surpreendê-los.

REFERÊNCIAS

- [AdobeFlash14] Adobe Flash, <http://get.adobe.com/br/flashplayer/> (accessed: 2/05/2014).
- [AdobeIllustrator14] Adobe Illustrator, <http://www.adobe.com/br/products/illustrator.html> (accessed: 2/05/2014).
- [Barillé87], Barillé, A., Il était une fois. 1987
- [Bayliss07] Jessica D. Bayliss. The effects of games in cs1-3. *Journal of Game Development*, 2(2), 2007.
- [Lester97] Lester, J. C. and Stone, B. (1997). Increasing Believability in Animated Pedagogical Agents. *International Conference on Autonomous Agents*. Marina del Rey, p. 16-21.
- [Logas05] Logas, Heather; Muller, Daniel. *Mise-en-scène Applied to Level Design: Adapting a Holistic Approach to Level Design*, DiGRA 2005 Conference: Changing Views – Worlds in Play.
- [Nealen11] Andrew Nealen, Adam Saltsman and Eddy Boxerman. *Towards Minimalist Game Design*. In *proceedings of Foundations of Digital Games (FDG) 2011*.
- [Sicart08] Sicart, M. (2008). *Defining Game Mechanics*. *Game Studies: The international Journal of Computer Games Research*, 8(2).
- [Smith08] Smith, G., Cha, M. and Whitehead, J. A *Framework for Analysis of 2D Platformer Levels*. Los Angeles, CA : ACM, 2008. *Proceedings of ACM SIGGRAPH Sandbox Symposium 2008*.

Sessão - Interação

EduVis: Visualização Interativa de Dados Educacionais

Sandra Gama Vilma Jordão Daniel Gonçalves
INESC-ID e Instituto Superior Técnico, Universidade de Lisboa
R. Alves Redol, 9, 1000-029 Lisboa, Portugal

sandra.gama@ist.utl.pt, vilma.jordao@ist.utl.pt, daniel.goncalves@inesc-id.pt

Resumo

Uma análise efetiva dos processos de ensino permite proporcionar à comunidade educativa um meio para analisar os resultados dos processos de ensino, apresentando potencial para ajudar a prevenir e corrigir situações problemáticas, levando a uma educação mais eficaz e bem sucedida. A aplicação de técnicas de data mining no âmbito da educação fornece ferramentas para analisar dados de ambientes educacionais, desde o comportamento dos alunos até às estratégias de ensino e coordenação de programas de estudos. No entanto, estes resultados incluem um amplo conjunto de comportamentos que consistem em padrões simbólicos, muitas vezes representados sob a forma textual, que normalmente é difícil de entender. Dado o potencial da visualização para representar grandes quantidades de dados, esta pode superar essa limitação, aliviando a carga cognitiva associada à interpretação de informação. Assim, utilizando os resultados de técnicas de data mining educacional que já tinha sido aplicado para analisar a interdependência entre as disciplinas de um curso universitário, foi desenvolvida uma visualização que permite a análise desses mesmos padrões. A visualização consiste em dois mecanismos distintos, mas coordenados, de representação de informação. O primeiro corresponde a uma visualização interativa multi-nível das disciplinas dos vários semestres do curso, em que as relações de dependência são representadas através de conetores visuais. A segunda ferramenta, que consiste em matrizes multicamada, permite aplicar mecanismos de filtragem e explorar mais detalhadamente as relações entre as disciplinas. A complementaridade dos dois métodos de visualização permite tornar vários padrões imediatamente perceptíveis e permite a navegação e análise de informação. A elaboração de testes com utilizadores reiterou estas características, provando o potencial do sistema para possibilitar uma solução eficaz para a visualização de padrões educacionais.

Palavras-Chave

Visualização de Informação, Análise Interativa de Dados Educacionais, Interação Pessoa-Máquina

1. INTRODUÇÃO

O número de alunos, no ensino tradicional e nos cursos on-line, tem vindo a crescer nas últimas décadas. Quanto aos estudantes universitários, a taxa de escolarização mundial aumentou de 8.5%, em 1970, para 24.7%, em 2006 [Gao 10]. Quanto ao ensino on-line, a criação e a profusão de MOOC (Cursos Massivos Abertos Online), em que pessoas de todo o mundo podem participar, também contribuiu para um crescimento no número de estudantes a nível mundial. Como resultado, os CMS (Sistemas de Gestão de Cursos) e LMS (Sistemas de Gestão da Aprendizagem) tornaram-se populares e tiveram um grande impacto no reforço da educação à distância [Kay 13].

Compreender a eficácia de um programa de estudos ou de uma disciplina implica perceber como os alunos atingem o sucesso. Tendo em conta os cursos universitários, é importante analisar as taxas de sucesso das disciplinas e como se influenciam entre si. Este processo potencialmente ajudará a perceber e corrigir problemas e limitações, permitindo uma melhoria da adequação dos currículos, levando a melhores

resultados. Por outro lado, com o crescente número de estudantes no ensino tradicional e a profusão de cursos on-line, um extenso conjunto de dados emerge dos currículos dos alunos. Esta informação, se for explorada eficazmente, pode ser crucial para melhorar os processos de ensino.

A aplicação de técnicas de *data mining* neste contexto é um tema de pesquisa emergente, que tem vindo a fornecer meios para analisar dados educacionais, desde o comportamento dos alunos às estratégias de ensino e coordenação de cursos, assumindo a denominação de EDM (*Data Mining Educacional*). Tanto na educação on-line como na tradicional, a informação original não permite a análise direta de aspetos como, por exemplo, o comportamento dos alunos ou o sucesso das estratégias de ensino. No entanto, a EDM fornece padrões relevantes com base em informações disponíveis e faz com que seja possível fazer previsões com base nos dados educacionais. Contudo, o resultado normalmente consiste num extenso conjunto de comportamentos que são descritos sob a forma de padrões textuais que normalmente são difíceis de entender e inter-relacionar devido à sua complexidade visual. Adicionalmente, a com-

preensão deste tipo de informação muitas vezes implica um conhecimento razoável de algoritmos e estatística matemática, o que assume competências específicas que o utilizador pode não possuir. No entanto, essa limitação deve ser superada, uma vez que a interpretação desta informação é de extrema importância para que a EDM seja útil e eficaz.

Em termos gerais, não só é necessário perceber a informação a nível global, como um todo coerente, mas também compreender os padrões específicos e inter-relacionar informação. Fornecer os instrumentos para permitir uma interpretação fácil e correta dos dados educacionais é, portanto, de extrema importância. Para superar esse obstáculo e obter análise de dados eficaz, o utilizador deve estar envolvido no processo de exploração de uma forma que combine criatividade, flexibilidade e conhecimento geral [Keim 02]. A visualização tem potencial para superar este desafio, uma vez que é uma excelente forma para representar grandes quantidades de dados e alivia a carga cognitiva associada à interpretação de informação [Ware 12]. Como resultado, a criação de uma visualização que represente os resultados de *data mining* educacional terá boas possibilidades de fornecer à comunidade educativa informação relevante e perceptível neste contexto, permitindo compreender problemas que de outra forma permaneceriam despercebidos e procurar soluções adequadas.

Esta necessidade levou-nos a criar uma visualização que representa padrões educacionais, permitindo a sua navegação e análise eficaz. É representada informação resultante da aplicação de técnicas de EDM para analisar a interdependência entre os cursos num curso universitário [Antunes 08]. Estes padrões fornecem o número de alunos que obtêm sucesso a cada disciplina do curso, assim como informações relativas a relações de dependência entre disciplinas. Exemplos de relações de dependência são: *50% dos estudantes que não conseguiram obter sucesso à cadeira A num determinado semestre, provavelmente não conseguirão completar a disciplina B no semestre seguinte* ou *25% dos alunos que foram aprovados na disciplina C num determinado semestre, também foram aprovados na cadeira D no mesmo semestre*. Assim, foi criada uma visualização que representa estes padrões e permite sua exploração através da coordenação de dois métodos interativos complementares: (i) uma visualização em camadas que tira partido de conectores visuais para representar relações de dependência; (ii) uma visualização que fornece mecanismos de filtragem para exploração mais detalhada de padrões, utilizando mapas de cor para evidenciar relações entre disciplinas.

Na secção 2 são analisados e discutidos estudos relevantes no contexto da visualização de dados educacionais. Na secção 3 apresentamos os padrões obtidos a partir de EDM. Em seguida, é descrita a abordagem adotada relativamente à visualização e são detalhados os aspetos mais importantes da mesma. Na secção 5 são apresentados e analisados os resultados de avaliação com utilizadores da visualização criada. Por fim, são discutidas algumas conclusões relativas ao presente estudo.

2. TRABALHO RELACIONADO

Com o crescimento do número de alunos no ensino tradicional e online, várias ferramentas foram criadas no âmbito da representação gráfica de informação educacional.

Tendo em conta a educação online, os CMS permitem a criação de salas de aula virtuais, onde alunos e professores podem partilhar informação, tornando possível a participação remota em discussões e a gestão de aulas. Este tipo de interação gera grandes quantidades de dados que precisam de ser geridos de uma forma que forneça aos professores informação relevante sobre o desempenho dos alunos. Para superar esse desafio, foi criado o CourseVis [Mazza 05], sendo usado como uma extensão ao CMS que permite a exploração interativa de dados e manipulação através de diferentes mecanismos de visualização. Um dos mecanismos é a representação tridimensional de participantes num fórum, em que os tópicos são representados como esferas com tamanho proporcional ao número de alunos envolvidos, permitindo *zoom* e visualização panorâmica. Outro mecanismo é a Matriz Cognitiva, que consiste de uma matriz na qual os nomes dos alunos são representados ao longo de um eixo e os conceitos da disciplina no outro eixo, sendo associada uma escala de cor ao desempenho, variando entre o verde (sucesso) e o vermelho (insucesso). Uma terceira ferramenta, que retrata o comportamento dos alunos através de gráficos e texto ordenados de forma matricial, tem como objetivo mostrar informação como o acesso aos conteúdos, a assiduidade e o progresso escolar. Um estudo com utilizadores mostrou que as representações gráficas do CourseVis permitem obter informação sobre os aspectos cognitivos dos estudantes de forma rápida e precisa. No entanto, os utilizadores tiveram alguns problemas na compreensão da informação devido à sobreposição de elementos visuais, resultando em dificuldade na leitura dos gráficos na representação do comportamento dos estudantes [Iliinsky 11]. Estas limitações levaram à criação de GISMO (Sistema Gráfico Interativo de Monitorização de Estudantes), que representa visualmente dados de LMS, bastante complexos e difíceis de ler e entender. O GISMO está integrado no LMS Moodle mas pode também ser adaptado para outras plataformas de aprendizagem. Focado numa visualização de comportamentos bidimensional, o GISMO oferece uma visualização simples para parâmetros específicos. Permite a exploração interativa de acessos e detalhes de recursos, e fornece os meios para a exploração dos comportamentos dos alunos que tinham sido considerados relevantes no CourseVis [Mazza 05]. Tendo sido usado num curso online, o GISMO mostrou ser eficaz para a compreensão dos comportamentos individuais, bem como dos métodos de avaliação, permitindo redesenhar o curso de acordo com as necessidades dos alunos.

Dado que o ensino tradicional resulta de processos de avaliação, tais como notas de avaliação e outras competências, como assiduidade e participação, a informação resultante pode ser difícil de interpretar. Adicionalmente, a descoberta de padrões e troca de informação são quase

impraticáveis neste contexto. Um dos estudos que têm sido feitos no âmbito da visualização de informação para apoiar os processos educacionais é o AVOJ [Xiaohuan 13]. De modo a refletir as diferentes capacidades dos alunos, esta ferramenta permite a exploração e visualização de dados sobre o desempenho. Fornece meios para agrupar os alunos de acordo com suas notas e outros aspetos, como os hábitos de estudo. No entanto, faz uso de um esquema de cores com níveis máximos de brilho e saturação, competindo pela atenção visual do utilizador, o que torna mais difícil encontrar padrões [Iliinsky 11]. O sistema oferece um mecanismo de visualização adicional para comparar tendências gerais, composto por um gráfico de barras que mostra estatísticas em tempo real, tornando possível compreender alguns aspectos específicos, como a forma como os alunos gerem as suas sessões de estudo. Os autores acreditam que, uma vez fornecendo um amplo conjunto de dados que não estavam disponíveis anteriormente, será possível obter mais informação sobre os processos educacionais e melhorar a eficácia do ensino. Outro estudo interessante é o de Xiaoya et al. [Xiaoya 09], uma visualização para análise de resultados dos estudantes universitários num curso de Inglês. Seguem uma visualização de coordenadas paralelas, em que N eixos equidistantes são usados para representar as dimensões de um conjunto de dados multidimensional. No entanto, vários modelos adicionais têm sido utilizados: (i) classificação, que permite aos utilizadores dividir dados em conjuntos arbitrários de aulas, (ii) média, o que torna possível, por meio de interações com o gráfico principal, obter os valores médios de cada conjunto de dados ou subconjuntos de dados, (iii) *box plot*, para medir a dispersão dos dados, (iv) permuta de eixos, o que evidencia relações internas entre atributos; (v) correlação, para calcular a correlação entre dois conjuntos de dados ao longo de dois eixos paralelos, (vi) associação, no qual se pode prever a ocorrência de atributo A com base na ocorrência de atributo B, e (vii) *roll-up* e *drill*, permitindo a representação de dados de diferentes níveis hierárquicos. Estes modelos, além de permitirem a representação e manipulação interativa de dados, permitem uma visão geral imediata e mostram um número considerável de variáveis simultaneamente, permitindo uma análise mais eficiente e eficaz. Trimm et al. [Trimm 12] criaram uma visualização em que os alunos são agrupados de acordo com suas notas. Estes grupos podem ser visualizados utilizando composições que mostram as suas características e variações no tempo. A fim de utilizar a composição, a informação sobre o historial de cada aluno é constituída por uma trajetória bidimensional, representada em dois eixos. Esta representação, embora simples, não é muito eficiente para mostrar simultaneamente características e tendências de uma grande quantidade de dados dos estudantes. A visualização utiliza recursos espaciais para representar trajetórias dos alunos ao longo do tempo usando o algoritmo de nível definido. É utilizada uma técnica de mistura de cor para mostrar características como a média e o desvio padrão em relação aos resultados dos alunos, representando a informação de uma forma na-

tural, e é utilizada a técnica de *color weaving* para mostrar o valor de um determinado atributo em relação a uma trajetória selecionada aleatoriamente. No entanto, o uso de gradientes pode adicionar ruído à imagem e a variação de brilho e saturação pode também causar a ilusão de proximidade, levando a uma interpretação imprecisa dos dados. No entanto, testes com utilizadores mostraram que esta visualização fornece meios para encontrar novos padrões relevantes nos dados. A fim de entender o abandono escolar de um número significativo de estudantes relativamente a um curso de ciência da computação, foi desenvolvida uma ferramenta que permite a visualização de padrões de repetição de sucesso ou insucesso [Wortman 07]. Devido à grande quantidade de informação disponível, foi usada uma estrutura visual de nós e arestas. Os nós representam eventos, com largura proporcional ao número de estudantes de cada evento, e as arestas representam a trajetória dos estudantes relativamente aos eventos (exames ou trabalhos), sendo a largura representativa do número de alunos a ser descritos neste comportamento. As cores representam o desempenho dos alunos, permitindo diferenciar grupos de estudantes, identificando os que têm comportamento similar. A visualização é interativa, permitindo a seleção de categorias de estudantes, como *alunos que repetem pelo menos uma disciplina* ou *alunos que nunca reprovaram uma disciplina*. Como resultado, foi possível aos professores obter um conjunto de conclusões a respeito de falhas repetitivas ou implicações de uma determinada cadeira no sucesso noutras disciplinas. A estrutura visual tem, contudo, uma importante lacuna no que diz respeito à sobreposição das linhas mais grossas às mais finas, o que pode tornar difícil perceber alguma informação.

Todas as abordagens mencionadas focam a visualização de informação relativamente aos processos de ensino, tradicional ou on-line. Apresentam diferentes técnicas e metodologias para visualizar os dados neste contexto em particular. Dado o âmbito do presente estudo e as particularidades dos dados que pretendemos analisar, em que uma visualização que permita inter-relacionar disciplinas e mostrar as suas interdependências apresenta maior relevância, destacamos o trabalho de Wortman e Rheingans [Wortman 07] e, mais recentemente, o de de Trimm et al. [Trimm 12]. Uma visualização interativa que permita mecanismos interativos, como destaque, comparação e filtragem, é de extrema importância no âmbito deste estudo. No entanto, o último trabalho supracitado, de Trimm et al. [Trimm 12], oferece interação limitada sobre as particularidades do nosso contexto. Além disso, o estudo de Wortman e Rheingans [Wortman 07], apesar de proporcionar mecanismos de interação, não permite comparação entre disciplinas ou destaque de padrões específicos. Além disso, o esquema de cores usado não alivia a confusão visual que está presente quando muitas arestas diferentes são representadas. Desenvolvemos, assim, uma técnica de visualização que representa as inter-relações entre as disciplinas de um programa de estudos na tentativa de colmatar as lacunas das soluções existentes neste contexto.

3. PADRÕES EDUCACIONAIS

Um dos desafios na utilização de dados de ensino de modo a melhorar o sucesso é a aquisição de conhecimento do contexto, no que diz respeito a estratégias de ensino atuais e aos comportamentos dos alunos. Neste estudo foi utilizado o resultado de *data mining* sequencial que tinha sido anteriormente aplicado a dados recolhidos durante nove anos num programa de estudos em ciência da computação [Antunes 08]. O objetivo do *data mining* sequencial é, dado um conjunto de sequências e um limiar de suporte mínimo, descobrir o conjunto de sequências que estão contidas em pelo menos σ sequências do conjunto de dados, isto é, o conjunto de sequências frequentes [Agrawal 95]. Esta técnica permite a descoberta de padrões sequenciais frequentes, sendo estes consistentes com o conhecimento de fundo existente. Tal conhecimento pode ser representado por uma linguagem livre de contexto, que desempenha o papel de uma restrição no processo de *data mining* sequencial. Este método não só reúne padrões esperados com base no conhecimento de fundo mas, com o uso de relaxamentos, permite também a descoberta de padrões que correspondem a desvios ao comportamento esperado, tornando evidentes algumas tendências potencialmente relevantes que antes eram desconhecidas [Antunes 08]. Assim, o conhecimento do currículo escolar foi representado como um autómato finito, estabelecendo a ordem das disciplinas que os alunos deviam completar para concluir o curso. O *data mining* sequencial com três diferentes valores de limiar de suporte (50 %, 25 % e 20 %) foi realizado, resultando em três diferentes conjuntos de padrões. Evidentemente, quanto menor é o valor do limiar de suporte, maior é o número de padrões resultante da aplicação do *data mining* sequencial. Como um resultado da aplicação do método acima referido, foi gerado um conjunto de padrões textuais. Os padrões observam a seguinte estrutura:

$$\text{Padrao}_i = (\text{semestre}_1, \dots, \text{semestre}_N, \text{total}_{\text{alunos}}),$$

$$\text{semestre}_j = \text{disciplina}_1 \vee (\text{disciplina}_1, \dots, \text{disciplina}_M)$$

Alguns exemplos são os seguintes:

- $(fex, 2000)$: 2000 alunos concluíram *fex* no primeiro semestre;
- $(fex, tc, 1400)$: 1400 alunos concluíram *fex* e *tc* no segundo semestre;
- $((fex, AM1), tc, 1000)$: 1000 alunos concluíram *fex* e *am1* no primeiro semestre e *tc* no segundo semestre;
- $((fex, AM1), (fisica1, tc), AM2, 800)$: 800 alunos completaram *fex* e *am1* no primeiro semestre, seguido por *fisica1* e *tc* no segundo semestre e *am2* no terceiro semestre.

Embora a informação textual torne difícil entender padrões específicos e forneça pouca informação geral, esta estrutura de padrões disponibiliza informação sobre as interações entre os diferentes semestres do curso que uma visualização eficaz será capaz de tornar evidente.

4. VISUALIZAÇÃO DE DADOS EDUCACIONAIS

Tendo em conta o contexto do estudo e os padrões a representar, foi criada uma visualização que mostra relações de dependência entre cadeiras, permitindo saber qual ou quais as disciplinas mais importantes por influenciarem o sucesso dos alunos ao longo do percurso escolar. Para tal, foi criado o EduVis, uma visualização coordenada, representada na Figura 1, que tira partido da conjugação de duas ferramentas principais: (i) uma representação multi-camada das disciplinas do programa curricular, inter-relacionadas através de conetores visuais; e (ii) uma representação multi-matricial em que são apresentadas as disciplinas, sendo os respetivos padrões representados através de mapas de cor. As duas ferramentas possuem características distintas e têm funcionalidades complementares. As secções seguintes descrevem em detalhe cada uma das ferramentas principais da visualização e descrevem os mecanismos de interação.

4.1. Visualização multi-camada

Nesta visualização, cada nível corresponde a um semestre de um programa curricular, representando as disciplinas desse semestre que têm algum tipo de relação com outras. Assim, as cadeiras são geralmente representadas como círculos verdes com tamanho proporcional ao número total de estudantes que as completaram. Quando existem dados sobre o insucesso, o círculo da cadeira é sub-dividido em dois semicírculos, mostrando informação sobre a aprovação e a reprovação através do código convencional ocidental de cor [Ware 12]. O verde, à esquerda, representa o número de alunos aprovados e o vermelho, à direita, mostra o número de estudantes que reprovaram, como representado na Figura 2. Este mecanismo, embora simples, torna possível entender imediatamente as disciplinas com elevada ou baixa taxa de sucesso, assim como comparar visualmente o sucesso e o insucesso de uma disciplina, sem a necessidade de exploração adicional.

4.2. Visualização multi-matricial

Nesta visualização cada matriz representa um semestre do curso, sendo cada disciplina correspondente a um quadrado, que se divide em dois triângulos: o superior representa sucesso e o inferior representa insucesso. Antes de qualquer interação, ou seja, no estado inicial da visualização, como representado na Figura 3, as cadeiras sobre as quais existe informação são representadas por cores entre o amarelo e o azul escuro, enquanto as disciplinas sem padrões são representadas a cinzento. O brilho representa o número de padrões em que as cadeiras estão envolvidas: quanto menor o brilho (mais "escura" a cor), mais relações com outras cadeiras essa disciplina tem. Se existirem poucas ou nenhuma dependências, então fica associada a um tom com brilho elevado (um tom mais "claro").

4.3. Interação

A interação inicia-se quando uma disciplina é selecionada, em qualquer uma das visualizações. Na representação multi-camada, esta ação corresponde ao movimento do



Figura 1. EduVis: Visualização principal

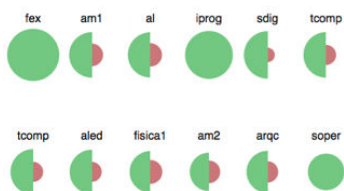


Figura 2. Visualização multi-camada (detalhe)



Figura 3. Visualização multi-matricial (detalhe)

cursor sobre o semi-círculo correspondente à aprovação ou reprovação, enquanto que na representação multi-matricial ocorre com o movimento do cursor sobre um triângulo, correspondente também à aprovação ou reprovação de uma cadeira. Sendo uma disciplina selecionada, ambas as visualizações refletem essa seleção. Na visualização multi-camada é destacado o círculo atual, sendo atribuída mais saturação ao mesmo, e é representada informação sobre as relações desta cadeira com as restantes através de conetores visuais. Estes consistem em curvas de Bézier cúbicas com espessura proporcional ao número de alunos que verificam o padrão, informação que é reforçada pela cor, utilizando convenções ocidentais de mapas de calor [Ware 12]. Assim, padrões que surjam de cadeiras com aprovação (semi-círculos verdes) são representados com cores entre o azul (menor quantidade de alunos) e o verde (maior quantidade), como ilustrado na Figura 4. De forma análoga, padrões que surjam de reprovações (semi-círculos vermelhos) são representados com cores entre o amarelo (menor quantidade de alunos) e o vermelho (maior quantidade). Não são, no entanto, usadas cores totalmente sa-

turadas, a fim de evitar que os artefactos visuais compitam pela atenção do utilizador [Ware 12]. Quando o cursor sai do círculo, a informação relativa às relações com outras disciplinas deixa de ser representada.

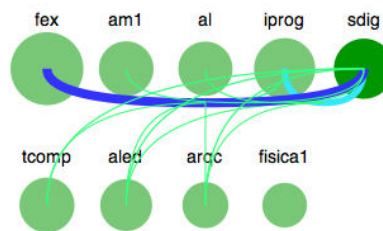


Figura 4. Visualização multi-camada (sdig)

Na visualização multi-matricial, as alterações refletem-se essencialmente a nível da cor. Assim, para demonstrar

as dependências, são usadas duas cores, azul e vermelho, que correspondem a um mapa de calor relativamente à aprovação e reprovação: o triângulo superior, representado em azul, está associado às aprovações, e o inferior, vermelho, representa as reprovações. Ao ser colocado o cursor sobre um triângulo, são representadas as cadeiras com que a cadeira selecionada está relacionada, a nível de aprovações e reprovações, como ilustrado na Figura 5, correspondente à seleção da disciplina *am1* no primeiro semestre. De forma semelhante à visualização multi-camada, quando o cursor sai do triângulo, a informação relativa às relações com outras disciplinas deixa de ser representada.

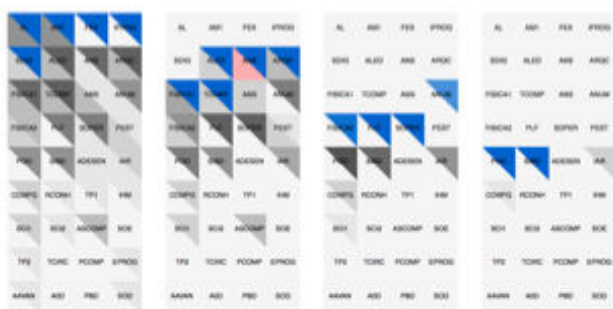


Figura 5. Visualização multi-matricial (*am1*)

Adicionalmente, sempre que uma disciplina é selecionada, é apresentada, no painel retangular entre as duas visualizações, informação detalhada sobre os padrões relacionados com essa disciplina. Quando um padrão é selecionado nesta lista, a informação correspondente reflete-se na visualização principal.

O EduVis possibilita a comparação simultânea de disciplinas. Na visualização multi-camada, ao clicar sobre um semi-círculo, este fica selecionado, não desaparecendo os seus padrões com o deslocar do cursor. De forma similar, na visualização multi-matricial, clicar nos triângulos fixa os padrões correspondentes e a disciplina selecionada fica em destaque com um contorno laranja. Desta forma, ao ser selecionada outra disciplina (através do deslocamento do cursor sobre a mesma), é possível comparar duas cadeiras simultaneamente em ambas as visualizações.

A visualização matricial disponibiliza ainda mecanismos de filtragem para exploração de padrões. Ao passar o cursor sobre os padrões de uma cadeira previamente selecionada, são visíveis as relações comuns entre a cadeira inicial e a cadeira onde se encontra o cursor atualmente. Ao clicar num triângulo dessa cadeira, apenas ficam visíveis os padrões comuns às duas cadeiras selecionadas, como representado na Figura 1. Faz-se assim uma restrição ao conjunto de relações iniciais da primeira disciplina selecionada. Assim, é possível fazer restrições indefinidamente, até ao ponto limite em que todas as cadeiras representadas fazem parte do mesmo conjunto de relações.

O EduVis tem, assim, três aspectos principais. O primeiro é leitura imediata de informação: pode-se perceber o número de semestres, bem como o número de dis-

ciplinas para cada nível e reconhecer de imediato as cadeiras concluídas por um maior número de alunos e as mais problemáticas. Outro aspecto importante é a exploração de dados: quando uma disciplina está selecionada, é representada informação, nomeadamente relativa às relações com outras cadeiras, que é o foco dos padrões que pretendemos representar. Finalmente, o sistema fornece mecanismos de comparação e filtragem interativas, permitindo a comparação entre cadeiras e permitindo criar restrições à informação representada, fornecendo ferramentas para uma exploração simples e eficiente.

5. AVALIAÇÃO

Foi realizado um estudo com utilizadores para avaliar a solução criada para a visualização de padrões educacionais. Como resultado, tendo em conta o contexto do estudo supracitado, pretendemos inferir: (i) a eficácia e eficiência da nossa solução, tendo para isso registado o tempo e número de erros associado ao desempenho de um conjunto de tarefas; (ii) a usabilidade, em geral, e a capacidade de aprendizagem, em particular, que o contacto com a visualização criada proporciona; (iii) o grau de satisfação dos inquiridos em relação ao desempenho das tarefas.

5.1. Tarefas Representativas do Sistema

Relativamente às tarefas selecionadas para avaliar a visualização desenvolvida, foi criado um conjunto de perguntas representativas: (1) *Quantos semestres são representados?* (2) *A nível geral, quais são as duas cadeiras com mais alunos?* (3) *Qual o conjunto de cadeiras envolvidas em mais padrões no segundo semestre?* (4) *Quais as cadeiras relacionadas com iar no quarto semestre?* (5) *Considerando os alunos que fizeram am1 e fisica1, quais são as outras cadeiras em que também obtiveram sucesso no segundo semestre?* (6) *Quais são as cadeiras comuns a quem fez sdig no primeiro semestre e obteve reprovação a aled no segundo semestre?* (7) *Considerando a cadeira fisica2 no terceiro semestre, embora seja a que tem mais padrões associados, é esta a cadeira com mais alunos nesse semestre?*

As perguntas 1, 2 e 3 correspondem a aspetos gerais imediatamente perceptíveis na visualização: o número de semestres é visível em ambas as visualizações, representado pelas camadas de círculos na visualização superior e matrizes na inferior. Na primeira visualização consegue-se imediatamente verificar, através do tamanho dos círculos, as cadeiras com mais alunos. Finalmente, na representação inferior consegue ver-se imediatamente quais são as cadeiras envolvidas em mais padrões, através do destaque que é feito com recurso à cor. A questão 4 requer exploração de qualquer um dos mecanismos de visualização, selecionando a cadeira de interesse em qualquer uma das ferramentas. A questão 5 requer a aplicação de filtros na visualização inferior para se conseguir responder, enquanto a questão 6 obriga a que se utilizem filtros e se perceba a mistura de cor. Finalmente, a questão 7 requer que o utilizador utilize uma conjugação de ambas as visualizações, a inferior para perceber que de facto a ca-

deira é a que tem mais padrões e a superior para perceber o número de alunos da cadeira mencionada, face às restantes.

5.2. Protocolo de teste

Antes de iniciar os testes, os participantes foram abordados individualmente, tendo-lhes sido explicado o contexto do presente estudo e pedida a sua colaboração. Depois de uma descrição verbal da visualização e dos seus componentes fundamentais, as principais funcionalidades foram exemplificadas. De seguida, foi entregue aos participantes um questionário com algumas perguntas de perfil e uma lista de outras perguntas, correspondentes ao conjunto supracitado de tarefas a realizar. Estas perguntas deviam ser respondidas durante a interação com a visualização, através de um computador portátil (comum a todos os utilizadores). Durante esta fase de teste, foi medido o tempo e o número de erros para cada tarefa. No final, os participantes foram encorajados a fazer comentários verbais ao sistema. Por último, foi-lhes pedido que preenchessem um questionário *online* de satisfação, consistindo em duas partes: a primeira parte corresponde ao SUS (*System Usability Scale*) [Brooke 96] e a segunda diz respeito a um pequeno conjunto de perguntas que têm como objetivo avaliar, também usando uma escala de Likert de 5 pontos, o grau de dificuldade sentido pelos participantes ao realizar cada tarefa, na compreensão dos seguintes aspetos: (i) o número de semestres representados, (ii) o sucesso e o insucesso, (iii) as cadeiras envolvidas em mais padrões, (iv) as relações de dependência entre as cadeiras, e (v) os padrões que envolvem diferentes cadeiras simultaneamente.

5.3. Resultados

O estudo foi realizado com 20 participantes, dos quais 15 (75%) são do género masculino e 5 (25%) do género feminino. Do total de participantes, 3 (15%) têm entre 18 e 24 anos de idade, 10 (50%) entre 25 e 34 anos e 3 (15%) têm entre 35 e 44 anos, além de que 2 (10%) têm entre 45 e 54, 1 (5%) entre 55 e 64 anos e 1 (5%) está situado na faixa etária superior aos 65 anos de idade. Quanto à escolaridade, 16 (80%) têm um curso superior, enquanto 2 (10%) completaram o ensino secundário e 2 (10%) concluíram os estudos do ensino básico.

Tabela 1. Tempo e número de erros médio

Tarefa	Tempo (segundos)	Número de erros
1	3.20	0.35
2	17.60	0.10
3	28.30	0.55
4	23.95	0.30
5	27.70	0.30
6	49.45	0.75
7	45.70	0.30

Relativamente aos tempos médios e número de erros médio relativos ao desempenho de cada uma das tarefas, estes são sumarizados na tabela 1. Embora as tarefas 1, 2 e 3 sejam imediatas e portanto não necessitem de exploração adicional, os valores medidos não permitem criar uma

generalização acerca do tempo e número de erros para desempenho destas tarefas face às tarefas de exploração (4 a 7), com possível exceção das tarefas 6 e 7. Para aprofundamento dos resultados, conduzimos uma análise estatística. Foi aplicado um teste estatístico Shapiro-Wilk, que mostrou evidência contra uma distribuição normal na maior parte dos dados ($p < 0.05$), pelo que aplicámos um teste não-paramétrico, o teste *Wilcoxon signed-rank* para descobrir diferenças significativas entre amostras. De facto, a nível de tempo, a tarefa 1 é significativamente mais rápida que as restantes ($z_{1-2} = -3.72$, $z_{1-3} = -3.92$, $z_{1-4} = -3.92$, $z_{1-5} = -3.92$, $z_{1-6} = -3.92$, $z_{1-7} = -3.92$, $p < 0.05$), enquanto a tarefa 2 é significativamente mais rápida que as tarefas 3, 5, 6 e 7 ($z_{2-3} = -2.39$, $z_{2-5} = -2.54$, $z_{2-6} = -3.32$, $z_{2-7} = -3.25$, $p < 0.05$). Por outro lado, as tarefas 3, 4 e 5 são significativamente mais rápidas que as tarefas 6 e 7 ($z_{3-6} = -2.31$, $z_{3-7} = -2.17$, $z_{4-6} = -3.45$, $z_{4-7} = -2.76$, $z_{5-6} = -3.36$, $z_{5-7} = -2.63$, $p < 0.05$). No entanto, no que diz respeito ao número de erros, existem diferenças significativas apenas entre a tarefa 6 e as que têm menos erros, a tarefa 2 ($W = 6$, $cv = 17$, $p < 0.05$) e a tarefa 4 ($W = 0$, $cv = 3$, $p < 0.05$), e entre as tarefas 2 e 3 ($W = 5$, $cv = 8$, $p < 0.05$). Numa tentativa de encontrar uma correlação entre o tempo e o número de erros, calculámos os coeficientes de Pearson, encontrando uma correlação, embora relativamente fraca, na tarefa 2 ($r = 0.45$, $p < 0.05$) e na tarefa 7 (0.49, $p < 0.05$), o que não nos permite generalizar uma correlação entre um maior tempo para desempenho da tarefa e um maior número de erros. É possível concluir, assim, que a complexidade temporal de uma tarefa no EduVis não leva a um maior número de erros no desempenho da mesma.

Tendo em conta o questionário de satisfação, a pontuação relativa ao SUS segundo os correspondentes parâmetros de cálculo ([Sauro 11]), foi de 79.47 pontos, mostrando resultados bastante elevados no que diz respeito à usabilidade e capacidade de aprendizagem do sistema. Usando o mesmo método para calcular as respostas de satisfação contextuais, obtivemos um resultado 92.11 pontos, mostrando que nossos objetivos foram alcançados no que diz respeito à satisfação no desempenho das tarefas.

5.4. Discussão

Os resultados da avaliação mostraram que os utilizadores foram capazes de obter as informações necessárias para concluir as tarefas em tempo razoável e com número reduzido de erros. Para tarefas mais complexas, a necessidade de maior interação com o sistema reflete um natural aumento do tempo de desempenho da tarefa que, no entanto, não se traduz num aumento do número de erros, o que mostra que a visualização permite obter a informação necessária ao desempenho das tarefas de uma forma eficaz. Estes resultados foram corroborados pelos questionários de satisfação, que provaram a usabilidade do sistema e mostraram elevada satisfação por parte dos utilizadores. A conjugação destes aspetos mostra, assim, a eficácia do EduVis para a visualização de padrões educacionais.

6. CONCLUSÕES

Uma grande quantidade de dados emerge das atividades educacionais com o crescente número de estudantes no ensino tradicional e on-line. Analisada eficazmente, esta informação pode ajudar a refinar os processos de educação. Neste contexto, as técnicas de *data mining* têm mostrado ser relevantes para encontrar padrões nos dados, mas o resultado da aplicação dos mesmos leva muitas vezes a conjuntos de dados difíceis de ler, interpretar e analisar. Uma vez ultrapassada essa limitação será possível representar esta informação como um todo consistente e ter noção de aspetos particulares nos dados. Face a este desafio, criámos uma visualização que coordena dois diferentes mecanismos de interação complementares para visualizar relações entre unidades curriculares de um curso universitário. O EduVis fornece mecanismos de exploração, comparação simultânea e filtragem, permitindo gerir a informação. Tira partido da cor para destacar elementos visuais para enfatizar a informação relevante e aliviar a confusão visual associada à representação de um grande número de artefactos com propriedades semelhantes.

Os testes com utilizadores mostraram que a visualização torna imediatamente evidente um conjunto relevante de informação geral e permite uma fácil recolha de dados através de mecanismos de exploração e comparação interativa. Os participantes perceberam imediatamente aspetos como o número de semestres representados, tal como as disciplinas com mais ou menos sucesso. Utilizaram os mecanismos disponíveis para comparar, filtrar e recolher informação e mostraram satisfação no desempenho das tarefas. Concluímos que o EduVis fornece os meios para representar os resultados de informação educacional de uma forma que proporciona à comunidade educativa perceber um conjunto de padrões que não seriam evidentes de outra forma. Fornecendo os meios para diagnosticar determinados problemas, o sistema será útil para que os coordenadores de programas de estudo, tal como os professores, tenham os meios para encontrar soluções para as limitações existentes, promovendo o sucesso na educação.

7. AGRADECIMENTOS

Este trabalho foi apoiado por fundos nacionais através da FCT (Fundação para a Ciência e a Tecnologia), no âmbito do projeto Educare - PTDC/EIA-EIA/110058/2009 e pelo fundo plurianual do INESC-ID - PEst-OE/EEI/LA0021/2013..

Referências

- [Agrawal 95] R. Agrawal e R. Srikant. Mining sequential patterns. Em *Proceedings of the IEEE International Conference on Data Engineering*, páginas 3–14, 1995.
- [Antunes 08] C. Antunes. Acquiring background knowledge for intelligent tutoring systems. Em *Proceedings of the International Conference on Educational Data Mining*, páginas 18–27, 2008.

- [Brooke 96] J. Brooke. SUS: A quick and dirty usability scale, 1996.
- [Gao 10] Y. Gao. A study on mass higher education in the world-based on comparative perspectives. Em *International Conference on Education and Management Technology (ICEMT)*, páginas 528–530, 2010.
- [Iliinsky 11] N. Iliinsky e J. Steele. *Designing Data Visualizations*. O'Reilly, 2011.
- [Kay 13] J. Kay, P. Reimann, E. Diebold, e B. Kummerfeld. Moocs: So many learners, so much potential ... *IEEE Intelligent Systems*, 28(3):70–77, 2013.
- [Keim 02] D. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, 2002.
- [Mazza 05] R. Mazza e V. Dimitrova. Generation of graphical representations of student tracking data in course management systems. Em *Ninth International Conference on Information Visualisation, 2005. Proceedings*, páginas 253–258, 2005.
- [Sauro 11] J. Sauro. *A Practical Guide to the System Usability Scale: Background, Benchmarks and Best Practices*. CreateSpace, 2011.
- [Trimm 12] D. Trimm, P. Rheingans, e M. desJardins. Visualizing student histories using clustering and composition. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2809–2818, 2012.
- [Ware 12] C. Ware. *Information Visualization: Perception for Design*. Elsevier, 2012.
- [Wortman 07] D. Wortman e P. Rheingans. Visualizing trends in student performance across computer science courses. Em *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education*, páginas 430–434, 2007.
- [Xiaohuan 13] W. Xiaohuan, Y. Guodong, W. Huan, e H. Wei. Visual exploration for time series data using multivariate analysis method. Em *8th International Conference on Computer Science Education (ICCSE)*, páginas 1189–1193, 2013.
- [Xiaoya 09] G. Xiaoya, L. Kan, e L. Ping. Visual analysis of college students' scores in english test. Em *4th International Conference on Computer Science Education (ICCSE)*, páginas 1816–1819, 2009.

IN2TUNNEL – Terapia de Exposição para Fobia de Atravessar Túneis

Inês Laureano Gomes Ana Paula Cláudio Maria Beatriz Carmo
LabMAg, Faculdade de Ciências, Universidade de Lisboa
1749-016 Lisboa, Portugal

fc37411@alunos.fc.ul.pt, apclaudio@fc.ul.pt, mbcarmo@fc.ul.pt

Augusta Gaspar

CIS- Centro de Investigação e Intervenção Social, ISCTE Instituto Universitário de Lisboa,
1649-026 Lisboa, Portugal

augusta.gaspar@iscte.pt

Resumo

O uso da Realidade Virtual na Terapia de Exposição (RVTE) permite que um paciente seja exposto à situação temida, sem recorrer a uma exposição real o que é particularmente vantajoso quando a reprodução da situação real é dispendiosa ou envolve risco físico. Neste artigo apresentamos uma aplicação Web que apoia o terapeuta no uso da terapia de exposição em pacientes com medo de atravessar túneis. Esta aplicação contém um conjunto de animações de cenários virtuais que recriam travessias de túneis, observadas do ponto de vista do passageiro ao lado do condutor, recriando uma situação em que o paciente não tem controlo relativamente ao que vai experienciar.

Descrevemos ainda um estudo com utilizadores que envolveu duas vertentes: uma com terapeutas para avaliar a utilidade potencial da presente aplicação no domínio da RVTE e outra com participantes voluntários para comparar a sensação de presença quando as animações são observadas numa projeção ou num modelo recente de equipamento de Realidade Virtual imersiva, constituído por uma estrutura que suporta um *smartphone*.

Palavras-chave Realidade Virtual, Realidade Virtual na Terapia de Exposição, RVTE, ansiedade, medo de atravessar túneis, sensação de presença.

1. INTRODUÇÃO

Ter medo de atravessar túneis pode ser um problema sério para pessoas que têm de o fazer diariamente, por exemplo, para chegar ao trabalho ou à escola. Muitas vezes, estas pessoas escolhem trajetos alternativos, apenas para fugir aos túneis, o que nalguns casos pode não ser uma opção racional ou viável.

O medo de atravessar túneis aflige um número não negligenciável de pessoas como mostra um estudo divulgado pela DEKRA¹, uma empresa de inspeções automóvel alemã. Esta empresa apresentou recentemente um estudo com base num questionário realizado a 1200 condutores² em que concluiu que 46% destes teve

“sempre” ou “algumas vezes” medo quando atravessou um túnel.

Atualmente, as pessoas são cada vez mais dominadas pelo *stress* do seu dia-a-dia, estando mais expostas a problemas de ansiedade. Os problemas de ansiedade são geralmente difíceis de diagnosticar e, o que é grave, se não forem convenientemente tratados podem transformar-se em fobias.

A fobia é um distúrbio de ansiedade, um medo incontroável que uma pessoa sente em determinada situação ou relativamente a algum objeto, podendo gerar ansiedade antecipatória, que se revela quando a pessoa fica ansiosa apenas por imaginar a situação ou o objeto [DSM-IV]. A fobia é distinta do medo dito normal, ou

¹ <http://www.dekra.com/en/home> (último acesso set/2014)

² http://www.dekra.com/en/pressemitteilung?p_p_lifecycle=0&p_p_id=ArticleDisplay_WAR_ArticleDisplay&ArticleDisplay_WAR_ArticleDisplay_articleID=15594973 (último acesso set/2014)

racional, pois o medo fóbico impede a pessoa de estar perante a situação que a perturba. Actualmente os terapeutas recorrem a dois tipos de técnicas de tratamento destas pessoas: *in vivo* e *in imagino*. Na primeira o paciente é exposto diretamente à situação ou objeto temido, enquanto na segunda se recorre à capacidade de imaginação do paciente para recriar mentalmente essa situação.

Na década de 90, surgiu uma nova ferramenta, a Realidade Virtual na Terapia de Exposição (RVTE). O recurso à Realidade Virtual permite criar cenários virtuais à medida do problema, muito semelhantes à situação real. A RVTE tem como principais vantagens não expor o paciente ao eventual perigo da situação real, e o facto de ser possível ao terapeuta controlar o nível de *stress* induzido ao paciente em cada sessão de terapia. Adicionalmente, é um processo que mantém a privacidade do paciente e a confidencialidade, fatores importantes para que o paciente mantenha a frequência do tratamento.

Vários autores concordam que a utilização e a eficácia da RVTE em diversos tipos de fobias é semelhante à terapia tradicional [Grillon09; Regenbrecht06; Haworth12]. Em particular, em estudos relativos à fobia de atravessar túneis, encontramos conclusões semelhantes [Eder09; Gotestam09; Mühlberger07; Mühlberger08].

Os equipamentos de RV, mais ou menos caros, como HMD (*Head-Mounted-Displays*) ou CAVE são a escolha natural para a aplicação deste tipo de terapia. Contudo, alguns utilizadores que experimentam estes equipamentos revelam algum desconforto, como náuseas ou enjoos. Este é o motivo principal que leva a pensar aparatos alternativos para a aplicação da terapia, como as projeções em ecrã grande, ou os monitores comuns, invocando a “Imaginação” referida por Burdea et al. como um dos três I’s da RV, juntamente com a “Imersão” e a “Interação” [Burdea03].

Muitas vezes, para um paciente fóbico até mesmo situações com baixo realismo e presença são suficientes para desencadear alguma forma de ansiedade. Isto acontece porque os pacientes fóbicos são extremamente sensíveis, daí que ao mínimo estímulo a ansiedade seja desencadeada [Medeiros06]. Este é um dos motivos que nos leva a pensar que não seja necessário que a RVTE seja exclusivamente usada com equipamentos de RV imersiva, e que possa ser utilizado por terapeutas no seu gabinete com o recurso a equipamento vulgar.

A abordagem descrita neste artigo, concretizada numa aplicação denominada IN2TUNNEL, destina-se a apoiar o trabalho do terapeuta no tratamento dos pacientes com fobia de túneis. Esta aplicação oferece animações de cenários virtuais que recriam travessias de túnel, acompanhadas por sons reais. Os cenários desenvolvidos foram pensados de forma a conterem potenciais indutores de ansiedade, que variam de cenário para cenário, permitindo assim ao terapeuta fazer uma gradação de níveis de intensidade de *stress*.

As principais vantagens do uso destas animações em ambiente virtual por oposição a filmes gravados dentro de

um túnel real são a possibilidade de controlar exatamente as características desejadas em cada cenário e introduzir os diferentes níveis de dificuldade e *stress* que permitem a gradação de ambientes. A equipa deste trabalho incluiu psicólogos que identificaram as características indutoras desse *stress*.

Neste artigo apresentamos também um estudo com utilizadores. Com um grupo de terapeutas avaliámos a utilidade potencial da aplicação no domínio da RVTE; com este mesmo grupo e com um grupo de participantes voluntários comparámos a sensação de presença quando as animações são observadas numa projeção de grande dimensão (2m largura x 1,125m altura) e num equipamento de Realidade Virtual imersiva. Neste estudo não tivemos acesso a pacientes diagnosticados com este tipo particular de fobia.

No seguimento deste documento é possível encontrar na secção 2 a descrição de alguns trabalhos já desenvolvidos relacionados com esta temática, na secção 3 os pormenores do desenvolvimento da aplicação IN2TUNNEL, na secção 4 os testes com utilizadores e por fim na última secção, as conclusões e o trabalho futuro.

2. TRABALHO RELACIONADO

Existe um número considerável de aplicações e abordagens RVTE em áreas distintas, por exemplo, para o tratamento de acrofobia (medo de alturas), para o medo de voar [Mühlberger03], para o medo de falar em público [Cláudio13; Cláudio14; Grillon09], para o medo de conduzir, entre outros também referenciados [Haworth12; Grillon09; Regenbrecht06].

Alguns autores estudaram em particular a aplicação da RVTE no tratamento do medo de circular em túneis.

Em 2002 Wauke et al. aplicaram RV no tratamento de fobias nos grandes centros urbanos, sendo que uma das fobias analisada foi a fobia de túneis [Wauke02; Wauke05]. Os cenários incluíam túneis vazios e túneis com tráfego intenso e um espelho retrovisor mostrando o que acontecia atrás do veículo do paciente. Neste caso destaca-se um pormenor interessante, a presença de um botão de saída de emergência que permite ao paciente em caso de pânico abandonar imediatamente o cenário virtual.

O simulador foi testado com os óculos de RV e um monitor, sendo os melhores resultados obtidos com a primeira abordagem. Entre os participantes no teste, aqueles que tinham um diagnóstico de transtornos intensos de ansiedade, sentiram desejo de fugir, forte desconforto físico, como taquicardia, sendo incapazes de terminar a experiência. Os utilizadores que não tinham diagnóstico de transtorno de ansiedade sentiram ansiedade e impaciência durante o período de exposição e alguns também não conseguiram terminar a experiência.

Em 2006, Gustavo de Medeiros voltou a abordar o tema com mais detalhe passando a haver dois tipos de túneis, com e sem tráfego, em que a circulação é feita num bairro e obriga a parar num sinal vermelho antes de prosseguir para o túnel [Medeiros06]. Neste caso, as curvas do túnel eram demasiado pronunciadas, mas tinha um cenário em

que havia demasiado tráfego devido a uma obstrução de um veículo pesado. Este autor optou pelo uso de modelos em VRML e um HMD, no entanto, os seus modelos não eram fáceis de navegar daí que os eventos na aplicação sejam uma sequência de eventos fixos, controlados por um terapeuta e não livremente conduzidos pelo paciente.

Mühlberger et al. usaram três cenários de condução distintos: um ambiente aberto, um túnel parcialmente aberto e um túnel fechado. Estes cenários foram testados por 15 pessoas com medo de túneis e 15 participantes sem medo. Neste caso os autores concluíram que os ambientes virtuais são uma ferramenta importante para a avaliação de reações de medo e que devem ser usados em pesquisa experimental [Mühlberger07].

O estudo realizado por estes autores pretendia verificar se haveria alguma diferença na reação dos pacientes de acordo com a variação de luz no interior do túnel [Mühlberger08]. O nível de *stress* induzido pela experiência foi medido usando o reflexo de *Startle* que é uma resposta defensiva a estímulos súbitos ou ameaçadores, sendo assim associado a um efeito negativo. O estado emocional em que se encontra o indivíduo faz variar o tipo de resposta que dá ao estímulo negativo.

Os túneis virtuais usados, cuja principal característica é a variação de luz dentro do túnel, têm cerca de 10 km e velocidade máxima de circulação de 80 km/h. Foram utilizados por pacientes que sofrem de fobias de túneis ou claustrofobia. Os resultados do estudo indicam que os reflexos de um paciente que viaja ao lado do condutor são mais intensos quando o túnel é escuro, muito provavelmente porque o passageiro não tem controlo sobre os comandos de condução.

O principal objetivo do nosso trabalho foi desenvolver uma ferramenta de fácil utilização, eficaz e que possa fazer uso de equipamento não dispendioso. Deste modo será possível ao terapeuta realizar no seu consultório uma terapia de exposição para medo de conduzir em túneis. IN2TUNNEL é uma aplicação Web que oferece ao terapeuta um conjunto de animações contendo diferentes indutores de *stress* de modo a permitir a evolução gradual da terapia. Além disso, permite guardar dados do paciente numa base de dados local para evitar problemas quer éticos quer de segurança.

3. A APLICAÇÃO IN2TUNNEL

A aplicação IN2TUNNEL destina-se a ser usada como suporte na terapia de exposição. Os utilizadores desta aplicação são o terapeuta e o paciente, sendo que cada um deles desempenha papéis distintos. Enquanto o primeiro deve escolher o cenário apropriado para a sessão com o paciente, observar as reações deste e registá-las na aplicação, o segundo deve apenas observar as animações e dar o seu *feedback* ao terapeuta.

Os cenários construídos foram analisados em conjunto com os psicólogos que integram a equipa de trabalho. Optou-se por colocar o paciente como um utilizador

passivo, sendo conduzido através do túnel, no lugar ao lado do condutor, não podendo assim tomar qualquer decisão relativamente à velocidade exercida ou a qualquer outro fator da condução. Os cenários foram selecionados de modo a causarem impacto no paciente.

Escolheram-se dois modelos base: um túnel de montanha e outro de cidade. O primeiro é um túnel com uma faixa em cada sentido, estreito e sem saídas de emergência, enquanto o túnel de cidade é um túnel amplo, com duas faixas em cada sentido, saídas de emergência e que tem como foco principal a presença de um carro parado na via da esquerda. Com o intuito de criar diferentes níveis de *stress* criámos algumas variantes nos túneis. Existem assim túneis de montanha com e sem curvas, com e sem outros veículos dentro do túnel, com veículos que impedem que se veja a saída do túnel. Fizemos também variar, entre o dia e a noite, a altura do dia em que se simulam as entradas e saídas dos túneis. Foram também gravados sons reais dentro de um veículo atravessando um túnel com música a tocar no hipotético rádio, gravando-se vários estilos de música.

Logo desde o início do projeto um dos principais objetivos foi criar uma solução de RVTE que não recorra necessariamente a equipamentos dispendiosos, de modo a que pudesse ser utilizada pelo terapeuta numa sessão no seu consultório. Por outro lado, os custos de desenvolvimento também foram minimizados recorrendo a software gratuito, bem como a modelos disponíveis *online*. Para a produção das animações e criação dos cenários foi utilizado o Blender³ muito utilizado para a criação de modelos 3D e videojogos. Explica-se em seguida como o terapeuta interage com a aplicação e como procedemos à criação dos cenários e das animações.

3.1 A Interface

Quando o terapeuta inicia a aplicação são abertas simultaneamente duas janelas que devem ser usadas em separado. A primeira janela deve ser visionada apenas pelo terapeuta (por exemplo no ecrã do seu portátil), pois é nela que lhe é permitido ter acesso à base de dados dos seus pacientes, onde regista as reações dos mesmos aos cenários a que é exposto. A segunda janela deve ser visionada num outro ecrã ou na tela de projeção visível pelo paciente e pelo terapeuta, e nela são projetados os cenários das simulações. Deste modo o paciente é exposto à simulação mas sem visualizar a informação restrita ao terapeuta.

Numa sessão de terapia é suposto que o paciente visiona os fotogramas dos vídeos e os vídeos propostos pelo terapeuta, enquanto este o confronta sobre os estímulos que os cenários têm sobre ele. No decorrer da sessão o terapeuta pode analisar diferentes vídeos e fotogramas e tomar notas sobre as reações do paciente, através da interface da aplicação com campos de texto editáveis. O terapeuta dispõe de uma lista de pacientes e ao escolher o

³ <http://www.blender.com> (último acesso set/2014)

nome de um paciente aparecem as ações disponíveis: a edição dos dados pessoais, como morada e número de telefone, e de notas acerca dos fotogramas e dos vídeos já visualizados; se o terapeuta pretender eliminar um paciente é-lhe sempre pedida confirmação, para prevenir erros ocasionais. Nas páginas das notas dos fotogramas e dos vídeos, o terapeuta consegue ver uma miniatura do fotograma ou vídeo para se poder orientar sem ter de os memorizar.

O terapeuta tem um menu que permite a escolha entre o fotograma (Figura 1) e os vídeos. Na página dos vídeos é apresentada a listagem dos vídeos disponíveis com uma breve descrição do que pode ser visionado em cada um deles.

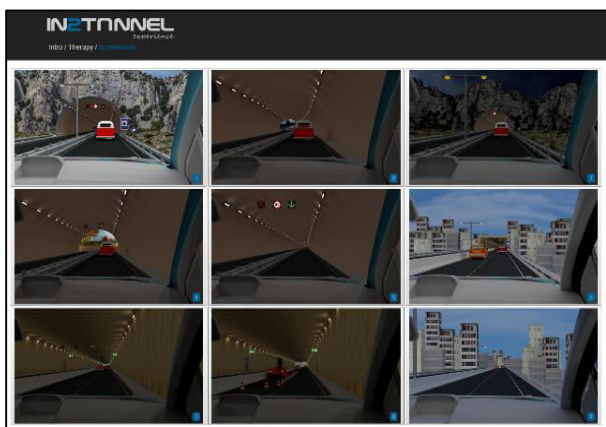


Figura 1 – Fotogramas das animações produzidas.

A interface Web da aplicação foi desenvolvida no Adobe Dreamweaver CS5. Usámos HTML5, CSS3, Javascript e JQuery. Com o aparecimento do HTML5 surgiram novas capacidades que são vantajosas quer para os programadores, quer para os utilizadores. Neste caso, decidimos utilizar a funcionalidade de bases de dados indexadas (IndexedDB), em que a cada objeto é associada uma chave unívoca e que permite guardar grandes quantidades de dados, localmente no *browser*.

A grande vantagem desta opção é o facto da não dependência do funcionamento dos servidores em que normalmente os dados são alocados, pois este passa a ser um servidor local, a própria máquina. Em comparação com a utilização das *cookies*, estas últimas implicam questões legais, têm um limite de armazenamento e além disso têm de estar sempre a estabelecer contacto com o servidor.

3.2. Os Cenários e as Animações

Os vídeos de simulação são o elemento central desta aplicação, pois são objetos relevantes na terapia de exposição. A vantagem de ter vídeos de simulação e não vídeos gravados num cenário real é que nos primeiros podemos ter características e eventos perfeitamente planeados, o que num vídeo real é, em alguns casos, potencialmente impossível de garantir, como por exemplo, a simulação de acidentes.

Uma hipótese que considerámos no início do trabalho foi a possibilidade de permitir ao terapeuta que fosse ele



Figura 2 - Entrada de noite em túnel de montanha (durante a noite).

próprio a escolher as situações e a fazer a montagem do cenário e gerar em seguida o vídeo para aquela terapia. No entanto, tal não nos pareceu exequível, pois a geração da totalidade das imagens que compõem o vídeo com a qualidade pretendida, revelou-se muito demorada, muito longe de um tempo de geração que poderia ser considerado “tempo-real”. Posto isto, o que fizemos foi gerar um número significativo de animações que

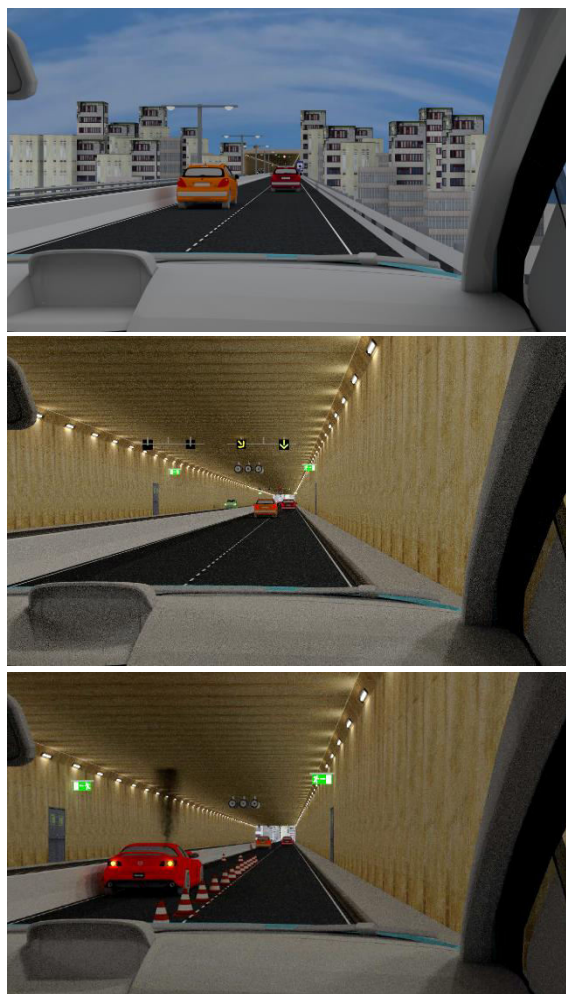


Figura 3 – Uma sequência num túnel de cidade: (de cima para baixo) entrada durante o dia, dentro do túnel, carro parado libertando fumo e bloqueando a faixa esquerda (ouve-se uma sirene de ambulância).

correspondem a simulações distintas. Com a experiência dos psicólogos presentes na equipa concluímos que deveríamos controlar fatores como a luminosidade (dentro e fora do túnel) e acima de tudo guiar o paciente obrigando-o assim a atravessar o túnel.

Os cenários contemplados na aplicação cobrem dois tipos de túneis distintos, como referimos anteriormente: um túnel de montanha e um túnel de cidade. O túnel de montanha é um túnel estreito, apenas com uma faixa em cada sentido e a única hipótese de fuga é pela entrada e pela saída (Figura 2). Por outro lado, o túnel de cidade é amplo, com duas faixas em cada sentido e saídas de emergência (Figura 3). Em ambas as situações introduzimos variantes de modo a introduzir diferentes níveis de impacto no paciente. Nos dois túneis optámos por fazer variar as entradas e saídas do túnel entre o dia e a noite (Figuras 2 e 3).

No túnel de cidade introduziu-se um obstáculo, um carro a deitar fumo parado na faixa da esquerda (Figura 3, em baixo), enquanto no túnel de montanha temos mais variantes, carro lento na frente e com uma dimensão que impossibilita a visibilidade para a frente, túnel sem qualquer movimento, túnel com e sem curvas, e túnel sem veículo à frente mas com uma curva que impossibilita ver a saída do túnel.

3.2.1. Criação dos cenários

Ambos os túneis foram totalmente modelados e texturados no *Blender* usando as técnicas de modelação 3D disponíveis, entre elas, *modeling*, *extrude* e alguns *modifiers* (*array* e *mirror*). Todas as texturas presentes foram retiradas quer da biblioteca CG Textures⁴ como da Plain Textures⁵, com exceção da sinalização que foi retirada de um documento da legislação da União Europeia⁶. Quer os veículos, quer os humanos virtuais que simulam os condutores das viaturas, foram igualmente retirados da Internet.

No túnel de cidade, o cenário exterior, quer à entrada, quer à saída do túnel é um modelo 3D, no entanto, o mesmo não acontece no túnel de montanha. A modelação das montanhas era um fator que iria pesar muito na fase de *rendering*, portanto optámos por modelar a montanha através de texturas.

Quase todos os cenários foram gerados “frame a frame” com o motor Cycles Render. A única exceção é a geração de fumo, uma vez que quando o trabalho foi iniciado a versão do *Blender* disponível (2.68) não o permitia. Daí que os cenários do túnel de montanha tenham sido feitos através do *Node Editor* do *Blender* com o *Composite*, juntando depois o cenário renderizado no *Cycles Render* e o fumo renderizado pelo *Blender Render*.

3.2.2 Criação das animações

Cada animação é composta por 3000 frames e tem a duração aproximada de 1:40 minutos. Depois da renderização para frames em formato PNG, são todas colocadas no *Blender* Video editor e feito um novo *rendering* no formato H.264 que permite converter todas as *frames* para uma sequência de vídeo. Este vídeo é gerado no formato AVI, no entanto, para ser posteriormente colocado na aplicação web, deve ser novamente convertido, desta vez para o formato .MP4.

Através do *Adobe Premiere Pro CS5* conseguimos incluir os sons, bem como os efeitos de *fade in* e *fade out* nas animações depois de convertido o vídeo para .MP4, pois o formato feito pelo *Blender* (.AVI) não é suportado por este software. Depois de termos a animação já com estes elementos convertemos novamente o vídeo de .MPEG para .MP4 no *Miro Video Converter*⁷.

O equipamento usado neste trabalho foi um computador com um processador AMD A8 5600K 3.6GHz; 8GB de memória RAM; placa gráfica GeForce GTX 760 2GB.

3.3. Aparatos Usados para Visualizar as Animações

Nos testes que realizámos, descritos na secção seguinte, os vídeos foram visualizados em dois aparatos distintos, ambos numa sala escura. O primeiro aparato é composto por um projetor de vídeo e uma tela ou parede branca; a imagem projetada tem cerca de 2m largura e 1,125m altura (16x9) e é observada a uma distância aproximada de 1,5 metros.

O segundo aparato usa um *smartphone* como dispositivo de geração de duas imagens e um dispositivo que o suporta para que possa ser usado como um equipamento imersivo de RV. O suporte usado é um Durovis Dive⁸ (Figura 4). O *smartphone* é colocado horizontalmente no suporte e este dispõe de um par de lentes ajustáveis, uma lente em frente de cada imagem. Deste modo cada olho do observador visualiza uma imagem e cria-se assim o efeito 3D. Este equipamento tem a principal vantagem de ser relativamente barato, dado que, como nos dias de hoje, praticamente todas as pessoas possuem um *smartphone* já



Figura 4: Equipamento Durovis. Imagem dupla no *Smartphone* e as lentes (à esquerda). Usando o equipamento imersivo (à direita).

⁴ www.cgtextures.com (último acesso set/2014)

⁵ www.plaintextures.com (último acesso set/2014)

⁶ eur-lex.europa.eu/resource.html?uri=cellar:00de7851-bf21-4eec-9621-626931ad9df5.0009.01/DOC_5&format=PDF (último acesso set/2014)

⁷ <http://www.mirovideoconverter.com/> (último acesso set/2014)

⁸ <http://www.durovis.com> (último acesso set/2014)

compatível com o mesmo, só é necessário adquirir o próprio suporte.

Nos nossos testes usámos um *smartphone* Galaxy S4 – QuadCore 1.9 GHz e um ecrã de 5”⁹. Para poder usar este equipamento é necessário instalar no *smartphone* a aplicação Side-By-Side Video¹⁰, de modo que seja possível visualizar dois vídeos lado a lado, um para cada olho.

Tendo em conta que o ajustamento das lentes pode ser um processo moroso, optámos por ter, no início de cada vídeo, a imagem parada durante 1 minuto para que cada pessoa tenha tempo suficiente de focar e depois ficar apenas concentrada a observar o vídeo. Os utilizadores que usam óculos graduados têm de os retirar para poder usar o dispositivo.

Note-se que optámos também por manter a sala escura já que o suporte tem algumas aberturas pelas quais entra luz ambiente, nomeadamente na zona junto ao nariz e nos encaixes laterais onde se procede ao ajuste das lentes.

4. TESTES COM UTILIZADORES

Os testes com utilizadores envolveram dois grupos de participantes: terapeutas e participantes voluntários. Os primeiros testaram a aplicação na qualidade de peritos e compararam a sensação de presença nos dois tipos de aparatos; os voluntários apenas realizaram esta comparação.

Durante os testes o entrevistador estava sentado junto dos participantes, tendo colocado questões aos utilizadores antes e depois de realizada a experiência. Além das respostas aos questionários, o entrevistador registou as opiniões e as sugestões dos participantes para serem tidas em conta em melhorias futuras da aplicação.

4.1 Testes com Terapeutas

O primeiro teste com terapeutas teve o objetivo de avaliar a utilidade potencial da aplicação no domínio da RVTE para o tratamento do medo de atravessar túneis. O segundo destinou-se a comparar a sensação de presença dentro do túnel virtual quando as animações são observadas numa projeção grande e num equipamento de Realidade Virtual imersiva, nos aparatos descritos anteriormente.

Participaram 4 terapeutas (2 Homens, de 43 e 60 anos, 2 Mulheres, de 39 e 43 anos), todos familiarizados com o uso da terapia de exposição. Dois destes participantes são terapeutas e também investigadores e os restantes exercem exclusivamente a profissão de terapeutas. Todos os terapeutas se sentiam antes dos testes confortáveis com a ideia de usar a RV para apoiar a terapia de exposição.

A avaliação foi dividida em duas fases: uma primeira de familiarização com a aplicação em que se pretendeu avaliar a interface propriamente dita (facilidade e clareza de utilização); e uma segunda em que solicitámos aos psicólogos que usassem a aplicação para conceber duas

sessões de terapia supondo que teriam de tratar um paciente com um caso grave de medo de atravessar túneis. Foi-lhes solicitado que uma das sessões correspondesse a uma sessão no início do tratamento e outra mais próxima do final. Deste modo, o psicólogo pode naturalmente fazer variar o nível de ansiedade causado sobre o paciente. No final os utilizadores expressaram as suas opiniões num conjunto de respostas abertas.

No que diz respeito à facilidade de uso da interface no geral as opiniões foram muito positivas. Depois de experimentarem a aplicação realizando a tarefa proposta pelo entrevistador, os terapeutas consideraram unanimemente que a ferramenta é adequada para a terapia de exposição, realçando a ausência de risco para o paciente e a confidencialidade do tratamento, dois fatores que podem motivar indivíduos com este tipo de fobia a procurar auxílio terapêutico e a levar a cabo uma terapia até ao final.

Três dos terapeutas pensam que a utilização da ferramenta nos passos intermédios da terapia pode permitir reduzir a exposição real a apenas uma sessão final. Um dos terapeutas considerou que o uso desta ferramenta poderia perfeitamente substituir por completo as sessões *in vivo* durante toda a terapia. Todos se mostraram recetivos à ideia de usarem a ferramenta neste tipo de terapia.

Todos os terapeutas participantes usam óculos no seu dia-a-dia e no segundo teste, conseguiram focar a imagem no dispositivo de RV imersiva, alguns com mais dificuldade que outros. Apenas um dos terapeutas referiu algum desconforto quanto ao peso do equipamento.

Classificando a sensação de presença dentro do mundo virtual usando equipamento de RV, três dos terapeutas deram, numa escala de 1-*pouco* a 5-*muito*, o valor 4 e um deles deu o valor 5. Todos consideraram que a sensação de presença é mais intensa com este equipamento do que com a projeção na tela.

A dimensão da imagem projetada e a distância a que foi visualizada foram unanimemente consideradas adequadas.

4.2 Testes com Participantes Voluntários

Os testes com pacientes voluntários envolveram um total de 16 participantes, 5 mulheres (25, 32, 36, 43 e 52 anos – média de 39 anos) e 11 homens (6 deles com idades entre os 22 e os 32 anos e os restantes 5 com idades entre os 34 e 46 anos – média de 30 anos).

Tendo em conta que não tivemos acesso a pacientes diagnosticados com o tipo particular de fobia em estudo, decidimos incluir um questionário inicial, respondido antes da visualização das simulações, para tentar aferir o seu nível de medo de túneis. Todos os utilizadores foram questionados sobre a regularidade com que conduzem e se alguma vez tinham experimentado outro caminho apenas para não passar num túnel. Todas as outras

⁹ <http://www.samsung.com/br/consumer/cellular-phone-tablets/smartphones/GT-I9505ZWLZTA-spec> (último acesso set/2014)

¹⁰ https://play.google.com/store/apps/details?id=com.funghisoft.sbsvideoplayer&hl=pt_PT (último acesso set/2014)

questões foram classificadas usando uma escala de 1-*fácil* a 5-*difícil* e envolvem as seguintes situações e características do túnel:

- Paragem forçada dentro do túnel
- Atravessar um túnel estreito
- Atravessar um túnel pouco iluminado
- Entrar num túnel cuja saída não esteja visível

Apenas uma participante expressou verbalmente, e antes mesmo de responder a estas primeiras questões, que sofre de um grau elevado de ansiedade quando atravessa túneis. Esta participante conduz frequentemente e, sente um enorme desconforto quando é necessário parar dentro do túnel, quando este é pouco iluminado ou muito estreito e ainda se a saída do túnel não for visível. Contudo, nenhum destes motivos a levou a escolher um caminho alternativo. Outros dois participantes dizem sofrer de um nível moderado de ansiedade em túneis. Os restantes não têm qualquer tipo de desconforto e responderam a todas as perguntas enumeradas antes com a classificação de “fácil”. Os participantes foram também questionados sobre a importância que davam à presença de saídas de emergência num túnel.

Entre os participantes, 11 consideram “muito importante” ou “importante” a existência de saídas de emergência, 2 participantes classificaram como “nada importante” e os restantes 3 deram valores intermédios de importância.

Pressupondo que a qualidade das animações também pode influenciar as opiniões dos participantes quanto aos aparatos usados na visualização, solicitámos que classificassem as animações quanto:

- à semelhança dos cenários com a realidade,
- à semelhança dos sons com os reais,
- ao aspeto credível das animações.

As respostas foram dadas numa escala de 1 a 5, em que 1 é “mau” e 5 é “excelente”. Os resultados estão resumidos no gráfico da Figura 5 e é possível verificar que a classificação mínima obtida foi 3. As classificações são bastante favoráveis, o que nos leva a pensar que a qualidade das animações não poderá ter pesado negativamente nos testes relativos à sensação de presença, cujos resultados apresentamos de seguida.

Todos os participantes exceto 1 (93%) consideraram que o equipamento de RV imersiva gera uma maior sensação de presença pois classificam-no como sendo muito mais imersivo do que a imagem projetada. Revelaram

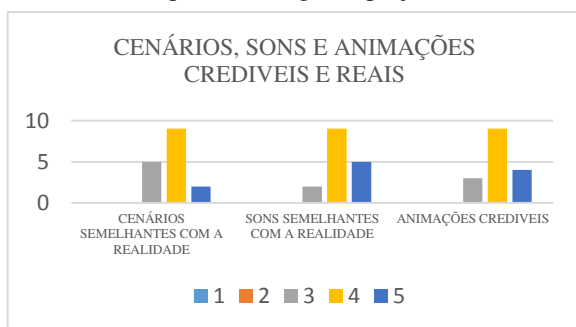


Figura 5 – Classificação das animações (voluntários).

dificuldade em focar a imagem 12 participantes (87%) sendo que 10 usam óculos graduados; 80% dos participantes revelou também ter sentido algum desconforto com o equipamento de RV imersivo (alguns pelo peso do equipamento, outros pelo calor provocado na cara) e 1 participante revelou ter-se sentido enjoado ao experienciar as animações também através do equipamento de RV. Toda a informação relativa à utilização do equipamento imersivo está compactada no gráfico da Figura 6.



Figura 6 – Desconforto na utilização do equipamento de RV.

Todos os participantes tiveram a sensação 3D no equipamento imersivo; no ecrã, apenas a participante que assumiu verbalmente o seu medo de túneis afirmou ter tido essa perceção. De todos os participantes, 11 referiram que a projeção deveria ser maior, enquanto os restantes pensam que o tamanho é adequado, mas que deveria o observador dever estar mais próximo estar posicionado mais perto da projeção.

4.3 Discussão

Apesar da dimensão limitada do conjunto de participantes, é possível observar uma clara tendência: a sensação de presença sentida pelo observador é maior com o equipamento de RV imersiva, apesar de na opinião geral o uso do equipamento não ter sido considerado confortável. É difícil obter uma imagem perfeitamente focada, especialmente para pessoas que usam óculos graduados. O equipamento tem aberturas que permitem a entrada de luz exterior, duas laterais para o ajuste das lentes e uma inferior junto ao nariz. Consideramos que este pode ser um fator de distração do mundo virtual, impedindo uma total abstração da realidade. Para colmatar este facto a aplicação foi testada numa sala escura; mesmo assim, 6 dos 20 participantes (30%) incluindo terapeutas e voluntários, notaram entrada de luz no equipamento de RV.

Numa próxima fase, e de modo a tirarmos conclusões sobre a utilidade da aplicação para pessoas que sofrem de fobia de atravessar túneis, esta deveria ser testada por pessoas já diagnosticadas de forma a termos uma noção do impacto produzido neste caso.

5. CONCLUSÕES E TRABALHO FUTURO

Neste artigo apresentamos uma aplicação web, denominada IN2TUNNEL, que apoia o terapeuta no uso da RVTE em pacientes com medo de atravessar túneis. O terapeuta dispõe de um conjunto de animações de cenários virtuais que recriam travessias de túneis, acompanhadas por sons reais. As animações foram concebidas de forma a conterem características ou eventos potencialmente indutores de ansiedade, que variam de cenário para cenário, permitindo que o terapeuta possa fazer uma gradação de níveis de intensidade de *stress* ao longo de um tratamento terapêutico. A interface da aplicação dispõe também de um conjunto de funcionalidades de apoio ao registo de informações sobre o paciente e que são guardadas na própria máquina do terapeuta.

Descrevemos ainda um estudo com terapeutas e participantes voluntários. Os terapeutas deram um retorno global positivo acerca da utilidade da ferramenta em terapia e da sua interface com o utilizador. Quanto à comparação da sensação de presença dentro do túnel virtual quando as animações são observadas numa projeção grande e num equipamento de RV imersiva que envolve um *smartphone* e um suporte com duas lentes, os resultados mostram uma preferência clara por este último equipamento. Foi no entanto identificada alguma dificuldade em obter uma imagem perfeitamente focada, tendo sido manifestado um certo grau de desconforto na sua utilização.

No futuro próximo temos a intenção de repetir estes testes com um HMD com a expectativa de observar uma menor dificuldade em focar a imagem. Gostaríamos de levar a cabo testes com uma amostra de pacientes reais, diagnosticados com medo de túneis. Só assim se podem tirar conclusões sobre a efetiva utilidade da aplicação desenvolvida. Se tivermos acesso a uma amostra com estas características, será igualmente interessante comprovar a convicção expressa pelos 4 terapeutas de que para pacientes com fobia, uma animação visualizada num equipamento vulgar não imersivo, como um monitor de computador, é já suficiente para desencadear algum nível de ansiedade e servir de apoio a uma sessão em consultório. Esta convicção parece-nos perfeitamente razoável quando pensamos que os terapeutas recorrem com frequência no início dos tratamentos à técnica *in imagino* em que o paciente simplesmente imagina a situação que lhe causa ansiedade ou medo. Comprovar este facto pode ser um fator importante para a disseminação do uso por parte dos terapeutas da RVTE para este e outros tipos de fobia.

6. AGRADECIMENTOS

Agradecemos à Fundação para a Ciência e Tecnologia (FCT) e ao laboratório de I&D LabMag o apoio financeiro dado para a realização deste trabalho, ao abrigo do projeto estratégico PEst-OE/EEI/UI0434/2014. Agradecemos ao Professor Francisco Esteves o acompanhamento inicial do trabalho e também a todos os participantes nos testes.

7. REFERÊNCIAS

- [Burdea03] Burdea, Grigore C., & Coiffet, Philippe, (2003). *Virtual Reality Technology*, 2nd ed. Wiley-Interscience.,
- [Cláudio13] Cláudio, A. P., Carmo, M.B., Pinheiro, T., Esteves, F. (2013). A Virtual Reality Solution to Handle Social Anxiety. *International Journal of Creative Interfaces and Computer Graphics* 12/2013; 4(2):57-72.
- [Cláudio14] Cláudio, A. P., Gaspar, A., Lopes, E., Carmo, M.B. (2014). Characters with Affective Facial Behavior. In *Proceedings of GRAPP 2014- International Conference on Computer Graphics Theory and Applications*, pp 348-355, Lisboa.
- [DSM-IV] DSM-IV, 2000- American Psychiatric Association. *Diagnostic and statistical manual of mental disorders: DSM-IV-TR*. American Psychiatric Publishing, Inc.
- [Eder09] Eder, S., Brutting, J., Muhlberger, A., & Pauli, P. (2009). Human Behaviour in Tunnel Accidents. In *Proc of the 4th International Symposium Human Behaviour in Fire*.
- [Gotestam09] Gotestam, K. G., & Svebak, S. (2009). Treatment of tunnel phobia: an experimental field study. *Cognitive behaviour therapy*, 38(3), 146–52.
- [Grillon09] Grillon, H. (2009). *Simulating interactions with virtual characters for the treatment of social phobia* (Doctoral dissertation, École Polytechnique Fédérale de Lausanne).
- [Haworth12] Haworth, M. B., Baljko, M., & Faloutsos, P. (2012, December). PhoVR: a virtual reality system to treat phobias. In *Proc of the 11th ACM SIGGRAPH Int Conf on VR Continuum and its Applications in Industry* (pp. 171-174). ACM.
- [Medeiros06] Medeiros, G. A. (2006). *Sistema de Realidade Virtual para Tratamento de Fobias*. Vasa. Obtido em <http://medcontent.metapress.com/index/A65RM03P4874243N.pdf>
- [Mühlberger03] Mühlberger, A.; Wiedemann, G.; Pauli, P., (2003), “Efficacy of a one-session virtual reality exposure treatment for fear of flying”, *Psychotherapy Research*, n. 13(3), pp 326-336.
- [Mühlberger07] Mühlberger, A.; Bühlhoff, H. H.; Wiedemann, G.; Pauli, P., (2007). Virtual reality for the psychophysiological assessment of phobic fear: Responses during virtual tunnel driving. *Psychological Assessment*, Vol 19(3), 340-346.
- [Mühlberger08] Mühlberger, A., Wieser, M. J., & Pauli, P. (2008). Darkness-enhanced startle responses in ecologically valid environments: a virtual tunnel driving experiment. *Biological psychology*, 77(1), 47–52.
- [Regenbrecht06] Regenbrecht, H., Wickerth, D., Dixon, B., & Mueller, S. (2006, November). Collaborative mixed reality exposure therapy. In *Proc of the IEEE Int Conf on Cyberworlds*, 25-32, Lausanne.
- [Wauke02] Wauke, A., Costa, R., & Carvalho, L.. (2002). *A Realidade Virtual no Tratamento de Fobias em Grandes Centros Urbanos*. Obtido em <http://www.lbd.dcc.ufmg.br/colecoes/wim/2002/0015.pdf> (último acesso set/2014)
- [Wauke04] Wauke, A. T., Costa, R., Carvalho, L., (2004). "VESUP: O uso de Ambientes Virtuais no tratamento de Fobias Urbanas." IX Congresso Brasileiro de Informática em Saúde, Ribeirão Preto, SP, Brasil.

LS3D: Pesquisa de Blocos LEGO em Ambientes Imersivos

Pedro B. Pascoal Daniel Mendes Diogo Henriques Isabel Trancoso Alfredo Ferreira
Departamento Engenharia Informática

INESC-ID / IST / Universidade de Lisboa

pmbp@tecnico.ulisboa.pt, danielmendes@tecnico.ulisboa.pt, diogo.henriques@tecnico.ulisboa.pt,
isabel.trancoso@inesc-id.pt, alfredo.ferreira@inesc-id.pt

Resumo

A crescente popularidade dos ambientes imersivos tem aumentado o interesse em interações pessoa-máquina mais naturais. Contudo, no contexto de seleção de objetos e construção de modelos virtuais, o uso de ambientes imersivos tem sido pouco explorado. Neste trabalho apresentamos o LS3D, um protótipo que aborda este desafio, usando fala e gestos como forma de interação, desenvolvido para o cenário de construção com blocos LEGO. Neste âmbito, foram desenvolvidas novas abordagens para explorar resultados de uma pesquisa neste tipo de ambientes, que permitem uma melhor interpretação dos resultados e que oferecem uma interação direta aos utilizadores. Para avaliar a nossa solução, comparámos o LS3D com a aplicação comercial LEGO Digital Designer. Os resultados sugerem que a nossa proposta oferece uma forma mais simples e natural para procurar objetos 3D virtuais, garantindo também melhores desempenhos e percepção dos resultados do que abordagens tradicionais para a recuperação deste tipo de conteúdo. Os participantes afirmaram ainda que a nossa solução oferece uma experiência mais divertida e enriquecedora.

Palavras-Chave

Interfaces multi-modais, interação natural, fala, gestos, descrição de objetos 3D, testes com utilizadores

1. INTRODUÇÃO

Os ambientes virtuais utilizados em aplicações de entretenimento estão a tornar-se vez mais imersivos [Trowbridge 09]. Recorrendo a CAVE [Jacobson 05] e *Heads-Mounted Displays* (por exemplo Oculus Rift¹ ou o recentemente anunciado Project Morpheus da Sony), este tipo de ambientes é cada vez mais comum. Diversas áreas podem tirar partido da capacidade de colocar o utilizador dentro do mundo virtual de uma forma mais imersiva em relação ao que as configurações tradicionais permitem, sendo possível oferecer interações mais naturais com o conteúdo virtual. Um dos exemplos destas aplicações é a criação de cenários virtuais [Cabral 11], que permitam ao utilizador posicionar objetos virtuais na cena, simulando interações do mundo físico.

No entanto, quando confrontados com o desafio de seleção de objetos em coleções, as soluções tradicionais baseiam-se normalmente em listas ou grelhas com vistas 2D dos objetos, o que pode não ser viável dada a mudança no paradigma de interação. Por outro lado, o uso de ambientes imersivos demonstra melhorias na visualização e exploração dos resultados de recuperação relativamente a abordagens tradicionais [Henriques 14, Pascoal 12]. Neste trabalho, focamos em formas de exploração de objetos 3D em grandes coleções através de ambientes imersivos,

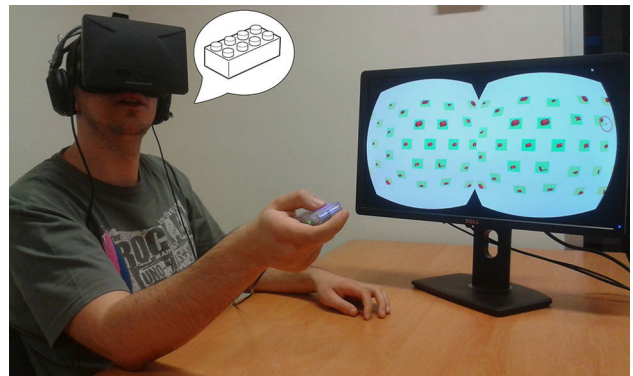


Figura 1. Utilizador a interagir com o protótipo LS3D.

com o propósito de selecionar um modelo específico. Para isso, usámos como cenário blocos de construção LEGO, que já provou ser um bom caso de estudo, que já originou aplicações de entretenimento interessantes [Santos 08, Mendes 11, Miller 12, Gupta 12]. Desenvolvemos um protótipo, o LS3D, que permite aos utilizadores pesquisar, selecionar, explorar e colocar blocos LEGO 3D num ambiente totalmente imersivo através do uso de interações multimodais, como ilustrado na Figura 1.

¹Oculus Rift: <http://www.oculusvr.com/rift/>

Ao longo do documento iremos discutir o estado da arte relativamente a pesquisa de conteúdo multimédia, com foco especial em objetos 3D virtuais. Procederemos à descrição da solução implementada no nosso protótipo, assim como da avaliação onde o comparamos com uma aplicação comercial. Finalmente, apresentamos as nossas conclusões e indicamos algumas direções para trabalho futuro.

2. TRABALHO RELACIONADO

Com o rápido aumento da informação multimédia disponível, a pesquisa e recuperação de um conteúdo específico tem-se apresentado como um desafio cada vez mais importante. As abordagens tradicionais permitem a recuperação recorrendo à utilização de pesquisas textuais. Contudo, este método não é trivial, visto que normalmente conteúdo multimédia não contém informação intrínseca suficiente. De forma geral, os motores de pesquisa fazem uso de anotações, tais como o nome dos ficheiros, legendas, ou mesmo referências associadas, que podem não estar diretamente relacionadas com conteúdo de média. Este conceito também se aplica na recuperação de imagens [Smith 97] e de objetos 3D [Funkhouser 03]. Funkhouser et al., faz uso de sinónimos para aumentar as palavras possíveis para descrever objetos 3D e minimizar a escassez do vocabulário.

Apesar dos trabalhos existentes, as descrições disponíveis para objetos 3D continuam a ser insuficientes, em especial relativamente à geometria dos objetos. Para facilitar, algumas das soluções propostas fazem uso da pesquisa-por-exemplo. O objetivo da pesquisa-por-exemplo é a obtenção de resultados semelhantes em termos do aspecto visual, tal como a cor [Paquet 99] ou forma [Laga 11]. Contudo, este método de interrogação exige que o utilizador tenha *a priori* um objeto semelhante ao procurado, o que geralmente não é o caso.

Uma forma de contornar essa limitação, é usar como interrogação um esboço do objeto pretendido, oferecendo aos utilizadores a possibilidade de pesquisarem objetos semelhantes aos esboços produzidos [Pu 05]. No trabalho de Santos et al. [Santos 08], os utilizadores realizam esboços que correspondem às dimensões de um determinado bloco LEGO desejado. Funkhouser et al. [Funkhouser 03] também apresenta um método de desenho, que permite aos utilizadores desenharem até três vistas 2D diferentes do modelo procurado, como forma de minimizarem o número de falsos positivos. Uma abordagem diferente é apresentada por Liu et al. [Liu 13] que tentou melhorar a pesquisa por esboços usando o perfil do utilizador, tendo em conta os hábitos do utilizador quando este desenha. Isso levou a melhorias nos resultados dos utilizadores que realizavam mais pesquisas.

Holz e Wilson [Holz 11] apresentam uma abordagem diferente, onde a descrição da interrogação é feita através de gestos. Este trabalho consiste na captura dos gestos dos utilizadores, usando a forma e a movimentação das mãos para criar esboços tridimensionais. Os autores concluíram que os participantes conseguiam manter as proporções dos objetos físicos relativamente corretas e que, em zonas com

mais detalhes, realizavam os gestos de forma mais lenta.

No entanto, a apresentação dos resultados nestes trabalhos foca-se principalmente no uso da abordagem tradicional de listas de vistas. Nakazako et al. [Nakazato 01] apresentaram o 3D MARS, que demonstra os benefícios no uso de ambientes imersivos no contexto da recuperação de conteúdo multimédia. O seu trabalho centra-se essencialmente na forma como são apresentados os resultados de pesquisas em sistemas de recuperação de imagens por conteúdo, usando uma CAVE. Estendendo a abordagem do MARS 3D a objetos 3D, Pascoal et al. [Pascoal 12] demonstram que alguns desafios na recuperação de objetos 3D podem ser superados ao apresentar os resultados em ambientes imersivos. Em ambas soluções, os resultados são distribuídos no espaço virtual de acordo com a sua semelhança face à interrogação. Após cada pesquisa, os utilizadores podem explorar os resultados, navegando no ambiente envolvente através do uso de dispositivos como *Head-Mounted Display's*. É também possível o uso diversificado de outros dispositivos de visualização e interação, que foram utilizados para testar vários paradigmas de interação para a recuperação de objetos 3D.

Recentemente, na área da recuperação de informação, temos testemunhado o aparecimento de um vasto leque de soluções com pesquisa-por-fala, especialmente em dispositivos móveis. Isto levou à preferência desta modalidade à tradicional de pesquisa por texto [Kamvar 10]. Na maioria dos casos, na procura por fala, a voz é convertida em texto, o qual é então utilizado como um parâmetro de pesquisa [Wang 11, Lee 10]. Mais recentemente, Lee e Kawahara [Lee 12] realizaram uma análise semântica das pesquisas por fala usadas para procurar livros, de forma a alcançar um maior entendimento sobre o elemento desejado.

Em conclusão, a pesquisa de conteúdos multimédia, nomeadamente de objetos tridimensionais, já foi alvo de estudos anteriores. Contudo, embora a maioria das soluções já comecem a explorar métodos mais naturais de interação, como gestos, ainda não exploram convenientemente todas as potencialidades da interação dos seres humanos e do seu poder descritivo. Em outras áreas, as descrições verbais têm sido utilizadas para recuperação de conteúdo. No entanto, as mesmas ainda não foram aplicadas no contexto dos objetos 3D nem complementadas com outros métodos multimodais. Algumas propostas usam ambientes imersivos, mas em alguns casos os resultados aparecem sobrepostos se forem muito semelhantes. As abordagens tradicionais não sobrepõem resultados, mas a sua visualização com base em vistas não tem a representação adequada dos objetos 3D e não permite interagir com os resultados da mesma forma que os sistemas imersivos permitem.

3. LS3D

Como solução para testar e entender quais os métodos mais naturais para a pesquisa de blocos LEGO, foi desenvolvido um protótipo, o LS3D (*LEGO Search combining Speech and Stereoscopic 3D*). Com o nosso protótipo, pretendemos validar algumas técnicas que ofereçam uma forma

mais natural de pesquisa de objetos 3D, tirando partido de ambientes imersivos e de dispositivos de interação e visualização de última geração. No seu desenvolvimento, tivemos em consideração o estudo realizado por Mendes et al. [Mendes 13], onde é demonstrada a preferência dos utilizadores pela descrição verbal para peças LEGO. Adicionalmente, oferecemos também *feedback* auditivo ao utilizador através da síntese de voz (*text-to-speech*). À semelhança com o trabalho de Mendes et al. [Mendes 13], o nosso protótipo irá representar o papel de fornecedor, onde as pesquisas realizadas são interpretadas pelo sistema, sendo de seguida devolvida uma lista de resultados. Para os resultados retornados, é possível ao utilizador interagir com eles através de gestos ou usando um dispositivo apontador (por exemplo o *Space Point Fusion*). É também possível movimentar a câmara e controlar o ponto de vista do utilizador.

De forma a acelerar o processo de pesquisa das peças, foi necessário criar uma estrutura indexada de toda a coleção de modelos dos blocos de LEGO, que contivesse tanto informação textual como descrições de forma de todos os modelos. Para obter a informação das peças, recorremos ao *LDraw*², uma biblioteca de código aberto que contém descrições textuais e geométricas. Desta biblioteca seleccionámos um subconjunto de 1.810 blocos de LEGO.

3.1 Indexação

Na biblioteca *LDraw*, cada bloco de LEGO tem associado descrições textuais, usando termos consistentes ao longo de toda a biblioteca, como ilustrado na Figura 2. Cada anotação é definida pelo designer após a criação do modelo virtual do bloco, e de seguida é validada pelos restantes membros. Através deste processo são garantidas a consistência e a qualidade das descrições textuais. Para cada bloco é também armazenada a sua informação geométrica, como número de linhas, triângulos, quadriláteros e cores. Para termos o acesso fácil e rápida a toda esta informação, criamos índices invertidos.

Considerando as soluções existentes para o texto, aplicámos uma abordagem proposta por Gennaro et al. [Gennaro 10] ao contexto de recuperação de objetos 3D. Como descritor de forma, usámos o algoritmo D2, que em trabalhos anteriores [Funkhouser 03] demonstrou oferecer resultados razoáveis, apesar do reduzido custo computacional associado. Após extraídas as características da forma para cada objeto, é criado um vetor de 128 características. Com base na ideia de que objetos semelhantes terão descritores semelhantes [Gennaro 10], seleccionamos um conjunto de objetos de referência (RO) da coleção. Para cada objeto da coleção é calculada a distância euclidiana entre ele e cada RO. Com base nestas distâncias, os RO são ordenados e, atribuindo um termo único a cada RO, repetimos este termo mais vezes quanto mais próximo o RO está do modelo. Desta forma criamos uma representação textual para a descrição de forma do objeto que pode ser processada de forma semelhante às descrições textuais que

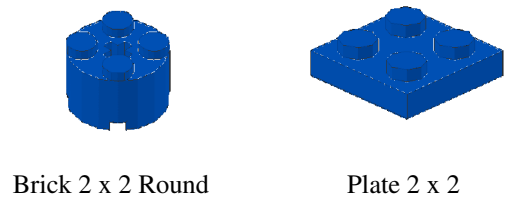


Figura 2. Exemplos de blocos LEGO e respectivas descrições LDraw.

acompanham cada bloco, mantendo a relação entre objetos semelhantes.

Foram seleccionados 85 RO da coleção para representar os 1.810 objetos, com base na equação sugerida por Amato e Savino [Amato 08]: $\#RO \geq 2 \cdot \sqrt{\#X}$. Desta forma, alcançamos um compromisso aceitável entre tempo de consulta e devolução de resultados. Usando apenas os 30 RO mais próximos do objeto, é ainda possível reduzir significativamente o tamanho dos índices com as descrições. Adicionalmente, para cada RO, os objetos da coleção que estão mais próximos a ele estão associados num índice invertido.

Finalmente, quando na recuperação de objetos, baseamos-nos numa abordagem de TF-IDF para calcular a relevância dos resultados. Desenvolvemos duas modalidades de pesquisa: pesquisa-por-texto e pesquisa-por-exemplo. Na pesquisa-por-texto seguimos uma implementação direta semelhante à de um motor de pesquisa padrão para documentos de texto. Para cada termo da consulta, o sistema recolhe os objetos que contenham esse termo no seu índice invertido, e são combinados todos os resultados, e ordenados os resultados de acordo com a sua semelhança com a interrogação. Na pesquisa-por-exemplo combinamos a descrição textual com os descritores de forma. Uma vez que tanto a interrogação com base nas descrições textuais como usando a informação geométrica geram valores de similaridade contidos em intervalos [0,1], é possível calcular uma média de similaridade de todos os resultados, mantendo o valor final de semelhança dentro desse intervalo. Combinando estas duas abordagens, é possível contrariar alguns falsos positivos e falsos negativos que resultariam da aplicação de apenas uma delas.

3.2 Especificação da Interrogação

Como anteriormente indicado, para as pesquisas podem ser realizadas usando pesquisa-por-voz, através da integração de reconhecimento [Meinedo 10] e síntese [Paulo 08] de voz. Usando as expressões mais comuns, indicadas no estudo preliminar de Mendes et al. [Mendes 13], construímos uma gramática usando o formato GRXML. O vocabulário gerado inclui mais de 500 palavras diferentes, devido à inclusão de formas flexionadas das palavras.

Assim sendo, com a nossa gramática é possível utilizadores descreverem um bloco LEGO com mais de que uma

²LDraw: <http://www.ldraw.org/>

descrição. Desta forma, é possível descrever os blocos LEGO através das suas dimensões, forma, cor usando um conjunto diverso de adjetivos. Por exemplo, considerando o bloco na Figura 2 (à esquerda), que pertence à classe dos blocos de LEGO redondos, este poderia ser descrito usando 14 adjetivos diferentes (por exemplo: *redondo, circular, cilíndrica, circunferência*). O bloco na Figura 2 (à direita), que pertence à classe de placas, poderia por sua vez ser descrito com 21 adjetivos diferentes (por exemplo: *fina, baixinha, espalhada, plana*).

No nosso protótipo, cada pesquisa começa com o uso da expressão *Acorda LEGO*, em que o sistema responde verbalmente com um *Sim*. Um exemplo de uma pesquisa pode ser: *Acorda LEGO. Quero uma peça fina, dois por dois*. No nosso trabalho, visto que cada bloco poder ter um vasto conjunto de cores, os utilizadores concentravam-se primeiro na descrição de forma e dimensão dos blocos. Uma vez identificado o bloco desejado, é fácil mudar a sua cor usando um novo comando de voz como *Acorda LEGO. Pinta de encarnado*. Dada a grande quantidade de blocos de LEGO, usamos três estratégias adicionais para restringir o número de peças mostradas. Os utilizadores podem filtrar verbalmente por característica (por exemplo: *Filtra por curva*), ou excluir por tipo (por exemplo: *Exclui DUPLO*) ou obter blocos visualmente semelhantes ao selecionado (por exemplo: *Dá-me semelhante a esta*).

Depois de o utilizador especificar uma pesquisa, o sistema responde imediatamente dependendo do que foi entendido. Por exemplo, na consulta *Acorda LEGO. Quero um bloco dois por dois*, o sistema irá sintetizar *Ilustrando peças dois por dois*. Após isso, o sistema exibe os blocos que correspondem à pesquisa do utilizador.

3.3 Exploração de Resultados

Para a exploração dos resultados das pesquisas, criamos uma nova abordagem que combina as visualizações cilíndricas e esféricas, apresentadas no trabalho de Henriques et al. [Henriques 14] como as preferidas dos utilizadores. Na nossa abordagem, mostramos os resultados distribuídos na metade frontal de um barril (Figura 3), situando o utilizador no interior do mesmo e colocando os blocos com mais relevantes na frente do utilizador. Adicionamos um quadrado colorido por trás de cada bloco para

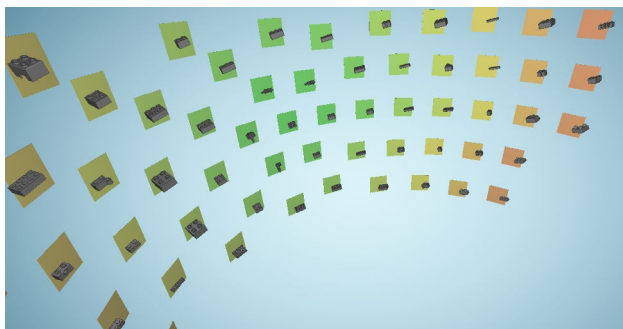


Figura 3. Modo de visualização barril.

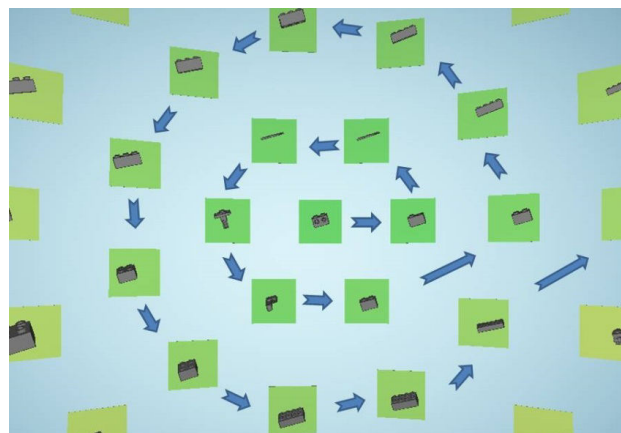


Figura 4. Disposição dos objetos por relevância.

ilustrar sua posição, com cor mais verde para os mais relevantes e vermelho os menos relevantes, conforme ilustrado na Figura 4. Além disso, para proporcionar uma melhor percepção dos objetos, todos os blocos vão girando lentamente. Isso permite aos utilizadores perceberem alguns detalhes adicionais. Quando o utilizador seleciona um objeto, os restantes param de rodar a fim de não chamar atenção desnecessariamente.

De forma semelhante aos modos desenvolvidos por Henriques et al. [Henriques 14], começámos por usar o dispositivo Leap Motion, mas depois de algumas experiências piloto apercebemo-nos que as pessoas muitas vezes perdiam a noção do volume onde poderiam interagir. Deste modo, exploramos diferentes interações, desta utilizando o dispositivo *Space Point Fusion*³, que não apresenta esta restrição. Após a integração deste dispositivo, foram comparadas as interações entre os dois dispositivos através de uma breve avaliação de usabilidade. A avaliação foi realizada por oito participantes, onde 50% deles não tinha qualquer experiência no uso de nenhum desses dispositivos, 5% já haviam experimentado usar ambos os dispositivos e os restantes 25% apenas um dos dispositivos. Verificou-se também que não só os utilizadores preferiram usar o *Space Point Fusion*, como este também foi mais rápido para realização dos testes do que o Leap Motion.

Recorrendo ao giroscópio do *Space Point Fusion*, os utilizadores podem apontar para um bloco específico (Figura 5). Após apontar para um bloco, o botão esquerdo do dispositivo pode ser premido para trazer o objeto para mais próximo do utilizador. Adicionalmente, o utilizador pode também rodar livremente o dispositivo, permitindo que o bloco seleccionado possa ser visto de todos os ângulos, conforme ilustrado na Figura 6.

3.4 Contrução do modelo de LEGO

Depois de navegar e encontrar o bloco desejado, o nosso protótipo permite ao utilizador construir um modelo LEGO

³PNI Sensor, <http://www.pnicorp.com/products/spacepoint-gaming>

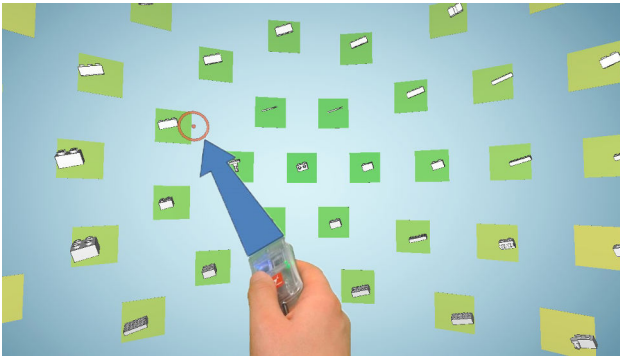


Figura 5. Selecionar um objeto usando o Space Point Fusion.

virtual. Depois de selecionado o bloco, o utilizador pode esconder os resultados da pesquisa e alterar para o modo de construção, mantendo o bloco selecionado. Neste modo é exibida uma grelha de LEGO, como representado na Figura 7. O utilizador pode então apontar em qualquer posição para colocar o bloco na localização desejada.

No modo de construção, é também possível manipular os blocos já colocados no modelo. Mantendo um bloco selecionado, o utilizador pode pesquisar por blocos semelhantes ou colori-los, usando as instruções de voz acima mencionados, ou mesmo descartá-lo.

4. AVALIAÇÃO

Para validar a nossa solução, comparámos LS3D contra uma aplicação comercial, Lego Digital Designer (LDD), através de um conjunto de testes com utilizadores. O LDD utiliza o paradigma *Windows Icons Menus and Pointing* (WIMP) apresentando uma coleção de blocos LEGO divididos em categorias numa grelha tradicional 2D, e cada objeto é representado por uma vista 2D.

4.1 Participantes

A nossa avaliação foi realizada por 20 participantes (quatro do sexo feminino), com idades entre os 18 e 50 anos (70% entre 24 e 29). Todos os participantes eram falantes nati-

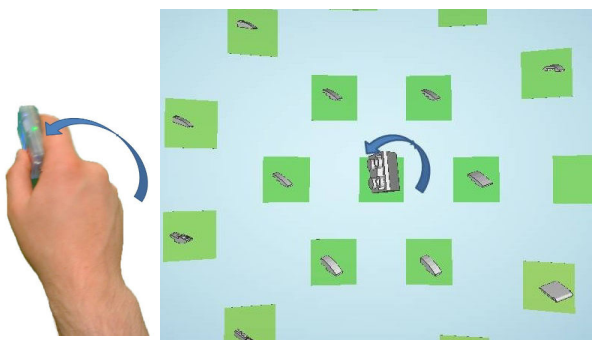


Figura 6. Utilizador rodando um objeto usando o Space Point Fusion.

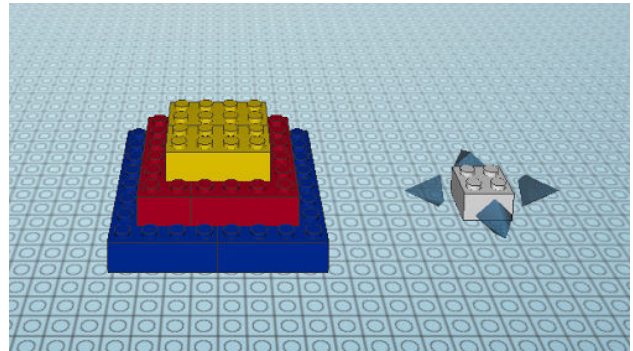


Figura 7. Construindo um modelo de LEGO virtual, com a peça selecionada.

vos da língua portuguesa, sendo a maioria deles residentes da região urbana de Lisboa, e nenhum deles tinha anteriormente usado o LDD. No entanto, os participantes já tinham bastante experiência no uso de motores de busca, por exemplo de pesquisa de imagens, sendo usado numa base diária. Apenas 30% dos participantes já estavam familiarizados com motores de pesquisa de objetos 3D, onde em todos eles os objetos eram representados usando vistas 2D. 60% dos participantes já tinham usado sistemas de pesquisa por voz, e também 60% dos participantes já haviam experimentado algum tipo de visualização estereoscópica, contudo apenas 50% tinham anteriormente usado Head-Mounted Display's (HMD).

4.2 Equipamento

As sessões de testes, foram conduzidas num ambiente controlado, sem influências externas. Para executar LDD, foi usado um computador padrão com rato e monitor, como ilustrado na Figura 8 (esquerda). O LS3D foi testado usando o *Space Point Fusion* para interações, os Oculus Rift para a visualização e um conjunto com microfone e auriculares para as interação por voz, como pode ser visto na Figura 8 (direita).

4.3 Metodologia

Os testes com os utilizadores foram estruturados em quatro etapas: questionário inicial para avaliar a experiência e perfil de cada utilizador; resumo dos objectivos dos testes; execução das tarefas nos dois sistemas; e questionário comparativo da usabilidade de ambos os sistemas. Para garantir uma distribuição uniforme dos testes, 50% dos utilizadores começaram por testar o LDD e os outros 50% começaram pelo nosso protótipo.

A execução das tarefas em cada sistema começou com um breve tempo de adaptação, onde os utilizadores experimentaram a interface. Após isso, foram convidados a procurar oito blocos, sendo mostrado exemplos de blocos de LEGO físicos. A ordem de apresentação dos blocos foi aleatoria, de forma a diferentes utilizadores não terem a mesma sequência, a fim de garantir uma distribuição uniforme. Depois de terminada a procura dos oito blocos, os



Figura 8. Participantes numa sessão de avaliação, usando o LDD (esquerda) e o LS3D (direita).

utilizadores foram convidados a responder a um pequeno questionário sobre sua experiência de utilização em relativamente aos dois sistemas.

4.4 Resultados

Realizamos três perspectivas diferentes na análise dos resultados dos nossos testes com utilizadores. Em primeiro lugar, apresentamos uma análise quantitativa sobre as medidas objetivas, ou seja, tempo e erros. Em seguida, apresentamos também a análise qualitativa com base nas respostas do questionário. Finalmente, discutimos várias observações capturadas durante as sessões de teste.

4.4.1 Análise Quantitativa

Durante os testes com utilizadores, registamos o tempo necessário para a realização da pesquisa de cada uma dos oito blocos, bem como o número de blocos errados selecionados. Estes valores estão representados no gráfico da Figura 9. Na análise dos resultados, na relação do tempo gasto na procura dos oito blocos, foi utilizado o teste de *signed-ranks* de Wilcoxon, com o qual podemos concluir

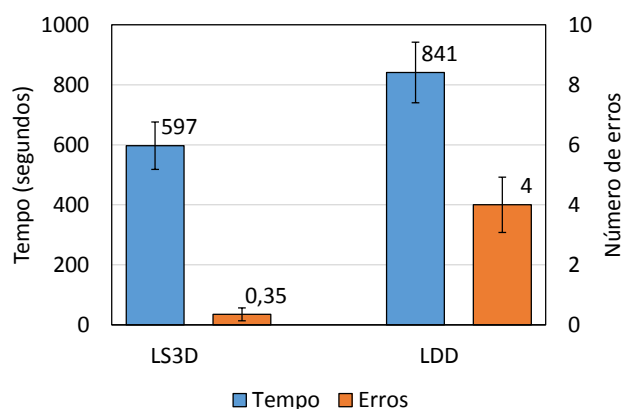


Figura 9. Tempo e número de erros na pesquisa dos oito blocos de LEGO (médias e intervalos de confiança a 95%).

que existem diferenças estatisticamente significativas: os utilizadores realizaram as pesquisas mais rápidas no LS3D do que com o LDD ($Z = -3,509$, $p = 0,000$). Quanto ao número de blocos errados, também foi utilizado o teste de *signed-ranks* de Wilcoxon, onde se conclui que também houveram diferenças estatisticamente significativas, sendo que a solução LS3D voltou a demonstrar melhor desempenho do que o LDD ($Z = -6,697$, $p = 0,000$), originando menos erros.

4.4.2 Análise Qualitativa

Depois de completar as tarefas em ambos os sistemas, os utilizadores foram convidados a classificar as interfaces de pesquisa, usando uma escala de Likert de 4 valores, sobre: quão divertido foi de usar; quão fácil foi de ver os blocos; e quão simples foi de usar o sistema. Os resultados deste questionário são apresentados na Tabela 1. O teste de Wilcoxon foi novamente utilizado para encontrar diferenças estatisticamente significativas. Os utilizadores concordaram que LS3D foi o mais fácil de usar ($Z = -2,441$, $p = 0,015$). No nosso sistema, até mesmo os participantes que nunca usaram para pesquisas por fala, conseguiram concluir as tarefas com sucesso. Relativamente à visualização dos objetos, os utilizadores concordaram que LS3D permitiu melhor interpretação da representação 3D ($Z = -2,780$, $p = 0,005$). Finalmente, também concordaram que LDD foi menos divertido face ao LS3D ($Z = -3,626$, $p = 0,000$).

4.4.3 Observações

Durante as avaliações, notámos alguns aspectos relevantes no comportamento dos utilizadores, bem como alguns comentários. Ao longo da avaliação, vários utilizadores

System	LDD	LS3D
Utilização	2 (1)	3 (1)
Visualização	2 (1)	3 (0)
Divertimento	2 (1)	4 (0)

Tabela 1. Classificações de cada sistema (Média, amplitude inter-quartil).

mencionaram que a divisão por categorias do LDD era bastante útil. No entanto, essa divisão não ajudava em dois dos blocos a procurar, dado que esses blocos não aparentavam pertencer a nenhuma das categorias, e não eram semelhantes aos objetos representados. Isso obrigou esses utilizadores a abrir todas as categorias e navegar nelas. Contudo, este processo facilitou mais tarde a procura pelo blocos seguintes. Utilizadores que não começaram por esses dois blocos fizeram comentários que refletem frustração, enquanto procuravam no LDD, algo que não ocorreu no LS3D, mesmo em pesquisas mais longas.

No LDD, os utilizadores mostraram ter problemas em entender os blocos representados, o que não aconteceu no LS3D. Comentários sugeriram que o nosso protótipo oferece uma melhor percepção dos objetos 3D, do seu tamanho e detalhes, principalmente em graças à rotação dos blocos listados. Isso foi particularmente visível em blocos sem pinos, em que os utilizadores muitas vezes selecionavam um bloco com um tamanho diferente quando usando o LDD.

Além disso, alguns utilizadores comentaram que era mais fácil usar o LS3D para procurar objetos pela sua dimensão, apesar de alguns mencionarem dimensões erradas durante a pesquisa. Estes erros eram notados e corrigidas após serem mostrados os resultados da pesquisa e o sistema responder o que havia sido entendido. Notámos também que os utilizadores recorreram mais a metáforas e adjetivos nas últimas pesquisas. Isso deveu-se ao facto de não terem inicialmente o conhecimento de todas as palavras que poderiam ser usadas e apenas no final já tinham ideia do potencial do nosso protótipo.

Observamos que as funcionalidades disponibilizadas no LS3D ajudaram bastante os utilizadores a encontrar os objetos pretendidos. Além disso, foi também mencionado que o refinamento da exclusão foi o menos necessário, porque as outras possibilidades de refinamento, exemplo e filtragem, demonstraram ser suficientes para cumprimento das tarefas. Isso foi também observado na distribuição dos refinamentos, onde 55% da estes eram para filtrar, 36% para exemplos e apenas 9% para exclusões.

5. CONCLUSÕES E TRABALHO FUTURO

Com os ambientes virtuais imersivos sendo cada vez mais comuns, permitindo que utilizadores se coloquem de uma forma mais credível dentro de um mundo virtual, novos desafios têm surgido. Um exemplo disso é o seu uso na exploração de objetos de uma coleção dentro de um ambiente imersivo, ou de aplicações para a montagem de modelos virtuais. Com esta motivação, desenvolvemos um protótipo que permite a construção de modelos LEGO virtuais, a fim de melhor compreender os métodos mais naturais e simples para a recuperação de objetos em tais sistemas.

Com base no conhecimento obtido, desenvolvemos o nosso protótipo LS3D usando consultas faladas para recuperar os blocos de LEGO virtuais. Para ver a coleção de blocos, criámos um novo modo de visualização em barril,

no qual o utilizador é cercado por objetos virtuais.

Realizámos uma avaliação com 20 utilizadores, comparando o nosso protótipo com uma aplicação comercial da marca LEGO. Os resultados sugerem que o nosso protótipo é capaz de superar a aplicação tradicional, que usa uma grelha 2D com miniaturas, apesar de esta ser uma interface mais familiar aos utilizadores. Através dos resultados obtidos, podémos concluir que a nossa solução ofereceu uma melhor percepção dos objetos, sendo menos sujeita a erros do que a aplicação comercial. Além disso, através de comentários dos utilizadores e análise dos questionários, podémos concluir que a maioria dos participantes achou fácil usar a nossa solução de pesquisa, mesmo sem um período de treino extensivo.

Como trabalho futuro, consideramos que seria interessante maior investigação para o uso de modelos de linguagem mais sofisticados. Para mostrar resultados de consultas, melhorias poderiam ser feitas usando agrupamentos de objetos, como disponível no LDD, mas com mais de um objeto ilustrando cada grupo. Como mencionado, o nosso protótipo oferece uma melhor experiência ao utilizador, mas acreditamos que usando a nova versão dos Oculus Rift que possui seguimento da posição do utilizador, se conseguirá alcançar uma maior imersão do utilizador.

Embora nós exploremos blocos de LEGO, acreditamos que a nossa solução pode ser aplicada a outros cenários que utilizam modelos virtuais, como por exemplo no contexto de montagem nas indústrias automóvel e de construção civil.

6. AGRADECIMENTOS

O trabalho apresentado neste artigo foi parcialmente financiado pela Fundação para a Ciência e a Tecnologia (FCT) através do projeto TECTON-3D (PTDC/EEI-SII/3154/2012), Pest-OE/EEI/LA0021/2013 e através da bolsa SFRH/BD/91372/2012.

Referências

- [Amato 08] Giuseppe Amato e Pasquale Savino. Approximate similarity search in metric spaces using inverted files. Em *Proceedings of the International Conference on Scalable Information Systems*, 2008.
- [Cabral 11] M. Cabral, P. Vangorp, G. Chaurasia, E. Chapoulie, Martin Hachet, e G. Drettakis. A multimode immersive conceptual design system for architectural modeling and lighting. Em *3D User Interfaces (3DUI), 2011 IEEE Symposium on*, páginas 15–18, March 2011.
- [Funkhouser 03] Thomas Funkhouser, Patrick Min, Michael Kazhdan, Joyce Chen, Alex Halderman, David Dobkin, e David Jacobs. A search engine for 3D models. volume 22, páginas 83–105, Janeiro 2003.
- [Gennaro 10] Claudio Gennaro, Giuseppe Amato, Paolo Bolettieri, e Pasquale Savino. An approach to content-based image retrieval based on the lucene search engine library. Em *Research and Advanced Technology for Digital Libraries*, páginas 55–66. Springer, 2010.

- [Gupta 12] Ankit Gupta, Dieter Fox, Brian Curless, e Michael Cohen. Duplotrack: a real-time system for authoring and guiding duplo block assembly. Em *Proceedings of the 25th annual ACM symposium on User interface software and technology*, páginas 389–402. ACM, 2012.
- [Henriques 14] D. Henriques, D. Mendes, P. Pascoal, I. Trancoso, e A. Ferreira. Poster: Evaluation of immersive visualization techniques for 3d object retrieval. Em *3D User Interfaces (3DUI), 2014 IEEE Symposium on*, páginas 145–146, March 2014.
- [Holz 11] Christian Holz e Andrew Wilson. Data miming: inferring spatial object descriptions from human gesture. páginas 811–820, 2011.
- [Jacobson 05] Jeffrey Jacobson, Marc Le Renard, Jean-Luc Lugin, e Marc Cavazza. The caveat system: Immersive entertainment based on a game engine. Em *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, ACE '05, páginas 184–187, New York, NY, USA, 2005. ACM.
- [Kamvar 10] Maryam Kamvar e Doug Beeferman. Say what? why users choose to speak their web queries. Em *Interspeech*, 2010.
- [Laga 11] H Laga, T Schreck, A Ferreira, A Godil, I Pratikakis, e R Veltkamp. Bag of words and local spectral descriptor for 3d partial shape retrieval. 2011.
- [Lee 10] C Lee, Alexander Rudnicky, e GG Lee. Let's buy books: finding ebooks using voice search. páginas 85–90, 2010.
- [Lee 12] Cheongjae Lee e Tatsuya Kawahara. Hybrid vector space model for flexible voice search. Em *Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*, páginas 1–4. IEEE, 2012.
- [Liu 13] Yong-Jin Liu, Xi Luo, Ajay Joneja, Cui-Xia Ma, Xiao-Lan Fu, e Dawei Song. User-Adaptive Sketch-Based 3-D CAD Model Retrieval. páginas 1–13, 2013.
- [Meinedo 10] Hugo Meinedo, Alberto Abad, T Pellegrini, I Trancoso, e J Neto. The L2F Broadcast News Speech Recognition System. páginas 93–96, 2010.
- [Mendes 11] Daniel Mendes, Pedro Lopes, e Alfredo Ferreira. Hands-on interactive tabletop LEGO application. Em *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology - ACE '11*, Novembro 2011.
- [Mendes 13] D. Mendes, D. Henriques, I. Trancoso, e A. Ferreira. Qual peça? identificação de métodos naturais para a descrição de objectos 3d. Em *INTERAÇÃO 2013*, 2013.
- [Miller 12] Andrew Miller, Brandyn White, Emiko Charbonneau, Zach Kanzler, e Joseph J LaViola. Interactive 3d model acquisition and tracking of building block structures. *Visualization and Computer Graphics, IEEE Transactions on*, 18(4):651–659, 2012.
- [Nakazato 01] M. Nakazato e T.S. Huang. 3D MARS: immersive virtual reality for content-based image retrieval. páginas 44–47. Ieee, 2001.
- [Paquet 99] Eric Paquet e Marc Rioux. Nefertiti: a query by content system for three-dimensional model and image databases management. volume 17, páginas 157–166, 1999.
- [Pascoal 12] Pedro Pascoal, Alfredo Ferreira, e Joaquim Jorge. Towards an immersive interface for 3d object retrieval. Em *Proceedings of the 5th Eurographics conference on 3D Object Retrieval*, EG 3DOR'12, páginas 51–54, Aire-la-Ville, Switzerland, Switzerland, 2012. Eurographics Association.
- [Paulo 08] Sérgio Paulo, Luís C Oliveira, Carlos Mendes, Luís Figueira, Renato Cassaca, Céu Viana, e Helena Moniz. Dixi—a generic text-to-speech system for european portuguese. Em *Computational Processing of the Portuguese Language*, páginas 91–100. Springer, 2008.
- [Pu 05] Jiantao Pu, Kuiyang Lou, e Karthik Ramani. A 2 d sketch-based user interface for 3 d cad model retrieval. *Computer-Aided Design and Applications*, 2(6):717–725, 2005.
- [Santos 08] Tiago Santos, Alfredo Ferreira, Filipe Dias, e Manuel J. Fonseca. Using sketches and retrieval to create lego models. 2008.
- [Smith 97] JR Smith e SF Chang. Visually searching the web for content. 1997.
- [Trowbridge 09] S. Trowbridge e C. Stapleton. Melting the boundaries between fantasy and reality. *Computer*, 42(7):57–62, July 2009.
- [Wang 11] Robert Wang, Sylvain Paris, e Jovan Popović. 6d hands: markerless hand-tracking for computer aided design. Em *Proceedings of the 24th annual ACM symposium on User interface software and technology*, UIST '11, páginas 549–558, New York, NY, USA, 2011.

Older Adult Performance Using Body Gesture Interaction

Micael Carreira

INESC-ID, Instituto Superior Técnico, Universidade de Lisboa
Rua Alves Redol, 9, 1000-029 Lisboa, Portugal

micaelcarreira@ist.utl.pt

Daniel Gonçalves

daniel.goncalves@inesc-id.pt

Karine Ting

ICD, Troyes University of Technology
Troyes, France

karine.lan@gmail.com

Abstract

Gesture interfaces are becoming an increasingly popular way to interact with technology, as they are considered very easy to learn and use. However, most gesture interactions studies focus on the average adult or, when focusing on older adults, it is usually in the gaming or physical activity contexts. In this study, we evaluate the suitability of gestural interfaces for older adults, in order to interact with a general technological interface. To this end, we asked 14 older users to perform a set of navigation and selection tasks; two tasks required to interact on most technological interfaces. For each of these tasks, we evaluated two alternative gestures. All senior participants were able to complete almost all the proposed tasks, and enjoyed using this type of interface. We concluded that gestural interfaces are adequate for the senior users, and derived a set of design implications that future developers should take into account when developing gestural interactions for the older people.

Keywords

Body gestures, older adults, Kinect, natural user interface.

1. INTRODUCTION

Over the last decade, gestural interfaces have earned increasing interest, both in the commercial industry as well as in research. This type of interface gained popularity in the video game context where, usually, users move their body that acts as a controller to play video games. However, due to the broad availability and low cost of gesture recognition hardware, several applications are being developed out of the gaming context.

Since people express themselves and interact in everyday social life through gestures, gestural interfaces are considered very natural and easy to use [Correia13]. Therefore, body gesture interfaces provide for easier technological interactions for groups that, until now, have shown some resistance in adopting technology, such as the older adults. In general, seniors find traditional computer interfaces overly technical and difficult to use [Cody99, Namazi03], which deprives them from the benefits technology has to offer. The number of ageing adults is growing and due to our healthier lifestyles we live longer, and are likely to be physically, socially and cognitively active until older ages. By captivating the interest of senior users in technology, it is possible to fight isolation and exclusion and allow them to be more productive, independent and to have a more social and fulfilling life.

However, seniors have particular physical characteristics that can be a hindrance when using gestural interfaces. Research shows that ageing brings along a significant decline in cognitive, perceptual and motor abilities. Motor issues of older adults include slower motion, less strength, less fine motor control and decreased range of motion and grip force [Nichols06]. Therefore, interac-

tions should be designed carefully in order to avoid fatigue, exhaustion and fine motor control. On the other hand, since some degree of physical activity is required to interact with gestural interfaces, it is likely to positively impact the health of the senior users, even if the intensity of physical activity is low [Saposnik10].



Figure 1. Participant using gestures to select a target on a TV screen.

Current literature focuses mainly on gesture interactions for average adults or, when it focuses on older adults, it is usually in the gaming context. Therefore, seniors' performance and acceptance towards body gesture interfaces are not well understood, particularly considering their specific needs and abilities out of the gaming context. In this study, we aim to understand how older adults can benefit from gesture based interactions, in terms of suitability and acceptance, when interacting with technological interfaces in general. To this end, we focused on two types of tasks required to interact on most technological interfaces: navigation and selection. For each task, we developed and evaluated two alternative gestures (Figure 2). Regarding navigation, we defined a Swipe gesture and a Grab and Drag gesture. For the selection task, we de-

veloped a Point and Push gesture and a Point and Hold gesture. An experimental evaluation with the target user group was performed, where we measured both performance and acceptance of the defined gestures. Our results showed that that this type of interface can, indeed, be successfully used by seniors, since most participants enjoyed using gestures and completed all the proposed tasks more or less easily. We also systematically compare the results for each task, concluding with the better suited alternative for the seniors. The results of this study are transversal to many applications, since most interfaces require navigating through information and selecting a particular target in a set.

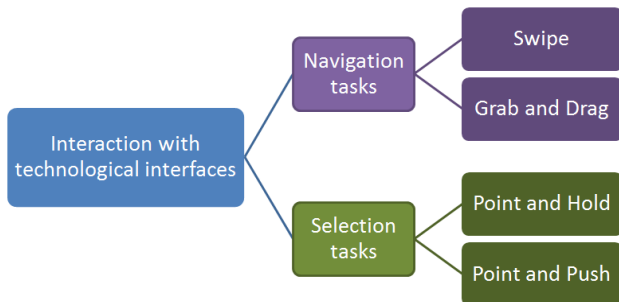


Figure 2. Diagram showing the evaluated gestures to interact with a general technological interface.

2. RELATED WORK

Gesture recognition can be seen as a way for computers to understand human body language, interpreting body gestures via mathematical algorithms. There are two main ways of achieving gesture recognition: with devices that have motion sensing capabilities (e.g. accelerometer, gyroscope, magnetometer) or video capturing and processing – also called computer vision or remote sensing – to detect users’ movements. The usage of gestures to control technological devices is a discipline being investigated for some years now [Hummels98]. However, in recent years, body gesture interfaces have become more popular due to the video game industry. Most gestural interactions we reviewed are based on the Microsoft’s Kinect device, since it has advantages over the competition as we depict in the next section. In the first subsection, we analyze general gestural interaction interfaces for the average adult, while in the second subsection we analyze the studies performed with older users.

2.1 General Body Gestures Studies

Despite the Kinect having been primarily developed to interact in video games, several applications are emerging in a diversified range of scenarios. Maldi et al. [Maldi13] developed a gestural interaction that allows controlling a photo viewer application. Authors defined four gestures: a push gesture to select a picture; a pull movement to enable returning to the parent level of hierarchy; and moving the hand to the left or right in order to look around all the photos. Although this study has the novelty of controlling a media interface with hand motion, no experimental evaluation was performed. Henze et al. [Henze10] analyzed static and dynamic gestures to interact with a music application. Static gestures refer to the

user’s pose or spatial configuration, and dynamic gestures to users’ movements in a certain time interval. Authors performed an experimental evaluation with 12 participants and found that dynamic gestures are easier to remember, more intuitive and simpler for controlling their application. Panger [Panger12] focused on using Kinect in real-life kitchens, which allows interacting even when users’ hands are messy. He implemented a recipe navigator, a timer and music player. One of the main challenges was preventing accidental commands, since intentional commands are interspersed with the cooking movements. A five subject experiment, in each user’s home, revealed that installing the Kinect was simple and that subjects felt successful interacting in this context.

Indeed, a major concern regarding gesture recognition is the recognition of a few significant gestures from a continuous sequence of movements, as studied in [Kim06]. Authors proposed a sequential identification scheme that performs gesture segmentation and recognition simultaneously and achieved a 95% recognition rate for smart home environments to control lights and curtains. Kim et al. [Kim11] purpose Ambient Wall, a prototype of a smart home system that can display the current status of the house through a projection on a wall. Their scenarios include changing channel on the TV, control the room temperature, check for messages, and turn off all devices. Their main focus is to enable users to monitor what is happening in their house at a glance, and control their surroundings by performing simple gestures that do not require any physical interface device. Another system that allows to control various devices in the house is HandsUp [Oh12], which uses the Kinect device together with a projector to display an interactive interface onto the ceiling of a room. Authors argue that this surface is a perfect screen for when people lay down on the bed or sofa.

2.2 Studies Focused on Older People

As we already stated, gesture recognition interfaces gained popularity in the video game industry. Probably due to this, most studies evaluating body gestures interfaces with older people still fall on the scope of games. Jung et al. [Jung09] examined the impact of playing Nintendo Wii games on the psychological and physical well-being of seniors in a long-term care facility. Although the game was not specifically adapted to older users, the seniors enjoyed playing it and found it stimulating. Moreover, a substantial amount of physical activity is required to play these games, which is likely to be beneficial in the health of the older users. Identical findings were observed in a similar study [Sapostnik10].

Other approaches focus on developing gesture controlled games for seniors, taking into account their physical limitations [Gerling12]. Authors developed and tested 4 static and 4 dynamic gestures with institutionalized older adults. Results showed that the gestural game was successful among older adults, and even had a positive effect on the participants’ mood. Static gestures were generally easier to perform. However, they also found that recalling gestures was too challenging for some participants. Au-

thors also found that institutionalized older adults represent an extremely heterogeneous group, and defend that gestural interfaces should be individually adjusted to each user. Similar to this study, Ganesan et al. [Ganesan12] aims to find the factors that play an important role in motivating older adults to maintain a physical exercise routine, a habit recommended by doctors but difficult to sustain. An early game prototype was developed for the Microsoft Kinect and preliminary results were promising.

More recent studies are adapting gestural gaming devices to interact with technological devices in general. Bobeth et al. [Bobeth12] evaluated how older users performed using in-air gestures to control a TV menu. Authors compared 4 gestures and results showed that directly transferring tracked hand movements to control a cursor on a TV achieved the best performance and was preferred by the users. Hassani et al. [Znagui11] developed an assistive robot that used Kinect in order to help older people perform physical exercises. Results showed that seniors enjoyed using this simple gestural interface.

3. EXPERIMENTAL DESIGN

Gestural interfaces became more prevalent in the video game industry, through devices such as the Nintendo Wii Remote, Microsoft Kinect and PlayStation Eye/Move. How gestural-based interfaces fare, for older users, in more traditional tasks such as navigation and selection of items remained largely unexplored. When comparing gesture based interaction using the Kinect to other gesture based devices, Kinect has the benefit of not requiring any accessory to operate, making it more practical and comfortable to use. The direct competitors, Nintendo Wii Remote and PlayStation Move, require a handheld controller to perform the gesture detection. Therefore, Kinect allows more direct and natural interactions and also the tracking of the whole body, as opposed to gestural interfaces that require a controller and only track the forces applied to that controller – usually handheld. Moreover, considering the case of the older adults, remote sensing is better suited since users that suffer from arthritis may have difficulties in holding a controller. Therefore, we decided to use Microsoft’s Kinect sensor in our study.

3.1 Research Questions

This user study aims to answer 3 main research questions:

1. Are gestural interfaces adequate for older adults to interact with general technological interfaces?
2. Which type of gesture allows for fastest navigation and selection with the lowest error rate?
3. Do older users enjoy and easily adapt to gestural interfaces? Which gestures do older users prefer?

3.2 Implementation

In order to understand if gestural interfaces are suited for seniors when interacting with a general technological interface, we focused on two types of tasks: navigation and selection. For each task, we evaluated two alternative gestures (Figure 2). We designed simple one hand gestures, thus avoiding problems that may arise with biman-

ual interactions [Nichols06]. For all the defined gestures, it is only required that seniors move their dominant hand above the hip and in front of their body for a short period of time. Therefore, all the gestures are relatively simple and physically easy to achieve.

Regarding navigation, we evaluated Swipe and Grab and Drag gestures. To perform a Swipe, users should drag either hand in the air and perform a horizontal motion to the desired direction. A Swipe gesture is only considered when users horizontally move their hand for at least 30cm. The vertical motion of the hand should not exceed 10cm, or the gesture is not considered a horizontal Swipe. The time interval of the gesture should be between 0.25 and 1.5 seconds. Regarding the Grab and Drag gesture, we used the implementation of Microsoft’s Kinect SDK. To perform the Grab and Drag, users should raise either hand so that a hand cursor appears on screen. The hand should be open, and the palm should be facing the Kinect sensor. Then, users can close their hand to “grab” the content and then they can drag the hand in the desired direction to scroll. To scroll more, users have to open their hand to “release”, so they can Grab and Drag again. This alternative may require more movements and coordination than the Swipe gesture, but we expect users to have more control on the navigation process. The Swipe gesture strives for simplicity.

For the selection task, we developed Point and Push and Point and Hold gestures. For both gestures, users should raise either hand towards the screen so a hand cursor appears. Then, to perform a selection through the push gesture, users should move their hand in the direction of the screen, as if they were touching the target. For this gesture, we also used the implementation in Microsoft’s Kinect SDK. Regarding the Point and Hold gesture, users should keep the hand cursor over a target for 1.5 seconds to select it. The interface gives feedback about the selection state of the target by progressively filling its background with a lighter color, like a sandglass. When the target is completely filled, it is selected. We expect the Point and Push gesture to be more precise, since it will not restrict the time users have to aim. The Point and Hold is simpler, as the users only have to keep pointing for a while to perform a selection. Prior to performing the user tests, all the developed gestures were evaluated by a physical therapist to access their suitability taking into account seniors’ potential physical limitations. The physical therapist concluded that these gestures posed no danger of overexertion or lesion on older people.

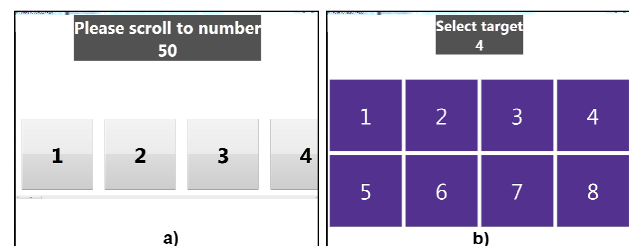


Figure 3. The navigation (a) and the selection (b) screens.

Our test prototype was implemented as a Windows Presentation Foundation application, and the gesture tracking was developed using Kinect for Windows SDK. To test the navigation gestures, we had a list of horizontally scrollable numbers, as shown in Figure 3a. For the selection gestures, a different number of targets were displayed on the screen, and users were asked to select a particular target from the set, as shown in Figure 3b.

4. USER STUDY

We conducted the user study described in this section in order to evaluate which gestures allow better performance and user satisfaction on navigation and selection tasks.

4.1 Participants

Fourteen older people, 3 men and 11 women, took part in our user study. All participants were aged over 60 years old and all had some experience with computers, being that only one of them did not own a computer at home. None of them had prior experience with gestural interfaces. Most users had some sort of physical movement limitations, such as slight rheumatism, tendinitis, osteoarthritis, ankylosing spondylitis, but nothing particularly severe. These conditions did not prevent them from using gestural interfaces. All precautions were taken to let them rest if they felt tired or had aching articulations.

4.2 Apparatus and Setup

In order to simulate an interaction scenario as close as possible to a real-life scenario where users could benefit from using a gestural interface to interact with technology, we performed the user study in an environment that resembled a living room (Figure 1). The output device was a 55" Samsung LED TV with 1080p, connected to a Dell laptop with quad core 3.2 GHz processor and 4 GB of RAM. We used a Kinect for Xbox, connected to the laptop through an adaptor cable. Participants were at a distance of 2.5 meters from the TV and the Kinect sensor.

4.3 Procedure

The user study had two main phases: training and evaluation. At the beginning of the training phase, we explained how the gestures were performed and participants were allowed to try each gesture for a maximum of 2 minutes. However, if the monitor found that the senior had understood and was already comfortable performing the gesture before this time interval, he would skip to the next gesture. Participants were free to choose the pose that was more comfortable to them, either standing or sitting.

In the evaluation phase, users were asked to perform specific tasks for navigation and selection. To test the navigation gestures, participants were successively asked to scroll to a predetermined number that was displayed on the screen (Figure 3a). After the user scrolls to the required number and that target stays visible for 2 seconds, the application automatically shows a new target. We imposed this 2 second target visibility period in order to exclude cases where the user did not have enough precision to scroll to a particular number, and thus avoid activation when participants just quickly passed by the target. The required navigation numbers order was chosen in a

way to cover 3 conditions: large, medium, and small ranges of scroll. A total of 8 navigations were required for each navigation gesture.

Regarding the selection task, the application asks to select a random target in a grid of 2 targets, then in a grid of 4, then in grid of 8 (Figure 3b), and finally in a grid of 16 targets. The varying number of selectable targets allows us to access the performance and precision of the developed gestures relative to the target number and size. This procedure is repeated 3 times, so a total of 12 selections were performed per participant. When users select the desired target, the application automatically moves to the next target selection task. In case of a wrong selection, the application logs it as a missed hit and the user is asked to select the same target again. If a participant makes 3 wrong selections, the application assumes the user failed completing that task and would automatically switch to the next target selection task.

Between each successfully completed navigation and selection tasks, a 5 second period was imposed where users could not interact. This allowed for frequent relaxation of the older users' arm, as well as simulate more realistic interactions, since users typically have to process the newly displayed information after interacting with a technological system. In order to avoid any bias related to the sequence of the performed gestures, the application randomizes the order of the tested gestures for the navigation and selection tasks. Participants' performance was automatically measured by logging the task completion time, as well as the number of errors. After performing all the required gestures, the users answered a simple questionnaire with 3 questions for each gesture regarding the easiness of performing that gesture, whether it was tiring, and the accuracy of the gesture detection. We opted to perform a simple usability questionnaire since, from our previous experience, older users find exhaustive questionnaires like TAM3 [Venkatesh08] too complex and have difficulty discerning between questions. A whole user test took an average of 25 minutes to complete.

4.4 Dependent Measures and Analysis

We used a within-subjects design where each participant tested all conditions. We performed Shapiro-Wilkinson tests on the observed values for the task completion time and number of errors to assess if dependent variables were normally distributed. If they were, we used the parametric paired t-test. If measures were not normally distributed, we used the non-parametric Wilcoxon test.

5. RESULTS

In this section, we quantitatively analyze the time required to complete the proposed tasks, as well as the number of errors that participants made while performing those tasks. We also qualitatively analyze the questionnaires' answers and the users' comments.

We must note that the results we are presenting are not completely uniform, as we slightly changed the way our application gave feedback to users in Swipe tasks. In the first 5 user tests, no visual feedback was given about the

state of the gesture recognition in Swipe tasks. However, participants reported that they missed visual feedback, which is present for the Grab and Drag gesture. So, after the 5th user, we decided to incorporate a simple feedback mechanism. It consists in showing a hand icon on the bottom-right corner or bottom-left corner of the screen (depending on the hand used) when users raise their hand above the hip, thus indicating that the system is ready to detect Swipes. The hand icon is static and does not replicate users' movements; it is just meant to be a simple indicator of the system's gesture recognition state. After integrating this simple feedback mechanism, users commented that this type of feedback was very useful. It did not change, however, the performance nor the number of errors, as the means are similar before and after the improvement, so we consider the results comparable.

5.1 Task completion time

The boxplot in Figure 4 illustrates the time required to perform the proposed tasks, grouped by gesture. Regarding navigation tasks, users completed them faster when using the Swipe gesture. Indeed, a paired t test revealed that the differences are statistically significant ($p < 0.005$). This occurred mainly because Swipes allow to scroll bigger distances faster. Moreover, most senior users found the Grab and Drag gesture to be more complex and harder to perform than the Swipe gesture. Some participants reported that they needed to be very focused in order to coordinate the motions required to perform the Grab and Drag gesture. Indeed, one participant (who recently had a stroke) was having so many difficulties performing this gesture, we had to end this task before the user completed it. On the other hand, some users preferred the Grab and Drag gesture because it allowed a finer control, especially for small distances. For the Swipe gesture, only one senior user had major problems using it.

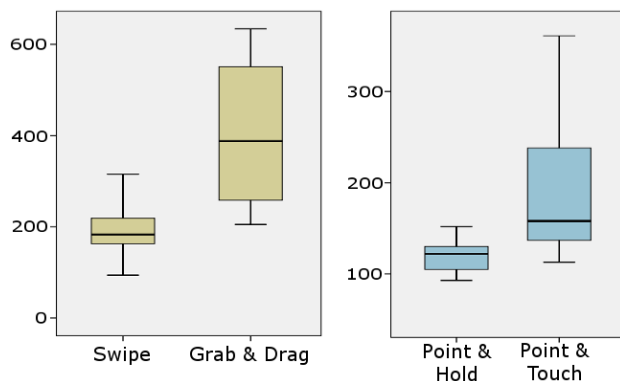


Figure 4. Time required, in seconds, that users took to complete the proposed tasks.

Regarding selection tasks, the Point and Hold gesture allowed for a better performance when compared to the Point and Push alternative. A paired t test showed that this difference is statistically significant ($p = 0.009$). Since both gestures require pointing at the screen, we can conclude that users took more than 1.5 seconds to perform the push gesture – the time required in Point and Hold to perform a selection. Almost all users performed the selection tasks without difficulties, finding the gestures easy to

perform. The exception was the same user unable to perform the Grab and Drag gesture, which also did not manage to conclude all the Point and Push gesture tasks.

5.2 Error rate

Regarding navigation tasks, we considered and classified errors into three categories: Direction, No Output and Precision errors. Direction errors occur when users are asked to navigate in one direction but end up scrolling in the opposite direction. This can happen when participants did not fully understand or did not perform the gesture correctly, or when the system fails to recognize the users' movements. No Output errors are considered when users move their hand with the intention of navigating, but no actual scrolling occurs. This may occur when the gesture is not wide or fast enough or when the Kinect failed to precisely recognize the motions of the user. Finally, Precision errors happen when users are scrolling in one direction to get to a particular number but, due to lack of precision of the gesture, pass it by. In this case, users have to perform another gesture to acquire the desired number. Regarding selection tasks, we considered an error when users selected a different target from the one they were required to select. We must note that not all errors were due to users' fault, but because the technology still lacks accuracy in some cases. Nevertheless, since this type of technology is still being improved, it is expected to be more accurate in the near future.

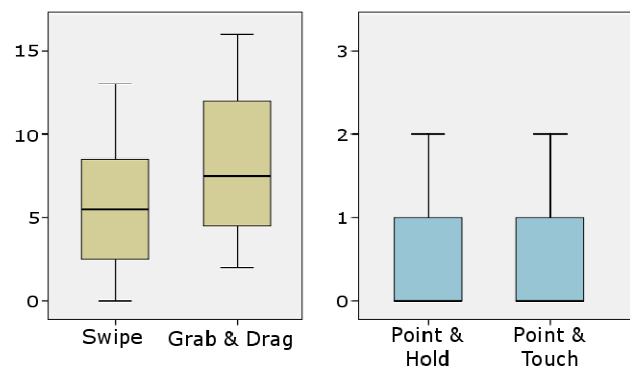


Figure 5. Number of errors for each gesture.

Figure 5 summarizes the number of errors that users made when performing the proposed tasks. As we can see, the total number of errors in both navigation tasks is similar. Indeed, a paired t test showed that there are no statistically significant differences ($p = 0.18$). However, the types of errors users committed on alternative navigation gestures are slightly different (Figure 6). By far, the most common type of error was Direction, and users made more of these errors using the Grab and Drag gesture. When performing this gesture, direction errors occurred mostly because of lack of coordination. In order to perform consecutive scrolls in the same direction using this gesture, users have to close their hand to grab, drag the hand to the desired direction, then open the hand and bring it back again to repeat this motion. However, some users would forget to open the hand between these steps, which made them scroll in the wrong direction, to the point where they started. This error occurred more frequently in the begin-

ning of the test, when users did not have so much experience. Regarding the Swipe gesture, direction errors occurred mainly because sometimes it is difficult to algorithmically interpret the intention of the user. To segment and recognize the intentions of the user in a continuous space of gestures is a complex challenge, magnified by the fact that each user has his/her own way of interacting.

Regarding the No Output errors, both navigation gestures had similar results (Figure 6). These errors occurred when users' movements were so slight that the system did not recognize them as intentions to scroll. Finally, regarding precision errors, the Swipe gesture had 50% more than the Grab and Drag (Figure 6). This is mainly because the Grab and Drag gesture allows users to get instant feedback and direct mapping of hand movements to scrolling. However, for the Swipe, the scrolling only happens after the gesture is performed. Users do not have instant feedback while performing this gesture, which does not allow a precision as good as on the Grab and Drag gesture.

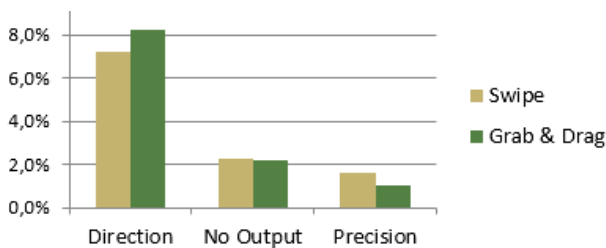


Figure 6. Percentage of errors, grouped by type of error, that participants committed in navigation tasks.

In Figure 5 we can see that for selection tasks both gestures achieved very similar results. However, for the Point and Hold gesture, most errors occurred when there were only 2 or 4 targets on screen (83% of errors). This happened because the users would start pointing at an undesired target, but they did not have enough time (selection is effective in 1.5 seconds) to adjust the hand position to the desired target, and then an erroneous target selection would occur. On the other hand, most errors that were made on the Point and Push gesture occurred when there were 8 and 16 targets on screen (60% of errors). In this case, the reason was the lack of precision users had when performing the push gesture. Indeed, users had no trouble pre-selecting the desired target by putting the hand cursor above it, but when they were performing the push gesture they would slightly move their hand and would accidentally select another target.

5.3 User Satisfaction

At the end of the user study, participants were asked to answer a satisfaction questionnaire regarding the easiness of performing the gestures, whether it was tiring, and the accuracy of the gesture detection. A 5 point Likert scale was used, with the higher score being the better. Figure 7 shows a boxplot of the results of the satisfaction questionnaire. Regarding navigation tasks, we found that both gestures achieved similar satisfaction results. A Wilcoxon signed-rank test showed no statistically significant differences between the Swipe and the Grab and Drag gestures, for every measured metric. Participants were divided be-

tween these two gestures: some would prefer the Swipes and others the Grab and Drag. For selection tasks, a Wilcoxon signed-rank test showed that there were statistical significant differences, being the Point and Hold gesture easier to perform, although with lower confidence ($Z=-1.813$, $p=0.07$). For the tiring and accuracy measures, no statistically significant differences were found.

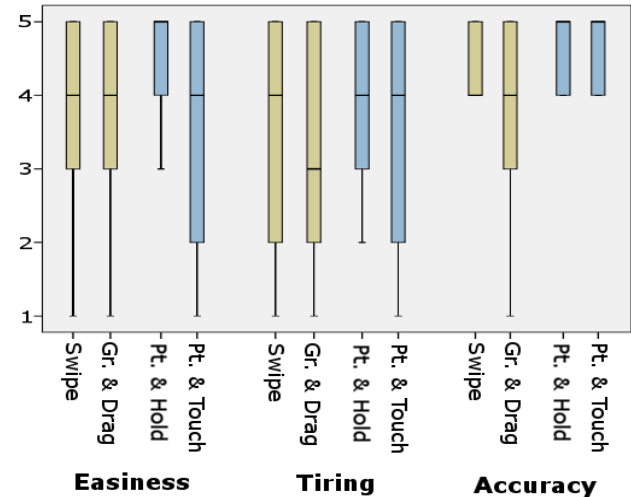


Figure 7. Results of the satisfaction questionnaire.

Besides the satisfaction questionnaire, we performed an informal interview and also gathered participants' comments while interacting. Regarding the navigation gestures, participants reported that the Swipe is easier to learn and execute than the Grab and Drag gesture. Therefore, Swipe was considered a more natural gesture. Some participants found the Grab and Drag too complex and demanding in terms of coordination, considering it a gesture that is not usually performed in everyday life and thus harder to master. Some participants also linked the difficulty of performing navigation gestures with the lack of practice. However, they were optimistic that if they had more time to practice, they would get used to it and would be able to use both gestures more proficiently.

Regarding the precision of the gestures for the navigation tasks, users reported that the Swipe did not allow for very precise scrolling, particularly when users wanted to scroll very little. Indeed, participants who were able to perform the Grab and Drag gesture usually preferred this alternative over the Swipe, since it allowed for more control and precision. However, some users reported discomfort while performing the Grab and Drag gesture, stating that since the hand palm needs to be facing the television screen, it is an uncomfortable position. Besides this, seniors did not regard either navigation gestures as tiring, except for a couple of participants who suffered from arthritis and mobility and balance issues.

The selection tasks were performed more easily when compared to the navigation tasks. For selection tasks, most users preferred the Point and Hold gesture. They reported this gesture to be very simple and easy to perform. Even when the targets got smaller, users reported that it was easy to aim and select the desired target. Participants also enjoyed the Point and Push gesture, but

reported that it was a bit more tiring to the arm. Some users started to perform the gesture with the arm already stretched. In this case, there was no room for the arm to stretch more and perform the “push” part of the gesture, resulting in users stretching the whole upper body painfully. Some users had no problem pre-selecting, i.e. getting hand over target, but when performing the push would lose precision and press on another target.

6. DISCUSSION

After analyzing all data, we can now answer the research questions proposed at the beginning of this study.

1. *Are gestural interfaces adequate for older adults to interact with general technological interfaces?* Despite not having any previous experience with gestural interfaces, most senior users performed all the proposed tasks without major problems. Only one user was not able to complete all the tasks in the Grab and Drag and the Push gestures, but she had expressed her motivation in participating in tests of “natural interaction” precisely because she had a stroke and had difficulties in understanding and concentrating. All the other participants, even the ones suffering from other health problems, had no major hindrances performing the whole user test. Moreover, previous studies found that even these low intensity exercises positively impact the health of the older people [Sapounik10]. Therefore, in short, gestural interfaces proved to be a solid alternative for older people to interact with simple technological interfaces.

2. *Which type of gesture allows for fastest navigation and selection with the lowest error rate?* Regarding navigation, the Swipe gesture outperformed the Grab and Drag gesture in terms of speed. The Swipe gesture is simpler and easier to learn and perform, and also allows users to scroll bigger distances faster. In terms of number of errors, both alternatives achieved similar results. For selection tasks, the Point and Hold gesture allows for faster selections than the Point and Push. Both gestures are similar in terms of error rate, although the Point and Hold allows for greater precision.

3. *Do older users enjoy and easily adapt to gestural interfaces? Which gestures do older users prefer?* Most seniors adapted very well to our gestural interaction. They found it easy to use and enjoyed using it. All the developed gestures achieved good rates on our satisfaction questionnaire. In terms of preference for navigation gestures, some older adults preferred the Swipe, while others preferred the Grab and Drag, which resulted in a tied satisfaction score. Regarding the selection gestures, participants agreed that the Point and Hold is easier to perform than the Point and Push.

This user study also allowed us to have some insights that may improve our original solution. One of the problems of the Swipe gesture was recognizing the direction the user intended to scroll, as some users had a way of interacting that led the system in generating a Swipe in the opposite direction. A possible solution to this problem is to only allow each hand to Swipe to a particular direction.

However, despite certainly reducing number of errors, this solution limits the number of possible interaction scenarios as it requires the user to have both hands free in order to navigate in both directions. Regarding the time and distance thresholds we imposed for the Swipe gesture, we found that they were not perfect for all users. For some users it was too wide, for other users it was too short. Therefore, despite having defined reasonable thresholds that allowed all users to adapt and perform all the proposed tasks, we conclude that each user has his/her own particular way of interacting.

We also observed that the Swipe gesture allowed users to fulfill the navigation tasks faster, but the Grab and Drag gesture allowed for more control and precision. Therefore, for technological interfaces where precision plays a big role, the Grab and Drag gesture may be a better alternative. Nevertheless, we must stress that some of our senior participants had difficulties in coordinating and performing this gesture, so it may not be the best choice for this particular age group. Regarding the Point and Hold gesture, the time required to hold over a target to select it should be increased, as participants erroneously made some errors because they did not had the time to adjust to the right target before the selection was made.

Transversal to all gestures, for we noticed that users had a tendency to perform better in the second gesture that was tested. This probably occurred due to the increased experience with gestural interfaces and also to the reduction of stress associated with a test environment, which is usually higher at the beginning. Participants also expressed that they would certainly perform better if they had more time to practice. We must also note that the conclusions we present on this paper may not apply to other cultures, as very different cultural backgrounds, either on different meanings of gestures as well as familiarity with technology, may induce differences on the final results.

7. DESIGN IMPLICATIONS

From our results, we derive the following design implications for gestural interfaces:

Keep the defined gestures as basic as possible. Gestures that may look simple for the average adult, such as Grab and Drag, may prove to be coordination challenges for older adults. Our gestures that were composed by two distinct steps (Grab and Drag, Point and Push), demanded more concentration from seniors, which led to a reduction in performance. The simpler the gesture, the easier it is to learn, which also increases motivation to keep using it.

Develop gestures that allow to be used by any hand. In this study we tried to simulate a real life scenario where gestural interfaces bring value, such as a living room. In this scenario, users may not have their dominant hand free. Therefore, and related to the previous design implication, the gesture must be simple enough to be used by the non-dominant hand. All participants have only used their dominant hand to perform the Grab and Drag gesture, but some seniors used both hands to perform the Swipe gesture in both directions. The Swipe, by being

simple enough to be performed by any hand, allows for greater freedom in interactions.

Give visual feedback of the state of the gesture recognition. In our first implementation of the Swipe gesture there was no visual feedback simply because, for this gesture, there is no direct mapping from hand movements to the elements on screen. However, senior participants felt lost when no visual cues were given, wondering on when they could perform the gesture. Participants felt more confident performing Swipes when visual feedback was given, even if it was as minimal as simply displaying an icon on screen when the user's hand was raised.

Allow personalization and adaptation. Each user has his own particularities in the way he moves, both in speed and distance. This makes static thresholds not optimal for all population. Gesture recognition is a great challenge per se, but the optimal solution involves adapting these thresholds to each user, preferably automatically. Otherwise, manual personalization should be available.

8. CONCLUSION

In this study, we showed that gesture interactions are an appropriate way for older adults to control a general technological interface. Our results showed that older people enjoyed using gestural interfaces, finding most of the evaluated gestures easy to learn and use. We found that the simpler the gestures, the better performance participants had in completing the tasks we proposed. The Swipe gesture, used for navigation, was simpler and therefore allowed users to complete the proposed tasks faster. The Grab and Drag gesture, did not have such a good performance, but allowed users to have more precision and control over the navigation process. Regarding selection tasks, the Point and Hold gesture was better since it allowed for accurate and fast selections. Older users highly rated all gestures in the satisfaction questionnaire, which means that this type of interaction was widely accepted. In general, the senior participants showed a positive attitude towards gesture-based interactions.

9. ACKNOWLEDGMENTS

We thank ActivAgeing Living Lab of Troyes University of Technology for hosting the user tests in their facilities. This work was supported by the Portuguese Foundation for Science and Technology (FCT): individual grant SFRH/BD/72735/2010; project PAELife AAL/0014/2009; and project PEst-OE/EEI/LA0021/2013.

10. REFERENCES

- [Bobeth12] Bobeth, J., Schmehl, S., and Kruijff, E. Evaluating performance and acceptance of older adults using freehand gestures for TV menu control. In Proc. of EuroTV '12, ACM (2012), New York, NY, USA, 35-44.
- [Cody99] Cody, M. J., D. Dunn, S. Hoppin, and P. Wendt (1999). Silver surfers: Training and evaluating internet use among older adult learners. *Communication Education* 48(4), 269–286.
- [Ganesan12] Ganesan, S., and Anthony, L. "Using the kinect to encourage older adults to exercise: a prototype." CHI'12 Extended Abstracts on Human Factors in Computing Systems. ACM, 2012.
- [Gerling12] Gerling, K., et al. "Full-body motion-based game interaction for older adults." Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems. ACM, 2012.
- [Henze10] Henze, N., Löcken, A., Boll, S., Hesselmann, T., Pielot, M.: Free-Hand Gestures for Music Playback: Deriving Gestures with a User-Centred Process. In: MUM'10, pp. 1–10, ACM (2010).
- [Hummels98] Hummels, C. and Stapers, P. J. (1998). Meaningful gestures for human computer interaction. Proc. of the 3rd International Conference on Automatic Face and Gesture Recognition.
- [Jung09] Jung, Y., Li, K. and Janissa, N. Games for a better life: effects of playing wii games on the well-being of seniors in a long-term care facility. In Proc. 6th Australasian Conference, IE '09, NY, USA, ACM.
- [Kim11] Kim, H., Jeong, K., Kim, S., Han, T.: Ambient Wall: Smart Wall Display Interface which can be Controlled by Simple Gesture for Smart Home. In: SA'11, pp. 1–2, Hong Kong, China (2011).
- [Kim06] Kim, D. and Kim, D. An Intelligent Smart Home Control Using Body Gestures. In: ICHIT'06, pp. 439–446, IEEE (2006).
- [Maidi13] Maidi, M., and Preda, M. Interactive media control using natural interaction-based Kinect. Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE, 2013.
- [Namazi03] Namazi, H. and McClintic, M. (2003). Computer use among elderly persons in long-term care facilities. *Educational Gerontology* 29(6), 535–550.
- [Nichols06] Nichols, T., Rogers, W. and Fisk, A. (2006). Design for Ageing. In: Salvendy, G. (Ed.). *Handbook of human factors and ergonomics*. (pp. 1418-1458).
- [Oh12] Oh, O., Jung, Y., and Cho, Y. Hands-up: Motion Recognition Using Kinect and a Ceiling to Improve the Convenience of Human Life. In: CHI EA'12, pp. 1655–1660, Texas, USA (2012).
- [Panger12] Panger, G.: Kinect in the Kitchen: Testing Depth Camera Interactions in Practical Home Environments. In: CHI'12, pp. 1985–1990, ACM (2012).
- [Saposnik10] Saposnik, G., Mamdani, M., and Bayley, M. (2010). Clinical trial protocols Effectiveness of Virtual Reality Exercises in STroke Re-habilitation. *International Journal Of Stroke*, 5(2) (pp. 47-51).
- [Venkatesh08] Venkatesh, V. and Bala, H. (2008), Technology Acceptance Model 3 and a Research Agenda on Interventions. *Decision Sciences* (pp. 273–315).
- [Znagui11] Znagui, A., Dijk, B. and Ludden, G. (2011). Touch versus in-air hand gestures: Evaluating the acceptance by seniors of human-robot interaction. In *Ambient Intelligence*, pp. 309–313. Springer.

Studying color blending for visualizing social artifacts

Sandra Gama Daniel Gonçalves
INESC-ID and Instituto Superior Técnico, Universidade de Lisboa
R. Alves Redol, 9, 1000-029 Lisboa, Portugal
sandra.gama@ist.utl.pt, daniel.goncalves@inesc-id.pt

Abstract

Given the tools and applications that are available today, we often communicate at any time and place, generating high volumes of social personal information. Visually representing such data may enable us to navigate and manage it in an effective manner. Color, due to its effectiveness in labeling and categorizing information, may highlight relevant data and further alleviate cognitive load associated with information interpretation. We have studied the use of color blending for representing social artifacts. In this paper we present a user study which we have performed to understand whether humans associate particular colors with different social artifacts. It has allowed us to understand which colors humans associate with contacts and conversation topics regarding either sent and received messages, as well as volume and variety. Results have enabled us to derive a set of design implications on how to use color blending to represent social facets when visualizing social personal information.

Keywords

Color Blending, Information Visualization, Social Interaction Visualization, Design Principles, Human-Machine Interfaces

1. INTRODUCTION

Not only is visualization a powerful way to represent information, but it also alleviates cognitive load associated with data interpretation. When it comes to coding information, using color to display data categories is usually the best choice. This suggests representing data entities as visual objects, in which colors naturally code their attributes [Ware 12]. When visualizing Social Personal Information (SPI), there are specific aspects that seem particularly relevant to users, namely contacts and keywords [fBR10]. Furthermore, representing trends associated with the ratio between sent and received messages seems relevant as well.

To take advantage of the potential of color to emphasize relevant SPI, we must understand how to use color to represent and highlight social artifacts. Hence, we have conducted a user study with focus on understanding how to use color blending in social information visualization. We have thus studied which colors to use to visualize both contacts and conversation topics, focusing on the representation of the ratio between sent and received messages regarding each facet. We also analyzed to which extent message volume and contacts' and topics' variety may be associated with color features such as brightness and saturation.

Naturally, color sets needed to be chosen to convey either contacts and topics, allowing the representation of sent-received message ratio. Color blending provides a natural choice to code this information, in which different color pairs are used to represent contacts and topics, and the ra-

tio between sent and received messages ranges from one of the original colors (meaning, for instance, *In all my communications with John, he sent all the messages*) to the other original color (meaning *I sent all the messages to John*). The same applies to messages exchanged regarding a particular topic. For instance, one of the original colors represents *In all the messages about computers I have exchanged, all were sent by me*, while the other represents *All messages about computers were sent to me*. So that visual confusion is avoided, we must choose two pairs of colors with different components. To perform such choice, we only took into account the four pairs of colors which produce good results when blended together: (*red, yellow*), (*green, yellow*), (*green, blue*) and (*red, blue*) [Gama 14].

This paper is organized as follows: in section 2 we introduce several concepts regarding color representation and present relevant work in the context of our study. Then, in section 3 we describe our study and then present and discuss the results of user tests, deriving a number of design implications. We then draw a set of conclusions regarding color for visualizing social information.

2. BACKGROUND

Color is a sensation produced in the brain [Chapman 04] that, if related to measurable phenomena, allows digital representation. It presents great potential, allowing the visualization of data entities in a way that alleviates cognitive load associated with information interpretation. One interesting use of color in visualization is color blending,

where each variable is assigned a different color and the color of the resulting data is computed as the weighted sum of the original colors. Various authors have studied different color blending techniques and a number of researchers have also considered color blending for visualization. For instance, Gossett and Baoquan [Gossett 04] aim at improving visualization using color to convey data properties. A subtractive color space has been adopted which uses red, yellow and blue as primary colors. Noise patterns are procedurally generated to create subregions of easily identifiable colors within a mixed region as a complement to color blending. On a different note, Hagh-Shenas et al. [HS06] studied information-carrying capacities of color blending and color weaving to encode multivariate information in map-reading. Livingston and Decker [Livingston 13] have also studied color blending, among other techniques, to represent trends among data layers on a demographic survey. Even though color blending yielded excellent response times, accuracy was not as promising. Although relevant in such particular contexts, previous research has not specifically studied the use of color blending in a social context, particularly regarding which colors are associated with different social facets. We analyzed such aspects, by understanding how human subjects are capable of assigning social artifacts to certain colors in particular.

3. STUDYING COLOR FOR SPI VISUALIZATION

In order to understand the best way to use color blending to represent SPI, we performed a user study with the following goals: (i) to verify which pairs of colors are more easily associated with each conversation facet (*contacts* and *subjects*); (ii) taking these pairs of colors into account, to understand which colors are associated with sent and received messages; (iii) whether other color features, such as brightness and saturation, were associated with measures such as volume and variety of contacts or topics.

3.1. Designing the study

We created a four-stage study. The first part consisted of a small number of profiling questions: age, gender, education, nationality and country of residence. Since our focus is color, at the second stage we performed a validated simplified 6-plate Ishihara color blindness test [DA92]. The third part consisted of color code questions, described in section 3.1.1. Lastly, subjects were asked to rate, given a 5-point Likert scale (1=completely disagree; 5=completely agree), the statement: *I found it easy to decide which color to associate with different facets of social information.*

3.1.1. Color code questions

This test stage was designed to understand the best colors for representing social artifacts. We started by asking subjects the aspect they consider more relevant in computer-mediated interactions (*contacts* or *topics*). Since we need two color pairs to represent sent and received messages for each facet, we created two scenarios using distinct color pairs build from different original colors. In Scenario A we considered the (*red, yellow*) and (*green, blue*) pairs and in Scenario B we considered the (*green, yellow*) and (*red,*

blue) pairs. For each scenario, we created a weekly calendar, as depicted in Figures 1 and 2.

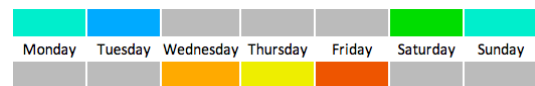


Figure 1. Weekly Calendar View (Scenario A)

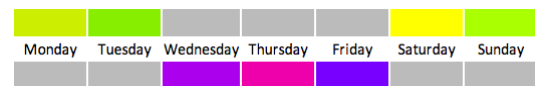


Figure 2. Weekly Calendar View (Scenario B)

On calendar views, colors from each pair were randomly assigned to each of two different sets of days (*Monday, Tuesday, Saturday and Sunday*) and (*Wednesday, Thursday and Friday*). For both scenarios, subjects were asked on which of the two sets of groups they (i) communicate with more people; (ii) discuss a wider variety of topics.

In order to find out the best colors of each pair to represent either sent or received messages, we created two new calendar views for each scenario, in which we assigned pure color components to each of the aforementioned sets of days, as depicted in Figures 3, 4, 5 and 6. Subjects were asked which set of days corresponded to a higher number of received (over sent) messages.



Figure 3. Weekly Calendar View (Scenario A: (*red,yellow*) pair).



Figure 4. Weekly Calendar View (Scenario A: (*green,blue*) pair).

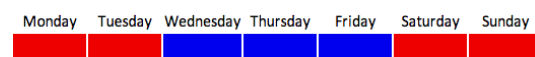


Figure 5. Weekly Calendar View (Scenario B: (*red,blue*) pair).

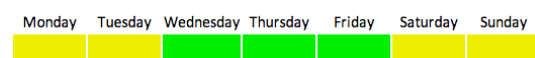


Figure 6. Weekly Calendar View (Scenario B: (*green,yellow*) pair).

In order to study possible variation of the perception of message volume and quantity of contacts or topics with brightness and saturation, we created two additional questions. On the first question, subjects were presented with a calendar view in which a group of days was represented in full brightness while the other was depicted in half the original brightness. Participants were asked to choose the group of days in which they exchanged a higher quantity of messages. On the second question, a calendar view was depicted in which a group of days was represented in full saturation while the other was depicted in half the original saturation. Subjects were asked to choose the group of days in which a wider variety of subjects were discussed or messages were exchanged with a higher number of people.

3.2. Conducting the study

After designing our study, we created a web page to perform our study as a five-step questionnaire using HTML, Javascript and Perl/CGI. We broadcasted it on social networks and kept the questionnaire open for a week.

3.3. Results

We obtained 54 responses of which 14 were incomplete, hence discarded. 40 responses were thus analyzed.

3.3.1. Subject profile

Concerning age, out of 40 participants, 11 (27.5%) were between 16 and 25, while 29 (72.5%) were between 26 and 35. Considering gender, 28 (70%) were male and the remaining 20 (30%) were female. Regarding academic background, 7 (17.5%) completed high school, 8 (20%) have BsC degrees, 23 (57.5%) hold a MsC and 2 (5%) have completed a PhD. Out of our subjects, all 40 (100%) are european. Concerning current residence, 39 (97.5%) live in Europe and 1 (2.5%) lives in Asia.

3.3.2. Color blindness

The Ishihara test showed that all 40 (100%) participants have normal color vision.

3.3.3. Color coding questions

We started the color coding questions by asking subjects which aspect was considered the most important: contacts or subjects. In fact, 21 participants (52.5%), were not able to decide. Nevertheless, 18 participants (45%) considered contacts to be the most relevant aspect, while only one person (2.5%) thought subjects to be the most important facet.

Regarding color scenario questions, results are summarized in Table 1. In scenario A, results do not establish a preference for the representation of contacts: 20 participants (50%) chose the (*red, yellow*) color pair while the remaining 20 subjects (50%) chose the (*green, blue*) pair. However, there is a stronger preference for conversation subjects. They seem to be associated with colors situated within the (*red, yellow*) color range, as picked by 28 subjects (70%) over the (*green, blue*) color pair, which was selected by only 12 participants (30%). As for scenario B, despite the slight preference for the use of colors within the (*red, blue*) range for representing contacts

Table 1. Colors associated with social facets

Scenario A	Contacts	Subjects
(<i>red, yellow</i>)	20	28
(<i>green, blue</i>)	20	12
Scenario B	Contacts	Subjects
(<i>green, yellow</i>)	18	12
(<i>red, blue</i>)	22	28

(22 subjects (55%)) over the (*green, yellow*) color pair (18 participants (45%)), there is a stronger bias for using the same color range for representing conversation subjects. In fact, 28 (70%) participants selected colors within the (*red, blue*) range for representing subjects, while only 12 subjects (30%) chose the (*green, yellow*) color pair.

Concerning the selection of colors for representing both send and received messages among the color pairs we have considered, user test results are summarized in table 2. Regarding scenario A, concerning the first color pair (*red, yellow*) there seems to be a strong preference for using *red* (28 participants (70%)) over *yellow* (12 participants (30%)) to represent received messages. Considering the second color pair (*green, blue*), there is a slight preference for using *blue* (18 participants (45%)) for representing received messages over *green* (22 participants (55%)). Regarding scenario B, there is also a strong preference for *red* in the first color set (*red, blue*) (28 participants (70%)) over *blue* (12 participants (30%)). As for the second color pair (*green, yellow*), there seems to be a slight preference on *yellow* for representing received messages (17 participants (42.5%)) over *green* (23 participants (57.5%)). Interestingly, *red* is strongly associated with received messages, either over another *hot color* such as yellow and a *cold color* such as blue. This fact suggests that participants associate the color *red* with action or attention, which is concomitant with occidental cultures' color conventions, in which *red* is used to draw people's attention to important situations, such as the need for action or imminent danger. On the other hand, *green* has had the opposite results, not being chosen to represent received messages over either yellow (17 (42.5%) against 23 (57.5%)) or blue (18 (45%) against 22 (55%)), which is concomitant with the color convention for this color and easily associated with sent messages, in which no further action is required. Since all participants have occidental provenience, there is a strong coherence between cultural color convention and unconscious perception of action associated with received messages.

Table 2. Colors for received messages

Scenario A		
(<i>red, yellow</i>)	29: Red	11: Yellow
(<i>green, blue</i>)	18: Green	22: Blue
Scenario B		
(<i>red, blue</i>)	28: Red	12: Blue
(<i>green, yellow</i>)	17: Green	23: Yellow

Regarding the possible relationship between either message volume and color brightness and variety of contacts or conversation topics and color saturation, results are summarized as follows. The majority of participants (28 (70%)) associated message volume with colors with lower brightness over colors with higher brightness (12 (30%)). Regarding contacts' and subjects' variety in exchanged messages, a very strong interrelation was evident from our test results: 36 subjects (90%) associated highly saturated colors with a higher variety of contacts and subjects while only 4 (10%) associated these less saturated colors. Hence, we may conclude that not only does lower brightness strongly suggest a higher message volume (darker colors represent a larger amount of messages) but also higher saturation is strongly associated with the exchange of messages with a greater variety of people or about a higher number of topics.

3.3.4. Satisfaction questionnaire

Regarding results for user satisfaction on the association of color with social facets, even though subjects did not find it extremely difficult to associate color with social facets (Score=1, 12.5%), an important part found it slightly difficult (Score=2, 25%). A smaller number of subjects believed this decision to be of medium difficulty (Score=3, 17.5%) and the majority found it relatively easy (Score=4, 35%), while a small number of people (Score=5, 10%) found it extremely easy. Calculating a weighted average of the scores showed users did not find it particularly natural to associate colors with social artifacts ($\bar{X}_W = 3.05$).

3.4. Design Implications

The analysis of our study results has shown that (i) Even though subjects had difficulties in deciding the most important aspect to represent, contacts are more relevant; (ii) There are stronger preferences for a particular pair to represent conversation topics, namely (*red, yellow*) (Scenario A) and (*red, blue*) (Scenario B); (iii) In one of the selected scenarios (Scenario B), there are conflicting decisions regarding the representation of either contacts or topics using the pair (*red, blue*). This suggests that color pairs considered in this scenario should not be used, in order to avoid cognitive confusion. Hence, the best color pairs to use are (*red, yellow*) and (*green, blue*); (iv) When there is a color pair which includes the color *red*, received messages are associated with that color; (v) When there is a color pair which includes the color *green*, received messages are associated with the other color of the pair (either *yellow* or *blue*); (vi) Colors with lower brightness correspond to higher message volume; (vi) Colors with higher saturation are associated with a higher variety of subjects or conversation topics in exchanged messages.

4. CONCLUSIONS AND FUTURE WORK

We have conducted a user study which aims at understanding how to take advantage of color in social information visualization. Even though participants were not able to make a decision regarding which of the aforementioned social facets is the most important, they were able to mentally

assign color to different facets: colors within the (*green, blue*) range for contacts and colors corresponding to the (*red, yellow*) pair for conversation topics. Regarding the first pair, *green* must be used to represent sent messages and *blue* to depict received messages. Concerning subjects, *yellow* must be used to represent sent messages and *red* to depict received messages. Furthermore, our study suggest that associating brightness and saturation with message volume and subjects' or topics' variety, respectively, will succeed in delivering further information. Using these results will enable our system to meet users' expectations, showing them meaningful social information in a natural, understandable way, while alleviating cognitive load.

5. ACKNOWLEDGEMENTS

This work was supported by national funds through FCT Fundação para a Ciência e a Tecnologia, under project Educare - PTDC/EIA-EIA/110058/2009 and INESC-ID multianual funding - PEst-OE/EEI/LA0021/2013.

References

- [Chapman 04] Nigel Chapman and Jenny Chapman. *Digital Multimedia*. John Wiley & Sons, 2004.
- [DA92] Dilogen D. Alwis and C. Kon. A new way to use the ishikawa test. *Journal of neurology*, 239(8):451–454, 1992.
- [fBR10] Authors Omitted for Blind Review. Title omitted for blind review. In ACM, editor, *ACM 6th Nordic Conference on Human-Computer Interaction*. ACM, 2010.
- [Gama 14] Sandra Gama and Daniel Goncalves. Studying color blending perception for data visualization. In *Eurovis 2014 - Eurographics Conference on Visualization*. Eurographics, 2014.
- [Gossett 04] N. Gossett and C. Baoquan. Paint inspired color mixing and compositing for visualization. In *IEEE Symposium on Information Visualization*, pages 113–118, 2004.
- [HS06] H. Hagh-Shenas, V. Interrante, C. Healey, and S. Kim. Weaving versus blending: A quantitative assessment of the information carrying capacities of two alternative methods for conveying multivariate data with color. In *3rd Symposium on Applied Perception in Graphics and Visualization*, pages 164–164, 2006.
- [Livingston 13] M. Livingston, J. Decker, and Z. Ai. Evaluating multivariate visualizations on time-varying data. In *Proceedings of the Conference on Visualization and Data Analysis*, 2013.
- [Ware 12] Colin Ware. *Information Visualization: Perception for Design*. Elsevier, 2012.

Sessão - Síntese e Visualização

Atmospheric Scattering - State of the Art

Diogo A. R. Lopes António Ramires Fernandes
Departamento de Informática
Universidade do Minho, Portugal
pg22830@alunos.uminho.pt, arf@di.uminho.pt

Abstract

Atmospheric scattering is the natural phenomenon mainly responsible for the colours we observe in the sky. Over the years, several realistic computer graphics algorithms have been proposed in order to reproduce these colours. This state of the art is motivated by the large amount of scattered information, and by its great potential usage in a wide range of applications like flight simulators, video games and movies. This paper will cover the most important models and will present their evolution over the years. The first part contains a small introduction to the mechanics behind the light scattering phenomena. The second part will cover the earlier models, very much focused in the physical phenomenon. The third section will cover GPU based models, more focused on performance.

Keywords

Modelling of Natural Phenomena, Atmospheric Scattering, GPU Graphics

1. INTRODUCTION

Atmospheric scattering is a natural phenomenon that can be described as the result of the interaction between sunlight and particles in the atmosphere. When we look to the sky during the day we see mostly blue, while at sunset we get a more reddish colour, specially near the horizon. Lord Rayleigh, in 1871, and Gustav Mie, in 1908, have described the behaviour of the light-particles interaction that produces these colours([Strutt 71, Mie 08]).

Rayleigh scattering describes the behaviour of light when interacting with very small particles (most of the particles present in the atmosphere). Mie scattering can be used to describe the behaviour of light interacting with any kind of particles, nevertheless, it is mostly used to describe the interaction with larger particles such as haze.

The core of the atmospheric scattering computational models is the method to solve the scattering equations of Mie and Rayleigh.

Earlier models like [Klassen 87] and [Nishita 93] used ray tracing algorithms in order to solve those equations. Latter models tended to use analytical expressions to reduce the computation complexity in the equations. Examples of this approach are the models presented in [Preetham 99] and [O'Neal 05]. The most recent models, such as [Schafhitzel 07] and [Bruneton 08], resort to pre-computation of the light scattering values in order to achieve better performances.

2. ATMOSPHERIC SCATTERING

In daylight models, the sun is considered the only light source. Due to the distance between the sun and the earth,

it is assumed that the light rays travel parallel to each other, and unmolested through the empty space until they enter the earth's atmosphere.

Upon entering the atmosphere, these rays will interact with particles in the atmosphere, the most relevant interaction, regarding colour, being scattering. Scattering occurs when a photon's electromagnetic field hits a particle's electric field in the atmosphere and is deflected into another direction [Strutt 71].

Most models consider two types of particles: air molecules and aerosol particles, the latter being much larger than the former. Air molecules can be found everywhere in the atmosphere where aerosols are more common closer to earth's surface. The result of the interaction is a function of the photon's wavelength and the size of the particle.

To better illustrate the light scattering phenomenon, let's follow a ray's path since it enters the atmosphere until it reaches the camera, ray R_1 in figure 1.

A ray R_1 , a set of photons travelling along a common trajectory, travels from the sun and reaches the top of the atmosphere (P_{sun1}) with a certain intensity I_0 . Following the same direction we see that the ray interacts by the first time with a particle at point P_1 .

This interaction has the potential to deflect the ray's photons in all directions. Only part of the photons will maintain the initial direction. Each time the ray interacts with a particle it gets further attenuation.

Still considering the initial direction, the ray further interacts with other particles until it reaches P_{scatt} , and the intensity of the ray upon reaching P_{scatt} can be computed

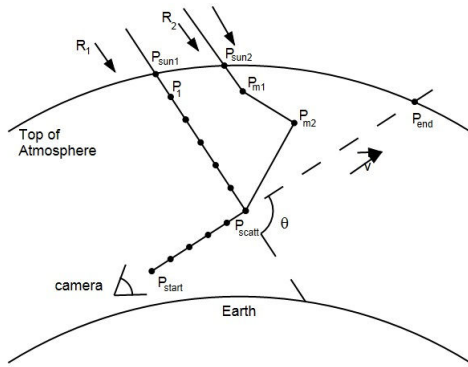


Figure 1. Evaluation of the intensity of a ray along its path

as $I_2 = I_1 att(p_A)$, where p_A is the path from P_{sun1} to P_{scatt} . The value of $att()$ represents the light attenuation, or optical depth, which can be translated as the percentage of photons that will follow the same direction.

Consider that at P_{scatt} some of the photons are scattered towards the camera. So R_1 will leave P_{scatt} with an intensity value of I_{scatt} that is given by $I_{scatt} = I_2 phase(\theta)$, where $phase(\theta)$ determines the percentage of the photons that will travel in a direction with an angle θ (figure 1), relative to original direction. The process of deflecting the light towards a given direction being considered, in this case the direction towards the camera, is also called in-scattering.

Between P_{scatt} and the camera location the ray will suffer further attenuation as it interacts with particles on the way, hence, the ray will reach the camera with an intensity $I_{final} = I_{scatt} att(p_B)$, where p_B is the path between P_{scatt} and the camera. The loss of intensity by the deflection of photons out of the original direction is called out-scattering.

The total amount of intensity reaching the camera in a direction v can be computed as the sum of all the in-scattering contributions caused by rays, parallel to R_1 , that interact with particles in the path from the camera to the exit point of the atmosphere along v .

2.1. Computing the Attenuation

Computing the attenuation requires knowing the particle density. In general, according to [Nishita 93], the density for a particular height h can be computed as $\rho(h) = exp(-h/H_{den})$, where H_{den} is a reference height, representing the atmosphere thickness where the particles can be found. For example, in [Nishita 93], for air molecules $H_{den} = 7994m$ and for aerosol particles $H_{den} = 1200m$. We shall refer to the density of air molecules as ρ_M and the density of aerosols as ρ_A . This reflects the fact that both types of particles have an exponential distribution along the atmosphere being more dense near the surface and the concentration of aerosols is only meaningful much closer to the surface.

The size of the particles also affects the interaction with different wavelengths. Attenuation is therefore a function of both the wavelength, λ , and particle density, ρ . Equation 1 shows how to compute the attenuation for particles (of type T, where T can be A or M) of a ray after travelling in path s .

$$att_T(s, \lambda) = exp\left(-\frac{4\pi K}{\lambda^4} \int^s \rho_T(p) dp\right) \quad (1)$$

where K is a constant for the standard atmosphere (molecular density at sea level).

2.2. Computing the Phase Function

The phase function determines the amount of light that is scattered in a certain direction. Gustav Mie presented a phase function that works for all particle sizes, but it is too computationally expensive. Rayleigh presented an approximation that works for particles with small sizes (10 times inferior to the photon wavelength) like air molecules (equation 2).

$$phase_M(\theta) = \frac{1}{4\pi} \frac{3}{4} (1 + \cos^2(\theta)) \quad (2)$$

Graphically the function behaves as depicted in figure 2.

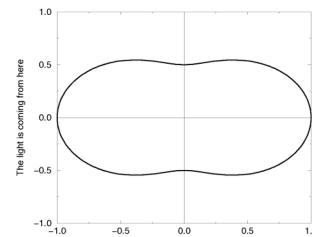


Figure 2. Rayleigh scattering phase function

Some models like [Nishita 93] proposed to use the Henyey-Greenstein phase function as an approximation to Mie's phase function for larger particles. (equation 3).

$$phase_A(\theta) = \frac{1}{4\pi} \frac{1 - g^2}{(1 + g^2 - 2g \cos(\theta))^{3/2}} \quad (3)$$

Graphically the function behaves as depicted in figure 3.

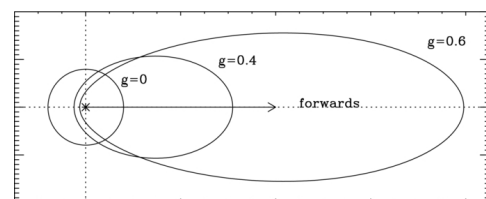


Figure 3. Rayleigh scattering phase function

The difference of these functions results can be better seen in figure 4. This figure helps to explain the behaviour of the light-particle interactions (smaller particles deflect particles more evenly than larger particles).

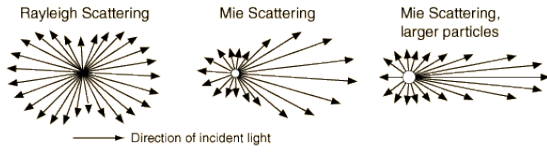


Figure 4. Types of scattering and description of its behaviour

2.3. Putting it all Together

The equation to compute the light intensity in a single wavelength λ that reaches the camera located at P_{start} , along a path s , and considering only a type of particle (A or M) is given by:

$$I_T(\lambda, s, P_{start}) = \int_{P_{start}}^{P_{end}} I_0(\lambda) att_{Tsun}(p) phase_T(\theta) att_{Tview}(p) dp \quad (4)$$

where $att_{Tsun}(p)$ is the attenuation, for particles of type T , between the entrance in the atmosphere until point p in path s , and $att_{Tview}(p)$ is the attenuation between p and the camera position.

When considering the result from both types of particles, the final intensity can be computed as:

$$I(\lambda, s, P_{start}) = I_A(\lambda, s, P_{start}) + I_M(\lambda, s, P_{start}) \quad (5)$$

The description above only considers single scattering events, i.e., the ray suffered only one event of in-scattering towards the camera. The full in-scattering value of light intensity must also take into account the scattering of light from rays that have been scattered multiple times. An example appears in figure 1 where ray R_2 is scattered multiple times until it is finally scattered into the viewing direction at point P_{scatt} . The computation of the final intensity of R_2 that reaches the camera is similar to R_1 , however, all rays reaching the atmosphere must be taken into account, as well as all scattering directions in every light-particle interaction, thus making this process highly computationally intensive.

To take into account multiple scattering up to n scattering events the term att_{sun} , in equation 4, needs to be replaced by att_{sunMS} (equation 6).

$$att_{sunMS}(n) = \sum_{i=0}^n \prod_{j=1}^{j=i} att(path(j)) phase(\theta_j) \quad (6)$$

where $path(j)$ is the path travelled by the ray between two consecutive interactions, $j - 1$ and j , and $phase(\theta_j)$ is the

angle between the direction of the ray prior to interaction j and the direction after interaction j .

In the next sections the mechanisms implemented by the various models in order to compute approximations to the scattering equations will be presented together with some resulting images.

All images were obtained by placing the camera in a latitude of 40° and longitude of 0° at 05 : 00, 14 : 00 and 20 : 00 hours.

3. EARLIEST MODELS

The earliest models attempted to solve the scattering equations by using ray-tracing algorithms and implemented a sampling strategy to solve the integrals presented in section 2.

3.1. Klassen Simple Layer Atmosphere Model

In 1987, Victor Klassen ([Klassen 87]) proposed a model of the atmosphere composed by 2 layers, each with constant density. Furthermore, the top layer, is composed only by air molecules (haze free layer) and the bottom layer, closest to the Earth's surface, is composed by both particle types (haze and air molecules). Klassen's model considers only single scattering and assumes that it is very unlikely that a in-scattering event will occur in the top layer.

Since in-scattering events occur only in the haze layer, in the top layer the attenuation is proportional to the distance covered by the rays in that layer. Hence, when entering the haze layer the intensity is given by equation 7.

$$I_1 = att(s_t, D_t) I_0 \quad (7)$$

where I_0 is the intensity before entering the atmosphere, s_t is the path covered by the ray until it reaches the bottom layer, and D_t is the density of the top layer.

Then the intensity at the in-scattering point P_{scatt} can be computed as the product of I_1 and the attenuation between the entrance point on the haze layer, and the intersection with the view direction, see equation 8.

$$I_{P_{scatt}} = att(s_h, D_h) I_1 \quad (8)$$

where s_h is the path covered by the ray in the haze layer until it reaches the view direction, and D_h is the density of the haze layer.

From P_{scatt} to the camera the light travels through path s . The full amount of light in the view direction is obtained by sampling the view direction, between the camera and the exit point of the haze layer, and compute the attenuation of all rays that are in-scattered towards the camera. The final equation, for a particular type of particle, is:

$$I_{final} = \sum_{i=0}^n att_{view}(s_i, D_h) phase(\theta) att(s_{hi}, D_h) att(s_{ti}, D_t) I_0 \quad (9)$$

where $att_{view}(s_i, D_h)$ is the attenuation between the camera and sample i along path s , and $theta$ is the angle between the view direction and the sun rays.

The attenuation itself is greatly simplified due to the assumptions of constant density within each layer as shown in equation 10.

$$att(s, D) = exp\left(-\frac{4\pi K}{\lambda^4} length(s)D\right) \quad (10)$$

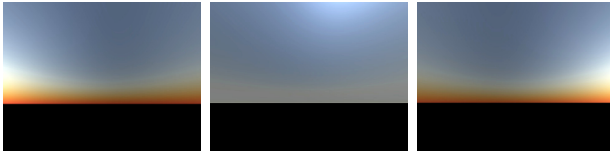


Figure 5. Images using Klassen model of the atmosphere with 2 layers. Sunrise, Midday and Sunset, respectively

The results (see Figure 5) were obtained by using the scattering coefficient values present in [Nishita 93], [Bucholt 95] and [Preetham 99], namely $D_{haze} = D_{top} = 2$, $K = 3e - 32$. Eight samples were taken in each viewing direction, and only three wavelengths were considered: red, green and blue.

Klassen’s model can reproduce various types of day, like foggy days or days with clear sky by adjusting the particle density and the height of the haze layer.

3.2. Multiple Layer Atmosphere Models

Later, [Kaneda 91] implemented single scattering with a continuous atmosphere as described in section 2. This work uses Rayleigh’s scattering equation for air molecules and Henyey-Greenstein’s scattering equation for aerosols. With a non-uniform atmosphere density, it needs to evaluate the density of air molecules and haze particles along the view direction resulting in a closer approximation to the attenuation equation (equation 1).

Optical depth integrals are approximated by sampling the ray. Where in Klassen’s model the ray is only sampled from the view direction ($P_{cam} \rightarrow P_{end}$), this model extends the sampling to the full path of the ray since it enters the atmosphere ($P_{sun1} \rightarrow P_{cam}$) (see figure 1). This model is more physically accurate than the one proposed by Klassen. On the other hand it requires a much larger computational effort.

In [Nishita 93] several performance improvements are presented. The two major computations, performance wise, are the density and the attenuation. Regarding the density, multiple spherical shells are created such that between each two consecutive shells the density difference is constant. This implies that the radius of the shells varies exponentially, resulting on a larger concentration of shells near the surface. As that the sun rays are parallel, consider a cylinder with its axis in the line that connects the centre of the earth to the sun, and a particular radius. Then all

the points in the line resulting from the intersection of a shell with the cylinder (a circle) will have the same attenuation. Consider now a set of cylinders with radius varying inversely proportional to the length of the intersection. A 2D table was built such that for each entry (i,j) the cell contains the attenuation for points in the intersection of shell i with cylinder j . To compute the attenuation for a particular point P , interpolation is used based on the altitude of the point and the distance to the centre of the earth of its projection on a plane perpendicular to the axis of the cylinders. The altitude determines the smallest outer shell that contains P , consider that this is shell i . The projection determines the smallest radius of the cylinder that contains P , assume that this is cylinder j . To determine the value of the attenuation for P linear interpolation of the closest grid locations (i,j) , $(i,j-1)$, $(i-1,j)$ and $(i-1,j-1)$ are used. This approach allows for a significant reduction of the required computations when compared to [Kaneda 91]. When comparing to Klassen’s, we get a physically more accurate model, although performance wise it is still slower despite of the pre-computations performed.

Results of using the model by Nishita et al. can produce results like the snapshots in figure 6:

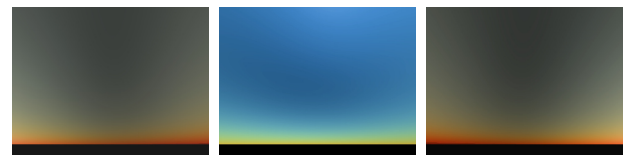


Figure 6. Images using Nishita model of the atmosphere in clear sky, Sunrise, Midday and Sunset respectively

These results were obtained by using coefficients like air and aerosol particles density, present in [Nishita 93], [Bucholt 95] and [Nishita 96], which were mentioned above in section 3.1. The g coefficient of Henyey-Greenstein phase function used was $g = 0.99$.

3.3. Nishita Multiple Scattering Model

The early models only considered single scattering events, but that changed in 1996 when Nishita et al. [Nishita 96] proposed to compute the colours of the sky by considering the contribution of multiple scattering, thus providing a better fit to the theoretical model.

For that purpose the atmosphere is divided in voxels and intensity values are stored at the vertices of the grid.

Initially, prior to runtime, the intensity reaching each voxel is computed based only on the optical depth, i.e. the direct path the rays traverse from entering the atmosphere until reaching the voxel. Furthermore, for each voxel, the intensity contributions from other voxels are stored separately. This enables second order scattering. The process can be performed recursively to achieve higher orders of scattering, but as noted by the authors the contributions beyond second order scattering are not very meaningful.

At run time, for each sample on the view direction, first they seek the voxel where the sample is located. Then all

the contributions of the vertices in that voxel are multiplied by the phase function and added together in a linear combination weighted by the distance from the sample to the vertices.

Since this method takes multiple scattering into account the results become more accurate, see `image 7`.

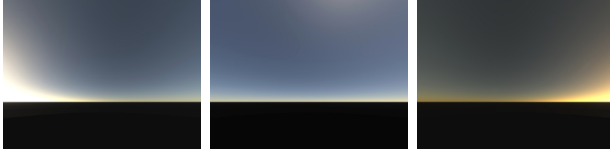


Figure 7. Images using Nishita et al. model of the atmosphere in clear sky with multiple scattering

This results were obtained by gathering 2 levels of scattering, and coefficients such as air and aerosol particles density are those presented in [Nishita 93], [Bucholt 95] and [Nishita 96], and are the same as the ones presented in the previous subsection.

3.4. Analytical Models

The main problems of the previous models is the computational time required to produce results with good quality. So, in 1995 Dobashi et al. [Dobashi 95] and Preetham et al. [Preetham 99] in 1999 proposed to reduce the number of computations necessary to obtain the colours of the sky.

To achieve this, both proposed a set of analytical functions that are capable of, given a viewing direction and the position of the sun, describe the light intensity for each visible light wavelength, and consequently, the colour of the sky in that viewing direction.

To create a set of functions capable of describing the distribution of light in the atmosphere, Dobashi et al. proposed to use a number of basis functions to express the light distribution on the atmosphere. The basis functions in [Dobashi 95] were cosine functions. The expression can be seen as the product of two terms where the first expresses the light distribution all over the atmosphere ($\Psi_i(v)$), and the second expresses the weight ($W_i^{(\lambda)}(\phi_{sun})$) that each cosine function has for a particular position in the atmosphere and for a particular viewing direction (equation 11).

$$L_\lambda(v, \phi_{sun}) = \sum_{i=0}^N W_i^{(\lambda)}(\phi_{sun}) \Psi_i(v) \quad (11)$$

These functions parameters and specifications are modelled by approximating their results with the ones produced by high quality ray tracing models, i.e., Dobashi et al. used the AS model of [Kaneda 91] to compute the values of light intensity for a series of configurations, then the functions were modelled according to the results obtained. The usage of this method reduces the quality of the result but it speeds up the rendering process, because, instead of solving the scattering integrals by sampling the ray's path, analytical expressions are used.

In 1999, Preetham et al. followed the same idea as Dobashi et al. but instead of using basis functions to compute the sky colours, a different set of functions proposed by Perez et al. in 1993 [Perez 93] were used. Preetham et al. developed a method that based on measurements of light intensity distributions in different weather conditions, modelled analytic expressions that could reproduce the colour of the sky with great accuracy and speed.

For his model, Preetham et al. considers a term called turbidity that is a measure of the fraction of scattering due to haze as opposed to molecules [Preetham 99], hence, turbidity is a measure of the thickness of the atmosphere. So, a higher value of turbidity means a denser haze, generating fog, and a lower value means clear sky.

To compute the distribution of sunlight for a given direction Preetham et al. uses equation 12 presented by Perez et al.

$$Y = Y_z F(\theta, \phi) / F(0, \theta_s);$$

$$F(\theta, \phi) = (1 + A e^{B/\cos\theta})(1 + C e^{D\phi} + E \cos^2\phi); \quad (12)$$

where $F(\theta, \phi)$ is Perez et al. model of luminance, and Y_z is the value of intensity at zenith. This formula of luminance describes the sky light distribution according to the direction of incident sunlight and some factors, such as, darkening or brightening of the horizon (A), luminance gradient near the horizon (B), relative intensity of the circumsolar region (C), width of the circumsolar region (D) and relative backscattered light (E). To model these parameters, Preetham et al. used the results produced by the model of Nishita et al. with multiple scattering [Nishita 96].

With these functions computing the sunlight intensity, in order to obtain the colour of the sky, it is only necessary to compute the light attenuation in every viewing direction. For the attenuation Preetham et al. uses pre-computed values for the fraction of light that is scattered ($S(\theta, t)$), and for the "extinction" of light ($\beta(\theta, \phi, s)$). The intensity value is computed by:

$$L(\lambda, s, \theta, \phi, t) = Y S(\theta, t) \beta(\theta, \phi, s, t) \quad (13)$$

where t is the turbidity, and $\beta(\theta, \phi, s, t)$ and $S(\theta, t)$ are:

$$\beta(\theta, \phi, s, t) = e^{(-\alpha \frac{\beta_{air}(\theta) + \beta_{aero}(\theta)}{\beta_{air} + \beta_{aero}})s} \quad (14)$$

$$S(\theta, t) = I_{air} \frac{\beta_{air}(\theta)}{\beta_{air}} + I_{aero} \frac{\beta_{aero}(\theta)}{\beta_{aero}} \quad (15)$$

α is a decay constant, s is the distance between the camera and the object (in the case of the sky, the top of the atmosphere) and the other coefficients represent the angular coefficients or phase functions ($\beta_{air/aero}(\theta)$) and the extinction coefficients or light attenuation ($\beta_{air/aero}$).

$I_{air/aero}$ is a polynomial function and represents the amount of light that is being scattered in the full path of the view direction. This function can be solved using a system of linear equations. Due to space restrictions the

system is not presented here, however, is fully detailed in [Preetham 99].

Note that the values of $S(\theta, t)$, with the exclusion of $I_{air/aero}$, and the values of $\beta(\theta, \phi, s, t)$ are pre-computed for all range of θ and ϕ . So the heaviest part of this method is computing $I_{air/aero}$. The other values are quickly obtained by a single equation (value of Y) or by just accessing a table ($\beta(\theta, \phi, s, t)$).

The turbidity value used to obtain figure 8 is 2. The remaining values can be found in [Preetham 99].



Figure 8. Images using Preetham model of the atmosphere in clear sky

The work of Preetham et al. allowed to obtain the colours of the sky faster without a significant loss in quality due to the parameter fitting according to a high quality model [Nishita 96].

This work was deeply analyzed in [Zotti 07]. The authors compared the results in [Preetham 99] with actual measurements and the ISO/CIE 2003 [oIC04] for standard general sky luminance distributions and conclude that the results for values of turbidity below 2.0 are not correct. Furthermore, the Preetham model does not properly reproduce the noticeable darkening of the sky in the antisolar hemisphere when the sun is low, with luminance values about 2–5 times too high. Also, the brightness peak towards the sun is not as steep as it can be measured or is modelled by the CIE Clear-Sky models. As a consequence of this analysis a modified version was proposed in [Hosek 12] that claims to significantly improve the rendition of sunsets and high atmospheric turbidity setups. Additionally, this model takes into account the ground albedo. From a mathematical point of view the models are very similar, with [Hosek 12] adding more coefficients to gain more control and be able to produce more accurate results.

4. Modern Models

When the graphic units became programmable, the AS models presented started to take advantage of its properties in order to enhance the performance of some already existent AS models.

4.1. GPU technique based Models

One of the first GPU models was presented by Dobashi et al. in 2002 [Dobashi 02]. The process of computing the colours of the sky is very similar to the one proposed in [Nishita 93]. The main difference consisted in using sampling planes instead of sampling directions, and with that small difference the GPU's could optimise the computation times.

In 2004 Sean O'Neal analysed [Nishita 93] and tried to optimise its performance [O'Neal 04]. O'Neal realised that

the number of required computation for a single pixel was too costly, so he proposed 2 major changes. One to improve the look-up table, another to fit some of the optical depth calculations into analytical functions.

The look-up table proposed by Nishita was able to reduce the number of calculations by half by pre-calculating the optical depth of the sunlight, and so reducing the cost of calculating the in-scattering component. O'Neal proposed an improved look-up table that also pre-computed the optical depth of the viewing ray. For that purpose two dimensions were considered, (x, y) , where x represents a specific altitude from the ground and y represents a vertical angle. For each pair (x, y) a ray is fired and it collects the atmosphere density in that viewing direction. With this optimisation both $att_{sun}(s)$ and att_{view} can be obtained interpolating the values in the look-up tables. In 2005, O'Neal,

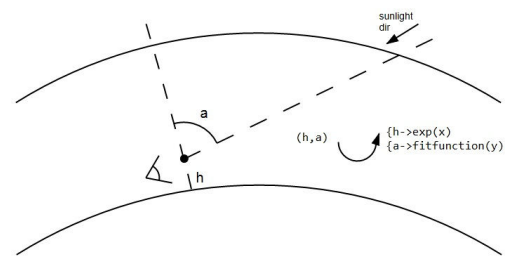


Figure 9. Look-up table elaboration by O'Neal

[O'Neal 05], went even further and tried to replace these look-up tables by analytical expressions in order to use the potential of the GPU. In order to do that, O'Neal plotted the results obtained for the various heights of x with the several angles of y and realised that the results followed an exponential distribution for the values of x with different scales in y . Knowing this, O'Neal proposed the usage of 2 analytical expressions. One that explained the exponential distribution in x , and a scale/fit function that explained the difference in scale in angle y .

To obtain the colours of the sky this method uses equation 4 and replaces the attenuation components of $\rho(s) = exp(x)$ and the $f^s = scale(y)D$, where D is the length of path s . For att_{view} , y is the angle of vision from the camera, and for att_{sun} , y is the angle made between the camera and the sun.

An important aspect of this method is that as opposed to the method in [Nishita 93], this model can use only one sample to compute the colour of the sky with only a slight reduction in quality. However the method allows the inclusion of more samples in the view ray in order to increase the quality of the result. Some results produced by O'Neal's AS model are shown in figure 10.

The exp and $scale$ functions used to obtain the results are the ones in [O'Neal 05], as are the scattering coefficients. Five samples were considered for each view ray. With this optimisations it was possible to compute the colours of the

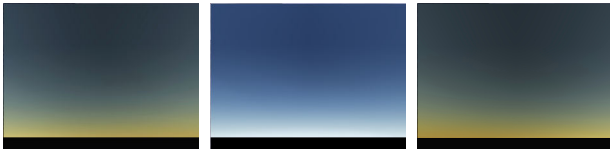


Figure 10. Images using O’Neal model of the atmosphere in clear sky

sky in real-time and still obtain a good quality result.

4.2. Pre-computed Models

After the work of O’Neal, Schafhitzel et al. in 2007 [Schafhitzel 07] and Bruneton et al. in 2008 [Bruneton 08] also took advantage of the programmability of the GPUs.

These models wanted to combine high performance with rendering quality, hence, they tried to perform the highest number of pre-computations possible in order to perform less operations in rendering time.

Nishita et al. [Nishita 93] proposed to pre-compute optical depth for sunlight and so removing the in-scatter integral. O’Neal [O’Neal 04] went further and pre-computed the out-scatter integral for the viewing direction and used analytic expressions to approximate the other integrals. Schafhitzel et al. pre-computed every single integral by assuming not only every possible sunlight and viewing direction but also every possible camera height, and store them in a 3D texture. The stored values were obtained with equation 4 with the exception of the sun light intensity (I_0).

With this approach Schafhitzel et al. only needs to fetch the values of the texture and use them to obtain the amount of light that travels towards the camera.

So they replaced the calculations of complex integrals for the calculations of indexes in a 3D table, and so the amount of in-scattered light can be given by equation 16. However, Schafhitzel et al. only computes single scattering events.

$$I(\lambda, h, a, b) = I_0(\lambda) \text{texture}(h, a, b) \quad (16)$$

In 2008 Bruneton et al. went further and pre-computed calculations for multiple scattering as well. For this purpose Bruneton et al. pre-computes and stores the values of the multiple scattering values in textures. In a process similar to the one used by Nishita et al. with multiple scattering [Nishita 96], Bruneton et al. uses multiple passes to compute this texture. At each pass the result texture sums the values of n scattering order for each sunlight direction, viewing direction and camera position/height.

The difference between Schafhitzel et al. and Bruneton et al. resides basically in the usage of equation 4 att_{sum} , where the first is using equation 1 and in the second case the equation is 6.

Bruneton et al. model produces results like the ones in figure 11.

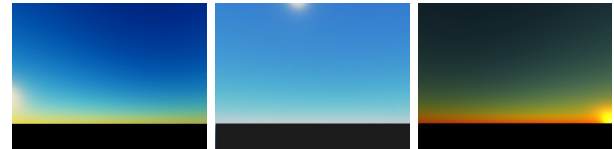


Figure 11. Images using Bruneton model of the atmosphere in clear sky

These images were obtained using the coefficients present in [Nishita 93] and [Nishita 96] with 4 orders of scattering computed.

This scheme of pre-computing the heavy integrals of the scattering equations allowed these modern models to produce high quality results in real-time. Bruneton et al. proposed some improvements for the usage of the pre-computed textures specially when rendering detailed terrains, but due to text space limitation its not approached here, however its is fully explained in [Bruneton 08].

5. PERFORMANCE OVERVIEW

The results presented for [Klassen 87], [Nishita 93], [Nishita 96] and [Preetham 99] were obtained with CPU running algorithms. The results obtained with [O’Neal 05] and [Bruneton 08] were based in GPU algorithms. The test machine had an Intel *i5-3210M* processor with 2 2.5GHz cores and a Nvidia GeForce 630m. The table presents average frame rates for our implementation of these algorithms.

Algorithm	Time
Klassen	1.42fps
Nishita 1993	0.85fps
Nishita 1996	0.056fps
Preetham	1.1fps
O’Neal 2005	230fps
Bruneton	250fps

As can be seen the simple model of Klassen produces faster results than [Nishita 93] with single scattering because it does not need to compute the density of the atmosphere particles along the view direction. We can also see that the computational cost for the multiple scattering algorithm of [Nishita 96] is about 15 times heavier than the model with single scattering [Nishita 93]. That is to be expect because far more calculations were added per frame.

The pre-computations allied to the analytical expression used in [Preetham 99] improve the time of rendering, but the most notorious aspect of these results is the power of the GPU an its usage. The results of [O’Neal 05] and [Bruneton 08] achieved real-time rendering, which is essential in several applications. Note that Bruneton et al. produces results with multiple scattering, whereas O’Neal does not.

6. CONCLUSION

The physical formulation of the problem is highly complex from a computational point of view. Sampling the integrals

and making assumptions that simplify the required computations, such as considering only single scattering and a constant density atmosphere, was the approach adopted in [Klassen 87]. In [Nishita 93] exponential decreasing density atmosphere is considered but with a performance penalty. Later, in [Nishita 96] multiple scattering is introduced, again with a penalty performance. Pre-processing and curve-fitting was the approach used the later models. Programmable GPUs also play a major role in this history as major contributors performance wise.

From Klassen's single scattering CPU based ray-tracing algorithm to Brunetton et al. GPU multiple scattering model there has been a great evolution. Quality wise earlier models are equivalent to their more recent counterparts. Performance is the key factor that has really changed. Thanks to modern graphic architectures and some pre-processing it is now possible to have realistic skies in real-time computer graphics applications.

7. ACKNOWLEDGEMENTS

Work partially funded by National Funds through FCT - Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project PEst-OE/EEI/UI0752/2011

References

- [Bruneton 08] Eric Bruneton and Fabrice Neyret. Pre-computed atmospheric scattering. *Comput. Graph. Forum*, 27(4):1079–1086, 2008.
- [Bucholt 95] Anthony Bucholt. Rayleigh-scattering calculations for the terrestrial atmosphere. 1995.
- [Dobashi 95] Yoshinori Dobashi, Tomoyuki Nishita, Kazufumi Kaneda, and Hideo Yamashita. A fast display method of sky colour using basis functions. *Journal of Visualization and Computer Animation*, 8(2):115–127, 1995.
- [Dobashi 02] Yoshinori Dobashi, Tsuyoshi Yamamoto, and Tomoyuki Nishita. Interactive rendering of atmospheric scattering effects using graphics hardware. pages 99–107, 2002.
- [Hosek 12] Lukas Hosek and Alexander Wilkie. An analytic model for full spectral sky-dome radiance. *ACM Trans. Graph.*, 31(4):95:1–95:9, July 2012.
- [Kaneda 91] Kazufumi Kaneda, Takashi Okamoto, Eihachiro Nakamae, and Tomoyuki Nishita. Photorealistic image synthesis for outdoor scenery under various atmospheric conditions. *The Visual Computer*, 7(5-6):247–258, 1991.
- [Klassen 87] R. Victor Klassen. Modeling the effect of the atmosphere on light. *ACM Trans. Graph.*, 6(3):215–237, July 1987.
- [Mie 08] G. Mie. Beiträge zur optik trüber medien, speziell kolloidaler metallösungen. *Ann. Phys.*, 330(3):377–445, January 1908.
- [Nishita 93] Tomoyuki Nishita, Takao Sirai, Katsumi Tadamura, and Eihachiro Nakamae. Display of the earth taking into account atmospheric scattering. pages 175–182, 1993.
- [Nishita 96] Tomoyuki Nishita, Yoshinori Dobashi, Kazufumi Kaneda, and Hideo Yamashita. Display method of the sky color taking into account multiple scattering. 1996.
- [oIC04] International Commission on Illumination CIE. Spatial distribution of daylight - cie standard general sky. 2004.
- [O'Neal 04] Sean O'Neal. Real-time atmospheric scattering. 2004.
- [O'Neal 05] Sean O'Neal. Accurate atmospheric scattering. 2005.
- [Perez 93] Richard Perez and Others. All-Weather Model for Sky Luminance Distribution—Preliminary Configuration and Validation. *Solar Energy*, 50(3):235–245, 1993.
- [Preetham 99] A. J. Preetham, Peter Shirley, and Brian Smits. A practical analytic model for daylight. pages 91–100, 1999.
- [Schafhitzel 07] Tobias Schafhitzel, Martin Falk, and Thomas Ertl. Real-time rendering of planets with atmospheres. *Journal of WSCG*, 15(1-3):91–98, 2007.
- [Strutt 71] J. W. Strutt. XV. On the light from the sky, its polarization and colour. *Philosophical Magazine Series 4*, 41(271):107–120, February 1871.
- [Zotti 07] Georg Zotti, Alexander Wilkie, and Werner Purgathofer. A critical review of the preetham skylight model. In Vaclav Skala, editor, *WSCG ' 2007 Short Communications Proceedings I*, pages 23–30. University of West Bohemia, January 2007.

Ray Reordering Techniques for GPU Ray-Cast Ambient Occlusion

Vasco Costa João M. Pereira Joaquim A. Jorge
Department of Information Systems and Computer Science
INESC-ID/Instituto Superior Técnico, University of Lisbon
Rua Alves Redol, 9, 1000-029 Lisboa, Portugal
vasco.costa@ist.utl.pt, jap@inesc-id.pt, jaj@inesc-id.pt

Abstract

Global illumination techniques, such as ambient occlusion, can be performed in a physically accurate way via ray casting. However ambient occlusion rays are incoherent. This means their computation is divergent causing a degradation of rendering performance. This problem is particularly acute on the GPU stream computing architectures which have performance issues with thread divergence. We reorder the rays, prior to the rendering step, to reduce this thread divergence issue. Rays which traverse the same region of space are reordered in bundles in order to increase memory coherency. We demonstrate that ray reordering techniques enhance performance while rendering scenes with ambient occlusion rays. The question is how to best perform this ray reordering. Ray reordering for ambient occlusion requires the classification of millions of rays. Spending too much time reordering these rays can negate any rendering performance benefits. Our work surveys and tests several techniques for ray reordering. We achieved the best performance results using a compress-sort-decompress technique, which sorts hashed rays, where the hash key has 32 bits of size.

Keywords

ray casting, ray reordering, gpu, ambient occlusion

1. INTRODUCTION

The availability of high performance hardware, namely GPUs, enables the use of more realistic rendering schemes. However GPUs have some limitations. These hardware architectures feature small sized caches and have simplified branch prediction hardware. Thus they are less tolerant of algorithms which feature divergent program paths or that have poor memory coherency. Unfortunately this is the case for several global illumination rendering algorithms.

In our case we are interested in performing ray-cast ambient occlusion. This technique is used, for example, to generate the baked shadow textures used in computer games. Other global illumination techniques have the same issues with lack of coherency and branch divergence. The processing of secondary rays while performing distributed ray tracing [Cook 84] is one such example.

In order to mitigate coherency issues while performing ray-casting we reorder the rays. Rays which traverse the same region of space are processed together minimizing branch divergence and improving memory coherency. This way we can improve the overall rendering performance. This work examines the performance of several techniques in regards to the ray-casting of ambient occlusion rays.

We can determine the ambient occlusion term, for a given point in the surface of an object, by ray-casting N random ray samples (see Figure 1) across the hemisphere centered

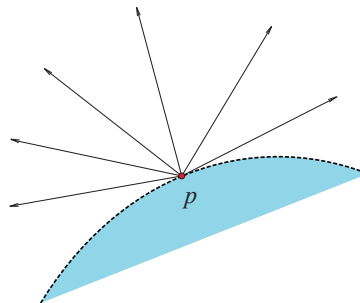


Figure 1. Ambient occlusion sampling.

around that point p oriented towards the same direction as the surface normal at that same point. These random ray samples can be generated with the aid of an R^2 Halton quasi-random number sequence.

A simple and expedient way to order these rays would be to group together rays with the same origin point. However as can be easily understood this does not guarantee the rays will not diverge significantly further along their path. Hence we need to take into account the ray directions as well while performing ray reordering.

To determine the ambient occlusion term for a 1024×1024 image with 16 samples per pixel we need to process over 16 million rays. Performing reordering with sorting on

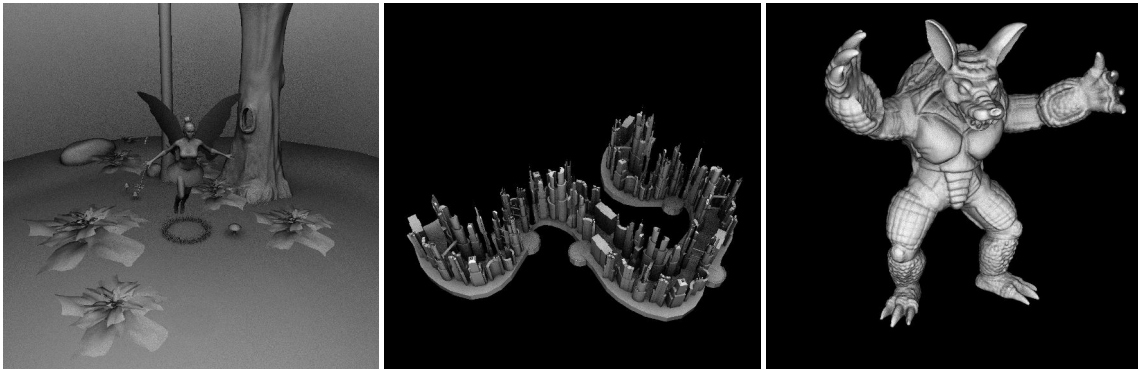


Figure 2. Fairy Forest, Serpentine City, Armadillo test scenes with ambient occlusion.

such huge lists, even in a GPU platform, can take tens or hundreds of milliseconds depending on the sorting algorithm and how accurate we want the ray sorting to be done.

Rays have five degrees of freedom. Three degrees of freedom for the origin point. Two degrees of freedom for the direction normal using spherical coordinates. If we represent each of these degrees of freedom with a 32-bit number then sorting these rays at full precision would at best require sorting large 160-bit keys. Since we are doing the sort operation just to maximize coherency it is possible to sort these rays at less than full precision using an hashing scheme. Thus speeding up the sorting process. To increase sorting speed even further it is possible to employ a compress-sort-decompress scheme, where adjacent rays with the same hash values are bundled together and sorted as a single unit, decreasing the size of the list which needs to be sorted.

Our contributions include the evaluation of ray reordering techniques in the context of GPU ray-casting. We have measured a 1.73x speedup on the rendering of scenes, with ambient occlusion rays, in our tests when using these techniques. The ray reordering techniques discussed in this work are independent of the ray tracing acceleration structure but should provide more of a performance boost in acceleration structures which degrade less gracefully as the ray coherence decreases.

The organization of this paper is as follows: we survey previous related work in detail, then we describe ray reordering techniques suitable for stream computing architectures. Next, we describe the rendering pipeline, the testing methods we adopted and present performance figures. Finally we discuss our results and present ideas for future work.

2. RELATED WORK

Arvo and Kirk [Arvo 87] described a ray tracing acceleration scheme employing *ray classification*. In this scheme a scene is partitioned in 5D ray space. To determine the primitives intersected by a ray using this scheme we simply consult the primitives list of the subspace which contains that ray.

Moon et al [Moon 10] improved ray tracing performance

on huge out-of-core scenes by reordering rays according to their hit point location. The technique computes an approximation of the hit point of each ray, i.e. the hit point heuristic, against a decimated mesh which has a quarter of the amount of triangles in the original mesh. This decimated mesh can fit into main memory unlike the original mesh at full resolution. These tentative hit points are then reordered in Z-curve order. Since this technique requires the construction of a decimated mesh it is hard to integrate with existing real-time or interactive applications.

Boulos et al [Boulos 08] used a breadth-first bounding volume hierarchy (BVH) ray packet traversal scheme. Multiple rays can be processed at a time in order to make use of SIMD parallelism. Their algorithm reorders rays when the packet utilization, i.e. the amount of active rays in a packet, drops below a certain threshold. Their algorithm also reorders shading requests by grouping together rays which intersect the same materials.

Barringer and Möller [Barringer 14] traverse a BVH using a stream based approach where the traversal loop also can process multiple rays at a time. Rays are reordered according to their directions prior to processing by sorting them into eight bins. One bin for each possible x, y, z ray sign direction.

Both of the previous techniques are limited in that that they are restricted to applications which use a bounding volume hierarchy ray tracing acceleration structure. While these approaches can be adapted for the traversal of other tree based acceleration structures, namely kd-trees, they do not map well to the traversal of grids and other non-tree based acceleration structures. The focus of our work is on acceleration structure *agnostic* ray reordering. These particular techniques are also highly tuned to the specific hardware configuration. They require extensive recoding work in order to map onto other compute architectures such as GPUs.

Aila and Laine [Aila 09] reorder secondary rays using 192-bit keys in the CPU in order to improve memory coherency and minimize divergence while performing the rendering on the GPU.

Hoberock et al [Hoberock 09] showed how to reorder shading operations in order to avoid divergence in GPU archi-

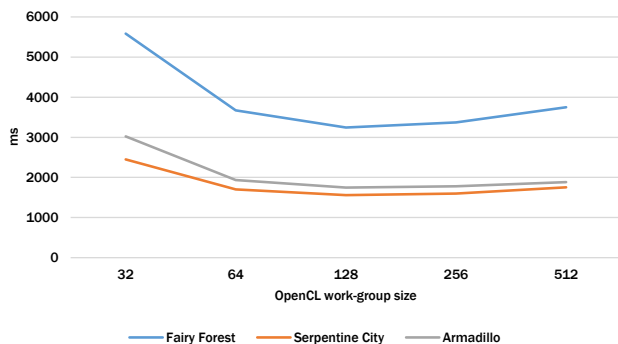


Figure 3. Time required to render the test scenes depending on the work group size.

tectures. The shading operation reordering is done with the aid of prefix sum and radix sort operations.

Garanzha and Loop [Garanzha 10] describe a breadth-first packet traversal algorithm for ray tracing which employs fast ray sorting using a compress-sort-decompress scheme to generate the ray packet bundles.

Blelloch [Blelloch 90] describes uses for the prefix sum operation in parallel architectures. We use prefix sum in our work in order to perform the compression and decompression steps. Harris et al [Harris 07] describe how to implement the prefix sum operation efficiently on GPU architectures.

Efficient GPU parallel sorting algorithms include bitonic sort [Lang 10] and radix sort [Satish 09]. The bitonic sort algorithm implementation we used has $O(N \log^2 N)$ time complexity. Radix sort implementations have $O(kN)$ complexity where k is the key size and N is the number of elements in the list to be sorted. Bitonic sort has the advantage that it can sort any items which have a comparison operator whether they are numbers or not. Radix sort is faster while sorting large lists of numbers with small key sizes.

Our work tests ray reordering techniques for ambient occlusion rays with different key sizes. From the full precision 192-bit keys to the hashed 32-bit keys. We also examine the benefit of the use of the compress-sort-decompress technique.

3. RAY REORDERING TECHNIQUES

In order to examine the performance advantage of ray reordering techniques which employ ray sorting we established a baseline set of techniques which do not employ ray sorting. Ray reordering in that case is only done with the use of the OpenCL [Munshi 11] work group construct which states which and how many compute threads should be executed in the same stream computing unit.

We devised a taxonomy for ray reordering techniques which is as follows: *static ray reordering* techniques apply a strictly fixed ray ordering scheme using the work group construct; *dynamic ray reordering* techniques sort rays according to their origin and direction values.

The ray reordering techniques we use are described in detail in the next subsections.

3.1. Static

We use two different static ray reordering techniques. These are named according to their work group arity.

The *static 1D* technique uses a 1D work group size. The optimum work group size for running a compute kernel is dependent on the characteristics of the hardware used and how many machine registers the kernel uses. In order to determine the optimum size of the work group for our hardware we conducted tests where we ran the ambient occlusion term computation algorithm with different work group sizes. As can be seen in Figure 3 the optimum work group size is 128 for all the test scenes as this provides the smallest ambient occlusion computation time. Rays with the same origin are processed together.

The *static 3D* technique uses the notion that rays with similar directions should be processed at the same time in order to maximize performance. This ray reordering technique does not require any ray sorting to be done either because the ray directions are pseudo-randomly generated using the same Halton sequence. We used a work group size of $8 \times 16 \times 1$ for a set of ambient occlusion rays of dimensions $width \times height \times nsamples$. This technique does not order the rays perfectly, since the pseudo-random generator does not always generate rays in the same overall direction for each n th sample as the object surface varies, but in practice this provides better results than just grouping together rays with the same origin as we did in the previous technique.

Both of these static techniques should also be applicable to shadow rays. The dynamic ray reordering techniques which use sorting should be much better suited for rendering reflection and refraction rays. We describe these dynamic techniques in the next subsection.

3.2. Dynamic

In the *dynamic 32-bit* technique [Garanzha 10] the 5D rays are hashed into 32-bit unsigned integer keys. These keys are then subsequently sorted. This reduces the amount of time spent doing comparisons. Since sort precision is reduced with this hashing scheme the sorted rays will be less coherent than using an exact ray sorting scheme. The HASHKEY32 function maps the 3D origin coordinates into the lower 24-bits of the key and the 2D spherical coordinates of the ray direction into the upper 8-bits of the key as can be seen in Algorithm 1.

The *dynamic 192-bit* technique [Aila 09] sorts the rays at a greater precision. The floating point ray coordinates are converted into a 192-bit key which is then sorted. The HASHKEY192 function first maps the ray origin and direction into the range $[0, 1]$, then scales the 3D origin coordinates into 24-bits each, the 3D ray direction coordinates into 21-bits each, and finally it interleaves the bits of each component with COLLECTBITS into a 6D array of 32-bit unsigned integer values. As we shall see later this tech-

Algorithm 1 Ray hashing functions.

```

function TRANSLATE(aabb, point)
  return  $\frac{\text{point} - \text{aabb}_{max}}{\text{aabb}_{max} - \text{aabb}_{min}}$ 
end function
function HASHKEY32(aabbrays, origin, direction)
   $o_{x,y,z} \leftarrow \text{TRANSLATE}(\text{aabb}_{rays}, \text{origin}) \times 256$   $\triangleright 8 \text{ bits} \times 3$ 
   $\triangleright$  translates ray origin into ray bounding box coordinates [0,1]
   $\theta \leftarrow \text{ACOS}(\text{direction}_z)$ 
   $\phi \leftarrow \text{ATAN2}(\text{direction}_y, \text{direction}_x)$ 
   $d_x, d_y \leftarrow \frac{\phi + \pi}{2\pi} \times 8, \frac{\theta}{\pi} \times 8$   $\triangleright 4 \text{ bits} \times 2$ 
  return  $\text{HASH3}(o_x, o_y, o_z) \vee (\text{HASH2}(d_x, d_y) \ll 24)$ 
end function
function COLLECTBITS(hash, idx, x)
  for all  $i \in 0 \dots 31$  do
     $k \leftarrow (idx + i \times 6) \gg 5$ 
     $\text{hash}[k] \leftarrow \text{hash}[k] \vee ((x \gg i) \wedge 1) \ll ((idx + i \times 6) \wedge 31)$ 
  end for
end function
function HASHKEY192(aabbrays, origin, direction)
   $o_{x,y,z} \leftarrow \text{TRANSLATE}(\text{aabb}_{rays}, \text{origin})$ 
   $d_{xy} \leftarrow (\text{direction} + 1.0) \times 0.5$ 
   $\text{hash}[0 \dots 5] \leftarrow 0$ 
  COLLECTBITS(hash, 0,  $o_x \times 256.0 \times 65536.0$ )
  COLLECTBITS(hash, 1,  $o_y \times 256.0 \times 65536.0$ )
  COLLECTBITS(hash, 2,  $o_z \times 256.0 \times 65536.0$ )
  COLLECTBITS(hash, 3,  $d_x \times 32.0 \times 65536.0$ )
  COLLECTBITS(hash, 4,  $d_y \times 32.0 \times 65536.0$ )
  COLLECTBITS(hash, 5,  $d_z \times 32.0 \times 65536.0$ )
  return hash
end function

```

nique gives better rendering performance at a higher sort time cost.

3.3. Compress-Sort-Decompress

The compress-sort-decompress technique [Garanzha 10] compresses the rays using run-length-encoding. Adjacent rays with the same hash values are grouped together. This reduces the amount of elements in the list to be sorted. Afterwards compressed ray bundles are decompressed into a final sorted list. This technique uses 32-bit ray hash keys. If there is too much variance between the ray keys the run-length-encoding scheme will not compress the rays well.

Prior to running the compress-sort-decompress algorithm ray hashes are computed using HASHKEY32. The algorithm works in the following order: compression via run-length-encoding (RLE), sorting of the compressed rays, decompression of the compressed rays. The overhead of the compression and decompression steps is minor as we can see in Figure 4. Once again this algorithm reduces the time spent to sort the rays since the total number of RLE blocks to sort is smaller than the total number of rays.

3.3.1. Compression

This is the run-length-encoding step. Here the *header* array is computed. Each array element is an RLE block which states the *start* position of the first ray of the block in coordinates of the initial array of rays, the *size* which stores the number of adjacent rays with an identical hash, as well as the actual *hash* value for that ray block.

The *header* array is computed as follows. We create an array H with size equal to the number of rays to process. $H[i]$ is initialized as 1 when the ray at position i differs from the previous ray at $i - 1$ and is initialized as 0 in other cases. We compute the exclusive prefix sum of H . This initializes H with the *start* positions for each RLE

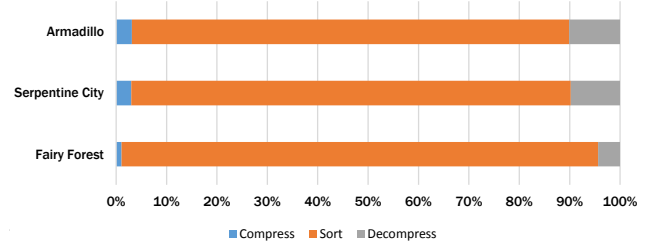


Figure 4. Percentage of total ray reordering time spent on the compress, sort, decompress stages for each test scene.

block. It also computes the total amount of blocks. We can now assign the *start*, *size*, *hash* values to each block where *size* at position i is computed via $H[i + 1] - H[i]$. The blocks have now been computed and are ready for the sort step.

3.3.2. Sorting

In this step the RLE blocks are sorted by their *hash* values. Our implementation uses a bitonic sort of $O(N \log^2 N)$ time complexity. Sorting tens of millions of 32-bit key value pairs on our system, with this algorithm, takes hundreds of milliseconds. We also tested the radix sort algorithm with $O(kN)$ linear time complexity. Radix sort takes tens of milliseconds on the same system to sort the same number of key value pairs.

3.3.3. Decompression

This is the final decompression step which returns a sorted list of rays given the intermediate list of sorted RLE blocks. We decompress these RLE blocks by first building an array with the sorted positions for each ray and secondly copying rays from their initial positions in the ray list to their final positions in the sorted ray list.

We compute the sorted positions array as follows. We create an array O with size equal to the number of RLE blocks to process. $O[i]$ is initialized with the *size* of the block at position i . We compute the exclusive prefix sum of O in order to determine the start positions, in the final ray list, for each block. Then we create an H array which will contain the final ray positions. H has the same size as the number of rays we are processing with all its elements initialized as ∞ . We compute the ray positions by initializing the ray positions array H thus. $H[O[i]]$ is initialized with the *start* ray position of a block. We do this for all the RLE blocks. Then we compute the inclusive segmented prefix sum of H which will generate the final offsets. The segmented prefix sum works in intervals, delimited by the non- ∞ elements, where the ∞ elements are treated as if they had a value of 1.

Thus we generate the final ray positions in consecutive order for each block. Once we finish copying the rays from the original list to the sorted list we can render the scene.

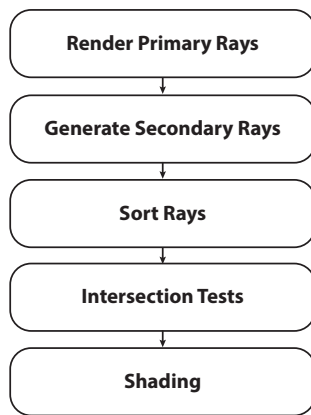


Figure 5. Rendering Pipeline.

4. RENDERING PIPELINE

The traditional ray tracing rendering pipeline works on what is known as a megakernel approach [Laine 13]. It uses an expensive, typically recursive, function aka kernel which computes the color contribution at a given pixel. This leads to an unbalanced rendering load since not all pixels require the same computation time.

Factors which can influence the computation time include: the time to traverse the ray tracing acceleration structure, the time to test the polygons intersected along the ray path, and the time to shade the materials. If the materials are elaborate the shading may take more time to compute than the time required for the polygon intersections task. In addition megakernels increase register pressure which, on a SIMT architecture such as a GPU, leads to problems when trying to issue enough concurrent threads to fully utilize available compute resources.

In order to mitigate the load balancing issues of a megakernel approach we devised a rendering pipeline which splits the rendering computation into several pipeline stages, as can be seen in Figure 5, at the cost of some additional temporary memory, for storing all the rays, in each of the intermediate stages. This allows us to optimize the load balancing at each of the processing steps required to compute the rendered image in a more fine-grained way leading to improved performance versus a megakernel. In the case of secondary rays we are also interested in applying ray reordering in order to improve coherency. This is not something which can be done using a traditional megakernel approach since, in that case, it is not possible to stop the computation midway to globally reorder all the rays. Each thread executes in a concurrent fashion and any two threads may be in completely different rendering stages at the same time.

To have acceptable rendering quality, in a modern interactive application, we require physically accurate shadows and ambient occlusion term computation. Reflections and refractions are more expensive to compute with more complicated memory access patterns. Their applicability is also much less than the previous global illumination effects

in typical scenes so we did not consider these in our current rendering pipeline in order to simplify it and further enhance performance. For example since we do not have refractions we can apply the back-face culling technique.

Our rendering pipeline works by first generating the primary rays, also known as eye rays, traversing the acceleration structure and computing the nearest intersection for each of the primary rays. Then we generate the secondary rays, which in the case of this work consist of ambient occlusion rays. The secondary rays are sorted according to their spatial distribution, where rays that traverse the same region of space are grouped together, and intersection tests are done in order to compute the ambient occlusion term. Finally we compute the shading for all the pixels in the screen taking into account the material, normal, uv coordinates, ambient occlusion term, at each pixel.

Our application supports texture mapping but this was not used for the test in this work since our main concern is profiling the performance of each of the ray reordering techniques and texture mapping would further unbalance the rendering load thus complicating the results analysis.

5. TESTING METHODOLOGY

The test platform uses an AMD FX 8350 8-core CPU @ 4.0 GHz powered machine with 8 GB of RAM. The graphics card includes a NVIDIA GeForce GTX TITAN GPU with 6 GB of RAM. The performance of the CPU is irrelevant, in our case, since all rendering and ray reordering algorithms run on the GPU. The algorithms were implemented in the OpenCL programming language.

All test images were rendered at 1024×1024 resolution with one primary ray per pixel using dot-normal (i.e. Lambertian) shading and sixteen ambient occlusion samples per primary ray. Thus rendering an image requires ray-casting 17.83 million rays.

We selected three test scenes (Serpentine City, Fairy Forest, Armadillo) representative of typical 3D gaming applications which can be seen in Figure 2.

Ray-triangle intersection adopts the Möller-Trumbore [Möller 97] algorithm since it does not have additional memory or precomputation costs. Each triangle needs 36 bytes of memory to store vertexes. Triangles with normals require an additional 36 bytes of memory. This is done to ensure more coherent memory accesses than an indexed vertex scheme would be able to provide. In our experience storing the triangles this way improves the rendering performance by 10% over the indexed method.

Sorting is generally done using the $O(N \log^2 N)$ bitonic sort algorithm. In the cases where we used the $O(kN)$ radix sort this is stated in the text.

6. RESULTS DISCUSSION

The speedup obtained with ray reordering techniques increases as the amount of incoherent rays as part of the total work load increases. As can be seen on the left of Figure 6 the Fairy Forest scene benefited the most from ray reordering. This is because the camera parameters we

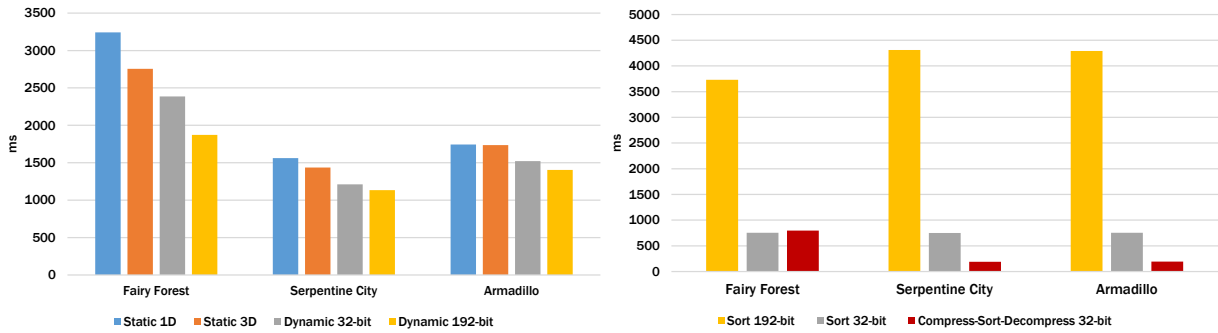


Figure 6. The chart on the left displays the time to render the test scenes for each of the ray ordering methods. The chart on the right displays the time required to reorder the rays for the dynamic reordering methods. The static reordering methods do not require any additional ray reordering time.

used for that scene require more ambient occlusion rays to be traced. In the Fairy Forest scene geometry covers the whole screen space whereas for the Serpentine City and Armadillo scenes a lot of the screen space is an empty background. In addition Serpentine City and Armadillo are manifolds so the computation of the ambient occlusion term is much more cache coherent than for the Fairy Forest scene which is more representative of a typical game scene.

As expected the more accurate hashing schemes have better rendering performance. The static techniques, which use a fixed ordering scheme, lead to worse rendering times than the dynamic techniques which compute actual ray hash keys and group together those rays which traverse the same regions of space.

On the right of Figure 6 we can see that the dynamic 192-bit key technique takes a long time to sort the ambient occlusion rays. In fact it takes longer to sort the 192-bit ray keys, using the bitonic sort algorithm, than to render the whole scene using the simplest static technique. When we reduce the key size to a sixth of 192-bits i.e. 32-bits the time required to sort the rays is reduced by a similar amount. The total amount of memory operations and comparisons is much less. If we use radix sort instead of bitonic sort the sort times are reduced from hundreds of milliseconds to tens of milliseconds thus making the dynamic ray reordering methods the most viable choice to improve ambient occlusion ray rendering performance.

The compress-sort-decompress scheme further improves sort performance for the Serpentine City and Armadillo scenes. This is due to it grouping together the background rays which take less time to render. In other words by simultaneously processing bundles of rays which take more or less the same amount of time to finish we are reducing the amount of bubbles in the GPU workload thus improving performance. This is an important fact which should not be underestimated as this can also be applied to other rendering tasks such as shading operations where not all materials have the same computational complexity level.

The performance gap between the 192-bit and 32-bit key

dynamic techniques is large enough to see that there is still more work to be done with the hash functions. One thing which could also be improved is the way the ray origins are hashed. The current hashing algorithm uses a virtual grid with the same resolution in all x, y, z axis but the scene geometry is seldom uniformly distributed across all axis. The virtual grid should therefore be able to better adapt to the distribution of geometry in the scene. Instead of using this virtual grid for the ambient occlusion rays we could use the acceleration structure (e.g. bvh, kd-tree, grid) cell id which contains the ray origin instead as this is compact and the acceleration structure cell id has previously been computed when doing the prior rendering pass for the primary rays.

	SERPENTINE CITY (138.63 KTriangles)	FAIRY FOREST (173.98, KTriangles)	ARMADILLO (345.94 KTriangles)
RENDER TIME			
<i>static</i>			
1D	1561 ms	3242 ms	1744 ms
3D	1436 ms	2757 ms	1736 ms
<i>dynamic</i>			
32-bit	1212 ms	2387 ms	1522 ms
192-bit	1133 ms	1873 ms	1403 ms
REORDER TIME			
<i>dynamic</i>			
sort 192-bit	4310 ms	3732 ms	4289 ms
sort 32-bit	752 ms	754 ms	753 ms
c.s.d. 32-bit	189 ms	798 ms	192 ms

Table 1. Performance results of the techniques for the test scenes. The static techniques do not require a reordering step prior to rendering. The dynamic techniques perform ray reordering with bitonic sorting.

As can be seen in Figure 4 most of the time that is spent with the compress-sort-decompress technique is sort time. Despite neighboring rays with the same hash being grouped together. When the hash function has more collisions rays are compressed better resulting in less sort time however hash functions with more collisions are more inaccurate and result in worse ambient occlusion computation time. It is necessary to strike a fine balance between the sort time and the render time to provide the best overall performance.

7. CONCLUSIONS AND FUTURE WORK

In this work we demonstrated that ray reordering speeds up the rendering of incoherent rays on stream computing architectures. We have measured a 1.73x speedup while rendering scenes with a lot of incoherent rays where we had 16 ambient occlusion samples per pixel. Instead of sorting the rays themselves it is much faster and more effective to sort the 32-bit hashes of the rays reducing the time spent on sorting. For interactive applications it is best to use a sorting algorithm such as radix sort since bitonic sort does not have enough performance to render scenes with tens of millions of rays at interactive frame rates.

There is still a lot of work to be done regarding ray hash functions in order to improve ray reordering performance. There is a large rendering performance gap between the 32-bit and 192-bit hash functions which we intend to explore in the future by experimenting with different key sizes and hash functions.

Since ray reordering is a task scheduling technique we should in the future not only take into account memory coherency, where neighboring rays are coalesced, but we should also take into account the ray intersection times into the hash function in order to guarantee better performance.

8. ACKNOWLEDGEMENTS

We thank the NVIDIA Corporation for the donation of the GeForce GTX Titan used for this research.

This work was supported by national funds through FCT - Fundação para a Ciência e Tecnologia, under project PEst-OE/EEI/LA0021/2013.

We would like to thank the Utah Animation Repository (Fairy Forest), Herminio Nieves (Serpentine City), and the Stanford 3D Scanning Repository (Armadillo) for the test scenes.

9. REFERENCES

- [Aila 09] Timo Aila and Samuli Laine. Understanding the Efficiency of Ray Traversal on GPUs. In *Proceedings of the Conference on High Performance Graphics 2009*, pages 145–149, 2009.
- [Arvo 87] James Arvo and David Kirk. Fast Ray Tracing by Ray Classification. *ACM SIGGRAPH Computer Graphics*, 21(4):55–64, 1987.
- [Barringer 14] Rasmus Barringer and Tomas Akenine-Möller. Dynamic Ray Stream Traversal. *ACM Transactions on Graphics (TOG)*, 33(4), 2014.
- [Blelloch 90] Guy E Blelloch. Prefix Sums and Their Applications. Technical report, Carnegie Mellon University, 1990.
- [Boulos 08] Solomon Boulos, Ingo Wald, and Carsten Benthin. Adaptive Ray Packet Reordering. In *Proceedings of the IEEE Symposium on Interactive Ray Tracing 2008*, pages 131–138, 2008.
- [Cook 84] Robert L Cook, Thomas Porter, and Loren Carpenter. Distributed Ray Tracing. *ACM SIGGRAPH Computer Graphics*, 18(3):137–145, 1984.
- [Garanzha 10] Kirill Garanzha and Charles Loop. Fast Ray Sorting and Breadth-First Packet Traversal for GPU Ray Tracing. *Computer Graphics Forum*, 29(2):289–298, 2010.
- [Harris 07] Mark Harris, Shubhabrata Sengupta, and John D Owens. Parallel Prefix Sum (Scan) with CUDA. *GPU gems*, 3(39):851–876, 2007.
- [Hoberock 09] Jared Hoberock, Victor Lu, Yuntao Jia, and John C Hart. Stream Compaction for Deferred Shading. In *Proceedings of the Conference on High Performance Graphics 2009*, pages 173–180, 2009.
- [Laine 13] Samuli Laine, Tero Karras, and Timo Aila. Megakernels Considered Harmful: Wavefront Path Tracing on GPUs. In *Proceedings of the Conference on High Performance Graphics 2013*, pages 137–143, 2013.
- [Lang 10] HW Lang and FH Flensburg. Bitonic sort. <http://www.iti.fh-flensburg.de/lang/algorithmen/sortieren/bitonic/bitonicen.htm>, May 2010.
- [Möller 97] T. Möller and B. Trumbore. Fast, Minimum Storage Ray-Triangle Intersection. *Journal of Graphics Tools*, 2(1):21–28, 1997.
- [Moon 10] Bochang Moon, Yongyoung Byun, Tae-Joon Kim, Pio Claudio, Hye-Sun Kim, Yun-Ji Ban, Seung Woo Nam, and Sung-Eui Yoon. Cache-oblivious ray reordering. *ACM Transactions on Graphics (TOG)*, 29(3):28, 2010.
- [Munshi 11] Aaftab Munshi, Benedict Gaster, Timothy G Mattson, and Dan Ginsburg. *OpenCL Programming Guide*. Pearson Education, 2011.
- [Satish 09] Nadathur Satish, Mark Harris, and Michael Garland. Designing Efficient Sorting Algorithms for Manycore GPUs. In *Proceedings of IEEE International Symposium on Parallel & Distributed Processing 2009*, pages 1–10, 2009.

Visualização Dinâmica de Nuvens através de Diagramas Termodinâmicos

Rui P. Duarte José F. Morgado
Escola Superior de Tecnologia e Gestão de Viseu, IPV
Viseu
{pduarte, fmorgado}@estv.ipv.pt

Abel J. P. Gomes
Instituto de Telecomunicações, UBI
Covilhã
agomes@di.ubi.pt

Resumo

A simulação e renderização de fenómenos naturais tem sido um dos grandes desafios em computação gráfica devido às suas aplicações em filmes de animação, ambientes virtuais e jogos. Este problema advém do facto de a formação, movimento e extinção de nuvens serem, naturalmente, processos amorfos e dinâmicos. Este artigo propõe-se a resolver este problema através de diagramas termodinâmicos SkewT/LogP. Estes diagramas constituem uma técnica 2D para simular nuvens em 3D a partir de dados atmosféricos disponibilizados por agências meteorológicas. Para atingir taxas de tempo real, o nosso sistema baseia-se na física, mas evita resolver equações diferenciais para representar, controlar, simular e renderizar o processo termodinâmico da ascensão de nuvens na atmosfera.

Palavras-Chave

Simulação de Nuvens, Métodos Procedimentais, Diagramas SkewT/LogP, Vento, Dados Atmosféricos

1. INTRODUÇÃO

A simulação realística de nuvens em ambientes sintéticos é, há muito, um tópico de investigação em computação gráfica. Contudo, a simulação de nuvens em computador é uma tarefa complexa, uma vez que estas possuem formas dinâmicas que variam ao longo do tempo.

No nosso trabalho implementámos uma ferramenta 2D dinâmica baseada em diagramas SkewT/LogP para representar, controlar e simular o processo termodinâmico das nuvens na atmosfera, usando dados obtidos através de agências meteorológicas existentes em várias partes do mundo. Este simulador 2D é usado para ‘alimentar’ o nosso ambiente 3D onde as nuvens se desenvolvem por influência de forças verticais de fluatibilidade, bem como forças horizontais devidas ao vento.

Desta forma, as principais contribuições do nosso trabalho são as seguintes:

- *Diagramas SkewT/LogP*. Tanto quanto sabemos, este é o primeiro simulador de nuvens que usa diagramas SkewT/LogP em computação gráfica. Em vez de usar taxas de decréscimo fixas para a temperatura ambiente ([Neyret 97] e [Grudzinski 07]), usamos dados obtidos de previsões atmosféricas, nomeadamente <http://weather.uwyo.edu>, <http://www.twisterdata.com>, e <http://www.woweather.com>. Estes dados incluem perfis de temperatura que permitem determinar automaticamente a base da nuvem e o seu topo através de sim-

ples intersecções geométricas realizadas no diagrama SkewT/LogP. Esta abordagem permite resolver um problema comum em métodos procedurais que é o ajustamento de parâmetros atmosféricos.

- *Resolução de Equações de Movimento em Tempo Real*. O processo de formação de nuvens está principalmente relacionado com a convecção térmica e estabilidade atmosférica. Usando diagramas SkewT/LogP conseguimos, explicitamente, determinar a força vertical necessária para resolver a equação de movimento sem resolver as equações diferenciais correspondentes.

O artigo está organizado da seguinte forma: A Secção 2 descreve brevemente o trabalho realizado em simulação de nuvens. A Secção 3 aborda a física das nuvens, em particular a equação do movimento. A Secção 4 apresenta em detalhe os diagramas SkewT/LogP e as curvas que o compõem de modo a resolver a equação do movimento. Na Secção 5 é apresentado o método de simulação. A Secção 6 apresenta resultados relevantes produzidos pelo simulador e a Secção 7 conclui o artigo apresentando trabalho futuro.

2. TRABALHO PRÉVIO

A simulação de nuvens baseia-se na teoria física de dinâmica de fluídos ou em técnicas procedimentais. A maior parte dos métodos baseados em física requer a resolução das equações de Navier-Stokes que consomem

bastantes recursos computacionais. Versões simplificadas destas equações têm sido propostas na literatura, nomeadamente os *Stable Fluids* [Stam 99] e *Coupled Map Lattice (CML)* ([Miyazaki 01] e [Harris 03]) usando a GPU (*Graphics Processing Unit*), embora não se consiga garantir que se obtenha taxas de tempo real.

Métodos procedimentais na simulação de nuvens constituem uma tentativa de evitar o peso computacional inerente aos métodos baseados em física. Tipicamente, estes métodos geram a distribuição de densidade de nuvens, usando a ideia de fractais [Voss 85], ruído ([Perlin 85], [Schpok 03], [Man 06]), Síntese de Fourier ([Gardner 85] e [Elinas 00]) funções implícitas volumétricas ([Trembilski 02] e [Wither 08]), autómatos celulares [Dobashi 00] e sistemas de partículas ([Neyret 97], [Bouthors 04] e [Grudzinski 07]).

No que respeita à simulação de nuvens a partir de dados atmosféricos, pouco trabalho foi feito. Trembilski [Trembilski 02] gerou uma superfície de nuvens a partir de dados gerados por simulações atmosféricas. A iso-superfície é determinada usando o algoritmo de *marching cubes*. Wenke et al. [Wenke 12] modelou a forma de nuvens ao nível do globo onde cada ponto da grelha é tratado com um sistema de partículas.

A nossa abordagem usa sistemas de partículas e considera a advecção térmica baseada em forças verticais de flutuabilidade e forças horizontais relacionadas com o vento, obtidas a partir de dados atmosféricos. Neyret [Neyret 97] gera nuvens considerando que a altitude a que as nuvens se formam é constante e conhecida e também que a variação da temperatura ambiente e da partícula dentro da nuvem são constantes. No nosso método, estes parâmetros são determinados a partir dos dados atmosféricos para cada partícula da massa de ar. Grudzinsky [Grudzinski 07] usa uma função exponencial para gerar trajetórias para as partículas de ar e uma função geradora que atribui a cada partícula um conjunto de parâmetros. No nosso método, a trajetória da partícula é determinada pelo diagrama SkewT/LogP.

3. FÍSICA das NUVENS

A ascensão de nuvens na atmosfera é feita através de forças de flutuabilidade ou mecânicas que trazem o arrefecimento da parcela de ar quando a pressão atmosférica diminui, resultando daí a expansão da nuvem. Eventualmente, esta massa de ar fica saturada pelo que se condensa em gotículas para formar uma nuvem. Isto implica que haja libertação de calor latente que aquece a parcela de ar, originando um aumento da sua flutuabilidade, o que, por sua vez, induz a sua ascensão na atmosfera. Quando a parcela de ar atinge a sua altitude máxima, começa a descer como consequência de flutuabilidade negativa, até que o equilíbrio seja atingido. Embora intuitivo, este processo convectivo é regulado por leis da termodinâmica. Compreender estas leis permite prever a formação de nuvens, bem como determinar o perfil vertical da sua temperatura na atmosfera.

3.1 Equações do Movimento para a Nuvem

Uma parcela de ar seco que evolui na atmosfera é descrita por três propriedades: temperatura (T , em K), pressão (P , em N/m^2) e densidade (ρ , em kg/m^3). Estas propriedades estão relacionadas pela lei do gás ideal [Cengel 11], que estabelece que

$$p = \rho RT \quad (1)$$

onde R é a constante do gás, que assume o valor $R_d = 287.05307 \text{ Jkg}^{-1}\text{K}^{-1}$ para ar seco; para ar húmido, R é dado por $R_m = 461.5 \text{ Jkg}^{-1}\text{K}^{-1}$ e T é substituída por T_v , designada por temperatura virtual, que é dada, aproximadamente, por

$$T_v = T(1 + 0.6 d) \quad (2)$$

onde d é a densidade do vapor de água.

Na direção vertical, a gravidade é a força externa mais importante que actua na atmosfera. A relação entre pressão e gravidade é descrita pelo equilíbrio hidrostático dado por

$$\Delta P = -\rho g \Delta z \quad (3)$$

onde ΔP é a variação da pressão atmosférica, Δz é a variação da altitude e g é a aceleração devida à gravidade ($g = 9.8 \text{ m/s}^2$). Assim sendo, a pressão do ambiente e a pressão da parcela de ar em z e $z + \Delta z$ dizem-se em equilíbrio hidrostático quando

$$\left(\frac{dP_e}{dz}\right)_z = -\rho_e(z)g = -\rho_p(z)g, \quad (4)$$

$$\left(\frac{dP_e}{dz}\right)_{z+\Delta z} = -\rho_e(z + \Delta z)g \quad (5)$$

onde P_e denota a pressão do ambiente e ρ_e e ρ_p são a densidade do ambiente e da parcela, respectivamente. Para a parcela de ar em $z + \Delta z$, a força vertical ascendente é apenas o gradiente de pressão do ambiente, $(dP_e/dz)_{z+\Delta z}$ e a força vertical descendente é $-\rho_p(z + \Delta z)g$. Uma vez que $\rho_p(z + \Delta z) \neq \rho_e(z + \Delta z)$, existe um desequilíbrio na força resultante que actua sobre a parcela de ar que dá origem à sua aceleração. Da lei do movimento de Newton [Crowell 00], tem-se

$$\rho_p(z + \Delta z) \frac{dv}{dt} = -(dP_e/dz)_{z+\Delta z} - \rho_p(z + \Delta z)g \quad (6)$$

onde v é a velocidade vertical da parcela de ar em $z + \Delta z$, dv/dt é a aceleração vertical da parcela de ar.

A lei do gás ideal e as leis da termodinâmica podem ser usadas para derivar a equação de Poisson para processos adiabáticos [Rogers 89], que é dada por:

$$\left(\frac{T}{T_0}\right) = \left(\frac{P}{P_0}\right)^k, \quad (7)$$

onde T_0 e P_0 são os valores iniciais de temperatura e pressão, e T e P são os correspondentes valores após uma alteração adiabática em altitude e

$$k = \frac{R_d}{c_p} = \frac{c_p - c_v}{c_p} \approx 0.286, \quad (8)$$

PRES (hPa)	HGHT (m)	TEMP (C)	DWPT (C)	RELH (%)	MIXR (g/Kg)	DRCT (deg)	SKNT (knot)	TWTB (K)	TVRT (K)	THTA (K)	THTE (K)	THTV (K)
1004.6	3	22.2	20.9	91	15.7	173	12	21.4	25.0	295.0	340.2	297.8
1000.0	39	22.8	21.1	90	16.0	168	20	21.7	25.6	295.9	342.2	298.7
975.0	260	21.6	20.2	91	15.5	170	31	20.7	24.3	296.8	341.8	299.6
950.0	486	20.4	19.2	93	15.0	171	38	19.6	23.0	297.8	341.5	300.5
925.0	716	18.9	17.8	93	14.1	177	41	18.2	21.3	298.6	339.7	301.1
900.0	952	17.4	16.3	93	13.1	186	42	16.7	19.6	299.4	337.9	301.7
875.0	1193	15.8	14.9	94	12.3	194	43	15.2	17.9	300.1	336.4	302.3
850.0	1439	14.2	13.4	95	11.5	202	42	13.7	16.1	300.9	335.0	303.0

Figura 1. Exemplo de uma sondagem atmosférica. Os rótulos das colunas da tabela são dados por: PRES - pressão atmosférica, HGHT - altitude; TEMP - temperatura atmosférica; DWPT - temperatura de condensação; RELH - Humidade relativa; MIXR - razão de mistura; DRCT - direção do vento; SKNT - intensidade do vento.

onde c_p e c_v são os valores da capacidade do calor específico do ar seco considerando pressão e volume constantes.

Em ciências atmosféricas, a temperatura potencial (θ) é uma variável mais conveniente para considerar alterações de temperatura e pressão, uma vez que o seu valor se mantém constante com alterações adiabáticas de altitude, a qual é dada por:

$$\theta = T \left(\frac{P_0}{P} \right)^{R_d/c_p}, \quad (9)$$

Usando temperatura potencial em vez da temperatura absoluta na Eq. (6), obtém-se a forma fina da equação do movimento em termos da temperatura potencial:

$$\frac{dv}{dt} = g \left[\theta_p(z + \Delta z) - \theta_0(z + \Delta z) \right] / \theta_p(z + \Delta z) \quad (10)$$

Portanto, a Eq. (10) pode ser usada para determinar o deslocamento vertical de uma parcela de ar na atmosfera, calculando-se para isso a temperatura potencial da parcela de ar e do ambiente, num determinado nível de pressão. Contudo, para diminuir o tempo de cálculo, usamos diagramas SkewT/LogP para calcular essas duas temperaturas potenciais, como será explicado mais à frente.

3.2 Vento

O vento é considerado uma força horizontal que resulta de diferenças de temperatura na atmosfera, causando diferenças de pressão que têm um grande impacto na forma e no movimento das nuvens. Para o cálculo desta força são necessários dois parâmetros: direção α (em graus) e intensidade r (em nós), fornecidos em formato tabular (colunas 7 e 8 dos dados da Fig. 1, respetivamente).

A representação do vento é feita usando coordenadas polares (r, α), contudo para incorporar o vento na advecção de uma nuvem torna-se necessário converter as coordenadas polares para cartesianas, usando a relação:

$$\begin{aligned} x &= r \cos(\alpha) \\ y &= r \sin(\alpha) \end{aligned} \quad (11)$$

Desta forma, é possível determinar o perfil de vento 3D a partir de dados atmosféricos, que é fundamental para

complementar a equação do movimento da nuvem com variações horizontais que resultam da influência do vento, como será mostrado no Algoritmo 3.

4 DIAGRAMAS SkewT/LogP

Um diagrama SkewT/LogP [AWS 79] é uma ferramenta meteorológica usada para a observação de elementos atmosféricos em qualquer plano de pressão da atmosfera, representado por uma linha isobárica (ver Fig. 2(a)). Estes diagramas permitem-nos determinar três parâmetros fundamentais para as três fases da formação de uma nuvem (Fig. 2(b)): temperatura convectiva (CT), nível de condensação convectiva (CCL) e nível de equilíbrio (EL).

4.1 Curvas de um Diagrama SkewT/LogP

Num diagrama SkewT/LogP existe um conjunto fundamental de curvas (ou linhas) que representam propriedades *standard* da atmosfera e serão fundamentais para calcular os parâmetros referidos anteriormente. Essas curvas são de cinco tipos (conforme se pode ver na Fig. 2(a)):

- *Isobáricas*. Isobáricas são linhas horizontais (a preto) que representam pontos do diagrama de pressão constante.
- *Isotérmicas*. Isotérmicas são linhas retas oblíquas (a verde musgo) traçadas do canto inferior esquerdo para o canto superior direito do diagrama. São linhas paralelas traçadas a 45° relativamente à linha $y = 0$ (i.e., com declive de 45°) em coordenadas cartesianas.
- *Linhas de razão de mistura*. A razão de mistura w refere-se à quantidade de vapor de água numa massa de ar e é dada pela seguinte expressão:

$$w(T, P) = \frac{\epsilon \cdot e(T)}{P - e(T)} \quad (12)$$

onde $\epsilon = R_d/R_m$ e $e(T)$ define a pressão do vapor saturado. Num diagrama SkewT/LogP, linhas de razão de mistura são traçadas como linhas retas tracejadas (a castanho) a partir do canto inferior esquerdo até ao canto superior direito do diagrama.

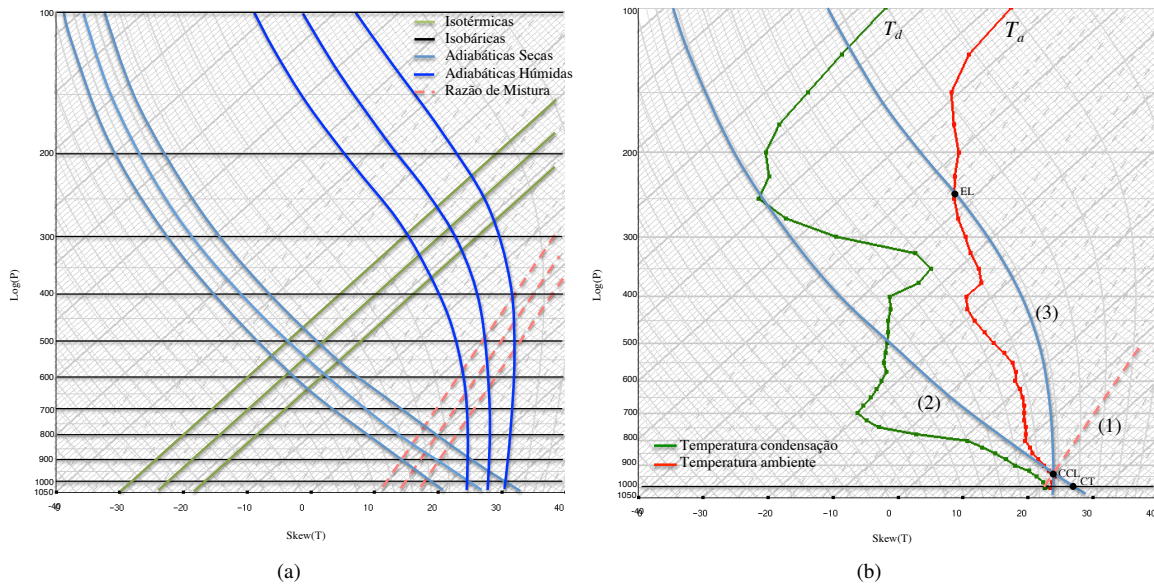


Figura 2. Diagramas SkewT/LogP. (a) Curvas que definem o diagrama (b) estimação de diversos parâmetros necessários para a formação de nuvens: (1) Cálculo do CCL, (2) Cálculo de CT e (3) Cálculo de EL.

- *Adiabáticas secas.* As curvas adiabáticas secas são curvas do tipo $1/\log P$ (em azul marinho) no diagrama SkewT/LogP. Estabelecem o comportamento termodinâmico de parcelas de ar não saturadas que se movem ascendentemente (os descendente), ou seja, descrevem a taxa de variação para o processo adiabático seco (DALR).
- *Adiabáticas húmidas.* Curvas adiabáticas húmidas, também denominadas de pseudo-adiabáticas são as curvas representadas a azul no diagrama que vão desde o fundo e vão gradualmente curvando para o topo direito tornando-se quase paralelas às curvas adiabáticas secas. Representam a taxa de variação para o processo pseudo-adiabático (SALR).

4.2 Curvas de Sondagens Meteorológicas

Para além dos cinco tipos de curvas definidos na secção anterior (ver Fig. 2(a)), há ainda mais dois tipos de curvas —denominadas curvas de sondagens meteorológicas— retratadas na Fig. 2(b) para resolver a equação de movimento associada à nuvem (i.e., Eq. (10)). A primeira destas curvas refere-se à temperatura ambiente T_a , enquanto que a segunda está relacionada com a temperatura de condensação T_d . Ambas são representadas através de aproximações lineares a vermelho e verde na Fig. 2(b), respectivamente, e são geradas especificamente a partir dos dados listados na terceira e quarta coluna da tabela apresentada na Fig. 1. A ideia principal do nosso método é usar estes parâmetros meteorológicos para ‘alimentar’ e explicitamente resolver a equação do movimento (10) de modo a simular e renderizar realisticamente a formação de nuvens.

5 FORMAÇÃO DE NUVENS

A formação de nuvens está relacionada com o movimento vertical de parcelas de ar que num determinado instante se convertem em nuvens. Mais concretamente, a formação de nuvens comporta dois passos distintos: geração e ascensão.

5.1 Geração

A geração de partículas inclui gestação e a libertação de partículas. Durante a fase de gestação, partículas são geradas para cada parcela de ar e à medida que o sol aquece o solo, as partículas vão absorvendo a radiação solar o que implica no aumento da sua temperatura. Assim que cada partícula atinge a temperatura convectiva T_c , libertam-se do solo (i.e., P_0) e inicia-se a segunda fase.

Algoritmo 1: Cálculo da temperatura convectiva T_c .

início

- Obter ponto de condensação (T_d, P) ao nível do solo;
- Calcular a linha de razão de mistura que passa no ponto de condensação determinado;
- Calcular o CCL como sendo o ponto de interseção entre a linha de razão de mistura e a curva da temperatura ambiente;
- Calcular a curva adiabática seca que passa no CCL;
- Calcular a temperatura convectiva T_c como sendo a interseção entre a curva adiabática seca e a pressão ao nível do chão.

Este processo é descrito no Algoritmo 1, que consiste nos seguintes cinco passos:

- *Passo 1.* O ponto de condensação (T_d, P_0) no solo é

o primeiro par de valores da curva da temperatura de condensação a ser calculado.

- *Passo 2.* A partir do ponto de condensação (T_d, P_0) , determina-se a linha de razão de mistura que passa nesse ponto (dada por (1) na Fig. 2(b)).
- *Passo 3.* O ponto CCL (T_{CCL}, P_{CCL}) é o ponto que resulta da intersecção entre a linha de razão de mistura (ver passo anterior) e um dos segmentos de reta da curva C_a relativa à temperatura ambiente.
- *Passo 4.* O cálculo da temperatura potencial no CCL é dado pela Eq. (9); mais especificamente, $\theta = T_{CCL} \left(\frac{P_0}{P_{CCL}} \right)^{R/c_p}$, onde (T_{CCL}, P_{CCL}) refere-se à temperatura e pressão no CCL.
- *Passo 5.* A temperatura potencial θ de uma partícula é constante ao longo da adiabatúca seca dada pela Eq. (9) e representada por (2) na Fig. 2(b), de modo que se pode usar o valor da temperatura potencial calculado anteriormente para calcular a temperatura convectiva T_c de cada partícula no ponto (T_0, P_0) , sendo $T_0 = T_c$. Portanto, $\theta = T_0 \left(\frac{P_0}{P_0} \right)^{R/c_p}$, ou seja, a temperatura convectiva é dada por $T_c = T_0 = \theta$.

5.2 Ascensão da Massa de Ar

Após atribuir uma temperatura convectiva a cada partícula da parcela de ar, tem lugar a ascensão das parcelas de ar, que se estabelece em duas fases: ascensão seca e ascensão húmida. A ascensão seca corresponde à ascensão das partículas entre o solo e o nível de condensação convectivo (CCL). Quando a temperatura da partícula fica

Algoritmo 2: EncontraCCLPartícula($\pi, \Pi, T_c, n, \Delta t$)

Dados: Π, T_a \triangleleft parcela de partículas Π , temperaturas ambiente T_a

Saída: θ \triangleleft temperaturas potenciais θ para pontos de T_a

Saída: Θ \triangleleft temperaturas potenciais Θ para partículas de Π

início

$k \leftarrow 0.286$ \triangleleft ver Eq. (8)

$P_0 \leftarrow$ pressão \triangleleft 1.ª coluna da tabela na Fig. 1

para cada $T_a[i]$ **faça**

$P \leftarrow$ pressão para $T_a[i]$ \triangleleft 1.ª coluna da Fig. 1

$T \leftarrow$ temperatura para $T_a[i]$ \triangleleft 3.ª coluna da Fig. 1

$\theta_i \leftarrow T \left(\frac{P_0}{P} \right)^k$ \triangleleft temperatura potencial, Eq. (9)

para cada $\Pi[j]$ **faça**

$\Theta_j \leftarrow T_c[j]$;

$[\theta_i, \theta_{i+1}] \leftarrow$ determina o intervalo;

$\Theta_j \in [\theta_i, \theta_{i+1}]$;

$(T, P)_j \leftarrow$ encontra o ponto CCL:

$\theta_p(T, P) = \Theta_j \in [\theta_i, \theta_{i+1}]$

igual à temperatura ambiente, diz-se que o CCL foi atingido, e as partículas de ar seco condensam em partículas de vapor de água. Portanto, necessitamos de calcular o

CCL para cada partícula π_i de uma parcela de ar π , conforme descrito no Algoritmo 2. Note-se que o CCL de uma partícula j é o ponto (T_j, P_j) que resulta da intersecção entre a sua curva adiabatúca seca (dada por (2) na Fig. 2(b)) e a curva C_a referente a curvas de sondagens de temperatura T_a (cf. 3.ª coluna da tabela na Fig. 1). Portanto, calculamos em primeiro lugar a temperatura potencial $\theta(T_j, P_j)$ em cada vértice de C_a , tal como ilustrado no primeiro ciclo do Algoritmo 2. Isto é equivalente a determinar a adiabatúca seca que passa em cada vértice de C_a , obtendo assim uma sequência $\theta_0, \dots, \theta_n$ de adiabatúcas secas (ou temperaturas potenciais) que cruzam C_a nos seus vértices.

Considerando que a temperatura convectiva T_{c_j} de uma partícula j coincide com a temperatura potencial Θ_j da sua adiabatúca seca, é necessário determinar o intervalo $[\theta_i, \theta_{i+1}]$ onde Θ_j se insira. Identificado o segmento $(i, i+1)$ de C_a que cruza a adiabatúca seca da partícula j , usamos um algoritmo de identificação de zeros (de análise numérica) para determinar o ponto de CCL (T_j, P_j) de modo que $\theta_p(T_j, P_j) = \Theta_j$ (cf. Eq. (9)), conforme pode ser observado no segundo ciclo ‘para’ do Algoritmo 2. Para resolver a Eq. (10) num ponto (T, P) localizado numa

Algoritmo 3: MovimentoAscensão()

início

para cada $\pi_i \in \Pi$ **faça**

para cada $p \in \pi_i$ **faça**

$C_0 \leftarrow$ curva da temperatura ambiente;

enquanto $P > CCL_P$ **faça**

$C_1 \leftarrow$ curva adiabatúca seca para a partícula;

$RP \leftarrow t_{min}, t_{max}$ na pressão P ;

$A \leftarrow (RP \cap C_0)$;

$B \leftarrow (RP \cap C_1)$;

$\theta_p \leftarrow$ DetermineTheta(t, A_P);

$\theta_a \leftarrow$ DetermineTheta(B_T, B_P);

$F \leftarrow 9.81 \cdot (\theta_p - \theta_a) / \theta_p$;

$v \leftarrow v_0 + F \cdot t$;

$\Delta z \leftarrow z_{init} + v_0 \cdot t + 1/2 \cdot F \cdot t^2$;

$z_1 \leftarrow z_0 + \Delta z$;

$x_1 \leftarrow x_0 + v_x$;

$y_1 \leftarrow y_0 + v_y$;

altitude z , é necessário conhecer dois valores, θ_p e θ_a . O valor da temperatura potencial θ_p de cada partícula é constante e conhecido do passo anterior e é igual à temperatura convectiva da partícula. Assim sendo, apenas é necessário determinar a temperatura potencial do ambiente θ_a (na curva de temperatura ambiente C_a) no mesmo nível de pressão P . Este cálculo consiste em determinar o ponto (θ_a, P) que resulta da intersecção entre a isobárica P e um segmento da curva C_a .

De seguida é necessário calcular a força de ascensão F para mover a partícula, bem como a sua velocidade v e o correspondente deslocamento vertical Δz que permite determinar a nova posição da partícula $z + \Delta z$. É também ne-



Figura 3. Nuvens *Cumulus* geradas para simular a variação diurna da temperatura no intervalo $T \in [25.63^\circ, 25.93^\circ]$. (a) A forma de torre é alcançada uma vez que há mais partículas a serem geradas a temperaturas mais baixas no intervalo. (b) Partículas são geradas com distribuição uniforme de temperaturas.

cessário calcular a evolução da partícula na direção (x, y) através da inclusão das componentes de vento, v_x e v_y , conforme se vê no Algoritmo 3.

O algoritmo para determinar o EL para cada partícula é similar ao Algoritmo 2, com a diferença que, em vez de determinar a temperatura para a curva adiabática seca, a temperatura é determinada para a curva adiabática húmida (dada por (3) na Fig. 2(b)) que passa pelo CCL.

6 RESULTADOS

Nesta secção são apresentados resultados referentes à simulação de diferentes nuvens, bem como resultados de desempenho para essas simulações.

6.1 Renderização de Nuvens

De forma a gerar imagens de nuvens realísticas, é necessário considerar a natureza complexa da sua interação com a luz. A luz numa nuvem é espalhada muitas vezes pelas pequenas gotículas que a compõem. É este efeito que lhes confere uma aparência suave e difusa. Neste sentido, foi implementado um ambiente 3D, em C++, com bibliotecas de OpenGL/GLUT onde se usou a técnica de renderização proposta por [Harris 01] que descreve um algoritmo para a renderização em tempo real de nuvens com base num processo de iluminação a dois passos, similar à apresentada em [Dobashi 00].

6.2 Forma das Nuvens

São vários os fatores que influenciam a forma das nuvens: dados da sondagem atmosférica, ventos horizontais e temperatura do solo no momento da geração da nuvem.

Quando vento horizontal incide numa nuvem, o seu efeito é mais óbvio no topo do que na base [Emanuel 94]. A Fig. 4 mostra o efeito final do vento numa nuvem (composta por 5000 partículas), aplicado desde a sua geração ao nível do solo até ao EL. O perfil de vento da Fig. 4(b) foi definido a partir dos dados da sondagem.

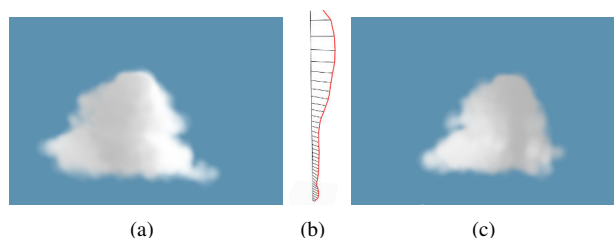


Figura 4. Evolução de uma nuvem em diferentes condições de vento. (a) nuvem gerada sem vento, (b) exemplo de um perfil de vento, (c) curva gerada com o perfil de vento definido em (b).

Dados atmosféricos diferentes produzem diferentes formas de nuvens, uma vez que os níveis de pressão onde o CCL e o EL são atingidos variam entre sondagens atmosféricas. Isto, combinado com o perfil de temperatura ambiente, determina a forma como as partículas evoluem na atmosfera. Na Fig. 5 são apresentadas duas nuvens geradas com base em duas sondagens diferentes. Ao nível do solo, a temperatura aumenta à mesma taxa. Contudo, a forma da nuvem é diferente devido a alterações nos perfis da temperatura ambiente e da temperatura da partícula.

Outro fator que influencia a forma de nuvens é o perfil de temperatura que as partículas seguem dentro da nuvem. Dependendo da temperatura ao nível do solo, as partículas dentro da nuvem atingem o equilíbrio a altitudes maiores ou menores na atmosfera (Fig. 3).

6.3 Desempenho

Foram realizados vários testes à eficiência do nosso algoritmo na simulação e renderização de nuvens e cenas de nuvens com complexidade crescente. Os primeiros testes focaram na simulação e renderização de uma nuvem com

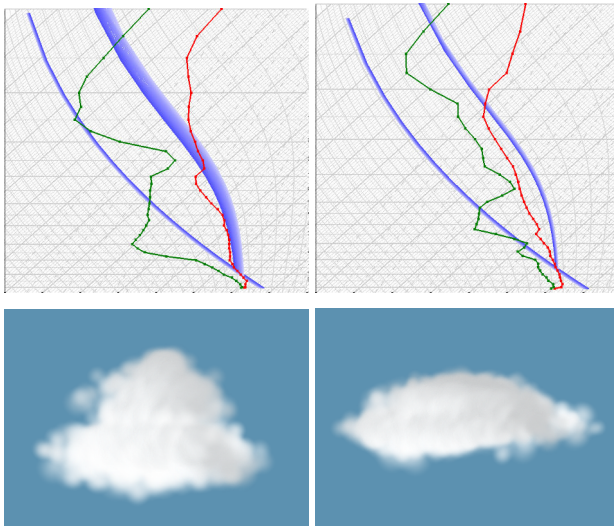


Figura 5. Efeito da utilização de diferentes sondagens atmosféricas na forma da nuvem.

variação do número de partículas entre as 200 e 3000.

Para cada passo de simulação, é aplicado o algoritmo de renderização que consome mais tempo do que o passo de simulação, conforme se apresenta na Fig. 7(a). Além disso, à medida que o número de partículas aumenta, o desempenho temporal de simulação tende para a linearidade. Na fase de renderização, e tendo em consideração que não estamos a usar aceleração CPU/GPU, os resultados apresentados na Fig. 7(b) indicam que para cenas compostas de centenas de milhares de partículas, conseguimos atingir valores interessantes em *frames* por segundo.

Nos segundos testes, foram renderizadas várias nuvens anteriormente calculadas usando diagramas SkewT/LogP. O número de nuvens varia entre 3 e 120 sendo que o número de partículas em cada nuvem varia entre 200 e 3000. Para além disso, foram geradas cenas contendo até 150000 partículas, tal como mostrado na Fig. 6.

7 CONCLUSÕES

Neste artigo demonstrámos que é possível usar diagramas SkewT/LogP na visualização realística de nuvens convectivas, em tempo real. Estes diagramas permitem-nos determinar os parâmetros necessários para a advecção de uma massa de ar usando a equação do movimento (cf. Eq. (10)), portanto, a nossa abordagem procedural não envolve a complexidade associada à resolução de equações diferenciais. A eficiência do algoritmo permite-nos simular nuvens até 1000 partículas em 5 milissegundos. Entre 10000 e 20000 partículas é um valor adequado para modelar uma cena de nuvens onde o utilizador pode navegar livremente em tempo real. Mesmo com 50000 partículas, o número de *frames* é ainda da ordem de 24 FPS.

Nesta fase apenas nuvens *cumulus* que resultam da convecção são bem descritas pelo nosso modelo. O nosso objetivo é simular outros tipos de nuvens, como por



(a)



(b)

Figura 6. Vista de um sistema de nuvens composto de 150000 partículas.

exemplo *stratus* ou *cirrus*. Estas nuvens são não conectivas, contudo podem ser determinadas num diagrama SkewT/LogP.

Referências

- [AWS 79] AWS. The use of the skew t, log p diagram in analysis and forecasting. 1979.
- [Bouthors 04] Antoine Bouthors e Fabrice Neyret. Modeling clouds shape. Em *Eurographics (short papers)*, Grenoble France, 2004.
- [Cengel 11] Yunus A Cengel, Michael A Boles, e Mehmet Kanoğlu. *Thermodynamics: an engineering approach*, volume 5. McGraw-Hill New York, 2011.
- [Crowell 00] Benjamin Crowell. *Newtonian physics*, volume 1. Light and Matter, 2000.
- [Dobashi 00] Yoshinori Dobashi, Kazufumi Kaneda, Hideo Yamashita, Tsuyoshi Okita, e Tomoyuki Nishita. A simple, efficient method for realistic animation of clouds. Em *Proc. of the 27th annual conference on Computer graphics and interactive techniques*, páginas 19–28, 2000.
- [Elinas 00] Pantelis Elinas e Wolfgang Stürzlinger. Real-time rendering of 3d clouds. *J. Graph. Tools*, 5:33–45, 2000.

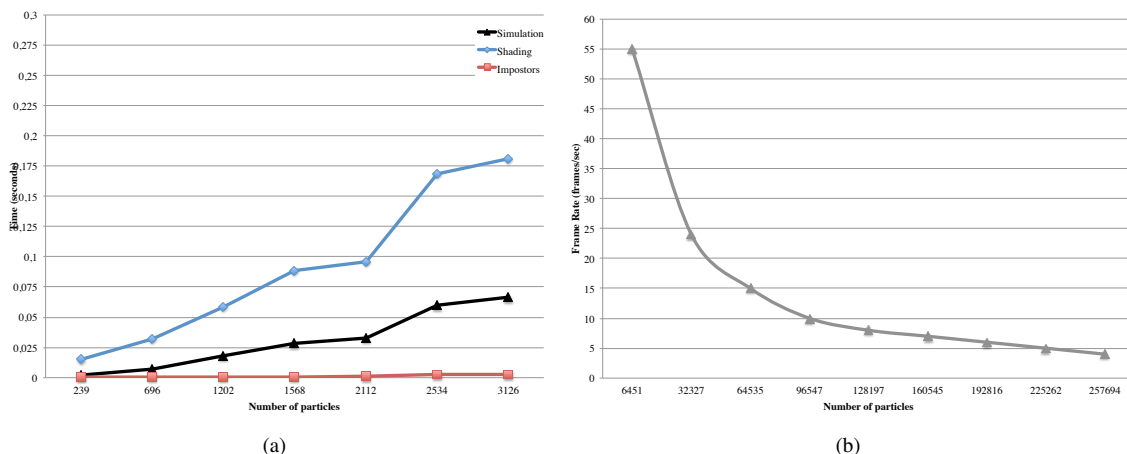


Figura 7. Resultados obtidos. (a) tempo de simulação e render de uma nuvem (b) número de frames necessárias para renderizar uma cena de nuvens.

- [Emanuel 94] Kerry A Emanuel. *Atmospheric convection*. Oxford University Press, 1994.
- [Gardner 85] Geoffrey Y. Gardner. Visual simulation of clouds. *ACM SIGGRAPH Comput. Graph.*, 19:297–304, July 1985.
- [Grudzinski 07] Jakub Grudzinski e Adrian Debowski. Clouds and atmospheric phenomena simulation in real-time 3d graphics. volume 4418 de *Lecture Notes in Computer Science*, páginas 117–127. 2007.
- [Harris 01] Mark J Harris e Anselmo Lastra. Real-time cloud rendering. Em *Computer Graphics Forum*, volume 20, páginas 76–85, 2001.
- [Harris 03] Mark J. Harris, William V. Baxter, Thorsten Scheuermann, e Anselmo Lastra. Simulation of cloud dynamics on graphics hardware. Em *Proc. of the ACM conference on Graphics hardware*, páginas 92–101, 2003.
- [Man 06] P. Man. Generating and real-time rendering of clouds. Em *Central European seminar on computer graphics*, páginas 1–9, 2006.
- [Miyazaki 01] Ryo Miyazaki, Satoru Yoshida, Tomoyuki Nishita, e Yoshinori Dobashi. A method for modeling clouds based on atmospheric fluid dynamics. Em *Proc. of the 9th Pacific Conference on Computer Graphics and Applications*, 2001.
- [Neyret 97] Fabrice Neyret. Qualitative simulation of convective cloud formation and evolution. Em *Proc of Eurographics Computer Animation and Simulation Workshop'97*, páginas 113–124, 1997.
- [Perlin 85] Ken Perlin. An image synthesizer. *Proceedings of SIGGRAPH*, 19(3):287–296, 1985.
- [Rogers 89] RR Rogers e MK Yau. A short course in cloud physics, international series in natural philosophy, 1989.
- [Schpok 03] Joshua Schpok, Joseph Simons, David S. Ebert, e Charles Hansen. A real-time cloud modeling, rendering, and animation system. Em *Proc. of the symposium on Computer animation*, páginas 160–166, 2003.
- [Stam 99] Jos Stam. Stable fluids. Em *Proc. of the 26th conference on Computer graphics and interactive techniques*, páginas 121–128, 1999.
- [Trembilski 02] Andrzej Trembilski, Andreas Broßler, e Abt Visualisierung Und Virtuelle. Surface-based efficient cloud visualisation for animation applications. Em *Journal of WSCG*, 2002.
- [Voss 85] R.F. Voss. Random fractal forgeries. *Fundamental algorithms for computer graphics*, 17:805–835, 1985.
- [Wenke 12] Wang Wenke, Li Sikun, Guo Yumeng, e Xiong Min. Automatic generation of large scale 3d cloud based on weather forecast data. Em *2012 International Conference on Virtual Reality and Visualization*, páginas 69–73, 2012.
- [Wither 08] Jamie Wither, Antoine Bouthors, e Marie-Paule Cani. Rapid sketch modeling of clouds. Em *Eurographics Workshop on Sketch-Based Interfaces and Modeling (SBIM)*, 2008.

Sessão - Modelação e Reconstrução

Geração Procedimental de Traçados Rodoviários para Simulação de Condução

Carlos Campos
ISEP/FEUP
Porto, Portugal
crc@isep.ipp.pt

João Miguel Leitão
ISEP
Porto, Portugal
jml@isep.ipp.pt

António Fernando Coelho
FEUP
Porto, Portugal
acoelho@fe.up.pt

Resumo

Os ambientes virtuais para simulação de condução para fins científicos requerem modelos tridimensionais de estradas realistas. A geração de modelos de estradas com este nível de qualidade requer a prévia definição dos traçados rodoviários. Dependendo dos requisitos da experiência, o traçado rodoviário de uma estrada pode ser obtido recorrendo a especialistas da área de projeto de vias comunicação. A conceção do traçado de uma via pode tornar-se uma tarefa morosa e dispendiosa, mesmo recorrendo a ferramentas CAD.

Na preparação de experiências científicas em simulador, o processo de obtenção de um traçado pode ser otimizado recorrendo a métodos de geração automática. Este artigo apresenta um método que permite gerar de forma automática traçados rodoviários adequados para implementação de experiências de simulação de condução em diversas áreas científicas, como ergonomia, psicologia e engenharia de tráfego. O processo de geração de traçados foi inspirado nos métodos utilizados em engenharia de vias, produzindo traçados segundo as normas e estradas semelhantes às encontradas no mundo real. Os traçados rodoviários gerados proceduralmente são adequados para realização de experiências de simulação de condução para fins científicos.

Com a implementação deste método é reduzido significativamente o recurso a especialistas de vias de comunicação para elaboração do traçado rodoviário adequando para a realização de experiências científicas em simulador de condução.

Palavras-Chave

Modelação Procedimental, Traçado Rodoviário, Projeto de vias, Simulação de Condução.

1. INTRODUÇÃO

A modelação procedimental de ambientes rodoviários realistas é uma área de investigação de grande interesse, que se dedica à criação de modelos não só para o entretenimento mas também para aplicações científicas. Neste artigo apresenta-se o estado atual da implementação de um método para a criação automática de traçados rodoviários para simulação de condução para fins científicos.

As experiências de simulação de condução para fins científicos requerem a criação de traçados rodoviários extensos e com características específicas. Uma forma de obter traçados realistas, é recorrer a especialistas da área de projeto de vias para produzirem traçados rodoviários que cumpram os elevados requisitos de realismo exigidos pela especificação da experiência. A conceção do projeto de uma via que cumpra os requisitos exigidos pode tornar-se uma tarefa morosa e dispendiosa, mesmo possuindo ferramentas de auxílio como o civil3D¹. Após obter os dados de projeto de um traçado rodoviário, é ainda necessário que o preparador da experiência despenda recursos para proceder à geração do modelo visual. Uma forma de oti-

mizar o processo de geração do modelo da estrada a partir do traçado é utilizar ferramentas de modelação automática que facilitem essa tarefa, como as apresentadas em [Campos07]. Adicionalmente ainda é necessário proceder à edição do modelo de terreno em função do traçado da estrada.

Para além dos elevados requisitos de realismo do traçado rodoviário, a simulação de condução requer também uma descrição semântica de todo o ambiente virtual gerado, como também referido em [Thomas00]. Esta descrição é fundamental em sistemas de simulação para colocação de atores (e.g. peões, outros veículos), monitorização de toda a simulação e produção de relatórios. Uma característica particular da simulação de condução para fins científicos é a produção de relatórios. Estes relatórios são fundamentais para uma posterior análise e tratamento de dados relativos ao ensaio experimental.

1.1 Organização do artigo

O restante artigo está organizado da seguinte forma: na secção 2, é apresentado o estado da arte na conceção de vias de comunicação e são apresentados os trabalhos relacionados mais significativos. O método concebido é apresentado na secção 3, e na secção 4 é descrita a im-

¹ Autocad civil3D – www.autodesk.pt

plementação de um protótipo funcional. Os resultados obtidos são apresentados na secção 5, e na secção 6, são apresentadas as principais conclusões e trabalho futuro.

2. ESTADO DA ARTE

2.1 Conceção de vias de comunicação

Como descrito em [Campos12], uma rede rodoviária pode ser definida por um conjunto de nós e ligações que os interligam. As redes rodoviárias são planeadas para ligar pontos de interesse, permitindo a deslocação de veículos entre os diversos pontos.

A conceção de uma via parte inicialmente de uma decisão estratégica de interligar dois pontos na rede rodoviária existente. Essencialmente, esta decisão é tomada tendo em consideração parâmetros como: crescimento demográfico, nível de serviço, estudos de tráfego e custo estimado de construção. Em engenharia de vias, um projeto de uma estrada desenvolve-se de acordo com as seguintes fases: programa preliminar, estudo prévio e projeto de execução [França11]. O programa preliminar é basicamente um caderno de encargos, constituído por disposições gerais, com pouca pormenorização, onde figuram as características gerais pretendidas e que condicionaram logo à partida a escolha do traçado. O estudo prévio não tem como função fornecer os elementos definitivos, nem a pormenorização necessária à execução da obra. O objetivo principal é o de apontar uma ou várias soluções alternativas para o traçado. Finalizado e aprovado o estudo prévio em alinhamentos retos, segue-se o projeto de execução. O traçado está aproximadamente definido e o trabalho consiste agora no detalhe do estudo. O projeto de execução deve conter a pormenorização necessária à implantação da via rodoviária. No traçado geométrico da via, começa-se por especificar a diretriz (traçado em planta), com a introdução de curvas compostas. Segue-se o traçado em perfil longitudinal (altimetria), onde são definidas as concordâncias verticais. Por último é especificado o perfil transversal da via. A definição do traçado deverá obedecer aos critérios estabelecidos nas normas aplicáveis. Em Portugal essas normas são definidas pela entidade Estradas de Portugal, que é responsável pela tutela da rede rodoviária [EP94].

2.2 Trabalhos relacionados

Vários trabalhos têm sido desenvolvidos para criação de ambientes rodoviários de cidade, focando-se essencialmente em traçados urbanos [Parish01] [Chen08] [Teoh08] [Coelho07]. Os resultados obtidos por estas metodologias são de boa qualidade, no entanto o problema destas abordagens é que a obtenção de modelos realistas implica um controlo demasiado elevado por parte do utilizador. Experiências de simulação de condução tipicamente requerem ambientes rodoviários mistos, com estradas corretamente modeladas e que combinem traçado em meio urbano e traçado em meio rural.

Uma proposta para a modelação procedimental de um ambiente virtual e adequado para simulação em tempo real é apresentada por Smelik et al. em 2008 [Smelik08]. O autor descreve um processo em que o ambiente virtual

é criado de forma hierárquica em camadas, a partir de um esboço inicial, incluindo o tratamento da rede rodoviária. Na base do modelo hierárquico encontra-se a definição do modelo do terreno, seguindo-se o modelo hidrográfico, o modelo da vegetação, o mapa rodoviário e por último o modelo urbano (edifícios). Uma vantagem desta hierarquização é que cada camada pode ser tratada individualmente. No processo de geração do modelo final é tida em consideração a adequação do modelo de terreno às restantes instanciações de objetos 3D, como abordado em [Latham06]. Na proposta apresentada por Smelik et al. não se percebe qual a metodologia utilizada para geração do traçado rodoviário. Smelik et al. em 2011 apresentam uma abordagem de modelação de um ambiente virtual, recorrendo a um modelador interativo [Smelik11]. A geração do ambiente virtual obriga a um controlo demasiado elevado por parte do utilizador, fazendo com que o processo deixe de ser totalmente automático. Para a geração do modelo geométrico rodoviário são exploradas as técnicas de modelação procedimental de redes urbanas apresentadas por Kelly et al. em 2008 [Kelly08].

Galin et al. apresentam uma metodologia de geração procedimental de traçados de estradas [Galin10]. A geração do traçado é realizada recorrendo a um algoritmo complexo, sobre um terreno definido em mapa de elevação. Em cada etapa do processo de geração do traçado da estrada sobre o modelo do terreno, a direção a seguir é determinada em função da avaliação local do custo de cada alternativa. O custo de cada alternativa é determinado recorrendo a uma função que avalia vários indicadores, como a relação com o terreno e a relação com as condicionantes. O facto do uso de uma avaliação local impede a correta avaliação e otimização global da solução final. O uso de uma abordagem que analise o custo por níveis de forma global conduz a uma melhor otimização dos traçados finais, produzindo modelos de qualidade. Dependendo do modelo de terreno e do tipo de estrada a gerar, determinar o número de alternativas a estudar pode-se tornar num problema demasiado complexo. Pela descrição do autor não se conclui que esta metodologia produza traçados rodoviários semelhantes aos encontrados em estradas reais, não sendo apresentada uma avaliação relativa ao realismo dos modelos obtidos. Uma metodologia de geração de uma rede rodoviária que integra diferentes tipologias de estradas (autoestradas, estradas principais e estradas secundárias) é apresentada por Galin et al. em [Galin11]. Cada tipologia de estradas é criada de forma independente e no final é realizada a sobreposição dos diferentes tipos de estradas numa única rede rodoviária. A forma como é tratado este problema resulta em que o traçado das estradas tenha que ser recalculado. Na geração do modelo visual são instanciados modelos 3D, como pontes, túneis e cruzamento.

Thomas et al. apresentam uma cidade virtual em que é possível encontrar diferentes atores, como: peões, veículos a circular nos diferentes sentidos, transportes públicos e veículos conduzidos por outros condutores, num complexo ambiente rodoviário urbano [Thomas00]. O ambi-

ente urbano é criado recorrendo ao modelador interativo VUEMS sobre uma definição do terreno em mapa de elevação [Donikian97]. O uso de um modelador interativo não é adequado à geração de um ambiente rodoviário extenso de forma automática e eficiente.

Nos trabalhos apresentados por Bayarri et al., são descritas metodologias para a geração de ambientes rodoviários adequados para simulação de condução a partir de dados de projeto [Bayarri96] [Pareja99]. Cada vez que se pretenda obter um modelo de uma via exige a prévia interação com a área de engenharia de vias, ou outras fontes, para obter a definição do traçado da via. Algumas destas metodologias já foram exploradas em trabalhos anteriores [Campos07].

Para a implementação de experiências de simulação de condução, além da geração do traçado rodoviário e modelação do ambiente virtual, deverá ser criada a correspondente definição semântica. Como referido em [Thomas00], a definição semântica é uma descrição de alto nível, de caracterização do ambiente rodoviário, que permite suportar o modelo dinâmico do ambiente virtual. A criação de uma descrição semântica de acordo com o mapa rodoviário é também abordada em outros trabalhos relacionados [Campos07] [Bayarri96] [Pareja99].

O projeto OpenDRIVE desenvolveu um *standard* para a completa descrição de uma rede viária, de modo a ser compatível com diversos sistemas de simulação. É referido que o *standard* proposto permite a parametrização das principais características de estradas reais. Um modelador interativo que usa este *standard* é o Road Designer [OpenDrive10]. Este modelador além de gerar o modelo visual gera uma descrição do ambiente rodoviário neste formato. No entanto não são conhecidas ferramentas de geração automática de modelos neste formato.

A criação de modelos de estradas realistas para simulação de condução passa muitas vezes por processos manuais, trabalhosos e morosos, mesmo quando são disponibilizadas ferramentas interativas. Sempre que se pretende criar um ambiente de traçado realista para tarefas de condução para fins científicos é necessário um elevado nível de interação com o preparador do modelo, que muitas vezes passa por recorrer a especialistas na área de projeto de vias para obtenção do traçado. Uma alternativa a este problema passa por conceber um método de geração automática de traçados rodoviários.

3. GERAÇÃO DE ESTRADAS

O processo concebido para a geração de um traçado tem como base um modelo hierárquico em níveis. Partindo da definição de uma ligação entre dois pontos, começa-se por gerar um conjunto de soluções de alto nível, com pouca pormenorização. A escolha da melhor hipótese em cada etapa do processo é realizada recorrendo a uma função de custo que avalia os vários parâmetros, como relação com o relevo do terreno e respetivas condicionantes. À medida que o processo de geração de uma estrada avança, o traçado rodoviário vai sendo refinado, até se obter uma definição com nível de detalhe adequado em

alinhamentos retos. A partir da definição em alinhamentos retos é desenvolvido o traçado em planta, com a introdução de curvas compostas. Após se obter o traçado em planta, segue-se o traçado em altimetria com a introdução de concordâncias verticais.

3.1 Geração de traçados rodoviários

O problema da geração do traçado rodoviário pode ser formulado da seguinte forma: Um traçado rodoviário T é constituído por um conjunto ordenado de vértices V e segmentos de traçado S (4).

$$T = (V, S) \quad (4)$$

Um segmento de traçado S_i é constituído por um conjunto de retas, arcos e clotoides (5).

$$S_i = \{(\text{reta}, \text{arco}, \text{clotóide})\} \quad (5)$$

Considerando uma área A e dois pontos designado por a e b , P é o conjunto de soluções de traçado T em A que interligam os pontos a e b (6).

$$P = \{T\} \quad (6)$$

O objetivo consiste em determinar um traçado T' entre os pontos a e b que minimize a função de custo $fc(I)$, que avalia a influência dos diferentes custos I , como a relação com o relevo do terreno e com as condicionantes. A função $fc(I)$ permite determinar o custo de um traçado $C(T)$ (7).

$$C(T') = \min C(T), \text{ em que } T \in P \quad (7)$$

Para a geração de um traçado rodoviário, a partir da definição de uma ligação entre dois pontos, são geradas definições de traçado com um nível de detalhe crescente. Em cada nível de detalhe são geradas várias alternativas de traçado, e em função da avaliação feita é selecionada uma, avançando-se para o nível de detalhe seguinte (figura 1).

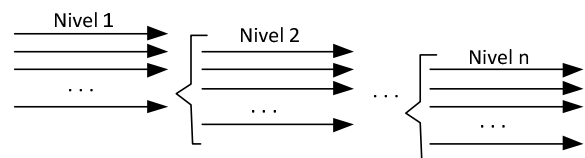


Figura 1. Geração de traçado por níveis de detalhe

A utilização do nível de detalhe adequado permite gerar e estudar um qualquer número de propostas sem prejudicar o processo em termos de peso computacional. O uso de muitos níveis de detalhe dificulta que a solução escolhida seja a ideal, porque em cada nível a avaliação com o relevo do terreno e respetivas condicionantes é feita com pouco detalhe, sendo que as decisões tomadas nos níveis anteriores condicionam as escolhas nos níveis seguintes. A partir do momento em que a localização de um vértice do traçado é determinada, esta não volta a ser processada. Por outro lado, poucos níveis implicam maior peso computacional a processar a relação com o relevo do terreno e condicionantes, ao analisar com maior detalhe cada alternativa de traçado. A estratégia da geração do traçado por níveis de detalhe corresponde ao método utilizado em engenharia de vias para conceção de um traçado rodoviário (secção 2.1), o que facilita a parametrização do número de níveis adequado.

Uma característica inovadora deste método é o facto de a obtenção do traçado final de uma estrada ser realizado de forma hierárquica em níveis, permitindo a partir da definição de uma ligação entre dois pontos, caminhar no sentido de obter a solução ótima de traçado, considerando a definição do modelo do terreno e respetivas condicionantes. A definição de traçado resultante contém a pormenorização necessária à geração do modelo virtual.

3.2 Avaliação de traçado

Na geração procedimental de um traçado rodoviário são realizadas avaliações em diferentes níveis. Ao nível da escolha do vértice do corredor é analisado a sua localização (ver secção 4.2), de forma a posteriormente garantir o desenvolvimento do traçado em planta, com a implantação de curvas que cumpram as normas de projeto. Ao nível do corredor, a escolha da melhor alternativa é realizada recorrendo a uma função de custo que avalia os diversos parâmetros considerados. Após obter o traçado em planta é realizada uma avaliação às características de sinuosidade da estrada.

3.2.1 Avaliação de alternativas de traçado

Os custos utilizados nesta abordagem para a avaliação de alternativas de traçado de uma estrada refletem a influência do relevo do terreno e respetivas condicionantes. Para avaliar as diferentes alternativas de traçado, calcula-se o custo total de uma alternativa de traçado, como sendo o soma pesada de todos os custos parciais. Os custos parciais considerados na avaliação de um traçado são: zona classificada, construção em ponte, construção em túnel, movimento de terras, construção da via e custo de utilização. Uma zona classificada é definida como uma área geográfica em planta que representa uma determinada condicionante, como por exemplo água, floresta ou edificações. Para determinar o impacto do traçado de uma via sobre uma zona classificada é necessário proceder ao cálculo da área da zona classificada afetada pela passagem da estrada. Para determinar se uma porção de extensão da estrada é construída em ponte ou túnel é feita uma análise à diferença entre a cota do terreno e a cota da estrada. Ao longo do traçado da via é considerada construção em ponte se a diferença entre a cota da estrada e a cota do terreno for superior ao valor de referência. Também é considerada construção em ponte sempre que a estrada atravessa água, por exemplo: rio ou lago. É considerada construção em túnel sempre que a altura entre a cota do terreno e a cota da estrada for superior aos valores de referência, na combinação entre altura e extensão. A extensão da via considerada em construção diretamente sobre o terreno é obtida retirando à extensão total, a extensão de construção em ponte e em túnel. Para efetuar o cálculo do volume de terras é necessário analisar a diferença (h) entre a cota do terreno e a cota da estrada. O volume de aterro é calculado quando se verificam simultaneamente as seguintes condições: $0 < |h| < h_{\text{refill}}$ e a estrada não atravessa água (h_{refill} corresponde à altura (h) de referência para aterro). O volume de escavação é calculado quando se verificam simultaneamente as seguintes condições: $0 < |h| < h_{\text{refexc}}$ (h_{refexc} corresponde à altura

(h) de referência para escavação). Nas extensões de estrada ocupadas por ponte ou túnel não é calculado movimento de terras, pois têm custos específicos. Na avaliação de traçados é calculado um custo, designado por custo de utilização, que está relacionado com a variação em altimetria do traçado da via (Δz) e com o índice de utilização. Considerando que cada ligação a tratar não tem nós de interligação, o índice de utilização (U) é calculado em função da extensão total da via e da população dos nós extremos (8).

$$U = PA * PB / L^2 \quad (8)$$

O índice de utilização (U) reflete a circulação de veículos (pessoas), nos dois sentidos, entre os dois nós (cidades). Os valores de referência para os custos analisados no estudo de alternativas de traçado foram determinados recorrendo a especialistas da área de projeto de vias de comunicação.

3.2.2 Avaliação das características de traçado

Os traçados rodoviários gerados pelo método proposto devem apresentar características geométricas semelhantes às encontradas em estradas reais. De forma a validar o método de geração de traçados recorreu-se aos seguintes indicadores: índice de sinuosidade em planta (I_s), análise no domínio da curvatura recorrendo à transformada discreta de Fourier (Discret Fourier Transform, DFT) e índice de curvatura (I_c). O índice de sinuosidade em planta é calculado através da divisão do valor acumulando de todas as variações de direção (α_i) da estrada pela extensão total sobre o eixo da via L (9).

$$I_s = \sum \alpha_i / L \quad (9)$$

Os valores apresentados na tabela 1, foram determinados analisando traçados reais do tipo autoestrada (A25).

Tabela 1. Valores de referência de indicadores

Indicador	Valor de referência	Desv. padrão(σ)
Índice de sinuosidade	737,40	72,02
Índice de curvatura	0,96	0,4

A transformada discreta de Fourier permite analisar a sinuosidade de uma estrada pela decomposição do traçado em séries sinusoidais. A transformada discreta de Fourier, quando aplicada a um conjunto de amostras no domínio dos tempos, resulta num conjunto de amplitudes de componentes sinusoidais no domínio das frequências [Mitra05] [Ingle12]. Neste caso quando aplicada a amostras no domínio do comprimento (m), resulta numa série de coeficientes no domínio da curvatura (m^{-1}). A transformada rápida de Fourier é amplamente usada em diferentes áreas científicas como por exemplo: engenharia, ciências e matemática. Do trabalho de investigação realizado anteriormente, conclui-se que os coeficientes da transformada discreta de Fourier para traçados do tipo autoestrada deixam de ter significado a partir do elemento 10. O índice de curvatura (I_c) é determinado pela relação entre a extensão da estrada em curva e a sua extensão total (10).

$$I_c = \text{Extensão em curva} / \text{Extensão da via} \quad (10)$$

Este indicador permite aferir qual porção do traçado da via que é descrito em curva face à extensão total. Tipica-

mente em traçados do tipo autoestrada, este índice tende a ter um valor próximo da unidade.

4. IMPLEMENTAÇÃO

Na implementação do protótipo, começou-se por desenvolver o traçado em planta, desde a escolha dos vértices

Tabela 2. Parâmetros utilizados para definição dos corredores (km)

Nível	$L_{min} >$	$L_{max} \leq$	Corredor inicial (largura)	$L_{paragem} <$	Corredor gerado (largura)
1	$0,02 \cdot V_b (i)$	10	5	$0,02 \cdot V_b (i)$	1
2	10	50	20	10	5
3	50	200	100	50	20
4	200	1000	500	200	100
5	1000	5000	2000	1000	500

(i) V_b , velocidade base de projeto em km/h. Por exemplo, $V_b=120$ km/h, $L_{paragem} = L_{min} = 2400$ m.

4.1 Geração de corredores

O processo de geração de um traçado é realizado hierarquicamente em níveis, aumentando o detalhe à medida que o processo avança para os níveis seguintes. Em cada nível são geradas diferentes alternativas de traçado. Cada alternativa representa um corredor com uma determinada largura. O nível em que o processo se inicia depende da extensão inicial da ligação entre os dois pontos. Segundo a tabela 2, é possível determinar em que nível o processo se inicia e quantas iterações serão realizadas até obter o traçado rodoviário em alinhamentos retos. Em engenharia de vias, a geração de um traçado rodoviário em alinhamentos retos também é realizada em níveis. A construção da tabela 2 foi inspirada nos métodos utilizados em engenharia de vias para conceção de um projeto de traçado rodoviário. Na tabela 2 os parâmetros L_{min} e L_{max} correspondem aos limites mínimos e máximos da extensão inicial entre dois pontos, que permitem determinar em que nível se inicia o processo de geração de corredores. A geração de um novo corredor, com largura definida pelo parâmetro *Corredor gerado*, é realizada dentro de um corredor com largura definida pelo parâmetro *Corredor inicial*. Essa geração termina quando já não existir alinhamentos com extensão superior à condição de paragem definida pelo parâmetro $L_{paragem}$. Como exemplo: uma ligação inicial entre dois pontos com extensão de 150km, pela tabela 2, o processo de geração do traçado inicia-se no nível 3 ($50 < 150 \leq 200$). A partir de um corredor inicial com 100 km de largura é gerado um novo corredor com 20 km de largura.

4.2 Geração de vértices do corredor

O processo de geração de um novo corredor consiste na divisão binária sucessiva da ligação inicial entre dois pontos, adicionando-se novos vértices à definição do traçado. A figura 2 exemplifica a obtenção de um novo vértice, na geração de corredores.

Entre os vértices A e B da figura 2, é determinado o ponto médio (m), obtendo-se o vetor \overrightarrow{Am} . É introduzida uma perturbação no módulo do vetor \overrightarrow{Am} de forma a que os novos alinhamentos gerados tenham diferentes extensões. Do resultado dessa perturbação obtém-se o vetor \overrightarrow{AP} . De seguida calcula-se o vetor tensor $\overrightarrow{PP'}$, que é perpendicular

do corredor até à obtenção do traçado em alinhamentos retos. Para a escolha de corredores desenvolveu-se o módulo de avaliação de alternativas. De seguida passou-se ao desenvolvimento do traçado em altimetria com a introdução de concordâncias verticais.

ao vetor \overrightarrow{Am} , permitindo deslocar o novo vértice para o ponto P'. No cálculo do novo vértice P' é considerada a largura do corredor, para que o novo vértice não se desloque para além dos limites do corredor.

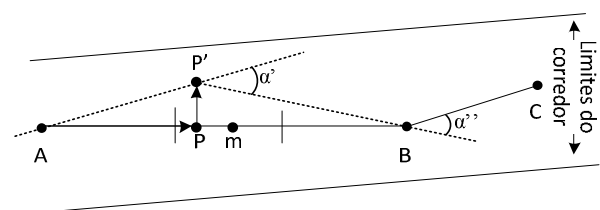


Figura 2. Vértice do corredor

Na determinação do novo vértice P' também é considerado o ângulo formado pelos alinhamentos adjacentes, α' e α'' da figura 2, impondo um limite máximo de deslocamento para o vértice P'. Dependendo do valor do ângulo dos alinhamentos adjacentes, este limite pode ser inferior aos limites do corredor. Esta consideração permite garantir que posteriormente é possível desenvolver o traçado em planta de acordo com as normas, cumprindo as curvaturas mínimas de traçado [EP94].

A cota dos novos vértices é determinada depois de conhecida a suas localizações em planta. Considerando que a cota dos pontos extremos foi previamente definida pelo preparador do traçado ou por instâncias superiores no processo de geração de corredores, estas não são alteradas. No cálculo da cota do vértice é tida em consideração a inclinação dos segmentos, de forma que esta inclinação se enquadre nos limites impostos pelas normas para o tipo de via em estudo.

4.3 Desenvolvimento do traçado em planta

O traçado em planta tem como principal objetivo a introdução de curvas entre os alinhamentos retos, resultantes do processo iterativo de geração de traçado. É nesta fase que se define o traçado final da via rodoviária, com o pormenor necessário à geração do modelo visual. O traçado em planta vai definir o eixo da via de uma estrada por uma sequência de retas, arcos e clotoides. Uma clotóide, também conhecida por curva de cornú, tem como função manter a continuidade da 1.ª derivada no traçado de uma via rodoviária, de forma a permitir a transição gradual da variação da aceleração centrífuga entre o troço reto e a curva circular. Esta sequência em termos geomé-

tricos resulta numa curva composta, como se ilustra na figura 3.

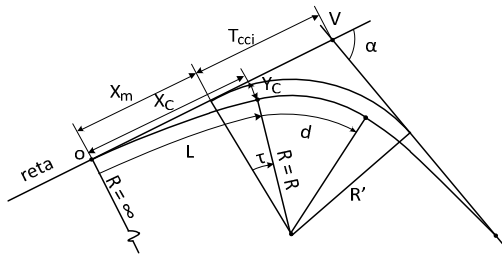


Figura 3. Curva composta simétrica

Dependo do tipo de via, existem parâmetros a nível geométrico que constam das normas de traçado que devem ser respeitados, como por exemplo: parâmetro da clotoide e raio mínimo em planta. Os valores mínimos admissíveis para estes parâmetros estão relacionados com a velocidade base do projeto da via. Conhecido o raio (R) do arco da curva circular, o parâmetro (A) da clotoide e o ângulo (α) entre os dois alinhamentos retos é possível determinar os restantes elementos. Os elementos em falta são: a extensão (L) de desenvolvimento da clotoide, o ângulo da tangente no ponto final da clotoide (tau), o valor do desvio do raio devido à introdução da clotoide, e o ponto de osculação (O) situado no fim do alinhamento reto e início da clotoide. Para determinar o ponto de osculação ainda é necessário determinar os parâmetros Tcci e Xm da figura 3. A distância Tcci é calculada pela seguinte equação:

$$Tcci = (R + \Delta R) * \tan(\alpha/2) \quad (11)$$

onde, R é o raio da curva circular, ΔR variação do raio com a introdução de clotoide e α o ângulo entre os dois alinhamentos. O parâmetro Xm é calculado pela equação (12) e o parâmetro Xc é calculado pela equação (13).

$$X_m = X_c - R * \sin(\tau_{Max}) \quad (12)$$

onde, τ_{Max} corresponde ao ângulo da tangente no ponto final da clotoide.

$$X_c = L - (L^5/40 A^4) + (L^9 / 3456 * A^8) \quad (13)$$

onde, L corresponde desenvolvimento da clotoide, e A ao parâmetro da clotoide.

4.4 Desenvolvimento do traçado em altimetria

O traçado em perfil longitudinal (tendo em consideração a altimetria) tem como objetivo adequar a cota da via ao relevo do terreno. O perfil longitudinal deve aproximar-se o mais possível da cota do terreno, minimizando desta forma o volume de movimentos de terras e outro tipo de construções como pontes e túneis, diminuindo os custos totais de construção. Para traçar o perfil longitudinal é necessário fazer um levantamento com espaçamento regular da cota do terreno sobre o eixo da via da estrada. Após conhecer a relação entre a definição do terreno e o traçado da estrada, procede-se à definição dos traneis, t_1 e t_2 , e dos vértices em altimetria (v_1, v_2, v_3) como se exemplifica na figura 4. Os traneis t_1 e t_2 correspondem aos alinhamentos retos em altimetria.

No método proposto, na escolha da localização dos vértices no traçado em planta é estudada a relação com o terreno, que posteriormente também é estudada em perfil longitudinal na escolha dos vértices dos traneis. Por

aproximação, nesta abordagem, os vértices dos traneis em perfil longitudinal são os mesmos vértices do traçado em planta, afetados pela introdução da curva composta. Conhecida a inclinação i_1 e i_2 dos traneis t_1 e t_2 , formados pelos vértices v_1, v_2 e v_3 da figura 5, e o raio de concordância vertical, estão definidos todos os elementos necessários para traçar a concordância vertical, definida matematicamente por uma parábola.

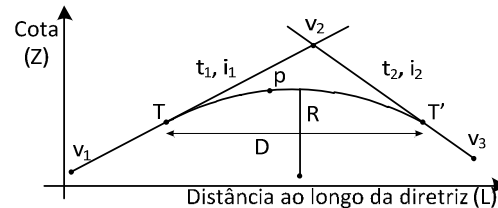


Figura 4. Concordância em perfil longitudinal

Como se pode ver na figura 4, a concordância é implementada entre os pontos T e T', com uma extensão (D) de desenvolvimento. A extensão (D) de desenvolvimento da concordância é obtida em função da inclinação (i_n) dos traneis e do raio (R) (14).

$$D = R * (i_1 - i_2) \quad (14)$$

O ponto T é determinado conhecendo a extensão do tranel t_1 , e o desenvolvimento (D) da concordância (15).

$$T = t_1 - D/2 \quad (15)$$

A cota do ponto T é determinada em função da cota do ponto inicial do tranel (C_1), a inclinação do tranel (i_1) e a extensão (t_1) do tranel (16).

$$CT = C_1 + i_1 * (t_1 - D/2) \quad (16)$$

A cota de um ponto p na concordância é calculada em função da cota do ponto T, da inclinação do tranel e da distância percorrida ao longo da concordância (17).

$$C_p = CT + (i_1 * d - d^2/2 * R) \quad (17)$$

Ao longo do traçado é determinada a cota dos pontos com o mesmo intervalo, que é utilizada no traçado em planta. Assim pode-se facilmente fazer a correspondência entre os pontos do traçado em planta e os pontos do perfil longitudinal, como se exemplifica na tabela 3.

Tabela 3. Piquetagem em perfil longitudinal

Km	M	P	Km	Cota
0+000	0	0	0+000	101,321
0+025	20	15	0+025	101,329
0+050	40	30	0+050	101,336

5. RESULTADOS

Para avaliação do método proposto e para efetuar uma análise crítica dos resultados obtidos, foi elaborado o seguinte caso de estudo: Pretende-se obter o traçado rodoviário para uma ligação do tipo autoestrada, sem nós de interligação, entre a cidade do Porto e a cidade de Braga. Considerou-se a população do Porto de 273584 habitantes e a população de Braga de 93587 habitantes. Esta ligação terá uma faixa com duas vias em cada sentido correspondendo no total a uma largura de referência de 40m. A velocidade base de projeto é de 120 km/h. Para o

problema apresentado foram gerados proceduralmente 8 traçados rodoviários do tipo autoestrada, utilizando o protótipo implementado e descrito na secção 4.

O modelo de terreno real em mapa de elevação no formato GEOTIFF afeto à ligação, foi obtido do repositório United States Geological Survey (earthexplorer.usgs.gov). As condicionantes foram definidas para uma área geográfica afetada pelo traçado da via. Nesta implementação, as áreas que representam condicionante são definidas num ficheiro de imagem no formato tiff [Libtiff14]. No processo de geração do traçado, obtém-se diferentes representações do eixo da via. Na figura 5, apresenta-se na imagem da esquerda o traçado em alinhamentos retos (preto) e o traçado em planta (azul), na imagem da direita o traçado em altimetria.

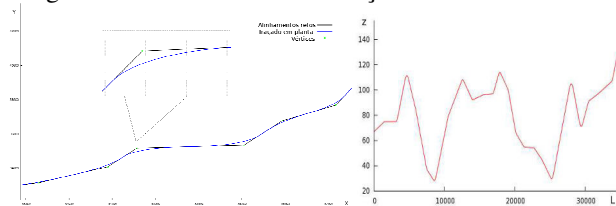


Figura 5. Traçado em alinhamentos

Na figura 6, apresenta-se um ambiente 3D gerado a partir de uma definição de traçado obtida proceduralmente.



Figura 6. Ambiente rodoviário (autoestrada)

Para o traçado final gerado, fez-se uma análise de custos. Considerando os vários parâmetros, zona floresta, construção da via, em ponte e túnel, movimento de terras e custo de utilização, obteve-se um custo total de 314,00 M€, o que corresponde a um valor de 6,70 M€ por quilómetro. Este valor enquadra-se dentro dos valores expectáveis para a construção de vias do tipo autoestrada.

O índice de sinuosidade em planta (I_s) é um dos indicadores a usar para analisar a sinuosidade de traçados rodoviários gerados pela abordagem apresentada, com estradas reais. Para cada um dos 8 traçados gerados proceduralmente, calculou-se o índice de sinuosidade, obtendo-se o valor médio de 728,75. O valor de sinuosidade em planta calculado para os traçados gerados enquadra-se na gama de valores admissíveis (valor de referência $\pm 1\sigma$), tendo como base o valor de referência da tabela 1. Pelo valor médio obtido para o índice de sinuosidade em planta, pode-se concluir que os traçados gerados pelo método proposto são semelhantes a traçados reais do tipo autoestrada.

Para se poder calcular a transformada discreta de Fourier é necessário efetuar uma discretização do traçado, sendo este amostrado com uma frequência de uma amostra por

metro ($f_s = 1 \text{ m}^{-1}$). Nesta análise apenas forma considerados 4 dos traçados rodoviários gerados proceduralmente, calculando-se as respetivas transformadas rápidas de Fourier, obtendo-se as representações apresentadas na figura 7.

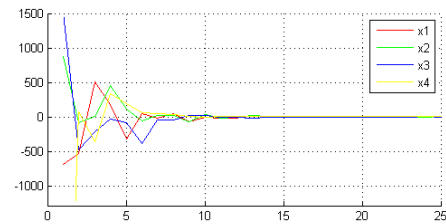


Figura 7. Análise de curvatura (autoestrada)

Como se pode ver pela figura 7, as respetivas transformadas apresentam um valor eficaz próximo de zero a partir do elemento 10. Em estudos realizados no âmbito deste trabalho conclui-se que traçados reais do tipo autoestrada apresentam curvas com valor eficaz próximo de zero a partir do elemento 10. A análise efetuada no domínio da curvatura permite concluir que os traçados do tipo autoestrada gerados apresentam características de sinuosidade em planta semelhantes aos traçados reais.

Calculou-se o índice de curvatura (10) para os traçados gerados proceduralmente, obtendo-se o valor médio de $I_c = 0,93$. Fazendo uma análise comparativa entre o valor médio obtido para os traçados gerados proceduralmente com o valor de referência para este indicador, conclui-se que os valores de I_c são da mesma ordem de grandeza e enquadrados nos limites de variância admissível (valor de referência $\pm 1\sigma$). Com a análise efetuada ao índice de curvatura, pode-se afirmar que os traçados obtidos proceduralmente do tipo autoestrada são semelhantes aos traçados reais, em termos de índice de curvatura.

Efetuada uma análise em termos de tempo, conclui-se que o método proposto comparado com metodologias mais recentes apresentadas por Galin et al. em 2010, permite reduzir significativamente o tempo necessário para a obtenção de traçados (15%), mantendo os padrões de realismo exigidos. Esta redução de tempo é mais significativa em traçados de grandes dimensões ($> 50 \text{ km}$), independentemente da dimensão da grelha do terreno.

6. CONCLUSÕES E TRABALHO FUTURO

Experiências realistas de simulação de condução requerem a prévia preparação de modelos de ambientes rodoviários corretamente modelados, com características semelhantes às encontradas em traçado real. O método apresentado neste artigo permite gerar de forma automática qualquer tipo de traçado rodoviário, de grandes dimensões, adequado para simulação de condução para fins científicos.

O processo de geração de traçados rodoviários concebido foi inspirado nos métodos utilizados em engenharia de vias, produzindo estradas segundo as normas e com traçados semelhantes aos encontrados em terreno real. A introdução de curvas compostas em planta e concordâncias verticais em altimetria são dois fatores de relevo que contribuem para o realismo dos traçados obtidos. Os tra-

çados gerados possuem características geométricas segundo as práticas comuns em projeto de vias.

O problema da geração de um traçado entre dois pontos é analisado de forma global, sendo tratado em diferentes níveis de detalhe. A solução final de traçado, gerada em cada nível de detalhe, é determinada recorrendo ao cálculo de diferentes custos, que avaliam a relação com o relevo do terreno e respetivas condicionantes, custo de construção e custo de utilização (secção 3.2.1). Ao longo do processo de geração automática, o traçado rodoviário vai sendo refinado, produzindo no final definições de estradas com a pormenorização necessária à geração de modelos visuais de qualidade.

O método proposto permite obter uma grande diversidade de traçados rodoviários, reduzindo drasticamente o trabalho e os custos envolvidos na sua conceção, uma vez que reduz significativamente o recurso a especialistas da área de projeto de vias de comunicação para obtenção de dados de projeto de estradas. A realização deste trabalho traduz-se num contributo inovador na geração automática de traçados rodoviários realistas. Uma das principais vertentes de aplicação desta abordagem será a geração de traçados rodoviários adequados à simulação de condução, permitindo a realização de estudos científicos nas mais diversas áreas.

Num futuro próximo, será possível apresentar modelos de estradas de diferentes tipos (rural, autoestrada) e os resultados da avaliação experimental em simulador com especialista da área de projeto.

7. AGRADECIMENTOS

A realização deste trabalho de investigação contou com o contributo especial do diretor do laboratório de análise de tráfego, onde se encontra instalado o simulador de condução DriS, Prof. Dr. Carlos Rodrigues do Departamento de Engenharia Civil da FEUP, e do Prof. Ângelo Jacob do Departamento de Engenharia Civil do ISEP.

O projeto MAT (Media Arts and Technologies - NORTE-07-0124-FEDER-000061), é financiado pelo Programa Operacional Regional do Norte (ON.2 – O Novo Norte), através do Quadro de Referência Estratégico Nacional 2007/2013, financiado pelo Fundo Europeu de Desenvolvimento Regional (Feder) e por fundos nacionais através da Fundação para a Ciência e a Tecnologia (FCT).

8. REFERENCIAS

- [Bayarri96] Bayarri, S; Fernandez, M; Perez, M; Virtual Reality for driving simulation-Bayarri, Vol. 39, n.º 5, Communications of the ACM, 1996.
- [Campos07] Campos, C.; Leitão, J.; Rodrigues, C.; Modelação de Ambientes Rodoviários de Grandes Dimensões, 15.º Encontro Português de Computação Gráfica, Outubro de 2007.
- [Campos12] Campos, C., Leitão, J., Coelho, A.; Modelação Procedimental de Ambientes Rodoviários para Simulação de Condução, 20.º Encontro Português de Computação Gráfica, Outubro de 2012.
- [Coelho07] Coelho, A.; Bessa, M.; Sousa, A.; Ferreira, N.; Expedient modeling of virtual urban environments with geospatial L-systems; Computer Graphics Forum, Vol. 26, N. 4, pp. 769–782, 2007.
- [EP94] Normas de traçado, Junta Autónoma das Estradas, Atual Estradas de Portugal (EP), ISBN-96379-6-2.
- [França11] França, A.; Apontamentos de vias de comunicação, Faculdade de Engenharia da Universidade do Porto, Julho de 2011.
- [Galin10] Galin, E.; Peytavie, A.; Maréchal, N.; Guérin, E.; Procedural Generation of Roads, EUROGRAPHICS, Volume 29, 2010.
- [Galin11] Galin, E.; Peytavie, A.; Guérin, E.; Benes, B.; Authoring Hierarchical Road Networks, Pacific Graphics, Volume 30, 2011.
- [Ingle12] Ingle, V., Proakis, J.; Digital Signal Processing Using MatLab, Third Edition, cap. 3, Gengage Learning, 2012.
- [Kelly08] Kelly, G.; McCabe, H.; Citygen: An Interactive System for Procedural City Generation. GDTW, UK, 2008.
- [Latham06] Latham, R.; Burns, D.: Dynamic Terrain Modification Using a Correction Algorithm, IMAGE 2006 Conference Scottsdale, Arizona, Julho de 2006.
- [Libtiff14] TIFF Library and Utilities, informação disponível em: <http://www.libtiff.org/>, Abril de 2014.
- [Mitra05] Mitra, S.; Digital Signal Processing, A Computed-Based Approach, third Edition, cap. 3, McGraw-Hill, 2005
- [OpenDrive10] OpenDRIVE, Format Specification REV 1.3, documento disponível em: <http://www.opendrive.org/docs/OpenDRIVEFormatSpecRev1.3D.pdf>, Agosto de 2010.
- [Pareja99] Pareja, I.; Bayarri, S.; Rueda, S.; Modelado de calidad para la visualización interactiva de carreteras, IX Congreso de Ingeniería Informática, 1999.
- [Parish01] Parish, Y.; Muller, P.; Procedural Modeling of Cities; ACM Computer Graphics (Proceedings of SIGGRAPH'2001), pp. 301-308, 2001.
- [Smelik08] Smelik R. M.; Tutenel T.; Kraker K.J.; Bidarra R.; A Proposal for a Procedural Terrain Modelling Framework, EGVE Symposium, 2008.
- [Smelik11] Smelik, R., Tutenel, T., Kraker, J., Bidarra, R.; A declarative approach to procedural modeling of virtual worlds, Computer & Graphics Journal, Elsevier, 2011.
- [Teoh08] Teoh, S.; Algorithms for the Automatic Generation of Urban Streets and Buildings, Proceedings of the International Conference on Computer Graphics and Virtual Reality (CGVR'08), Julho 2008.
- [Thomas00] Thomas, G. and Donikian, S.; Modelling Virtual Cities Dedicated to Behavioural Animation. EUROGRAPHICS 2000, vol. 19(3), 2000.

Modelação Expedita de Terrenos e Subsolos via Interação Espacial e Superfícies Interactivas

Daniel Simões Lopes
INESC-ID Lisboa
daniel.lopes@inesc-id.pt

Daniel D. Simão Fernando Fonseca
Daniel Mendes Joaquim A. Jorge
INESC-ID Lisboa,
Instituto Superior Técnico, Universidade de Lisboa
{daniel.simao, fernando.fonseca,
danielmendes, jorgej}@tecnico.ulisboa.pt

Resumo

A ilustração e visualização 3D de modelos geológicos são tarefas comuns em muitas áreas da geologia. Na larga maioria dos casos, tais tarefas são realizadas recorrendo ao desenho sobre meios bidimensionais, com o intuito de transmitir uma percepção 3D aos conceitos geológicos traçados. Este artigo apresenta um sistema de interação espacial baseada em gestos para a construção rápida de modelos e ilustrações geológicas. Os modelos caracterizam-se por serem compostos por camadas homogéneas (isto é, sem falhas nem dobras geológicas) empilhadas entre si. Partindo de cenários sem dados, o sistema permite a modelação, edição e exploração expedita de terrenos e subsolos num ambiente 3D. O princípio de modelação baseia-se no facto da geometria de um terreno ou subcamada poder ser modelada por interação espacial. Aqui, este princípio é explorado e são analisadas a representatividade e expressividade geológica contidas em gestos. As interações do utilizador com o sistema são adquiridas através de tecnologias de captura do movimento e mesas multitoque. A visualização é garantida por tecnologia estereoscópica que proporciona uma percepção 3D do modelo. Os resultados preliminares de estudos com utilizadores mostraram que o sistema proposto é muito útil para a visualização ilustrativa e para a rápida modelação de objectos estratigráficos, permitindo realizar uma modelação expedita de características geológicas grosseiras com um número limitado de gestos. O sistema ainda permite modelar detalhes mais finos ao interagir com a superfície interactiva via toque com os dedos. A principal conclusão é que gestos com a mão revelam-se um input bastante eficaz tanto para a criação de terrenos como de camadas estratigráficas.

Palavras-Chave

Geologia, terrenos, subcamadas, modelação expedita, interação baseada em gestos, visualização ilustrativa, estereoscopia.

1. INTRODUÇÃO

Ilustrar e visualizar conceitos geológicos em 3D são tarefas comuns em muitas áreas da geologia. Tais tarefas visam a comunicação de relações espaciais e temporais de possíveis configurações estruturais apresentadas por um objecto geológico [Lidal 2013; Turner 2006]. O desenho de ilustrações tridimensionais é muitas vezes realizado sobre meios bi-dimensionais como tentativa de

transmitir uma percepção espacial fora do plano de desenho. A externalização de uma ideia ou conceito, principalmente se se partir de um cenário sem dados, é uma tarefa muito comum em geologia [Natali et al. 2013]. A falta ou ausência de dados obriga o ilustrador/modelador a focar-se nas características globais do modelo, podendo então ser complementadas *a posteriori* quando mais dados estiverem disponíveis.

Assim, o processo de modelação é orientado pelo conhecimento geral do geólogo, podendo necessitar de várias iterações para produzir o resultado desejado. Note-se que cada iteração consiste num esboço rápido já que não há informações como base. Por forma a construir modelos geológicos mais detalhados são obrigatoriamente necessários dados reais.

Especialistas em geologia já expressaram a necessidade de sistemas computacionais que permitam a modelação geológica rápida de terrenos e elementos estratigráficos recorrendo a esboços bidimensionais [Lidal 2013]. No entanto, sistemas de interacção espacial, isto é, sistemas que permitem a modelação de conteúdos em 3D recorrendo à interacção espacial, conseguem fornecer um meio ainda mais rápido para construir rapidamente um modelo geológico.

Aplicações geológicas onde sistemas de desenho baseados em gestos se poderão revelar úteis são a exploração mineira e exploração de petróleo. Na primeira aplicação, a triagem inicial de uma prospecção requer a visualização de conceitos iniciais sobre arranjos estruturais. Para tal, são necessários modelos geológicos altamente precisos com uma grande quantidade de detalhes e a definição precisa das condições do subsolo, algo que pode levar vários meses para construir o modelo final [Turner 2006]. Pelo contrário em exploração petrolífera, durante a fase inicial de exploração são necessários sistemas de modelação rápida. Esta fase de exploração é caracterizada por ter muito poucos, ou mesmo nenhuns, dados disponíveis o que dificulta tremendamente a procura de recursos inexplorados de petróleo. Desta forma, dado que a exploração e avaliação de modelos geológicos alternativos é realizada sob fortes restrições de tempo é uma tarefa adequada para sistemas de modelação rápida.

Este artigo propõe um sistema de interacção espacial baseada em gestos para a construção expedita de modelos geológicos em camadas (ou seja, terreno e subsolo). As superfícies são representadas como mapas de alturas (representação comum para modelação de terrenos e camadas). O objectivo consiste em usar o sistema para construir modelos estratigráficos para efeitos de ilustração e visualização. A simulação geológica não é aqui considerada. O sistema permite agilizar a modelação e edição assim como a exploração de modelos geológicos partindo de um cenário sem dados. Os modelos considerados são conceitos e ideias iniciais não apresentando detalhes geológicos intrincados. Uma das novidades deste trabalho consiste na realização da modelação rápida de características geológicas grosseiras com apenas um conjunto limitado de gestos no ar.

2. TRABALHO RELACIONADO

A larga maioria dos sistemas de modelação geológica, tais como os software Petrel e GOCAD, visam construir modelos precisos, estando focados na modelação de pequenos detalhes e pormenores geológicos com grande

exactidão e realismo. Estes sistemas requerem grandes quantidades de dados e o processo de modelação leva vários meses para ser concluído. Desta forma, estes sistemas não permitem expressar conceitos iniciais ou esboços de uma forma expedita, logo são impraticáveis para comunicar propostas alternativas de geometrias em tempo útil, uma tarefa recorrente na exploração mineira e petrolífera.

As necessidades e os desafios em desenvolver sistemas computacionais para construir de forma expedita modelos geológicos tem sido relatada em vários trabalhos [Lidal 2013; Turner 2006; Olsen 2004; Peytavie et al. 2009]. No entanto, poucos adoptam uma abordagem baseada em esboços [Natali et al. 2012; Lidal et al. 2013, Amorim et al. 2012]. Provavelmente, o trabalho mais notório relativo a sistemas computacionais para desenho de esboços geológicos é apresentado por [Lidal 2013]. Na sua tese, foram apresentados vários sistemas baseados em esboços para modelar rapidamente geometrias estruturadas por camadas. Apesar de terem sido considerados diferentes abordagens para criar esboços 3D, nenhum deles faz uso de interfaces multimodais interactivas onde gestos são usados para modelar explicitamente superfícies.

Embora a abordagem para modelação 3D de esboços usando o espaço livre (e.g., espaço em cima de uma mesa) não seja novidade [Galyean & Hughes, 1991; Schkolne et al. 2001, Keefe et al. 2001], o desenvolvimento de um sistema de interacção espacial baseada em gestos, especificamente desenhado para construir rapidamente modelos geológicos com camadas, nunca fora desenvolvido antes.

3. SISTEMA DE INTERACÇÃO ESPACIAL BASEADA EM GESTOS

O sistema proposto tem como objectivo modelar superfícies, nomeadamente, mapas de alturas, fazendo uso da rapidez e expressividade de gestos de mão. Por forma a exteriorizar rapidamente conceitos em superfícies, a interacção homem-computador é baseada em gestos e no toque através do recurso de tecnologias de captura de movimento assim como de uma mesa de grandes dimensões, colocado numa posição horizontal à altura da cintura e sensível ao toque (Figura 1). Já a percepção 3D do conteúdo é conseguida através de um ecrã estereoscópico usando óculos polarizados juntamente com uma câmara de profundidade que regista o posicionamento da cabeça relativamente ao ecrã (Figura 1).

O utilizador encontra-se cercado por câmaras de captura de movimento que monitorizam continuamente os marcadores ópticos que instrumentam o dispositivo de desenho (i.e., caneta). Ao interagir com a aplicação, o utilizador visualiza uma caixa virtual (*proxy*) no interior da qual pode desenhar as superfícies e aceder a um menu para evocar as diversas funcionalidades de modelação e visualização (Figura 2). A superfície do ecrã juntamente com o espaço acima da superfície, confere ao utilizador

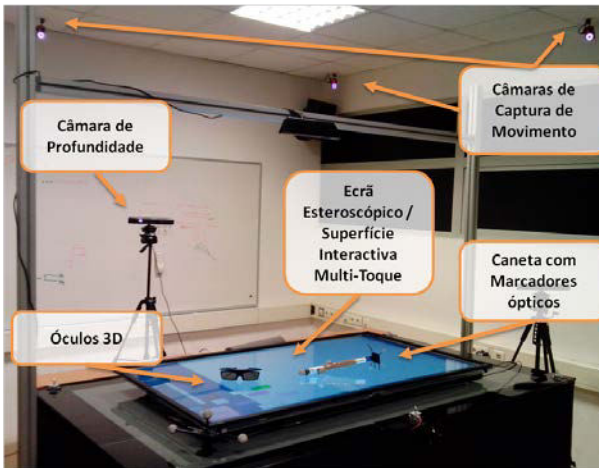


Figura 1: O equipamento do sistema de interacção espacial baseada em gestos.

toda a liberdade necessária para desenhar livremente no ar.

A interacção com o sistema segue um modelo bimanual assimétrico [Guiard 1987] que considera as acções do utilizador, dominância manual e espaços de interacção (Figura 3). Para este efeito, o sistema exige que o utilizador use a mão dominante (MD) para esboçar em pleno ar recorrendo à caneta com marcadores ópticos ou desenhar no ecrã multitoque. A MD pode também ser usada para mover a caixa de *proxy* e para complementar as tarefas de rotação e escalonamento realizadas em conjunto com a mão não-dominante (MND). Já as tarefas de corte da *proxy*, selecção da camada e alternar entre visão de câmara e edição de modelo são tarefas mais adequadas para a MND.

Ainda relativamente à geração de conteúdo 3D, este é

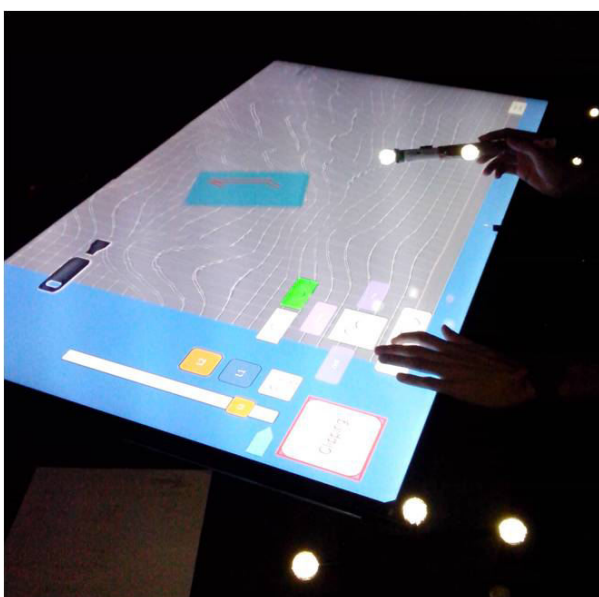


Figura 2: Interface de modelação e visualização 3D com exposição dos menus, caneta e *proxy*.



Figura 3: Acções do utilizador, dominância manual e espaços de interacção do sistema de interacção espacial proposto.

produzido no interior da *proxy* através de gestos no ar (apenas com a MD) e toques (com MD e MND) que actuam como inputs de modelação ou edição geométrica (Figura 2). A MD segura uma caneta devidamente instrumentada com marcadores ópticos para que as coordenadas e orientações 3D sejam adquiridas usando a tecnologia de captura de movimento em tempo real (Figura 2). Com base na posição 3D e orientação da caneta, os vértices da malha das superfícies podem ser editados no interior da *proxy*. Adicionalmente, detalhes geológicos mais finos podem ser esboçados com os dedos sobre a superfície interactiva (Figura 4).

3.1 Transformações Afins

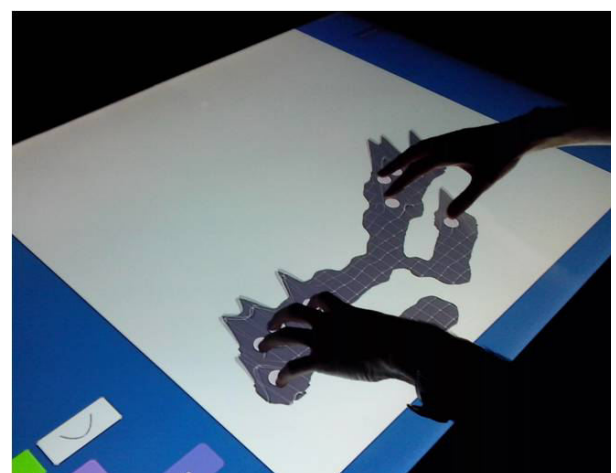


Figura 4: Edição de uma superfície com ambas as mãos MD e MND recorrendo a multi-toque sobre a mesa estereoscópica.

O processo de modelação exige a visualização do conteúdo 3D segundo diferentes pontos de vista. Para visualizar o conteúdo modelado, transformações afim são aplicados sobre o conteúdo da mesa recorrendo ao uso fácil do multi-toque [Kim et al. 2006]. Estas transformações são realizadas usando o seguinte conjunto de interações com o ambiente virtual: (i) gesto de arrastar o dedo sobre a superfície para a translação; (ii) gesto de rodar dois dedos, um em relação ao outro para girar a *proxy* em torno do eixo que atravessa verticalmente o seu centro; e (iii) gesto de arrastar dois dedos juntando/afastando-os para realizar o *zoom in* e *zoom out*, respectivamente.

3.2 Produção Expedita de Terrenos e Camadas de Subsolo

O sistema de interação espacial aqui desenvolvido baseia-se no seguinte princípio de modelação: partindo de um cenário sem dados, a estrutura geral de uma superfície geológica pode ser modelado por um conjunto bastante limitado de interações espaciais. Com base neste princípio, a criação de modelos geológicos faz uso de dois tipos de interações gestuais: gestos no ar para criar o conteúdo em geral, e gestos de toque para adicionar detalhes e suavizar qualquer artefacto introduzido por gestos. Em particular, a modelação e edição de superfícies consistem na alteração de um mapa de alturas através da introdução de várias elevações e depressões. Para interações espaciais, a altura da caneta relativamente ao ecrã táctil corresponde à amplitude de uma função super-Gaussiana elíptica,

$$F_{SGE}(x, y) = A \exp \left(- \left(\frac{(x-x_0)^2}{2\sigma_x^2} + \frac{(y-y_0)^2}{2\sigma_y^2} \right)^\gamma \right) \quad (1)$$

onde A é a amplitude ou altura; x_0 e y_0 o centro; σ_x e σ_y a variância segundo x e y , respectivamente; γ , $\gamma \geq 1$, o expoente que confere um formato circular ($\gamma \approx 2$) ou quadrado ($\gamma > 2$) à gaussiana. Note-se que uma Gaussiana circular é um caso particular da (Equação 1) quando $\sigma_x = \sigma_y$ e $\gamma = 1$. Fazendo uso da orientação da caneta relativamente ao plano do ecrã é possível ainda de rodar a função F_{SGE} em torno do eixo que sai da mesa (Figura 5 (a)). Para gestos de toque, são usadas Gaussianas

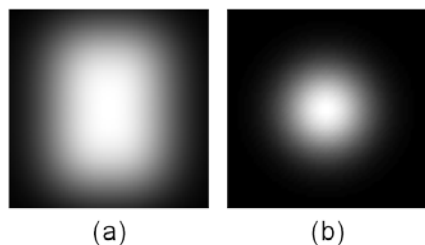


Figure 5 – Funções matemáticas usadas para representar elevações e depressões do mapa de alturas através (a) interação espacial (super gaussiana elíptica) e (b) toque com o dedo (gaussiana circular).

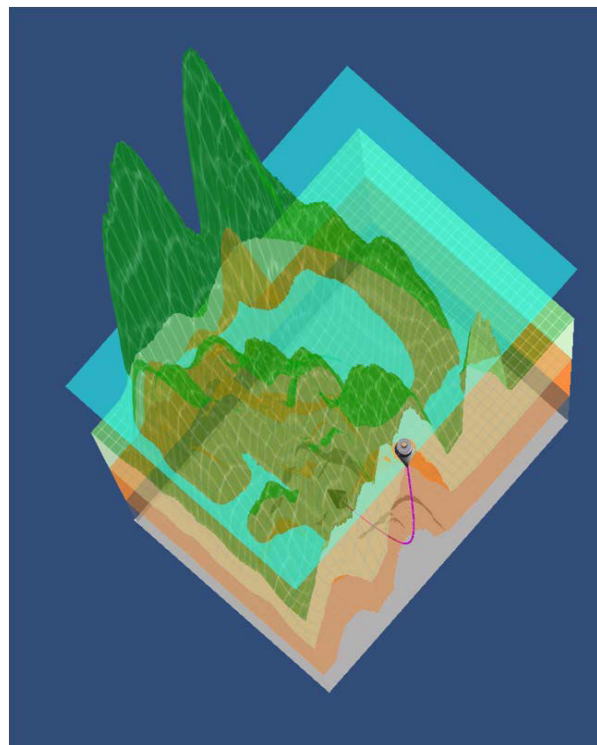


Figure 6 – Exemplo de modelo geológico gerado pelo sistema de interação espacial baseado em gestos.

circulares para editar elevações ou depressões mas de área mais reduzida (Figura 5 (b)). É ainda possível introduzir várias Gaussianas circulares tocando na tela em vários pontos (Figura 4). Aqui, a convenção adoptada para contruir um modelo geológico consiste em partir de baixo para cima, i.e., a primeira superfície corresponde à camada mais profunda e as camadas seguintes são empilhadas por ordem estratigráfica, terminando-se com a criação da superfície do terreno (Figura 6).

Note-se que, devido à semelhança óbvia, os modelos aqui considerados designam-se por "camadas de bolo" por terem como característica dominante o empilhamento homogéneo de camadas, isto é, não apresentam falhas nem dobras geológicas.

3.3 Corte

Explorar o interior de uma estrutura composta por camadas recorrendo a planos de corte é um requisito necessário para revelar a quantidade de informação contida dentro do objecto volumétrico (Figura 7). Foram implementados planos de corte que se podem deslocar ao longo dos eixos canónicos horizontais.

O utilizador pode colocar planos de corte numa determinada zona de interesse, aparecendo sobre os planos de corte as linhas de fronteira entre as diferentes camadas (Figura 7 (b)).

4. ESTUDO DE USABILIDADE

Um estudo de usabilidade foi realizado para verificar as vantagens e limitações do sistema de interacção espacial baseada em gestos e, mais importante, para estudar o impacto do gesto em modelação geológica.

A um total de oito utilizadores (25 ± 6 anos de idade), sem experiência prévia em modelação de terrenos ou objectos geológicos, foi pedidos para desenhar três camadas simples, na seguinte ordem: a camada inferior como uma superfície plana, a camada do meio com um canal no centro e a camada de topo como uma montanha com um único pináculo.

Todos os utilizadores relataram que o sistema permitia criar estruturas em camadas de uma forma rápida e simples. Relataram também que os gestos são adequados para definir o conteúdo geral das superfícies enquanto o toque com dedos são óptimos para atenuar os artefactos não-suaves e para adicionar pequenos detalhes. Para além disto, as transformações afim e as funcionalidades de corte foram considerados muito úteis para explorar o conteúdo gerado. A maioria dos utilizadores indicou que o rastreamento da cabeça e a estereoscopia 3D aumentam a percepção espacial do que foi modelado, enquanto que uma minoria considerou que a estereoscopia não trazia valor acrescentado.

O feedback mais notável ocorreu aquando do desenho de uma superfície plana. Embora pareça simples, tal tarefa mostrou ser bastante exigente. Os resultados obtidos consistiram em superfícies quase planas mas sempre ligeiramente inclinadas face ao plano horizontal. Tal indica que gestos no ar introduzem artefactos sempre que a modelação é restrita a superfícies ideais, como é o caso do plano.

5. DISCUSSÃO

Neste trabalho apresentamos um sistema de interacção espacial por gestos especificamente concebido para modelar objectos geológicos constituído por camadas

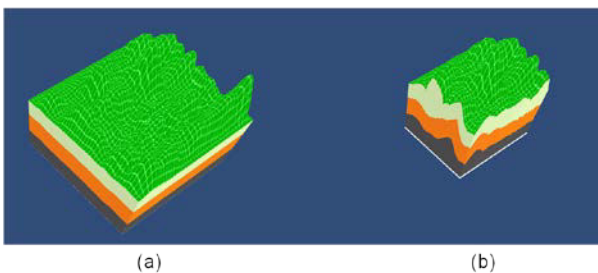


Figura 7 – Cortes aplicados à proxy para efeitos de exploração do conteúdo no seu interior: (a) proxy original, (b) proxy cortada.

homogéneas. O sistema permite que os utilizadores externalizem às suas ideias assim como esboçar alguns detalhes mais finos. Explora-se o princípio de modelação em que a estrutura geral de um terreno ou de uma camada estratigráfica consegue ser modelada por um conjunto de gestos grosseiros. Analiza-se também a expressividade geológica por gestos e verifica-se quão rápido um objecto estratigráfico pode ser criado.

Os resultados obtidos foram satisfatórios em termos de rápida visualização ilustrativa e para efeitos de comunicação de ideias. Os testes preliminares com utilizadores indicam que o uso de gestos dentro de uma *proxy* consiste numa interface natural e fácil de usar para esboçar rapidamente modelos geológicos simples.

Além disso, o sistema permite produzir um grande número de modelos geológicos alternativos num intervalo de tempo relativamente curto, comparativamente aos sistemas de desenho tradicionais que são feitos construir modelos geológicos muito precisos, que requerem grandes quantidades de dados e que exigem meses de trabalho de modelação.

Mesmo permitindo uma modelação expedita, esboçar com gestos é um processo que requer algum treino para adquirir proficiência. Ao contrário de outras ferramentas de desenho, a metodologia é menos demorada. A novidade do método apresentado consiste em desenhar rapidamente em todo o domínio de uma caixa *proxy*, a fim de criar camadas de espessura variável. Detalhes mais finos podem ser adicionados às superfícies com toques.

O estudo empírico realizado com os utilizadores mostra que a técnica de interacção de esboços no ar é altamente apropriado para obter rapidamente um esboço geral, enquanto o esboço na mesa tem a vantagem de modelar uma superfície em vários lugares, com diferentes tipos de pincel e de forma colaborativa.

Como uma nota sobre ergonomia, dado que o ecrã interactivo é uma superfície bastante larga, a modelação pode tornar-se uma tarefa cansativa, difícil e desconfortável caso o alvo desejado esteja fora de alcance [Toney e Thomas 2006]. Por exemplo, tal acontece com frequência nas extremidades da superfície em relação ao utilizador. Esta limitação é parcialmente resolvida movendo o conteúdo para mais próximo do modelador na tela de toque recorrendo à opção de translação.

6. CONCLUSÕES & TRABALHO FUTURO

Este trabalho apresenta um sistema baseado esboços para modelação e edição de camadas geológicas com gestos, onde as superfícies são definidas como mapas de altura. Os resultados preliminares dos testes de utilizadores mostram que o sistema proposto se revelou útil para a visualização ilustrativa e modelação rápida de objectos geológicos. A principal conclusão é que gestos no ar revelam ser eficientes na modelação de terrenos e camadas geológicas.

Como trabalho futuro, ir-se-à explorar cenários geológicos na presença de poucos dados e de muitos dados. Para tal, recorre-se ao sistema para efectivar a conclusão rápida de um modelo recorrendo a técnicas de extrapolação e interpolação de nuvens de pontos. Futuramente, apontamos para simulações geológicas dependentes do tempo para representar cenários hipotéticos ao longo da história natural, também conhecidos como métodos inversos [Wijns 2003 Guillen et al. 2008, Caumon 2010]. Ainda como desenvolvimentos futuros, pretende-se introduzir descontinuidades geométricas nos os modelos para modelar falhas e fracturas, dado que são características comuns em muitos cenários geológicos.

7. AGRADECIMENTOS

Os autores agradecem o financiamento proporcionado pela Fundação para a Ciência e Tecnologia (FCT). O primeiro autor agradece a bolsa de pós-doutoramento SFRH/BPD/97449/2013. O quarto autor agradece a bolsa de doutoramento SFRH/BD/91372/2012. Este trabalho foi também parcialmente financiado através dos projetos TECTON-3D PTDC/EEI-SII/3154/2012 e Pest-OE/EEI/LA0021/2013.

8. REFERÊNCIAS

- AMORIM R., BRAZIL E. V., PATEL D., SOUSA M. C., 2012. Sketch Modeling of Seismic Horizons from Uncertainty. Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling (SBIM) '12, 1-10.
- CAUMON, G. 2010. Towards Stochastic Time-Varying Geological Modeling. *Mathematical Geosciences* 42, 5, 555-569.
- GALYEAN T. A., HUGHES J. F., 1991. Sculpting: an interactive volumetric modeling technique. Proceedings of the 18th annual conference on Computer graphics and interactive techniques '91 (SIGGRAPH), 267-274.
- GUIARD, Y. 1987. Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model. *Journal of Motor Behavior* 19, 486-517.
- GUILLEN, A. CALCAGNO, PH., COURRIOUX, G., JOLY, A., AND LEDRU, P. 2008. Geological modeling from field data and geological knowledge: Part II. Modelling validation using gravity and magnetic data inversion. *Physics of the Earth and Planetary Interiors* 171, Recent Advances in Computational Geodynamics: Theory, Numerics and Applications, 1-4, 158-169.
- KEEFE D., ACEVEDO D., MOSCOVICH T., LAIDLAW D. H., LAVIOLA J., 2001. CavePainting: A Fully Immersive 3D Artistic Medium and Interactive Experience. *Proceedings of ACM Symposium on Interactive 3D Graphics '01*, 85-93.
- KIM, S.-G., KIM, J.-W., BAE, K.-T., AND LEE, C.-W. 2006. Multi-touch Interaction for Table-Top Display. *Advances in Artificial Reality and Tele-Existence, Lecture Notes in Computer Science* 4282, Springer Berlin Heidelberg, 1273-1282.
- LIDAL, E. M., 2013. Sketch-based Storytelling for Cognitive Problem Solving Externalization, Evaluation, and Communication in Geology. PhD Thesis. University of Bergen.
- LIDAL, E. M., NATALI, M., PATEL, D., HAUSER, H., AND VIOLA, I. 2013. Geological storytelling. *Computers & Graphics* 37, 5, 445-459.
- NATALI, M. AND VIOLA, I. AND PATEL, D. 2012. Rapid Visualization of Geological Concepts. In Proceedings of the 25th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), 150-157.
- NATALI, M., LIDAL, E. M., PARULEK, J., VIOLA, I., AND PATEL, D. 2013. Modeling Terrains and Sub-surface Geology. In EuroGraphics 2013 State of the Art Reports (STARs), 155-173.
- OLSEN, J., 2004. Realtime procedural terrain generation. Technical report, Department of Mathematics And Computer Science, (IMADA) University of Southern Denmark.
- PARADIGM GOCAD Software framework for subsurface modeling.
<<http://www.pdgm.com/Products/GOCAD.aspx>>, Accessed: April 201
- PEYTAVIE, A., GALIN, E., MERILLOU, S., AND GROSJEAN, J. 2009. Procedural Generation of Rock Piles Using Aperiodic Tiling. In Proceedings of Pacific Graphics 28, 7, 1801-1810.
- SCHKOLNE S., PRUETT M., SCHRÖDER P., 2001. Surface Drawing: creating organic 3D shapes with the hand and tangible tools. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems '01, 261-268.
- SCHLUMBERGER LIMITED Petrel EP Software Platform. Software package for oil and gas exploration and production.
<<http://www.slb.com/petrel.aspx>>, Accessed: April 2014.
- TONEY, A. AND THOMAS, B.H. 2006. Considering reach in tangible and table top design. First IEEE International Workshop on Horizontal Interactive Human-Computer Systems, 2006. Table-Top 2006, january, 2 pp.
- TURNER, K., 2006. Challenges and trends for geological modelling and visualisation. *Bulletin of Engineering Geology and the Environment* 65, 2 (May), 109-127.
- WIJNS, C., BOSCHETTI, F., AND MORESI, L. 2003. Inverse modelling in geology by interactive evolutionary computation. *Journal of Structural Geology Computers* 25, 10, 1615-1621.

Segmentação de Malhas Triangulares Baseada em Convexidade Relaxada

Rui S. V. Rodrigues José F. M. Morgado
Escola Superior de Tecnologia e Gestão de Viseu, IPV
Viseu
{rsrodrigues, fmorgado}@estv.ipv.pt

Abel J. P. Gomes
Instituto de Telecomunicações, UBI
Covilhã
agomes@di.ubi.pt

Resumo

Este artigo introduz uma nova técnica de segmentação de malhas triangulares, designada por segmentação de montes e vales (MeV), que combina técnicas de segmentação baseadas na fronteira e no interior de regiões. O algoritmo MeV usa uma função altura com sinal com o objectivo de distinguir entre montes (+) e vales (-) e zonas planares (0). Cada região é construída à volta de extremos locais (i.e., máximos e mínimos) da malha. Um monte é construído a partir de um máximo e um vale a partir de um mínimo. No entanto, é possível às fronteiras dos montes invadirem parcialmente os vales e vice-versa. Consequentemente, vamos ter montes que formam regiões convexas relaxadas e vales que formam regiões côncavas relaxadas. Ao contrário do actual estado da arte da segmentação de malhas, a existência destas regiões relaxadas torna o algoritmo eficaz na segmentação de diferentes tipos de objectos (de forma livre ou não).

Palavras-Chave

Malhas triangulares, segmentação de malhas, convexidade relaxada.

1 INTRODUÇÃO

A segmentação de malhas é importante em áreas tão diversas como a modelação geométrica, desenho assistido por computador e computação gráfica. A segmentação de malhas de triângulos consiste na partição da malha num conjunto de sub-malhas. Mas, como refere Attene et al. [Attene 06a], a segmentação de malhas pode ser feita com base em critérios geométricos ou critérios semânticos. Nas técnicas de segmentação baseadas em propriedades geométricas, a malha é dividida num conjunto de sub-malhas que satisfazem uma determinada propriedade geométrica (por exemplo, curvatura ou distância a um plano). Por outro lado, nas técnicas de segmentação baseadas em semântica, a divisão da malha em sub-malhas tem lugar quando cada sub-malha delimita um região perceptualmente significativa (por exemplo, um braço do corpo humano).

Na verdade, os métodos de análise dos objectos não fornecem uma descrição semântica das suas formas, mas eles permitem uma caracterização geométrica e estrutural dos objectos. No entanto, acredita-se que esta informação, aliada ao conhecimento das ciências cognitivas acerca da percepção humana, permite a definição de novos mecanismos para estruturar e perceber a forma dos objectos. Neste âmbito, Biederman [Biederman 87] diz que as pessoas percebem os objectos como uma coleção de partes, enquanto Hoffman [Hoffman 97] refere que a visão humana define fronteiras das partes ao longo dos mínimos negati-

vos das curvaturas principais. Note que, de certa maneira, a asserção de Hoffman implica que as partes significativas dos objectos são convexas. Como vamos ver mais à frente, o algoritmo proposto neste artigo segue esta asserção de Hoffman.

1.1 Trabalho relacionado

A segmentação de objetos consiste na divisão de um objeto em sub-objetos significativos em termos de forma. Na nossa opinião, a melhor maneira de classificar os algoritmos de segmentação de malhas é ter em conta a dimensão estrutural dos objetos. Assim sendo, pode dizer-se que há três categorias principais de técnicas de segmentação:

1. *Segmentação baseada em volume.* Neste caso, os segmentos são volumes. A entrada é uma malha volumétrica 3D, que é particionada em sub-malhas volumétricas 3D [Chazelle 81] [Bajaj 92] [Lien 04] [Ghosh 13].
2. *Segmentação baseada em superfície.* Nesta técnica, os segmentos são regiões definidas por malhas de triângulos 2D. Cada região consiste num conjunto de faces ligadas que tem uma propriedade geométrica semelhante (por exemplo, convexidade e curvatura).
3. *Segmentação baseada em esqueleto (skeleton).* Nesta técnica, também conhecida como esqueletização, os segmentos são segmentos de reta. A entrada é uma



Figura 1. Um conjunto diverso de malhas, segmentadas usando o algoritmo MeV.

malha volumétrica 3D ou uma malha de superfície 2D, mas a saída é um esqueleto 1D que representa a forma estrutural da malha. Gerar este *esqueleto* é um processo conhecido como esqueletização (*skeleton extraction* ou *skeletonization*) [Li 01] [Capell 02].

No que diz respeito aos algoritmos de *segmentação baseada em superfície*, temos as seguintes sub-categorias principais: *region growing* [Chazelle 95] [Kalvin 96], *watersheds* [Mangan 99] [Zuckerberger 02], *hierarchical clustering* [Garland 01], [Katz 05], *iterative clustering* [Lloyd 82], *spectral clustering* [Chung 96], [Zhang 07], [Reuter 09], e *fuzzy clustering* [Katz 03]. De notar que o nosso algoritmo está incluído na sub-categoria dos algoritmos de *region growing*. Basicamente, este consiste na decomposição da malha em regiões poligonais convexas. Cada região começa com um vértice inicial ou polígono e cresce em número de polígonos (que vão sendo agregados) até ser satisfeita uma condição pré-definida ou um critério de paragem. Isto é, a região cresce enquanto é válida uma condição pré-definida relacionada com a convexidade. A maioria dos algoritmos baseados em *region growing* têm duas etapas: *region growing* (expansão de região) e *region merging* (fusão de regiões) [Zuckerberger 02], [Kalvin 96], [Lavoué 05], [Kim 10].

1.2 Contribuições

As principais contribuições do algoritmo de segmentação de malhas proposto neste artigo são as seguintes:

- *Convexidade relaxada*. Este tipo de convexidade permite-nos formar regiões que não são estritamente convexas nem estritamente côncavas. Isto é particularmente apropriado para a segmentação de objectos de forma livre, como por exemplo, o polvo e a formiga que podemos ver na Figura 1.
- *Point membership test* (PMT). PMT é um caso particular do teste SMC (*set membership classification*), que é muito popular na modelação CSG (*constructive solid geometry*) [Tilove 80]. Este teste é neste artigo usado pela primeira vez como teste de convexidade na segmentação de malhas.
- *Função de altura com sinal*. Ao contrário de outros algoritmos que usam funções de altura sem sinal (ver, por exemplo, [Mangan 99]), nós adoptamos uma função de altura com sinal que nos permite organizar

a malha em regiões de dois tipos: montes e vales. Esta função de altura com sinal é obtida pelo sinal da curvatura dada por

$$\kappa(i) = (-1)^i K \quad (1)$$

onde K é a curvatura de Frobenius e i é índice de convexidade que toma o valor 0 para vértices convexas e 1 para vértices côncavos.

O restante artigo está organizado como se segue. A secção 2 faz a descrição do conceito de convexidade relaxada e descreve, ainda, de forma breve o algoritmo MeV. As Secções 3-9 descrevem o algoritmo passo a passo. A Secção 10 apresenta os resultados mais relevantes, em particular os resultados obtidos com o *benchmark* de Princeton. Finalmente, a Secção 11 apresenta as conclusões finais.

2 SEGMENTAÇÃO RELAXADA

2.1 Conceitos Teóricos

De acordo com a asserção de Hoffman [Hoffman 97] referida na Secção 1, a visão humana delimita as regiões do objeto ao longo dos mínimos negativos das curvaturas principais, o que implica que as regiões significativas de um objeto são convexas. Isto é ilustrado na Figura 3, onde a malha se divide em regiões convexas relaxadas; nenhuma região côncava relaxada faz parte desta malha. Em suma, ao contrário das outras técnicas de segmentação que dividem uma malha em regiões convexas, côncavas e de sela, a nossa divide uma malha em regiões convexas relaxadas e regiões côncavas relaxadas, evitando assim a sobre-segmentação da malha tanto quanto que possível.

Uma região não precisa ser estritamente convexa (respectivamente, côncava) para ser classificada como um monte (respectivamente, vale). Uma região *convexa relaxada* ou monte é uma região que é delimitada por um ciclo de arestas côncavas. Por outro lado, uma região *côncava relaxada* ou vale é uma região que é delimitada por um ciclo de arestas convexas. Isto significa que uma região convexa relaxada (respectivamente, região côncava relaxada) pode conter arestas côncavas (respectivamente, arestas convexas) no seu interior, desde que não formem ciclos. Assim, uma região convexa também é convexa relaxada, mas não vice-versa; de modo semelhante, uma região côncava é também côncava relaxada, mas não vice-versa. A convexidade relaxada é particularmente importante na segmentação de ob-

jetos de forma livre; por exemplo, as pernas do polvo descrito na Figura 1 são exemplos de regiões convexas relaxadas.

Um monte tem um pico, i.e., um ponto com curvatura positiva máxima. Um vale tem um nadir ou ponto mais baixo, i.e., um ponto com curvatura negativa mínima. Começando pelos picos e nadires, podemos construir montes e vales, respectivamente, até eles satisfazerem as suas condições de paragem.

2.2 Esboço do Algoritmo

De modo a criar uma segmentação baseada nos conceitos apresentados, foi implementado o algoritmo de segmentação MeV composto pelas seguintes etapas:

1. *Cálculo da curvatura dos vértices.* Cálculo da curvatura de Frobenius em cada vértice da malha (Ver Secção 3).
2. *Cálculo da curvatura das faces.* Cálculo da curvatura de cada face, calculando a média da curvatura de Frobenius dos seus vértices (Ver Secção 4).
3. *Classificação das arestas.* Classificar cada aresta da malha como aresta convexa, aresta côncava ou aresta plana. Esta classificação é feita usando o PMT, em vez do tradicional ângulo diedral (Ver Secção 5).
4. *Classificação das faces.* Classificar cada face da malha como face convexa ou face côncava. A convexidade de cada face é determinada pela tipo de convexidade das suas arestas (Ver Secção 6).
5. *Encontrar Picos e Nadires.* Encontrar faces de curvatura máxima (picos dos montes) e faces de curvatura mínima (nadir dos vales). Estas faces são as faces iniciais das regiões da malha a criar no próximo passo (Ver Secção 7).
6. *Expansão de Regiões.* Construir regiões a partir das faces iniciais, que são faces pico e faces nadir. De recordar que uma região da malha corresponde a um conjunto conexo de faces duma malha de triângulos (Ver Secção 8).
7. *Fusão de Regiões.* Juntar regiões adjacentes se for apropriado (Ver Secção 9).

Em resumo, o algoritmo MeV criar uma segmentação geométrica, na qual as regiões são montes e vales. Cada região termina o seu processo de expansão quando a sua fronteira (i.e., arestas que a delimitam) se torna estritamente convexa ou côncava.

3 CÁLCULO da CURVATURA dos VÉRTICES

A curvatura num vértice é calculada através da norma da matriz de covariância, das normais $\{\mathbf{N}_i\}$ das faces incidentes no vértice, onde $\mathbf{N}_i = (x_i, y_i, z_i)$ e $i = 1, \dots, n$.

Para essa finalidade, temos que determinar o vetor médio das normais, que é:

$$\bar{\mathbf{N}} = (\bar{x}, \bar{y}, \bar{z}), \text{ com } \bar{x} = \sum_{i=1}^n x_i, \bar{y} = \sum_{i=1}^n y_i, \text{ e } \bar{z} = \sum_{i=1}^n z_i.$$

A fórmula geral para calcular a covariância de duas variáveis $u \in \{x, y, z\}$ and $v \in \{x, y, z\}$ é

$$\sigma_{uv}^2 = \frac{1}{n} \sum_{i=1}^n (u_i - \bar{u})(v_i - \bar{v}) \quad (2)$$

de modo que a matriz de covariância é a seguinte:

$$C = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} \quad (3)$$

Consequentemente, a norma da matriz de covariância (isto é, matriz de Frobenius) é a seguinte:

$$K = \sqrt{\sum_{\substack{u \in \{x, y, z\} \\ v \in \{x, y, z\}}} \sigma_{uv}^2} \quad (4)$$

que representa a curvatura combinatória num dado vértice rodeado por n faces. Note que, de acordo com a Eq. (4), o valor de K é sempre maior ou igual a zero, de modo que não há possibilidade de distinguir entre um vértice convexo e um vértice côncavo, se estes tiverem a mesma curvatura. Como mostramos mais à frente, este problema resolve-se através do PMT.

4 CÁLCULO da CURVATURA das FACES

Depois de calcular a curvatura de Frobenius em cada vértice, calcula-se a curvatura de cada face como sendo a média da curvatura dos seus vértices. Mas, como vimos na secção anterior, isso não nos fornece qualquer informação sobre a convexidade de qualquer vértice, aresta, face, ou região da malha. A análise da convexidade da malha inicia-se com a análise da convexidade das suas arestas através do PMT, o que é então propagado para as suas faces, podendo-se então começar a formar regiões da malha.

5 CLASSIFICAÇÃO das ARESTAS

A análise da convexidade de cada aresta é baseada no PMT.

5.1 Classificação pelo PMT

De acordo com o critério PMT, todo o ponto 3D é classificado em relação à malha do objecto da seguinte forma: o ponto está do lado de *fora* da malha; o ponto está *sobre* a malha; o ponto está do lado de *dentro* da malha.

Classificar pontos desta forma, em relação a uma malha de triângulo, reduz-se à classificação de um ponto relativamente ao plano definido pela face da malha que lhe está mais próximo. Vamos, então, usar a equação vetorial do plano α dada pelo produto interno $\vec{N} \cdot (\vec{CP}) = 0$, onde \vec{N} é o vector normal à face de centro C e \vec{CP} representa um vector em α definido por C e qualquer outro ponto P pertencente a α . Assumindo que $N = (a, b, c)$, $C = (x_0, y_0, z_0)$

e $P = (x, y, z)$, obtemos facilmente a equação Cartesiana do plano como $\alpha(x, y, z) : ax + by + cz + d = 0$, onde $d = (-ax_0 - by_0 - cz_0)$. É claro que o plano $\vec{N} \cdot (\vec{CP}) = 0$ divide \mathbb{R}^3 em dois semi-espacos, i.e., semi-espaco positivo $\vec{N} \cdot (P - C) > 0$ e o semi-espaco negativo $\vec{N} \cdot (P - C) < 0$. A normal \vec{N} aponta para o semi-espaco positivo (i.e., para fora), de modo que podemos facilmente testar se um ponto está dentro ou fora da malha usando um único triângulo nas proximidades.

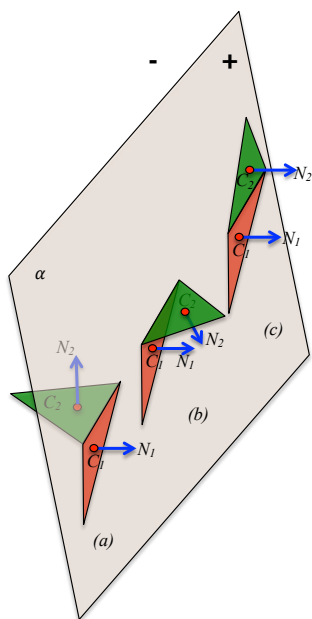


Figura 2. Análise da convexidade de duas faces vizinhas (vermelha e verde) baseada no PMT.

5.2 Análise da convexidade das aresta via PMT

O critério PMT pode ser usado para avaliar a convexidade de uma aresta, que é partilhada pelas duas faces nela incidentes (Figura 2). Seja C_1 e \vec{N}_1 o centro e vector normal da primeira face, e C_2 e \vec{N}_2 o centro e vector normal da segunda face. Então, nós temos:

- Se $\vec{N}_1 \cdot (C_2 - C_1) < 0$, a segunda face situa-se no lado negativo do plano α , assim sendo, a aresta partilhada pelas faces é definida como *convexa* (Figura 2(a));
- Se $\vec{N}_1 \cdot (C_2 - C_1) > 0$, a segunda face situa-se do lado positivo do plano α , assim sendo, a aresta partilhada pelas faces é definida como *côncava* (Figura 2(b)).
- Se $\vec{N}_1 \cdot (C_2 - C_1) = 0$, então as faces são co-planares (Figura 2(c)), e a aresta partilhada é definida como *plana*;

Note-se que este critério pode ser usado para saber se uma aresta é ou não convexa, sem calcular o ângulo diedral entre os triângulos incidentes na aresta. Além disso, o PMT

não sofre da ambiguidade do ângulo diedral na avaliação da convexidade das arestas. Após a realização de alguns testes, observou-se que o critério PMT é mais rápido do que o critério de ângulo de diedral, pois gasta menos de 50 por cento do tempo.

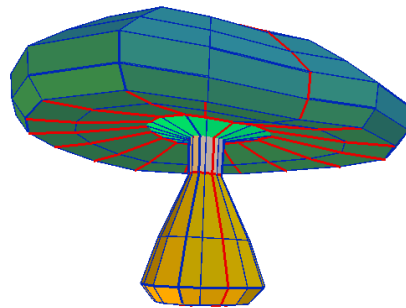


Figura 3. Cogumelo segmentado usando o algoritmo MeV, com as arestas convexas e côncavas representadas

6 CLASSIFICAÇÃO das FACES

Antes de iniciar a formação das regiões da malha, temos que avaliar e classificar cada face em relação à convexidade. Para esse efeito, adotamos a seguinte classificação para as faces:

- *Face convexa.* Uma face é *convexa* se o seu número de arestas convexas é maior ou igual ao número de arestas côncavas. Neste caso, tendo em consideração que o descritor de convexidade i igual a zero para faces convexas, obtemos o sinal da curvatura $\kappa(0) = (-1)^0 K = K$, onde K representa a média da curvatura de Frobenius da face.
- *Face côncava.* A face é *côncava* se o seu número de arestas côncavas é maior do que o número de arestas convexas. Neste caso, o descritor de convexidade i igual a 1, de modo que o sinal da curvatura $\kappa(1) = (-1)^1 K = -K$.
- *Face plana.* A face é *plana* se todas as arestas delimitadoras são planas. Neste caso, o sinal da curvatura $\kappa(i) = 0$ porque $K = 0$, não importa o valor de seu descritor de convexidade i .

7 ENCONTRAR PICOS e NADIRES

Nesta etapa o objetivo é encontrar os picos dos montes e os nadires dos vales da malha de triângulos.

Intuitivamente, os picos são vértices convexas da malha, como por exemplo, os cantos convexas de um cubo. Por outro lado, os pontos de nadir são vértices côncavos da malha. Pontos planares são pontos localizados em zonas planas, isto é, pontos de curvatura nula.

No entanto, estamos interessados em encontrar faces de pico e faces de nadir. Uma face de pico é uma face de

curvatura máxima positiva, isto é, uma face rodeada por faces com menor ou igual curvatura; uma face de nadir é uma face de curvatura mínima negativa, isto é, uma face rodeada por faces com maior ou igual curvatura. As faces iniciais dos montes são especificamente faces de pico, enquanto para os vales são as faces de nadir. Faces de pico e de nadir são agrupadas em arrays separados, para serem facilmente seleccionados no etapa seguinte.

8 EXPANSÃO DE REGIÕES

A etapa de expansão de regiões do algoritmo baseia-se no conceito de convexidade relaxada. Seguindo a asserção de Hoffman, o nosso algoritmo de segmentação cria em primeiro lugar todas as regiões convexas relaxadas; depois, o algoritmo constrói as restantes regiões côncavas relaxadas.

Algoritmo 1: Construir Montes da Malha

Verificar : F : array de faces *seed* para os montes

Verificar : H : array vazio de montes

- 1: Ordenar F por curvatura decrescente
 - 2: $n \leftarrow$ tamanho de F
 - 3: **para** $i \leftarrow 0$ até $n - 1$ **fazer**
 - 4: **se** $F[i] \notin$ a uma região já criada da malha **então**
 - 5: Criar novo monte m
 - 6: $m \leftarrow m \cup F[i]$
 - 7: **enquanto** \exists aresta não-côncava a delimita m
 fazer
 - 8: $f \leftarrow$ face adjacente a a
 - 9: $m \leftarrow m \cup f$
 - 10: **fim enquanto**
 - 11: $H \leftarrow H \cup m$
 - 12: **fim se**
 - 13: **fim para**
-

O procedimento para criar as regiões convexas relaxadas (ou montes) de uma malha é descrito no Algoritmo 1. A construção de um monte começa com uma face inicial convexa (cf. linhas 4 a 12 no Algoritmo1). Com o objectivo de expandir a região, examinam-se as faces vizinhas para verificar se podem ser adicionadas ao (cf. linha 7) monte, repetindo-se então o processo de crescimento iterativo para cada face adicionada anteriormente. Somente faces planas e convexas podem ser adicionadas à região corrente. A condição de paragem verifica-se quando toda a fronteira do monte é formada só por arestas côncavas, como expresso pela condição do ciclo na linha 7. No Algoritmo 1, supomos que temos n montes porque temos no início N faces iniciais. Mas, se alguma face inicial é inserida num outro monte durante a formação deste, a condição na linha 4 não será satisfeita, por isso, termina tendo um número de montes que é menor que n . Por exemplo, todas as faces de um pico em torno dos vértices de um cubo irão ser incluídas num único monte, que compreende toda a malha que cobre o cubo.

Da mesma forma, o processo para formar as regiões côncavas relaxadas (ou vales) de uma malha está descrito no Algoritmo 2. Construir um vale começa com uma face

de nadir. A região do vale cresce a partir de uma face côncava inicial adicionando a seguir faces vizinhas que são côncavas ou planas. O crescimento de um vale pára quando a sua fronteira é formada só por arestas convexas (cf. linha 7 no Algoritmo 2).

Algoritmo 2: Construir Vales da Malha

Verificar : F : array de faces *seed* para os vales

Verificar : V : array vazio de vales

- 1: Ordenar F por curvatura decrescente
 - 2: $n \leftarrow$ tamanho de F
 - 3: **para** $i \leftarrow 0$ até $n - 1$ **fazer**
 - 4: **se** $F[i] \notin$ a uma região já criada da malha **então**
 - 5: Criar um novo vale v
 - 6: $v \leftarrow v \cup F[i]$
 - 7: **enquanto** \exists aresta não-convexa a delimita v
 fazer
 - 8: $f \leftarrow$ face adjacente a a
 - 9: $v \leftarrow v \cup f$
 - 10: **fim enquanto**
 - 11: $V \leftarrow V \cup v$
 - 12: **fim se**
 - 13: **fim para**
-

Recorde-se que os montes são construídos antes dos vales, o que está em conformidade com a afirmação de Hoffman [Hoffman 97], isto é, de acordo com o modo como os seres humanos percebem as formas do mundo envolvente. Em termos de algoritmos, isto traduz-se no fato de que qualquer região ao ser expandida vai preservando a sua convexidade relaxada. Isto significa que uma região convexa relaxada pode compreender arestas côncavas, desde que essas arestas da malha não formem um ciclo. Isso vale também para a região côncava relaxada de uma malha que possuem arestas convexas.

Além disso, a condição que paragem da formação de uma região da malha é bastante importante no nosso algoritmo pois termina a formação de um monte (e também um vale) no ciclo de inflexão que separa o monte (e também um vale) das outras regiões da malha. O ciclo de inflexão que delimita um monte é definido por arestas côncavas, enquanto o que delimita um vale apenas compreende arestas convexas. Uma consequência subtil das condições de paragem para os montes e vales é que o crescimento da região pára automaticamente, sendo o suficiente para efectuar a segmentação da superfície de objectos de forma não livre (por exemplo, peças mecânicas na parte superior da Figura 1); nenhuma etapa de fusão de regiões é necessária a não ser para objetos de forma livre. O mesmo se passa para alguns objectos de forma livre mais simples que também não necessitam do passo suplementar de fusão de regiões para alcançar as segmentações finais, como é o caso do cogumelo apresentado na Figura 3.

9 FUSÃO DE REGIÕES

A fusão de regiões é uma etapa essencialmente necessária para objetos de forma livre como, por exemplo, os seres humanos e animais. Estes objetos de forma livre

são os objetos mais difíceis de segmentar no processo de segmentação do nosso algoritmo (e na maioria dos outros métodos) porque eles apresentam muitas variações na curvatura, o que provoca o aparecimento de *sobre-segmentação* na malha. Em grande parte, os altos e baixos dos objetos de forma livre levou-nos a chegar à noção de geometria relaxada, que está no cerne do nosso algoritmo. Assim, a fusão de regiões visa reduzir o excesso de segmentação da malha.

Na etapa de fusão de regiões, decidimos usar o algoritmo proposto por Chen e Georganas [Chen 06]. A ideia principal deste algoritmo é reduzir o número de pequenas regiões (segmentos), através da junção das regiões mais pequenas com regiões maiores, que lhes são adjacentes. Basicamente, uma pequena região é fundida com a região adjacente com o qual compartilha a parte mais longa da sua fronteira.

Observa-se que a *sobre-segmentação* também pode fazer a sua aparição em objetos de forma não livre (por exemplo, peças mecânicas), isto é, com as regiões de malhas com faces co-planares. Isso acontece como consequência dos erros de arredondamento em cálculos aritméticos com vírgula flutuante. A fim de evitar este problema, é necessário usar uma tolerância para garantir a co-planaridade de triângulos adjacentes, mas esta operação é feita na etapa de anterior.

10 RESULTADOS EXPERIMENTAIS

10.1 Configuração de Testes

O nosso algoritmo de segmentação foi projetado e implementado num computador com um processador 2.4 Intel Core Duo e um sistema operativo Mac OS X, usando o OpenGL e a *User Interface Library* (GLUI), que é uma biblioteca em C++ baseada no *OpenGL Utility Toolkit* (GLUT).

10.2 Aferição (Benchmarking)

Utilizou-se o *Princeton benchmarking* [Chen 09] para segmentações, com o objectivo de comparar o nosso algoritmo com o estado-da-arte dos algoritmos de segmentação de malhas. Esta ferramenta de *benchmarking* fornece 19 categorias de modelos, em que cada categoria é composta por 20 objectos. Esta ferramenta também fornece 4300 segmentações criadas por seres humanos que servem de referência para a avaliação dos métodos, sendo fornecidos para cada objecto, em média, 11 segmentações criadas por seres humanos.

O *Princeton benchmarking* disponibiliza comparações quantitativas das segmentações geradas pelos seres humanos e das segmentações geradas por computador produzidos pelos sete algoritmos seguintes: *k-means* [Shlafman 02], *random walks* [Lai 08], *fitting primitives* [Attene 06b], *normalized cuts* [Golovinskiy 08] *randomized cuts* [Golovinskiy 08], *core extraction* [Katz 05] e *shape diameter function* [Shapira 08]. De notar que o software de *benchmarking* não inclui os códigos destes sete algoritmos, mas apenas segmentações produzidos por eles,

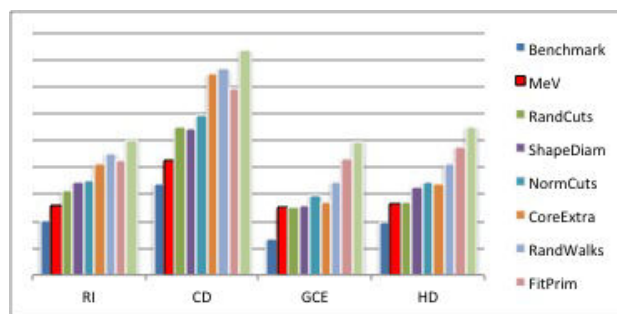


Figura 4. Resultados das métricas.

que funcionam como dados de entrada para o software de *benchmarking*. Portanto, antes de executar a avaliação dos oito algoritmos (sete do *benchmarking* mais o nosso), adicionamos as segmentações geradas pelo nosso algoritmo ao *benchmark*.

Para comparar as segmentações, a ferramenta de *benchmarking* usa as seguintes quatro métricas:

- *Rand Index* (RI). Mede a probabilidade de dois triângulos pertencerem à mesma ou a diferentes segmentações.
- *Cut Discrepancy* (CD). Intuitivamente, é um método baseado na fronteira e que mede as distâncias entre os cortes definidos pelas fronteiras das regiões (do método e os criados pelo ser humano), de modo que quanto menores forem estas distâncias, melhor será a segmentação gerada pelo método.
- *Consistency Error* (CE). Esta métrica tenta determinar semelhanças e diferenças hierárquicas nas segmentações.
- *Hamming Distances* (HD). Mede as diferenças globais entre regiões de duas segmentações. A principal vantagem destas métricas é que elas permitem encontrar correspondências entre regiões produzidas por um algoritmo de segmentação e as segmentações criadas pelos seres humanos.

Em resumo, uma das métricas concentra-se em erros de fronteira (*Cut Discrepancy*), e as outras três focam-se em diferenças de região (*Hamming Distance*, *Rand Index*, e *Consistency Error*).

A Figura 4 mostra os resultados do nosso algoritmo em comparação com os outros sete algoritmos de *benchmarking* usando as quatro métricas referidas anteriormente. Note-se que as métricas são consistentes umas com as outras, no sentido de que os algoritmos de segmentação têm o mesmo desempenho relativamente a essas métricas. Como pode ser observado, o algoritmo MeV tem a melhor *média* de desempenho para todas as métricas.

Numa comparação mais detalhada dos diferentes algoritmos de segmentações, usámos a métrica RI, que mede o grau de correspondência da forma da região entre duas

segmentações do mesmo modelo (normalmente uma delas foi criada por seres humanos). A Tabela 1 mostra os resultados comparativos da segmentação para cada categoria de objetos, onde as entradas da tabela apresentam os rankings destes oito algoritmos de acordo com a métrica de RI, em que 1 é o melhor e 8 é o pior. Os resultados mostram que o nosso algoritmo está classificado como o melhor na métrica RI em relação às segmentações criadas pelos seres humanos, uma posição que ele mantém para quase todas as categorias de objetos.

O software de *benchmark* também nos permitiu visualizar um conjunto de imagens resultantes da segmentação MeV para um conjunto de 380 modelos, alguns dos quais se mostram na Figura 1.

Tabela 1. Resultados do Rand Index.

Rand Index	Object Category															Average				
	Human	Cup	Glasses	Airplane	Ant	Chair	Table	Teddy	Hand	Plier	Fish	Bird	Armadillo	Bust	Mech		Bearing	Vase	FourLeg	
MeV	1	1	2	2	3	3	1	1	1	2	5	1	1	5	2	1	3	1	2	1
RandCuts	2	2	1	3	2	6	5	7	2	1	2	4	2	1	1	6	2	2	3	2
ShapeDiam	4	6	5	1	1	2	2	4	3	6	8	2	3	2	3	3	1	4	1	3
NormCuts	5	3	3	5	4	1	4	2	5	4	4	5	7	7	5	2	4	5	5	4
CoreExtra	7	4	7	7	5	5	3	6	4	3	1	3	4	8	7	7	8	3	6	5
RandWalks	8	5	8	8	6	4	6	3	6	8	6	6	8	4	6	4	7	6	8	7
FitPrim	3	7	6	4	7	8	8	5	7	7	3	8	6	3	4	5	5	7	4	6
KMeans	6	8	4	6	8	7	7	8	8	5	7	7	5	6	8	8	6	8	7	8

11 CONCLUSÕES

Neste artigo apresenta-se um novo algoritmo de segmentação de malhas triangulares, designado por MeV, cuja ideia principal é encontrar, em primeiro lugar, os pontos extremos da malha onde a curvatura atinge um valor máximo local ou um mínimo local, os quais denunciam a presença de montes e vales respectivamente. Aliás, as correspondentes faces de pico e faces de nadir são as faces iniciais de expansão de montes e vales respectivamente.

Note-se que é suficiente usar só o passo de expansão de regiões para segmentar corretamente objetos de forma não-livre, enquanto objetos de forma livre, em alguns casos, precisam da etapa extra de fusão de regiões para mitigar o problema de sobre-segmentação que possa acontecer. No entanto, em muitos dos casos, o excesso de segmentação dos objetos de forma livre já é evitado pela aplicação do conceito de convexidade relaxada na construção das regiões.

No futuro, mesmo tendo em conta que o algoritmo MeV produz segmentações mais próximas da "ground-truth" do que os atuais algoritmos do estado da arte, pretende-se aumentar ainda mais o desempenho em relação a esses algoritmos quer em termos de complexidade quer em relação à diminuição do ruído.

Referências

[Attene 06a] M. Attene, S. Katz, M. Mortara, G. Patane, M. Spagnuolo, e A. Tal. Mesh segmentation — a comparative study.

Em *Proceedings of the IEEE International Conference on Shape Modeling and Applications (SMI'2006)*, páginas 7–19. IEEE Computer Society, 2006.

[Attene 06b] Marco Attene, Bianca Falcidieno, e Michela Spagnuolo. Hierarchical mesh segmentation based on fitting primitives. *The Visual Computer*, 22:181–193, 2006.

[Bajaj 92] C. Bajaj e T. Dey. Convex decomposition of polyhedra and robustness. *SIAM Journal on Computing*, 21(2):339–364, 1992.

[Biederman 87] Irving Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–147, 1987.

[Capell 02] Steve Capell, Seth Green, Brian Curless, Tom Duchamp, e Zoran Popovic. Interactive skeleton-driven dynamic deformations. *ACM Transactions on Graphics*, 21(3):586–593, 2002.

[Chazelle 81] Bernard M. Chazelle. Convex decompositions of polyhedra. Em *Proceedings of the 13th Annual ACM Symposium on Theory of Computing*, STOC '81, páginas 70–79, New York, NY, USA, 1981. ACM.

[Chazelle 95] Bernard Chazelle, David P. Dobkin, Nadia Shouraboura, e Ayellet Tal. Strategies for polyhedral surface decomposition: an experimental study. Em *Proceedings of the 11th Annual Symposium on Computational Geometry*, SCG '95, páginas 297–305, New York, NY, USA, 1995. ACM Press.

[Chen 06] Lijun Chen e Nicolas D. Georganas. An efficient and robust algorithm for 3d mesh segmentation. *Multimedia Tools and Applications*, 29(2):109–125, 2006.

[Chen 09] Xiaobai Chen, Aleksey Golovinskiy, e Thomas Funkhouser. A benchmark for 3D mesh segmentation. *ACM Transactions on Graphics*, 28(3):73:1–73:12, August 2009.

[Chung 96] Fan Chung. *Spectral Graph Theory*. CBMS Regional Conference Series in Mathematics. American Mathematical Society, 1996.

[Garland 01] Michael Garland, Andrew Willmott, e Paul S. Heckbert. Hierarchical face

- clustering on polygonal surfaces. Em *Proceedings of the 2001 Symposium on Interactive 3D Graphics*, I3D '01, páginas 49–58, New York, NY, USA, 2001. ACM.
- [Ghosh 13] Mukulika Ghosh, Nancy M. Amato, Yanyan Lu, e Jyh-Ming Lien. Fast approximate convex decomposition using relative concavity. *Computer-Aided Design*, 45(2):494 – 504, 2013.
- [Golovinskiy 08] Aleksey Golovinskiy e Thomas Funkhouser. Randomized cuts for 3D mesh analysis. *ACM Transactions on Graphics*, 27(5):145:1–145:12, Dezembro 2008.
- [Hoffman 97] Donald D. Hoffman e Manish Singh. Saliency of visual parts. *Cognition*, 63(1):29 – 78, 1997.
- [Kalvin 96] Alan D. Kalvin e Russell H. Taylor. Superfaces: Polygonal mesh simplification with bounded error. *IEEE Computer Graphics & Applications*, 16:64–77, May 1996.
- [Katz 03] Sagi Katz e Ayellet Tal. Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Transactions on Graphics*, 22(3):954–961, 2003.
- [Katz 05] Sagi Katz, George Leifman, e Ayellet Tal. Mesh segmentation using feature point and core extraction. *The Visual Computer*, 21:649–658, 2005.
- [Kim 10] Hyoungseok B. Kim e Hosook Kim. Mesh segmentation based on local geometric properties. *International Journal of Computer Science and Network Security*, 10(1), 2010.
- [Lai 08] Yu-Kun Lai, Shi-Min Hu, Ralph R. Martin, e Paul L. Rosin. Fast mesh segmentation using random walks. Em *Proceedings of the 2008 ACM Symposium on Solid and Physical Modeling*, SPM '08, páginas 183–191, New York, NY, USA, 2008. ACM.
- [Lavoué 05] Guillaume Lavoué, Florent Dupont, e Atilla Baskurt. A new cad mesh segmentation method, based on curvature tensor analysis. *Computer Aided Design*, 37:975–987, September 2005.
- [Li 01] Xuetao Li, Tong Wing Woon, Tiow Seng Tan, e Zhiyong Huang. Decomposing polygon meshes for interactive applications. Em *Proceedings of the 2001 Symposium on Interactive 3D Graphics*, I3D '01, páginas 35–42, New York, NY, USA, 2001. ACM.
- [Lien 04] Jyh-Ming Lien e Nancy M. Amato. Approximate convex decomposition of polyhedra. Em *ACM SIGGRAPH 2004 Posters*, SIGGRAPH '04, New York, NY, USA, 2004. ACM.
- [Lloyd 82] S. Lloyd. Least square quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.
- [Mangan 99] Alan Mangan e Ross Whitaker. Partitioning 3D surface meshes using watershed segmentation. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):308–321, October 1999.
- [Reuter 09] Martin Reuter, Silvia Biasotti, Daniela Giorgi, Giuseppe Patanè, e Michela Spagnuolo. Discrete laplace-beltrami operators for shape analysis and segmentation. *Computers and Graphics*, 33:381–390, June 2009.
- [Shapira 08] L. Shapira, A. Shamir, e D. Cohen-Or. Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer*, 24(4):249–259, 2008.
- [Shlafman 02] Shymon Shlafman, Ayellet Tal, e Sagi Katz. Metamorphosis of polyhedral surfaces using decomposition. *Computer Graphics Forum*, 21(3):219–228, 2002.
- [Tilove 80] R.B. Tilove. Set membership classification: a unified approach to geometric intersection problems. *IEEE Transactions on Computers*, C-29(10):874–883, October 1980.
- [Zhang 07] Hao Zhang, Oliver van Kaick, e Ramsay Dyer. Spectral methods for mesh processing and analysis, 2007.
- [Zuckerberger 02] Emanoil Zuckerberger, Ayellet Tal, e Shymon Shlafman. Polyhedral surface decomposition with applications. *Computers & Graphics*, 26(5):733–743, 2002.

Sessão - Aplicações

Galactica, a Digital Planetarium that explores the Solar System and the Milky Way

Jorge d'Alpuim
ISCTE – Instituto Universitário de Lisboa
ISTAR-IUL, Lisbon
jorge_alpuim@iscte.pt

Miguel Sales Dias
Microsoft Language Development Center, Lisbon
ISCTE – Instituto Universitário de Lisboa
ISTAR-IUL, Lisbon
miguel.dias@microsoft.com

Abstract

This paper describes a new Digital Planetarium system that allows interactive visualization of astrophysical data and phenomena in an immersive virtual reality (VR) setting. Taking advantage of the Cave Hollowspace at Lousal infrastructure, we have created a large-scale immersive VR experience, by adopting its Openscenegraph (OSG) based VR middleware, as a basis for our development. Since our goal was to create an underlying system that could scale to arbitrary large astrophysical datasets, we have splitted our architecture in offline and runtime subsystems, where the former is responsible for parsing the available data sources into a SQL database, which will then be used by the runtime system to generate the entire VR scene graph environment, for the interactive user experience. Real-time computer graphics requirements lead us to adopt some visualization optimization techniques, namely, GPU calculation of textured billboards representing stars, view-frustum culling with octree organization of scene objects and object occlusion culling, to keep the user experience within the interactivity limits. We have built a storyboard (the “Galactica” storyboard), which describes and narrates a visual and aural user experience, while navigating through the Solar System and the Milky Way, and which was used to measure and evaluate the performance of our visualization acceleration algorithms. The system was tested with an available dataset of the complete Milky Way (including the solar system), featuring 100.639 textured billboards representing stars and additional 104.328 polygons, representing constellations and planets of the solar system. We have computed the frame rate, GPU traverse time, Cull traverse time and Draw traverse time for three visualization conditions: (A) using standard OSG view frustum culling technique; (B) using view frustum culling with and our octree organizing the scene’s objects; (C) using view frustum culling with our octree organizing the scene’s objects and our occlusion culling algorithm. We have generally concluded that our octree organization and octree plus object culling techniques out-performs the standard OSG view frustum culling, when around half or less than half of the dataset is in view of the virtual camera.

Keywords

Virtual reality, CAVE, Lousal, Immersive Virtual reality, octree, occlusion culling, astrophysical, Solar System, Milky Way, Hipparcos, Openscenegraph

1. INTRODUCTION

If we approach Immersive Virtual Reality (VR) from the perspective of its requirements for real-time 3D Computer Graphics, we certainly conclude that we need to put in place in our VR system, visualization acceleration techniques in order to provide the user with real-time interactive experiences, within an environment capable of depicting photorealistic scenes in stereoscopy, in higher resolution multi-display projection systems, such as the one available at the CAVE Hollowspace of Lousal (CaveH) [Soares10], in southern Portugal (near the city of Grândola) and used in this work. There are in fact available a quite wide range of different approaches to accelerate the processing of large and complex datasets, depending on the kind of dataset we are willing to represent. In this

paper, we are especially interested in visualizing large datasets of astrophysical celestial objects, such as the ones included in the Digital Universe Atlas (DUA), assembled by the American Museum of Natural History (AMNH) [Abbott02]. This catalogue features a high-precision dataset with 100.639 stars, which was sensed by the Hipparcos scientific satellite of the European Space Agency (ESA), launched in 1989 and operated until 1993. Hipparcos, was targeted to precision astrometry and provided the community with accurate measurement of the positions of celestial objects. DUA is a multi-dimensional atlas intended to represent the known universe and used as a basis for further data analysis and exploration. In this framework, we present in this paper, a new interactive Digital Planetarium tool for the CaveH at Lousal, able to paint with the opportunity to experience the visualization and interactive navigation across arbitrary large astrophysical

datasets, such as the DUA. Displaying an arbitrary large dataset of rich 3D content in an Immersive VR setting with high resolution and real-time stereoscopic visualization requirements, relying solely on the Core CPU power, without any specific 3D acceleration technique in mind, would naturally compromise the user experience. We could easily break the real-time interactivity performance, since the time taken to process the entire scene complexity would become noticeable. Visualization acceleration techniques are meant to help decreasing the CPU effort to process and draw the geometry in each frame, by using appropriate scene organization techniques, real-time visualization technique and leveraging the power of the available GPU Cores (in the order of hundreds) to unload the burden of the few existing CPU Cores, in the computing system. In this context, the identified real-time 3D Computer Graphics requirements, namely, a scene graph to describe the 3D environment, a volumetric organization technique of the scene, an object culling approach and GPU rendering of most of the celestial objects, were matched by the adoption of the Openscenegraph (OSG) [Osfield05] C++ development environment, specially targeted to the processing and visualization of large and complex 3D environments (in terms of polygon count), which has already some built-in acceleration techniques. In addition to OSG, we have used the CaveH middleware [Dias07] [Soares10], an in-house developed logic to produce and manage 3D content in the multi-display Cave Automatic Virtual Environment (CAVE) at Lousal, an immersive large-scale VR system much like the one proposed by [Cruz-Neira92] in 1992. Since the CaveH supports projection in many different computing architectures (from tablet and laptop displays to large-scale multi-projection planes) and it is based on the OSG graphics platform, which is a layer above the OpenGL library, we adopted the same development environment in order to avoid later unexpected compatibility issues. This paper presents also the results obtained with the Galactica application that provides the user with an experience of a space travel across the Solar System and its surrounding galaxy (Milky Way). We provide a comparison between the OSG standard pipeline and our customized 3D rendering pipeline, with our visualization acceleration techniques embedded in the OSG rendering traversal cycle, specifically developed to support the requirements of Galactica.

The paper is structured as follows. In the section 2, we describe in detail the datasets used by Galactica. The third section details the underlying system architecture, responsible for managing, simulating and running the application. Section 4 explains the overall VR simulation and visualization pipeline, by showing how the data is represented and explaining in detail the real-time 3D algorithms developed in order to achieve real-time performance. Section 5 presents and discusses the results on the measured performance metrics obtained by simulating the Galactica application using the different acceleration algorithms. In Section 6 we extract some conclusions and provide some ideas for further research. We also acknowledge our thanks to all persons that directly or indirectly helped achieving the Galactica application results.

2. THE DIGITAL UNIVERSE ATLAS DATASET

As mentioned before we used the DUA dataset, loading only the data of the Milky Way subset (corresponding to the Hipparcos sensed data). We were willing to provide an application to navigate through the Solar System's astrophysical data and phenomena and the Milky Way. Since the DUA lacks on specific Solar System data, we gathered the complementary data from another publicly available dataset contained in the Celestia [Laurel01] software package. As a result, the data depicted in the Galactica application comprise planets, stars and polylines, this last ones describing the constellations and the planetary orbits. Using the above mentioned data sources we were able to paint the following objects:

Planets: The Solar System's planets are drawn as spheres with their surface textures mapped into it, simulating an approximation of the real planet's shapes. The used surface textures were taken from [Hastings-Trew00], an online resource providing free high-resolution images to the scientific community. The planets are represented by their appearance, dimensions, orbital lines and rotation's axis. The planet's orbits are also present in our VR simulation and will be drawn by polylines, with a color assigned, as a sequence of points around the Sun, forming the elliptical planet's orbits.

Stars: Representing the stars is a complex task due to the fact that, when a star is quite distant from the viewer, we only perceive it as a shiny small point in the sky and this effect is computationally difficult to reproduce. Therefore, our stars' representation, are an approximation of the visual effect our eye can perceive, which is a point with a colored halo around. That being said, we decided to use a texture of a halo provided by the DUA (Figure 1), transform it into a halo transparency texture and map it into a colored surface. The corresponding surface is a quadrangle 3D sprite, simulating the billboard effect, in order to ensure that the star's halo is always facing the virtual camera that represents the viewer pose (position and orientation).

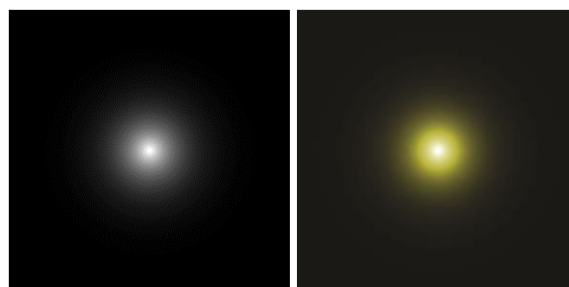


Figure 1: [Left]: Original halo texture given by DUA. [Right]: Simulation of a yellow star halo effect by mapping the halo transparent texture into a yellow surface.

The left image present in Figure 1 is a full opaque image of a white halo, which was the basis to generate the transparent mask image created to allow representing colored hallos (Figure 1 – [Right]), where its black pixels represents the transparent pixels.



Figure 2: Landscape taken from our VR application depicting the Scorpius constellation, one of the Zodiac's constellations.

Drawing polylines joining the stars' position will generate the stars' constellations effect, as it can be observed on Figure 2. Since the constellations are 3D objects drawn in the VR scene, those constellations are only realistically represented when they are observed from a perspective closer to the Earth's surface. Other way, the constellations will be "deformed" as perceived by the viewer, once the viewer's relative pose to the Earth is different.

3. VR SYSTEM ARCHITECTURE

The system built to simulate the required VR scene is divided in two: offline and runtime subsystems (Figure 3). The offline subsystem is responsible for selecting and parsing available astrophysical data and metadata, both from local and/or remote sources, into a relational database that will be read by the runtime subsystem. The runtime subsystem, builds the VR scene based on the data stored in the relational database, which was previously populated by the offline subsystem.

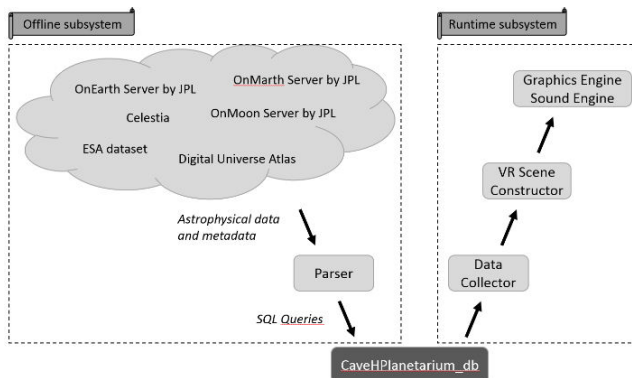


Figure 3: Overall VR system architecture, described with two subsystems.

This logical split made on the overall system brings scalability advantages to the application due to its ability to visually simulate different datasets of arbitrary sizes.

4. VR SIMULATION AND VISUALIZATION PIPELINE

As it was stated before, our 3D development environment was based on CaveH, which is a C++ middleware that builds on the OSG scene graph and 3D rendering SDK. An intrinsic feature regarding OSG, is that it will organize all scene objects in an acyclic graph, in order to efficiently traverse all scene's objects in each 3D visualization pipeline traversal, comprising 4 traversal passes per frame: an event, an update, a cull and a draw pass in each frame

[Wang10]. This order should be respected in order to keep the data consistency and to prevent unexpected behaviors. The event traversal, is the one responsible for collecting user inputs, such as keyboard and mouse operations, which is extremely important in an interactive application, due to its ability to provide the application, the possibility of reacting to user inputs. The update traversal, will inspect all the scene's objects and will transform them as a callback response to the user's inputs and/or by their own geometric transformation parents, in case of dynamic objects. Cull traversal, will filter which objects are seen/not seen by a virtual camera and therefore, which ones should be processed and drawn, by different metrics, such as their visibility, their importance in final image, their translucency, etc, preventing irrelevant objects to keep being processed in further traversals. Object that are elicited to be drawn in the current frame, are added to a "Draw List". The last traversal, the draw traversal, will process objects in the internal "Draw List", to be sent to the display, directly creating the OpenGL calls that will create the rendered scene. The DUA's Milky Way subset, our main testing dataset, comprises 100.639 stars which will be represented by textured billboards, each one with unique characteristics (taken from the dataset metadata), hence it may lead to performance issues if we do not take any pipeline precaution, limiting the amount of data traversed and drawn, in every frame. As stated by [Akenine-Möller08] "Acceleration algorithms will always be needed" in any VR application. Our visualization acceleration tackled, in particular, the object's culling task. As a conclusion, our algorithms will operate during the pipeline's culling traversal.

OSG already provides by default some culling algorithms such as the "small feature culling", which clips away the objects that will contribute with only few pixels in the rendered image. Although, since we want to draw stars in the night sky we need to draw even the smaller objects, and so we disabled this small feature culling approach. Another important culling approach already built-in the OSG pipeline is the standard "view frustum culling" algorithm, which clips away all the objects not seen by the virtual camera, by intersecting each object's bounding volume against the virtual camera's view frustum (that is, the truncated visualization pyramid). We will be using it in our approach because it will prevent all the invisible objects from being drawn, reducing the number of OpenGL calls which would reduce the time taken to draw the scene. In our application we adopted the standard octree approach, as our spatial organization data structure. This technique divides the 3D space in a regular way, which allows rapid building of the entire tree and provides an efficient and fast way to search for all the visible objects, stored in such octree data structure. Another real-time acceleration algorithm we found useful to include in our approach is the occlusion culling algorithm, which clips away the occluded objects, this is, the algorithm will test if any object inside the view frustum is hidden behind another object or a set of objects and, in that case, will also clip it away.

4.1 Data representation

Choosing an optimized way to represent the data is an important part of an efficient real-time VR application. As it

was stated previously, we represented each star as textured billboards and the planets as textured spheres. Drawing each star as an independent billboard, using the default billboard’s OSG object, implies high computational demand, breaking the application’s interactivity (see Table 1).

OSG billboard structure		
# Objects	# GPU calls	Frame rate (fps)
11	11	~ 195
100	100	~ 189
1 009	1 009	~ 150
10 221	10 221	~ 52
100 629	100 629	~ 8

Table 1: Preliminary performance results for a variable number of objects drawn in the scene, using standard OSG Billboard objects with increasing number of objects, running in mono projection mode.

Therefore, we adopted a different approach to simulate the billboard effect of each star, based on direct GPU drawing of such objects, by the mean of OpenGL’s shading language programming, the GLSL. Our approach allows to draw all the stars in a single GPU call, while changing each star object pose to face the camera every frame. In order to keep each star facing the camera, while drawing each vertex of each star’s billboard (a quadrangle), we need to rotate such quadrangle by its corresponding star’s pivot point.

	# Objects	# GPU calls	FPS Mono	FPS Stereo
OSG	3.934	3.934	~ 97	~ 27
OSG	100.629	100.629	~ 4	~ 2
GPU	3.934	1	~ 855	~ 462
GPU	100.629	1	~ 743	~ 396

Table 2 – Comparative table depicting the system’s performance drawing all the stars by the two considered approaches (OSG or GPU Billboards), both in mono and stereo projection modes.

The vertex shader needs to receive the pivot points list because while drawing each star’s billboard vertex, it will need to rotate according to its position regarding the corresponding pivot point.

As it may be observed from Table 2, drawing the stars with our GPU approach generates lower GPU calls and therefore higher performance, even when viewing all the objects present in the VR scene, which led us to realize that this was the best approach to create the star’s billboard effect.

4.2 Octree spatial organization

In complex scenes with large number of objects to be processed, testing all object’s visibility is a great struggle to the rendering pipeline, introducing a big delay in cull and following traversal phases. Organizing and sorting those objects in a spatial organization data structure is a common

approach to reduce the number of visibility tests, boosting the time taken to process the entire scene. These data structures organize the scene objects in any N-dimensional space and typically are hierarchical, meaning that each level of the organization encompasses an arbitrary number of smaller levels which, during traversing time of the entire scene, can accelerate the quest of finding all the visible objects.

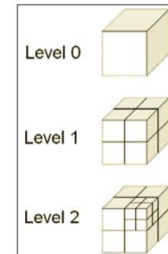


Figure 4: Octree sub-nodes division example [Wang10].

Even though these data structures may be dynamic, updating themselves while the contained objects change their state (position, size, orientation, etc...), would be time-consuming, introducing bigger delays and therefore, worse performance. In our Galactica storyboard (a travel through the Solar System), we do not consider at this stage the orbital motion of the planets and its satellites. Since in our case, the objects represented in the scene do not change position and size, we decided to adopt a static octree data structure. With this approach, traversing the entire scene during the cull phase, would reduce the number of visibility tests because when a level would not pass the test it would prevent any object, inside that level, to be tested.

The octree data structure construction [Akenine-Möller08] is performed in the beginning of the runtime subsystem cycle (see Figure 3). The octree is built based on the global scene bounding box, which is the octree’s root node (see Figure 4). Starting in this node, the octree will be constructed recursively by splitting each node in two equal parts in each direction, generating eight new octants. To manage the octree’s balance and its ability to be traversed rapidly, we set two rules to decide if a new octant should be created. Those rules are the maximum octree depth level and the other is the minimum number of objects per octant. All the stars stored in an octant will be drawn as part of the same OSG geometry object (with one vertex list and one polygon list), which means we will have as much OSG geometry objects, as the number of octants holding such objects.

Max. depth	2	4	4	8	8	16
Min. obj.	128	64	128	64	128	128
Pose 1	889	941	948	1128	1160	1175
Pose 2	910	926	918	1208	1190	1202
Pose 3	909	897	906	750	798	801
Max. dif.	21	44	42	458	392	401

Table 3: Frame rate measurements along three poses with incrementing number of objects built with different octree parameters. Data was captured in mono projection mode.

To create a balanced and efficient Octree, we present (in Table 3) the results achieved for different octree construction configuration. Pose 1 depicts a simulation of around 3.000 objects, Pose 2 depicts around 12.000 objects and Pose 3 depicts, the entire scene, around 100.600 objects. The table's last row represents the maximum frame rate difference between the best and worst performance pose. We set our goal to find the configuration that would bring the most balanced object's organization, in order to traverse the entire scene, with different number of visible objects, keeping the frame rate as high and uniform as possible. We may realize, observing the Table 3, that using a maximum depth higher than 4 brings lower frame rates with higher number of visible objects and bigger frame rate variations, and using a maximum depth lower than 4 brings lower frame rate performance in pose 1 because the scene is more unbalanced than with depth 4, providing slower scene traversing while observing fewer objects, which led us to choose the maximum depth of 4.

Having a maximum depth of 4 and minimum objects per octant of 64, we achieved a better performance in pose 2 than the results measured with 128 minimum objects per octant. Since this pose (pose 2) has a large amount of objects (but not the entire scene), which is the most common situation while simulating the VR application, we decided to use the following configuration to build our octree:

- Maximum depth: 4
- Minimum primitives per octant: 64

Embedding this octree organization structure with the standard view frustum culling algorithm, will prevent many visibility tests while traversing the entire scene on the quest of finding the visible nodes because, before testing each object inside an octant, the test is performed with the octant itself, and if the visibility test fails, it will prevent as many object's testing as the number of objects stored in that octant and its sub-octants children.

4.3 Occlusion culling

Our approach considered the development of an occlusion culling algorithm based in the literature review [Hanson98] [Papaioannou06], which has the benefit of bringing higher frame rate performance, by discarding occluded objects in the culling traversal phase. This algorithm has the goal of avoiding drawing and processing objects that passed the octree intersection test, after the "view frustum culling algorithm", but are hidden by other objects or groups of objects inside the view frustum, therefore we would say it could also be understood as the hidden objects culling algorithm. The occlusion culling algorithm comprises two phases. The first one, corresponds to a standard Z-Buffer technique provided by OSG, where a two-dimensional array with one element per each pixel in the display, holds a list of depth values of objects that will be mapped into that Z-Buffer element. In the second phase, we compute the distance from each object to the camera eye, based on a ray casting technique, where we trace a ray from the eye point (of the virtual camera) to each pixel in the display intersecting all the objects on the ray path, storing in the Z-Buffer only the depth of the closest object intersected.

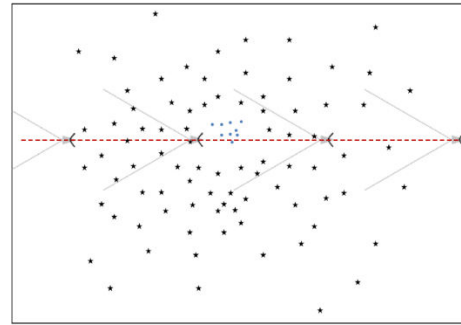


Figure 5: Top projection of the camera path across the Galatica scene, during the data collection for the performance evaluation experiment. The black stars represent the density of stars scattered all over the spatial scene, the blue circles represent the Solar System's planets and the red dashed line represents the projection of the path the virtual camera, during the simulation. In the final pose the camera includes the full dataset in its view frustum.

This way, while traversing the objects' scene we will be drawing only the objects with a Z-value lower than the value on the Z-Buffer, meaning that they are closer and therefore they are not occluded. Since our objects are organized in an octree structure, the ray casting method will test the depth value against the octants before testing the objects themselves, which will bring fewer tests by assuming that if an octant is occluded, all the objects contained within it are occluded too.

5. GALACTICA'S RESULTS AND DISCUSSION

To evaluate our system algorithms in different conditions we have created a camera animation path, crossing the whole VR scene from one side to the other passing through the scene center, where the Solar system is located. The animation lasts 30 seconds. Using this measurement approach allows us to evaluate and compare the algorithms performance in different conditions: we will start with a pose where there are no objects inside the view frustum, then the camera will be travelling backwards inside the scene, increasing the number of visible objects, until the camera reaches the opposite side, where the entire scene is inside the camera's view frustum.

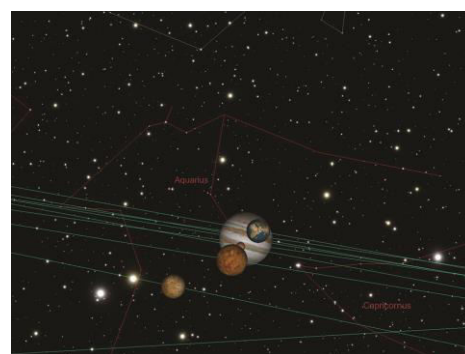


Figure 6: Screenshot taken from our Galatica application with the camera at the center of the scene viewing Solar System's planets, part of the planet's orbits and stars constellations (3D polylines linking stars).



Figure 7: Screenshot taken from our Galatica application of a Milky Way's landscape.

A picture depicting the camera's path during the animation is presented in Figure 5 where we illustrate the different types and amounts of objects inside the view frustum along the camera's animation path. We may observe that this amount is increasing during the camera animation until all objects are visible. This approach will provide us a good way to compare the behavior from when there are no visible objects to when all objects are visible, and consequently need to be drawn by the graphics pipeline. Figure 6 and Figure 7 depict two different camera's poses that are available during the camera animation while the data is being collected.

5.1 Evaluation metrics

To evaluate our application and its performance we defined a set of metrics to measure the application's performance, by reading the continuous frame rate and times taken in the Cull, Draw and GPU traversals during the camera animation, which are the most relevant traversals affected by our custom algorithms. In order to validate and compare our algorithms, we simulated our application under different conditions:

1. OSG standard visualization techniques (per object view frustum culling);
2. OSG standard visualization techniques (per object view frustum), plus our Octree view frustum approach;
3. OSG standard visualization techniques (per object view frustum), plus our Octree view frustum approach and our occlusion culling algorithm.

The measurements were taken simulating the application in a PC with the following configuration:

- Intel Core i7-3970X CPU @3.5GHz
- 16 GB of RAM DDR3
- NVIDIA Quadro 5000 (2.5 GB GDDR5)
- Data set of 100.639 star objects and 104.328 polygons, representing planets of the solar system and constellations.

The measurements presented regard to a simulation in stereo projection since our application is intended to be operating in the CaveH at Lousal, which is a large-scale immersive VR system with stereo projection.

5.2 Measurements

The following diagrams depict the results measured with the metrics and conditions stated in the previous sub-section, where the vertical axis represents the frame rate and the horizontal axis, the camera's animation normalized

key frames. As it may be clearly observed in Diagram 1, both Octree and Octree plus Occlusion culling achieve higher frame rates from the beginning until almost half of the entire animation, and will become inverted from that point on, which is the expected behavior.

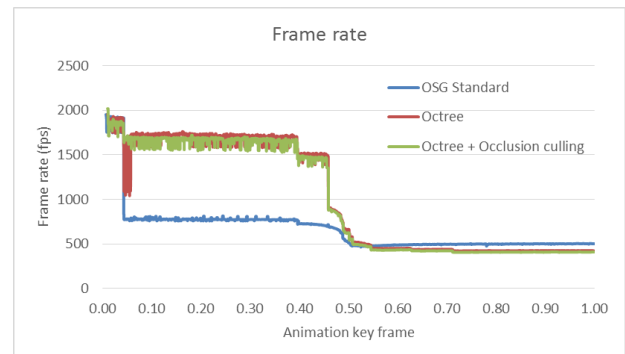


Diagram 1: Camera's animation frame rate variation in stereo projection mode.

In a first analysis, when any object hasn't yet appeared inside the view frustum, all algorithms behave quite similarly, with frame rates around 1800 fps. This similar behavior happens due to the fact that with an octree spatial organization, the octree's root node won't pass the culling test and no other octant or nested object will be tested, resulting in only one test. Only one culling test is performed with the same visualization conditions, in a scene with the standard OSG scene graph, which won't pass the culling test as well and will prevent testing all the scene's objects. By the time the root node will pass the test, all objects will have to become tested in order to decide if they need to be drawn, and this moment happens around the 0,05 key frame, where we observe a big drop of the frame rate of the OSG standard algorithm. This big drop, in our case, has another explanation besides the delay introduced by the object visibility search tasks. As we have stated previously, all our stars are stored in only one geometry (a single OSG node) which is drawn during the GPU phase, which means that if there is any visible star inside an octant, the system will draw all the stars encapsulated in that OSG node, which is very inefficient. This phenomenon can be observed, by the large increase in the duration of the GPU pipeline's phase at the 0,05 key frame, in Diagram 2.

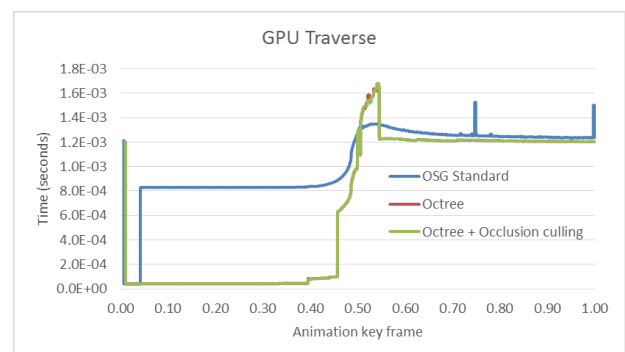


Diagram 2: Camera's animation GPU pipeline's traverse duration in stereo projection mode.

Since all the stars are embedded in the same scene graph object, with the OSG standard view frustum culling technique, the system performs only one visibility test to decide if the stars are visible or not. When the first star passes the test, all of them will be selected to be included in the cull list and therefore drawn in the respective phase, and this is the reason why this OSG standard algorithm, is the one who spends less time in the cull phase throughout the entire camera's animation (see Diagram 3).

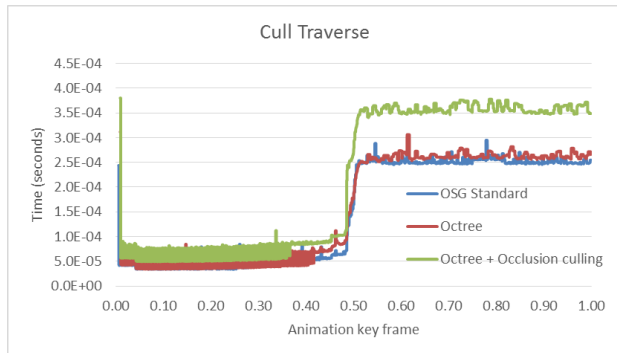


Diagram 3: Camera's animation cull pipeline's traverse duration in stereo projection mode.

From the 0,05 to 0,4 key frames we do not observe much differences in the various algorithms behaviors, regarding both frame rate or traverse duration, because the amount of visible objects is increasing at a low pace. During this period, the OSG standard algorithm average frame rate is around 750 fps. The Octree's frame rate is the highest, with around 1700 fps and the Octree plus Occlusion algorithm is 1600 fps, in average. This last algorithm spends a bit more time to search for occluded objects and this is the reason why its frame rate is a little bit lower than the Octree approach. The 0,4 key frame corresponds to the moment when the camera is approaching to the scene's center, where the smallest octants with planets are located, bringing a higher effort to compute what needs to be drawn, as it can be observed in Diagram 4. Between 0,4 and 0,55 key frames the camera is experiencing the visualization of a new kind of astrophysical objects, the Solar System's planets and their orbits, which will bring different phenomena to the different graphics pipeline's traversal phases, for all algorithms. Since the OSG standard algorithm only uses a single OSG node to represent all the stars, their only visible objects are one stars object, the planets and their orbits, which is not verified in the other two algorithms. These have the scene organized in an Octree and by this time they have a lot more data to traverse and all the new octants as well as the planets and their orbits, intersected by the view frustum. Consequently, this phase is the point where the algorithms' performances will suffer a change. In the cull phase, with OSG standard algorithm, the system only needs to traverse an object for all the stars and another per planet and orbit, whereas with the Octree approach, it needs to traverse each planet and orbit, as well as all the visible octants, which are nearly half of the total because we're in the center of the entire scene. The Octree plus Occlusion culling algorithm, will have to traverse the same as the Octree approach and will have also to test each planet for their occluded visibility, which

means that this algorithm will be the one to take more time to compute the objects visibility (a small set, corresponding to just the Solar System planets), followed by the Octree approach, making the OSG standard approach the faster to process this pipeline's phase.

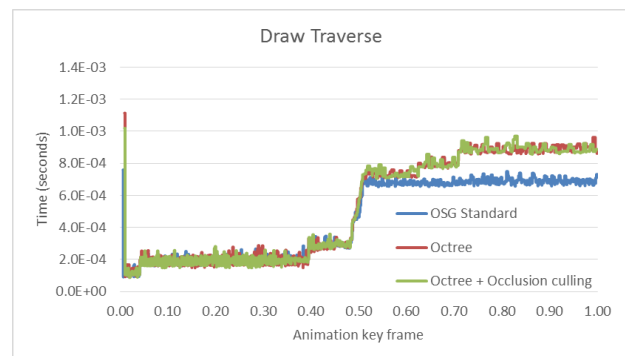


Diagram 4: Camera's animation draw pipeline's traverse duration in stereo projection mode. Regarding the draw pipeline's phase (Diagram 4), we may realize that in this phase the OSG standard approach takes less time, because it has to process less number of objects, followed by the Octree and Octree and Occlusion approaches, which behave almost the same way, since they have to process almost the same number of objects. This result shows that the occlusion calculation brings no relevant effect when the planets (occluders) are occluding each other, because they are only a few of such objects in the dataset, which is an expected result. In the GPU phase (Diagram 2), we can observe the same behavior as in the other pipeline's steps, because there are more objects to process than in the previous camera's animation phase. In this pipeline's traversal step, we may observe that there is a moment when the OSG standard algorithm take a little lower time to process the scene than the other approaches, which happens in the moment when all planets and orbits are visible, and will be followed by an inversion again, with the OSG standard to take more time to perform the processing, as the planets will become so small that they will occupy no pixels in the final image and won't be drawn. After this animation's phase and until the end of the experiment, the frame rates decrease a lot and bring better efficiency for the OSG standard algorithm. At this point the system is about to draw the entire scene and this standard technique is the faster approach given the small size of the dataset. In the worst case scenario, when everything is visible, the frame rate is in average 500 fps for the OSG standard algorithm. Using the Octree approach the frame rates are a little bit lower, around 420 fps in average. Finally, with Octree plus Occlusion culling, the frame rate is in average 400 fps. This happens because the OSG standard approach is, in this case, the one who has less objects to compute in all the pipeline's traversals.

6. CONCLUSIONS AND FUTURE WORK

In this paper we have described the requirements, architectural elements, development and performance evaluation of a Digital Planetarium appropriate for an immersive VR environment, such as the CaveH at Lousal. Via the support and improvement of an existing VR system

(CaveH), we have developed and tested an application, referred to as Galatica, which is able to run a realistic and rich visual and aural astrophysical user experience. We have developed advanced visualization acceleration techniques to ensure that our application is efficient, runs at interactive rates in stereoscopy and is scalable to datasets of arbitrary dimensions, as can be seen by the performance results (see Diagram 1). The application is based on a storyboard of a thematic travel through the Solar System, with visual and aural elements. The application is developed such that it can support other storyboards that might be defined for the DUA or other datasets. In fact, we have developed features to remove, update and add new datasets to the database that will feed the VR application, allowing to bring continuously diversified and up-to-date astronomical experiences to our immersive VR experience. We have computed the frame rate, GPU traverse time, Cull traverse time and Draw traverse time for three visualization conditions: (A) using standard OSG view frustum culling technique; (B) using view frustum culling with and our octree organizing the scene's objects; (C) using view frustum culling with our octree organizing the scene's objects and our occlusion culling algorithm. In our experimental analysis we observed the entire dataset of 100.639 star objects and 104.328 polygons. We have generally concluded that our octree organization and octree plus object culling techniques outperforms the standard OSG view frustum culling, when around half or less than half of the dataset is in view. This is due to the use of a balanced spatial organization structure to store the tested objects (a set of around 100K stars with their constellations and some planets with their orbital lines), preventing testing each object in the scene for its visibility. We have realized that when the virtual camera is navigating through the scene's objects, almost all the time the view frustum will hold many objects, but not the whole set contained in the entire scene, which is when our octree brings more value in efficiency, rather than the standard OSG approach. Our occlusion culling approach, in the case of our test VR scene (with the DUA dataset), didn't brought improvements on the frame rate because the only occluders present in the scene are the planets (the object occlusion culling algorithm does not apply to stars billboards) and they are only a small number of them, being only visible in a small part of the whole scene. We have taken this conclusions for the case of Galatica with a quite small planet dataset. The cases when our visualization optimization techniques aren't more efficient than the standard OSG approach, are when the virtual camera is observing almost the entire VR scene. We expect that for larger datasets the advantage of our visualization and optimization techniques based in octree spatial organization and object culling, become more noticeable. As for future work, we expect to include more and larger data sources available in the community, which can provide more diversified types of data, like specific phenomena in an arbitrary galaxy, or the exploration of a nebulae or a black hole. We plan also to bring even more realistic and impressive landscapes to the user experience, by creating new or remodel the existing geometric models of the celestial bodies, with increasing level of detail and

resolution. Although not tackled in the paper, we plan to improve the virtual navigation model by adding motion blur on the camera's movement, since it traverses many different ranges of speed. Creating new sonification cues for different types of astronomical bodies and managing how each sound would influence the user's experience, is an additional interesting direction for future research. As it can be expected, increasing the quality and resolution of the visual and aural content will bring a struggle to the performance, so a promising option is try new approaches to manage the storage of the objects. Performing the object testing at the polygon level rather than at the bounding box level, would bring a more reliable visual representation. Using the view-dependent level-of-detail approach to draw the objects might be a good direction to pursue, since it will make the draw phase to spend more effort on parts of the objects that are relevant to each view. The occlusion culling algorithm could be explored deeper, now that there are some new and faster GPU-based alternative ways of calculating the objects' visibility through occluders.

7. ACKNOWLEDGEMENTS

The authors would like to pay special appreciation to Lara Alegre and Alexandre Loureiro, that helped designing and writing the Galatica storyboard. To Mafalda Abrunhosa, a special thanks for her availability to record the narration of Galatica. To Filipe Gaspar and Vando Pereira a special appreciation for their guidance in the topics of GPU computing.

8. REFERENCES

- [Abbott02] Abbott, B., Emmart, C., Marx, S. & Wyatt, R., "Digital Universe Atlas", <http://www.haydenplanetarium.org/universe/>, 2002.
- [Osfield05] Osfield, R., "OpenSceneGraph", <http://www.openscenegraph.org>, 2005.
- [Dias07] Dias, M. S., Soares, L. P., Varella, R., Pires, F., Bastos, R., Carvalho, N. & Costa, V., "CAVE-HOLLOWSPACE do Lousal - Princípios Teóricos e Desenvolvimento, Curso Curto", *15ª Encontro Português de Computação Gráfica*, Microsoft Portugal, Oeiras, 17th October 2007.
- [Cruz-Neira92] Cruz-Neira, C., Sandin, D., DeFanti, T., Kenyon, R. & Hart, J., "The CAVE: Audio Visual Experience Automatic Virtual Environment", *Communications of the ACM* 35, 1992, pp. 65–72.
- [Laurel01] Laurel, C., "Celestia", <http://www.shatters.net/celestia/>, 2001.
- [Hastings-Trew00] Hastings-Trew, J., "JHT's Planetary Pixel Emporium", <http://planetpixlemporium.com/>, 2000.
- [Wang10] Wang, R. & Qian, X., "OpenSceneGraph 3.0: Beginner's Guide", 2010, Packt Publishing Ltd.
- [Akenine-Möller08] Akenine-Möller, T., Haines, E. & Hoffman, N., "Real-Time Rendering", Third Edition, 2008, Wellesley, Massachusetts: A K Peters, Ltd.
- [Hansong98] Hansong, Z., "Effective Occlusion Culling for the Interactive Display of Arbitrary Models", Ph.D. Thesis, 1998, University of North Carolina: USA.
- [Papaioannou06] Papaioannou, G., Gaitatzes, A. & Christopoulos, D., "Efficient Occlusion Culling using Solid Occluders", *Proceedings of the 14th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, January 30th - February 3rd 2006, Plzen: Czech Republic.
- [Soares10] Soares, L. P., Pires, F., Varella, R., Bastos, R., Carvalho, N., Gaspar, F. and Dias, M. S., "Designing a Highly Immersive Interactive Environment: The Virtual Mine", *Computer Graphics Forum*, The Eurographics Association and Blackwell Publishing Ltd., Volume 29, Issue 6, pages 1756–1769, September 2010.

Medidas e Detecção de Estruturas Morfológicas Para Craniometria Usando Modelos 3D

Luís Neves

Departamento de Eletrónica, Tele-
comunicações e Informática
Univ. Aveiro, Portugal
luispneves@ua.pt

Paulo Dias

Departamento de Eletrónica, Tele-
comunicações e Informática
DETI / IEETA
Univ. Aveiro, Portugal
paulo.dias@ua.pt

Catarina Coelho

Departamento de Ciências da Vi-
da, Faculdade de Ciências e Tec-
nologia
Univ. Coimbra, Portugal
catarina.coelho.rs@gmail.com

Maria Teresa Ferreira

Centro de Ciências Forenses
Univ. Coimbra, Portugal
mtsferreira76@gmail.com

Hélder Santos

iDryas, Grupo Dryas Octopetala
Coimbra, Portugal
helder.santos@dryas.pt

Beatriz Sousa Santos

Departamento de Eletrónica, Tele-
comunicações e Informática
DETI / IEETA
Univ. Aveiro, Portugal
bss@ua.pt

Resumo

Este artigo descreve os recentes desenvolvimentos num trabalho ainda em curso referente a uma nova abordagem à craniometria tradicional baseada na análise de modelos digitais 3D de crânios através da aplicação CraMs (CranioMetric Measurements). O desenvolvimento desta aplicação tem como primeiro objetivo apoiar os antropólogos na obtenção de medidas craniométricas. Recentemente foi acrescentada a possibilidade de obter duas medidas adicionais baseadas em projeções de planos auxiliares. Foram ainda desenvolvidos e integrados na aplicação métodos para detetar estruturas de interesse através da análise da morfologia dos crânios. Estes métodos são baseados na curvatura e na deteção de sharp edges e suportam a extração de informação morfológica das órbitas e suturas cranianas, estruturas consideradas de interesse para o estudo dos crânios. Apresentam-se os métodos usados e os resultados obtidos em oito crânios.

Keywords

Craniometry, Morphology, 3D Models, Feature Detection.

1. INTRODUÇÃO

A craniometria é usada no campo da Antropologia para identificar o sexo, a ancestralidade e algumas variações populacionais [Oxnard74]. Na craniometria tradicional, os antropólogos fazem medições diretamente nos crânios. Contudo, esta abordagem tem alguns problemas: verifica-se uma baixa repetibilidade nas medições (intra- e inter-observador), os crânios tendem a ficar danificados devido ao manuseamento e a análise métrica de fragmentos de crânios é muito restrita [Utermohle82], [Stephens00].

O trabalho apresentado neste artigo foi desenvolvido para acrescentar novas funcionalidades à aplicação CraMs (CranioMetric Measurements). Esta aplicação visa auxiliar os antropólogos a obter medidas craniométricas, tentando diminuir o erro na análise e preservando os espécimes através do uso de modelos digitais 3D adquiridos por *scanning* dos crânios [Santos13].

A aplicação CraMs, tal como a craniometria tradicional, é baseada em medidas craniométricas definidas entre pontos craniométricos. No entanto, um dos campos que se integra na craniometria é a Geometria Morfométrica que consiste na análise e quantificação morfológicas dos crâ-

nios, ou seja, a avaliação de contornos, proporções, e volumes. Porém, estas avaliações são subjectivas se se utilizar apenas a craniometria tradicional [Gonzalez11], [Hefner09] tendo portanto uma aplicação limitada. Um dos objetivos do trabalho aqui apresentado é integrar na aplicação CraMs a possibilidade de cálculo de mais medidas e desenvolver métodos para a deteção de algumas estruturas morfológicas. A ideia subjacente é que a deteção de estruturas por análise morfológica usando métodos automatizados possa ser mais objetiva permitindo novas formas de analisar os crânios. De seguida apresenta-se brevemente a aplicação CraMs, a nova funcionalidade (possibilidade de deteção de novas medidas e análise de estruturas), os métodos que a suportam, bem como resultados obtidos em oito crânios.

2. APLICAÇÃO CRAMS

A aplicação *CraMs* (CranioMetric Measurements) foi desenvolvida em C++ usando o VTK para visualização e processamento 3D, e o QT para a interface gráfica [Dias13; Santos13]. A aplicação permite o alinhamento de crânios e a deteção de pontos craniométricos. Estes pontos são depois usados para calcular as medidas craniométricas.

2.1 Aquisição dos modelos

Os recursos fundamentais para o funcionamento da aplicação são os modelos 3D dos crânios. Estes são adquiridos através de um scanner de luz estruturada (*Breukmann SmartSCAN¹*). Uma equipa que inclui um técnico especialista em aquisição 3D (HS) e uma antropóloga (CC) estão envolvidos no processo para garantir melhores resultados. Tipicamente, o modelo 3D final é composto por 1.5 milhões de pontos, com um erro de aquisição abaixo dos 30µm. Mais informações sobre este processo podem ser encontradas em [Dias13].

2.2 Medidas disponíveis

Como já foi referido anteriormente, as medidas craniométricas são obtidas a partir dos pontos craniométricos. Estes pontos podem ser definidos na aplicação CraMs de três formas: automaticamente, sem interação do utilizador; semi-automaticamente, onde o utilizador providencia manualmente uma estimativa inicial da localização do ponto que depois é refinada usando informação de curvatura do modelo e conhecimento *a priori* do ponto a determinar; manualmente, dependendo unicamente da sugestão do utilizador. Na Figura 1 estão representados alguns dos pontos craniométricos.

Na versão da aplicação desenvolvida em [Santos13] era possível obter 12 das 24 medidas selecionadas para a análise antropológica dos crânios, cujas denominações se apresentam na Tabela 1.

Tabela 1: Medidas disponíveis na aplicação inicial do CraMs.

ZYB – largura bizigomática	BBH – altura basion-bregma	BPL – comprimento basion-prostion	AUB – largura biauricular
XCB – largura máxima craniana	BNL – comprimento nasion-basion	UFHT – altura facial superior	NLH - altura nasal
GOL – comprimento crânio máximo	FRC – corda frontal	MAB – largura maxilo-alveolar	UFBR – largura facial superior

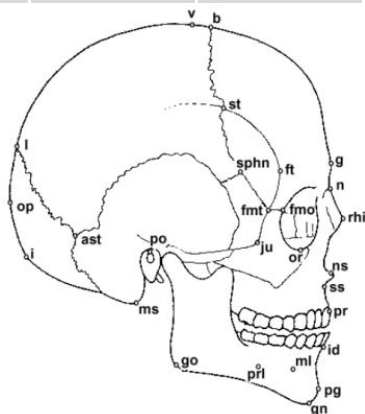


Figura 1: Vista lateral do crânio com anotações em alguns pontos craniométricos.

2.3 Medidas adicionais

Uma das limitações da versão inicial da aplicação CraMs era o número limitado de medidas disponibilizadas, que

permitia uma análise demasiado simplista da morfologia de um crânio. Para resolver esse problema, foram desenvolvidos métodos para obter as restantes 12 medidas craniométricas. A denominação das 12 medidas que foram adicionadas é apresentada na Tabela 2.

Tabela 2: Medidas adicionadas à aplicação CraMs.

WFB – largura frontal mínima	NLB – largura nasal	FOB – largura do foramen magnum	EKB – largura biorbitária
DKB – largura interorbital	PAC – corda parietal	OCC – corda occipital	FOL – comprimento do foramen magnum
OBB – largura orbital	OBH – altura orbital	MAL – comprimento maxilo-alveolar	MDH – altura do processo mastóide

Algumas destas medidas são difíceis de obter usando os métodos tradicionais manuais (medições com recurso à craveira ou compasso), nomeadamente as medidas MAL e MDH. Estas requerem o uso de projeções e foram obtidas na aplicação através do uso de planos definidos pelo utilizador. A Figura 2 exemplifica a obtenção destas medidas através dos métodos tradicionais e com a aplicação CraMs.

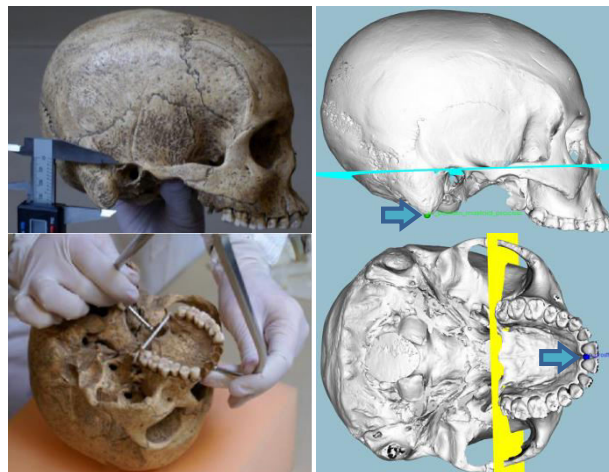


Figura 2: Medidas MDH (em cima) e MAL (em baixo) obtidas com métodos tradicionais (esquerda) [Coelho12] e com a aplicação CraMs (direita).

Na aplicação, os planos de referência estão representados a azul claro e a amarelo, os pontos craniométricos a verde e a azul, para as medidas MDH e MAL respetivamente.

3. ANÁLISE DA GEOMETRIA MORFOMÉTRICA

Na craniometria, apesar de ser uma área de investigação em expansão, ainda são poucos os estudos que usam informações sobre geometria morfométrica. Isto deve-se ao facto de ser difícil, usando apenas métodos tradicionais, descrever e classificar formas morfológicas complexas como o crânio de forma objetiva [Gonzalez11], [Hefner09]. Contudo, com o desenvolvimento de aplicações de *software* e utilização de modelos 3D, torna-se possível explorar novos métodos para obter informações sobre estruturas morfológicas do crânio de forma objetiva.

Embora o crânio possua diversas estruturas morfológicas de interesse, foi necessário começar por definir quais as primeiras a abordar, tendo sido selecionadas as órbitas

¹ <http://aicon3D.com/>

(estruturas ósseas onde, no vivo, se alojam os olhos – Figura 3) e as suturas (que são as articulações fibrosas que unem os ossos do crânio). A Figura 4 mostra a sutura Sagital (localiza-se na região superior do crânio, na zona de união dos dois ossos parietais) que foi usada para testes dado a sua importância na análise da morfologia craniana.

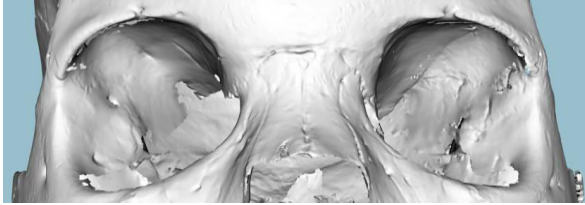


Figura 3: Órbitas num modelo 3D (vista anterior).



Figura 4: Sutura sagital num modelo 3D (vista superior).

3.1 Conceitos Utilizados

Para detetar as estruturas de interesse nos crânios, decidiu-se usar inicialmente os valores de curvatura por já serem usados na aplicação CraMs. Depois de obter alguns resultados, optou-se por recorrer a uma melhor filtragem dos vértices através de um algoritmo de segmentação para definir quais os vértices que pertencem a zonas de interesse (*Sharp Edges*).

3.1.1 Valores de curvatura

Estes valores são obtidos através da extração de informação sobre o tipo e intensidade da curvatura numa superfície. Para extrair esta informação são definidas as Curvaturas Principais. Estas consistem no máximo e mínimo das Curvaturas Normais para cada vértice da superfície. São depois usadas para calcular os valores de curvatura Média e Gaussiana para cada vértice da superfície. Com estes valores é possível classificar uma superfície como sendo côncava, plana ou convexa (Figura 5).

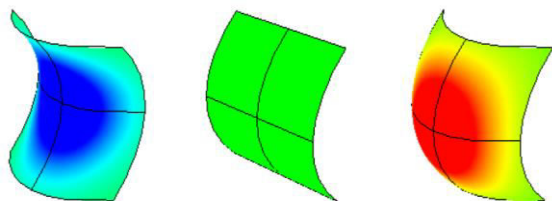


Figura 5: Tipos de superfície. Côncava (esquerda), plana (centro) e convexa (direita) [Santos13].

3.1.2 Sharp edges

Foi usado um algoritmo que permite determinar quais os vértices que definem zonas distintas na superfície. Este é descrito por Vieira e Shimada [Vieira05] e usa um conceito ao qual os autores dão o nome de *Sharp Edges*. O algoritmo é simples e fácil de implementar, contudo o seu desempenho é influenciado pela resolução do modelo usado.

O algoritmo baseia-se em duas expressões. Inicialmente, para cada vértice x_i do modelo, o tamanho da vizinhança é determinado calculando o comprimento médio das arestas adjacentes aplicando a expressão (1).

$$I_{avg,i} = \frac{1}{N} \sum_{j \in N(i)} \|x_j - x_i\| \quad (1)$$

Depois, o vértice é considerado como sendo parte de uma *sharp edge* se a expressão (2) se verificar.

$$\frac{1}{|k_i|} < \lambda \cdot I_{avg,i} \quad (2)$$

Onde N é o número de vizinhos de x_i , $N(i)$ é o conjunto de vizinhos de x_i , $\frac{1}{|k_i|}$ é o raio de curvatura mínimo para o vértice x_i e k_i é o valor de curvatura para esse vértice. O valor do λ deve ser ajustado de acordo com a resolução do modelo e a precisão dos resultados esperados. Na Figura 6 estão representados a vermelho os *sharp edges* detetados usando os valores de curvatura gaussiana e um fator λ igual a 55, numa órbita e em algumas suturas da calote craniana.

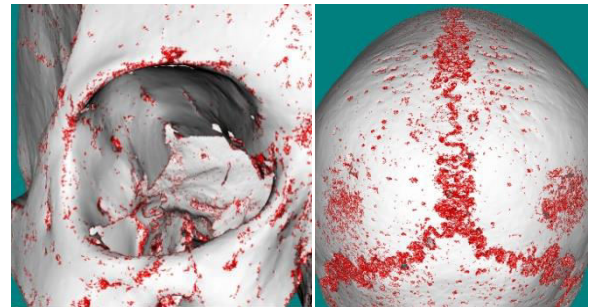


Figura 6: Representação dos *sharp edges* (vermelho) nas zonas de interesse: órbita (esquerda) e suturas sagital e lambóide (direita).

É de referir que foram feitos testes com os dois tipos de curvatura (média e gaussiana) e usando diversos fatores λ (entre 1 e 60) para determinar quais seriam os parâmetros mais adequados. Depois de alguns testes, decidiu-se empiricamente usar a curvatura gaussiana e um fator λ que varia diretamente com o número de vértices do modelo (dividindo este por 30.000).

3.2 Métodos de deteção das órbitas

3.2.1 Usando curvatura

Inicialmente foi desenvolvido um método para a deteção das órbitas baseado unicamente nos valores de curvatura. Com base em três pontos de referência definidos pelo utilizador, o método define uma estrutura fechada constituída por pontos de curvatura máxima. Os resultados deste método estão representados na Figura 7 para a mesma órbita, usando modelos 3D de baixa e de alta resolução. A cor das esferas representa a ordem pela qual as estruturas foram definidas pelo método (começando no preto e seguindo para o amarelo).

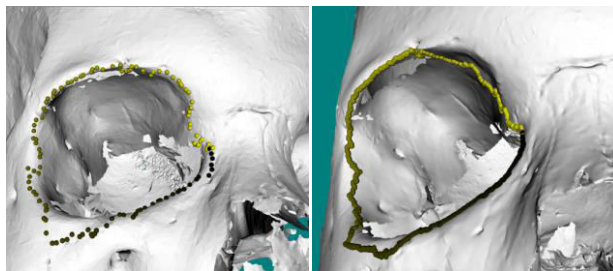


Figura 7: Detecção de uma órbita com curvatura em modelos 3D de resolução baixa (esquerda) e alta (direita).

Estes resultados preliminares, usando exclusivamente os valores de curvatura, mostraram claramente que apenas a curvatura não seria suficiente para permitir uma deteção rigorosa das estruturas sendo necessário introduzir uma filtragem adicional.

3.2.2 Usando sharp edges

Por forma a refinar o método de deteção foi usado o conceito de *sharp edges* para obter uma melhor filtragem dos vértices usados na constituição da estrutura. Inicialmente, aumentou-se o número de pontos de referência marcados pelo utilizador de três para quatro. A estrutura detetada no mesmo crânio, constituída por *sharp edges*, está representada na Figura 8 à esquerda. Os quatro pontos de referência estão assinalados a verde (e por setas azuis), a cor das esferas indica a ordem com que esta foi definida.

Depois de definida a estrutura é encontrado o plano que melhor se ajusta aos pontos desta. O resultado deste processo está representado na Figura 8 à direita.

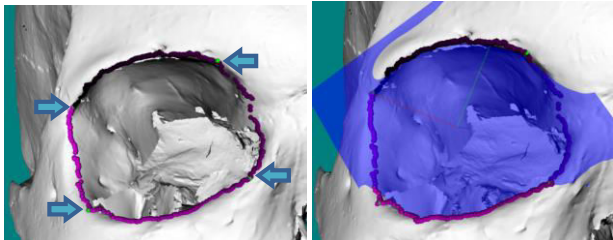


Figura 8: Resultado da deteção de uma órbita com *sharp edges* num modelo de alta resolução (esquerda) e resultado do ajuste do plano à estrutura (direita).

Finalmente é feita a projeção dos pontos da estrutura sobre o plano e definido um novo sistema de coordenadas. Os pontos estão representados na Figura 9 pelas esferas azuis (coincidentes com o plano) e os semi-eixos coordenados pelas linhas vermelha, verde e amarelo.

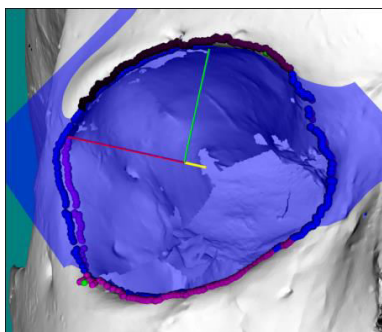


Figura 9: Resultado da projeção dos pontos e definição do sistema de coordenadas na deteção de uma órbita.

Neste novo sistema de coordenadas (definido no plano da órbita) é calculada a elipse que melhor se ajusta a este conjunto de pontos. O resultado deste processo está representado na Figura 10. As esferas azuis são os pontos projetados, as esferas vermelhas são os pontos da elipse, as linhas amarela e azul claro representam os eixos da elipse. O comprimento destes eixos é depois usado para definir as medidas OBB e OBH. Estas duas medidas são difíceis de obter usando métodos tradicionais pois a sua definição implica que tenham que ser perpendiculares entre si, sendo esta restrição sempre garantida ao usar os eixos da elipse para as definir. A forma como estas medidas são obtidas usando métodos tradicionais está demonstrada na Figura 11.

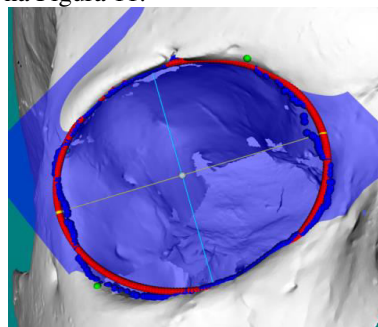


Figura 10: Resultados do ajuste da elipse aos pontos projetados.



Figura 11: Demonstração da obtenção das medidas OBB (esquerda) e OBH (direita) usando métodos tradicionais [Coelho12].

3.3 Método de interação

No método de deteção das órbitas usando *sharp edges* (ver seção 3.2.2), como foi referido, inicialmente eram utilizados quatro pontos de referência, marcados pelo utilizador, para detetar a estrutura constituída por *sharp edges* (representada na Figura 8). Contudo, após efetuar alguns testes ao método de deteção notou-se que ter um número fixo de pontos de referência prejudicava a interação e os resultados da deteção. De modo a resolver este problema foi implementado um novo método de interação para permitir um número de pontos de referência variável. Neste método, a estrutura constituída por *sharp edges* é inicializada com base em dois pontos marcados pelo utilizador. Depois de estar inicializada, o utilizador pode ir marcando pontos para guiar a estrutura. Em qualquer momento, o utilizador pode finalizar o processo indicando que quer fechar a estrutura (unindo o último ponto que marcou com o primeiro) e confirmando o resultado final. É importante referir que é possível ir apagando as seções que já foram detetadas caso estas não sejam satisfatórias e adicionar mais pontos para guiar o algoritmo. Na Figura 12 estão exemplificadas as estrutu-

ras obtidas no mesmo crânio usando três e sete pontos de referência (assinalados a verde e por setas azuis).

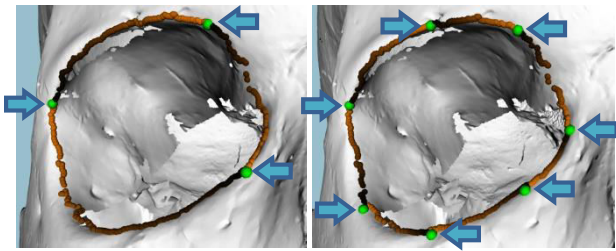


Figura 12: Estrutura da órbita detetada com três (esquerda) e sete (direita) pontos de referência.

3.4 Métodos de deteção das suturas

3.4.1 Usando curvatura

Foi usado para deteção das suturas um método semelhante ao método de deteção das órbitas baseado apenas nos valores de curvatura. Com base em dois pontos de referência marcados pelo utilizador, o método encontra um caminho que os liga através de pontos de curvatura mínima. Os resultados deste método estão representados na Figura 13 para a mesma sutura, usando modelos de baixa e de alta resolução. A cor das esferas representa a ordem com que a estrutura foi definida pelo método.

Como se pode ver na Figura 13, no modelo de alta resolução este método já mostrava alguns problemas em detetar uma sutura simples.

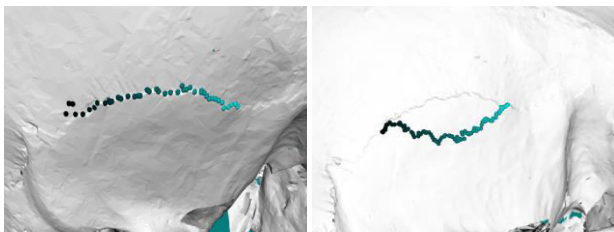


Figura 13: Deteção da sutura escamosa com curvatura em modelos de resolução baixa (esquerda) e alta (direita).

3.4.2 Usando sharp edges

O método de deteção das órbitas usando *sharp edges* (explicado em 3.2.2), foi adaptado para a deteção das suturas. Com base em dois pontos de referência marcados pelo utilizador nas extremidades da sutura, o método começa por definir uma estrutura que os une diretamente constituída por *sharp edges*. Para cada ponto desta estrutura a primeira vizinhança (pontos adjacentes) é verificada e os pontos considerados como *sharp edges* são guardados. Este processo é repetido iterativamente até não serem encontrados novos *sharp edges* ou até ser atingido um limite de vizinhança (definido na aplicação). O resultado deste método está representado na Figura 14. A estrutura inicial está representada pela linha de esferas azuis, na qual a cor indica a ordem com que foi definida, e os pontos considerados como sendo parte da sutura estão representados por esferas ciano.

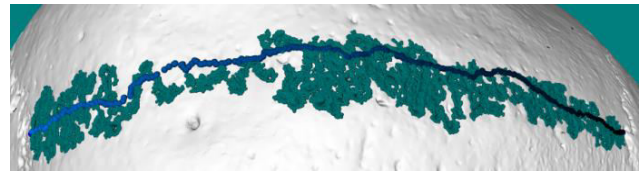


Figura 14: Resultado da deteção com *sharp edges* da sutura sagital num modelo de alta resolução.

4. RESULTADOS E DISCUSSÃO

Nesta secção são apresentados os resultados obtidos com os métodos de deteção usando *sharp edges*. Os resultados obtidos com os métodos usando somente os valores de curvatura são preliminares e não foram quantificados. À exceção das suturas todos os valores foram obtidos pelas antropólogas (CC e MTF).

4.1 Medidas com projeções

Foram obtidas as medidas MAL e MDH usando a aplicação CraMs e comparadas com as obtidas por uma antropóloga usando craniometria tradicional (M.T.) (Tabela 3). Para dois crânios (#21 e #38) não foi possível obter o valor da medida MAL devido à fragmentação da região anatómica a analisar. Na secção 8 apresentam-se os resultados para os oito crânios.

Tabela 3: Valores das medidas MAL e MDH obtidos usando os dois métodos.

Medida	Crânio #67			
	Antr #1		Antr #2	
	CraMs	M.T.	CraMs	M.T.
MAL	54	57	53,6	59
MDH	25,8	27	23,5	25

Para fazer uma análise mais rigorosa dos valores obtidos foi calculado o índice de precisão TEM (Erro Técnico de Medição), que também é usado regularmente em estudos craniométricos para exprimir a margem de erro das medidas [Perini05]. O valor do TEM é calculado aplicando a expressão (3) seguida da expressão (4).

$$TEM \text{ absoluto} = \sqrt{\frac{\sum d_i^2}{2n}} \quad (3)$$

$$TEM \text{ relativo} = \frac{TEM \text{ absoluto}}{VAV} * 100 \quad (4)$$

Nas quais $\sum d_i^2$ é a soma dos desvios elevados ao quadrado, i é o número de desvios, n é o número de medidas, TEM relativo é o erro técnico expresso em percentagem e VAV é o valor médio variável (média da média entre todas as observações, para cada medida). Os valores de TEM relativo calculados para as duas medidas obtidas com os dois métodos estão apresentados na Tabela 4.

Tabela 4: Valores de TEM relativo obtidos para as medidas MDH e MAL usando os dois métodos em oito crânios.

Medida	CraMs (%)	M.T. (%)	Número de medidas
MAL	1,19	3,12	6 (em 8)
MDH	5,12	6,03	8 (em 8)

Analisando estes valores verifica-se que há uma diminuição do erro inter-observador usando a aplicação CraMs. Estas medidas são frequentemente excluídas em estudos craniométricos devido ao grande erro que é usual obter através dos métodos tradicionais. Contudo, ao comprovar-se uma redução constante do erro através da utilização do CraMs num estudo mais abrangente (efetuado em mais crânios), será possível que estas medidas passem a ser utilizadas recorrentemente, melhorando assim a análise craniométrica.

4.2 Detecção das Órbitas

Foram obtidas as medidas OBB e OBH para oito crânios usando o método de deteção abordado em 3.4 (com *sharp edges*). A Tabela 5 mostra os valores obtidos para um dos oito crânios.

Tabela 5: Valores das medidas OBB e OBH obtidos usando os dois métodos.

Medida	Crânio #57			
	Antr #1		Antr #2	
	CraMs	M. T.	CraMs	M. T.
OBB	42,7	43	43	43
OBH	32,8	33	32,7	32

Foram calculadas as percentagens de diferença entre as medidas obtidas nos oito crânios usando os dois métodos. Estas percentagens estão representadas na Figura 15

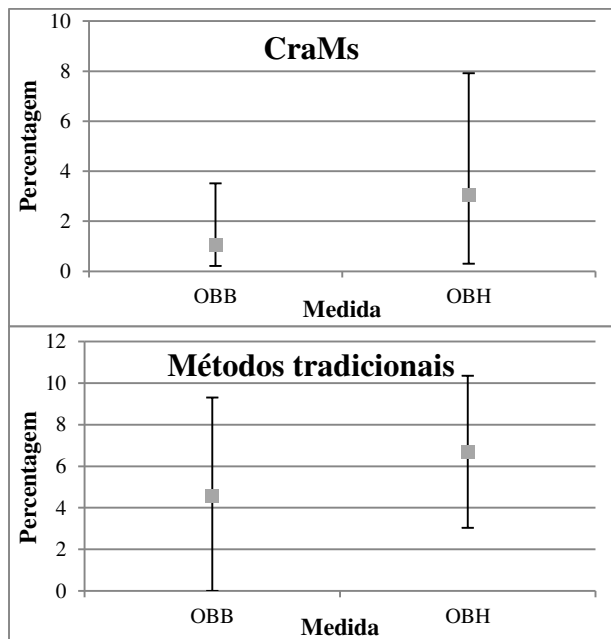


Figura 15: Percentagens de diferença entre medidas usando a aplicação (em cima) e usando métodos tradicionais (em baixo).

Analisando a Figura 15, verifica-se que há uma descida nas percentagens de diferença média e nos limites obtidos usando a aplicação. Também para estas medidas foram calculados os valores de TEM relativo (ver seção 4.1). Estes estão apresentados na Tabela 6 para os dois métodos, usando a aplicação e os métodos tradicionais.

Tabela 6: Valores de TEM relativo obtidos para as medidas OBB e OBH usando os dois métodos em oito crânios.

Medida	CraMs (%)	M.T. (%)	Número de medidas
OBB	1,01	3,77	8 (em 8)
OBH	3,17	4,84	8 (em 8)

Os valores de TEM confirmam que há uma diminuição no erro inter-observador usando a aplicação. Esta diminuição é bastante mais evidente para a medida OBB com uma redução do erro para quase um quarto do correspondente usando métodos tradicionais.

4.3 Detecção das Suturas

Em Antropologia, a classificação da complexidade das suturas varia de autor para autor, sendo comum a classificação em simples, intermédia ou complexa (Figura 16) [Rhine90]. Tendo este facto em mente, um dos autores (LN) classificou a sutura sagital nos oito crânios.

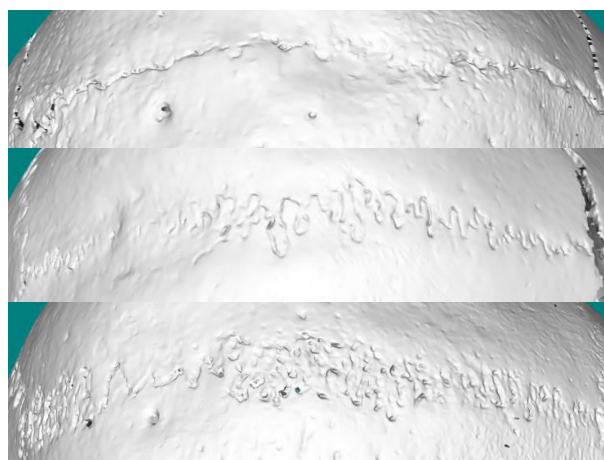


Figura 16: Exemplos dos três tipos de suturas. Simples (em cima), intermédia (centro) e complexa (em baixo).

Na Tabela 7 são apresentadas as classificações da mesma sutura nos oito crânios, o número de vértices que constituem o modelo, o número de pontos considerados como sendo parte da sutura e o rácio em percentagem entre o número de pontos na sutura e o número de vértices total no modelo.

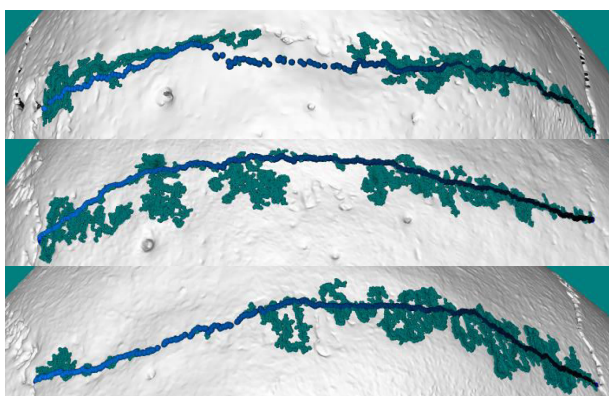
À exceção dos três últimos valores (a negrito) parece haver uma relação direta entre o rácio e a classificação atribuída, verificando-se um aumento no rácio de acordo com a complexidade da sutura.

Nos três crânios em que esta relação não se verificou nota-se a obliteração² da sutura que pode explicar esta discrepância. No crânio número 66, por exemplo, o caminho inicial encontrado pelo método não é o ideal. A Figura 17 mostra os três crânios com problemas na deteção da sutura sagital.

² Com o avançar da idade, dá-se a obliteração das suturas craniais. Ou seja, as suturas começam a encerrar, ficando menos nítidas e podendo mesmo desaparecer.

Tabela 7: Resultados da deteção da sutura Sagital em oito crânios.

Crânio	Classificação	Vértices M.	Pontos S.	Rácio (%)
#21	Intermédia	1.370.198	17.787	1,30
#25	Complexa	1.512.605	46.645	3,08
#38	Simples	1.801.451	6.869	0,38
#39	Intermédia	1.849.701	35.778	1,93
#57	Intermédia	2.096.164	35.128	1,68
#65	Simples	1.711.944	18.280	1,07
#66	Intermédia	2.440.999	22.428	0,92
#67	Intermédia	2.229.845	24.609	1,10

**Figura 17: Problemas na deteção das suturas. Crânio 65 (em cima), crânio 66 (centro) e crânio 67 (em baixo).**

5. CONCLUSÕES E PERSPETIVAS FUTURAS

Com a aplicação CraMs é possível obter todas as 24 medidas craniométricas mais comumente utilizadas, incluindo quatro difíceis de obter com os métodos tradicionais (MAL, MDH, OBB e OBH). Os resultados aqui apresentados para estas medidas, embora só tenham incidido sobre oito crânios, são interessantes e sugerem claramente que estas medidas poderão vir a ser utilizadas mais frequentemente em estudos craniométricos devido às reduções do erro quando obtidas através da aplicação CraMs.

Em relação à geometria morfométrica, os métodos aqui apresentados podem ser considerados como um ponto de partida para futuros estudos craniométricos. Estes métodos foram desenvolvidos com o objetivo de explorar novas áreas computacionais capazes de proporcionar ferramentas simples, precisas e, por conseguinte, replicáveis, à prática quotidiana de um antropólogo físico e/ou forense. O método de deteção das órbitas produziu resultados interessantes com reduções significativas do erro nas medidas quando aplicado a um número significativo de crânios (oito). Quanto à deteção das suturas foram obtidos resultados preliminares promissores. Contudo, de forma a comprovar a sua mais-valia, estes métodos carecem de ser testados num maior número de crânios, designadamente crânios de diferentes origens populacionais e de diferentes cronologias, análise esta que será o próximo passo deste projeto de investigação transdisciplinar.

Em relação à aplicação, com base nos resultados obtidos, o passo seguinte será desenvolver os métodos computacionais, tornando-os mais precisos e automatizados. A exploração de novos métodos computacionais e/ou a deteção de outras estruturas morfológicas do crânio é também um objectivo futuro. Por fim, e mais ambicioso, será o desenvolvimento de um método para alinhar e reconstruir zonas do crânio fragmentadas. Além de inovador, é de uma grande mais-valia na prática antropológica onde a fragmentação dos crânios é frequente, dificultando ou mesmo inviabilizando a sua análise.

Há que salientar a extrema importância do crânio na análise antropológica, já que é das peças ósseas mais informativas. Como se referiu brevemente na introdução, através da análise do crânio é possível aceder ao sexo, à ancestralidade e à idade à morte do indivíduo, parâmetros essenciais a qualquer análise antropológica. Além disso, é a estrutura óssea de base da face do indivíduo em vida. Ou seja, um crânio completo permite a reconstrução da face do indivíduo, parâmetro relevante na análise de esqueletos de personagens históricas e de extrema importância na identificação de desaparecidos em casos forenses. Por conseguinte, o desenvolvimento de métodos que permitam uma análise precisa, replicável e cuja taxa de erro é quantificável, mesmo de crânios fragmentados, é fundamental, sobretudo na prática forense, onde os resultados da análise antropológica devem ser particularmente robustos para servirem de prova judicial.

6. AGRADECIMENTOS

Os autores querem agradecer ao Daniel Santos pelo trabalho desenvolvido na sua tese de mestrado que é o ponto de partida para os métodos aqui apresentados. Um agradecimento à iDryas, Grupo Dryas Octopetala, pelo financiamento dos trabalhos de aquisição e processamento dos modelos usados neste trabalho, em particular ao Hélder Santos e ao Miguel Almeida.

Um agradecimento ainda ao Samuel Silva pela sua sugestão e pela sua disponibilidade que foram essenciais para o desenvolvimento dos métodos de deteção de morfologia.

Este trabalho foi parcialmente financiado pela Fundação para a Ciência e a Tecnologia no âmbito do projeto PEst-OE/EEI/UI0127/2014).

7. REFERÊNCIAS

- [Coelho12] C. Coelho. Uma identidade perdida no mar e reencontrada nos ossos: Avaliação das afinidades populacionais de uma amostra de escravos do século XV-XVI. *Dissertação de Mestrado, Universidade de Coimbra*, 2012.
- [Dias13] P. Dias, D. Santos, D. Souza, H. Santos, C. Coelho, M. T. Ferreira, E. Cunha e B. Sousa Santos. A new Approach for 3D Craniometric Measurements Using 3D Skull Models. *Proceedings of 17th International Conference on Information Visualization (IV'13)*, pp. 462-467, Jul 2013.
- [Gonzalez11] P. N. Gonzalez, V. Bernal e S. I. Perez. Analysis of sexual dimorphism of craniofacial traits using geometric morphometric techniques. *Intern-*

tional Journal of Osteoarchaeology, vol. 21, pp. 82-91, 2011.

[Hefner09] J. T. Hefner. Cranial nonmetric variation and estimating ancestry*. *Journal of Forensic Sciences*, vol. 54, pp. 985-995, 2009.

[Oxnard74] C. E. Oxnard. Cranial Variation in Man - Study by Multivariate Analysis of Patterns of Difference among Recent Human Populations - Howells, Wh. *Am J Phys Anthropol*, vol. 41, pp. 349-351, 1974.

[Perini05] T. A. Perini, G. L. Oliveira, J. d. S. Ornellas e F. P. Oliveira. Technical error of measurement in anthropology. *Rev Bras Med Esporte*, vol. 11, no. 1, pp. 86-90, 2005.

[Rhine90] Rhine, S. 1990. Non-metric skull racing. In: Gill, G.W.; Rhine, S. (eds.). *Skeletal attribution of race: methods for forensic anthropology*. New Mexico. Maxwell Museum of Anthropology: 9-20.

[Santos13] D. Santos. Modelos cranianos 3D: nova abordagem cranométrica. *Dissertação de Mestrado, Universidade de Aveiro*, Jul 2013.

[Stephens00] C. Stephens. Measure of Truth? An Examination of the History, Methodology, Theory and Practice of Craniometric Analysis: With Special Focus on its Application in Forensic Identification. *Totem: The University of Western Ontario Journal of Anthropology*, vol. 8, pp. 83-108, 2000.

[Utermohle82] C. J. Utermohle e S. L. Zegura. Intra- and interobserver error in craniometry: a cautionary tale. *Am J Phys Anthropol*, vol. 57, pp. 303-10, Mar 1982.

[Vieira05] M. Vieira e K. Shimada. Surface mesh segmentation and smooth surface extraction through region growing. *Computer-Aided Geometric Design*, vol. 22, pp. 771-792, 2005.

8. APÊNDICE

Neste apêndice são apresentadas as tabelas de valores obtidas para todos os oito crânios. Os valores assinalados com α não foram possíveis de obter devido à fragmentação.

Medida	Crânio #21				Crânio #25				Crânio #38				Crânio #39			
	Antr #1		Antr #2		Antr #1		Antr #2		Antr #1		Antr #2		Antr #1		Antr #2	
	CraMs	M.T.	CraMs	M.T.	CraMs	M.T.	CraMs	M.T.	CraMs	M.T.	CraMs	M.T.	CraMs	M.T.	CraMs	M.T.
MAL	α	α	α	α	54,3	56	53	58	α	α	α	α	51,7	50	50,7	50
MDH	25,4	24	22	22	31,5	31	28,8	32	27,7	26	27,6	27	22,6	20	22,1	22

Medida	Crânio #57				Crânio #65				Crânio #66				Crânio #67			
	Antr #1		Antr #2		Antr #1		Antr #2		Antr #1		Antr #2		Antr #1		Antr #2	
	CraMs	M.T.	CraMs	M.T.	CraMs	M.T.	CraMs	M.T.	CraMs	M.T.	CraMs	M.T.	CraMs	M.T.	CraMs	M.T.
MAL	51,9	55	50,9	54	57,3	57	56,3	57	57,1	56	57,6	60	54	57	53,6	59
MDH	25,4	23	27	22	29,6	31	29,5	32	27,7	28	32,9	33	25,8	27	23,5	25

Medida	Crânio #21				Crânio #25				Crânio #38				Crânio #39			
	Antr #1		Antr #2		Antr #1		Antr #2		Antr #1		Antr #2		Antr #1		Antr #2	
	CraMs	M.T.	CraMs	M.T.	CraMs	M.T.	CraMs	M.T.	CraMs	M.T.	CraMs	M.T.	CraMs	M.T.	CraMs	M.T.
OBB	39,8	40	39,6	38	44,1	47	42,6	43	43,1	45	42,4	42	40,4	43	40,7	40
OBH	31,8	34	32,2	31	35,3	35	34,9	33	39,5	39	36,6	36	30,9	32	29,4	29

Medida	Crânio #57				Crânio #65				Crânio #66				Crânio #67			
	Antr #1		Antr #2		Antr #1		Antr #2		Antr #1		Antr #2		Antr #1		Antr #2	
	CraMs	M.T.	CraMs	M.T.	CraMs	M.T.	CraMs	M.T.	CraMs	M.T.	CraMs	M.T.	CraMs	M.T.	CraMs	M.T.
OBB	42,7	43	43	43	43,2	43	42,9	42	43,9	42	44,1	43	46,5	43	46,6	42
OBH	32,8	33	32,7	32	33,8	34	33,2	33	31,2	32	33	30	33,8	34	34,2	32

Open Source Debuggers and Integration with a 3D Engine

Andre Alexandre Wang Liu António Ramires Fernandes

Departamento de Informática
Universidade do Minho, Portugal

andre.liu507@hotmail.com, arf@di.uminho.pt

Abstract

Debugging is always an important phase in an application life. Applications using 3D hardware accelerated graphics in general, and OpenGL in particular, are usually hard to debug since the computation is split between two processors, each with its own memory space. To be able to debug these applications one has to rely on the API's debugging mechanisms to inspect the output of the computations performed. Several tools are available for this purpose, some of them are open source. Open source tools are great to integrate in already existing projects such as 3D engines since they allow the required customizations. The goal of this paper is twofold. Firstly we provide an overview of the existing open source debugging tools for OpenGL and secondly we discuss the integration of one of such tools in an existing 3D engine.

Keywords

OpenGL; Debugging; GLSL; 3D engine

1. INTRODUCTION

OpenGL applications are prone to error for a number of reasons. Probably the most common reason is the mathematics behind 3D graphics which is complex and hard to trace in a multi stream processor environment such as the GPU. This may require the user to output partial results to output buffers and latter inspect them. This inspection is not straight forward to achieve in a regular debugger since these buffers live in the GPU memory space, and it is up to the programmer to retrieve them. The multi stream nature of the GPU also makes it harder to debug a particular instance of a shader.

Another reason lays on the drivers themselves. While the specification is unique, it is a known fact that there are significant differences between the implementations from the two main hardware makers. For instance, currently, when using uniform blocks NVIDIA accepts the instance block name, while AMD does not. Furthermore, not all features described in the specification are implemented. The same applies to OpenGL extensions, with different drivers having different degrees of implementation completeness. The third issue relates to the silent way drivers deal with many errors. When an error occurs usually life goes on in the application. It is up to the programmer to retrieve the compilation and linkage logs, and check for errors during execution.

Recently, OpenGL has came up with an extension dedicated to debugging [Khronos 14]. The goal is to provide the user with feedback when invalid operations are performed. Although a step in the right direction, it is still far

from perfect. Not all problematic situations are covered by this mechanism, and the debug messages provided by the different hardware manufacturers are far from helpful in most cases.

Each of the vendors provides a debugging tool, at least for Windows operating system. NVIDIA released NSight [NVIDIA 14], and AMD has CodeXL [AMD 13]. Both debuggers can work integrated with Visual Studio. While the list of features is impressive, including shader code tracing, and GPU memory inspection, these tools are not up to date with the latest version of OpenGL. CodeXL claims to support OpenGL 4.3 while NSight only supports OpenGL 4.2. Furthermore, NVIDIA Optimus equipped laptops do not to take advantage of the full list of features available in NSight, namely shader code tracing.

Open source tools on the other hand are not as powerful as they don't allow shader tracing. However, currently some are up to date with the latest OpenGL version and extensions making them useful for users who want to explore the latest features. It is also possible to integrate these tools within an application. Being open source allows for the required customizations to be performed.

This paper explores some of the available open source OpenGL debuggers (section 2). It also discusses the integration of one of such debuggers, GLIntercept, in an OpenGL based application (section 3). The selected application is the Nau 3D engine¹. This engine allows for multi-pass rendering using OpenGL with shaders, and integrates NVIDIA's Optix ray tracing engine and Bullet physics en-

¹<https://github.com/Nau3D/nau>

gine. It is a complex application covering a wide range of OpenGL functionality and therefore should provide a very rich case study. Conclusions are presented in section 4.

2. OPEN SOURCE DEBUGGERS

On the OpenGL wiki [OpenGL.org 12] four open source debuggers are listed:

- GLIntercept
- APITrace
- Bugle
- glslDevil

Recently Valve has released its own debugger, VOGL. Together these are the debuggers explored in this paper. Full sub sections are devoted to debuggers which support the latest OpenGL versions, namely: GLIntercept, APITrace and Bugle.

All these debuggers are based on a library that wraps OpenGL functions. This library replaces the OpenGL library as far as our application is concerned. When an OpenGL function is called, the application is actually calling the homonym function on the debugger library, enabling actions to be triggered before and after dispatching the call to the OpenGL library. The most common of these actions is logging function calls, and all the above mentioned debuggers perform this functionality.

Due to the open source nature of these tools there is no guarantee that future updates will be available when new OpenGL versions come out. Even if the team behind the tool is active there will be a period of time when the debugger is not fully updated. Some of these debuggers have a fallback mechanism to at least report the OpenGL function names called within an application. Nevertheless, for users wanting to try the latest features, it is important to have full support right away, or at least support for the features being tested.

In the next sub-sections some of the particular features of the debuggers will be discussed, namely their updatability.

2.1. GLIntercept

GLIntercept [Trebilco 13] is an OpenGL debugger originally designed for Windows, but a Linux port is available. The main features of interest for modern OpenGL users are:

- Full function logger: logs every OpenGL function call during the application life to a file;
- Frame logging: as an alternative to full frame logging it can selectively log only the function calls between two consecutive swap buffers calls;
- Dump textures and frame buffers to file;

The configuration of the debugger is based on text files and is performed manually. Nevertheless the syntax is simple and examples are provided.

2.1.1. Plugins

This debugger is unique due to the fact that it allows the inclusion of plugin libraries. The source code has its own plugin solution to help users to create them, and source code for several plugins is available. Some note worthy plugins for modern OpenGL are:

- Debug Context: Forces an OpenGL debug context and logs; ARB_debug_output and GL_KHR_debug messages;
- Shader Editor: Provides a text editor to edit and inspect shader code in real-time;
- Function Stats: Statistics regarding function calls;

Plugins allow to override the default functionality of GLIntercept by adding code to be executed before and after an OpenGL function call. It is also possible to define code to wrap render functions and the end of the frame.

Plugins are added to the default GLIntercept config file as follows:

```
Plugins
{
    FunctionStats = ("GLFuncStats\GLFuncStats.dll")
    {
        // plugin options
    }
}
```

The available options are provided in the template config file for each plugin.

2.1.2. Logging

GLIntercept works by overriding calls to `wglGetProcAddress`, wrapping the real function pointer with some code and returning it to the application being debugged. Since the usage of `wglGetProcAddress` is required by any application using post 1.1 OpenGL, this approach ensures that GLIntercept will log all functions. However, functions which are not defined in the header config files of GLIntercept will have their parameters represented by "???", for instance, assuming that `glGetProgramiv` is not defined the log would report `glGetProgramiv(???)`.

An example of a log snippet is:

```
...
glClear(GL_DEPTH_BUFFER_BIT |
        GL_STENCIL_BUFFER_BIT |
        GL_COLOR_BUFFER_BIT)
glDisable(GL_SCISSOR_TEST)
glUseProgram(7)
glUniformMatrix4fv(0,1,false,[1.732051,0.000000,
0.000000,0.000000,0.000000,1.732051,0.000000,
0.000000,0.000000,0.000000,-1.006018,-1.000000,
0.000000,0.000000,9.458376,10.000000])
glUniform4fv(1,1,[0.000000,0.000000,1.000000,1.000000])
...
```

By default, return values are not logged in GLIntercept. This functionality could be partially covered with a plugin that traces `glGet*` functions. Yet, to fully implement such a feature it requires the plugin programmer to identify all GL functions that return meaningful values to the application.

2.1.3. Updating for newer OpenGL versions

As mentioned before, GLIntercept can log all OpenGL functions. However, there is no automatic way of understanding the function parameters. GLIntercept requires header configuration files with the function signatures and constants to provide this information. Currently, GLIntercept is up to date with OpenGL 4.4.

Should a new OpenGL version come out, it is feasible to partially update the header configuration files manually by adding a few functions and constants.

For a full update, GLIntercept has a script for downloading the egl, gl, glx and wgl xml files from Khronos spec repository². It is possible to use the script to create an almost complete GLIntercept header configuration files. A small amount of code tuning is still required due to the fact that OpenGL has constants defined with the same value. For instance in `glxext.h` we can find:

```
#define GL_PIXEL_MODE_BIT 0x0020
//...
#define GL_COMPUTE_SHADER_BIT 0x0020
```

And in the header configuration files we have

```
//gl11_1 include file
enum Mask_Attributes {
    ...
    GL_PIXEL_MODE_BIT          = 0x0020,
    ...
};
//gli4_3 include file
enum Mask_ShaderProgramStages {
    GL_COMPUTE_SHADER_BIT     = 0x0020,
};
```

The conversion of functions also requires tuning to take the different enums into account. For instance, consider the function `glUseProgramStages`:

```
//glxext.h
GLAPI void APIENTRY glUseProgramStages (GLuint pipeline,
    GLbitfield stages, GLuint program);

//gli4_4 include file
void glUseProgramStages(GLuint pipeline,
    GLbitfield[Mask_ShaderProgramStages] stages,
    GLuint program);
```

On the positive side, this updates do not require a rebuild of GLIntercept.

2.2. APITrace

APITrace [Fonseca 13] is a unique tool among this set of debuggers. When the application being debugged is running the debugger limits itself to log all calls producing a binary file with all the information. APITrace's main goal is to replay the log file once the application terminates.

This debugger has the ability to replay its trace files allowing the user to check and verify its current state including the resources and uniforms used for the current function. Using the replay it's possible to dump images to ffmpeg in order to create a video.

Although the log file is binary it can be edited using APITrace's tools allowing the user to resize the file or even change some input. APITrace can read the frame uniforms and shaders by replaying the trace file, it replays each function until it reaches a user defined breakpoint (a user selected function) or the end of the file. Then it grants the user the ability to inspect the application state at that point. When replaying an application, APITrace provides a full GUI application that makes it very easy to use.

As far as logging goes, APITrace supports all OpenGL debugging extensions even if the extensions are not supported by the driver/hardware. In this case the extension functions themselves are not executed but they can be traced.

This debugger is also capable of profiling both CPU and GPU execution times for frames and draw calls.

APITrace has cross-platform compatibility allowing debugging for Linux, Windows, OSX, and Android. Besides OpenGL, APITrace can also work with Direct3D and OpenGL ES. However the profiling features are only available to OpenGL.

2.2.1. APITrace GUI

When replaying a log file, APITrace provides a full GUI application that makes it very easy to use. A log viewing example is shown in figure 1.

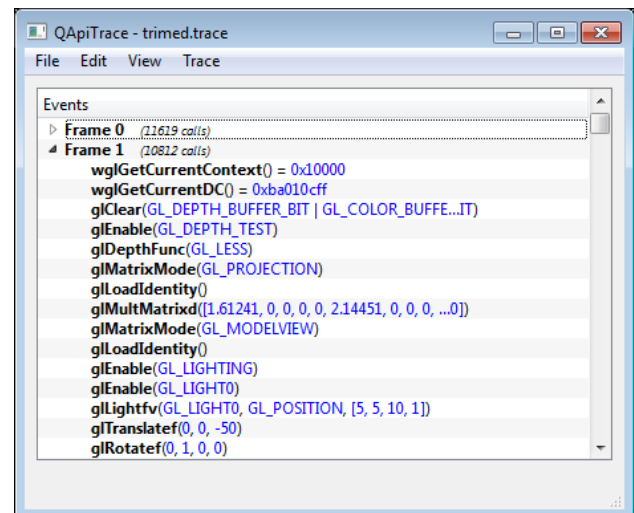


Figure 1. APITrace log viewer.

The GUI provides every bit of information regarding the application status for each draw call, uniform values, shader code, and even a graphical profiling tool.

2.2.2. Updating for newer OpenGL versions

APITrace suffers from similar problems as GLIntercept regarding the update to new OpenGL versions or extensions. Most of the process is automated, and it is based on the official OpenGL header files. Nevertheless, manual tuning is required for parameter types and some function signatures.

Again, while the update could be performed manually for a reduced number of functions, it is unfeasible for someone

²<https://cvs.khronos.org/svn/repos/ogl/trunk/doc/registry/public/api/>

outside the development group to achieve a full significant OpenGL version update.

Plus, when such an update is performed, a rebuild is required.

2.3. Bugle

Bugle [Merry 07] is mostly used in Linux operative systems. It can also be installed in Windows but, as mentioned on the web site, "it is significantly trickier than on UNIX-like systems, and currently only recommended for experts".

Bugle's configuration is based on filters (a set of actions used to extract, manage, or print information). Filters are chained together, like a production line from a factory, the product being debug or profile information.

Bugle comes with several filters that can be configured inside a chain. However, unlike GLIntercept plugins, no method other than changing the C code directly can create new filters. There are filters for tracing, collecting statistics, error logging, taking screen shoots, video capturing, amongst others. An example of a chain is as follows:

```
chain myshowstats
{
    filterset stats_basic
    filterset stats_primitives
    # Note: stats_fragments requires GL_ARB_occlusion_query
    filterset stats_fragments
    filterset stats_calls
    filterset showstats
    {
        show "frames per second"
        show "batches per frame"
        show "calls per frame"
        graph "triangles per second"
        graph "fragments per second"
    }
}
```

This chain will compute several statistics that will be used by the filter `filterset showstats` to display on the debugged application's interface according to the `show` and `graph` listed by the filter. The `filterset showstats` displays data according to the statistics configuration file, for example:

```
"frames per second" = d("frames") / d("seconds")
{
    precision 1
    label "fps"
}
```

2.3.1. Bugle GUI

Bugle displays statistics on top of the application being debugged (figure 2), as well as in its own GUI (figure 3). Bugle's own GUI allows the user to set breakpoints on any OpenGL function and inspect the current OpenGL state, including buffers, textures, framebuffer contents and shaders.

2.3.2. Updating for newer OpenGL versions

Bugle generates its API configuration files using the khronos-api xml files, which can be located in Khronos

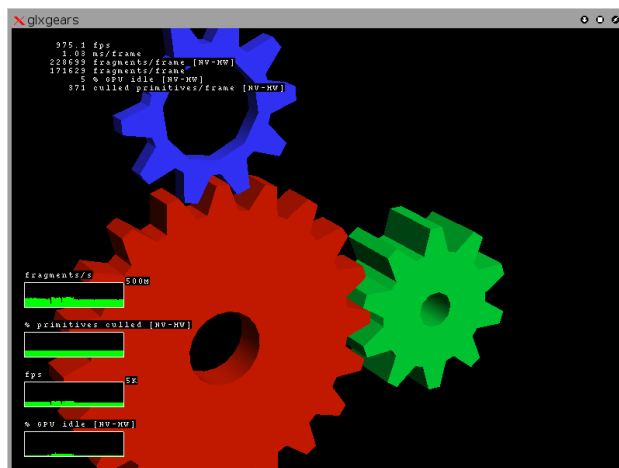


Figure 2. Bugle stats on top of the application.

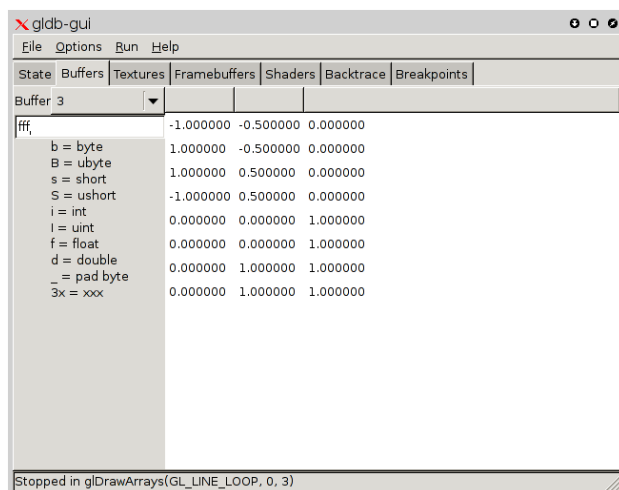


Figure 3. Bugle's GUI.

spec repository, making it the most simple debugger to update. Updating is a fully automated process with no manual tuning required.

On the downside, once the API configuration files are updated, a rebuild is required.

2.4. Other Debuggers

GLSL-Debugger [Hanson 13], a descendant of glslDevil [Klein 10], is the only open-source debugger that provides GLSL tracing capabilities, based on Mesa[Mesa.org 99], an open-source implementation of the OpenGL specification. However, at the time of this writing Mesa only supports GLSL 3.3, hence, it may be of little use for cutting edge developers. This debugger has a complete GUI and a large number of features, but its dependency on Mesa makes it hard to keep up to date with the latest OpenGL versions.

A recent addition to the list of available debuggers is

VOGL [Software 14], Valve’s OpenGL debugger. Still only supporting OpenGL 3.3, but with an interesting list of features that makes it worth while to monitor its development. Perhaps the most interesting feature is the ability to deal with very large trace files efficiently. OpenGL state management allows to efficiently start tracing from the middle of the trace file without having to replay the entire file, as required with APITrace. Trimming and editing of trace files is also considered an important feature and VOGL’s developers are working hard on these tools.

2.5. Comparison table

The following table provides a short feature comparison between the debuggers that have full support for current OpenGL, namely: GLIntercept, APITrace and Bugle.

	GLIntercept	APITrace	Bugle
GUI	no	yes	yes
Log trace	yes	yes	yes
Trace step-by-step	no	no	yes
Replay trace	no	yes	no
OpenGL debug ext.	yes	yes	yes
Trace statistics	yes	no	yes
Runtime statistics	no	no	yes
Show extensions	no	no	yes
Video capturing	yes	yes	yes
Screenshooting	yes	yes	yes
Capture frame log	yes	yes	yes
Capture frame buffer	yes	yes	yes
Application profiling	no	yes	yes
Shader information	yes	yes	yes
Uniform data reading	no	yes	yes
Plugin addition	yes	no	no
Man. tuning GL updates	yes	yes	no
Rebuild on GL updates	no	yes	yes
OS	Win	Win/Unix	Win+./Unix

Table 1. Open Source Applications Feature table

3. INCORPORATING THE DEBUGGER IN NAU

Nau [Ramires] is an open-source 3D graphics engine that works with OpenGL as its graphics API. Nau allows for multipass pipeline definition using XML project files. The material management system is very extensible and flexible allowing for complex pipelines, and enabling it to perform many graphical effects without the need to write a single line of C/C++ code.

Nau works with Optix [NVIDIA 09], from NVIDIA, enabling hybrid rendering algorithms in pipelines that contain both rasterization and ray-tracing passes.

An embedded profiler detailing both CPU and GPU times provides helpful information to fine tune projects.

Composer is an application GUI that works on top of Nau’s library and provides a simple GUI to explore the settings of the current project, allowing for shader recompilation in real time. It also provides information on materials, lights, cameras, and uniform variables.

Nau is continuously being updated to include new OpenGL features and to extend the XML project definition language. Although Nau already provides some debug information, both final users and developers would benefit from having extra debugging information available.

Integrating a debugger into Nau is certainly a path to explore, and this section reports on the progress achieved so far. Being able to have access to the OpenGL calls log file in runtime, will probably save many headaches when writing code for Nau.

Since Composer already provides a GUI, and as both Bugle and APITrace also have a GUI, we opted for GLIntercept. Furthermore, its plugin architecture might reveal very useful in this integration scenario as no source code editing in the debugger is required. Finally, GLIntercept is the only debugger that does not require a rebuild for each OpenGL update, or new extension.

GLIntercept was designed to provide a large amount of information, not only for debugging but profiling as well. Since Nau already contains debugging mechanisms not all of GLIntercept features will be used as they are better covered inside Nau itself. For instance, if enabled GLIntercept profiler records CPU times for every OpenGL call which is not very useful since we are more interested in the GPU times. Nau records both CPU and GPU times per user defined block. GLIntercept also allows the recording of errors, however this occurs with a significant performance cost. We plan to add to Nau the most recent OpenGL debugging extension [Khronos 14]. A GLIntercept plugin is available for this purpose, but implementing it directly in Nau provides more flexibility.

3.1. Changes on GLIntercept

By default GLIntercept’s behaviour can’t be changed when the target application starts. Since the debugger is now integrated in the target application it makes sense to be the application controlling the logging facilities. Therefore, the log is no longer automatically created, it has to be started manually with a function on the target application. Also, the plugins must be activated within the target application.

For the target application to control the debugger it was required to allow to edit in runtime the debugger configuration settings. To achieve this we created a C module integrated in the debuggers library to expose the required functions.

Because the settings can now be edited in run-time several changes regarding removing and adding new settings also had to be made. For instance, plugins alter an internal function table, and now the ability to remove plugins implies methods to clean the function table in order to avoid crashing.

3.2. Changes in Nau

Everything in Nau can be defined in XML project and material definition files. Hence, it makes sense to follow the same principle regarding the debugging options. Therefore, the XML definition was extended to allow the configuration of GLIntercept’s options and plugin usage. No source code editing is necessary to configure debugging options for a particular project.

The `<debug>` tag has been added in order to use the new features. On the absence of this tag, GLIntercept will not

be used by Nau. This allows Nau not to use GLIntercept at all, so the new features are not forced on the user, and are disabled by default, causing no impact relative to the original performance.

3.3. Changes on composer

As mentioned before, Composer is a GUI for Nau. Debugging information such as the log of OpenGL calls can become a powerful debugging tool if provided frame by frame, or even at a finer grain such as pass by pass. To achieve this Composer was expanded to control rendering: pausing rendering, rendering pass by pass, or frame by frame.

Once paused, it will load the txt logfile and split the log in frames and passes. Due to the plugin architecture of GLIntercept it is feasible to generate more info in the future.

3.4. Creating a Project File for Debugging

To activate Nau's debugger all that is required is the addition of the `debug` tag on the project XML file. The children tags are optional and allow the override of the default settings for GLIntercept as defined in `gliconfig.ini`. Without any children or options the default settings will be used.

This section will demonstrate how most of these tags work. Often the mentioned value for the attribute will be some generic type, this means that the value should be of the same type within the quotation mark. The generic values are:

- "bool" so the value should either be "true" or "false";
- "uint" means an unsigned int;
- "string" is for text values.

The `debug` tag has a set of children as in the following code. The `glilog` attribute is optional, when "false" it won't create the function log file.

```
<project>
...
<debug glilog="bool">
  <functionlog>
    ... see functionlog section
  </functionlog>
  <imagelog>
    ... see imagelog section
  </imagelog>
  <framelog>
    ... see framelog section
  </framelog>
  <plugins>
    <plugin>
      ... see plugins section
    </plugin>
    ...
  </plugins>
</assets>
</project>
```

Most of the available options are very similar to the standard GLIntercept's config file, however there are a few exceptions.

3.4.1. functionlog

This will define logging options such as the location of the log file and if the number of frames to be logged should be limited to a certain value to prevent excessively long logs. All these options have default values in `gliconfig.ini`, so the definition of these tags is only required to override the default settings.

```
<functionlog>
  <logmaxnumlogframes value="uint"/>
  <logpath value="string"/>
  <logname value="string"/>
</functionlog>
```

3.4.2. logperframe

This allows the user to select frames to log. If this tag is defined then the logging is not started as soon as the application starts. To start logging the user must press a key, or combination of keys, as defined in the children tags. To log only selected individual frames the tag `logoneframeonly` must have a value true. Otherwise, if `logmaxnumlogframes` is defined then only the defined number of frames will be logged. If none of the mentioned tags are defined then once logging is activated it keeps logging frames until the application is terminated.

```
<logperframe>
  <logoneframeonly value="bool"/>
  <logframekeys>
    <item value="key"/>
    <item value="key"/>
    ...
  </logframekeys>
</logperframe>
```

It should be noted that if `logframekeys` has multiple children then a key combination is being defined, for example `<item value="ctrl"/><item value="f"/>` means that the user needs to press `<ctrl-f>` to create the snapshot.

3.4.3. imagelog

Image log determines if the logger will log textures, which textures and in which format. In this section we can configure which texture types to save, 1D, 2D, 3D and cube textures. In here we can also configure the format in which images are saved, `<imagesavepng>`, `<imagesavetga>`, and `<imagesavejpg>`, are the allowed save formats and `<imageicon>` determines the icon type. Note that if the configuration is set to log all frames this will imply that all the images in the application will be saved every frame. This feature is designed to function with logging of single frames for performance reasons.

```
<imagelog>
  <imagesave1d value="bool">
  <imagesave2d value="bool">
  <imagesave3d value="bool">
  <imagesavecube value="bool">
  <imagesavepng value = "bool">
  <imageicon>
    ...
  </imageicon>
</imagelog>
```

3.4.4. framelog

This will save the frame buffer's pre/post/diff state on an additional frame folder. While in GLIntercept's `gliConfig.ini` the pre/post/diff flags are one single attribute, here they are 3 separate booleans (for example `ColorBufferLog` is now: `<frameprecolorsave>`, `<framepostcolorsave>`, `<framediffcolorsave>`).

There is also the possibility of creating a movie with the frame buffer, including all the specified buffers. For the stencil buffer it is possible to specify a color table for stencil values, for instance if the stencil value is 1 the color could be `#FF0000` (red).

```
<framelog>
  <framepostcolorsave value = "bool">
  <frameimageformat value="string"/>
  <framestencilcolors>
    <item value="uint"/>
    <item value="uint"/>
    ...
  </frameStencilColors>
  ...
  <frameicon>
  ...
  </frameicon>
  <framemovie>
  ...
  </framemovie>
</framelog>
```

3.4.5. plugins

Adding a plugin is slightly different on Nau because of the configuration format.

GLIntercept plugins require at least the plugin name and the location of its DLL. For example in `glicongif.ini` we could write:

```
FunctionStats = ("GLFuncStats.dll")
```

Some plugins can fit extra parameters and these extra parameters should be placed as described in the example, their format is the same as the GLIntercept's config file. These parameters can be found in the `config.ini` file of the plugin. Converting this information for a Nau project results in:

```
<plugins>
  <plugin name="FunctionStats"
    dll="GLFuncStats.dll">
    parameter1 = "parameter1 value";
    parameter2 = "parameter2 value";
    ...
  </plugin>
  ...
</plugins>
```

3.4.6. WORKING WITH COMPOSER

A debug menu was added to Composer. In order to access the debug options it is necessary to Pause Composer. Once paused the composer will start reading the GLIntercept main log file (if available) and fetch program data. This may take a while for large log files.

Once paused, the GLI Log, Program Info and Buffer Info options become available. Navigation op-

tions include executing the next pass, go until the end of the frame, or execute a full frame (see figure 4).

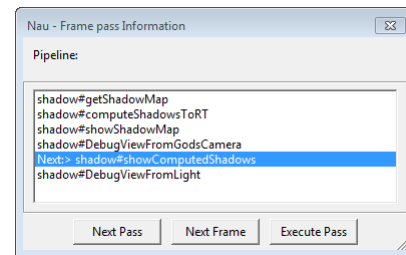


Figure 4. Nau's pass control.

The log window shows the function call log and stats per frame, see figure 7.

Program Info will provide information for each program including the actual uniform values, see figure 6. Composer already provides the values that uniforms *should* have according to the project file. Having the actual values allows developers to check if the uniforms are correctly being set.

The buffer information window allows the visualization of buffer contents. The user may specify the number of atomic elements and their type per struct of the buffer.

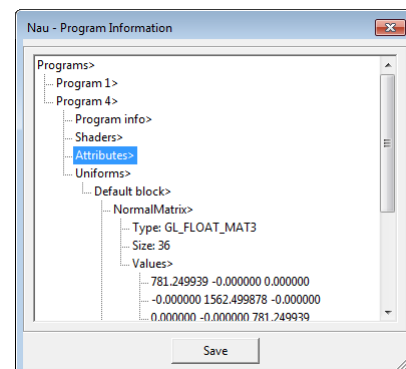


Figure 5. Nau's program uniforms.

4. CONCLUSIONS AND FUTURE WORK

Current Open Source debuggers, while not covering all desired features, can complement each other. While there is still a lot of work ahead, the debuggers presented in here are already very useful tools for OpenGL developers.

Nevertheless, actual shader debugging remains an issue. Proprietary debuggers like NSight or CodeXL, and GLSL Debugger are the only debuggers capable of this feature and even they are not fully up to date with the latest OpenGL version and extensions. A software implementation by Khronos could help to solve this issue.

Integrating GLIntercept with Nau turned out to be a much more simple task than expected. GLIntercept's code itself is simple to read once the programmer knows where

Type	INT	INT	BYTE	BYTE	BYTE	BYTE
1	0	1	2	0	0	0
2	3	4	5	0	0	0
3	6	7	8	0	0	0
4	9	10	11	0	0	0
5	12	13	14	0	0	0
6	15	16	17	0	0	0
7	18	19	20	0	0	0
8	21	22	23	0	0	0
9	24	25	26	0	0	0
10	27	28	29	0	0	0

Figure 6. Nau's buffer information.

Figure 7. Nau's log information.

to start looking and all that was required was to export the configuration functions.

On Nau's side, due to the highly modular architecture of the 3D engine, it was painless to introduce all the new features. Although we have implemented a substantial set of features, much remains to be done to achieve a really helpful debugging environment. For instance, as we know which values the GL uniforms *should* have, and the values actually set on those uniforms, an automatic verification matching procedure is a goal to be achieved.

State inspection is another feature we are working on. The goal is to be able to configure which OpenGL state variables are relevant and to be able to inspect them when the application is paused.

Regarding the integration, not all information generated by plugins is used by Nau, being available on text format only. Integrating this information in Nau's GUI, Composer, could be a helping hand in some debugging scenarios.

5. ACKNOWLEDGEMENTS

Work partially funded by National Funds through FCT - Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project PEst-OE/EEI/UI0752/2011

References

- [AMD 13] AMD. Codexl, November 2013. [Online - accessed 1 August 2014].
- [Fonseca 13] José Fonseca. Apitrace. GitHub, November 2013. [Online - accessed 1 August 2014].
- [Hanson 13] Hanson e Chris 'Xenon'. Glsldbg, October 2013. [Online - accessed 1 August 2014].
- [Khronos 14] Khronos. Khr_debug extension spec, March 2014.
- [Klein 10] Thomas Klein, Magnus Strengert, e Thomas Ertl. Glsldbg - opengl glsl debugger. Glsldbg. <http://www.vis.uni-stuttgart.de>, February 2010. [Online - accessed 1 August 2014].
- [Merry 07] Bruce Merry. OpenGL software development kit. BuGLe. OpenGL, 2007. [Online - accessed 1 August 2014].
- [Mesa.org 99] Mesa.org. Mesa 3d graphics library, January 1999. [Online - accessed 1 August 2014].
- [NVIDIA 09] NVIDIA. Nvidia optix ray tracing engine, 2009. [Online - accessed 1 August 2014].
- [NVIDIA 14] NVIDIA. Nvidia nsight visual studio edition. NVIDIA Nsight Visual Studio Edition. NVIDIA, 2014. [Online - accessed 1 August 2014].
- [OpenGL.org 12] OpenGL.org. Debugging tools. OpenGL.org, March 2012. [Online - accessed 1 August 2014].
- [Ramires] António Ramires. Nau - opengl + optix 3d engine. [Online - accessed 8 June 2014].
- [Software 14] Valve Software. Valve opengl debugger, May 2014. [Online - accessed 1 August 2014].
- [Trebilco 13] Damian Trebilco. Glinters - opengl call interceptor/logger. GLinters. Code.google.com, June 2013. [Online - accessed 1 August 2014].

Visualização de Música à Distância de um Gesto

João Tiago Gomes Maria Beatriz Carmo Ana Paula Cláudio
LabMAG, Faculdade de Ciências, Universidade de Lisboa
Campo Grande 1749-016
fc37415@alunos.fc.ul.pt, {bc,apc}@di.fc.ul.pt

Sumário

Este artigo apresenta uma aplicação que tem como objetivo dar a qualquer pessoa a possibilidade de criar música partindo de vários sons, que podem ser progressivamente compostos em níveis crescentes de complexidade. Inspirada no instrumento eletrónico Theremin, a interação será feita de modo a criar ou manipular música de uma forma natural. A composição de música é acompanhada de uma visualização tridimensional, que não só serve de guia ao utilizador no processo de composição, como também permite visualizar a música que está a ser criada.

Keywords

Visualização, música, composição musical, leap motion, interação natural

1. INTRODUÇÃO

Lev Sergeevich Termen patenteou em 1928 um instrumento musical eletrónico com o nome *Theremin*, que era controlado pelo músico sem qualquer contacto físico [Theremin28]. O instrumento dispõe de duas antenas que controlam respetivamente o *pitch* (ou frequência) e a amplitude (ou volume) do som. O músico controla cada antena com uma mão, afastando ou aproximando as mãos das antenas de modo a fazer variar o som do instrumento.

A 21 de Maio de 2012 foi anunciado pela empresa Leap Motion um controlador, com o mesmo nome, constituído por duas camaras infravermelhas e três LED infravermelhos. Este dispositivo permite fazer o *tracking* dos dedos e das mãos de um utilizador de forma extremamente precisa (precisão até 1/100 de milímetro) num espaço mais ou menos hemisférico com 1 metro de alcance [Leap].

Em meados de 2013 iniciou-se a comercialização do dispositivo Leap Motion e desde então surgiram aplicações em diversos domínios, como, por exemplo, na área da medicina ou jogos interativos, que tiram partido desta nova forma de interação. Em particular na área musical foi publicado, na loja da Leap Motion (Airspace [Airspace]), um considerável número de aplicações, grande parte das quais se concentrava em tornar a Leap Motion um controlador MIDI. No entanto este trabalho, ainda que inserido na área musical, segue outro rumo.

As semelhanças entre um *Theremin* e um controlador Leap Motion são óbvias. Assim surgiu a ideia de usar o dispositivo Leap Motion não só para tocar um instrumento virtual que produza apenas um tipo de som, tal como acontece com o *Theremin*, mas também para

tocar vários sons de vários tipos diferentes ao mesmo tempo, servindo assim de ferramenta de composição musical. O propósito deste trabalho é, então, sobrepor vários sons de modo a criar uma música. A composição da música e a sua complexidade ficam a cargo do utilizador.

O processo de produzir vários tipos de sons, quer seja através da voz quer seja através de um instrumento, e de juntar em tempo real esses sons de modo a que eles formem uma música não é algo novo. Vários músicos contemporâneos têm seguido esta abordagem só possível com recurso à tecnologia (hardware e software) que existe hoje. É importante também referir que esta tendência não se prende a um estilo musical. Existem hoje artistas das mais variadas áreas que utilizam este processo para produzir música, em áreas que vão *do Hip Hop* [Dub FX] à música erudita [Zoe Keating], passando pelo *Rap, Pop, Soul* [Bernhoft], *RnB* e *Jazz* [Tom Thum].

O objetivo deste trabalho foi desenvolver uma aplicação que reunisse três componentes: em primeiro lugar, a vertente musical, que possibilita ao utilizador tocar vários sons de diferentes formas e compô-los de modo a criar música fácil e interactivamente; em segundo lugar, a utilização do dispositivo *Leap Motion* como interface para a criação da música na aplicação, tomando especial atenção à forma como esta interação é feita, de modo a que seja o mais natural possível; e, finalmente, a visualização da música de forma a distinguir claramente os vários sons utilizados na composição. Estes três módulos, embora distintos, têm de funcionar em conjunto para proporcionar uma boa experiência de utilização. Isto torna-se particularmente relevante pelo facto de estarmos a trabalhar com um dispositivo de interação, que não o rato e o teclado, com o qual a maior parte dos utilizadores

não está familiarizada. Uma vez que estamos a utilizar o dispositivo Leap Motion, temos que ter em conta novos problemas e desafios que não encontramos com as interfaces tradicionais, que já são estudadas há bastante tempo.

Nesta secção foi dada uma pequena introdução aos vários temas que foram usados como inspiração para a aplicação. Na secção 2 é apresentado o trabalho relacionado com o descrito neste artigo. Na secção 3 são abordados os vários componentes que constituem a aplicação. Na secção 4 são apresentadas as principais conclusões. E finalmente na secção 5 são sugeridos melhoramentos futuros.

2. TRABALHO RELACIONADO

2.1 Processo Musical

Foi analisado o trabalho de vários artistas musicais (*DubFX*, *Zöe Keating*, *Tom Thum* e *Jarle Bernhoft*) que fazem música sozinhos recorrendo a vários instrumentos e sons para tentar perceber o processo utilizado por eles. Há aspectos comuns a todos estes artistas neste processo, variando em alguns detalhes, nomeadamente no nível de complexidade da música criada e no nível de preparação prévia necessária. O processo consiste no seguinte: em primeiro lugar, o artista dispõe sempre de equipamento (hardware) e/ou tecnologia (software) que lhe permite gravar várias *tracks* (em português, faixas, mas é importante não confundir com o termo faixas que é usado coloquialmente para nos referirmos a uma música de um álbum). O artista começa por gravar numa *track* um som, quer seja ele vindo da sua própria voz quer seja de um instrumento. Após a gravação, esse som começa a ser reproduzido imediatamente em *loop*. De seguida é gravado outro som noutra *track*, enquanto o anterior está a ser reproduzido. Este processo vai-se repetindo, de forma a que as várias *tracks* se vão sobrepondo para formar uma música cada vez mais complexa. A música produzida com este processo pode ser bastante complexa, com várias *tracks* sobrepostas e a reproduzirem vários sons, ou algo simples com poucas *tracks*: cabe ao artista decidir.

2.2 Visualização de Música em Tempo Real

Foram analisadas aplicações que têm a capacidade de apresentar uma visualização em tempo real de uma música que está a ser tocada. Verificou-se que em alguns casos a visualização é específica para uma dada música e é necessário alterar o código fonte para a substituir por outra. No entanto, a visualização dessa música é feita em tempo real.

As aplicações analisadas fazem uso da informação acerca das frequências do som num dado momento para construírem a respetiva visualização. O exemplo mais comum disto é um simples equalizador de barras que se encontra em inúmeros aparelhos de música.

Foram analisadas 4 aplicações com visualizações em 3D (*Loop Waveform Visualizer [Loop]*, *Music Colour Particles [Color Particles]*, *Cube Visualizer [Cube Visualizer]* e *A dive in Music [Dive]*). Três delas possuem

apenas uma visualização disponível e são, por isso, mais simples. Estas aplicações mais básicas fazem uso de visualizações que são já bastante conhecidas, tal como uma simples onda de som (figura 1), e transformam-na numa versão 3D.

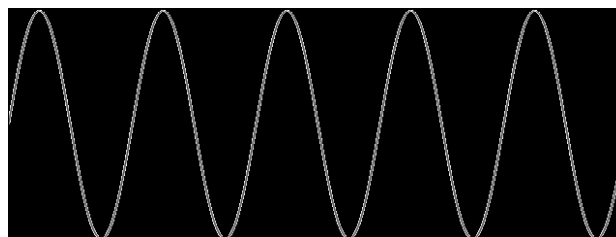


Figura 1 - Onda de som

Por outro lado, apesar de serem mais simples, estas aplicações permitem que se tenha uma clara noção da relação entre o som e a respetiva representação gráfica. Quando passamos para aplicações mais elaboradas visualmente, esta relação tende a perder-se. Caso disso é a aplicação *A dive in Music* que tem mais de 20 visualizações bastante complexas e que envolvem emissores de partículas. O resultado final que se obtém é mais agradável esteticamente mas perde-se a relação que existe entre som e imagem.

2.3 Visualização de Música com Pré-processamento

A vantagem clara da visualização de música com pré-processamento em relação às visualizações feitas em tempo real é que o resultado gráfico pode ser muito mais rico e pode usar mais poder de processamento para gerar a imagem final, não só relativamente à visualização, como à extração da informação da música que servirá de *input* para a visualização. Assim, e comparando, por exemplo, a visualização anteriormente referida *A dive in Music* com *Ljósið [Ljósið]*, é notória a diferença na qualidade das partículas usadas e no efeito visual que esta visualização causa.

Note-se que ambas as visualizações recorrem a partículas; no entanto, em *Ljósið* é possível distinguir claramente a que som pertence cada visualização. O mesmo não acontece no exemplo anterior. Uma explicação possível para que isto aconteça prende-se com a natureza pré-processada da visualização e o tempo e esforço que pode ter sido despendido a fazer uma separação mais cuidada dos sons individuais de cada instrumento (piano e violino), em vez de usar simplesmente o som como um todo, tal como é feito nas aplicações em tempo real. Outro exemplo de visualização de música com pré-processamento foi desenvolvido por Christopher Garcia [Garcia11]. Este autor, a partir da análise da música, gera um ficheiro de texto que serve de *input* à geração da visualização.

2.4 Visualização e criação de música simultaneamente

Nesta secção são abordadas algumas aplicações que além de gerarem uma visualização da música, tal como as anteriores, têm também a capacidade de permitir ao

utilizador criar música. Isto muda o propósito da aplicação, se compararmos com as anteriormente referidas, pois agora o foco é na criação da música e não simplesmente em assistir a uma visualização. A interatividade desempenha, portanto, um papel central.

Foram analisadas aplicações com visualizações tanto em 2D como em 3D mas todas eram relativamente parecidas (*ToneMatrix* [ToneMatrix], *Beat Petite* [BeatPetite] e *ToneCraft* [Tone Craft]). Nestas aplicações é possível tocar vários sons em diferentes momentos, sendo que o metrónomo que marca o tempo é um elemento bastante presente e notório pois é ele que dá estrutura à música. Em cada uma das aplicações é possível tocar um total de 16 notas até que o tempo volte ao início (encontrando-se o tempo num *loop* infinito). Em três das aplicações analisadas os elementos gráficos estão dispostos de acordo com uma grelha regular bidimensional. Nessas grelhas o eixo dos *xx* corresponde ao tempo e o eixo dos *yy* corresponde ou a um som diferente ou, tratando-se do mesmo som, um *pitch* diferente desse som, o que corresponde na prática a uma nota diferente que é tocada.

Existem neste conjunto de aplicações dois pontos aos quais vale a pena prestar alguma atenção. O primeiro é o facto de existir um tempo subjacente e que regula o momento a que as notas são tocadas, um metrónomo.

Este metrónomo tem a particularidade de, quando acabam de tocar as 16 notas possíveis, voltar ao início e começar a tocar de novo as mesmas notas. Isto produz um efeito interessante que, à partida, pode passar despercebido. O facto de existir uma repetição constante faz com que quaisquer que sejam as notas tocadas, a música que daí resulta soa agradavelmente. O segundo ponto prende-se em particular com a aplicação *ToneMatrix*, que, para além de fazer uso do metrónomo e da repetição, tal como foi referido anteriormente, junta uma outra funcionalidade, o ligeiro atraso entre cada batida do metrónomo. A repetição e o atraso tornam a sonoridade da música semelhante a um andamento de jazz.

3. TRABALHO DESENVOLVIDO

O trabalho desenvolvido envolveu três componentes principais: a) áudio, que trata da reprodução e gravação dos vários sons, da combinação dos sons gravados e da marcação do tempo da forma mais exata possível; b) interação do utilizador, com o recurso ao controlador Leap Motion, para a composição de música e a interface gráfica que auxilia a construção e a composição da música; c) visualização da música criada.

O trabalho foi desenvolvido com recurso à linguagem *javascript* e para o browser *Google Chrome* que é o único browser neste momento que suporta todas as funcionalidades de HTML5 necessárias para a aplicação funcionar. A aplicação resultante ficará disponível numa página web, que, uma vez *online*, permitirá que qualquer pessoa possa aceder à aplicação de forma rápida e fácil, não sendo preciso fazer qualquer *download* adicional. Esta foi uma das decisões que mais pesou na escolha das tecnologias a usar para desenvolver este projeto. Deste

modo, é apenas preciso possuir o dispositivo Leap Motion e aceder a um endereço *web* para usufruir da aplicação.

3.1 Audio

A componente de áudio foi a que se demonstrou mais desafiante. Nesta componente podemos encontrar duas vertentes. Por um lado desenvolver um metrónomo para a aplicação ser capaz de marcar o tempo de forma exata. Por outro lado, disponibilizar vários sons, organizar as várias *tracks* que compõem a aplicação e lidar com o processo de reprodução e gravação de *tracks*.

Tendo em conta a tecnologia utilizada, foi difícil conseguir acomodar estes requisitos e ao mesmo tempo ter em conta aspetos de desempenho, isto é evitar que os atrasos no som sejam perceptíveis aos utilizadores.

3.1.1 Metrónomo, marcação de tempo e desempenho

Ao começar o módulo de áudio foi claro que a marcação do tempo e a necessidade de construir um metrónomo que fosse capaz de o contar de forma precisa e igualmente espaçada seria crucial. Assim, e nunca esquecendo que toda a aplicação é em tempo real, pensou-se desde o início nas implicações em termos de desempenho. Desde logo pelo facto de a aplicação correr num *browser*, fica limitada em termos de precisão do seu relógio. O valor de atraso do relógio pode ser significativo quando se trata de som. Por isso, foram usadas várias técnicas para tentar mitigar os efeitos do atraso do relógio do *Google Chrome*, *browser* para o qual a aplicação foi desenvolvida.

A forma como o metrónomo funciona na aplicação é a seguinte: com base no BPM (*beats per minute*) e no número de notas que é possível dar num dado compasso, é definido um número de milissegundos de intervalo entre cada “batida” da música; cada vez que passa esse intervalo de tempo, o metrónomo “acorda” e sabe que necessita de efetuar uma série de ações; essas ações em *javascript* traduz-se na execução de uma função *callback*. Esta função é responsável por: a) começar a gravar uma *track*, caso tenha sido efetuada essa ação pelo utilizador; b) terminar de gravar uma *track*; c) atualizar o contador que é responsável por saber quando parar de gravar uma *track*; d) reproduzir som se o utilizador quer tocar o som de determinada *track*; e) reproduzir os sons das *tracks* base; f) caso haja alguma *track* com algum som já gravado e se for o momento de reproduzir esse som, de facto reproduzi-lo g) por último, acertar os valores da contagem das notas que servem para marcar o tempo (estes valores são apresentados na interface gráfica e serão explicados numa secção posterior).

Como já foi referido anteriormente, existe um atraso de alguns milissegundos no relógio das aplicações que usam o *browser*. Isto faz com que o intervalo de tempo que passa entre cada “acordar” do metrónomo não seja igual. O efeito cumulativo de todos os atrasos causa, naturalmente, problemas acrescidos em termos de desempenho.

Por esta razão, o que se fez para resolver este problema foi ajustar o número de milissegundos que leva o próximo “acordar” do metrônomo, não para o seu valor esperado, mas sim para um valor que tenha em conta o atraso anterior. Por exemplo, supondo que o metrônomo deveria “acordar” no instante 100 mas que, devido ao atraso, apenas acorda no instante 104, e supondo ainda que o intervalo entre cada “acordar” do metrônomo é de 100ms, então o próximo instante seria provavelmente o 208 (104 do instante original mais os 100ms de intervalo mais o atraso que este segundo “acordar” teria, $104 + 100 + 4 = 208$). Ora, este não é o efeito pretendido. O esperado seria o próximo “acordar” do metrônomo aos 200ms ou, na pior das hipóteses, no instante 204, que tem apenas o erro deste instante e não o erro acumulado do “acordar” inicial.

Para conseguir os tempos o mais regulares possível foram usadas algumas técnicas. A primeira foi ter em conta o atraso do *callback* passado e chamar o próximo com uma correção. No exemplo anterior, isto queria dizer que o segundo *callback* seria chamado não com 100ms de atraso mas sim com 96ms (100ms menos 4ms de atraso do *callback* anterior). Assim elimina-se o erro cumulativo e fica-se apenas com o erro de cada *callback*, sendo este ajustado para o próximo *callback* que é feito.

A técnica descrita acima reduz para níveis aceitáveis o erro de relógio entre as chamadas de *callback*. Mas dentro da função *callback* que é sistematicamente chamada é preciso também ter cuidados. Nesta função são feitas operações como começar a gravar, parar de gravar, tocar o som de várias *tracks*. Se estas operações não forem feitas em sequência resultam em discrepâncias no áudio posteriormente ouvido pelo utilizador. Assim, existe uma zona crítica na qual as operações têm de ser feitas consecutivamente. Para isso, é necessário que as operações não críticas sejam previamente preparadas.

Testou-se a utilização de *threads* para esta parte do áudio, mas verificou-se a sua ineficiência, pois a comunicação entre a *thread* principal e a secundária implicaria demasiado atraso, e isto seria inaceitável numa aplicação que corre em tempo real e que trata áudio.

3.1.2 Tracks

A aplicação é constituída por várias *tracks*, tocando cada uma um som específico. Esse som é obtido através de um ficheiro *.wav* carregado quando a aplicação inicia e que contém o som de um instrumento ou uma parte de um instrumento (maioritariamente sons de bateria: bombo, tarola, pratos). Existem tipos diferentes de *tracks*: as *tracks* base e as *tracks* normais.

A diferença entre as duas é o nível de controlo que o utilizador tem sobre elas: nas *tracks* base o utilizador não tem qualquer controlo, elas são tocadas automaticamente desde que a aplicação é iniciada até que é encerrada. O objetivo destas *tracks* é dar ao utilizador uma base sobre a qual trabalhar. Presentemente existem 2 *tracks* base,

uma que marca o ritmo com o som de um bombo de uma bateria e uma que dá um som envolvente e contínuo. Este som contínuo é sintetizado em tempo de execução, usando geração aleatória de sons limitados a um dado intervalo de frequências.

Passando para as *tracks* normais, estas podem ser controladas pelo utilizador, concretamente através do dispositivo Leap Motion. Será explicado em mais detalhe na secção seguinte como é que é feita esta interação.

Os sons para cada uma das *tracks* foram escolhidos tendo em conta como soam isoladamente e em conjunto e – o que é mais importante – pela sua duração. Tal como já foi explicado anteriormente, existe um intervalo de tempo específico que separa as chamadas consecutivas de um *callback* e os sons, ao serem escolhidos, não podem exceder esse limite para que não se sobreponham uns aos outros.

Finalmente, o som envolvente encontrado numa das *tracks* base é o único que não é real mas sim sintetizado. Foi usado código de uma biblioteca para obter este som. A forma como o som é obtido envolve o uso de vários *Biquad Filters* [Biquad Filter] sobrepostos (por defeito são usados 20) e a utilização de um *Panner* [Panner] para mudar a posição de onde o som vem, o que permite que o som ouvido não seja monótono e pareça variar, produzindo um efeito de envolvimento.

3.1.3 Gravar Som

Para gravar o som foram tentadas várias abordagens. A primeira foi gravar o som usando uma biblioteca que utilizava nós *javascript* e *threads*. O resultado não obtinha a precisão necessária para começar a gravar nem para parar num momento exato, devido à latência inerente às *threads* em *javascript* e aos nós *javascript* que são usados na *Web Audio API* [Web Audio API].

Depois de várias iterações de tentativa e erro, chegou-se finalmente a uma solução aceitável em termos de desempenho: não gravar o som que saía de determinada *track* e sim gravar o momento em que o som era tocado. Isto significa que na prática o que acontece é: quando a aplicação está em modo de gravação, sabe que está a gravar determinada *track* e sabe quando é tocado um som; assim, é gravado num *array* um valor booleano, se o valor for verdadeiro então, quando for altura de reproduzir o som posteriormente, este é reproduzido, se o valor for falso, o som não é reproduzido.

3.2 Interação

A interação com o utilizador é realizada através de duas componentes que se complementam: uma baseada na Leap Motion e outra na interface gráfica.

3.2.1 Interação com Leap Motion

A interação entre o utilizador e a aplicação no que respeita a composição da música e dos vários sons que se encontram nas várias *tracks* é feita através do dispositivo Leap Motion (figura 2).



Figura 2 - O dispositivo Leap Motion

Ao utilizar-se este dispositivo para fazer tocar sons e para controlar a parte musical da aplicação, foi necessário ter em conta novos aspetos de interação que são radicalmente diferentes de, por exemplo, um rato e teclado.

Na aplicação o utilizador consegue controlar sons de ritmo, ou seja, sons que não se prolongam no tempo, por exemplo, o som de um tambor de uma bateria. Um contra-exemplo disto seria o som de um violino, onde o músico pode prolongar o som de uma nota por algum tempo. Assim, visto que estes sons terminam rapidamente, escolheu-se dar ao utilizador a possibilidade de controlar dois parâmetros relativamente a eles: primeiro, o número de vezes que é tocado (tendo em conta os BPMs da aplicação), considerando sempre que o momento tocado é controlado pelo metrónomo da aplicação para que os sons “encaixem” todos no mesmo sítio; em segundo lugar, o volume do som.

Uma vez selecionada uma *track* e conseqüentemente um som, o utilizador pode tocar esse som. A mão esquerda controla o número de vezes que o som é tocado. Opcionalmente, a mão direita pode controlar o volume do som que está a ser tocado. Para conseguir identificar as várias posições das mãos foram definidos limites, tanto horizontais como verticais. Assim, definiu-se um plano vertical, paralelo aos eixos dos *yy* e dos *zz* colocado ao meio da imagem e que separa as zonas de ação de cada uma das mãos (figura 3). Para a mão esquerda foram definidas verticalmente quatro zonas, correspondendo a cada uma ações distintas: a mais alta impede que seja tocado qualquer som (*z1*); a seguinte toca o som a uma velocidade normal, e, de acordo com a batida do metrónomo (*z2*); abaixo desta o som toca a uma velocidade duas vezes maior que o normal (*z3*); e finalmente a última toca o som a uma velocidade quatro vezes superior à normal (*z4*).

A velocidade a que o som é tocado está diretamente relacionado com os BPMs do metrónomo e do número de notas que é possível tocar, bem como os momentos em que podem ser tocadas.

Na tabela 1 assinalam-se com “x” as células correspondentes aos momentos em que os sons podem ser reproduzidos, consoante a zona (*z1*, *z2*, *z3* ou *z4*) em que a mão esquerda do utilizador se encontra.

A mão direita controla o volume do som tocado. Para esta mão foram igualmente definidos limites que representam diferentes ações. Assim, as zonas verticais utilizadas para a mão direita consistem no seguinte: a zona mais alta corresponde ao volume mais alto (80%), a zona mais baixa corresponde ao volume mais baixo (10%) e na zona intermédia é feita uma proporção entre as coordenadas da mão do utilizador e o volume. De notar que as zonas nos extremos, mais alta e mais baixa, ocupam cada uma 25% do espaço e a zona intermédia ocupa os restantes 50% do espaço.

Tal como foi mencionado anteriormente, o utilizador pode escolher utilizar apenas a mão esquerda (a que controla se o som é tocado ou não) e se assim for o volume do som corresponde ao último volume utilizado pela mão direita ou, caso a mão direita nunca tenha sido utilizada naquela *track*, é usado um volume por omissão.

Uma vantagem que advém disto é a possibilidade de o utilizador escolher o volume desejado para tocar determinado som com a mão direita e depois usar unicamente a esquerda para escolher os momentos exatos em que quer que o som seja tocado.

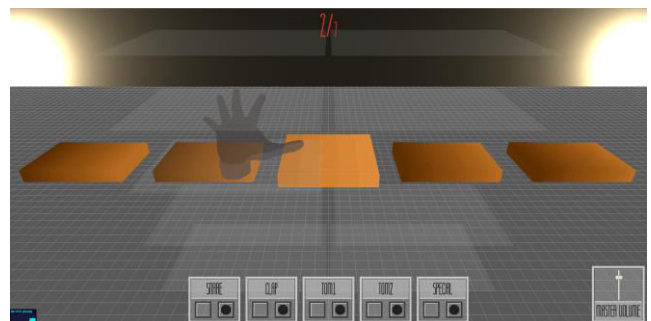


Figura 3 - Imagem da aplicação sem nenhuma *track* selecionada

<i>z1</i>																
<i>z2</i>	x				x				x				x			
<i>z3</i>	x		x		x		x		x		x		x		x	
<i>z4</i>	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Zona vs tempo	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Tabela 1 –Momentos em que os sons podem ser reproduzidos em cada zona de interação

Na figura 3 é possível ver a mão virtual de um utilizador que está a usar a aplicação. São visíveis também os limites discutidos anteriormente. Cinco planos dividem o lado esquerdo de modo a criar quatro zonas e quatro planos dividem o lado direito para criar três zonas. De notar, no entanto, que na imagem nenhuma *track* está seleccionada, e por isso tanto os limites como a mão do utilizador encontram-se preenchidas com uma cor neutra. Assim que uma *track* é seleccionada, tanto as mãos virtuais do utilizador como os limites mudam de cor para dar ao utilizador *feedback* acerca do que está a acontecer (figura 4).

No início da implementação da aplicação não existiam ainda as mãos virtuais para guiar o utilizador no ecrã. Existiam, no entanto, as mesmas zonas descritas anteriormente, delimitadas pelos mesmos limites. Constatou-se que era muito mais difícil para o utilizador colocar a mão na zona desejada, pois tinha apenas o *feedback* da visualização e o auditivo para se guiar (note-se que o *feedback* da visualização não engloba as mãos virtuais do utilizador, assunto abordado em maior detalhe na próxima secção). A partir do momento em que as mãos virtuais foram adicionadas à aplicação, tornou-se bastante mais simples a orientação do utilizador, pois o *feedback* que obtinha era relativo à localização das suas mãos no espaço virtual.

Outra consequência foi a melhoria na precisão. Antes da introdução das mãos virtuais era normal passar de uma zona para outra sem querer, obrigando o utilizador a olhar constantemente para as suas mãos reais para tentar mantê-las no mesmo sítio. Após a introdução das mãos virtuais o utilizador guia-se maioritariamente por elas e olha bastante menos para as suas mãos reais. Este facto sugeriu a possibilidade de ter ainda mais uma ou duas zonas, totalizando cinco ou seis, sem que isso se traduzisse numa perda de precisão das mãos. A hipótese, no entanto, teria de ser corroborada por testes com utilizadores efetivos.

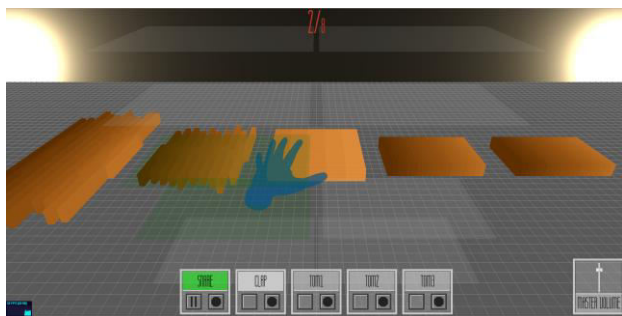


Figura 4 – Interação com uma *track* seleccionada

3.2.2 Interface gráfica

A interface gráfica localiza-se na zona inferior da imagem.

À direita existe um slider que permite regular o volume de som. Ao centro estão desenhadas áreas rectangulares,

cada uma delas associada a uma *track*. O utilizador, ao seleccionar um dos rectângulos, inicia a reprodução do som da *track* correspondente. Em cada rectângulo existem dois botões, um para gravar e outro para fazer *pause* ou *play*, consoante a ação corrente. Ao carregar no botão de gravar a cor da metade superior do rectângulo muda para laranja, indicando que a aplicação se está a preparar para gravar (o início da gravação dá-se no início de cada compasso, o utilizador pode guiar-se pelos números do metrónomo explicados anteriormente para saber quando é que a aplicação vai começar a gravar). Quando a gravação é iniciada, a cor da metade superior do rectângulo muda para vermelho. Uma vez acabada a gravação, a cor muda novamente para verde para indicar que a *track* está simplesmente a reproduzir. Neste momento o utilizador pode escolher parar o som da *track* que acabou de gravar. Para isso basta carregar no botão de pausa, o som é parado imediatamente e a cor da metade superior do rectângulo muda para castanho para indicar a pausa.

3.3 Visualização

Um dos objectivos da aplicação era conseguir a visualização clara da música criada. Visto que cada *track* toca um som único e distinto, foi também um objetivo desde o início que se conseguisse distinguir os sons de cada *track* na visualização.

Lembrando que a aplicação está dividida em dois tipos de *tracks*, as *tacks* base, que não podem ser alteradas pelo utilizador, e as *tracks* normais que podem ser manipuladas pelo utilizador, foram criados dois tipos de visualização.

Relativamente às *tracks* base, criou-se uma visualização relativa a apenas uma delas, a *track* que contém o som do bombo. Para este efeito o plano vertical ao fundo do espaço tridimensional contém dois círculos posicionados nos cantos inferiores direito e esquerdo, tal como pode ser observado na figura 5, e que reagem em simultâneo, aumentando de tamanho quando é reproduzido o som. Devido à cor que possuem, os círculos assemelham-se a uma luz pulsante. Esta visualização é iniciada assim que o som começa a ser reproduzido.

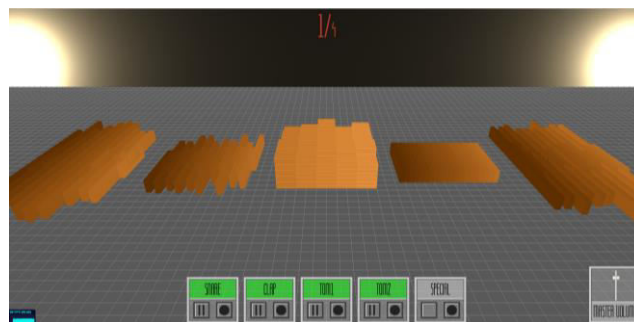


Figura 5 – Vários conjuntos de paralelepípedos a variar o seu tamanho consoante o som tocado

Relativamente às *tracks* normais, que podem ser manipuladas pelo utilizador, existem presentemente cinco, podendo no entanto ser adicionadas mais no futuro. A cada uma corresponde um dos blocos de paralelepípedos que estão dispostos horizontalmente, tal como se pode ver na figura 5.

Os blocos são animados quando é reproduzido o som da respectiva *track* e o seu movimento é função da média das frequências que compõe o som. Para facilitar a identificação da *track* que está a ser reproduzida definiram-se três tipos de animações uma para os blocos dos extremos, outra para o segundo e quarto blocos e uma terceira para o bloco do meio. Com esta distribuição de animações mantém-se a simetria da imagem.

Os blocos dos extremos são compostos por 20 paralelepípedos, com a dimensão maior paralela ao eixo dos *zz*, que efetuam uma mudança de escala segundo esse eixo. O bloco do meio é constituído por um conjunto de 5x5 paralelepípedos e efetua uma mudança de escala segundo o eixo dos *yy*. Finalmente os restantes dois conjuntos de blocos, igualmente constituídos por 20 paralelepípedos, dispostos também com a dimensão maior segundo o eixo dos *zz*, efetuam uma translação segundo este eixo de forma desencontrada, ou seja, se o primeiro paralelepípedo efectua uma translação positiva no eixo dos *zz*, o segundo efectua uma translação negativa e assim sucessivamente. Em todas as visualizações existe sempre alguma aleatoriedade no tamanho dos paralelepípedos quando reagem ao som reproduzido.

A complementar a visualização sobre as *tracks*, apresentam-se no topo da imagem e centrados dois números no formato X/X. O número da esquerda conta de um até quatro e está sincronizado com o som da batida da música, marca portanto o tempo. O segundo número conta de um até dezasseis e representa as notas que podem ser tocadas quando a mão do utilizador está na zona quatro vezes mais rápida - tabela 1.

Mostra-se também, na zona inferior esquerda da imagem, o número de *frames* por segundo (*fps*) em que a aplicação está a correr. Idealmente este valor encontra-se a 60fps. É usada para fazer testes de desempenho.

3.4 Organização dos módulos da aplicação

A figura 6 mostra a organização dos vários módulos que compõem a aplicação. O módulo *main* controla a componente de visualização e a sua ligação com os outros módulos: a componente de interação que se subdivide pela interface gráfica (módulo *Ui*) e pela interação com o dispositivo Leap Motion (módulo *Leap*); e a componente de áudio (módulos associados ao módulo *Audio*). Na componente de áudio temos os módulos relativos à simulação de um metrónomo (módulo *Metronome*), à reprodução de sons (módulo *Sounds*) e ao processo de gravação de *tracks* para composição de novos sons (módulo *Recorder*). Como já foi referido, uma das *tracks* base emite um som contínuo que serve de fundo à aplicação. Como a geração deste som é independente dos outros sons produzidos durante a execução da aplicação, há um módulo especial para este efeito (módulo *Drone*).

4. CONCLUSÕES

Desenvolveu-se uma aplicação interativa para a criação de música a partir de sons base. Esta aplicação integra a combinação de três componentes: áudio, para reprodução, gravação e combinação de sons; interação, com recurso à Leap Motion; e visualização. A aplicação baseada nas tecnologias *HTML5*, *CSS3* e *javascript*, tem um nível de desempenho que permite a gravação e reprodução de som de forma precisa e em tempo-real.

A aplicação foi construída de forma modular, de modo não só a tornar o código mais legível e a facilitar a manutenção como também a ser mais fácil estender o trabalho que foi desenvolvido até agora, implementando outras funcionalidades.

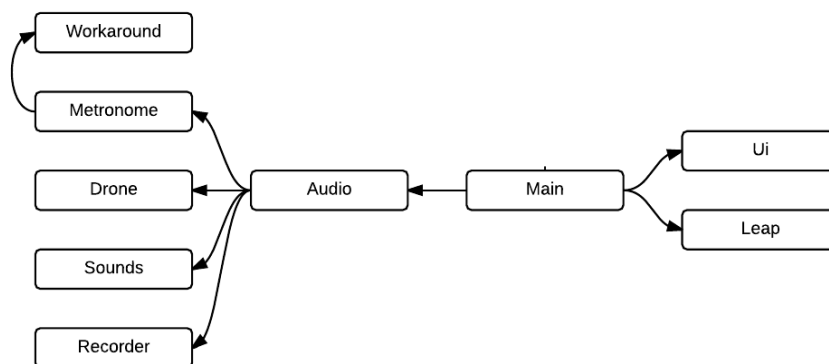


Figura 6 – Organização dos vários módulos da aplicação

O efeito final obtido com a aplicação e particularmente com a junção dos 3 módulos discutidos anteriormente (áudio, *Leap Motion* e Visualização) é algo que irá ser futuramente testado com utilizadores comuns e também com especialistas nestas áreas para obter *feedback* acerca dos vários aspetos que constituem a aplicação. Especificamente, será interessante ter *feedback* da interação do utilizador com a Leap Motion; saber se em termos de visualização esta é perceptível e se é claro qual a representação visual que corresponde a cada *track* e, conseqüentemente, cada som. Será útil ainda saber se a divisão do espaço tridimensional em zonas que têm várias ações associadas foi uma boa escolha em termos de interação com a Leap Motion.

Em termos de composição de sons e da criação de música, pretende-se apurar se o processo usado para a reprodução e gravação de som é fácil de utilizar. Visto que a aplicação tem como utilizador alvo alguém que não tem nenhum tipo de treino musical, gostaríamos de saber qual a sua opinião final relativamente ao resultado da música que criou.

5. TRABALHO FUTURO

A aplicação desenvolvida recorre unicamente a sons de ritmo. No futuro será interessante tentar usar sons com notas, discretos (como um xilofone) e contínuos (como um violino). A utilização deste tipo de sons levantará novos problemas, não só em termos de áudio e do seu processamento (mais desafiante nos sons contínuos do que nos discretos) como também de interação. Como interagir com sons que têm várias notas?

Relativamente aos sons contínuos, afigura-se aconselhável o recurso a sons sintetizados em vez de sons reais, tal como é feito com a maior parte dos sons usados nesta aplicação. Deste modo será mais fácil a mudança de nota sem a perceção pelo utilizador de qualquer problema, como cortes abruptos.

Relativamente à interação com a Leap Motion, a nova versão da *driver* do Leap Motion, lançada recentemente, inclui funcionalidades que permitem simplificar algumas das opções escolhidas para a interação. É possível agora quantificar o fechamento da mão do utilizador. Com esta funcionalidade poderíamos eliminar a zona mais alta associada à mão esquerda e usar o fechamento da mão ou a sua abertura para saber se o utilizador pretende tocar ou não determinado som.

6. AGRADECIMENTOS

Agradecemos à Fundação para a Ciência e Tecnologia (FCT) e ao laboratório de I&D LabMAg o apoio financeiro dado para a realização deste trabalho, ao abrigo do projeto estratégico PEst-OE/EEI/UI0434/2014. Agradecemos também a todos os participantes nos testes.

7. REFERÊNCIAS

- [Airspace] Leap Motion Store – Airspace (acedido em 28-07-2014). <https://airspace.leapmotion.com/>
- [BeatPetite] BeatPetite (acedido em 28-07-2014) <http://beatpetite.com/>.
- [Bernhoft] Bernhof (acedido em 28-07-2014) <http://bernhoft.org/>.
- [Biquad Filter] The Biquad Filter Node Interface (acedido em 28-07-2014) <http://webaudio.github.io/web-audio-api/#the-biquadfilternode-interface>
- [Color Particles] Music Colour Particles. jabtunes. (acedido em 28-07-2014) <http://jabtunes.com/labs/arabesque/>.
- [Cube Visualizer] Cube Visualizer (acedido em 28-07-2014) <http://shoffing.com/pages/projects/cubevis/>.
- [Dive] A dive in Music. (acedido em 28-07-2014) <http://do.ade.in/music/>.
- [Dub FX] Dub FX (acedido em 28-07-2014) <http://dubfx.net/>.
- [Leap] Leap Motion (acedido em 28-07-2014). <https://www.leapmotion.com/>
- [Ljósíð] Ólafur Arnalds - Ljósíð (Official Music Video) (acedido em 28-07-2014) <http://vimeo.com/6284199>.
- [Loop] Loop Waveform Visualizer. Air Tight Interactive. (acedido em 28-07-2014) <http://www.airtightinteractive.com/2012/01/loop-waveform-visualizer/>
- [Garcia11] Garcia, Christopher Michael. Study on the Procedural Generation of Visualization from Musical Input Using Generative Art Techniques. Master Thesis Texas A&M University, 2011.
- [Panner] Panner Node Interface (acedido em 28-07-2014) <http://webaudio.github.io/web-audio-api/#the-pannernode-interface>
- [Theremin28] Ssergejewitsch, Theremin Leo. *THEREMIN SIGNALING APPARATUS. US1658953* A USA, 14 de February de 1928. Grant.
- [Tom Thum] Beatbox brilliance: Tom Thum at TEDxSydney (acedido em 28-07-2014) <http://www.youtube.com/watch?v=GNZBSZD16cY&feature=youtu.be&t=8m12s>
- [Tone Craft] Tone Craft (acedido em 28-07-2014) <http://labs.dinahmoe.com/#tonecraft>.
- [ToneMatrix] ToneMatrix. AudioTool (acedido em 28-07-2014) <http://tonematrix.audiotool.com/>.
- [Web Audio API] Web Audio API (acedido em 28-07-2014) <http://webaudio.github.io/web-audio-api>
- [Zoe Keating] Zoe Keating - 'Lost' [HD] Sound Quality, ABC Radio National. Youtube. (acedido em 28-07-2014) <http://www.youtube.com/watch?v=sG9H5E2JN3s>

Artefactos e Exposições 3D Virtuais no Contexto duma Rede de Cultura e Arte Digital

Luís Romero
romero@estg.ipv.pt

Pedro Miguel Moreira
pmoreira@estg.ipv.pt

Carla Dias
carladias96@gmail.com

Luís Carrilho
lfcarrilho@gmail.com

Instituto Politécnico de Viana do Castelo
Escola Superior de Tecnologia e Gestão
Av. do Atlântico, Viana do Castelo

Resumo

Este documento apresenta o projecto duma REde de Cultura e ARte DIgital (RECARDI), uma das suas aplicações, eExpor, e expõe a criação, e visualização de artefactos 3D e a capacidade de exposição de artefactos digitais num ambiente virtual 3D. A rede no projecto RECARDI é suportada numa plataforma digital, e permite a Experimentação e Criação (individual ou colaborativa), a Exibição Virtual, o Ensino/Treino on-line e a Preservação Digital e os Direitos de Autor em Repositório Digital próprio, dos mais diversos artefactos de arte e cultura digital. A aplicação eExpor visa proporcionar um local para qualquer artista (talento emergente, ou profissional) expor e discutir arte. A aplicação eExpor, oferece capacidades para visualizar artefactos 3D e criar um espaço virtual tridimensional sob a forma de uma galeria onde o utilizador-público poderá aceder a exposições virtuais existente onde os artefactos digitais podem ser apreciados.

Palavras Chave

Galeria Virtual, Web3D, Modelos 3D, Visualização 3D, Realidade Virtual.

1. INTRODUÇÃO

O projecto RECARDI (REde de Cultura e ARte Digital) [Recardi12] tem como objectivo a criação, dinamização e expansão de uma "rede de excelência" em Cultura e Arte Digital activa, para a Web. A rede RECARDI é constituída, numa visão macro, por uma plataforma central, um repositório global de artefactos digitais e um conjunto de outros repositórios multimédia associados ao projecto assim como diversas aplicações piloto. No âmbito do projecto foram desenvolvidas as primeiras cinco aplicações piloto, nomeadamente: "eExperimentar", "eExpor", "eColaborar", "eAprender", "eNarrar" e "ePreservar". Estas constituem a "semente" da rede, isto é, são as primeiras de outras que se pretendem venham a ser desenvolvidas pela comunidade que fará uso da rede RECARDI, sejam elas derivadas destas, sejam de outras novas soluções totalmente desenvolvidas de raiz, explorando a modularidade e extensibilidade da plataforma. Em concreto, a aplicação eExpor visa proporcionar um local para qualquer artista expor a sua arte. Como tal pretende-se disponibilizar uma plataforma facilitadora da partilha de arte, capaz de produzir formas de interacção adaptadas aos desejos dos visitantes e dos artistas que promovem as suas obras. A experiência de contemplar arte deve ser centrada não só nos artefactos mas também no processo criativo que lhes deu origem. Adicionalmente oferece capacidades para criar um espaço virtual tridimensional, sob a forma de uma galeria, onde o criador poderá dispor artefactos digitais, sob um prisma expressivo do próprio, à qual o utilizador-público poderá aceder

para apreciar uma linha narrativa constituída por artefactos definida pelo artista. Este documento centra-se na descrição da solução obtida para a visualização de artefactos 3D e na criação de exposições 3D de artefactos digitais.

2. TRABALHO REALACIONADO

O interesse sobre a construção de galerias virtuais 3D vem de há bastantes anos. Um dos primeiros trabalhos pode ser encontrado em [Ullrich02]. Descreve uma aplicação com capacidades para criar um ambiente virtual e distribuir objectos 3D pelo espaço. Utiliza a tecnologia VRML97, um verdadeiro precursor de visualização 3D na Web. As galerias têm sido adoptadas pela comunidade de museus do mundo inteiro. A exposição de arte na Web permite não só apresentar as peças dos museus mas também possibilita a visita virtual, cativando desta forma potenciais visitantes do museu. Tomam diversas formas de interagir com o utilizador, utilizando tecnologias como Web3D [Sinclair03], Realidade Virtual [Kwon03], Realidade aumentada [Mase96], Realidade Mista [Hall01] entre outras. Um levantamento sobre trabalhos de museus virtuais pode ser encontrado em [Styliani09]. Actualmente existem algumas soluções de galerias de arte disponíveis para o público que fazem uso de tecnologias 3D para expor obras de arte. O Virtual Art Space (VAS) [VAS14] foi fundado por um grupo de artistas que identificou a necessidade de galerias 3D, para fins de exibir arte na internet. O VAS é uma rede de galerias virtuais 3D gratuita e está disponível para dispositivos móveis. Utiliza a tecnologia comercial Unity 3D que

implica a instalação de um *plugin* para poder usufruir desta tecnologia. O Artfolio 3D Gallery [ArtFolio14] é uma plataforma onde os artistas podem colocar as suas obras em forma de galerias 3D e utilizam a tecnologia WebGL [WebGL14]. Só é permitido colocar imagens como artefactos. A Kunstmatrix [Kunstmatrix14] é uma plataforma que permite criar galerias virtuais, com diferentes ambientes e onde é possível inserir imagens e esculturas num ambiente tridimensional. Utiliza a tecnologia Flash da Adobe.

3. ARQUITECTURA DO SISTEMA

A arquitectura lógica da plataforma RECARDI está representada de forma resumida na Figura 1. Descrita detalhadamente em documento próprio [Recardi12].

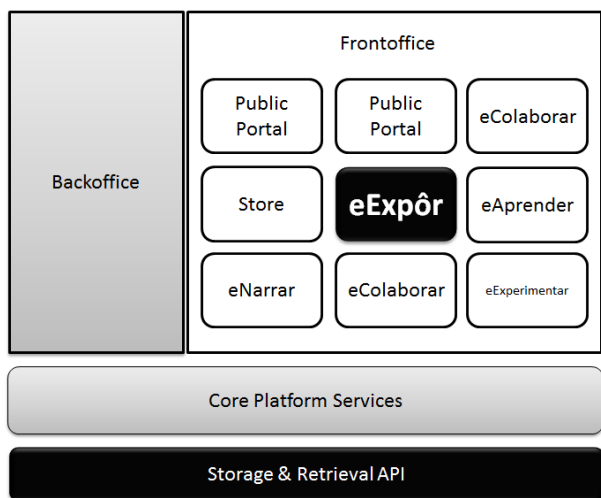


Figura 1: Arquitectura lógica do RECARDI

A aplicação eExpôr, enquanto elemento integrante da plataforma RECARDI, é responsável pela partilha de obras de arte. Depende de diversos serviços de plataforma, tais como o serviço de pesquisa e o serviço de gestão de utilizadores para permitir a localização de obras e artistas.

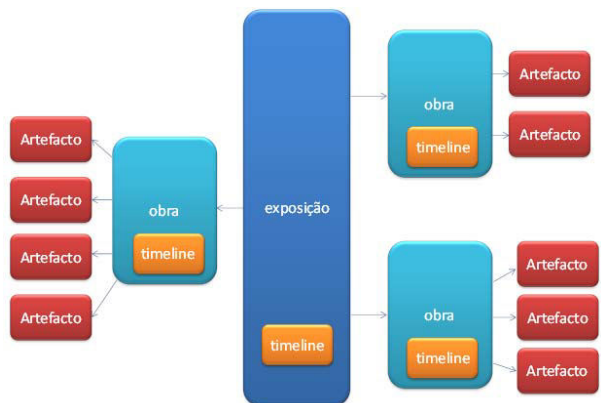


Figura 2: Elementos principais da aplicação eExpôr

Na Figura 2 estão representados as entidades mais importantes da aplicação eExpôr que são as seguintes: a obra, a exposição, a *timeline* e os artefactos.

Uma obra é representada através de artefactos tais como imagens, vídeo e elementos 3D, e tem associada uma *timeline* (cronograma da criação) que reflecte os avanços e recuos que a obra vai sofrendo durante o seu processo de construção e restante ciclo de vida. Uma exposição é composta por um conjunto de artefactos de obras cuja visita é alcançada através da navegação no ambiente virtual da exposição. Tal como as obras as exposições também disponibilizam uma *timeline* onde é registada a evolução da exposição ao longo do tempo.

4. ARTEFACTOS 3D

O manuseamento dos artefactos 3D assenta na tecnologia WebGL. A biblioteca de desenvolvimento utilizada é Three.js [Dirksen13] [Threejs14]. Permite criar e animar ambientes 3D e assenta no *standard* da WebGL que, por sua vez, faz uso do *standard* HTML5 e a sua definição do elemento *canvas*. A biblioteca Three.js foi criada para permitir uma programação com um maior nível de abstracção relativamente ao WebGL. No caso da aplicação eExpôr são relevantes as seguintes operações:

- Definição de câmaras e luzes;
- Leitura de objectos 3D em diversos formatos;
- Aplicação de texturas e materiais;
- Interação e Animação de objectos.

As câmaras e luzes são operações simples implementadas directamente com o javascript. A leitura de modelos e aplicação de texturas são dependentes da leitura dos modelos 3D. O ambiente virtual do espaço utiliza as características de geometria e texturas da biblioteca para criar o espaço virtual. A criação da exposição utiliza as características interactivas do WebGL/Three.js.

4.1 Formatos de Modelos 3D

Existem inúmeros formatos para representarem modelos 3D que são suportados pela biblioteca Three.js.

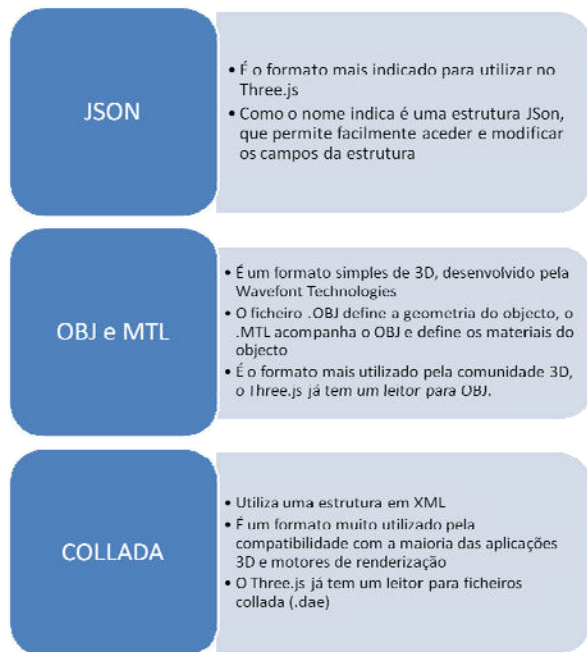


Figura 3: Formatos 3D suportados

Cada formato contém diversos componentes que representam a geometria, as texturas aplicadas, e outros dados relevantes para a definição dum modelo 3D. Na aplicação eExpor os modelos 3D dos artefactos, e respectivas texturas, podem ser carregados em três formatos possíveis. A Figura 3 ilustra os formatos suportados pela aplicação eExpor e as características de cada um deles.

4.2 Visualização 3D

Para visualizar um artefacto numa cena no Three.js é criado o objecto renderizador WebGL do tipo HTML canvas. Posteriormente é definido um ponto de vista através duma câmara, é utilizada uma visão em perspectiva com os respectivos parâmetros, e adicionadas luzes, passíveis de serem parametrizadas.

O conjunto de elementos presentes no visualizador de artefactos constitui a hierarquia de objectos ilustrada na Figura 4.

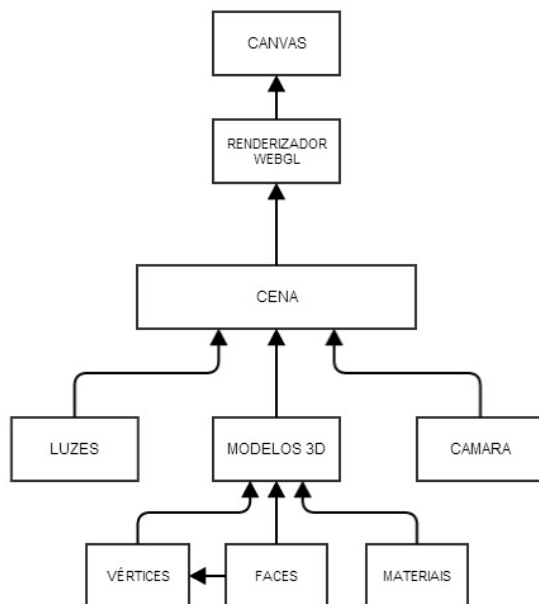


Figura 4: Hierarquia de objectos no visualizador 3D

Cada artefacto tem associado um conjunto de materiais, que representam as características da superfície do artefacto, e definem o aspecto visual do objecto. Os materiais podem ser definidos através de texturas, com possíveis efeitos, ou vídeo. São suportadas imagens do tipo PNG, GIF e JPEG como texturas e os formatos mp4, ogg/ogv e webm de vídeo.

4.3 Integração na plataforma

Tal como descrito (Figura 2) a aplicação eExpor define uma obra como um conjunto de artefactos. A integração dos objectos 3D nesta definição é ilustrada pela Figura 5.

Um artefacto 3D utiliza a estrutura modelos-3d para definir a geometria do objecto e o conjunto de materiais utilizados no mesmo. A estrutura MATERIAIS-3D representa a definição dos materiais utilizados pelo objecto. Os materiais de um objecto são utilizados para criar menus que permitem a alteração de materiais no artefacto.

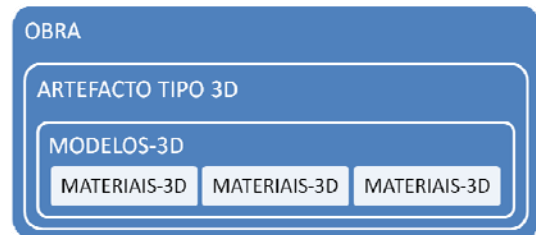


Figura 5: Estruturas dos Artefactos 3D

O processamento do visualizador 3D passa por três fases importantes:

- Pesquisa na API (*storage & retrieval*) sobre a existência de dados referentes ao modelo que se pretende visualizar.
- Escolha do *loader* mais adequado para a leitura do modelo 3D.
- Cálculo da dimensão do objecto e configuração da visualização, i.e., como os objectos podem ter dimensões diferentes, implica ajustar as configurações da câmara de modo a manter o modelo 3D centrado na cena. Construção do interface que permite alterar as cores do artefacto e as cores das luzes.

A Figura 6 ilustra o resultado final do visualizador de artefactos 3D integrado na aplicação eExpor.

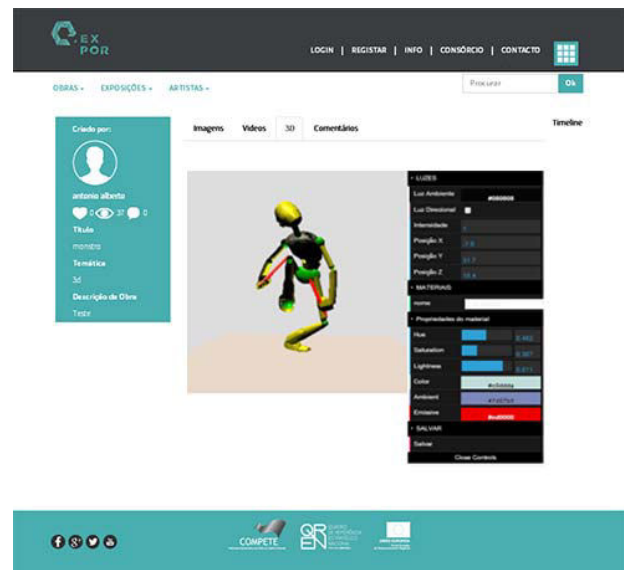


Figura 6: Visualizador 3D na aplicação eExpor

5. EXPOSIÇÃO 3D

Para a construção do espaço virtual da exposição, foi levado em consideração a sua forma, iluminação e organização. As salas serão aparelhadas como um *puzzle* numa grelha onde cada quadratura representa um quadrado com 6x6 metros e 3,5 metros de altura. De forma a simplificar a construção dos espaços é adoptado um conjunto de formatos de sala que o utilizador pode utilizar para criar as suas exposições, como ilustra a Figura 7.

Na plataforma uma sala é considerada como uma obra constituída pelos artefactos: chão, paredes, e tecto. Notar

que desta forma fica aberta a possibilidade de criação de espaços pelos utilizadores.

Ao criar uma exposição 3D no eExpor, o utilizador escolhe o formato do espaço da sala para criar a exposição. Após o autor escolher a sala pretendida, um segundo menu aparecerá com as obras disponíveis, e os respectivos artefactos, presentes na exposição geral, tal como mostra a Figura 7.

O autor coloca as obras e os artefactos na sala 3D com a ajuda da interface fornecida (menu na lateral direita). Será capaz de rodar, escalar e transladar os objectos no espaço 3D. A selecção de objectos poderá ser efectuada de duas formas: Através do rato ou da interface, seleccionando o identificador do objecto na cena.

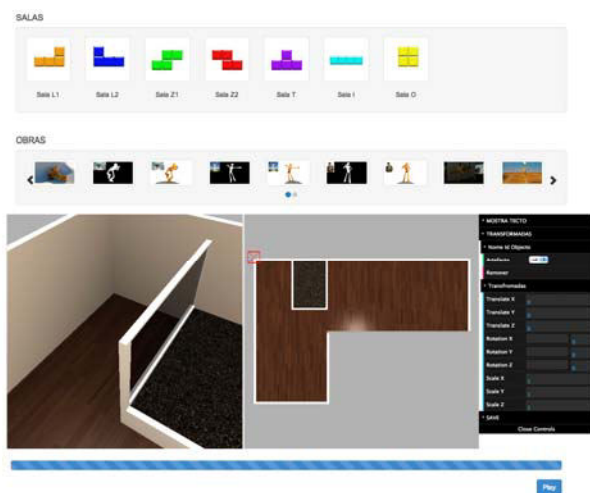


Figura 7: Formatos de espaços de salas

Para visualizar a exposição no espaço virtual basta escolher o botão *Play* (Figura 7). O utilizador poderá navegar no espaço, em modo de ecrã inteiro (*fullscreen*), e utilizando as teclas de direcção para se movimentar e o rato para se orientar.

6. CONCLUSÕES

A aplicação desenvolvida para criação e visualização de artefactos 3D e exposições virtuais 3D no âmbito do projecto RECARDI constituem um exemplo para a implementação de soluções de representação de objectos e criações 3D numa plataforma web. Apesar da computação gráfica tridimensional estar cada vez mais presente nas variadíssimas páginas Web disponíveis, as tecnologias existentes ainda não têm uma presença generalizada no mundo da internet. Existem na internet várias galerias de arte que expõe e comercializam objectos de arte. Poucas são as que permitem os utilizadores criarem exposições com artefactos digitais 3D. As tecnologias necessárias para a criação de soluções estão disponíveis para qualquer interessado, tal como este projecto demonstra ao utilizar ferramentas não comerciais e de código aberto. A solução exposta está sujeita a diversas melhorias tais como a junção de espaços diferentes, a introdução do áudio, a diversificação de tipos de media passíveis de serem integrados no espaço da exposição, optimização da visualização, proporcionar a criação de guias das obras

expostas, apenas mencionando algumas das possíveis melhorias.

7. AGRADECIMENTOS

Projecto financiado pelo Programa Compete, QREN, FEDER, projeto em co-promoção nº 22997.

8. REFERENCIAS

- [ArtFolio14] Artfolio 3D Gallery (ArteFolio). Acedido a Julho 2014 e disponível em: <<http://www.artfolio.de>>
- [Dirksen13] Jos Dirksen. Learning Three.js: The javascript 3D Library for WebGL. *PACKT* (2013).
- [Hall01] T. Hall, L. Ciolfi, M. Fraser, S. Benford, J. Bowers, C. Greenhalgh, S. Hellstrom, S. Izadi, H. Schnadelbach, The visitor as virtual archaeologist: using mixed reality technology to enhance education and social interaction in the museum, in: S. Spencer (Ed.), Proceedings of the VAST 2001 Conference, Greece, November 2001, ACM Press, New York, 2001.
- [Kunstmatrix14] Kunstmatrix. Acedido a Julho 2014 e disponível em: <<http://www.kunstmatrix.com>>
- [Kwon03] Y.M. Kwon, J.E. Hwang, T.S. Lee, M.J. Lee, J.K. Suhl, S.W. Ryul, Toward the synchronized experiences between real and virtual museum, in: Proceedings of Conference of APAN, January 2003, Japan, 2003.
- [Mase96] K. Mase, R. Kadobayashi, R. Nakatsu, Meta-Museum: A Supportive Augmented-Reality Environment for knowledge Sharing, in: Proceedings of the Conference VSMM '96, Japan, September 1996, IEEE Computer Society Press, 1996, pp. 107–110.
- [Recardi12] Consórcio RECARDI, Rede de Cultura e Arte Digital – Relatório Técnico. Julho 2012.
- [Recardi14] Rede de Cultura e Arte Digital, Agosto 2014. <<http://www.recardi.com/>>
- [Styliani09] Sylaiou Styliani, Liarokapis Fotis, Kotsakis Kostas, Patias Petros. Virtual museums, a survey and some issues for consideration. *Journal of Cultural Heritage* 10 (2009) 520–528.
- [Sinclair03] P.A.S. Sinclair, K. Martinez, D.E. Millard, M.J. Weal, Augmented reality as an interface to adaptive hypermedia systems. New review of hypermedia and multimedia, Special Issue on Hypermedia beyond the Desktop 9 (1) (2003) 117–136.
- [Threejs14] three.js - Javascript 3D library. Acedido a Julho 2014 e disponível em: <<http://threejs.org/>>
- [Ullrich02] Tomas Ullrich, Jiri Zara. Interactive Virtual Gallery. *Spring Conference on Computer Graphics, Budmerice, Slovakia*, ACM 2002, 43–47.
- [VAS14] Virtual Art Space (VAS). Acedido a Julho 2014 e disponível em: <<http://www.3Dvas.com>>
- [WebGL14] "WebGL 2 Specification". khronos.org. 2013-09-26. Acedido a Julho 2014 e disponível em: <<http://www.khronos.org/registry/webgl/specs/latest/2.0/>>

Sessão - Visão por Computador e Realidade Aumentada

Sistema de Realidade Aumentada para Apoio ao Projeto de Arquitetura

Mariana Lopes^{1,2}
t-marig@microsoft.com

Jorge Silva^{3,4}
jsilva@fe.up.pt

Miguel Sales Dias^{2,5,6}
miguel.dias@microsoft.com

Sara Eloy^{5,6}
sara.elay@iscte.pt

Filipe Gaspar^{2,5,6}
t-filipg@microsoft.com

Ricardo Miguel⁵
jricardomiguel@gmail.com

Nuno Mendonça⁵
nunoasmendonca@gmail.com

¹ FEUP, Faculdade de Engenharia da Universidade do Porto, Porto. ² MLDC, Microsoft Language Development Center. ³ FEUP-DEI, Porto. ⁴ INESC TEC, Porto. ⁵ ISCTE-IUL, Lisboa. ⁶ ISTAR-IUL, Lisboa.

Resumo

Em arquitetura, a maquete, seja para estudo ou para a comunicação da proposta de projeto, desempenha um papel fundamental como ferramenta nos processos criativos de planeamento e conceção de projetos. No entanto, existem por vezes dificuldades associadas a esta abordagem, como o sejam, o tempo gasto e os custos envolvidos na sua produção e modificação, sempre que é necessário prever o resultado de mudanças no projeto, ou quando se tentam ver os pormenores do interior do modelo, o que pode não ser viável. Com o objetivo de auxiliar os arquitetos a superar essas dificuldades, uma hipótese foi proposta, que aponta para o uso da tecnologia de visualização interativa num ambiente de realidade aumentada, onde um modelo virtual 3D do projeto de arquitetura, na mesma escala da maquete de estudo, pode ser visualizado por meio de um dispositivo do tipo tablet. A verificação da hipótese levou à criação de um protótipo, permitindo a visualização de modelos 3D digitais de edifícios e do seu ambiente circundante, usando realidade aumentada. A nossa abordagem de realidade aumentada baseada em visão, permite o seguimento de texturas, necessitando de uma marca visual planar (uma imagem normal), a qual funciona como um plano de referência para o registo e posicionamento do modelo 3D. O sistema oferece várias funcionalidades que facilitam a modificação e a visualização interativa do modelo 3D, como a oclusão de objetos ou o apoio de planos de corte interativos, para visualização mais minuciosa do interior, amplamente utilizados em arquitetura. Para enriquecer a experiência de visualização e interação com o modelo 3D, o sistema utiliza uma plataforma de computação móvel com ecrã tátil, permitindo uma interação baseada em toque. A usabilidade deste protótipo foi positivamente avaliada com um painel de 16 estudantes universitários e 4 arquitetos, utilizando uma metodologia baseada em tarefas.

Palavras-chave

Realidade aumentada, arquitetura, dispositivos móveis, seguimento de textura, modelos virtuais 3D, FIRST, maquete de estudo, interface de toque, avaliação da usabilidade

1. INTRODUÇÃO

A maquete e o desenho representam um papel fundamental no projeto de arquitetura e no respetivo processo criativo. Estes métodos de representação de ideias permitem ao arquiteto expressar o seu pensamento, materializando um objeto que ainda só faz parte da sua imaginação de forma a demonstrar aquilo que pretende transmitir. As maquetes para arquitetura são, hoje em dia, diretamente baseadas nos modelos virtuais e produzidas por fabricação digital. Esta pode ser aditiva, construindo a forma com impressoras 3D, subtrativa, seccionando e traçando o contorno do material com máquinas CNC e cortadores laser ou formativa, dobrando e torcendo o material através de movimen-

tos controlados por computador que dobram a forma inicial na definida. A maquete física, dita de estudo, permite ter uma visão completa do objeto arquitetónico e é, até hoje, um método insubstituível no auxílio ao trabalho de projeto dos arquitetos. No entanto, as alterações nos modelos são morosas, com custos financeiros e de tempo elevados e não são passíveis de repetição, o que leva a que os modelos 3D virtuais surjam como solução para a modificação e visualização de características que outra forma seriam impossíveis de realizar, como a visualização de pormenores dos interiores em maquetes a escalas pequenas. Os modelos 3D digitais são cada vez mais um complemento às maquetes físicas (de estudo) utilizadas no processo de projeto, pois para além de permitirem obter imagens de diversos pontos de vista, fiéis à realidade do

objeto, permitem ainda tratar de questões analíticas, tanto aos níveis da estrutura como da infraestrutura, facilitando assim a sua simulação e compreensão [Freitas10].

O arquiteto tem que ter a habilidade de conseguir expor com clareza as suas ideias, tanto com o cliente final como com o promotor imobiliário ou com os restantes técnicos de especialidades que com ele trabalham. Como tal, as ferramentas de representação utilizadas são vitais para a perceção do projeto, seja elas físicas ou virtuais (Figura 1).



Figura 1: Projeto de arquitetura representado de forma real (imagem da esquerda, em maquete) e virtual (imagem da direita), realizado por Ana Moural, ISCTE-IUL.

As tecnologias de Realidade Virtual (RV) e de Realidade Aumentada (RA) têm sido usadas maioritariamente para auxílio na visualização e interação com informação do projeto de arquitetura [Veliz12]. Em particular, com a RA, a qual permite ao utilizador ver e interagir em tempo real com o mundo real onde pode observar objetos virtuais sobrepostos e alinhados (registados) com objetos reais, o projeto de arquitetura pode ser representado fisicamente, por meio de uma maquete, onde se pode registar um modelo virtual realizado com ferramentas apropriadas de CAAD [Wang13]. A utilização da RA poderá potenciar maior dinâmica e simplificação na comunicação entre todas as partes envolvidas num projeto de construção de um edifício, por exemplo, desde o momento criativo, com as suas fases de análise e simulação, passando pelo desenvolvimento e representação do edifício, pela fabricação dos componentes, incluindo intervenção na obra, durante a execução do edifício [Amim07].

Com o objetivo de auxiliar os arquitetos a ultrapassar as dificuldades atrás enunciadas com a utilização da maquete de estudo no processo atual de conceção em arquitetura, foi proposta uma solução tecnológica alternativa à referida maquete. De facto, pretende-se demonstrar que a visualização e interação com modelos virtuais de projetos de arquitetura à escala das maquetes de estudo, utilizando tecnologia de realidade aumentada consumida num dispositivo do tipo *tablet*, são benéficas para o processo de conceção em arquitetura, quer do ponto de vista de análise iterativa de conceitos e ideias, quer da compreensão e comunicação do projeto de arquitetura. Para tal, definiram-se os seguintes objetivos:

- Levantar os requisitos, desenvolver e testar um protótipo demonstrável de um sistema que permita a visualização e interação com modelos virtuais 3D de edifícios, usando tecnologia de realidade aumentada. As funcionalidades suportadas incluem:

- Plataforma computacional móvel com ecrã tátil, permitindo uma interação baseada em toque com o ambiente aumentado.
- Sistema de realidade aumentada utilizando técnicas de visão baseadas em seguimento de textura, as quais necessitam de uma marca visual planar (com uma textura cromaticamente variada), funcionando como plano de referência para registo e posicionamento do modelo.
- Suporte à visualização, modificação e interação com o modelo 3D, como por exemplo a oclusão de objetos e o suporte de planos de corte interativos para inspeção de pormenores interiores, muito utilizados em arquitetura.
- Desenvolver um estudo de avaliação da usabilidade e satisfação do protótipo desenvolvido e analisar os respetivos resultados.

Seguiu-se uma metodologia de investigação que compreendeu uma fase de levantamentos de requisitos junto de alguns colaboradores, professores e alunos, da área de Arquitetura, seguida da priorização dos mesmos. Depois seguiu-se uma fase de planeamento pormenorizado do desenvolvimento de funcionalidades, com aprovação do cumprimento dos requisitos essenciais pelos colaboradores iniciais. Após nova priorização das funcionalidades, as mais relevantes foram desenvolvidas e testadas num protótipo, culminando com a realização e análise de um estudo de avaliação da usabilidade e satisfação do mesmo protótipo. Neste último, adotámos uma metodologia baseada em tarefas, com medição do tempo de cada utilizador, registo das quebras e erros cometidos, envolvendo ainda observação e entrevista dos participantes.

Este artigo está organizado da seguinte forma. Na secção 2, resume-se os trabalhos relacionado com sistemas de realidade aumentada para arquitetura. De seguida, na secção 3, é apresentada uma visão geral do sistema desenvolvido ARch (*Augmented Reality for Architecture Design*), como a sua arquitetura, as suas funcionalidades e o respetivo desenvolvimento. Na secção 4 é descrita a metodologia e apresentados e discutidos os resultados recolhidos na fase de avaliação da usabilidade do ARch. Finalmente, na secção 5, são retiradas conclusões e proposto um plano para trabalho futuro.

2. TRABALHO RELACIONADO

A realidade aumentada permite ao utilizador visualizar o mundo real com objetos virtuais sobrepostos e alinhados (registados) com este, dando uma sensação de fusão entre os dois mundos [Azuma97]. Esta tecnologia tem sido adaptada em áreas tão diversificadas como a militar, a do entretenimento, a medicina ou a arquitetura, entre outras. Na área da arquitetura, os sistemas de RA têm sido usados tanto em ambientes exteriores [Sareika10] para, por exemplo, antever o impacto visual de edifícios [Graf10] ou de pontes a construir [Broschart13], como em ambientes interiores, essencialmente na fase de conceção do projeto de arquitetura. A revisão de trabalhos relacionados que se segue cingiu-se a sistemas deste último tipo.

Desenvolvido por Broll et al. [Broll04], Arthur é um sistema de realidade aumentada usado para planeamento urbano e tomadas de decisões dos arquitetos, reunidos em torno de uma mesa. O sistema é baseado na utilização de HMDs (*Head Mounted Displays*) e marcas visuais (Figura 2), sendo a interação com o utilizador feita por meio de marcadores, apontadores e gestos. É possível executar ações como navegar em menus, desenhar linhas no espaço ou selecionar e manipular objetos virtuais.



Figura 2: Exemplo de utilização do sistema ARTHUR (figura adaptada de [Broll04])

Também o *Augmented Reality-based Urban Design System* (ARUDesigner), desenvolvido por Wang [Wang08] utilizava o mesmo sistema de marcas (ARToolkit) e HMD. Este sistema (Figura 3) foi criado para reduzir a possibilidade de mal interpretações entre os membros da equipa multidisciplinar, fornecendo uma informação visual mais detalhada e encorajando a colaboração e comunicação entre os referidos mesmos, no planeamento urbano. Apresentado por Belcher e Johnson [Belcher08a], o ARchitecture-View permite a visualização de modelos arquitetónicos de edifícios, especificamente modelos BIM (*Building Information Model*) (Figura 4).



Figura 3: Utilização do sistema ARUDesigner (figura adaptada de [Wang08])

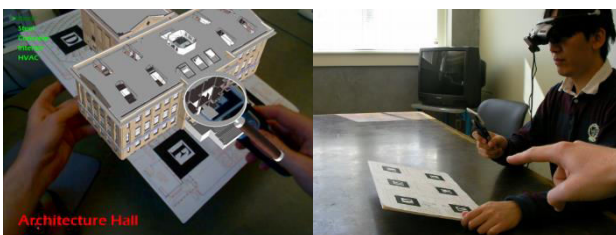


Figura 4: Exemplo de utilização do sistema ARchitectureView (figura adaptada de [Belcher08a])

MxR é um sistema de realidade aumentada baseado em marcas planares visuais, para suportar a colaboração durante as fases iniciais do desenvolvimento na área de arquitetura, apresentado por Belcher e Johnson [Belcher08b]. Este sistema permite visualizar o modelo 3D virtual sobre o físico, ou seja, uma maquete, como é ilustrado na Figura 5, permitindo que os vários intervenientes no projeto possam discutir e testar diversas hipóteses e visualizar os resultados, recorrendo a uma interface tangível.

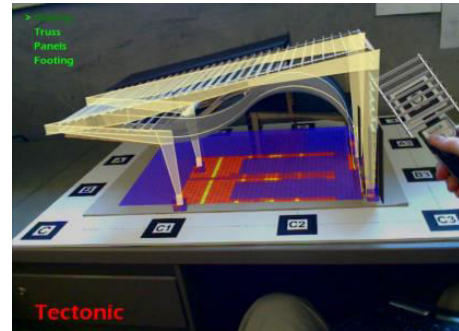


Figura 5: Vista em primeira pessoa do sistema MxR (figura adaptada de [Belcher08b])

AR-media Player [Am14a] permite ao utilizador visualizar modelos 3D criados com recurso a aplicações, como 3ds Max ou Vectorworks, para as quais esteja disponível o AR-media Plugin. O AR-media Player funciona como um sistema com recurso a marcas visuais, semelhantes às reconhecidas pelo ARToolkit [Kato99], permitindo o corte de objetos 3D, gestão de camadas, sombras em tempo real e também a ligação entre duas ou mais marcas visuais. A visualização da realidade aumentada tanto pode ser feita num ecrã de computador (Figura 6) como em dispositivos móveis.

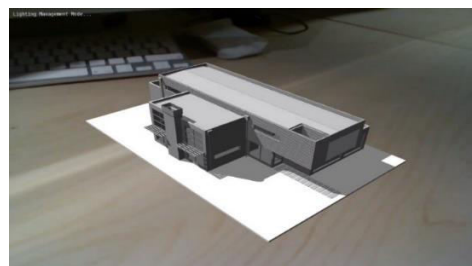


Figura 6: Modelo 3D visualizado através do AR-media Player (figura adaptada de [Am14b])

Os primeiros sistemas apresentados, apesar de serem utilizados no planeamento urbano para visualizar o modelo ou testar hipóteses, apresentam certas lacunas, como por exemplo, não permitirem cortes ou visualização por camadas do modelo. O AR-media, na versão para computador, suporta algumas dessas funcionalidades, mas não possui a mobilidade necessária para uma visualização mais interativa, que a versão móvel contém. Com exceção do AR-media, todos os sistemas apresentados utilizam HMDs, mas não são totalmente móveis. Os dispositivos móveis, como telemóveis, *smartphones* e *tablets*, surgiram como novos dispositivos alvo para a expansão da realidade aumentada, pois, em relação à alternativa anterior do uso de

HMD, são menos intrusivos, socialmente mais aceitáveis, sempre disponíveis e altamente móveis [Zhou08]. A aplicação AR-media para dispositivos móveis já fornece a mobilidade pretendida, mas não contém todas as funcionalidades de interação com o modelo que a aplicação de computador *desktop* possui. Neste contexto, a aplicação descrita neste trabalho, foi desenvolvida para *tablet*, apresentando um conjunto vasto de funcionalidades e permitindo uma maior mobilidade do utilizador e a interação com o sistema, através da manipulação da pose do dispositivo e do toque no seu ecrã, substituindo assim a utilização de HMD.

3. O SISTEMA ARCH

ARch, *Augmented Reality for Architecture Design*, é o sistema de RA que se descreve a seguir, desenvolvido pelos autores para apoio ao projeto de Arquitetura.

A utilização típica deste sistema desenvolve-se em torno de uma mesa sobre a qual está colocada uma marca visual constituída por uma superfície plana com um padrão de referência (por exemplo, uma fotografia), semelhante ao apresentado na Figura 7. Cada utilizador deverá dispor de um *tablet*, equipado com uma câmara, usado para captar imagens da marca, sendo este o único equipamento necessário. Nas secções seguintes serão apresentados a arquitetura (secção 3.1), o ambiente de desenvolvimento usado (secção 3.2), as funcionalidades (secção 3.3) e a interface de utilizador (secção 3.4) do sistema.



Figura 7: Caso de utilização do sistema

3.1 Arquitetura lógica da aplicação ARch

A aplicação desenvolvida, em execução no *tablet*, é constituída por quatro módulos principais (Figura 8). O módulo de “Vídeo” é responsável pela aquisição da imagem da câmara, passando a imagem adquirida ao módulo de “Visualização”.

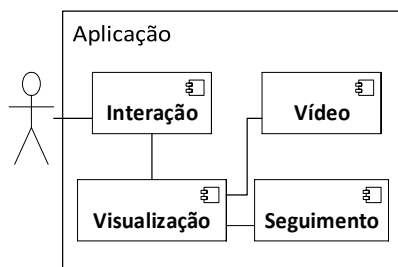


Figura 8: Arquitetura lógica da aplicação

Este módulo de “Visualização” é responsável pelo registo do modelo virtual com a imagem captada, em cada instante. Para garantir um alinhamento perfeito entre os dois mundos, real e virtual, este módulo usa a informação da pose da câmara, fornecida pelo módulo de “Seguimento”, o qual sintetiza uma câmara virtual com as mesmas características extrínsecas da câmara real usada na aquisição de imagem.

Os parâmetros intrínsecos da câmara são determinados previamente, com recurso a calibração, e fornecidos à aplicação. Para cada imagem capturada fornecida pelo módulo de “Visualização”, o módulo de “Seguimento” segue características previamente conhecidas na mesma (e previamente detetadas), e ainda elimina e descobre novas características, até que um número suficiente destas permita calcular a nova transformação de pose da câmara virtual, em relação à marca visual. O módulo de “Interação” recebe os comandos do utilizador e executa as ações solicitadas sobre o modelo virtual. Além disso, mostra informação de estado da aplicação, como, por exemplo, a funcionalidade ativa.

3.2 Ambiente de desenvolvimento

A aplicação foi desenvolvida para um *tablet*, correndo o sistema operativo Windows 8.1, com recurso às APIs (*Application Programming Interfaces*) NUTTS (*Natural Ubiquitous Texture Tracking System*) [Bastos08] e OSG (*OpenSceneGraph*) [Wang10]. A linguagem de programação utilizada foi C++, comum às APIs utilizadas.

Para captação do mundo real foi utilizada a câmara embutida do *tablet*. A Figura 9 mostra a arquitetura física de implantação da aplicação.

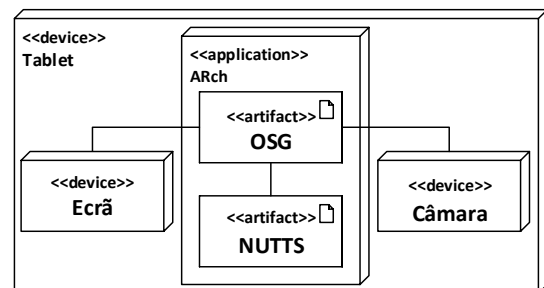


Figura 9: Arquitetura física da aplicação

O módulo de “Seguimento” adota a API NUTTS – *Natural Ubiquitous Texture Tracking System* para o reconhecimento e seguimento da marca visual. Esta API, que usa o descritor de características de imagem FIRST – *Fast Invariant to Rotation and Scale Transform* [Bastos09], uma técnica invariante à escala e rotação, tolerante a variações de luminosidade, ruído e *blurring* visuais e com repetibilidade semelhante ao SURF (*Speeded Up Robust Features*) [Bay08], mas mais eficiente e preciso, permite fazer a extração e correspondência de pontos característicos previamente conhecidos, da textura que constitui a marca visual. A técnica NUTTS utiliza técnicas de correção da distorção intrínseca da câmara, de *back-projection*, *template matching* e *optical flow*, para seguir em tempo real as características FIRST de uma ou várias imagens, permitindo assim fazer o seguimento de várias texturas em simultâneo.

A técnica de correção da distorção de perspetiva da textura numa determinada pose, permite recuperar pontos característicos perdidos na sequência de vídeo, evitando que o sistema fique sem informação desses pontos, permitindo a atualização da pose atual da câmara [Bastos09].

O módulo de “Visualização” usa a API OSG para visualização da cena 3D. Baseada no conceito de grafo de cena, esta API encapsula a maioria de funcionalidades incluídas na API do OpenGL [OpenGL14], fornecendo ao utilizador novas funcionalidades e um desenvolvimento mais rápido. O grafo de cena é uma estrutura que define as relações espaciais e lógicas de uma cena 3D para uma gestão e apresentação eficiente dos dados gráficos [Wang10]. É representado normalmente por um grafo acíclico organizado em árvore invertida, composto por um conjunto de nós e de folhas. Os nós podem representar uma transformação geométrica, um objeto virtual complexo, ou um agrupamento hierárquico de outros nós e folhas (conhecido como um sub-grafo), por exemplo. As folhas podem representar um objeto virtual primitivo ou até uma luz ou câmara virtual na cena. Esta API permite também carregar diversos formatos 3D com informação geométrica e topológica, como, por exemplo, FBX, que possibilita aos arquitetos definir as diferentes camadas da maquete fora do ambiente de desenvolvimento e facilita as funcionalidades de visualização ativando ou desativando grupo de nós no grafo.

O módulo de “Interação” suporta a interação baseada em toque e gestos planares (na superfície do ecrã). Após terem sido avaliadas possíveis APIs para fazer o desenvolvimento deste módulo, optou-se por utilizar também a OSG, devido à sua compatibilidade com a linguagem de programação utilizada (C++).

3.3 Funcionalidades

Para definir o conjunto das funcionalidades mais importantes para a aplicação, realizaram-se sessões de levantamento de requisitos junto de arquitetos, professores e alunos, da área de Arquitetura, utilizando técnicas de prototipagem rápida e entrevistas, seguidas da priorização dos mesmos. Dessas sessões resultaram as seguintes funcionalidades essenciais que o protótipo permite executar sobre o modelo virtual 3D:

- Visualizar o modelo através de um *tablet*, com interação através do toque.
- Realizar cortes verticais (alçados) e cortes horizontais (plantas).
- Realçar um grupo de objetos agrupados por “camadas” (correspondendo cada camada a uma especialidade: arquitetura, engenharia de estruturas, ou outra), sobre todos os outros objetos.
- Ocultar e selecionar um objeto ou grupo de objetos.
- Alterar a cor e a textura de um objeto previamente selecionado.

3.4 Interface Gráfica de Utilizador

A interface gráfica de utilizador permite a interação com toque e gestos planares e é constituída por duas zonas principais: a zona onde é visualizada a cena aumentada e a zona dos menus, com funções diferentes por cada modo de

visualização (Figura 10 e Figura 11). A aplicação arranca no “Modo de Apresentação” e, consoante a funcionalidade pretendida, comuta-se entre o “Modo de Cortes” e o “Modo de Seleção”, deslizando o dedo pelo ecrã.



Figura 10: Interação com a aplicação

No “Modo de Cortes” incluem-se todas as funcionalidades ligadas à realização de um corte horizontal ou vertical, enquanto no “Modo de Seleção”, se agrupam todas as funcionalidades ligadas à interação com objetos do modelo, como por exemplo, oclusão e seleção de objetos e modificação de cor e textura dos mesmos. Para a realização de um corte no edifício visualizado é necessário colocar a aplicação em “Modo de Cortes” e selecionar a direção do corte (horizontal ou vertical), arrastando seguidamente o dedo sobre o ecrã para deslocar o plano de corte (Figura 11). Os cortes podem ser realizados segundo um qualquer ângulo definido pelo utilizador e vistos de qualquer perspetiva.

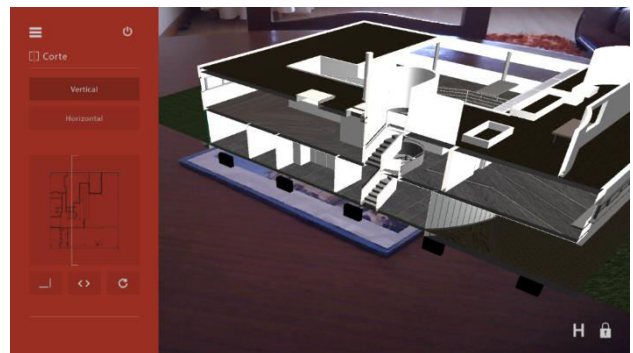


Figura 11: Realização de um corte no edifício

A funcionalidade de realçar um grupo de objetos, ilustrada na Figura 12, permite aos utilizadores visualizar infraestruturas ou estruturas que, de outra maneira, não seria possível, mantendo a perceção da localização das mesmas em relação ao edifício.

Cada grupo de objetos corresponde a uma “camada” do edifício, sendo as camadas designadas por “paredes”, “estrutura”, “infraestrutura” e “outras”. Para realçar qualquer uma das camadas, é necessário colocar a aplicação em “Modo de Seleção”, abrir o menu e selecionar a camada que se pretende realçar.

Outra das funcionalidades disponibilizadas permite editar objetos do modelo, como pintar, ocultar (Figura 13) ou modificar a textura de um objeto, de maneira a testar novas

ideias de visualização. Essas operações podem ser executadas selecionando as opções “Pintar” ou “Ocultar”, tocando de seguida sobre o objeto a editar.



Figura 12: Realçar uma camada

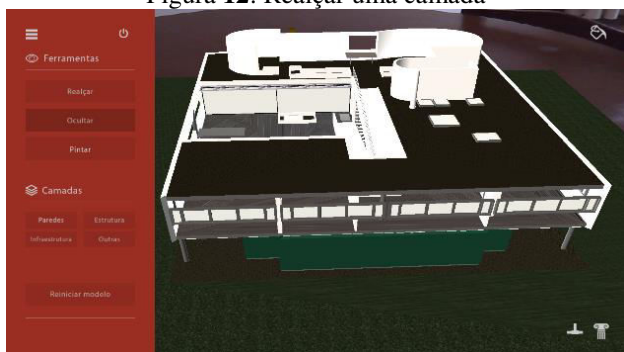


Figura 13: Ocultação de um objeto (parede frontal)

4. AVALIAÇÃO DA USABILIDADE

Realizou-se um estudo de avaliação da usabilidade e satisfação do sistema ARCh que envolveu a análise da execução de simples tarefas, como realizar um corte, realçar uma camada e ocultar um objeto. Para esta avaliação contamos com a participação de 16 estudantes do Mestrado Integrado em Arquitetura do ISCTE-IUL e 4 docentes do curso que participaram na qualidade de peritos.

4.1 Metodologia

Cada sessão foi realizada individualmente, num ambiente controlado e com recurso ao mesmo *hardware* e *software*. No início de cada sessão, os participantes foram informados do contexto e objetivo da avaliação e foi-lhes pedido que respondessem a um inquérito preliminar, usado para recolha de dados do participante. Seguiu-se a apresentação das funcionalidades do sistema e uma demonstração do seu funcionamento. Na Tabela 1 é apresentada a lista de tarefas e sub-tarefas que cada participante deveria executar, durante a sessão. Estas tarefas foram executadas na presença de um observador, envolvido no desenvolvimento do sistema. Durante a realização das tarefas, os participantes podiam falar com o observador, fazendo observações ou comentários sobre o protótipo ou as tarefas solicitadas. O observador não interferiu diretamente na realização das tarefas, limitando-se a dar algumas ajudas verbais, sempre que solicitadas, tomando notas sobre as ajudas solicitadas, o resultado de cada tarefa e outros factos considerados relevantes.

Tarefa	Sub-tarefa
Inicialização	1. Apontar câmara para a marca
	2. Visualizar/Explorar o modelo 3D
Realizar corte	3. Mudar para Modo de Cortes
	4. Alterar posição do corte
Ocultar	5. Mudar para Modo de Seleção
	6. Abrir o menu
	7. Selecionar a opção ocultar
	8. Selecionar um objeto
Realçar	9. Realçar uma camada

Tabela 1: Sub-tarefas realizadas pelos participantes por cada tarefa.

No final da sessão de avaliação, cada utilizador respondeu a dois pequenos inquéritos, um relativo à utilidade do protótipo e outro focando a usabilidade da sua interface gráfica.

4.2 Resultados e Discussão

Na Figura 14 são apresentados os resultados da avaliação do grau de cumprimento das tarefas, donde se pode concluir que todos os participantes concluíram com sucesso todas as sub-tarefas (de 1 a 9) propostas e que, maioritariamente, as mesmas foram executadas com facilidade. As maiores dificuldades ocorreram nas tarefas 3 e 4, para cuja concretização foi solicitado um maior número de ajudas.

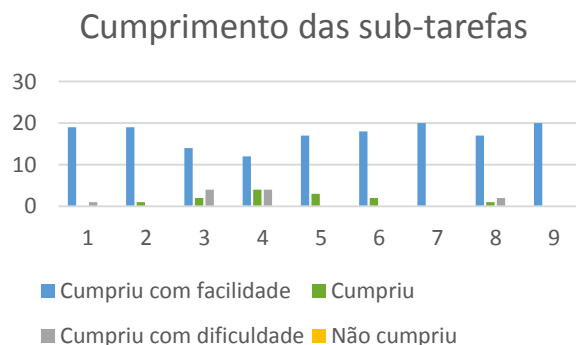


Figura 14: Resultados das sub-tarefas realizadas

Na Tabela 2 são apresentados os resultados do inquérito de satisfação a que os participantes responderam no final da sessão, que consistia em pontuar um conjunto de 7 afirmações sobre o sistema utilizado. As pontuações tomam valores inteiros numa escala de 1 a 5, correspondendo, respetivamente, às respostas: “discordo totalmente”, “discordo”, “indiferente”, “concordo” e “concordo totalmente”. Tanto os estudantes como os peritos consideram que o sistema desenvolvido facilita a visualização dos modelos 3D e a interação com estes e que é útil no processo de conceção em arquitetura, apesar de os peritos concordarem de forma menos assertiva que os estudantes.

No inquérito de satisfação, os participantes pronunciaram-se também quanto ao grau de confiança que depositam na tecnologia que utilizaram e às características que mais e menos valorizaram.

Afirmção:		Média	Mediana
1. Esta solução facilita a visualização e interação com modelos 3D em arquitetura.	E	4,6	5
	P	4	4
2. Esta solução promove o teste de novas ideias na fase de conceção	E	4,2	4
	P	3,8	4
3. A utilização da aplicação acrescenta informação ao processo de conceção	E	3,9	4
	P	3,8	4
4. A utilização da aplicação dá-me um maior controlo sobre o projeto	E	3,8	4
	P	3,8	3,5
5. A utilização da aplicação torna mais compreensível o projeto durante o processo de conceção	E	4,2	4
	P	3,8	4
6. Na generalidade, considero útil a aplicação no processo de conceção em arquitetura	E	4,2	4
	P	3,8	3,5
7. Sinto-me disposto a aprender a utilizar estas tecnologias /dispositivos?	E	4,4	5
	P	4,3	4

Tabela 2: Análise dos resultados do inquérito de satisfação realizado aos estudantes (E) e peritos (P)

Verificou-se que, em geral, que os participantes têm um grau de confiança médio-alto. As características mais valorizadas pelos estudantes foram a facilidade de utilização, a facilidade de demonstração de projetos e de visualizar novas ideias. Os peritos valorizaram as duas primeiras características referidas mas não deram relevância à facilidade de visualização de novas ideias.

Os participantes responderam também a um inquérito sobre a usabilidade da aplicação, cujos resultados são apresentados na Tabela 3. A escala utilizada foi de 1 (“muito má”) a 6 (“muito boa”), sendo os valores intermédios correspondentes a “má”, “insuficiente”, “razoável” e “boa”, respetivamente. O grau de adequação das funcionalidades da aplicação para a realização das tarefas propostas, foi elevado, em ambos os grupos de participantes, sendo de realçar a elevada pontuação atribuída à facilidade de aprendizagem de utilização.

Classificação da aplicação quanto:		Média	Mediana
1. Facilidade de utilização	E	4,9	5
	P	5,3	5,5
2. Grau de satisfação com a sua utilização	E	4,7	5
	P	4,8	4,5
3. Facilidade de aprendizagem	E	5,2	5,5
	P	5,3	5
4. Obtenção dos resultados esperados (como por exemplo, queria fazer um corte e conseguiu)	E	5,3	5
	P	5,3	5,5

5. Semelhança da forma de funcionamento nas diferentes tarefas (como por exemplo aceder a diferentes opções)	E	5,0	5
	P	5,0	5
6. Possibilidade de interagir pelo toque	E	5,0	5
	P	4,8	5
7. Entendimento dos textos apresentados	E	4,9	5
	P	5,5	5,5
8. Tamanho dos botões apresentados	E	5,4	5,5
	P	5,5	5,5
9. Respostas da aplicação às suas ações	E	4,7	5
	P	4,8	4,5
10. Saber o que estava a acontecer na aplicação durante a sua utilização	E	5,0	5
	P	5,0	5
11. Grau de satisfação global com a aplicação	E	5,3	5
	P	5,5	5,5

Tabela 3: Análise do inquérito de usabilidade realizado aos estudantes (E) e peritos (P)

Apesar dos resultados serem todos positivos, os participantes classificaram o grau de satisfação com a sua utilização e a resposta da aplicação às ações com valores mais baixos. Estes valores referem-se a pequenas falhas na interface, como a falta de *feedback* visível do toque. Face a essas falhas, foram sugeridas pelos participantes algumas melhorias, como, por exemplo, uma melhoria ao nível do *design* da interface gráfica da aplicação.

5. CONCLUSÕES E TRABALHO FUTURO

Neste artigo descrevemos as fases de levantamento de requisitos, desenvolvimento, teste e avaliação de um sistema de RA para apoio ao projeto de Arquitetura, ARch. Este sistema visou demonstrar que a visualização e interação com modelos virtuais 3D de edifícios, utilizando realidade aumentada disponibilizada num *tablet*, é benéfica para o processo de conceção em arquitetura.

Terminado o desenvolvimento, foi avaliada a sua usabilidade e satisfação e analisaram-se os resultados obtidos. Esses resultados, apresentados em pormenor na secção 4.2, indicaram que os participantes gostaram da aplicação e acharam-na fácil de usar, considerando que esta facilitava a visualização e interação com os modelos 3D em arquitetura. Os participantes também concordaram que a solução promovia o teste de novas ideias na fase de conceção, tornando o projeto mais compreensível durante esse processo. Na generalidade, consideraram a aplicação útil no processo de conceção em arquitetura.

Tendo em conta os resultados da avaliação podemos concluir que esta solução é benéfica para o processo de conceção em arquitetura, do ponto de vista de análise iterativa de conceitos e ideias e da compreensão e comunicação do projeto.

Como trabalho futuro, está prevista a introdução de várias melhorias na aplicação desenvolvida e o desenvolvimento de novas funcionalidades. Serão tidas em conta as sugestões dos participantes no estudo para melhorar o *design* da interface gráfica do utilizador. Quanto a novas funcionalidades, prevê-se a possibilidade de controlo da iluminação do modelo por forma a simular diferentes alturas do dia ou do ano, bem como a inserção de objetos animados que permitam, por exemplo, simular o movimento de pessoas ou de carros na rua. Estas funcionalidades permitirão aos arquitetos melhorar a experiência de utilização do sistema.

Outra possibilidade de desenvolvimento futuro é a substituição da marca visual planar, constituída por uma imagem texturada, por uma marca 3D (a própria maquete física), unificando o uso de maquetes e o uso de modelos 3D num novo conceito para conceção em arquitetura. Para isso, será necessário incorporar no sistema um sensor Kinect [Microsoft14], para fazer a reconstrução 3D da maquete física, e substituir o módulo de seguimento da aplicação, atualmente baseado na API NUTTS, por um novo módulo, utilizando a API KinectFusion [Izadi11].

6. REFERÊNCIAS

- [Am14a] AR-media. AR-media™ - Product ARPlayer, 2014. Consultado em Março de 2014 em <http://www.ar-media.it/arplayer>
- [Am14b] AR-media. 2014. AR-media™ - Product ARPlugin for 3ds Max. Consultado em Março de 2014 em http://www.armedia.it/3dsmax_arplugin
- [Amim07] Amim, R. R. *Realidade aumentada aplicada a arquitetura e urbanismo*. PhD Thesis, 2007.
- [Azuma97] Azuma, R. T. A survey of augmented reality. *Presence*, 6(4):355–385, 1997.
- [Bastos05] Bastos, R., Dias, J.M.S. Fully Automated Texture Tracking Based on Natural Features Extraction and Template Matching. *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, páginas 180–183. ACM, 2005.
- [Bastos08] Bastos, R. Automatic camera pose initialization, using scale, rotation and luminance invariant natural feature tracking. *Journal of WSCG*, 16(1-3):97–104, 2008.
- [Bastos09] Bastos, R e Dias, M. S. *FIRST - Fast Invariant to Rotation and Scale Transform: Invariant Features for Augmented Reality and Computer Vision*. VDM Verlag, 2009.
- [Bay08] Bay, H., Ess, A., Tuytelaars, T., Gool, L. V., "SURF: Speeded Up Robust Features", *Computer Vision and Image Understanding (CVIU)*, Vol. 110, No. 3, pp. 346–359, 2008.
- [Belcher08a] Belcher, D. e Johnson, B. R. An augmented reality interface for viewing 3d building information models. *26th eCAADe Conference Proceedings*, páginas 561–568, 2008.
- [Belcher08b] Belcher, D. e Johnson, B. Mx: A physical model-based mixed reality interface for design collaboration, simulation, visualization and form generation. *Proceedings of the 28th ACADIA Conference*, páginas 464–471, 2008.
- [Broll04] Broll, W., Lindt, I., Ohlenburg, J., Wittkämper, M., Yuan, C., Novotny, T., Mottram, C., Fatah gen Schieck, A. e Strothman, A. Arthur: A collaborative augmented environment for architectural design and urban planning. *Journal of Virtual Reality and Broadcasting*, páginas 1–10, 2004.
- [Broschart13] Broschart, D., Zeile, P. e Streich, B. Augmented reality as a communication tool in urban design processes. *Proceedings REAL CORP 2013 Tagungsband*, páginas 119–126, 2013.
- [Freitas10] Freitas, M. R. e Ruschel, R. C. Aplicação de realidade virtual e aumentada em arquitetura. *Arquitetura Revista*, 6(2):127–135, 2010.
- [Graf10] Graf, H., Santos, P. e Stork, A. Augmented reality framework supporting conceptual urban planning and enhancing the awareness for environmental impact. *Proceedings of the 2010 Spring Simulation Multiconference*, página 181. Society for Computer Simulation International, 2010.
- [Izadi11] Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D. e Davison, A. KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. *24th Annual ACM Symposium on User Interface Software and Technology*, páginas 559–568. ACM, 2011.
- [Kato99] Kato, H. e Billinghurst, M. Marker tracking and HMD calibration for a video-based augmented reality conferencing system. *2nd IEEE and ACM International Workshop on Augmented Reality (IWAR '99)*, páginas 85–94. IEEE, 1999.
- [Microsoft14] Microsoft. Kinect for Windows | Voice, Movement & Gesture Recognition Technology, 2014. Consultado em Janeiro de 2014 em <http://www.microsoft.com/en-us/kinectforwindows/>
- [OpenGL14] OpenGL. OpenGL Overview, 2014. Consultado em Março de 2014 em <http://www.opengl.org/about/>
- [Sareika10] Sareika, M. *Urban sketcher: mixing urban realities using mixed reality technology*. Thesis, 2010.
- [Veliz12] Veliz, A., Kocaturk, T., Medjdoub, B. e Balbo, R. Dialogs between physical and digital modelling methods on architectural design. *Proceedings of the 30th eCAADe Conference*, Vol. 2:281–289, 2012.
- [Wang08] Wang, X., Gu, N. e Marchant, D. An empirical case study on designer's perceptions of augmented reality within an architectural firm. *Journal of Information Technology in Construction*, 13:536–551, 2008.
- [Wang10] Wang, R. e Qian, X. *OpenSceneGraph 3.0: Beginner's guide*. Packt Publishing Ltd, 2010.
- [Wang13] Wang, X., Kim, M. J., Love, P. E. D. e Kang, S.-C. Augmented reality in built environment: Classification and implications for future research. *Automation in Construction*, 32:1–13, 2013.
- [Zhou08] Zhou, F., Duh, H. B.-L. e Billinghurst, M. Trends in Augmented Reality Tracking, Interaction and Display: A Review of Ten Years of ISMAR. *7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, páginas 193–202. IEEE Computer Society, 2008.

3D Reconstruction and Visualization of Liver and Vascular Networks from CT data using VTK and IGSTK

João Fradinho Oliveira
C3i/Instituto Politécnico de Portalegre
Portalegre, Portugal
jfoliveira@estgp.pt

Hugo Capote
Hospital Dr. José Maria Grande
Portalegre, Portugal
hugocapote@gmail.com

José Luis Moyano-Cuevas José Blas Pagador
Bioengineering and Health Technology Unit
Jesús Usón Minimally Invasive Surgery Center
Cáceres, Spain
{jmoyano, jbpagador}@cmmijesususon.com

Francisco Miguel Sánchez-Margallo
Jesús Usón Minimally Invasive Surgery Center
Cáceres, Spain
msanchez@cmmijesususon.com

Abstract

Spatial reasoning of vascular structures in organs such as the liver is an imperative task performed pre-operatively in resection planning when minimising risks of bleeding in a procedure and intra-operatively during surgery. Accurate automatic 3D reconstruction of surfaces from computerized tomography (CT) contours is complex or impossible without user intervention. Often the gap between scan slices is large enough to make contour correspondence between adjacent slices hard to establish and branching difficult to determine. Freely available open source libraries such as the image guided surgery toolkit and the visualization toolkit (IGSTK and VTK respectively) provide building blocks that enable one to speed up the development time whilst allowing one to focus on new algorithms that might help the user. In this paper we present a new automatic solution for visualization/spatial reasoning of vascular networks within the liver that uses two separate 3D reconstruction approaches respectively. In order to make the system automatic, instead of creating contour correspondences where often crucial data is missing between slices we create a layered approach, where the surface of the liver is represented as one or more layered closed surfaces, and vascular networks where correspondence is more complex are represented as stacks of extruded individual contour blocks. Since the geometric primitive used in either reconstruction is the triangle, other algorithms such as collision detection in resection planning can be used.

Keywords

3D reconstruction, IGSTK, VTK, computerized tomography

1. INTRODUCTION

Liver imaging has multiple applications such as measurements of liver volume [Sánchez-Margallo11]; diagnosis and quantification of tumours and other diseases [Goryawala12]; surgical planning prior to hepatic resection or surgical navigation systems [Maeda09]. All these applications need to previously solve the problem of the segmentation of the liver and the problems associated with reconstruction from CT contour polygons broadly set as contour correspondence, branching and tiling [Meyers94].

Segmentation

Segmentation of the liver is a complex task in the CT image because of the noise, the variations of grey level inside the liver, the morphological alterations and similarity of grey levels with other neighbouring

structures. There are different types of segmentation depending on the user participation in the process: manual, semiautomatic and automatic. In manual segmentation, the user draws the contour of the anatomical structure in all slices of the study. These contours must be performed by an experienced radiologist because sometimes the limits between the liver and neighboring structures is complicated. Semiautomatic and automatic segmentation are desirable since the contour annotation time is greatly reduced in a system that enables editing of any error in automatic results. The interest in the segmentation methods is reflected on the Segmentation of the Liver Competition 2007 (SLIVER07 [Sliver14]). This competition started as part of the workshop **3D Segmentation in the Clinic: A Grand Challenge (Miccai 2007)**, and aims to compare different algorithms to segment the liver from clinical 3D

computed tomography (CT) scans [Heimann09] , [Sliver14].

Contour correspondences and branching

A polygon contour on one slice can represent a section of an organ that is physically connected to another section/polygon contour on a different slice. The contour could be the bottom or top part of an organ which is normally implicit if there is no branching. Vascular networks have considerable branching, thus the problem of finding contour correspondence is harder.

Tiling

How vertices from one contour connect to vertices of a contour in a different slice to form triangles/tiles.

The following properties are desirable in a reconstruction system:

- the representation spatially represents the correct geometric level of detail for the task.
- fast to render.
- portability, the models/representation can be used in other systems.
- minimises the required user intervention in establishing contour correspondence, branching, correcting polygon tiling.
- resilient (errors do not compromise the whole model).

In this paper we present two different reconstruction solutions. A surface based reconstruction method for the reconstruction of the liver and simple genus objects of interest, and a block like structure for the reconstruction of vascular structures. For the surface reconstruction we take advantage that there is mostly an implicit 1 to 1 correspondence between all contours of the liver, thus not requiring user intervention establishing the links, for the vascular structures where branching is predominant we stack vertically individual extruded contours of each slice thus not requiring user intervention for establishing correspondence or tiling corrections. Three tools were developed in the scope of this work: a contour annotation tool (a), an intra-operative visualization/Navigation tool (b) and a simple resection planning tool (c). Re-using VTK (a, c) and IGSTK (b) functionality for different components in these tools was desirable for future projects but also presented some less well documented challenges that are also presented in this paper.

Our contour annotation tool uses VTK functionality to load, display DICOM data and specific widgets for spline editing. Our annotation tool exports both kinds of reconstructions into IGSTK's mesh and ply format. We use IGSTK functionality in a separate Navigation tool to amongst other tasks: load and visualize patient DICOM data, control the rendering opacity of loaded objects, and communicate with surgical tool position trackers in an intra-operative or training setting. Since the geometrical

primitive of either exported reconstructions is the triangle, we also show an example of collision detection in the Resection planning tool that also uses VTK functionality.

In section 2 we review related work on 3D reconstruction from polygon contours, in section 3 we present our system, in section 4 we present results and discuss future work and conclusions in section 5.

2. RELATED WORK

Several 3D reconstruction methods have been published. We focus on methods that assume that the manual or automatic segmentation of polygon contours on the CT data has already been made.

Amenta et al. [Amenta00] use a 3D Voronoi diagram to create a surface that has the same topology of the underlying point sample surface. Hoppe et al. create surfaces that are at a zero distance from the tangent planes of the surface [Hoppe92]. Whilst these reconstruction methods work with unorganized points, points derived from CT polygon contours present interesting modeling problems in that not enough slices exist to always capture where an organ starts or where exactly an object physically branches. Meyers decomposes the problems of reconstruction from CT polygon contours into three problems [Meyers94]: a) finding the correspondence between contours of different slices; b) the triangle tiling problem, the stitching of points between different slices and c) contour branching problem. Christiansen et. al measure the area of overlap of any polygon pair from adjacent slices and establishes a correspondence with a threshold [Christiansen78], Meyers also presents a method for automatically finding correspondences between contours, however the level of user correction required is not clear.

Regarding the problem of tiling, early work by [Keppel75] uses graph theory to find tiling arrangements that maximize the volume of the surfaces, Fuchs searches a directed graph, where the edge weights are surface area, finding the shortest path minimizes surface area [Fuchs77]. Christiansen et. al presents an elegant method for tiling that minimises the diagonal length of the connecting triangles with a simple heuristic for branching, branch pairs are grouped into one polygon, and a new vertex half-way the closest points between the branches is inserted [Christiansen78].

A comprehensive review of branching algorithms can be found in [Bajaj95]. Itk-snap [Itk14] is a tool that reads medical imaging files (DICOM) and allows one to annotate polygon contours and create a 3D model. Organs and vascular networks are built in the same way, with a vertical stack of individual extruded polygon contours (blocks). Simply extruding polygon contours eliminates the need to establish polygon correspondence, which with vascular networks could be quite time consuming. It is not clear however whether the organs and vascular

networks can be visualized at the same time with the added visual clutter of geometry internal to these blocks, it is also not clear whether the top and bottom of each block is triangulated with a centroid strategy which in some polygons might not be internal to a polygon, or if a more elaborate triangulation was used. We adopt a similar stack strategy for reconstructing vascular structures using ear clipping triangulation and reconstruct the surface of the liver rather than use blocks to avoid internal visual clutter of internal block structures that would prevent viewing the internal vascular objects.

IGSTK [Igstk14] builds on VTK, and provides useful functionality for creating intra-operative applications such as classes that communicate with known ultra sound devices (e.g. `igstk::UltrasoundProbeObjectRepresentation`) and has a class that represents vascular structures (`igstk::VascularNetworkObjectRepresentation`) as a network of tubes, where `vtkPolyLines` define the centerline of tubes rather than building them from polygon contours. Our system uses VTK functionality directly for DICOM file reading and viewing, and uses a useful `vtkwidget` for manipulating the polygon contours. Our system outputs geometry that can later be read in IGSTK. Other, volume based methods such as the Marching cubes exist, but are beyond the scope of this paper. Such methods create iso-surfaces in various cells, whereas surface based approaches enable the definition of iso-contours/polygon contour marking in each slice and then create the iso-surface from these contours [Bajaj95].

3. SYSTEM ARCHITECTURE

Our contour annotation tool, that exports a triangulated surface of an organ (a) or block models of vascular networks (b) is composed of four components:

- contour creation/editing (a, b)
- contour correspondence and branching (only optional in a)
- tiling algorithm (a, b)
- contour triangulation (a, b)

In other words the reconstruction of surfaces that do not have branching requires first contour annotation by the user (section 3.1), the algorithm then processes each slice and checks for polygons, if there is one, it checks to see if there are polygons present in the next or previous slice. If no polygons are present in adjacent slices the algorithm triangulates the hole from above or below (section 3.4) depending on whether it is the top or bottom slice of the organ. After the contour triangulation checks, the polygon is tiled with the polygon in the next slice using the minimal spanning diagonal criteria (section 3.3).

For the reconstruction of vascular structures, after polygon contour annotation (section 3.1), each slice is checked for polygons, if there is one, this contour polygon is simply copy and pasted automatically to the

next slice and tiled together (section 3.3). Each contour polygon is then always triangulated from below (section 3.4) and its extruded counter part is triangulated from above (section 3.4). Branching of surfaces is supported but requires user intervention to establish polygon correspondences (section 3.2).

3.1 Contour annotation

For both vascular models and organ models one first needs to manually define the contour polygons on each slice, or read a previous automatic or manual segmentation created by a different system. Our file layout supports multiple contours per slice and can record any type of branching correspondences. As mentioned previously our contour annotation tool uses VTK to load and view DICOM CT scans (`vtkDICOMImageReader` and `vtkImageViewer2` respectively). VTK provides a slider bar (`vtkSliderWidget`) that enables one to use the mouse to change the slice being viewed. We use a spline based widget (`vtkContourWidget`) to annotate the polygon contours (Figure 1). In our system, each slice can store multiple contour widgets, and the control points of each can be saved finished or unfinished to file for future editing operations.



Figure 1. Polygon contour annotation.

To speed up annotation, an option to copy and paste all the contours from another slice was added. VTK allows for different interaction styles to be defined to interact with different components, left click drag over the DICOM image changes the brightness of the image, this is particularly useful for the radiologist when segmenting the image/defining polygon contours. Right click drag over the image zooms the image. A left click inside the image creates a polygon control point, right click inside the image displays the VTK coordinates of the point click in the lower left of the window.

When extending the `vtkslider` widget we noticed that the slice number label on the slider did not correspond to the actual slice being viewed. To fix this we disabled the label and set it to invisible, we then used the slider image value to create the correct slice number displayed as text below the slider.

One problem we found with `vtkImageViewer2` was that the z coordinate calculation during picking would give the same z value for different slices. We solved this problem by computing the z value according to the

current slice number with respect to the z origin of the first slice (`reader->GetDataOrigin()`, `zorig`), the orientation of the data (`reader->GetImageOrientationPatient()`), and the z spacing between slices (`reader->GetPixelSpacing()`, `deltaz`).

We note that the first slice in this `vtkImageViewer2` widget corresponds to the last slice in IGSTK's DICOM viewer. In addition, with `vtkImageViewer2` the positive Y points up, whereas in IGSTK the positive Y points down. IGSTK adopts a LPS reference frame centered on the patient, left is the left of the patient (Figure 2). Since we wish to overlay the 3D models and DICOM data in IGSTK, the output vertex coordinates need to be converted to the target reference frame. The following equation computes z for each slice.

```
int nx=n-1; //n is the total number of slices
for(int i=0; i<n; i++)
{
    zcurrent=zorig+( ((float) (nx-i))*deltaz );
}
```

It is tempting to simply convert Y, by subtract Y from the the vertical image size in pixels, however the coordinates of the points in the VTK widget are projected coordinates, hence we need to subtract Y from the maximum Y display value instead. This can be retrieved with:

```
double *bounds = imageView->GetImageActor()->GetDisplayBounds();
```

Some DICOM files have a position offset stored, so the final x, and y conversion is:

```
float *offset=reader->GetImagePositionPatient();
p[0]=(p[0])+offset[0];
p[1]=bounds[1]-p[1]+offset[1];
```

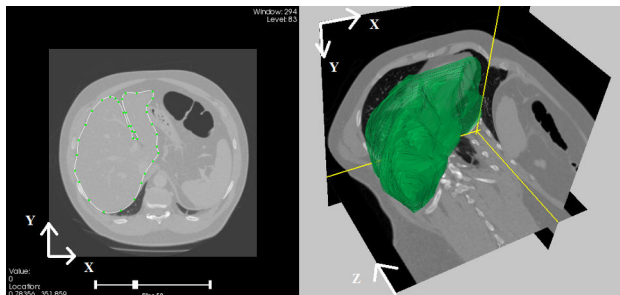


Figure 2. left) VTK coordinate system;right) IGSTK LPS coordinate system.

Finally both types of our reconstructed models can have a low and high resolution version. The low resolution versions simply extract the control points from the contour widget:

```
vtkPolyData *polyData(0);
polyData = vtkPolyData::New();
crep->GetNodePolyData(polyData);
```

Whereas the high resolution versions extracts additional points that make the curves more smooth:

```
polyData = crep->GetContourRepresentationAsPolyData();
```

Before any tiling or polygon triangulation can take place, it is important to establish a consistent vertex ordering in

all the contours. The user is likely to make polygon contours with arbitrary orientations, hence we enforce a counter clockwise ordering by firing a ray perpendicular to the first segment (V2-V1) and count the number of intersections (alike [Oliveira02] but in 2D); if there is an odd number of hits it was counter clockwise, if there are zero or an even number of hits it is clockwise and the vertex order needs to be reversed.

3.2 Contour correspondence and branching

Contours can be quite far apart horizontally from one slice to the next, making it hard [Bajaj95] or impossible for automatic 3D reconstruction solutions to know when is a contour supposed to stitch to another, or to simply end and start a new structure. Additionally the proximity of contours can make it equally difficult to establish how a structure actually branches.

For the vascular network reconstruction the user does not need to establish contour correspondence, as each individual extruded contour block is systematically stacked over each other.

For the surface reconstruction of the liver we point out that the genus of the object is quite simple, and that the outer surface typically only branches once at one of the lower tips of the organ. Since we wish to see the vascular networks inside the surface of the liver, the alpha blending actually allows for this extra organ tip to be treated as a separate closed surface. Much like Metaballs[Shen86] that are a collection of overlapping spheres representing the shape of an object, treating the organ as two separate objects has one considerable benefit in that a 1 to 1 correspondence can be inferred, instead of requiring the user to define all correspondences in the model (the user is likely to define the contours corresponding to each branch in arbitrary order, making the correspondence not automatic anymore, in addition with branching. simply checking whether the next slice has a polygon contour or not is not enough to establish whether the polygon is the top or bottom of a structure as shorter branch structures terminate before the main branch and this needs to be signalled by the user). To avoid considerable user intervention establishing the contour correspondence, the user merely needs to copy and paste the first polygon contour of the branch into the slice with the last section before the branch thus creating two overlapping objects. Each organ keeps a separate polygon contour definition without the other contours. Nevertheless we implemented branching, which affects slightly how the contour polygons are tiled and is explained in the next section. The user can establish correspondence between two polygon contours by first right-click a position near a polygon and pressing the 'c' key, the user can then change the slice being viewed, select a different contour in the same way and press 'v' to create a post-connection and a pre-connection between contours according to depth. To speed up the annotation of contour correspondences, when the contours of a slice are copy and pasted to the next, the 1-1 contour

correspondences between identical contours are automatically created. Multiple pre-connections and post-connections can be accumulated for a given polygon, when the user right clicks near a polygon, typically the following is printed in the console:

```
3-0 [4-0] 5-0, 5-1
```

which means that the first polygon contour of slice 4 was selected and this polygon has a correspondence/pre-connection with the first polygon of slice 3, in addition the polygon has a branch post-connection with the first and second polygon contours of slice 5.

3.3 Tiling

Several algorithms exist for tiling [Meyers94]. For the vascular structures we simply duplicate the entire contour polygon into the next adjacent slice, and as a 1 to 1 correspondence exists between the vertices the tiling configuration illustrated in Figure 3-a) was used.

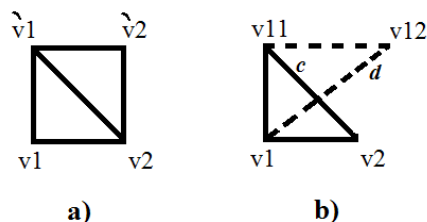


Figure 3. a) Tiling of vascular network polygons/polygon extrusion; b) Tiling in surface based reconstruction using the minimal length diagonal.

For the surface reconstruction we implemented Christiansen et al. minimal spanning diagonal method for tiling [Christiansen78] as it also offered an elegant solution for branching. This algorithm essentially finds two vertices that are in proximity to each other (v_{11} and v_1 in Figure 3-b form a bridge between the different slices) it then measures the length c between v_{11} and v_2 and the length d between v_1 and v_2 , the shortest length indicates the triangle that is created, in this case the triangle (v_{11} , v_1 , v_2) is inserted instead of the triangle (v_{11} , v_1 , v_{12}). The bridge then becomes v_{11} and v_2 , and the length from v_2 to v_2 is compared with the length from v_{11} to the vertex to the right of v_2 and so forth.

For branching, essentially the branched contour polygons are inserted into one ordered polygon that goes through a new vertex. This new vertex is halfway in-depth between the two slices, and is positioned half-way between the closest vertex pair between the branched polygons. The new polygon definition goes through the closest pairs and midpoint twice as it bridges the adjacent polygon and returns (see Figure 7).

3.4 Contour Triangulation

In the surface reconstruction, the first vertical contour and the last vertical contour of a structure need to be triangulated otherwise holes will result in the model. A simple algorithm for triangulating the planar contour,

would be to compute a centroid vertex and create triangles with adjacent contour vertices and the centroid (Figure 4, right). However more complex polygons (Figure 5, right) representing the start of an organ are possible, and the centroid computation might yield a vertex that actually is not inside the contour or can create intersections with other triangles in the contour. We implemented David Eberly's *Triangulation by Ear Clipping* [Eberly08] to address these issues.

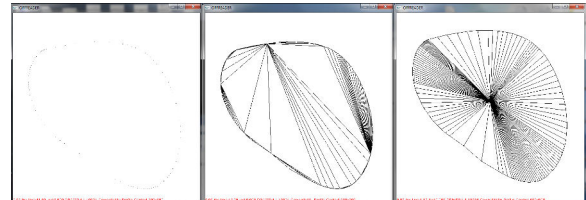


Figure 4. Contour triangulation: left) vertices; middle) Triangulation by Ear Clipping; right) centroid triangulation.

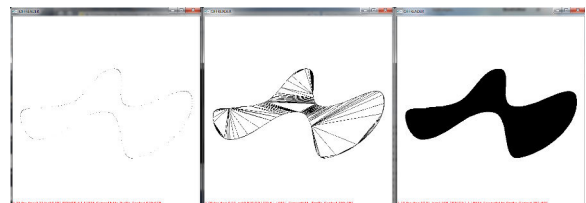


Figure 5. Contour triangulation by Ear Clipping: left) vertices middle) wireframe right) flat shading.

This algorithm works by first computing a list of ear vertices, an ear is a triangle formed by three consecutive vertices of a polygon that does not contain any other vertices inside. He calls this the triangle containment test. As each ear vertex is removed, a triangle is added, and the contour polygon shrinks. A simple polygon of n vertices always has $n-2$ triangles. After removing an ear vertex, the status of vertices adjacent to the ear vertex needs to be updated, vertices that were not ears could become an ear and need to be added to the ear list, and vertices that were ears might stop being an ear and need to be removed from the ear list. Eberly outlines an implementation that is $O(n^2)$.

In the vascular network reconstruction, each contour polygon that is extruded and tiled, has to be triangulated from the top and the bottom creating a collection of wedge like structures.

4. RESULTS

Numerical results are summarized in Table 1. The R column indicates which reconstruction technique was used. S and s indicate models reconstructed with the surface based approach using high resolution vertices of the contours and just the contour points respectively. B and b indicate vascular networks that were built with the block like approach, likewise B used the high resolution vertices and b used just the control points. The Object denoted object was a small object of interest created with high resolution vertices for tumour visualization. With rendering performance in mind, it can be seen that the full

resolution liver surface together with the full resolution vascular networks could present some performance issues in some tasks ($61048+275360=336408$ triangles[liver1 total]). Fortunately the lower resolution versions ($4980+76+11574=16630$ [liver1], Figure 8) and ($3248+550+7000=10798$ [liver2], Figure 9) respectively enable fast rendering and spatially convey all the bifurcations required for risk assessment in a planned intervention. In particular, the alpha blending set in IGSTK to 0.4 enables the additional surface structure in the smaller liver object to be almost unnoticeable whilst freeing the user from the polygon correspondence task all together. We argue that some polygon tiling will simply always create rendering artefacts unless for example a compromise is made in the detail (Figure 6 indicates corresponding problematic tiling locations). No user adjustment attempts were made with the models of Figure 8 and 9 where three out 111 slice connections had some difficulties in Figure 8. These difficulties did not occur with the models shown in Figure 9. Again the alpha blending lessens any visual clutter allowing one to focus on the structures that really matter, however the slice by slice tiling approach proved to have the resilience quality we sought.

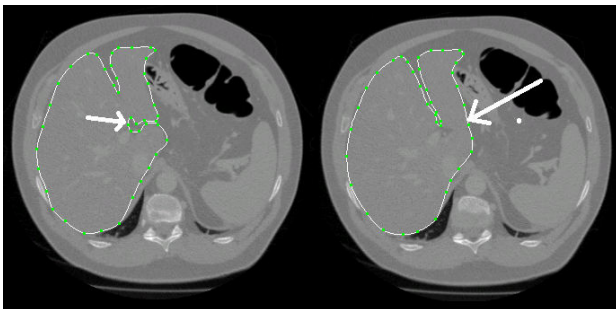


Figure 6. Polygon tiling challenge.

Contour annotation proved to be the longest task in the 3D reconstruction, the copy and paste of contours were only introduced at a later stage and were not used or timed. Manual annotation of 118 contours from 155 CT slices took around 1 hour and 20 minutes to annotate. The system generated a triangulated model (Figure 7) with 61048 triangles and 30526 vertices in under 1 second (Table 1) on a Sony Vaio laptop (Pentium i7-2640M 2.80 GHz with 6GB of RAM). The Liver2 dataset with 74 CT slices, took around 1h30 mins to annotate 59 contours, whilst the liver sub-parts (liver1_b and liver2_b) took around 10-15 minutes to annotate. Both vascular structures took close to 2 hours and 2.5 hours respectively to annotate. We note that the segmentation is inherently a non-linear task as the radiologist needs to browse image patterns in different slices for a given type of object.

Results of the surface reconstruction of the liver overlaid with the surface of the tip of the liver along with the reconstructed vascular network block structures can be seen in Figure 8 and Figure 9. We show an application that uses the reconstructed ply models in Figure 10. For this application we used VTK glyph and implicit plane classes for basic collision detection/resection planning.

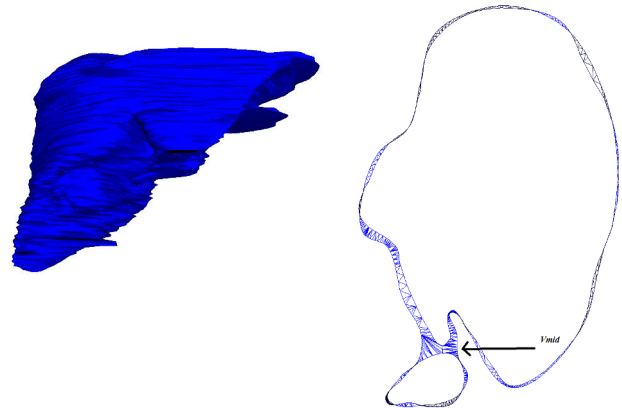


Figure 7. 3D surface reconstructed liver from a pig with 1 branching occurrence. Left) overall model; right) 2 slices with branching and newly inserted vertex Vmid.

Object	R	#loop	#tris	#verts	Time (s)
Liver w/branch	S	118	61048	30526	<1
	s	118	4966	2485	<1
liver1	s	113	4980	2492	<1
	S	113	61964	30984	<1
liver1_b	s	6	76	40	<1
veins1	b	590	11574	7420	<1
	B	590	275360	142338	3
object	S	7	1574	789	<1
liver2	s	59	3248	1626	<1
	S	59	32954	16479	<1
liver2_b	s	19	550	277	<1
veins2	b	300	7000	4388	<1
	B	300	147896	76572	2

Table 1: 3D (R)econstruced models. s/S surfaces (liver, tumor); b/B vascular network; s/b - just uses the control points; S/B - hi-res;

5. CONCLUSIONS AND FUTURE WORK

We presented two reconstruction algorithms for the surface reconstruction of the liver and the vascular structures respectively. We found that the models using only the control points of the polygons are enough to spatially represent the structures for decision making, and allow for fast rendering. Decomposing any branching from a surface into a separate overlapping object produced negligible visual clutter (Figure 8 and 9) while greatly reducing the need for user intervention in contour correspondence and adjusting any tiling mistake in the branching. We believe that the minimal length diagonal can be further improved if the algorithm builds tiles from both sides of the bridge connection, considering the minimal length of either side might allow the tiling to progress correctly until the other side of a problem area. Semi-automatic segmentation is an exciting area of

research where contour selection growing algorithms alike Photoshop's magic wand are possible and the use of texture information could perhaps further help segmentation, the authors would like to explore this area and help reduce the contour annotation time. This work aims to lessen the cognitive load of the surgeon during image-guided surgery. Many laparoscopic surgeries are performed just with the live video and ultrasound imaging. Our intra-operative setup uses our IGSTK customized navigation application to view the reconstructed models and communicate/visualize with the surgical instrument position trackers whilst the live-video from the endoscopic camera is broadcast in a separate viewing window. The system is currently ongoing clinical trials using pigs, but if successful we aim to use the system with humans. We are planning to make the code available open source.

6. ACKNOWLEDGEMENTS

This project is funded by Programa de Cooperación Transfronteriza España Portugal (POCTEP) and Fondo Europeo de Desarrollo Regional (FEDER) Reference code: 0401_RITECA_II_4_E.

7. REFERENCES

- [Amenta00] Amenta, N., Choi, S., Dey, T. K., Leekha, N., A Simple Algorithm for Homeomorphic Surface Reconstruction, In proc. of the sixteenth annual symp. on Computational geometry, 2000, 213-222.
- [Bajaj95] Bajaj, C. L., Coyle, E. J., Lin, K-N, Arbitrary Topology Shape Reconstruction from Planar Cross Sections, Computer Science Tech. Reports. Paper 1205. 1995. <<http://docs.lib.purdue.edu/cstech/1205/>>
- [Christiansen78] Christiansen, H. N., Sederberg, T. W., Conversion of complex contour line definitions into polygonal element mosaics, Computer Graphics XIII,2(August, 1978), 187-192.
- [Eberly08] Eberly, D., Triangulation by Ear Clipping, 2008. <<http://www.geometrictools.com/>>
- [Fuchs77] Fuchs, H., Kedem, Z. M., Uselton, S. P., Optimal Surface Reconstruction from PlanarContours, Communications of the ACM, XX-10(1977),693-702.
- [Goryawala12] Goryawala, M., Guillen, M. R., Cabrerizo, M., Barreto, A., Gulec, S., Barot, T.C., Suthar, R.R., Bhatt, R.N., McGoron, A., and Adjouadi, M, A 3-D Liver Segmentation Method with Parallel Computing for Selective Internal Radiation Therapy. IEEE Trans. Inf. Technology in Biomedicine, 16 (1), 2012, 62-69.
- [Heimann09] Heimann, T., et al., Comparison and Evaluation of Methods for Liver Segmentation from CT datasets. IEEE Transactions on Medical Imaging, volume 28, number 8, 2009, 1251-1265.
- [Hoppe92] Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W., Surface reconstruction from unorganized points. SIGGRAPH'92, 71-78.
- [Igstk14] IGSTK, The Image-Guided Surgery Toolkit, 2014, <<http://www.igstk.org/>>
- [Itk14] itk-SNAP, 2014, <<http://www.itksnap.org/>>
- [Keppel75] Keppel, E., Approximating complex surfaces by triangulation of contour lines, IBM J. Res. Develop. 19 (Jan. 1975), 2-11.
- [Maeda09] Maeda, T., Hong, J., Konishi, K et al., Tumor ablation therapy of liver cancers with an open magnetic resonance imaging-based navigation system. Surg Endosc 23(5), 2009, 1048-1053.
- [Meyers94] Meyers, D., Reconstruction of Surfaces From Planar Contours, PhD Thesis, University of Washington
- [Oliveira02] Oliveira, J. F., Steed, A., Determining orientation of Laser scanned surfaces, in proceedings of 1st Iberoamerican Symposium in Computer Graphics, SIACG2002, Guimarães, Portugal, 281-288.
- [Sánchez-Margallo11] Sánchez-Margallo, F. M., Moyano-Cuevas, J. L., Latorre, R., Maestre, J., Correa, L., Pagador, J. B., Sánchez-Peralta, L. F., Sánchez-Margallo, J. A., Usón-Gargallo, J., Anatomical changes due to pneumoperitoneum analyzed by MRI: an experimental study in pigs, Surg. Radiol Anat, 33(5), 2011, 389-96.
- [Shen86] Shen, J., Thalmann, D., Interactive shape design using metaballs and splines, SIGGRAPH86, 151-160.
- [Sliver14] Sliver, 2014, <<http://www.sliver07.org/>>

8. APPENDIX A – IGSTK INSTALLATION

In order to compile the IGSTK/VTK libraries with minimal problems on Windows 7, we found that the following steps should be taken:

- Use *git* to download and checkout VTK and IGSTK libraries, instead of any url distribution.
- Turn-off the shared library option in CMake2.8 when compiling every library.

The following libraries need to be installed: IGSTK 5.2, VTK 5.10.0, ITK 4.4.1, QT 4.8.5, FLTK 1.0.11

9. APPENDIX B – APPLICATION

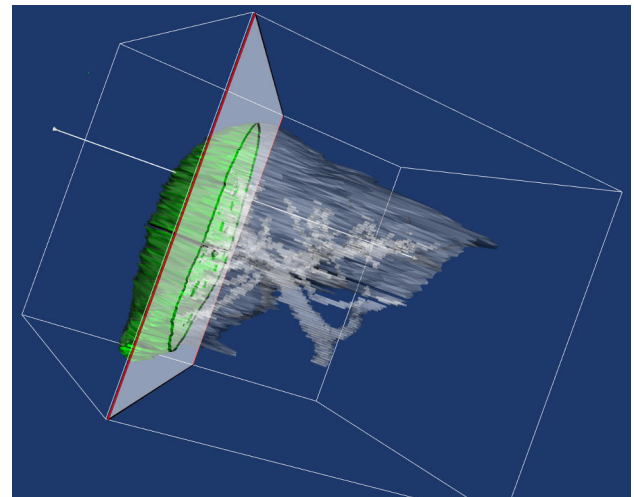


Figure 10. Resection Planning tool with liver1.

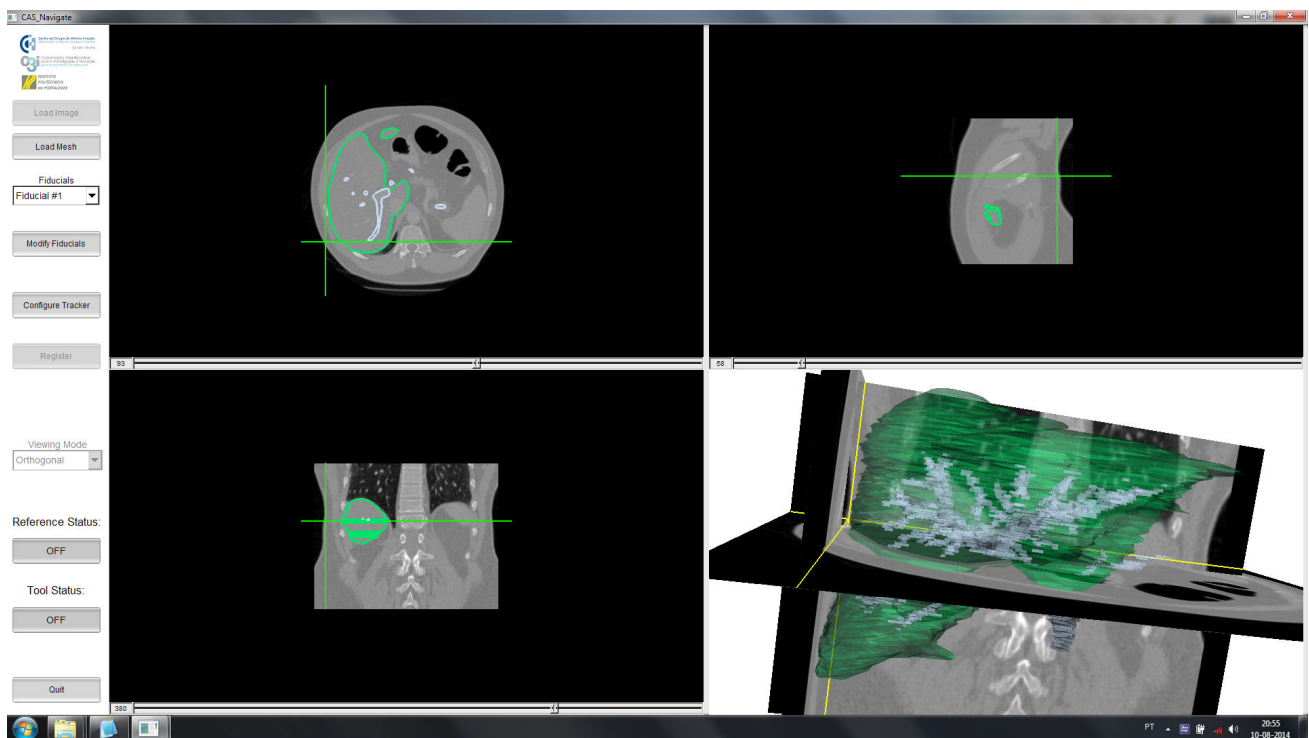


Figure 8. IGSTK visualization of liver1 result: liver1 (4980) + liver1_b (76) + veins1(11574) = 16630 triangles.

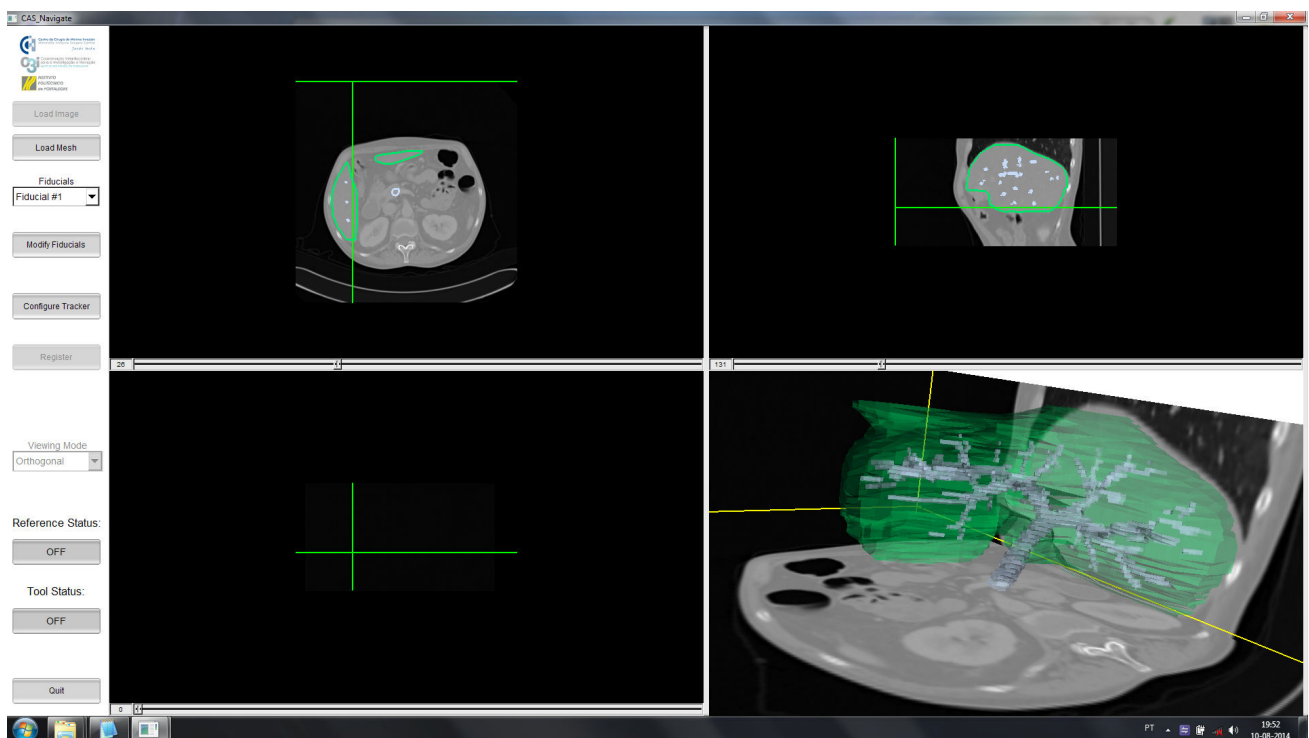


Figure 9. IGSTK visualization of liver2 result: liver2 (3248) + liver2_b (550) + veins2(7000) = 10798 triangles.

Indoor Image Geocoding using Synthetic Views

Nuno Mourão de Amorim Jorge Gustavo Rocha
Departamento de Informática
Universidade do Minho
4710-057 Braga, Portugal

ntma90@gmail.com, jgr@di.uminho.pt

Abstract

The Global Positioning System is well known for not reaching indoor environments. Several Indoor Positioning System's have been proposed, but most of these solutions either have high accuracy errors or use expensive material to attenuate positioning errors. In this paper we propose a Computer Vision routine which is able to compute the location and orientation on indoor environments. This routine is based on Structure from Motion, an incremental algorithm which recovers the 3D structure from related photographs. The 3D structures generated are geocoded, stored in a database, and new photographs can be added at any time. By combining these 3D structures with the already existing Synthetic Views method for fast location recognition, we are able to compute the indoor GPS coordinates and orientation of new photographs in less than a second.

Keywords

Indoor Positioning System, Geocoding, Structure from Motion, Synthetic Views.

1. INTRODUCTION

Used for civil, commercial and military purposes, the *Global Positioning System (GPS)* has proven to be a resourceful and useful service. By using 4 or more satellites, this system use a trilateration process for world wide location recognition. Since the process of trilateration requires the communication between the subject to be located and the satellites, occluded zones (by bad weather or buildings) often hinder this communication which renders the *GPS* ineffective.

To complement the *GPS* in these zones, several *Indoor Positioning System (IPS)* solutions were researched. These solutions were evaluated in several performance metrics were the most relevant are: accuracy, precision, complexity, robustness, scalability and equipment cost. Presently there is not any official *IPS* because the existing solutions does not balance these metrics.

Further tackling the indoor localization problem, the *Computer Vision* community has been supporting the use of *Structure from Motion*, a *Simultaneous Location and Mapping (SLAM)* technique to geo-registrate photographs without prior information of their location. The advantage of this system related to other research's is that it does not require expensive hardware. A simple photograph taken from cellphone is all it is needed. But most of the *SFM* solutions proposed were only tested on outdoor environments, where the *GPS* signal is strong the majority of time. Besides, the implementation of their complete pipeline is not publicly available.

So, to offer continuity to this research, we were motivated to develop a prototype which uses one of the existing *Computer Vision* fast localization methods to perform image geocoding on occluded zones, without any prior information of where photographs were taken.

In this paper we want to prove that already existing methods can be applied into indoor with just few modifications. By developing a prototype, we will offer the necessary tools for the *Computer Vision* community to experiment and improve this research. We also want to show that both scalability and performance may be achieved without expensive hardware.

2. RELATED WORK

The task of indoor localization is related with several areas of research. Taking advantage of *Inertial Measurement Unit's (IMU)* device, [Woodman 08, Patrick 09] proposed an indoor location system using this measurement device. The *IMU* are known for high drifting errors which is accumulated over time. To compensate errors, other research's propose to use a combined *IMU* system with *RFID's* [Ruiz 12] or *Ultra-Wideband* measurements [Hol 09].

By using Ultrasonic waves, [Priyantha 05, Minami 04, Hazas 06] propose solutions which use these high accuracy waves for indoor localization. Although these solutions compute locations with reduced positioning errors, the hardware needed for ultra-sonic waves is expensive.

Due to high range coverage and low cost hardware usage, [J. 00, NI 03, Zhang 10, Das 14] proposed *IPS* solutions

which use radio frequencies for location recognition. Subjects to be located are required to carry a small *RFID* device which acts as a receiver of tracking information. Although *RFID* solutions are able to compute positions on indoor environments, radio frequencies are affected by signal interference caused by infrastructures.

In [Mao 13, Jung 14], a solution for indoor location system using an infrared system was proposed. Subjects to be located are required to use an infrared device which periodically sends information to infrared sensors positioned along the building structure. The nearby sensors which are able to capture the messages, consequently compute the current position of users.

From the presented state of art it is noticeable that either the solution uses expensive hardware to attain precise localization or cheap and inaccurate hardware which is complemented with additional components to attenuate localization errors.

Tackling the localization problem with optical techniques, research such as [Schindler 07, Irschara 09, Li 10, Li 12] use Structure from Motion models for location recognition. Structure from Motion allows the reconstruction of 3D models seen by several related 2D photographs. Since these models store information of photographs which gave them origin, new related photographs may be added by comparing their visual information with the 3D structure. Because querying new photographs to a large database is highly inefficient, [Schindler 07, Irschara 09, Li 12] devised routines which accelerate the query phase by retrieving a set of potential useful information from the database to locate new photographs. If a new photograph validates a pose verification, then it is correctly placed within the model and the *GPS* position is returned. The advantage of these research's is allowing location recognition while only requiring cheap hardware such mobile-phone cameras.

3. IMAGE GEOCODING PROTOTYPE

From the state of art *Computer Vision* solutions, we decided to use the Synthetic Views method [Irschara 09] on our location recognition prototype. This choice was motivated by the result of their work which proves that image based localization in large environments can be made in real time.

Since this method was only experimented on outdoor environments, the following sub-sections offer a brief overview of the inherent problems of indoor *SFM* models. Afterwards we explain how the *SFM* model can be used to retrieve *GPS* coordinates when querying photographs to be geocoded. And the last subsection will provide the explanation of the implementation of our prototype and the available software used to complete the Synthetic Views pipeline.

3.1. Understanding Indoor Models

Although the theory of image geocoding on outdoor and indoor places should be the same, we questioned ourselves why the state of art research do not address experiments

into indoor environments. Driven by this curiosity, we started by exploring the *SFM* process to understand to output of 3D models. To do so, we chose *VisualSFM* [Wu b], one of the available free to use *SFM* software's to reconstruct indoor models. Data sets for both indoor and outdoor environments were gathered and we found that:

- Indoor environments are less descriptive than outdoor. These environments often contain areas with few decoration like blank walls. Since features are defined in areas where a sudden discrepancy on intensity occurs (edges, corners, ridges), indoor images often contain less number of features;
- Indoor narrow areas prevent good baseline photographs. The *SFM* reconstruction strongly depends on a chosen pair of photographs to start an incremental reconstruction. A good baseline pair is often a pair which contains a wide area of common features to support the further positioning of photographs. Due to the existence of narrow rooms, corridors, entrances, it is often difficult to take good baseline photographs.

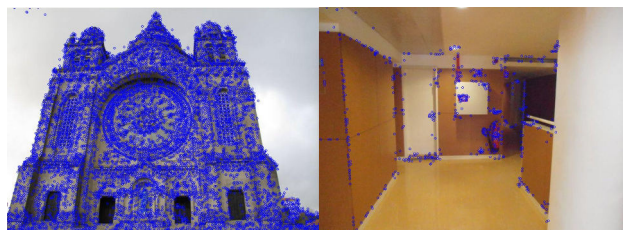


Figure 1. Example of the reduction of features extracted on indoor environments, where each blue point is a feature extracted. At the left, about 10100 features were extracted by how descriptive the church facade is. At the right, only 1621 were extracted.

Due to these facts 3D models were often partitioned into 3 or more sub-models to avoid degeneracy. Although theoretically, these models are still usable to perform image geocoding, practically each sub-model may contain different levels of drifting, which may create pose estimations with different levels of errors even when taken in the same environment. As a workaround to this problem, we increased the number of photographs and decreased the distance between photographs positions, which slightly increased the quality of 3D models. Since more photographs are required, each with less features than outdoor models, we had to adjust some thresholds used in *Synthetic Views*.

3.2. Geocoding with Structure from Motion

So, to geocode a photograph means to retrieve the associated *GPS* information through other geographic information. Using the *SFM* algorithm we are able to recover the structure seen by several photographs and place the



Figure 2. Example of an indoor reconstruction using VisualSFM (at the left). This structure represents an amphitheater (at the right). Analyzing the 3D structure recovered, few 3D points are visible and the structure semantic is hard to notice.

inputted photographs relatively to the generated structure. Although we may add new photographs to the reconstruction by continuing the incremental *SFM* process, none of the outputted data contains geographic information. Therefore, to relate the *3D* referential (where models are reconstructed) to the *GPS* coordinate system (where models are located in the world), the respective *GPS* coordinates of each photograph used on the model reconstruction must be injected in the system. With both coordinate systems (*3D* and *GPS*) available, it is possible to approximate a 4×4 transformation matrix G which maps *3D* positions X into *GPS* coordinates.

$$gps_{coords} = X.G \quad (1)$$

The G matrix can be obtained by computing an affine transformation using a set of coordinates from both *3D* and *GPS* systems. This matrix can also be used to transform the direction vector into *GPS*, since vectors can be represented as two points. In this case, the *GPS* direction can be represented in cardinal directions.

3.3. Synthetic Views Implementation

So, to develop the image geocoding prototype, we started by exploring an existing, uncompleted *SFM* code at [SFM]. Although we did not need the entire *SFM* routine (as we already had *VisualSFM* for reconstructions, we recycled functions which allow pose verification and pose estimations. For feature processing we decided to use *SIFT* [Lowe 04] as a feature descriptor, since it is also used in [Irschara 09].

The Synthetic Views method is divided into 2 phases:

- An offline phase where *3D* models are refined into a compressed yet representative structure, and stored into an image database;
- An online phase where the image database is used to geocode incoming images.

As stated in [Irschara 09, p. 2], *3D* models outputted by *SFM* often contain several redundant descriptors. In order to reduce the feature repeatability, [Irschara 09, p. 2]

propose the use of *Mean Shift Clustering* [Comanicu 02]. This method applies a global threshold to cluster features which have an high level of similarity, without losing the pose estimation capability. In our implementation we used the mean of the descriptors associated to each *3D* point. Although this may lead to a much more aggressive compression, after performing few tests we noticed that the information kept is still capable of allowing a good feature matching rate. As both orientation and scale will be needed afterwards, we also compute their mean for each *3D* point and extrapolate them into the *3D* coordinate system. While the *3D* orientation is given by the mean of the direction of each associated *2D* point to the respective camera view points, the *3D* scale is computed by the following equation.

$$scale_{3D} = \frac{scale * distance_{viewTo3DPoint}}{focallength} \quad (2)$$

After compressing relevant information, artificial cameras are placed uniformly around the *3D* model and each one takes a snapshot. To place each camera, we compute the model ground plane by approximating a plane to the position of each original view. This plane is then divided into a grid and 12 views (each with 30° viewing direction difference) are placed into each grid position. Since indoor models are narrower than outdoor models, we do not tilt the viewing direction of each synthetic view by 10° as in [Irschara 09, p. 4]. For each placed synthetic view, *3D* visible points are re-projected into these cameras and artificial photographs are created, each containing the compressed *2D* and *3D* point information. A *3D* point is only visible by a given synthetic view if:

- Its position lies within the camera view frustum culling;
- The scale of *3D* point is higher than 1 in terms of *Difference of Gaussian (DoG)*;
- The difference of its *3D* orientation with the viewing direction of the current synthetic view is lower than 30° (face culling).

This process is done iteratively until we fill the ground plane with synthetic views. With all the views placed, the best views to represent the model are selected. To do this selection, each synthetic view is evaluated by their view coverage. Both original views and synthetic views are used to build a square binary matrix, where 1 means that a view A covers B and 0 the opposite. For outdoor models we consider the proposed values for view coverage, so, a view A covers B if A contains at least 150 points seen by B . For indoor, to adapt to the reduction of points per view explained in 3.1 this threshold is set to 30 points. Beside this condition, it is also defined that each view covers itself. The best views to represent the *3D* model are retrieved by applying the greedy algorithm described in [Irschara 09, p. 4]).

The information of the best synthetic views are then stored in *3D* documents containing *3D* visible points, their associated *2D* points, compressed descriptors and a *GPS* matrix

G described in 3.2 which is computed before compressing the models. Also, their descriptors are injected into a vocabulary tree to support the geocoding process.

Vocabulary Tree is a structure which allows the creation of an image database, where the descriptors of images are transcribed into 'words' and propagated through a n -ary balanced tree. The main idea behind the vocabulary tree is that similar descriptors will be propagated to the same leafs. When querying an image against the vocabulary tree, the more similarity the propagation is, the higher the probability of 2 images positively match when performing feature matching. Furthermore, this image database structure is scalable and adaptable to higher databases as it allows the creation of trees with more levels (which refines the result of queries) and more branches (which allows a better descriptor distinction). Although [Irschara 09, p. 5] used a vocabulary tree which benefits the use of the GPU to speed up the top document retrieval and their probabilistic scoring function which allows a more precise retrieval of the top best documents to match, we are using the implementation available in [Snaveley] which corresponds to the vocabulary tree researched in [Nist 06]. This decision was based on the fact that we needed the full compression process available as soon as possible and the vocabulary tree document retrieval available at [Snaveley] was outputting the desired results in our experiments.

After compressing enough models, in the online phase new photographs are queried against the image database and a pose estimation is delivered whenever a positive match is returned.

The online phase of our prototype is defined by the following operations:

- Feature Extraction on the query photograph;
- Query vocabulary tree for top matches;
- Feature match between the query image and the synthetic views retrieved;
- Solve *Perspective-N-Point Ransac* to pose estimate the query image;
- Compute the *GPS* position and orientation.

To perform feature extraction and matching we used *SiftGPU* [Wu a] a fast and accurate implementation of *SIFT* [Lowe 04] which benefits the computers GPU and the $CUDA$ toolkit to fasten heavy matrix operations. For feature extraction we extract around 1280 features for the query image, as it is an high enough number to contain relevant features for the pose estimation phase. The descriptors extracted are then queried in the vocabulary tree and the top 10 best documents are retrieved. Feature matching is then performed between the query and 3D documents features. Once a positive match is found, we try to compute the projection matrix by applying the *Perspective-N-Point Ransac* supplied by the *OpenCV* library [Ope] to evaluate

the coherency of the 2D query points with the 3D document points. If the computed projection matrix validates at least 10 inlier points, we define the query image as pose estimated. Using the G matrix stored within the 3D document which matched the current image to geocode, the 3D position and orientation returned by the pose estimation are extrapolated into *GPS* coordinates.

The source code developed for our prototype can be found at [Amorim 14].

4. DATASET AND RESULTS

To evaluate the performance of *Synthetic Views* for geocoding photographs we gathered a data set of 443 indoor and 802 outdoor of geocoded photographs from locations in Braga, Montalegre and Viana do Castelo in Portugal.



Figure 3. Some of the indoor photographs used to evaluate the performance of our prototype.

From these photographs, we built 12 3D models with *VisualSFM*, which resulted in 215403 3D points projected by 808293 features extracted. We then applied the *Synthetic Views* compression on the reconstructed models to remove the excess of information stored in the vocabulary tree. Table 1 shows the compression value obtained with our prototype.

Original Number of Descriptors	808293
Compressed Number of Descriptors	231942

Table 1. Number of descriptors before and after applying the compression.

From the selected locations, another set of 25 indoor and 25 outdoor geocoded photographs were collected and sequentially geocoded to the compressed database. Their associated *GPS* coordinates were only used to compute precision errors between the estimated coordinates and their ground truth. Table 2 provides the estimation rate and time spent on geocoding each photograph and Table 3 presents the average time spent on each operation when geocoding a single photograph. Each query photograph used has 1000 pixels width and 750 height.

These results were attained on a computer with a CPU Intel i5-4200U 1.6 GHz with a GPU nVidia GeForce 820M, while using the $CUDA$ version of *SiftGPU* and a vocabulary tree of 5 levels and 10 branches to store documents.

So, analysing the provided tables, with *Synthetic Views* we were able to compress redundant descriptors on our

	Indoor	Outdoor
Geocode Rate	16 (64%)	22 (88%)
Precision Recall	[0.307, 4.050] m	[0.252, 7.620] m
Mean Precision	1.160 m	2.560 m
Mean Time	0.451 s	0.520 s

Table 2. Overall statistics from pose estimating 25 indoor and 25 outdoor 1000x750 photographs with Synthetic Views.

	Indoor	Outdoor
Keypoint Extraction (1280)	0.216	0.230
Vocabulary Tree (top 10)	0.178	0.181
F. Matcher (1280x2500)	0.015 * N	0.015 * N
PnP Ransac	0.001 * N	0.001 * N
Compute GPS Coordinates	0.0001	0.0001

Table 3. Computational time expressed in seconds for each operation on Synthetic Views. The value N indicates the number of top documents to retrieve. On feature matching, the computational time is based on a comparison of 1280 query features with 2500 document features. Both number of features are based on the mean of features extracted and features stored within 3D documents.

database by more than 71% of its initial value. As stated before, we increased the number of photographs taken per environment and decreased the distance between the photographs positions. This resulted in more stable indoor models, but with an high level of redundancy, which justifies the advantage of the compression phase made by Synthetic Views.

Although the compression rate is high, we were able to geocode 16 indoor and 22 outdoor photographs from our input data set of 50 photographs. Due to the discrepancy of indoor successful geocoded photographs to outdoor, we believe that this method is still not adapted to indoor environments. When compressing our database, we noticed that indoor models required more synthetic views placed compared to outdoor. Since each indoor view shares few features with other views due to reasons explained in 3.1, for some indoor models the 30 points coverage threshold was not allowing a good coverage, which justifies the need of an higher number of views to completely cover indoor models. Lowering this threshold degenerated all indoor models and hindered the geocoding process. As the source of this problem comes from the consistency of indoor models, we propose the use of special wide angle cameras to augment the environment coverage (more features per photograph) and increase the overlap between photographs. Reconstructing indoor environments with these photographs should deliver refined indoor models.

The accuracy of our prototype should also be improved.

Successfully geocoded indoor photographs were located with a mean precision error of 1.160 meters. Outdoor photographs reached the 2.560 meters errors. While, for outdoor models, 2 meters errors may not imply inaccurate localization, for indoor the same errors may place photograph behind walls or bellow floors more often. Here, we proposed the use of Multicore Bundle Adjustment [Wu 11], to locally optimize positioning errors. This algorithm benefits the GPU and CPU to fasten the optimization, which allows more accurate precision while maintaining the geocoding performance.

As for the geocoding speed of this method, with our database we are able to geocode new indoor photographs with a computational time ranging from 0.410 seconds to 0.554 seconds and outdoor photographs from 0.427 to 0.571 depending on which top document query images match. It is natural that outdoor photographs spend more time to be geocoded compared to indoor, since we are processing an higher amount of extracted features.

Although we did not evaluate our prototype with larger data sets, the only operation that should need adjusting is the vocabulary tree querying. As the database increases, more descriptor distinction is required in order to not confuse the top documents retrieved. As stated before, the number of levels and branches of the tree may be adjusted to the stored information.

5. CONCLUSION AND FUTURE WORK

We presented a method which leverages speed with scalability to solve the indoor localization problem. Our contribution was the development of a running prototype. Our indoor geocode system is able to receive images to be geocoded and delivers their *GPS* position and orientation. By making available our code to the *Computer Vision* community we are contributing to the discussion and refinement of state of the art mechanisms used in indoor localization problem.

As future work, we aim to improve our *Synthetic Views* based prototype, by integrating the vocabulary described in [Irschara 09, p. 5] and apply the Multicore Bundle Adjustment on the final geocoding position. We believe that this will greatly boost the performance and refine the accuracy of our system in larger scales.

In the short future, we will prepare and share some data sets, for indoor spaces, to use for benchmarking different tools and approaches. These data sets will improve and speed up the evaluation of new methods.

To further prove the utility of our image based geocoding system, we will develop a client-server service where users can send photographs from their mobile phones, to be geocoded. Our server returns the estimated *GPS* position of the photograph sent.

Image geocoding based on feature prioritization [Li 10] is also being explored in the recent literature. We are very interested in the development of another prototype, based on *Prioritized Features*, to see how it compares in terms of accuracy, speed and scalability, with the existent synthetic

views based prototype.

References

- [Amorim 14] Nuno Mourão Amorim. Image Geocoding Prototype. <https://bitbucket.org/nam81/qt-sfm.git>, 2014. Accessed 20 October 2014.
- [Comaniciu 02] Dorin Comaniciu, Peter Meer, and Senior Member. Mean Shift: A Robust Approach Toward Feature Space Analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, 2002.
- [Das 14] Piyali Das and Dharma. P. Argawal. RFID for Indoor Position Determination. *Instrumentation and Measurement, IEEE Transactions*, pages 561–567, 2014.
- [Hazas 06] M. Hazas and A. Hopper. Broadband ultrasonic location systems for improved indoor positioning. *Mobile Computing, IEEE Transactions*, pages 536–547, 2006.
- [Hol 09] Jeroen D. Hol, Fred. Dijkstra, and Schon Thomas B. Luinge, Henk. Tightly Coupled UWB/IMU Pose Estimation. *Proceedings of the IEEE International Conference on Ultra-Wideband (ICUWB)*, 2009.
- [Irschara 09] Arnold Irschara, Christopher Zach, Jan-Michael Frahm, and Horst Bischof. From structure-from-motion point clouds to fast location recognition. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2599–2606, June 2009.
- [J. 00] Hightower J., Want R., and Borriello G. SpotON: An indoor 3D location sensing technology based on RF signal strength. *UW CSE00-02-02*, 2000.
- [Jung 14] H. Oh. Jung, Kim Doojin, and H. Lee Beom. An Indoor Localization System for Mobile Robots Using an Active Infrared Positioning Sensor. *Journal of Industrial and Intelligent Information*, pages 35–38, 2014.
- [Li 10] Yunpeng Li, Noah Snavely, and Daniel P Huttenlocher. Location Recognition using Prioritized Feature Matching. *ECCV'10 Proceedings of the 11th European conference on Computer vision*, pages 791–804, 2010.
- [Li 12] Yunpeng Li, Noah Snavely, Dan Huttenlocher, and Pascal Fua. Worldwide Pose Estimation using 3D Point Clouds. *ECCV'12 Proceedings of the 12th European conference on Computer Vision*, pages 15–29, 2012.
- [Lowe 04] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, pages 91–110, 2004.
- [Mao 13] Ling Mao, Jiapin Chen, Zhenbo Li, and Dawei Zhang. Relative Localization Method of Multiple Micro Robots Based on Simple Sensors. *International Journal of Advanced Robotic Systems*, 2013.
- [Minami 04] M. Minami, Y. Fukuju, K. Hirasawa, S. Yokoyama, M. Mizumachi, H. Morikawa, and T. Aoyama. Dolphin: A practical approach for implementing a fully distributed indoor ultrasonic positioning system. *Ubicomp*, pages 347–365, 2004.
- [NI 03] Lionel M. NI, Liu Yunhao, Cho Lau Yiu, and Abhishek P. Patil. LANDMARC: Indoor Location Sensing Using Active RFID. *Pervasive Computing and Communications, 2003. (PerCom 2003). Proceedings of the First IEEE International Conference*, pages 407–415, 2003.
- [Nist 06] David Nist and Henrik Stew. Scalable Recognition with a Vocabulary Tree. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2161–2168, 2006.
- [Ope] OpenCV (Open Source Computer Vision). <http://opencv.org/>. Accessed 7 May 2014.
- [Patrick 09] Robertson Patrick, Michael Angermann, and Bernhard Krach. Simultaneous localization and mapping for pedestrians using only foot-mounted inertial sensors. *Proceedings of the 11th international conference on Ubiquitous computing*, pages 93–96, 2009.
- [Priyantha 05] N. B. Priyantha. The cricket indoor location system. *PhD Thesis, Massachusetts Institute of Technology*, page 199, 2005.
- [Ruiz 12] A.R.J. Ruiz, F.S. Granja, J.C. Prieto Honorato, and J.I.G. Rosas. Accurate Pedestrian Indoor Navigation by Tightly Coupling Foot-Mounted IMU and RFID Measurements. *Instrumentation and Measurement, IEEE Transactions*, pages 178–189, 2012.

- [Schindler 07] G. Schindler, M. Brown, and R. Szeliski. City-Scale Location Recognition. *Computer Vision and Pattern Recognition, CVPR '07. IEEE Conference*, pages 1–7, 2007.
- [SfM] Structure from Motion ToyLib. <https://github.com/royshil/SfM-Toy-Library>. Accessed 7 May 2014.
- [Snavely] Noah Snavely. VocabTree2. <https://github.com/snavely/VocabTree2>. Accessed 7 May 2014.
- [Woodman 08] Oliver Woodman and Robert Harle. Pedestrian localisation for indoor environments. *UbiComp '08 Proceedings of the 10th international conference on Ubiquitous computing*, pages 114–123, 2008.
- [Wu a] Changchang Wu. SiftGPU. <http://cs.unc.edu/~ccwu/siftgpu/>. Accessed 8 May 2014.
- [Wu b] Changchang Wu. VisualSfM - A Visual Structure from Motion System. <http://ccwu.me/vsfm/>. Accessed 7 May 2014.
- [Wu 11] Changchang Wu, Sameer Agarwal, Brian Curless, and Steven M. Seitz. Multicore Bundle Adjustment. *CVPR 2011*, pages 3057–3064, 2011.
- [Zhang 10] Dian Zhang, Yanyan Yang, Dachao Cheng, Shiyuan Liu, and Lionel NI. COCKTAIL: An RF-based Hybrid Approach for Indoor Localization. *IEEE International Conference on Communications (ICC)*, pages 1–5, 2010.

Anotações Multimédia Estruturadas para Ambientes Virtuais Colaborativos

João Guerreiro
INESC-ID, DEI/IST/ULisboa
Lisboa, Portugal
joao.p.guerreiro@ist.utl.pt

Daniel Medeiros
DEI/IST/ULisboa
CAPES Found.- Min. of Education of Brazil
Brasília - DF, Brazil
daniel.medeiros@tecnico.ulisboa.pt

Joaquim A. Jorge
INESC-ID, DEI/IST/ULisboa
Lisboa, Portugal
jorgej@tecnico.ulisboa.pt

Alberto Raposo
Instituto Tecgraf, PUC-Rio
Rio de Janeiro, Brasil
abraposo@tecgraf.puc-rio.br

Ismael Santos
Petrobras
Rio de Janeiro, Brasil
ismael@petrobras.com.br

Resumo

A globalização transformou o design de engenharia num processo distribuído geograficamente. A adoção disseminada de Realidade Virtual para o design e a necessidade de agir e deixar notas directamente nos objectos em discussão, levou a projectos sobre anotações em Ambientes Virtuais Colaborativos (AVCs). No entanto, as abordagens convencionais ainda estão limitadas à metáfora do post-it amarelo com texto. Na verdade, multimédia, como áudio, esboços, vídeo e animações permitem uma maior expressividade que poderia ser aproveitada nos AVCs. Além disso, anotações individuais não conseguem captar nem a lógica, nem o fluxo de discussão, que são fundamentais para compreender as decisões de design do projeto. Neste artigo, apresentamos uma abordagem com anotações multimédia estruturadas, que suportam a discussão e tomada de decisão em tarefas de revisão de design de projectos. Além disso, suportamos a argumentação baseada em *Issues* para revelar a proveniência de decisões de projetos de engenharia. Embora esta seja uma exploração inicial do espaço de soluções, exemplos mostram um maior apoio à revisão de projetos colaborativos, em relação às abordagens tradicionais.

Palavras-Chave

Ambientes Virtuais Colaborativos, Anotações Multimédia, CSCW, Revisão de Design, Proveniência

1 Introdução

A globalização transformou o design de engenharia num processo distribuído geograficamente. A adoção de Ambientes Virtuais Colaborativos (AVCs) para a revisão do design de projectos permite o trabalho colaborativo e a melhoria da eficiência [1]. Além disso, suporta reuniões que se centram e atuam sobre os modelos virtuais. A necessidade de colocar marcas diretamente sobre os objetos em discussão, levou à exploração de anotações.

Vários projetos contribuíram com anotações textuais ou esboços colocados no mundo virtual, seguindo a metáfora do post-it + texto (eg, [2, 3]). Estas anotações permitem identificar problemas e sugerir formas de os resolver. No entanto, as abordagens convencionais não têm a expressividade e eficiência de reuniões presenciais, onde os participantes podem recorrer ao discurso, esboços em papel, ou manipular objectos para ilustrar as suas opiniões. De facto, anotações multimédia podem fornecer uma expressividade semelhante através de áudio, esboços e reproduzindo mo-

vimentos de câmara em sincronia com fala e esboços.

Anotações são importantes para influenciar, guiar e documentar as decisões do projecto. Este factor torna-as importantes para as sessões colaborativas subsequentes. Ainda assim, a gestão do conhecimento trocado nas reuniões é um dos grandes desafios em AVCs [6]. A maioria das abordagens apresentam as anotações como itens isolados, o que prejudica uma discussão estruturada e a tomada de decisão. A proveniência constitui um princípio central da nossa abordagem, reflectindo a história de um produto de dados, a partir da sua fonte original [9], para suportar a discussão e revelar o fluxo de decisões.

A engenharia offshore pode beneficiar do uso de anotações multimédia. Projetos que envolvem plataformas de águas profundas, de petróleo e gás, demoram vários anos para ser concluídos e envolvem equipas geograficamente distribuídas. Além disso, estes projectos são caros e têm grandes riscos envolvidos [7]. As anotações multimédia poderiam acompanhar reuniões de revisão de projeto em AVCs, para



Figura 1. É possível criar anotações multimédia com uma estrutura em árvore, com a proveniência das decisões. (a) Anotação no AVC, criada por "Obama". (b) Anotação a ser reproduzida, usando movimentos de câmara, sincronizados com áudio, para mostrar uma anomalia no modelo. (c) Estrutura da discussão, onde os balões indicam o tipo de anotação e as cores o tipo de resposta. O utilizador está a adicionar um Pró através de uma anotação áudio.

enriquecer os argumentos dos participantes.

Neste artigo, apresentamos uma abordagem baseada em anotações multimédia estruturadas para suportar o processo de tomada de decisão em tarefas de revisão de design de projectos. As alternativas multimédia pretendem facilitar a criação de anotações (esboços e áudio; títulos textuais) e aumentá-las com movimentos de câmara, sincronizados. Para além disso, baseamo-nos numa estrutura em árvore e numa argumentação *Issue-Based* para suportar a discussão e revelar a proveniência das decisões de design. Neste processo, é possível complementar uma anotação ou contribuir com argumentos positivos/negativos.

2 Trabalho Relacionado

Anotações são importantes para complementar um AVC com informação adicional. Um exemplo bem conhecido é o *Microsoft Word* e a sua opção de *track changes*. Em AVCs, a literatura reporta uma predominância para anotações textuais semelhantes a post-its ou esboços (p.e. [2, 3]). Embora menos frequente, anotações áudio também são usadas [5].

Uma forma diferente de apoiar as discussões é através de viagens virtuais guiadas [7], onde um utilizador pode partilhar a sua visão do modelo com os restantes. Embora este recurso seja valioso em reuniões síncronas, fica aquém no que diz respeito a interações assíncronas, se não for suportado por informação adicional. A nossa abordagem baseia-se em movimentos de câmara do utilizador (como numa visita guiada), sincronizado com o áudio e esboços, para fornecer anotações multimédia mais expressivas e apoiar interações assíncronas.

Embora úteis para identificar problemas com o modelo, anotações isoladas não suportam a discussão. Um chat convencional tenta fornecer esse suporte [10], mas o seu uso é maioritariamente síncrono. [6] apresenta uma contribuição valiosa, pois adiciona semântica às anotações. No entanto, não é clara a forma como as anotações estão

relacionadas e qual o fluxo da discussão.

3 Anotações Multimédia e Proveniência

As abordagens atuais são insuficientes para fornecer uma plataforma para a tomada de decisão assíncrona em AVCs. Aqui, apresentamos a lógica que suporta a nossa abordagem para anotações multimédia estruturadas.

Revisões de projeto colaborativas requerem suporte para discussão entre os participantes. Estas sessões geralmente são acompanhadas por notas e minutas que apontam os problemas atuais, soluções e próximas etapas. Ao utilizar um AVC, a discussão pode ser centrada no modelo real através de anotações no próprio modelo.

A nossa abordagem suporta anotações multimédia como títulos textuais, esboços e áudio. Além disso, permite a criação de anotações dinâmicas que incluem movimentos de câmara sincronizados com os restantes. Esta sincronia possibilita a reprodução de todos os passos realizados pelo autor da anotação. Estes tipos de anotações permitem-nos criar anotações rapidamente através dos esboços (mais ilustrativos) e áudio, que são mais ricos (e rápidos) que notas textuais (usados apenas em títulos para um *overview* da discussão). Os movimentos de câmara permitem que um participante guie os restantes no AVC para, sincronizado com áudio e esboços, expôr os problemas de um projecto e fornecer sugestões.

As anotações podem apoiar um processo de tomada de decisão e, portanto, são importantes para as sessões colaborativas seguintes. No entanto, anotações isoladas não suportam esses processos e os chats convencionais dificultam discussões paralelas, visto que são seguem apenas uma linha sequencial. Num chat é mais difícil manter o controlo de discussões paralelas ou argumentos incorporados. Estes problemas são exacerbados em condições assíncronas.

Para apoiar a tomada de decisão, cada anotação deve tornar-se parte de um argumento estruturado e proporcionar uma base sólida para toda a discussão. Desta forma,

pode fornecer uma compreensão de todos os argumentos da discussão em curso para identificar as questões em aberto, ajudar na sua resolução e posteriormente revelar o processo usado para resolver um problema. Este conhecimento é semelhante ao conceito de proveniência, que é usado em vários contextos. Por exemplo, o Vistrails capta a proveniência de uma visualização (como ela foi criada), bem como os dados manipulados [8], o que permite a reprodutibilidade e compreensão de todo o processo.

Tal como [6], usamos metadados (p.e. autor, posição, tempo) para atribuir responsabilidade por anotações individuais. Mais importante ainda, propomos uma estrutura baseada em árvore, inspirada em modelos *Issue-Based* que pretendem capturar a estrutura de argumentação, apoiar a tomada de decisão e permitir a compreensão posterior das decisões tomadas (*Issue-Based Information Systems* [4]). Na nossa discussão baseada em *Issues*, qualquer participante pode abrir uma nova questão, que pode incluir uma pergunta ou a identificação de um problema (e até uma sugestão de solução). Em seguida, os participantes podem responder a uma anotação existente, complementando-a com informação adicional (questões, detalhes ou tomar uma posição), ou adicionar argumentos positivos/negativos (Prós/Contras). A capacidade de responder a todas as anotações, da raiz até as folhas, cria a árvore mencionada acima. Isto tanto permite uma identificação clara do fluxo de argumentação, como suporta discussões paralelas sobre questões em aberto que possam surgir.

4 Anotações no Mundo Virtual

A nossa abordagem para anotações multimédia estruturadas foi construída em cima de um AVC multi-plataforma, que tem como objectivo suportar a Visualização 3D e discussão em reuniões colaborativas. Este sistema permite o uso de vários dispositivos como *Laptops*, *Walls* ou *Tablets*, o que permite interações individuais e/ou em equipa. Por exemplo, é possível interagir com a árvore de discussão num smartphone, enquanto se apresenta a anotação na Wall, para todos os utilizadores na sala (Figura 2-b).

Revisões de design de projectos exigem flexibilidade

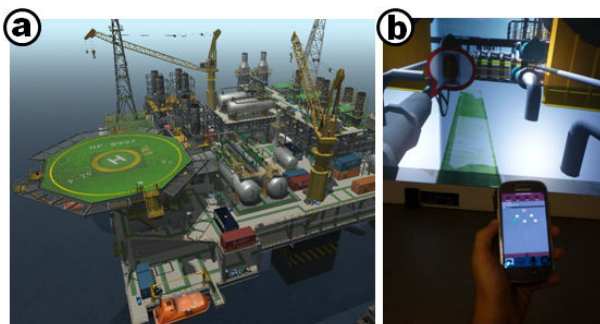


Figura 2. (a) Plataforma de petróleo utilizada. (b) Exploração da árvore no Smartphone, enquanto a Wall mostra o modelo.

quando as equipas estão distribuídas geograficamente. A nossa abordagem tenta superar esta dificuldade, principalmente em reuniões assíncronas, fornecendo uma estrutura flexível para suportar a discussão e tomada de decisão.

Os diferentes tipos de anotações são criados no mundo virtual através de um menu na parte inferior. (Figura 1-a). As anotações são retratadas no mundo virtual através de balões de banda desenhada, para metaforizar uma discussão. Para evitar a confusão no mundo virtual em tópicos populares, o número de balões desenhados é o logaritmo de base 2 de todas as anotações. Isto indica a atividade relativa (nº de anotações) da discussão de um problema, mas liberta o mundo virtual de confusão adicional.

Confiámos em elementos gráficos para retratar o estado atual das anotações e as relações entre elas. Por exemplo, a cor da margem de uma anotação no mundo virtual indica se esse problema já foi resolvido (verde) ou não (vermelho). Além disso, a imagem do perfil do autor da questão é apresentada no centro do balão de fala (Figura 1-a). Um clique/toque no balão exhibe toda a discussão em torno da anotação e permite participar na discussão.

Após seleccionar uma anotação no mundo virtual, o seu menu é apresentado no lado esquerdo do ecrã (Figura 1-b,c). É possível seleccionar uma das anotações e reproduzi-la (ou fazer *pause*) no botão na parte inferior. Todas as anotações são apresentadas nas mesmas condições em que foram criadas (mesmas posições e orientação da câmara).

Cada participante pode responder a uma anotação específica, adicionando um Pró, um Contra ou informação adicional (p.e. uma pergunta) e seleccionando o tipo de anotação (Figura 1-c). Após terminar, um novo nó é criado com uma imagem representativa do seu tipo e a cor da ligação indica o tipo de resposta (Pro a verde; Contra a vermelho, Mais Informação a amarelo). Esta informação é útil para fornecer uma visão geral do fluxo da discussão sem reproduzir todas as anotações. Para evitar sobrecarregar esta vista, a árvore é re-arranjada para ser centrada na anotação seleccionada. Por exemplo, a Figura 1-b mostra um nó colapsado (+) à esquerda, indicando maior atividade em resposta a essa anotação. Para finalizar, utilizadores com a permissão adequada podem declarar uma questão como resolvida (ou re-abri-la se necessário).

4.1 Cenário

Exemplificamos a nossa abordagem através de um cenário real, de modo a ilustrar todas as funcionalidades desenvolvidas. O cenário inclui uma revisão de projeto envolvendo engenheiros distribuídos geograficamente. Estes podem criar e reproduzir anotações multimédia, sobre as questões consideradas cruciais para a construção do modelo e/ou o modelo virtual (CAD) da plataforma (Figura 2). A Figura 3 retrata as anotações e o fluxo de discussão de uma discussão real numa reunião de projecto:

1. João pergunta se a válvula não deveriam ter um mecanismo que indicasse se está aberta ou fechada.
2. Rita, a Gestora do Projecto responde que este modelo não funciona dessa forma, e que esta é a válvula que usam

normalmente. Ela sugere que mantenham esta válvula.

3. Marco adiciona um Pró, mencionando que os trabalhadores estão habituados e sabem trabalhar com o modelo.
4. Rita adiciona que têm várias unidades deste modelo no armazém, outra vantagem para a sua utilização.
5. João insiste que a indicação é importante para trabalhadores novos que podem não ter esse conhecimento.
6. Rita diz que está incluído no treino e não é um problema.
7. João concorda em utilizar essa válvula.
8. Rita vê que a válvula está numa posição errada, e usa os movimentos de câmara para o mostrar. Ela pede ao designer para o corrigir o mais rápido possível.
9. Quando o problema é resolvido, Rita marca-o como tal.

5 Discussão e Conclusões

Revisão de design de projectos em AVCs requer uma estrutura flexível para apoiar a discussão. As anotações podem facilitar a comunicação ao colocar marcas diretamente no objeto de interesse. No entanto, as anotações convencionais não têm a flexibilidade e expressividade para visualizar e apoiar as discussões *multi-threaded* e argumentos complexos. Recorremos a diferentes opções multimédia para maximizar a expressividade. É possível criar e reproduzir anotações de áudio e esboços, onde as condições de visualização são replicadas para mostrar a perspectiva dos autores. Ainda assim, é com anotações de movimentos de câmara (posição e orientação) que os utilizadores podem explicar e demonstrar melhor os seus pontos de vista. Como visto no nosso cenário, pode-se manipular a câmara enquanto se explica uma anomalia que parece inexistente, mas que se torna visível de outra perspectiva.

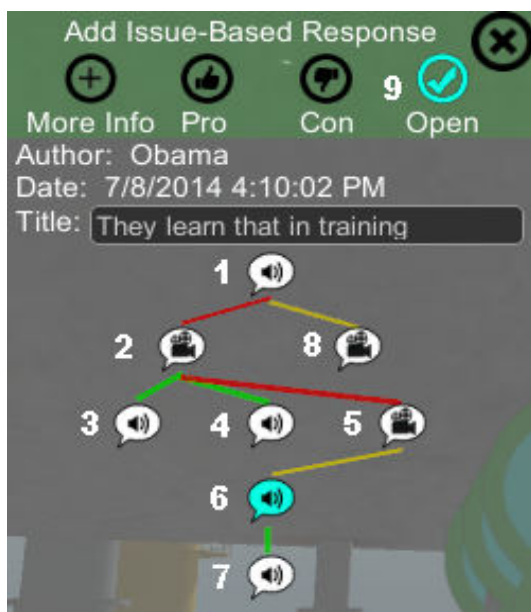


Figura 3. Árvore de discussão, baseada num cenário real. Numeração indica a ordem.

As anotações multimédia suportam argumentos completos e expressivos, mas não suportam a discussão por si só. A discussão do nosso cenário não seria suportada por anotações isoladas. A estrutura *Issue-Based* permite que os utilizadores respondam a anotações específicas com argumentos bem identificados. Tal estrutura é essencial para apoiar a tomada de decisão e para revelar a proveniência das decisões de design, o que não seria possível com um conjunto de post-its.

Embora este artigo apresente um modelo virtual numa fase de *design* avançada, cenários em fases mais precoces também podem ser explorados. No entanto, são necessários esforços adicionais para suportar o destaque de partes do modelo e para apresentar as mudanças que ocorrem num modelo virtual (CAD) 3D ao longo do tempo. Esta abordagem também pode ser estendida para suportar as actividades de formação e manutenção de forma natural.

6 Agradecimentos

Este trabalho foi parcialmente financiado pela FCT através dos projetos CEDAR PTDC/EIA-EIA/116070/2009, TECTON-3D PTDC/EEI-SII/3154/2012 e Pest-OE/EEI/LA0021/2013. Daniel Medeiros agradece à Capes pela bolsa recebida (ref 9040/13-7)

Referências

- [1] HUANG, G. Q. Web-based support for collaborative product design review. *Computers in Industry* (2002).
- [2] JOTA, R., AND ET AL. Immiview: a multi-user solution for design review in real-time. *Journal of Real-Time Image Processing* (2010).
- [3] JUNG, T., AND ET AL. Annotating and sketching on 3d web models. In *Proceedings of IUI* (2002), ACM.
- [4] KUNZ, W., AND RITTEL, H. W. *Issues as elements of information systems*, vol. 131. 1970.
- [5] LANGLOTZ, T., AND ET AL. Audio stickies: visually-guided spatial audio annotations on a mobile augmented reality platform. In *Proceedings of OZCHI* (2013), ACM.
- [6] LENNE, D., AND ET AL. Supporting design with 3d-annotations in a collaborative virtual environment. *Research in Engineering Design* (2009).
- [7] SANTOS, I. H., AND ET AL. A collaborative virtual reality oil & gas workflow. *International Journal of Virtual Reality* (2012).
- [8] SILVA, C. T., AND ET AL. Provenance for visualizations: Reproducibility and beyond. *Computing in Science & Engineering* (2007).
- [9] SIMMHAN, Y. L., AND ET AL. A survey of data provenance in e-science. *ACM Sigmod Record* (2005).
- [10] UVA, A. E., AND ET AL. Distributed design review using tangible augmented technical drawings. *Computer-Aided Design* (2010).

Eery Proxemics: Proximidade à Distância usando Múltiplas Superfícies

Maurício Sousa Daniel Mendes Alfredo Ferreira
João Madeiras Pereira Joaquim A. Jorge
INESC-ID/IST/Universidade Técnica de Lisboa

{antonio.sousa, danielmendes, alfredo.ferreira}@ist.utl.pt
{jap, jaj}@inesc-id.pt

Resumo

Com os avanços das ferramentas de video-conferência e de software colaborativo, as reuniões virtuais tornaram-se cada vez mais comuns, uma vez que permitem poupanças em tempo e recursos. No entanto, a sensação de presença ainda é diminuta. Na verdade, os participantes remotos queixam-se de ter uma presença reduzida, enquanto que os participantes locais têm dificuldades para perceber as atividades das pessoas remotas. Neste artigo, apresentamos o conceito de Eery Proxemics, uma extensão dos Proxemics que visa trazer a sintaxe das interações proxémicas para reuniões virtuais e aumentar a consciência das atividades e situação dos participantes remotos. O nosso trabalho centra-se em reuniões virtuais facilitadas por várias superfícies interactivas, que vão desde écrans de grandes dimensões (Walls), tablets e smartphones. Portanto, o nosso objetivo é aumentar o conhecimento mútuo de participantes em locais diferentes, que não se conseguem ver uns aos outros, através de um espaço virtual comum. Nós chamamos este espaço virtual Eery Space. Através dele, é possível tornar a área de interação proxémica visível para que os participantes distantes consigam usar as interações proxémicas naturais e estabelecer intercâmbios de informação. Foi realizada uma avaliação preliminar com pessoas fora do nosso grupo de investigação, onde os resultados indicam que a nossa abordagem é eficaz na melhoria do conhecimento mútuo entre os participantes e suficiente para iniciar as relações proxémicas, independentemente da localização física.

Palavras-Chave

Realidade Virtual, Realidade Aumentada, Colaboração, Interação Proxémica

1. INTRODUÇÃO

Quando as pessoas se reúnem, comunicam de várias maneiras além da forma verbal. Hall [Hall 66] observou que o espaço e a distância entre pessoas (*Proxemics*) tem impacto na comunicação interpessoal. Embora isso tenha sido explorado para ajudar a criação de conteúdo digital colaborativamente [Marquardt 12b], hoje em dia é cada vez mais comum as equipas de trabalho estarem geograficamente espalhadas pelo globo. Com apertados orçamentos de viagens e horários restritos, os membros da equipa contam com as reuniões virtuais. Estas reuniões conseguem reunir convenientemente pessoas de vários e diferentes locais. De facto, através de tecnologia adequada, torna-se possível ver os outros, bem como ouvi-los, o que significa que se torna mais fácil comunicar verbalmente, e até mesmo não-verbalmente, à distância. As novas soluções de videoconferência e telepresença suportam ambos os ambientes desktop comuns e as mais recentes tecnologias móveis, como smartphones e tablets. Exemplos notáveis incluem o Skype e o FaceTime. No entanto, apesar dos avanços tecnológicos, as pessoas remotas muitas vezes sentem-se negligenciadas devido à sua presença li-



Figura 1. Visão do Eery Space: uma pessoa remota controla a Wall, dois participantes separados geograficamente colaboram e uma quarta pessoa olha para eles através de uma janela virtual.

mitada [Neyfakh 14]. Além disso, embora a comunicação verbal e visual possa ser fácil em reuniões virtuais, outros modos de comunicação, ou seja, *proxemics*, ainda podem

ser explorados.

Neste trabalho, apresentamos *Remote Proxemics* como uma forma de interagir proxemicamente com pessoas remotas. Para este fim, nós exploramos o espaço à frente de dois ou mais ecrãs de grandes dimensões (Walls) em locais diferentes, onde as pessoas locais e remotas podem-se encontrar, partilhar recursos e envolverem-se em tarefas colaborativas, como ilustrado na Figura 1. Propomos técnicas que permitem que as pessoas interajam como se estivessem no mesmo espaço físico, bem como abordagens para melhorar a sensação mútua de presença. Finalmente, apresentamos a avaliação preliminar da nossa abordagem.

2. TRABALHO RELACIONADO

Os ambientes virtuais partilhados [Raskar 98] oferecem uma experiência diferente das "cabeças falantes", uma vez que as pessoas podem explorar uma localização remota. Os sistemas mais adequados para a colaboração são os imersivos, ou através de Walls, ou mesmo, através de ecrãs estilo CAVE. Estes sistemas fornecem o tamanho necessário para todas as pessoas numa reunião conseguirem ver os outros e utilizarem o espaço físico para o trabalho colaborativo. Por exemplo, [Cohen 14] apresentaram um sistema de vídeo-conferência com uma cena visual compartilhada para promover um jogo cooperativo com crianças. Os autores mostraram que a metáfora do espelho melhora a sensação de proximidade. Seguindo uma metáfora diferente, [Beck 13] apresentaram um sistema imersivo de telepresença que permite a grupos distribuídos de pessoas se encontrarem num mundo virtual 3D. Os participantes encontram-se frente-a-frente e exploram o modelo 3D. Enquanto que a maioria das interfaces de utilizador comuns exigem que seja a pessoa a despoletar uma ação, como um toque num botão, alguns sistemas têm a capacidade de reagir à presença do utilizador. Para isto, é importante detectar a presença e analisar as relações espaciais entre as pessoas. [Hall 66] declara que as relações espaciais podem evidenciar informação sobre a intenção das pessoas interagirem entre si. [Laga 09] sugere que o conceito de espaço privado pode ser utilizado como indicativo de comunicação não verbal e define um modelo matemático para identificar este espaço.

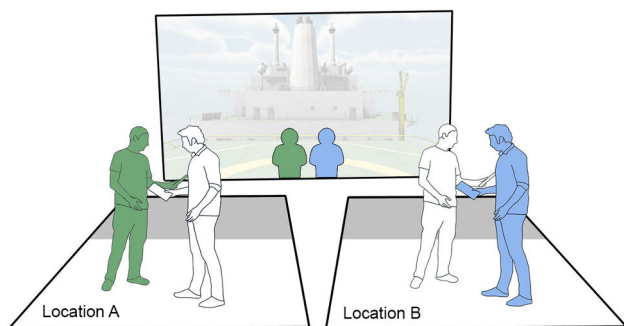


Figura 2. Duas pessoas em locais diferentes interagindo no Eery Space. As Walls mostram o mesmo em ambos os locais.

Mais recentemente, [Marquardt 12b, Marquardt 12a] propõem o uso de interações proxémicas para mediar pessoas, dispositivos e objectos não digitais. Os autores demonstram que através da análise da distância e orientação, as aplicações podem alterar os dados no ecrã ou reagir às pessoas para activar eventos implicitamente.

3. EERY SPACE

De forma a explorar as interações entre pessoas fisicamente separadas, criamos um espaço virtual comum, de forma a ultrapassar a distância que as separa. A este espaço chamamos de Eery Space, onde pessoas munidas de um dispositivo móvel conseguem se encontrar, colaborar e partilhar recursos à frente de uma Wall. Em vez de colocar os utilizadores em frente uns dos outros, como é típico nas soluções comerciais e outros trabalhos [Benko 12, Beck 13], colocamos ambas as pessoas remotas e as locais lado-a-lado, de forma similar ao descrito em [Cohen 14]. Consideramos as posições das pessoas ao longo da Wall e as suas distâncias até ela. Ao contrário da prática comum das interações com utilizadores remotos utilizando a metáfora do espelho, incutimos nas pessoas a sensação de que as pessoas remotas estão à volta das locais e presentes no mesmo espaço partilhado. Assim, o Eery Space cria e reforça o modelo da reunião partilhada onde as interações podem acontecer. Mais, todos os ecrãs Wall mostram a mesma perspectiva de forma a que o foco partilhado da reunião seja plausível.

4. DESENHO DA INTERAÇÃO

Ao colocar as pessoas no mesmo espaço virtual comum, apesar de estarem geograficamente distantes, novas formas de interação se tornam possíveis. Essas novas interações levam em conta o espaço pessoal de cada participante. O lugar de cada pessoa deve ser contabilizado, como se os participantes locais e remotos estivessem na mesma sala. Ao contrário dos sistemas convencionais, que se focam no contato visual, nós concentramos-nos em interações proxémicas.

4.1. Remote Proxemics

Criamos os *remote proxemics* para ser possível identificar as interações naturais que ocorrem entre as pessoas co-localizadas e torná-las disponíveis às pessoas que não estão fisicamente na mesma sala. Trabalhos anteriores indicaram que as pessoas respondem socialmente e naturalmente a elementos de mídia [Reeves 96]. Assim, permitimos que as pessoas remotas interajam através de proxies virtuais apropriados, fazendo tanto o espaço como as ações mutuamente visíveis. Dentro do Eery Space, quando uma pessoa entra no espaço pessoal de outra (um metro da sua posição), estas podem começar a interagir no que chamamos de *Bolha de Interação*. Esta bolha abrange duas ou mais pessoas, locais ou remotas. Quando localizados na mesma bolha, as pessoas podem participar em atividades colaborativas. No nosso protótipo, os participantes podem criar anotações conjuntas e têm a capacidade de ver o conteúdo dos outros em tempo real.

4.2. Moderador

O moderador é a pessoa que tem a autoridade especial de controlar a visualização comum nas Walls, espelhando ações feitas no seu dispositivo móvel. Esta autoridade é concedida a quem se aproxima da Wall, dentro do espaço do moderador (como demonstrado na Figura 2), aproveitando as interações próximas pessoa-dispositivo. O moderador atual abandona o seu papel ao sair deste espaço. Se isso acontecer e outra pessoa está neste espaço, então, essa passa a moderador. Caso contrário, o papel do moderador fica aberto para quem o quiser reclamar depois.

5. VISIBILIDADE E CONSCIÊNCIA

Permanecer conscientes dos outros é algo que tomamos como certo na vida quotidiana, mas manter essa consciência tem-se revelado difícil em sistemas distribuídos em tempo real [Gutwin 02]. Ao tentar manter as pessoas conscientes da presença de outras, um problema importante é como fornecer essa informação de forma não intrusiva, mas eficaz. Seguindo as orientações de colaboração propostas por [Erickson 00], utilizamos as técnicas descritas abaixo para aumentar a visibilidade e notoriedade das outras pessoas, ou seja, dos participantes remotos, seja através da Wall ou dos dispositivos móveis.

Sombras na Wall Cada pessoa tem uma sombra correspondente na Wall, que se distingue por um nome e uma cor única, como demonstrado na Figura 3, de forma similar a [Apperley 03]. O tamanho da sombra reflete a distância da pessoa à Wall para dar uma noção da relação espacial entre as pessoas e a superfície interativa. Além disso, cada pessoa tem uma aura colorida em torno da sua sombra. Quando duas ou mais pessoas têm a mesma cor da aura, encontram-se na mesma bolha e podem iniciar tarefas colaborativas.

Janelas Virtuais fornecem uma representação direta da posição e orientação das outras pessoas. Estas representam uma vista do mundo virtual, de forma similar a [Basu 12]. Combinando a posição das pessoas e a orientação dos seus dispositivos móveis, calculamos uma perspectiva indivi-

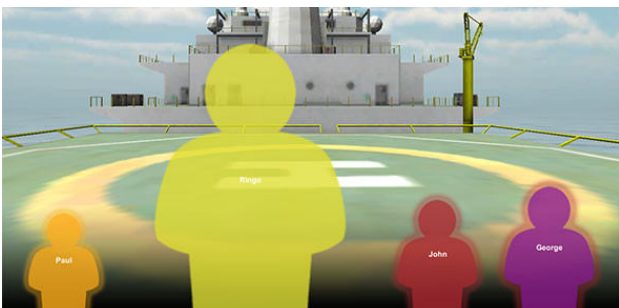


Figura 3. Sombras dos participantes na Wall. A maior representa o moderador. As duas pessoas na direita com auras vermelhas estão na mesma bolha. Quanto maior for a sombra mais próxima a pessoa está da Wall.



Figura 4. A Janela Virtual oferece uma visão pessoal para o mundo virtual, mostrando avatares com a posição e orientação de acordo com a Wall, do ponto de vista do proprietário do dispositivo. Neste caso, dois participantes, um local e um remoto.

dual. A Janela Virtual mostra ambas as pessoas locais e remotas (Figura 4), representadas por avatares dentro do mundo virtual.

Mapa da Bolha Sempre que um participante inclina o seu dispositivo para uma posição horizontal, uma vista superior parcial do espaço Eery é mostrada, como demonstrado na Figura 5. No centro, o dono do dispositivo é representado por um grande círculo branco. Outros participantes que estão perto o suficiente para estar na mesma bolha também são retratados por grandes círculos, pintados com a sua cor única. Participantes fora da bolha são representados fora do écran. Recorrendo a uma abordagem similar a [Gustafson 08], colocamos esses círculos (menores do que as pessoas na mesma bolha) na borda, indicando a sua direção de acordo com a sua posição no Eery Space.

Espaço Íntimo O Eery Space foi concebido, tendo em mente o respeito pelo espaço pessoal de cada participante. Todos os participantes têm o seu próprio espaço assegurado, mesmo quando não estão na mesma sala física que os outros. Para prevenir participantes de invadir o espaço de outros, oferecemos *feedback* háptico, vibrando os seus dispositivos móveis sempre que isto acontece.

6. AVALIAÇÃO PRELIMINAR

Para avaliar se as nossas técnicas fornecem *feedback* suficiente para que as pessoas possam interagir remotamente, realizámos uma pequena experiência. Construímos o nosso sistema usando um rastreador baseado em múltiplas câmeras Microsoft Kinect, que é capaz de identificar seis pessoas, lida com oclusões e resolve a posição de cada um. Utilizamos Unity3D para desenvolver um sistema distribuído multi-cliente para exploração de um ambiente virtual 3D, com suporte para múltiplos écrans, tablets e

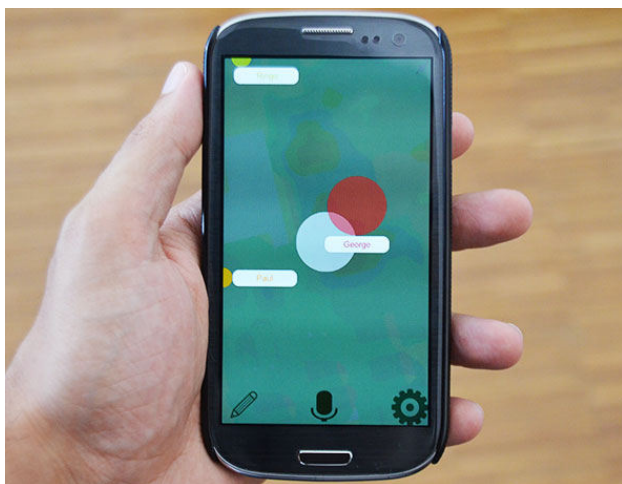


Figura 5. Mapa da Bolha. O círculo branco no centro representa o dono do dispositivo. O vermelho representa um participante na mesma bolha. Os pequenos círculos na borda são pessoas fora da bolha.

smartphones. Para esta experiência, dois participantes foram colocados em salas diferentes equipadas com um display Wall. O nosso cenário foi construído à volta de tarefas de desenho e revisão de modelos 3D. Ambos os participantes foram convidados a assumir o controlo Wall, em turnos, para navegar até um ponto no modelo e, em seguida, abordar uma pessoa local e outra remota para iniciar uma anotação colaborativa. Através de um questionário qualitativo usando uma escala Likert de 6 valores (1 - muito difícil, 6 - muito fácil), seis participantes indicaram que foi fácil (≥ 5) executar as tarefas. Ainda, descobrimos que não existem diferenças significativas em localizar a pessoa remota no *Eery Space* em relação a uma pessoa local (≥ 5 em ambos os casos).

7. CONCLUSÕES E TRABALHO FUTURO

Nas reuniões virtuais, os participantes remotos muitas vezes se sentem negligenciados devido à sua presença limitada. De forma a mitigar isto, o nosso *Eery Space* traz interações proxémicas a pessoas geograficamente distantes. Nós exploramos ambas as interações proxémicas pessoa-pessoa e pessoa-dispositivo, e técnicas desenvolvidas para fornecer a consciência adequada das ações dos participantes, tanto remotos como locais. Os resultados da avaliação preliminar com utilizadores sugerem que a nossa solução é capaz de fornecer os meios necessários para que as pessoas se envolvam em actividades de cooperação com base na sua localização dentro do espaço virtual comum e em relação às Walls. Para trabalho futuro, gostaríamos de verificar se avatars, que representem mais fielmente as pessoas, conseguem aumentar a sensação de presença dos participantes remotos e locais. Além disso, temos a intenção de aplicar o nosso conceito a diferentes campos, tais como edição e manipulação colaborativa de modelos de engenharia ou visualização de dados médicos.

8. AGRADECIMENTOS

O trabalho apresentado neste artigo foi parcialmente financiado pela Fundação para a Ciência e a Tecnologia (FCT) através dos projetos CEDAR (PTDC/EIA-EIA/116070/2009), TECTON-3D (PTDC/EEI-SII/3154/2012), Pest-OE/EEI/LA0021/2013 e através da bolsa SFRH/BD/91372/2012.

Referências

- [Apperley 03] Mark Apperley, Laurie McLeod, Masood Masoodian, Lance Paine, Malcolm Phillips, Bill Rogers, e Kirsten Thomson. Use of video shadow for small group interaction awareness on a large interactive display surface. 2003.
- [Basu 12] Aryabrata Basu, Andrew Raij, e Kyle Johnsen. Ubiquitous collaborative activity virtual environments. Em *Proc. of CSCW '12*, 2012.
- [Beck 13] S. Beck, A. Kunert, A. Kulik, e B. Froehlich. Immersive group-to-group telepresence. *Visualization and Computer Graphics, IEEE Transactions on*, 2013.
- [Benko 12] Hrvoje Benko, Ricardo Jota, e Andrew Wilson. Miragetable: freehand interaction on a projected augmented reality tabletop. Em *Proc. of CHI '12*, 2012.
- [Cohen 14] Maayan Cohen, Kody R. Dillman, Haley MacLeod, Seth Hunter, e Anthony Tang. Onespace: Shared visual scenes for active freeplay. Em *Proc. of CHI '14*, 2014.
- [Erickson 00] Thomas Erickson e Wendy A. Kellogg. Social translucence: An approach to designing systems that support social processes. *ACM TOCHI*, 2000.
- [Gustafson 08] Sean Gustafson, Patrick Baudisch, Carl Gutwin, e Pourang Irani. Wedge: Clutter-free visualization of off-screen locations. Em *Proc. of CHI '08*, 2008.
- [Gutwin 02] Carl Gutwin e Saul Greenberg. A descriptive framework of workspace awareness for real-time groupware. *CSCW*, 2002.
- [Hall 66] Edward T. Hall. *The Hidden Dimension*. Doubleday, 1966.
- [Laga 09] Hamid Laga e Toshitaka Amaoka. Modeling the spatial behavior of virtual agents in groups for non-verbal communication in virtual worlds. Em *Proc. of IUCS '09*, 2009.
- [Marquardt 12a] Nicolai Marquardt, Till Ballendat, Sebastian Boring, Saul Greenberg, e Ken Hinckley. Gradual engagement: Facilitating information exchange between digital devices as a function of proximity. Em *Proc. of ITS '12*, 2012.
- [Marquardt 12b] Nicolai Marquardt, Ken Hinckley, e Saul Greenberg. Cross-device interaction via micro-mobility and formations. Em *Proc. of UIST '12*, 2012.
- [Neyfakh 14] Leon Neyfakh. My day as a robot, May 2014. Online: <http://www.bostonglobe.com/ideas/2014/05/10/day-robot/6UAMgmUFn0mZhoMS8vy0GK/story.html>, accessed 14-June-2014.
- [Raskar 98] Ramesh Raskar, Greg Welch, Matt Cutts, Adam Lake, Lev Stesin, e Henry Fuchs. The office of the future: A unified approach to image-based modeling and spatially immersive displays. Em *Proc. of SIGGRAPH '98*, 1998.
- [Reeves 96] Byron Reeves e Clifford Nass. *The Media Equation: How People Treat Computers, Television, and New Media Like Real People and Places*. 1996.

Índice de Autores

- Alexandre, Isabel 3, 19
Almeida, Ricardo 3
Amorim, Nuno 167
- Baptista, Filipe 19
Brito, Luisa 3
- Calado, Jorge 19
Campos, Carlos 89
Capela, Marco 3
Capote, Hugo 159
Carmo, Maria Beatriz 11, 33, 137
Carreira, Micael 49
Carrilho, Luís 145
Cláudio, Ana Paula 11, 33, 137
Coelho, António 89
Coelho, Catarina 121
Costa, Vasco 71
Cuevas, José Luis 159
- D'Alpuim, Jorge 113
Dias, Carla 145
Dias, Miguel Sales 113, 151
Dias, Paulo 121
Duarte, Rui 79
- Eloy, Sara 151
- Fernandes, António Ramires 63, 129
Ferreira, Alfredo 41, 179
Ferreira, Maria Teresa 121
Fonseca, Fernando 97
- Gama, Sandra 25, 57
Gaspar, Augusta 11, 33
Gaspar, Filipe 151
Gomes, Abel 79, 103
Gomes, Inês Laureano 33
Gomes, João Tiago 137
Gonçalves, Daniel 25, 49, 57
Guerreiro, João 175
- Henriques, Diogo 41
- Jordão, Vilma 25
Jorge, Joaquim 71, 97, 175, 179
- Leitão, Miguel 89
Liu, André 129
Lopes, Daniel 97
Lopes, Diogo 63
Lopes, Mariana 151
Lopes, Pedro 19
Lopes, Pedro Faria 3
- Margallo, Francisco 159
Medeiros, Daniel 175
Mendes, Daniel 41, 97, 179
Mendonça, Nuno 151
Miguel, Ricardo 151
Moreira, Pedro Miguel 145
Morgado, José Francisco 79, 103
- Neves, Luís 121
- Oliveira, João 159
- Pagador, José Blas 159
Pascoal, Pedro 41
Pereira, João Madeiras 71, 179
- Raposo, Alberto 175
Rocha, Jorge Gustavo 167
Rodrigues, Rui Sérgio 103
Romero, Luis 145
- Santos, Beatriz Sousa 121
Santos, Hélder 121
Santos, Ismael 175
Silva, Inês 19
Silva, Jorge 151
Simão, Daniel 97
Sousa, Mauricio 179
Sousa, Pedro 3
- Teixeira, Renato 11
Ting, Karine 49
Trancoso, Isabel 41