

INSTITUTO POLITÉCNICO DE VIANA DO CASTELO

24 A 26 OUTUBRO 2012

20.º Encontro Português de Computação Gráfica

Livro de Atas

20^º12
Encontro Português
Computação Gráfica



INSTITUTO POLITÉCNICO
DE VIANA DO CASTELO



GPCG
Grupo Português de
Computação Gráfica

Atas
20^o Encontro Português de Computação Gráfica

24-26 de outubro, 2012

Viana do Castelo, Portugal

Atas do 20º Encontro Português de Computação Gráfica

Escola Superior de Tecnologia e Gestão
Instituto Politécnico de Viana do Castelo
Viana do Castelo, Portugal
24-26 de outubro, 2012

Pedro Branco, Pedro Miguel Moreira, Rui Rodrigues e Luís Romero (editores)

publicado por
Instituto Politécnico de Viana do Castelo
Praça General Barbosa
4900-347 Viana do Castelo

ISBN: 978-989-97491-5-3
ebook ISBN: 978-989-97491-4-6
CD-ROM ISBN: 978-989-97491-2-2

Depósito Legal: 350546/12

impresso por MediaDesign 2
em outubro de 2012

design da capa: Ana Filomena Curralo

edição apoiada por:



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

Prefácio

Entre 24 e 26 de outubro de 2012 realiza-se na Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Viana do Castelo a 20ª edição do Encontro Português de Computação Gráfica. Este evento tem reunido investigadores, docentes e profissionais da Computação Gráfica e áreas afins desde a sua génese, permitindo a divulgação de trabalhos realizados ou em curso, a troca de experiências e a discussão de temas relacionados com a Computação Gráfica.

Para a edição deste ano, contámos com variadas contribuições da comunidade científica, divididas em três grandes categorias: os artigos longos, os artigos curtos e os posters. Os artigos longos apresentam resultados de trabalho significativo que representam uma contribuição na área científica respetiva. Os artigos curtos permitem uma discussão mais focada, apresentando ainda assim contribuições significativas. Finalmente, os posters apresentam resultados de última hora, novas ideias ou trabalhos de qualidade que estão em curso.

As diversas contribuições foram revistas e avaliadas por membros da Comissão Científica, e neste processo foram selecionados 12 artigos longos, 6 artigos curtos e 6 posters, cobrindo variadas áreas de estudo: modelação e reconstrução, visão por computador, realidade aumentada, jogos e entretenimento, interação, síntese e visualização, entre outras. Essas contribuições são reproduzidas neste volume, organizadas pelas três categorias descritas.

O evento propriamente dito conta com três workshops no seu primeiro dia, com temáticas que passam pela impressão 3D, síntese de imagem e simulação e modelação de comportamentos de corpos vivos. Nos restantes dias decorrem as sessões técnicas, a outorga do “Prémio Professor José Luís Encarnação 2012” promovido pelo Grupo Português de Computação Gráfica, e duas palestras proferidas por ilustres investigadores convidados: José Luís Encarnação e Verónica Orvalho.

Gostaríamos de agradecer aos autores e oradores convidados pelas suas contribuições, aos membros da Comissão Científica pelo trabalho desenvolvido, à Comissão Organizadora pela realização da conferência no Instituto Politécnico de Viana do Castelo, e ainda, ao Grupo Português de Computação Gráfica pelo apoio institucional, e pelo esforço contínuo em manter este Encontro ao longo dos anos. Fica também um agradecimento especial a todos aqueles que, não estando diretamente envolvidos na organização, foram essenciais para o sucesso da mesma, através de diferentes formas de apoio. A Comissão Organizadora agradece ainda a todas as entidades empresariais e institucionais que a apoiaram na concretização do evento.

A eles, e a todos os que contribuíram ao longo destes anos para a história da Computação Gráfica em Portugal e deste Encontro, os nossos parabéns e um especial agradecimento pelo vosso trabalho.

Cordiais saudações,

Pedro Branco, UM
Rui Rodrigues, FE/UP
(Copresidentes da Comissão Científica do 20º EPCG)

Pedro Miguel Moreira, IPVC
Luís Romero, IPVC
(Copresidentes da Comissão Organizadora do 20º EPCG)

Viana do Castelo, outubro de 2012.

Comissão Científica

Co-presidentes

Rui Rodrigues, FE/UP

Pedro Branco, UM

Membros

Abel Gomes, UBI

Adérito Marcos, UAberta

Adriano Lopes, FCT/UNL

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 Ramires Fernandes, UM

Armando J. Pinho, IEETA/UA

Beatriz Sousa Santos, DET/UA

Caldas Pinto, IST/UTL

Daniel Gonçalves, INESC-ID

Elisabeth Simão Carvalho, UAtlântica

Fernando Birra, FCT/UNL

Fernando Nunes Ferreira, FE/UP

Fernando Pereira, IST/IT

Francisco Morgado, IPViseu

Frutuoso da Silva, UBI

Hélder Araújo, ISR/UC

João Cunha, LNEC

João Madeiras Pereira, IST/UTL

João Paulo Moura, UTAD

Joaquim Jorge, IST/UTL

Joaquim Madeira, DET/UA

José Creissac Campos, UM

Luís Gonzaga Magalhães, UTAD

Luís Marcelino, FCT/UNL

Luís Paulo Santos, UM

Mário Rui Gomes, IST/UTL

Manuel João Ferreira, UM

Manuel João Fonseca, IST/UTL

Manuel Próspero dos Santos, UNL

Maria Beatriz Carmo, UL

Mauro Figueiredo, UAlg

Maximino Bessa, UTAD

Miguel Leitão, ISEP/IPP

Miguel Velhote Correia, FE/UP

Nuno Correia, FCT/UNL

Óscar Mealha, DeCA/UA

Paulo Dias, IEETA/UA

Pedro Castro, IPVC

Pedro Faria Lopes, ISCTE/ADETTI

Rui Prada, INESC-ID/IST

Teresa Chambel, FC/UL

Teresa Romão, FCT/UNL

Vasco Branco, UA/INESC

Verónica Orvalho, FC/UP

Comissão Organizadora

Co-presidentes

Luís Romero, IPVC

Pedro Miguel Moreira, IPVC

Secretariado

Maria João Lousa, GCI-IPVC

Membros

Abel Barbosa Soares, IPVC

Alexandre Ulisses Silva, IPVC

Ana Filomena Curralo, IPVC

Carlos Sérgio Laranjeira, IPVC

João Nunes, IPVC

Paula Alexandra Rego, IPVC

Pedro Miguel Faria, IPVC

Organização, Parcerias e Apoios

Coorganização

Instituto Politécnico de Viana do Castelo

<http://www.ipvc.pt>

Grupo Português de Computação Gráfica

<http://www.gpcg.pt>

Parceria

AEESTG - Associação de Estudantes da Escola Superior de Tecnologia e Gestão do IPVC

<http://ae.estg.ipvc.pt>

Apoios Empresariais

CA - Crédito Agrícola

<http://www.creditoagricola.pt>

CPC.iS

<http://www.cpcis.pt>

Orbcom - IT & Consulting Services

<http://www.orbcom.pt>

OVNITUR - Viagens, Turismo e Rent-a-Car

<http://www.ovnitur.pt>

Apoios Institucionais

CMVC - Câmara Municipal de Viana do Castelo

<http://www.cm-viana-castelo.pt>

Porto e Norte - Turismo do Porto e Norte de Portugal

<http://www.portoenorte.pt>

Índice

Prefácio	iii
Comissão Científica	v
Comissão Organizadora	vii
Organização, Apoios e Parcerias	viii
Artigos longos	1
SketchTester: Analysis and Evaluation of Calligraphic Gesture Recognizers <i>Abílio Costa, João P. J. Pereira</i>	3
Interface Caligráfica de Escrita no Ar <i>Pedro Leitão, António Castro, João P. J. Pereira</i>	11
Sistema Integrado de Geração Automática de Conteúdo para Videojogos de Plataformas <i>Fausto Mourato, Fernando Birra, Manuel Próspero dos Santos</i>	19
Identificação de Objetos em Imagens Tomográficas através de GPGPUs <i>Bruno Preto, Fernando Birra, Pedro Medeiros</i>	27
Visualização em Tempo Real de um Modelo Esparso de Mistura Paramétrica para Síntese da BTF <i>Nuno Silva, Luís Paulo Santos</i>	35
Realidade Virtual Aplicada ao Tratamento da Ansiedade Social <i>Tânia Pinheiro, Ana Paula Cláudio, Maria Beatriz Carmo, Francisco Esteves</i>	43
Modelação Procedimental de Ambientes Rodoviários para Simulação de Condução <i>Carlos Campos, João Miguel Leitão, António Coelho</i>	51
CaS: Collection-aware Segmentation <i>Raquel Costa, Manuel J. Fonseca, Alfredo Ferreira</i>	59
Ray Tracing of Large Models on a Multi-Projection Display <i>Vasco Costa, João M. Pereira, Joaquim A. Jorge</i>	67
Adaptação de Simbologia em Realidade Aumentada <i>Maria Beatriz Carmo, Ana Paula Cláudio, António Ferreira, Ana Paula Afonso, Edgar Montez, Raúl Simplício</i>	73
Interaction with Large Displays in a Public Space Using the Kinect Sensor <i>Igor Duarte Cardoso, Paulo Dias, Beatriz Sousa Santos</i>	81
ERAS – An Ontology-Based Tool for Expeditious Reconstruction of Virtual Cultural Heritage Sites <i>Telmo Adão, Luís Magalhães, Maximino Bessa, António Coelho, Augusto Sousa, Nuno Rodrigues, Alexandrino Gonçalves, Roberto Rodrigues, Francisco Pereira, João Moura, Luís Paulo Reis</i>	89
Artigos curtos	97
Adaptação de Cor para Dicromatas na Visualização de Imagens <i>Madalena Ribeiro, Abel Gomes</i>	99
Inserção Dinâmica de Objectos Virtuais no Contexto de Fotografias Tiradas por Utilizadores <i>Rui Nóbrega, Nuno Correia</i>	103
Exploração Interactiva dos Hábitos de Audição de Música <i>Ricardo Dias, Manuel J. Fonseca, Daniel Gonçalves</i>	107
AUV Planning, An Application for AUV Mission Planning <i>António Coelho, Ricardo Gonçalves, Rui Rodrigues,</i>	111

COGNITO - Captura, Reconhecimento e Visualização de Atividades Manuais Complexas <i>Gustavo Mações, Hugo Domingues, Luís Almeida, Luís Paulo Santos</i>	115
Figuras de Ação como Interfaces Tangíveis para a Criação de Filmes de Animação <i>Pedro Ribeiro, Ido Iurgel</i>	119
Posters	123
Human Motion Analysis Tools <i>João Ferreira Nunes, Pedro Miguel Moreira, João Manuel R. S. Tavares</i>	125
Outdoor Sports Performance Analysis Enhancement and Multimedia Documentation <i>João Manuel Brisson Lopes</i>	127
Utilização de Objetos Reais em Aplicações de Realidade Aumentada para Dispositivos Móveis: Aplicação de Advertainment <i>Nuno Cardoso, Fábio Rocha, Pedro Miguel Moreira</i>	129
Realidade Aumentada para Caça ao Tesouro em Larga Escala <i>João Fonseca, Gilberto Felgueiras, Pedro Miguel Moreira</i>	131
Extração de Informação de Texto Formal para a Modelação Expedita de Edifícios Monumentais <i>Ricardo Baptista, Roberto Rodrigues, António Coelho, Luís Paulo Reis, Augusto Sousa, Luís Magalhães</i>	133
An Intuitive Navigation System: Augmented Reality on the Car Windshield <i>Waldir Pimenta</i>	135
Índice de autores	137

Artigos longos

SketchTester: Analysis and Evaluation of Calligraphic Gesture Recognizers

Abílio Costa

Dep. de Eng^a Informática, ISEP - IPP
R. Dr. António Bernardino de Almeida, 431, Porto
amfca1t@gmail.com

João P. Pereira

Dep. de Eng^a Informática / GECAD, ISEP - IPP
R. Dr. António Bernardino de Almeida, 431, Porto
jjp@isep.ipp.pt

Resumo

As interfaces caligráficas apresentam uma forma natural para utilizadores interagirem com aplicações. Dado que o núcleo de uma interface caligráfica é o seu reconhecedor, existe a necessidade de avaliação de diversos reconhecedores antes de optar pela utilização de um. Neste artigo apresentamos uma avaliação de três reconhecedores caligráficos: o reconhecedor de Rubine, o reconhecedor de \$1 e o reconhecedor CALI. A avaliação foi realizada com base em amostras caligráficas reais, desenhadas por 32 participantes, com um conjunto de símbolos selecionados para utilização num trabalho futuro. Para além disto, discutimos também alguns aperfeiçoamentos realizados à implementação dos reconhecedores e que ajudaram a obter taxas de reconhecimento superiores. No final, o CALI obteve a melhor taxa de reconhecimento com 94% de sucesso, seguido do reconhecedor de \$1 com 87% e finalmente pelo reconhecedor de Rubine com 79%.

Abstract

Sketch-based interfaces can provide a natural way for users to interact with applications. Since the core of a sketch-based interface is the gesture recognizer, there is a need to correctly evaluate various recognizers before choosing one. In this paper we present an evaluation of three gesture recognizers: Rubine's recognizer, CALI and the \$1 Recognizer. The evaluation was done using real gesture samples drawn by 32 subjects, with a gesture set arranged for use in a future work. We also discuss some improvements to the recognizers' implementation that helped achieving higher recognition rates. In the end, CALI had the best recognition rate with 94% accuracy, followed by \$1 Recognizer with 87% and finally by Rubine's recognizer with 79%.

Keywords

Gesture recognition, Calligraphic interfaces, Rubine, CALI, \$1 Recognizer.

1. INTRODUCTION

Using pen and paper to draw or sketch something in order to express an idea is very common and also very natural for us. By using this concept in user interfaces one can make the interaction process more natural and spontaneous.

In the future, we aim to develop a programming library to aid in the creation of applications for two-dimensional physics simulations in which the user interacts directly with the scene using a "pen and paper" style interaction. Thus, instead of selecting from a menu which objects compose the scene to be simulated, the user can simply draw the objects directly in the scene. We hope that developing a library that integrates a calligraphic interface and a physics simulation engine will provide a boost for developers to create new applications around this concept, be they for educational purposes, like an application used for teaching physics to students using an interactive whiteboard, or for entertainment purposes, such as a physics-based game where the user draws parts of the scene in order to reach a goal, in the same genre as Crayon Physics Deluxe [Purho09]. These are only two examples of a wide range of possibilities.

The library will support three gestures to draw primitives and other three to define relations between primitives. The first three gestures are used to draw rectangles, triangles and circles, which can be created by drawing these symbols directly. To establish relations between primitives the user can draw a zigzag to connect two primitives with a spring, a cross to pin a primitive over another and a small circle to connect one primitive over another with a rotation axis. Since both the circle primitive and the rotation axis relation use the same gesture¹, we only have in fact five gestures to recognize, presented in Figure 1. Given that the cross is the only gesture that cannot be drawn with only one stroke, we opted to replace it with an alpha, which is an intuitive single-stroke representation of a cross. We chose to use only single-stroke gestures because besides meeting the needs of our library it makes the interaction simpler, since using gestures formed with multi-strokes will force the user to specifically signalize when a gesture is completely drawn or, if

¹ The identification of whether the system should recognize a circle primitive or a rotation axis relation is done by analyzing the size of the gesture and whether or not it is drawn over two existing gestures.

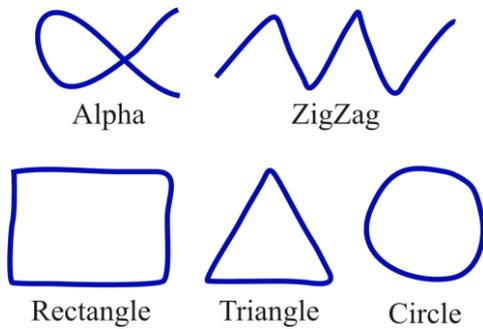


Figure 1 – Set of gestures used in our work

using a timer approach, to draw all the gesture's parts within a specific time and wait for the recognition to happen, which may lead to user frustration.

Given the importance of having good gesture recognition, since the user must feel the interaction to be as natural and unrestrictive as drawing with a pen and a paper, we conducted an evaluation of various gesture recognizers in order to select the one that best fits our needs. In this evaluation we have done two sessions to collect samples of the five gestures drawn by various subjects, in order to put the recognizers to test with a wide range of data. This paper describes that evaluation in detail, along with various considerations to achieve higher recognition rates.

In the next section we present an overview of the related work done in the gesture recognition field. This is followed by a description of the application we developed to test gesture recognizers and the implementation of these recognizers. We then present how the evaluation of the gesture recognizers was conducted and discuss its results. Finally we propose potential future developments of this work and present our conclusions.

2. RELATED WORK

Given the potential of automatic sketch recognition, a lot of work has been done in order to develop recognizers capable of dealing with the intrinsic ambiguity of hand-drawn sketches. Since there is a great variety of sketch recognition algorithms, it is only natural that there's also diversity in their characteristics. For example, some recognizers only work with single-stroke sketches, while others are oriented towards multi-stroke sketches. Also, whether or not the recognizer can identify sketches independently of their orientation, scale, and drawing order can greatly affect its usefulness in some domains. Another important characteristic is if the recognizer can be trained with new gestures, meaning that it can be easily expanded, or if its gestures are hardcoded, which makes it difficult to change its gesture set to fit a new domain.

Rubine's recognizer [Rubine91], a trainable gesture recognizer, classifies each gesture using a linear classifier algorithm with a set of distinct features. Rubine specifies 11 static geometric features, such as sin/cosine of the initial angle of the gesture, distance between the first and last points, total gesture's length, among others. Rubine also defined two dynamic features: the maximum speed of the gesture and its duration. The recognizer is very flexible since features can be easily added or removed to make the recognizer fit the application needs. For exam-

ple, [Plimmer07] shows how to improve the recognition and make it independent of gesture's size by removing features that involve absolute sizes and adding new ones that use ratios instead. For the training process, Rubine's recognizer calculates the features of the training templates of each gesture class² and computes the weights of those classes based on their features. As pointed by the author, the recognizer requires about 15 training templates per gesture class to be effective, which can make the training process a time consuming task. To recognize a gesture, the algorithm makes a linear combination of the gesture's features and the weights of each class. The class which maximizes that combination is selected as the one that the gesture belongs to. The major limitations of Rubine's recognizer are its sensibility to the drawing direction, scale and orientation and being unable to identify multi-stroke sketches. Pereira et al. [Pereira04] made some modifications to Rubine's recognizer in order to make the algorithm accept multi-stroke sketches, but only when drawn with a constant set of strokes as pointed out by [Stahovich11]. The authors also present a way to make the algorithm insensitive to drawing direction, by doing the recognition twice: first with the original sketch and then with an inverted sketch.

CALI [Fonseca02] is an easy to use multi-stroke recognizer that uses Fuzzy Logic and geometric features to classify gestures independently of their size or orientation. Instead of an individual algorithm, CALI is a complete library that can be easily built into an application. CALI separates gestures into two types: shapes and commands. Shapes can be drawn (and recognized) using solid, dashed and bold³ lines, while commands are only recognized with solid lines. The recognizer defines a set of geometric rules or features to identify each gesture, like its thinness, aspect ratio, and many others. For example, a Line shape is characterized by being "very thin". In addition to the global geometric features, some gesture classes are also characterized by local features such as the sub-gestures that compose the gesture or whether it has intersections. For example, the Cross command is identified by having two intersecting Line shapes. When an input gesture enters the recognition process its features are computed and checked against each defined gesture rules, using fuzzy sets to find the degree of membership to each rule and therefore to each class. Since CALI is a non-trainable recognizer, adding new gestures is not an easy task, involving hand-coding and analysis of which features characterize and distinguish the gesture. To solve this limitation the authors also present a trainable recognizer and compare three training algorithms: K-Nearest Neighbors, Inductive Decision Tree and Naïve Bayes, the latest being the one with highest training efficiency. Nevertheless, the trainable recognizer has a lowest recognition rate and requires numerous training templates for each gesture class.

² A gesture class represents a unique gesture, but can be made from multiple representations of that gesture, i.e. multiple templates.

³ Bold lines are made from multiple overlapping solid strokes.

In [Wobbrock07], Wobbrock et al. present the \$1 Recognizer which aims to be easy to understand and quick to implement. It is insensitive to scale and orientation of sketches, but is sensitive to the drawing direction. One major advantage of \$1 Recognizer is the simplicity to add support for new gestures, requiring only one training template per gesture class in most cases. The algorithm has basically four major steps, the first three being applied to both the training templates and the input gesture (the one that is to be recognized), and the fourth step only to the input gesture. The first step is to resample the point path, using simple linear interpolation, so that every gesture (including the training templates) has the same number of points. This enables a direct point-to-point comparison between input gesture and training templates, independently of drawing size and speed. The second step is to rotate the gesture to an orientation that is optimal for matching and thus reduce the recognition time later. This rotation is made based on the angle between the centroid of the gesture and the gesture's first point. The third step is to scale the gesture non-uniformly to a square and translate its centroid to the origin (0,0). The fourth and last step, which is only applied to input gestures, is where the actual recognition happens. The input gesture is compared to each training template to find the average distance between corresponding points and, based on that distance, a score is calculated. The training template with the biggest score is the one that, according to the recognizer, matches the input gesture. Templates with lower score can be used to deal with ambiguity, serving as alternative matches. When the algorithm is computing the average path-distance between an input gesture and a training template, the input gesture is rotated using the Golden Section Search algorithm to find the angle in which that distance is minimized. The authors also explain how to make the recognizer sensitive to scale or orientation, for some or all gesture templates.

In order to solve some of the limitations of the \$1 Recognizer, such as not being able to recognizing multi-stroke gestures, sensitiveness to the drawing direction, and problems recognizing uni-dimensional gestures such as lines, Anthony et al. extended it and created the \$N Recognizer [Anthony10]. The algorithm starts by computing all the possible combinations of stroke orders and directions for each multi-stroke gesture serving as training template and creates a single-stroke gesture for each combination, by connecting the individual strokes with the order and direction of that combination. These single-stroke gestures are used for comparison with the input gesture, using the same process as the \$1 Recognizer, since multi-stroke input gestures are also transformed into single-stroke gestures by connecting their individual strokes by the order they were drawn. The transformations used in \$1 Recognizer, such as point resampling, rotation to find the optimal orientation, and translation of the centroid to the origin are also applied by \$N to every combined single-stroke gesture. Despite the improvements over the \$1 Recognizer, \$N has problems recognizing gestures made with more strokes than defined in the training templates. Also, it is not well suited to recognize "messy" gestures

like a scratch-out, commonly used for erasing-like actions.

Lee et al. [Lee07] present a trainable graph-based recognizer that is insensitive to orientation, scale and drawing direction and is able to recognize multi-stroke gestures. The recognizer uses statistical models to define symbols, which makes it deal with the small variations associated with hand-drawn gestures naturally. Each gesture is represented by an attributed relational graph, in which nodes depict the type of primitive (line or arc) and its relative length⁴. The edges of the graph represent the geometric relationships between primitives, characterized by the number of intersections, the intersection angle and the intersection location. Gestures are segmented into individual primitives using a technique based on the drawing speed [Stahovich04], meaning that errors in the segmentation process will propagate to the recognition process. When an input gesture arrives, the recognizer compares it to each trained gesture class and computes a dissimilarity score based on six error metrics, each one with a different weight on the resulting score. This dissimilarity score is then converted to a similarity score which is used to identify the gesture class that classifies the input gesture. Since the same gesture can be drawn with varying number of primitives and drawing orders, comparing input gestures and training templates is not straightforward and presents a graph matching problem. To solve this, the authors evaluate and propose five approximate matching techniques. For the training process, an average attributed relational graph is created for each gesture class, by averaging the graphs of multiple training templates. One limitation of this approach is that all training templates of a gesture class must be drawn with a consistent drawing order or consistent orientation.

Vatavu et al. [Vatavu09] present a trainable recognizer that uses elastic deformation energies to classify single-stroke gestures. The recognizer is naturally insensitive to gesture scale and orientation, since the same gesture has similar curvature functions independently of the drawing orientation or size, but is sensitive to drawing direction and starting point within the gesture. To classify a gesture, the recognizer computes its curvature function, based on trajectory analysis, and calculates the alignment cost to each gesture class to find the one that minimizes that cost. Computing the curvature function for each class is done by averaging the functions of multiple training templates for that class.

In [Sezgin05], the authors present a multi-stroke sketch recognizer, based on Hidden Markov Models (HMM), that is capable of recognizing individual sketches in complex scenes even if the scene is not yet completed, i.e. while it is being drawn, and without the need to pre-segment⁵ it. On the other hand it can only recognize sketches in their trained orientations, thus being sensitive

⁴ The length of the primitive in relation to the total gesture's length.

⁵ Pre-segmenting a scene means isolating individual sketches or gestures in the scene.

to orientation. The algorithm first creates discrete observation sequences of the scene by identifying various geometric primitives such as sloped lines, horizontal/vertical lines, polylines, among others, with the aid of the Early Sketch Processing Toolkit [Sezgin01]. Then, for each trained HMM, it computes the likelihood for various subsections of the scene given that HMM. Using these likelihoods, it builds a graph in which the shortest path gives the most likely segmentation of the scene, that is, the individual sketches in the scene. Finally, it classifies each segment (the individual sketch) by finding the HMM that maximizes the probability of generating that segment. Since the recognition relies on the stroke order of the trained templates, it is not well suited for domains where the stroke ordering cannot be predicted. Also, because HMMs are suited for sequences, it cannot recognize single-stroke sketches, unless they are pre-segmented.

PaleoSketch [Paulson08] is a low-level non-trainable sketch recognizer for single-stroke primitives. When recognizing a gesture, it starts with a pre-recognition phase, where it removes consecutive duplicate points, cleans drawing noise from the beginning and the end of the gesture, and computes various graphs and values that characterize that gesture. After the pre-recognition process is done, the actual recognition is made. PaleoSketch uses multiple individual sub-recognizers, where each indicates whether or not the input gesture matches a given primitive. Since a gesture can be recognized by more than one sub-recognizer and since each sub-recognizer only returns whether the stroke matches or not, without any “matching score”, the results are passed to a hierarchy function for sorting. This function uses a corner finding algorithm to find the minimum number of lines to correctly describe the input gesture and compares that to the minimum number of lines defined for each sub-recognizer’s primitive, ordering the results accordingly. PaleoSketch is insensitive to orientation, scale and drawing order, but can only recognize low-level primitives. For more complex gestures (such as rectangles or triangles), one needs to add a higher-level recognizer on top of PaleoSketch.

The need to compare and evaluate the performance of various sketch recognizers is not something new. In [Schmieder09] a toolkit to automatically evaluate recognition algorithms is presented. In addition to sketch data collection and labeling, the toolkit allows the integration of multiple trainable and non-trainable recognizers, which can be tested simultaneously. After testing, the toolkit outputs the results in the form of a Microsoft Excel file or screenshots of the gestures that have been misclassified. As a proof of concept, the authors also present and discuss experimental results of the evaluation of six gesture recognizers: CALI, Microsoft Ink Analyser, \$1 Recognizer, Rubine’s recognizer with the extended features used in InkKit, PaleoSketch, and a recognizer using Dynamic Time Warping techniques.

3. IMPLEMENTATION

In order to test the sketch recognition algorithms, we developed SketchTester, an application that enables us to rapidly prototype and incorporate recognition algorithms. With this application we are able to individually test each algorithm against drawn gestures and immediately see the recognition results (recognized gesture, recognition score, and other alternative matches with lower score). It also provides a graphical interface to add/remove training templates to/from each trainable recognizer, automatically saving them to hard-disk in a recognizer-specific file. As shown in Figure 2, to collect sketch samples from subjects SketchTester offers a window that specifies what gesture should be drawn and, after the gesture is submitted, shows the recognition result of each recognizer. The application keeps asking for random gestures until a predefined number of samples of each gesture is collected, point at which the recognition rates of each recognizer are presented. Each of the submitted sample gesture and corresponding results are saved to a file, so that we can analyze it later.

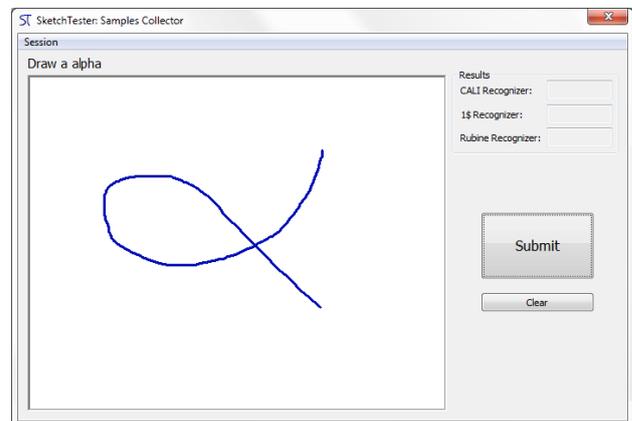


Figure 2 – Interface used to collect gesture samples from subjects

SketchTester also incorporates functionality to review collected gesture samples, save them as bitmap files, and extract data from them, such as total number of correctly recognized gestures. Another important feature of SketchTester is the possibility to reprocess multiple files of collected samples. This was particularly useful since we made some improvements to the recognizers after the collection of samples from subjects. By reprocessing those samples, their recognition results were updated according to the improved recognizers.

In SketchTester we implemented three popular recognizers: the \$1 Recognizer, Rubine’s recognizer, and CALI. The algorithms of the first two recognizers were implemented according to the descriptions given by their authors in [Wobbrock07] and [Rubine91], respectively. In the case of CALI, since it exists in the form of a library, it was only necessary to integrate it with SketchTester. Another important note is that we didn’t implement any rejection for gestures with low score, since we always want a result even if it is a low-scored match.

As regards to training templates, \$1 Recognizer was first trained with 2 templates for each gesture while Rubine’s

recognizer was trained with 15 templates for each gesture. Also, since \$1 and Rubine's recognizer are sensitive to the drawing direction, when adding a new training template the application automatically creates a copy of the template but with inverted drawing direction, adding it to the same gesture class as the original template in the case of \$1 or to a new class in the case of Rubine's recognizer

CALI was also subject to changes. First we disabled the recognition of gestures we don't need, such as the "copy" gesture, lines, arrows, and some others. Also, since we don't need to discriminate circles from ellipses, we always classify an input gesture as a circle whether it is recognized by CALI as circle or ellipse. The same logic applies to rectangles and diamonds, where they are both classified as rectangles. By removing unneeded gestures from the recognizer and grouping similar gestures we expect to increase the recognition success. Furthermore, we added support for the alpha gesture in CALI by hard-coding a new gesture class in which we defined two features that were selected based on empirical observations: the ratio between the area of the largest quadrilateral and the area of the convex hull (A_{lq}/A_{ch}), and the ratio between the perimeter of the largest quadrilateral and perimeter of the convex hull (P_{lq}/P_{ch}). The fuzzy sets associated with these features are presented in Figure 3. These two features alone were not enough, because most of the times CALI would classify an input alpha gesture as both alpha and zigzag⁶. To prevent alphas from being misrecognized as zigzags we defined that zigzags can't have any intersections, and that alphas must have an intersection situated at more than 10% away from the limits of the gesture, meaning that each of the alpha's tails must make more than 10% of the gesture.

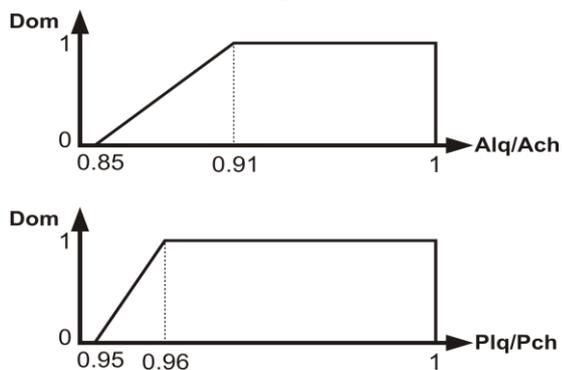


Figure 3 – Fuzzy sets defined for the alpha gesture

After analyzing the samples collected from subjects, we found that the implemented recognizers could be improved. In Rubine's recognizer we added three more classes (plus three "duplicates" with inverted drawing direction) per gesture to contemplate different orientations. Figure 4 exemplifies how the rectangle gesture would be represented in each of the four classes. Before the inclusion of new classes to represent various orientations, only the orientation represented in the top-left side of Figure 4 was present.

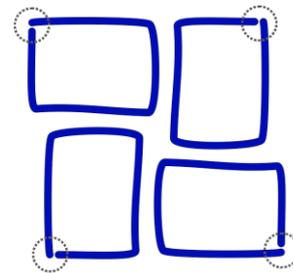


Figure 4 – Representation of four classes for the rectangle gesture contemplating different orientations; the circles mark start and end points

For the \$1 Recognizer, one possible improvement we found was adding training templates describing rectangles and triangles starting at the middle of an edge, since the first version only had templates starting at vertices. We also added training templates with different starting vertices for these two gestures in order to contemplate every drawing possibility, but without overdoing it, since the algorithm is insensitive to drawing orientation and so we don't need templates of the gesture starting in every vertex.

We have also improved CALI implementation, by setting the minimum size of each of the alpha's tails to 7% instead of 10%. Also, we defined that a zigzag can have intersections as long as the distance along the gesture between the two intersecting points is less than 13% of the total gesture's length. Finally, the distance along the gesture between the two intersecting points in an alpha must be more than 20%.

The rationale behind these improvements will be described in the next section of this paper.

4. EVALUATION

In our effort towards finding the recognizer which best fits our purpose and has the greatest recognition rate we collected 1550 gesture samples from 32 subjects.

4.1 Method

We conducted two sessions to collect gesture samples, each session with 16 subjects. Using the SketchTester application and following a guide document⁷, each subject was asked to draw 10 samples of every of gesture, for a total of 50 samples per subject which provided us with 1550 gesture samples⁸. Both sessions were conducted at our institution where 9 subjects used the institution's desktop computers with traditional mice. The remaining 23 subjects had personal laptops and about 25% of them used the laptop's built-in touchpads and the remaining subjects used conventional mice. All the subjects were MSc students.

Since our evaluation aims to test the recognition of single-stroke gestures with no drawing restrictions, we asked the subjects to freely draw the five gestures presented in Figure 1 with diversified sizes, orientations and shapes.

⁷ http://dei.isep.ipp.pt/~i060687/guiaao_recolha_caligrafica.pdf

⁸ 1550 and not 1600 because two subjects provided less than 50 samples each.

⁶ The zigzag gesture is originally called WavyLine in CALI.

4.2 Results

As previously described, we made some improvements to the recognizers and their training templates after analyzing the first recognition results of the collected gesture samples. Then, we reprocessed these samples with the improved recognizers. In this section we will present the recognition results obtained before and after the improvements and discuss how these improvements affect the recognition rates.

To make sure the collected samples fit the domain in which we intend to use the recognizers, we first cleaned them by removing samples that do not match the requested gesture (e.g., the subject was asked to draw a rectangle and drew a circle) and samples drawn incorrectly (e.g., gestures drawn with multiple overlapping lines or gestures that don't represent any of the five required gestures). As expected, cleaning the samples enhanced the recognition rates, especially with CALI whose improvement reached 11%. This enhancement is shown in Figure 5, with a side-by-side comparison of the recognition rates of the three algorithms before and after cleaning the samples, with CALI achieving the highest rates.

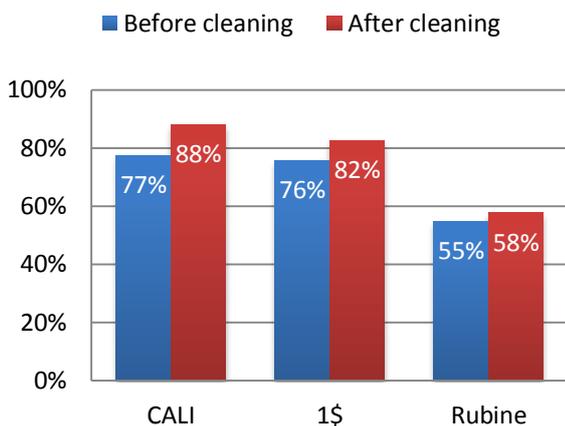


Figure 5 – Recognitions rates before and after cleaning the collected sample gestures

In order to understand why Rubine's recognizer obtained such low recognition rates we need to examine the individual recognition rates of each gesture. As presented in Figure 6, the recognition rates of the triangle and circle gestures are 26% and 27% respectively, which is extremely low and far from the remaining gestures' rates. After reviewing individual samples it was clear that what was affecting the recognition of these gestures was the diversification in drawing orientations. We also noticed that users have a tendency to draw rectangles starting with the top-left vertex, which explains why that particular gesture was not suffering much with the orientation sensitivity problem, since the training samples had that same orientation. Interestingly, the drawing orientation had a low impact on the recognition of the zigzag gesture, mainly because it is very distinctive from the remaining gestures in terms of features, and since we did not implement gesture rejection, it is recognized even with a low recognition score.

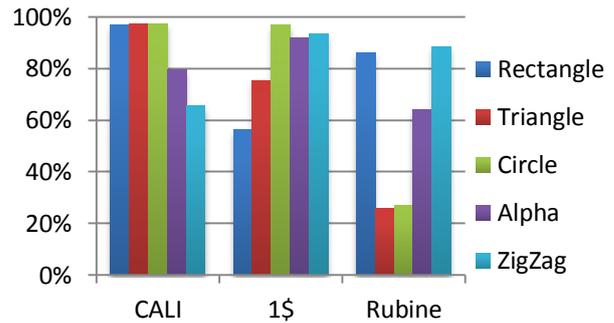


Figure 6 – Recognition rates of each gesture with each recognizer

To increase Rubine's recognizer recognition rates we needed to overcome the problem of rotation dependence. As described earlier, the solution was to add new classes representing each gesture in different orientations. As show in Figure 7, this greatly increased the recognition rates and, despite a slight decrease in the recognition of the rectangle gesture, the overall recognition rate with Rubine's recognizer was improved to 79%, against the previous 58%.

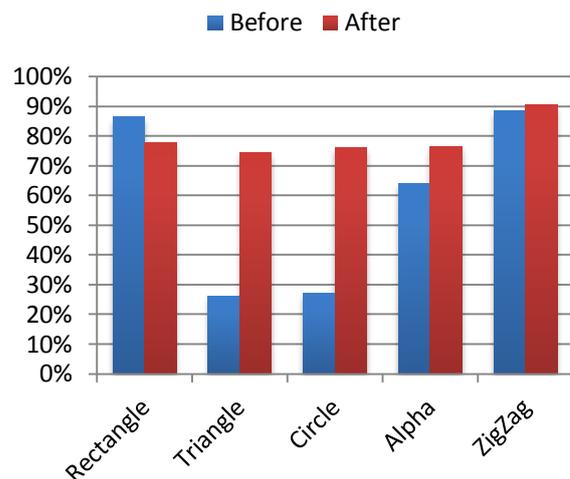


Figure 7 – Recognition rates for each gesture with Rubine's recognizer, before and after adding gesture classes representing multiple orientations

Regarding the \$1 Recognizer, the most problematic gestures were rectangles and triangles, with recognition rates of 56% and 75% respectively. We found that both gestures needed training templates with starting points at different vertices and also at the middle of edges. We also found that a training template representing right triangles was needed. After these additions to \$1 Recognizer's training templates the recognition rates of rectangles and triangles were improved and, despite a decrease in the recognition rates of circles and alphas, the global recognition rate of the algorithm was improved to 87% against the previous 82%. We then tried to improve the recognition of circles and alphas, since they were affected in these changes, but found no success. Figure 8 shows the recognition rates for each gesture before and after the addition of the new training templates.

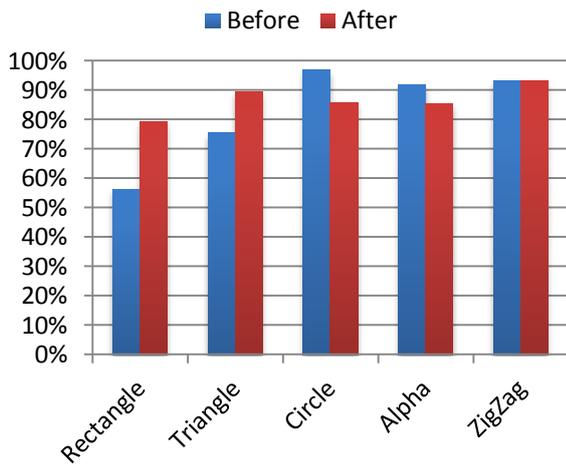


Figure 8 - Recognition rates for each gesture with \$1 Recognizer, before and after improving its training templates

Despite CALI already reaching a fairly high recognition rate of 88%, analyzing the results for individual gesture rates suggests that it could be further improved. As show in Figure 6, the alpha and zigzag gestures have low recognition rates when compared to the other gestures, which may be due to the changes we've made to the original CALI source code to include the alpha gesture, discussed previously. Further investigation showed that many alphas with short tails were simply not recognized. After detailed examination of the collected gestures we found that 7% was the optimal value as the minimum relative size of the gesture's tails. In relation to the zigzag gesture, we found that many subjects made small intersections when drawing zigzags and since we first specified that zigzags must not have intersections, many were recognized as alphas. After inspecting the collected samples, we specified that zigzags can have intersections but only when the two intersecting points are not more than 20% away from each other. We also updated the alpha gesture so that the distance between the two intersecting points must be more than 20% of the total gesture's length. As presented in Figure 9, these changes to CALI had a positive impact on the recognition of alphas and zigzags, without affecting the other gestures.

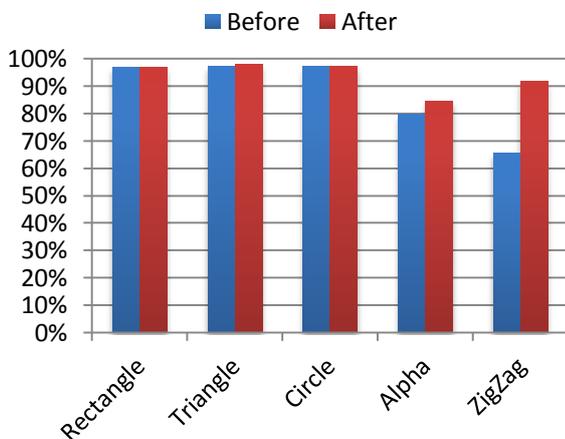


Figure 9 - Recognition rates for each gesture with CALI, before and after updating the recognition source code of the alpha and zigzag gestures

By analyzing the first recognition results of the samples collected from subjects, we were able to identify flaws in our implementation of the recognizers. As show in Figure 10, our attempts to correct these flaws where successful and all the recognizers reached higher recognition rates, with CALI achieving the best rates, followed by \$1 and then by Rubine's recognizer. Also, Figure 10 shows that even after improving the recognizers, the respective standard deviations are high, which is caused by a significant disparity in the individual recognition rates of each gesture. This suggests the possibility that the recognizers can still be improved. If we were able to improve the recognition of the gestures with lowest rates, the overall recognition rate would increase and the standard deviation would decrease, meaning that the recognizer would be recognizing all the gestures in a balanced way.

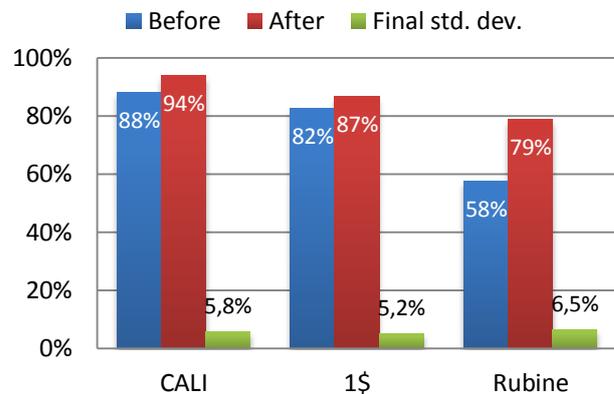


Figure 10 - Overall recognitions rates before and after improving the recognizers, along with the standard deviation of the final rates

In [Schmieder09], to show the potential of their automatized recognizer evaluation toolkit, the authors conducted an experiment in which they evaluate six recognizers, including the same three we evaluated, with three basic single-stroke gestures⁹: circle, rectangle and line. In their experiment, Rubine's recognizer had the best recognition with a 96% success rate, followed by \$1 Recognizer with 89% and CALI with 84%. While at first these results can seem to contradict our evaluation's results, with CALI and Rubine's recognizer inverting positions, they can be easily explained. While we used the original feature set in Rubine's algorithm, they implemented the extended feature set used in InkKit [Plimmer07], which explains why Rubine's recognizer achieved such high recognition rates. In respect to CALI, they consider circles and ellipses as independent gestures, unlike our evaluation where we don't need to differentiate these two gestures and consider both as one. In their results, there have been 94 circles misclassified as ellipses, in a total of 730 evaluated gestures. If we consider these 94 ellipses as being correctly classified, effectively merging circles and ellipses, the recognition rate for CALI increases to 96%. If we also merge rectangles and diamonds it rises to 98%. These

⁹ There's also a second experiment which we won't cover on this paper because it is done with Entity Relationship (ER) diagrams instead of basic gesture shapes.

rates are close to those obtained in our evaluation. Regarding \$1 Recognizer, both evaluations yield similar results. Finally, it is important to notice that since they used fewer gestures than us it is normal that they obtained higher recognition rates, as the misclassifications tend to increase with the number of gestures.

5. FUTURE WORK

In the future we could implement more recognizers in SketchTester, and even improve Rubine's features as described in [Plimmer07], in order to also evaluate them against the collected samples. The inclusion of other gestures could also be subject of study if the library is extended beyond five gestures.

An evaluation of the recognizers with gestures collected using touchscreens or interactive whiteboards would also be an interesting evaluation, since these are the kind of devices that most benefit from calligraphic interfaces.

Since the improvements to the recognizers were made and evaluated using the same set of gesture samples, it would be important to re-evaluate these improvements with new samples, in order to confirm that they are valid not only for our sample set but also to generic samples.

Finally, although the most relevant result of recognition is the gesture with the highest score, a study considering the first two or three high-score results could be relevant in cases where the application presents a list of alternative matches to solve ambiguity.

6. CONCLUSION

In this work we've collected sample gestures from various subjects and evaluated three popular gesture recognizers to find the one that best fits in the interaction layer of our physics simulation library. We've also presented some insights on how the implementations of these recognizers can be improved to yield better results. Also, despite the specificity of the tested data, our work can serve as a base to others exploring gesture recognizers.

To conclude, both CALI and \$1 are good candidates for our library since both achieved high recognition results. Also, we are confident that if the improved features described in [Plimmer07] were implemented in Rubine's recognizer, it could have results as good as the results of the other two recognizers. Nevertheless, since CALI archived higher rates it shall be selected to integrate our library.

7. ACKNOWLEDGMENTS

We wish to thank all the participants in the recognizer evaluation sessions. We are also grateful to Manuel J. Fonseca for providing us the CALI library source code.

8. REFERENCES

- [Anthony10] Lisa Anthony, Jacob O. Wobbrock. 2010. A lightweight multistroke recognizer for user interface prototypes. In *Proceedings of Graphics Interface 2010* (GI '10), 245-252.
- [Fonseca02] Manuel J. Fonseca, César Pimentel, and Joaquim A. Jorge. 2002. CALI: An online scribble recognizer for calligraphic interfaces. In *AAAI Spring Symposium on Sketch Understanding*, 51-58.
- [Lee07] WeeSan Lee, Levent Burak Kara, and Thomas F. Stahovich. 2007. An efficient graph-based recognizer for hand-drawn symbols. In *Computers & Graphics* 31, 554-567.
- [Paulson08] Brandon Paulson and Tracy Hammond. 2008. PaleoSketch: accurate primitive sketch recognition and beautification. In *Proceedings of the 13th international conference on Intelligent user interfaces* (IUI '08), 1-10.
- [Pereira04] Pereira, J. P., Branco, V. A., Jorge, J. A., Silva, N. F., Cardoso, T. D., and Ferreira, F. N. 2004. Cascading recognizers for ambiguous calligraphic interaction. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling*.
- [Plimmer07] Beryl Plimmer and Isaac Freeman. 2007. A toolkit approach to sketched diagram recognition. In *Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI...but not as we know it* (BCS-HCI '07), vol. 1.
- [Purho09] Petri Purho. 2009. Crayon Physics Deluxe. <<http://crayonphysics.com/>>
- [Rubine91] Rubine, Dean. 1991. Specifying gestures by example. In *Proceedings of the 18th annual conference on Computer graphics and interactive techniques* (SIGGRAPH '91), 329-337.
- [Schmieder09] Paul Schmieder, Beryl Plimmer, and Rachel Blagojevic. 2009. Automatic evaluation of sketch recognizers. In *Proceedings of the 6th Eurographics Symposium on Sketch-Based Interfaces and Modeling* (SBIM '09), 85-92.
- [Sezgin01] Tevfik Metin Sezgin, Thomas Stahovich, and Randall Davis. 2001. Sketch based interfaces: early processing for sketch understanding. In *Proceedings of the 2001 workshop on Perceptive user interfaces* (PUI '01), 1-8.
- [Sezgin05] Tevfik Metin Sezgin and Randall Davis. 2005. HMM-based efficient sketch recognition. In *Proceedings of the 10th international conference on Intelligent user interfaces* (IUI '05), 281-283.
- [Stahovich04] Thomas F. Stahovich. 2004. Segmentation of pen strokes using pen speed. In *Proceedings of 2004 AAAI fall symposium on making pen-based interaction intelligent and natural*, 152-158.
- [Stahovich11] Thomas F. Stahovich. 2011. Pen-based Interfaces for Engineering and Education. In *Sketch-based Interfaces and Modeling*, 119-152.
- [Vatavu09] Radu-Daniel Vatavu, Laurent Grisoni, and Stefan-Gheorghe Pentiu. 2009. Gesture Recognition Based on Elastic Deformation Energies. In *Gesture-Based Human-Computer Interaction and Simulation*, vol. 5085, 1-12.
- [Wobbrock07] Jacob O. Wobbrock, Andrew D. Wilson, and Yang Li. 2007. Gestures without libraries, toolkits or training: a \$1 recognizer for user interface prototypes. In *Proceedings of the 20th annual ACM symposium on User interface software and technology* (UIST '07), 159-168.

Interface Caligráfica de Escrita no Ar

Pedro Leitão

Dep. Eng. Informática, ISEP-Instituto
Politécnico do Porto

R. Dr. António Bernardino de Almeida, 431 Porto

pedromi-
guel_rs70@hotmail.com

António Castro

GILT/ISEP-Instituto Politécnico do
Porto

R. Dr. António Bernardino de Almeida, 431 Porto

avc@isep.ipp.pt

João J. Pereira

GECAD/ISEP-Instituto Politécnico
do Porto

R. Dr. António Bernardino de Almeida, 431 Porto

jjp@isep.ipp.pt

Resumo

Na actualidade, está a emergir um novo paradigma de interacção, designado por Natural User Interface (NUI), para reconhecimento de gestos produzidos com o corpo do utilizador. O dispositivo de interacção Microsoft Kinect, inicialmente concebido para controlo de videojogos, demonstra ser uma aposta viável para explorar outras áreas, como a do apoio ao processo de ensino e de aprendizagem para crianças do ensino básico. O protótipo desenvolvido visa definir um modo de interacção baseado no desenho de letras no ar, e realizar a interpretação dos símbolos desenhados, usando os reconhecedores de padrões KDA, SVM e \$N.

Abstract

A new interaction paradigm, named Natural User Interface (NUI), for the recognition of gestures produced with the user's body is emerging. The interaction device Microsoft Kinect, which was originally designed to control video games, proves to be a viable wager to explore other areas such as support to the teaching and learning process for children of elementary school. The developed prototype aims to define a mode of interaction for drawing letters in the air, and perform the interpretation of the symbols by means of the pattern recognizers KDA, SVM and \$N.

Keywords

Classificadores estatísticos, NUI, Kinect, Nuvens de pontos, Calibração, OpenNI, NITE, XNA, OSC, \$N.

1. INTRODUÇÃO

A interacção com equipamentos electrónicos usando teclado, rato ou outro controlador manual, tem vindo a ser bastante vulgarizada. O dispositivo Microsoft Kinect pretende colmatar as limitações humano-máquina existentes, através de uma interacção baseada em gestos realizados com o corpo. Este dispositivo, inicialmente desenvolvido para a utilização em videojogos, tem vindo a despertar interesse junto do público e a demonstrar diversas potencialidades. A aplicabilidade do Kinect não se limita unicamente aos videojogos, existindo um vasto leque de áreas nas quais pode ser utilizado. O âmbito deste projecto visa a aplicação do Kinect, juntamente com o reconhecimento de padrões, à área do ensino, integrando os algoritmos KDA, SVM e \$N de forma a interpretar e identificar as letras correspondentes aos símbolos desenhados pelo utilizador, auxiliando assim o desenvolvimento motor e apoiando o processo de aprendizagem.

Nos tempos que correm, com a globalização e o crescimento tecnológico acelerado, as pessoas têm vindo a aderir cada vez mais a equipamentos digitais, em detrimento

da escrita convencional (papel e lápis). Contudo, e apesar de a sociedade se encontrar cada vez mais informatizada e dependente da tecnologia, a escrita convencional continua a ter um papel indispensável no quotidiano das pessoas, sobrepondo-se, em determinadas situações, à escrita digital. A escrita de textos, equações ou o desenho de esquemas nas salas de aulas constituem um bom exemplo da sua indispensabilidade. Estas tarefas tornam-se mais funcionais quando aliadas a uma interface tradicional.

Ao longo do tempo têm surgido diversas aplicações com este tipo de interfaces, refere Tomas Sylverberg [Sylverberg et al., 07]. Este autor propõe, como forma de apoio a militares, o uso de dispositivos móveis que lhes permitam desenhar símbolos NATO¹, essenciais na orientação dos soldados no terreno. O uso de dispositivos permitiria a partilha de informações do terreno, reportando a informação desenhada aos restantes elementos dispersos no campo.

¹ Conjunto de símbolos militares padronizados, para marcações em mapas pela NATO.

Nas secções 2 e 3 serão apresentados respectivamente alguns dos principais dispositivos NUI existentes no mercado e as correspondentes bibliotecas de desenvolvimento. A secção 4 aborda os algoritmos de reconhecimento de padrões analisados no âmbito deste projecto. O trabalho realizado e a respectiva avaliação são descritos nas secções 5 e 6, respectivamente. Por último, na secção 7, procede-se à conclusão do artigo e traçam-se as perspectivas de trabalho futuro.

2. HARDWARE

Com a mudança do paradigma de interacção, várias marcas de consolas de videojogos estão a lançar para o mercado novos produtos inovadores na área da interacção natural.

2.1 Wii

Nos anos 80, a Nintendo tentou, pela primeira vez, mudar o modo de interacção dos seus produtos com a criação da *Nintendo PowerPad*. Contudo, apenas anos mais tarde, em 2006, com o lançamento da *Wii*, o público aderiu consideravelmente a este modo de jogo. A *Wii* vem equipada com um sensor e um dispositivo semelhante a um comando de televisão, designado por *Wii Remote*, que utiliza um microcontrolador² ADXL330. Este controla três acelerómetros em simultâneo, um para cada eixo, de forma a obter a velocidade e a posição da mão do jogador [Tanaka et al., 12].

2.2 PlayStation Move

A multinacional japonesa *Sony* lançou para o mercado, no ano de 2004, o dispositivo de captura visual baseado em gestos denominado *Eye Toy*. Este conseguiu alcançar alguma notoriedade, despertando o interesse do público por este novo modo de interacção, apesar da baixa capacidade de captura de informação bidimensional de que dispunha. A *PlayStation Move* inclui o *Move Eye*, que consiste numa câmara RGB com uma resolução máxima de 640 por 480 píxeis, a funcionar a uma frequência de 60 a 120 *frames* por segundo (FPS). Contém também um conjunto de microfones e um dispositivo, designado por *Motion Controller*, idêntico a um bastão com uma esfera luminosa. O reconhecimento dos movimentos realiza-se com base na obtenção das coordenadas tridimensionais da esfera luminosa pela câmara RGB [Tanaka et al., 12].

2.3 Microsoft Kinect

O Kinect surgiu de uma conferência realizada em Junho de 2009 (Figura 1), sendo que o seu lançamento para o mercado aconteceu apenas em Novembro de 2010 [Tscherrig, 11]. O projecto que esteve na origem do Microsoft Kinect designou-se por 'Projecto Natal' e teve como base a tecnologia PrimeSense. Adoptava um sistema de luz padronizada e o microcontrolador PS1080 SoC, concebendo assim um dispositivo de controlo de videojogos para a consola Xbox 360 da Microsoft. Na sua constituição podemos encontrar um conjunto de microfones para aplicações que envolvam o reconheci-

mento de comandos por voz, uma câmara RGB com uma frequência de amostragem de 30 FPS e uma resolução de 640x480 píxeis. Além desta, inclui uma câmara de infravermelhos (IR) com uma resolução de 640x480 píxeis e frequência de amostragem de 30 FPS, usada para medir a profundidade dos elementos inseridos no cenário através da luz estruturada projectada sob eles e posteriormente reflectida para a câmara IR.



Figura 1 - Constituição do Microsoft Kinect [Mehlmann et al., 11]

3. SOFTWARE

A comunicação e acesso às funções do dispositivo Kinect processam-se através de um *driver*³ de comunicação entre o dispositivo e o sistema operativo. Os vários *drivers* de comunicação existentes, garantem o acesso a diferentes funcionalidades e suporte para diferentes sistemas operativos.

3.1 Microsoft Kinect SDK

A 21 de Maio de 2012 foi disponibilizada a versão comercial 1.5 da SDK Kinect. Esta versão contém um conjunto de ferramentas pensado para programadores e empresas, permitindo-lhes desenvolverem aplicações para este dispositivo. Esta versão comercializável veio permitir tanto a empresas como ao público em geral usufruírem de um conjunto de ferramentas para Windows direccionadas para o dispositivo Kinect [Pinto, 12b].

3.2 OpenNI

A plataforma OpenNI é uma plataforma para dispositivos NUI, que surgiu em Dezembro de 2010, fruto da associação de empresas sem fins lucrativos como a Willow Garage, desenvolvedora de hardware e software livre para aplicações pessoais robóticas [Willow Garage, 11], a Side-Kick, líder no desenvolvimento de jogos e publicação especializada de videojogos baseados em NUI [SideKick, 2011], e a fabricante de equipamentos informáticos ASUS, que comercializa o dispositivo NUI WaviXtion, baseado na tecnologia PrimeSense [Rogers, 11]. Através de um esforço conjunto estabeleceram uma plataforma padrão para equipamentos NUI, assegurando a compatibilidade, interoperabilidade e a comunicação entre estes equipamentos [OpenNI, 11a].

A sua infra-estrutura é segmentada por camadas (Figura 2), nas quais disponibiliza a interface para módulos de

² Consiste num computador num *chip*, contendo um processador, memória e periféricos de entrada/saída.

³ Pequeno programa que faz a comunicação entre o sistema operativo e o hardware.

dispositivos físicos e módulos para *middleware* de comunicação, associando-os à plataforma e acabando assim com problemas de incompatibilidades entre *middleware* e sensores ou vice-versa. Com isto, passa a ser possível desenvolver aplicações e portá-las para funcionar com outro *middleware* ou dispositivo sem que sejam necessárias modificações suplementares.

As camadas que constituem a infra-estrutura OpenNI são as que a seguir se discriminam [OpenNI, 11b]:

1. A camada aplicação funciona com base nos dados fornecidos da camada inferior e permite interagir com aplicações NUI alto nível;
2. Camada que representa as interfaces OpenNI, e os vários *middlewares* que podem ser associados à plataforma. Tem como objectivo processar os dados adquiridos pelos dispositivos e traduzi-los para serem interpretados pela camada mais acima;
3. Camada responsável por fazer a associação à plataforma dos diferentes dispositivos de captura de informação visual e áudio.

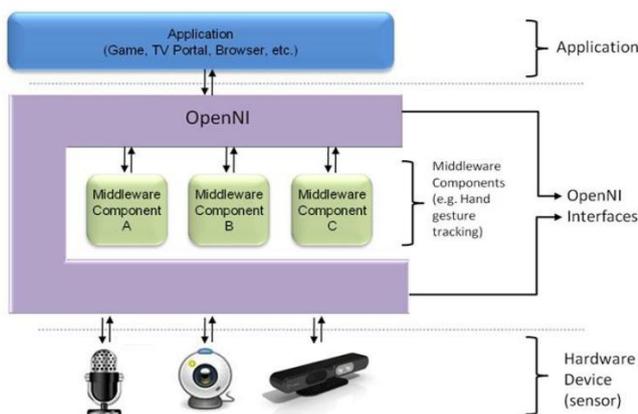


Figura 2 - Estrutura de camadas OpenNI [OpenNI, 11b]

4. ALGORITMOS DE RECONHECIMENTO CALIGRÁFICO

O reconhecimento de caracteres insere-se na área do reconhecimento de padrões, analisando características de uma imagem e decompondo-a de modo a extrair-lhe todas as particularidades que a caracterizam relativamente a outras amostras [Maxwell, 12]. Os reconhecedores dividem-se em dois grupos: estatísticos e conexionistas. Os classificadores estatísticos, nos quais se incluem os algoritmos KDA, SVM, \$1 e \$N, recorrem a um conjunto de amostras para aprendizagem antes do processo de classificação. Os classificadores conexionistas englobam algoritmos de redes neuronais e tentam simular as intercomunicações existentes entre os neurónios existentes num sistema neuronal biológico [Kolman and Margaliot, 09].

4.1 Kernel Discriminant Analysis

O KDA é uma extensão do algoritmo linear estatístico Linear Discriminant Analysis (LDA), usado no reconhe-

cimento de padrões e aprendizagem supervisionada [Yongmin et al., 01]. O KDA transpõe um problema para um espaço dimensional, representando as amostras através de vectores no espaço de características. Posteriormente, usando uma técnica Kernel, maximiza e determina não-linearmente a combinação de características representadas pelos vectores organizados em classes.

4.2 Support Vector Machine

O Support Vector Machine (SVM) é um algoritmo linear estatístico, descoberto por Vladimir Vapnik em 1963, aplicado no reconhecimento de padrões e aprendizagem supervisionada [Souza, 10b]. Este algoritmo procede a uma classificação linear binária dos dados de entrada, separando-os em duas categorias. O modelo de dados definido pelo SVM mapeia o conjunto de características através de vectores no espaço de características, separando-os em duas categorias, permitindo assim distinguir binariamente cada característica com base na categoria correspondente.

4.3 \$1

O \$1 é um método estatístico não-supervisionado conciso e simples, com uma taxa assinalável de reconhecimento de gestos [Anthony and Wobbrock, 2007a]. Funciona com base em geometria simples e trigonometria, que o torna simples e funcional, fazendo com que possa ser implementado em sistemas de prototipagem rápida. Neste são definidos dois conjuntos: o conjunto de pontos desenhados no momento para reconhecimento, designado de conjunto candidato C ; e as várias amostras de pontos existentes no momento para reconhecimento, designado de conjunto candidato T_i , que servem como modelos de comparação. O \$1 caracteriza-se pela sua total invariância⁴ no que respeita à orientação dos gestos, fazendo com que, por exemplo, os sinais '>' e '<', sejam considerados como sendo o mesmo símbolo no processo de reconhecimento, não obstante as suas diferentes orientações.

A comparação entre os dois conjuntos, C e T_i , T_i aplica o cálculo da distância euclidiana para determinar a melhor aproximação dos conjuntos de pontos existentes em T_i e que oferecem melhor alinhamento com o conjunto candidato C desenhado. Assim, com base nesta aproximação, é obtido o valor da menor distância d_i entre os vários conjuntos testados e quantificados, usando um sistema de pontuações, sendo que o que obtiver maior pontuação será o conjunto seleccionado [Wobbrock et al., 07b].

Antes do processo de procura e alinhamento do conjunto C , o caminho de pontos definidos por este é sujeito a quatro operações. A primeira passa por tornar os pontos de C equidistantes, reamostrando os pontos segundo um processo iterativo, no qual o distanciamento dos pontos varia consoante esse incremento. No segundo passo, procura rodar C através do ponto centróide⁵ definido pelo caminho dos pontos e pelo ponto em que se inicia o caminho dos pontos, realizando uma rotação para o ângulo indica-

⁴ Insensibilidade à rotação.

⁵ Ponto no interior de uma forma geométrica que define o seu centro geométrico.

tivo (0°) com base nestes dois pontos. Além de se encontrar o ângulo indicativo do conjunto C , são também procurados os ângulos indicativos dos conjuntos T_i aquando do seu carregamento, permitindo assim aperfeiçoar a procura do conjunto que melhor se alinha usando o método de alinhamento GSS⁶.

O terceiro passo resume-se a redimensionar não uniformemente o caminho de pontos para um quadrado tido como de referência, com um comprimento dos lados *size* igual em todas as amostras de T_i . Por fim, realiza a translação do centróide de C para a origem $(x, y) = (0, 0)$. Estes passos têm como intuito normalizar todos os gestos desenhados de forma a otimizar C , facilitando a correspondência dos pontos e o alinhamento com T_i , usando o GSS, como se pode observar na Figura 3 [Wobbrock et al., 07b].

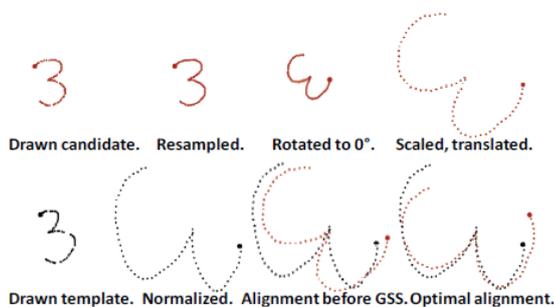


Figura 3 - Descrição dos passos de correspondência e alinhamento de um conjunto de pontos efectuado pelo \$1 [Anthony and Wobbrock, 07b]

4.4 \$N

O \$N baseia o seu funcionamento na mesma estratégia definida pelo seu predecessor \$1, realizando um conjunto de operações com base em geometria simples e trigonometria. Contudo, a grande melhoria encontra-se na versatilidade e capacidade de reconhecimento de gestos compostos por múltiplos traços [Anthony and Wobbrock, 07a].

Foram tidas em consideração duas questões, também abordadas por [Anthony and Wobbrock, 07b], que se prendem com a articulação dos gestos realizados e o efeito do número de amostras no rigor do reconhecimento. Relativamente à velocidade de articulação em que os gestos se processam, um aumento da velocidade não implica um aumento da taxa de erros no processo de reconhecimento, visto a distribuição dos pontos variar consoante o incremento do processo de reamostragem. Já o aumento do número de amostras melhora consideravelmente a precisão dos reconhecimentos.

O \$N foi sujeito a testes que permitiram determinar quais os melhores parâmetros a configurar, de forma a garantir uma taxa de reconhecimento razoável. As modificações

passaram pela alteração do número de pontos no processo de reamostragem em cada conjunto de pontos desenhados. A modificação do valor da reamostragem faz com que cada ponto seja representado com os espaçamentos entre pontos definido. Como resultado, ocorre um aumento ou diminuição da clareza dos pontos que constituem um traço. No gráfico da Figura 4 estão representados os 3 diferentes valores aplicados, 16, 64 e 96, num conjunto de 100 tentativas. Verifica-se que a reamostragem usando o incremento 64 oferece maior rigor.

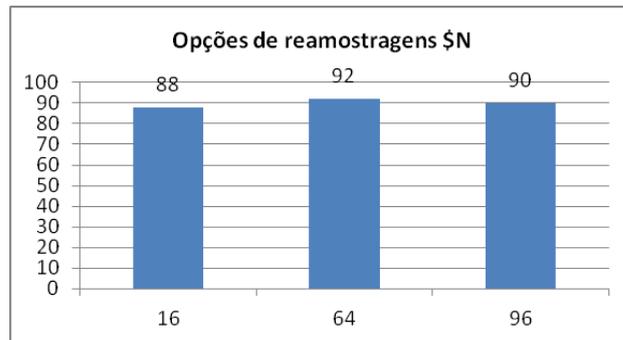


Figura 4 - Testes de eficiência do algoritmo \$N, usando diferentes valores de reamostragem

5. DESENVOLVIMENTO DA APLICAÇÃO

O software, desenvolvido em ambiente Windows, sobre a plataforma XNA Game Studio 3.1, visa definir uma interface de desenho de letras no ar que reconheça letras desenhadas por crianças do ensino básico. O dispositivo NUI Microsoft Kinect é usado como base para a concepção da aplicação, seguindo uma ordem de trabalhos que começa por localizar as articulações do corpo do utilizador, definir o plano de escrita e desenho das letras e, por fim, incorporar algoritmos de reconhecimento de padrões, para identificar as letras desenhadas.

Apesar de a versão oficial Microsoft Kinect SDK já se encontrar disponível actualmente, aquando do início deste projecto esta livreria estava apenas disponível na versão experimental, razão que levou a optar pelo uso da plataforma OpenNI. Outros factores influenciaram esta decisão, tais como o funcionamento com diferentes sistemas operativos, ser uma plataforma padronizada para qualquer equipamento, e tratar-se de uma plataforma de uso livre sujeita a constantes estudos, o que contribui para o aumento da sua maturidade e estabilidade.

Foram realizados testes com diferentes algoritmos de reconhecimento de padrões, que basearam a sua análise num conjunto de amostras representativas das 26 letras do alfabeto latino. Ambos os KDA e SVM usaram um conjunto de 260 amostras. Já o \$N usou um número bastante inferior, recorrendo para o efeito a apenas 41 amostras. A simplicidade e a flexibilidade na identificação de padrões, aliado ao reduzido número de amostras, fazem do \$N a melhor opção no processo de reconhecimento de padrões.

⁶ Técnica que consiste em encontrar o limite máximo ou mínimo recorrendo a uma função que analisa a distribuição dos valores dentro desses limites.

5.1 Interface com o utilizador

A aplicação desenvolvida pretende criar uma analogia com o modo de desenho de símbolos na areia da praia. Esta aplicação está dividida em seis grupos: detecção das articulações, configuração do painel de escrita, desenho de letras, reconhecimento de letras, gravação do desenho de letras e apresentação dos resultados das identificações.

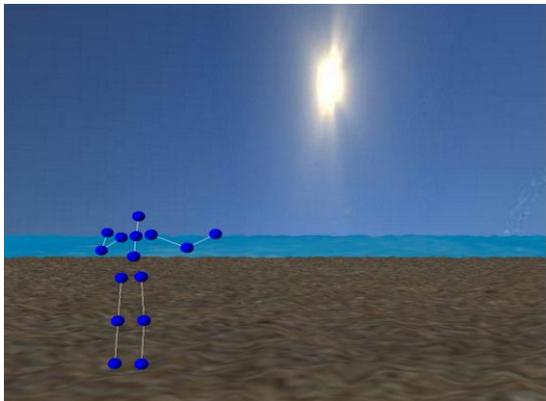


Figura 5 - Ecrã de localização das articulações do utilizador

O primeiro grupo, da detecção das articulações, destina-se à localização e rastreio do utilizador. Para isso, este deverá levantar os dois braços e baloiçá-los alternadamente para a esquerda e para a direita, até que o sistema o detecte e lhe reconheça as articulações do corpo. As articulações obtidas são posteriormente representadas através de esferas ligadas por intermédio de linhas, representando o esqueleto do utilizador e apresentando-o no cenário inicial da praia (Figura 5).



Figura 6 - Ecrã de indicação ao utilizador, destinado à selecção da mão de desenho que pretende usar

Depois de localizar o corpo, e ainda no cenário inicial da praia, segue-se a operação de Configuração do painel de escrita. Aqui, o utilizador deve estabelecer a distância entre si e o plano de escrita, movendo a mão direita para o lado direito, aproximadamente à mesma altura do ombro, e de seguida baixando a mão esquerda abaixo do nível da anca esquerda. Após definida a distância do uti-

lizador ao plano de escrita, aquele deve indicar qual das mãos pretende usar. Assim, quando aparece o ecrã ilustrado na Figura 6, deverá projectar para a frente a mão pretendida.



Figura 7 - Interface do ecrã de desenho

Depois de determinar os parâmetros inerentes à configuração do painel de desenho, é iniciada uma nova fase (Figura 7). Aí são concedidas cinco tentativas, nas quais o utilizador deverá acertar em cada uma das letras pedidas. Estas são geradas aleatoriamente de acordo com as 26 letras do alfabeto. O utilizador poderá movimentar o cursor (uma representação de um galho de uma árvore), movendo a mão de desenho ao longo de um plano situado mais próximo do seu corpo, ou então movimentar o cursor ao mesmo tempo que vai adicionando pontos ao ecrã, tendo a mão seleccionada de atravessar o plano mais afastado. Terminado o desenho estão disponíveis duas opções: a de limpar os pontos adicionados ou a de identificar o símbolo desenhado.



Figura 8 - Ecrã de selecção da letra a atribuir ao símbolo desenhada

Para identificar o conjunto de pontos que constituem o símbolo desenhado, surge um novo cenário para o grupo do reconhecimento de letras, onde é apresentado o resultado da identificação do símbolo, realizada pelo reconhecedor de padrões \$N. Aqui, estão unicamente disponíveis as opções para guardar ou não o referido símbolo. Caso não se pretenda guardar dever-se-á seleccionar a opção 'Não', sendo automaticamente direccionado para o grupo

de desenho de letras (o ecrã anterior, representado na Figura 7). Caso contrário, se seleccionar a opção para guardar o desenho, é então que surge um novo cenário (Figura 8), apresentando as 26 letras do alfabeto possíveis de associar ao conjunto de pontos do desenho que se pretende guardar. Para seleccionar a letra pretendida, o utilizador deverá projectar a mão de desenho até ao plano mais afastado, escolher a letra e, por fim, accionar o botão para guardar. Esta adição de um novo conjunto de pontos ao conjunto de amostras existentes enriquece o processo de reconhecimento e generalização de futuros desenhos.

Por fim, após terminarem as cinco tentativas disponíveis para o utilizador responder às letras geradas pela aplicação, são mostrados os resultados da identificação de cada símbolo pelo algoritmo de reconhecimento de padrões. As letras correctamente identificadas são assinaladas com um “visto” verde. Os erros são marcados com uma “cruz” vermelha.

5.2 Detalhes técnicos

De forma a obter as coordenadas das articulações, o utilizador deve começar por indicar a sua presença perante o dispositivo Microsoft Kinect (Figura 9), como referido anteriormente, até que as articulações sejam detectadas.

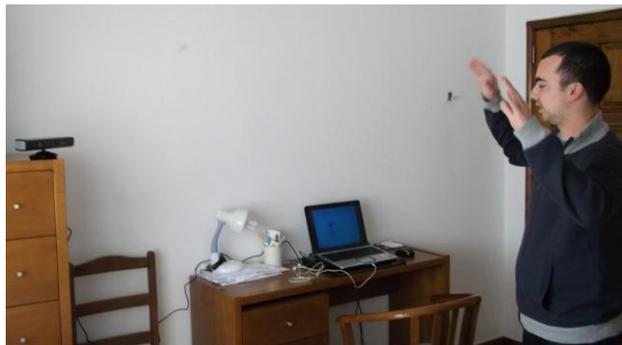


Figura 9 - Detecção da presença do utilizador, acenando com os braços acima da cintura

A definição do plano de escrita consiste, na realidade, num plano de dupla profundidade, estabelecido com base na medição da distância medida da mão direita ao ombro direito do utilizador (Figura 10).



Figura 10 - Postura definida para a determinação do duplo plano de interação

As áreas de interacção no ar foram pensadas como um conjunto de três segmentos, definidos em função do valor obtido da medição referida anteriormente. Esse valor é segmentado em três (Figura 11), ficando assim o segmento que vai dos 0% aos 25% a tratar-se de um segmento de ócio. Assim, sempre que a mão de escrita se encontrar dentro deste segmento, não será desencadeada qualquer acção.

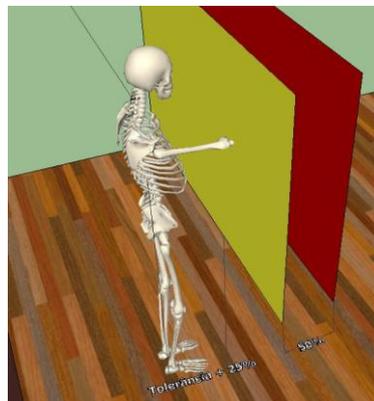


Figura 11 - Plano de escrita de duplo limite

O segmento seguinte é o maior dos três segmentos. Vai dos 25% aos 75% e é através deste que o utilizador deverá interagir maioritariamente, por se revelar essencial na movimentação e interacção com a aplicação. Sempre que o braço seja movimentado dentro destes limites, as coordenadas bidimensionais (x, y) da mão serão transpostas para o ecrã, movimentando o cursor também segundo as mesmas coordenadas. Neste segmento, os movimentos da mão do utilizador não se traduzem em desenhos no ecrã.

O último segmento vai dos 75% aos 100%. Nele, tal como no segmento anterior, é feita a transposição das coordenadas da mão para o cursor no ecrã. Neste caso, os gestos esboçados pelo utilizador são exibidos no ecrã.



Figura 12 - Utilizador dextro, com a mão projectada, prestes a ir de encontro ao limite estipulado para desenho

De seguida é necessário identificar qual das mãos será usada para desenhar as letras. Este processo foi pensado de forma a não limitar a interacção exclusivamente a utilizadores dextros, mas também ter em consideração o

público sinistro. Assim, o utilizador deverá projectar para a frente a mão que pretende usar de modo a interceptar o segmento de desenho (75%) como se encontra exemplificado na Figura 12.

Dos três algoritmos implementados, dois deles, o KDA e o SVM, recorrem a um método de classificação. Já o \$N recorre a trigonometria simples, o que o torna mais rápido no processo de carregamento inicial das amostras em comparação com os dois anteriores. O \$N apoia a sua análise num conjunto de 41 amostras, isto é, um número superior ao das 26 letras que constituem o alfabeto. Isto deve-se ao facto de cada uma das letras poder ser desenhada de diferentes formas, iniciando o traço de diferentes posições e variando o número de traços usados de acordo com a forma como a letra é traçada. Por exemplo, a letra 'A' pode ser desenhada com três traços ou simplesmente usando apenas dois.

Uma vez terminado um desenho, este é analisado pelo reconhecedor \$N, sendo indicado o resultado da identificação e apresentado o padrão que mais se assemelha de entre o conjunto de 41 amostras de aprendizagem usado.

6. VALIDAÇÕES

De modo a verificar a viabilidade da aplicação desenvolvida no âmbito deste projecto, procedeu-se a uma avaliação informal com um grupo de 20 participantes com idades compreendidas entre os 6 e os 25 anos, dos quais 7 elementos eram adultos, e os restantes crianças do ensino básico, tendo cada um deles realizado 10 tentativas. Ao longo dos testes foi possível verificar, por parte dos utilizadores, um certo embaraço nas duas tentativas iniciais, vindo progressivamente a adaptarem-se à interface ao longo das várias experiências realizadas. Considera-se positiva a adaptabilidade do utilizador à aplicação. No entanto, revelou-se essencial sujeitar o utilizador a um processo de aprendizagem. Ao longo das 10 tentativas, o utilizador vai-se familiarizando com os diferentes planos, adquirindo a prática necessária ao desenho das letras. Porém, após apenas três interações, o utilizador ainda demonstra debilidades relacionadas com a sensibilidade na transposição dos gestos, bem como dificuldades em situações em que se pretende remover a mão do desenho sem deixar rastros indesejados no ecrã. Estas situações carecem de um treino mais profundo para serem gradualmente corrigidas, melhorando consideravelmente a interação com o utilizador.

Relativamente ao reconhecedor de padrões eleito para a aplicação, o \$N, a definição de amostras com diferentes ordens de desenho das 26 letras do alfabeto revelou-se essencial para a precisão no reconhecimento, obtendo este algoritmo uma taxa de sucesso de 92%, em comparação com os outros reconhecedores de padrões Kernel Discriminant Analysis (KDA, 63%) e Support Vector Machine (SVM, 54%) testados (Figura 13).

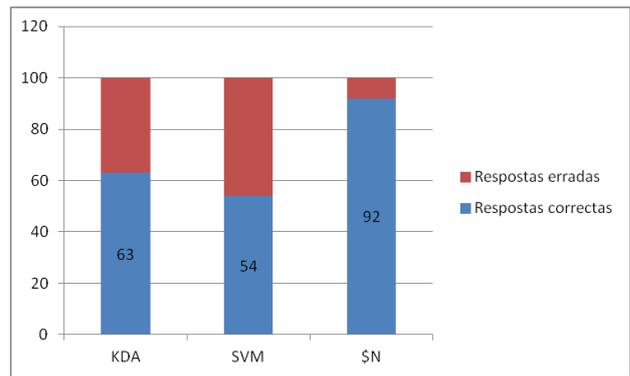


Figura 13 - Grau de eficiência dos algoritmos de reconhecimento de padrões testados

7. CONCLUSÃO E TRABALHO FUTURO

O estudo deste sistema de reconhecimento de gestos no ar, tem vindo a permitir conceber uma solução direccionada às crianças do ensino básico de forma a auxiliá-las, na aprendizagem da escrita. Além disso, os desenhos produzidos são posteriormente analisados pela aplicação com uma taxa de reconhecimento bastante positiva, nas identificações efectuadas pelo reconhecedor. Até ao momento, com base nos testes realizados em dois grupos de pessoas de diferentes faixas etárias, foi possível constatar as valias desta aplicação na área da educação e desenvolvimento motor das crianças, assim como diferentes níveis da percepção espacial, por parte dos utilizadores das diferentes faixas etárias. Apesar de os resultados obtidos indicarem que o público-alvo mais jovem demonstra uma maior dificuldade ao longo do processo de familiarização com os planos, em comparação com os utilizadores de idade mais avançada. Este público mais jovem demonstra, em contrapartida, uma maior capacidade de adaptação, melhorando gradualmente as suas capacidades de coordenação.

Um aspecto essencial, a considerar para os utilizadores de qualquer faixa etária, prende-se com a necessária familiarização aos planos de interação, por forma a interagir eficientemente com a aplicação.

Como perspectivas de trabalho futuro, considera-se a utilização de um modelo dinâmico com um esqueleto interno, constituído por um conjunto de junções que façam correspondência com as junções captadas pelo Kinect/OpenNI, permitindo desta forma animar um modelo com os gestos realizados pelo utilizador.

8. REFERÊNCIAS

- [Anthony and Wobbrock, 07a] Anthony L., Wobbrock J. \$N Multistroke Recognizer in JavaScript. *University of Maryland - Baltimore and University of Washington, USA, 2007.*
- [Anthony and Wobbrock, 07b] Anthony L., Wobbrock J. A Lightweight Multistroke Recognizer for User Interface Prototypes. *University of Maryland - Baltimore and University of Washington, USA, 2007.*

- [Kolman and Margaliot, 09] Kolman E., Margaliot M. Knowledge-Based Neurocomputing: A Fuzzy Logic Approach, Springer, p. 1-12, 2009
- [Maxwell, 12] Maxwell, Abril 2012.
<http://www.maxwell.lambda.ele.puc-rio.br/11498/11498_5.PDF>
- [Mehlmann et al., 11] Mehlmann G., Kistler F., Endrass B., André E., Klimmt C., Wagner J., Smirra R. Analysis of Interactive Storytelling Applications, UK, 2011.
- [OpenNI, 11a] OpenNI. Introducing OpenNI, Outubro 2011.
<http://75.98.78.94/images/stories/pdf/OpenNI_UserGuide_v4.pdf%20%5b1>
- [OpenNI, 11b] OpenNI. ProgrammerGuide, Outubro 2011.
< <http://openni.org/docs2/ProgrammerGuide.html>>
- [Pinto, 12b] Pinto P., 2012. SDK v1.5 do Kinect para Windows disponível para download, Maio 2012.
<<http://pplware.sapo.pt/windows/software/sdk-v1-5-do-kinect-para-windows-disponvel-para-download/>>
- [Rogers, 11] Rogers R., 2011. Journal. Kinect with Linux. Houston.
- [SideKick, 11] SideKick, , Outubro 2011.
< <http://www.sidekick.co.il/>>
- [Souza, 12a] Sousa C., 2010. - Kernel Support Vector Machines for Classification and Regression in C#, Abril 2012.
<<http://crsouza.blogspot.pt/2010/04/kernel-support-vector-machines-for.html>>
- [Souza, 10b] Souza C.,2010. Handwriting Recognition Revisited: Kernel Support Vector Machines. The *Code Project*. Brazil.
- [Sylverberg T., 07] Sylverberg T., Kristensson P., Leifler O., Berglund E., Drawing on Paper Maps: Reliable On-line Symbol Recognition of Handwritten Symbols Using a Digital Pen and a Mobile Phone, 2007.
- [Tanaka et al., 12] Tanaka K., Parker J., Baradoy G., Sheehan D., Holash J, Katz L. A Comparison of Exergaming Interfaces for Use in Rehabilitation Programs and Research, Vol 6, 2012.
- [Tscherrig, 11] Tscherrig J., 2011. Activity Recognition using Kinect.
- [Willow Garage, 11] Willow Garage, Outubro 2011.
<<http://www.willowgarage.com/blog?page=8>>
- [Wobbrock et al., 07a] Wobbrock J., Wilson A., Li Y. \$1 Unistroke Recognizer in JavaScript. University of Washington and Microsoft Research, USA, 2007.
- [Wobbrock et al., 07b] Wobbrock J., Wilson A., Li Y. Gestures without Libraries, Toolkits or Training:A \$1 Recognizer for User Interface Prototypes. University of Washington and Microsoft Research, USA, 2007.
- [Yongmin et al., 01] Yongmin L., Shaogang G., Liddell H.,Kernel Discriminant Analysis. University of London, *Department of Computer Science*, UK, 2001

Sistema integrado de geração automática de conteúdo para videojogos de plataformas

Fausto Mourato

CITI - Departamento de Informática
Faculdade de Ciências e Tecnologia
Universidade Nova de Lisboa
2829-516 Caparica
p21748@campus.fct.unl.pt

Fernando Birra

CITI - Departamento de Informática
Faculdade de Ciências e Tecnologia
Universidade Nova de Lisboa
2829-516 Caparica
fpb@di.fct.unl.pt

Manuel Próspero dos Santos

CITI - Departamento de Informática
Faculdade de Ciências e Tecnologia
Universidade Nova de Lisboa
2829-516 Caparica
ps@di.fct.unl.pt

Resumo

Neste artigo é apresentado um sistema de suporte à geração automática de níveis para videojogos de plataformas. Este assenta numa *framework* genérica de representação de níveis, compatível com grande parte dos conceitos existentes neste tipo de jogos. Numa perspectiva de edição de conteúdo, o sistema pode ser utilizado como editor de níveis genérico. A arquitectura desenvolvida é modular e baseada em *plugins*, permitindo a adaptação para outros formatos de níveis. Com um princípio similar, é possível integrar algoritmos de geração automática e adaptação de conteúdo. É abordada a arquitectura da solução proposta e a forma como é possível estabelecer um pipeline de algoritmos de geração automática. Finalmente, são também descritos alguns destes algoritmos.

Keywords

Videojogos de plataformas, Geração Procedimental de Conteúdo, Ajuste de Dificuldade

1. INTRODUÇÃO

A geração automática de conteúdo no contexto dos videojogos tem suscitado interesse na comunidade académica e em equipas de desenvolvimento de pequena dimensão. Por um lado, trata-se de uma forma de suprimir a escassez de recursos humanos comparativamente às grandes empresas do segmento e, por outro lado, trata-se de um desafio intelectual interessante de algoritmia onde diversos aspectos têm que ser considerados.

Neste artigo abordamos um sistema desenvolvido para o suporte de diversas técnicas de geração automática de níveis, em particular para videojogos de plataformas, onde serão salientados os seguintes aspectos:

- A arquitectura baseada em *plugins*, potenciando a expansão da aplicação para outros jogos e permitindo a inclusão de novas técnicas de geração automática.
- O mecanismo baseado em regras para extracção automática de possíveis caminhos dentro de um nível e a análise dos mesmos.
- Os principais algoritmos desenhados e implementados para a geração automática de níveis.
- Uma técnica de ajuste de dificuldade e adaptação de conteúdo, desenhada para a criação de desafios compostos por um trajecto não linear, baseado na recolha de objectos e activação de *triggers*.

O referido sistema pretende potenciar o desenvolvimento de novos videojogos de plataformas a partir de uma metodologia comum capaz de integrar diversos algoritmos

para geração automática. Contudo, é possível utilizar a aplicação enquanto simples editor de níveis sem recurso a geração procedimental.

Os casos de estudo apresentados abordam jogos existentes, considerando o potencial que a aplicação teria se existisse e tivesse sido utilizada como apoio aquando do desenvolvimento desses mesmos jogos.

Relativamente à estrutura deste documento, na Secção 2 são referidos os estudos mais importantes relacionados com a geração automática de conteúdo neste tipo de jogos. Na Secção 3 descreve-se o sistema implementado, com principal ênfase na *framework* para representação de níveis e na arquitectura modular da aplicação. De forma mais superficial, são também abordados alguns dos algoritmos implementados. A Secção 4 complementa a descrição com uma visão geral sobre a utilização da aplicação com recurso a alguns exemplos. Finalmente, as principais conclusões e aspectos para desenvolvimento futuro são apresentados na Secção 5.

2. TRABALHO RELACIONADO

O estudo dos videojogos de plataformas foi iniciado em contexto académico por Compton e Mateas [Compton06]. Os autores analisaram os principais componentes deste tipo de jogos e definiram um modelo conceptual para os caracterizar. Foram também identificados os principais padrões de trajectos.

Posteriormente, uma abordagem mais detalhada foi apresentada por Smith *et al.* [Smith08], com a definição de uma hierarquia para descrição do conteúdo de um nível

com vista à análise do mesmo. De uma perspectiva geral, um nível é apresentado como um conjunto de células interligadas por portais. Uma célula é vista como uma região do nível de tamanho variável, onde a personagem pode realizar um trajecto directo composto por um ou mais grupos de ritmo. Estes grupos representam uma sequência coerente de acções a realizar pelo jogador. Finalmente, a um grupo de ritmo são associados diversos elementos de jogo, nomeadamente: plataformas, obstáculos, itens, *triggers* e auxiliares de movimento. Os princípios apresentados deram seguimento a uma abordagem para a geração automática de conteúdo neste tipo de jogos, designada por *Rhythm Based* [Smith09]. A geração baseada em ritmo consiste na definição inicial de um conjunto coerente de acções para o utilizador realizar e, em seguida, na criação de uma geometria para o nível na qual essas acções sejam possíveis. Esta abordagem concretizou-se num protótipo intitulado de *Launchpad* [Smith11]. Paralelamente a esta linha de trabalho, há ainda a referir alguns estudos relacionados dentro do mesmo grupo de investigação. A capacidade de expressão desta abordagem foi explorada nas vertentes de linearidade do caminho e uma aproximação ao conceito de dificuldade [Smith10a]. Adicionalmente, realizaram-se testes sobre uma adaptação da ideia inicial numa abordagem mista de criação automática com troços fixos definidos por um utilizador humano [Smith10b]. Finalmente, os estudos incidiram ainda sobre a possibilidade de adaptação da geração baseada em ritmo para ter em conta o desempenho do utilizador, adequando o desafio das acções com a perícia demonstrada [Jennings-Teats10].

Uma abordagem mais directa foi proposta por Pedersen *et al.* [Pedersen09]. O gerador apresentado parte de uma parametrização inicial onde se define o número de saltos a realizar, a sua amplitude média e a frequência com que estes acontecem, entre outros. Este gerador foi usado para realizar sessões de jogo com utilizadores e extrair percepções como, por exemplo, a de dificuldade.

Finalmente, Mawhorter e Mateas [Mawhorter10] propuseram uma outra técnica alternativa interessante com o intuito de gerar níveis de forma automática. A abordagem parte da criação inicial de um conjunto de amostras de níveis, as quais são replicadas e compostas numa grelha pela aplicação do algoritmo *Occupancy-Regulated Extension* (ORE), o qual é apresentado nesse mesmo artigo.

Desde o ano de 2010, a *IEEE Conference on Computational Intelligence and Games* tem apresentado um *track* específico sobre o tema da geração automática de níveis, utilizando-se para o efeito uma versão adaptada do jogo *Infinite Mario Bros.* Algumas das técnicas previamente desenvolvidas foram testadas também neste contexto. Os detalhes sobre a primeira edição da competição podem ser consultados no artigo *The 2010 Mario AI Championship: Level Generation Track* [Shaker10].

3. O SISTEMA IMPLEMENTADO

No âmbito deste trabalho, concebeu-se e implementou-se uma aplicação na linguagem C# sobre a plataforma .NET funcionando como sistema para integração de diversas

técnicas de geração automática. Esta assenta numa *framework* que permite representar genericamente um qualquer nível de plataformas. Em seguida, na Secção 3.1 é apresentada a estrutura desta *framework* e a forma como os níveis podem ser representados. Na Secção 3.2, são apresentados os detalhes da integração com formatos de alguns jogos existentes. Posteriormente, na Secção 3.3, é explicada a arquitectura modular que permite a integração de algoritmos de geração automática. A Secção 3.4 aborda a questão da extracção automática de caminhos a partir de grafos e, finalmente, na Secção 3.5 são apresentados os principais algoritmos de geração incluídos no sistema.

3.1 Modelo de representação de níveis

Conforme referido na Secção 2, foi apresentada uma proposta de *framework* por [Smith08], com o foco principal na representação conceptual de um nível e na identificação de regiões. No caso do trabalho aqui apresentado, pretende-se uma representação mais concreta do conteúdo, capaz de mapear directamente a informação existente e com menor grau de ambiguidade.

Como ponto de partida para a definição dos níveis, assumiu-se uma representação baseada em enumeração espacial discreta sob a forma de uma matriz bidimensional. É comum em videogames de plataformas o recurso a representações neste formato. Por exemplo, a estrutura dos níveis dos jogos *Infinite Mario Bros.*, *Prince of Persia* e *Rick Dangerous* consiste precisamente numa matriz bidimensional. Adicionalmente, e embora se trate, aparentemente, de uma restrição ao domínio, este tipo de representação potencia um conjunto de técnicas para geração automática, como se irá constatar ao longo desta secção. Assim, um nível é composto por um conjunto de **células**, preenchidas a partir de um conjunto pré-definido de **blocos**. Para aumentar a capacidade de representação desta abordagem, o conjunto de blocos associado a um nível ou jogo é representado numa hierarquia. Em cada célula é ainda possível definir um conjunto de **propriedades** e respectivos **valores** estabelecido para cada tipo de bloco.

Para esclarecer os conceitos anteriores, considere-se como exemplo um jogo minimalista baseado no universo *Super Mario Bros.*, possuindo somente os seguintes tipos principais de blocos:

- *Preenchido*, representando um bloco de existência física onde a personagem não pode passar e que serve também de eventual suporte para a mesma.
- *Vazio*, representando um espaço por onde a personagem pode passar e em que o conteúdo gráfico existente é meramente estético.
- *Plataforma*, representando restrição somente ao movimento vertical descendente da personagem, servindo assim como suporte para a mesma sem impedir o seu movimento, como se estivesse numa camada distinta. A Figura 1 apresenta exemplos de utilização deste tipo de bloco nos jogos *Infinite Mario Bros.* e *XRick*.

Considere-se ainda que o bloco vazio possui variantes para a representação do início e fim do nível e duas reproduções visuais distintas de carácter estético. A sua representação poderia assentar na hierarquia apresentada na Figura 2. Com base neste modelo inicial é possível, por exemplo, descrever o nível apresentado na Figura 3.

Suponha-se também que, na implementação do hipotético jogo correspondente a esta descrição, as flores poderiam ser visualizadas com uma qualquer cor. Neste caso, poder-se-ia definir uma propriedade *Cor* para o bloco *Flor* na qual se poderia indicar o valor RGB que descrevesse essa mesma cor. De momento, do ponto de vista do sistema os valores são todos representados em formato de texto. No entanto, tal não impede que a efectiva implementação do jogo possa processar os dados e efectuar a sua conversão para um qualquer formato.

Adicionalmente, a *framework* contempla o conceito de **grupo**, o qual permite a definição de um conjunto de **blocos** que só fazem sentido se interpretados em conjunto. A definição de grupos é opcional mas, no entanto, é útil tanto para a edição manual de um nível como para geração automática, uma vez que providencia informação adicional sobre a possível organização das células. A existência deste tipo de estruturas é comum em abordagens baseadas em células. Por exemplo, no jogo *Prince of Persia* a representação das portas de início e fim do nível recorre ao uso de duas células, tal como é possível observar na Figura 4. Do mesmo modo, um cano no jogo *Infinite Mario Bros.* pode ser visto como um exemplo de um **grupo**. Contudo, este caso em particular possui uma característica complementar que também é tida em conta na *framework*. Apesar de possuírem uma largura fixa de duas células, os canos neste jogo têm altura variável. Este tipo de situações é mapeado sob o conceito de **grupo de dimensão variável**.

Com o intuito de potenciar a possível ligação entre a representação e os processos de geração automática, implementou-se no sistema um mecanismo para definição de **regras** baseadas em padrões, permitindo ao utilizador definir agrupamentos especiais de células que podem originar um determinado processamento. Actualmente, esta funcionalidade tem sido utilizada para a representação aproximada dos principais movimentos da personagem, facultando a posterior extracção automática de grafos de trajectos para os níveis. Deste modo, é possível definir conjuntos de blocos e indicar um sub-grafo associado a cada conjunto, representando os possíveis movimentos da personagem do jogo.

Durante o estudo da capacidade de representação dos níveis de diversos jogos, verificou-se que, por vezes, a hierarquia não é suficiente para designar directamente todos os conjuntos lógicos de blocos. No exemplo simples definido anteriormente, esta questão encontra-se presente. Os blocos *Plataforma* podem ser interpretados da mesma forma que *Vazio*, para algumas situações, ou *Preenchido*, para outras. No movimento horizontal descrito acima, o deslocamento indicado para a personagem

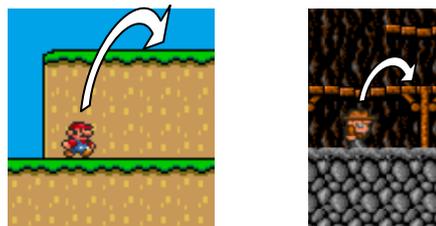


Figura 1 – Exemplo de simulação de múltiplas camadas em jogos de plataformas bidimensional (*Infinite Mario Bros.* à esquerda e *XRick* à direita)

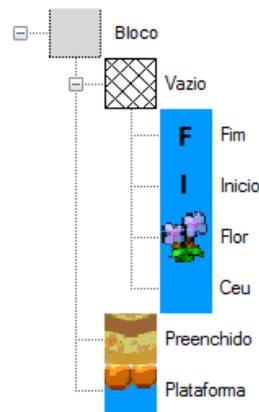


Figura 2 – Exemplo de uma hierarquia de blocos



Figura 3 – Nível de exemplo utilizando uma hierarquia de blocos simples

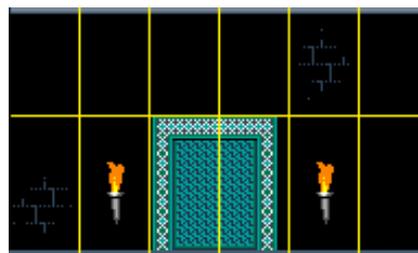


Figura 4 – Exemplo da representação em grelha no jogo *Prince of Persia*

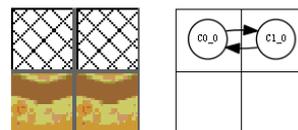


Figura 5 – Exemplo de uma regra para construção do grafo de trajectos. À esquerda é definido o padrão a procurar no nível e à direita é indicado o sub-grafo correspondente.

seria também válido com blocos do tipo *Plataforma* nas células de baixo servindo como suporte, bem como para células de cima, pois não impedem a passagem da personagem.

No entanto, não é possível definir o elemento *Plataforma* como nó filho dos nós *Vazio* e *Preenchido* simultaneamente numa hierarquia simples. Deste modo, estabeleceu-se também a possibilidade de definir **categorias** de blocos, construídas com referências à hierarquia e com recurso a operadores lógicos. Neste caso, poderia ser construída uma categoria *Suporte* através de uma disjunção, usando a expressão $\text{Bloco/Preenchido} \vee \text{Bloco/Plataforma}$. Do mesmo modo, poder-se-ia definir uma categoria *Movimentável* novamente com recurso a uma disjunção, neste caso com a expressão $\text{Bloco/Vazio} \vee \text{Bloco/Plataforma}$. Assim, a forma mais completa de descrever o movimento horizontal da personagem seria a regra apresentada na Figura 6.



Figura 6 – Exemplo de uma regra para construção de trajectos, utilizando *Categorias*.

Sintetizando a *framework* acima descrita, o modelo de representação de níveis é composto pelos seguintes elementos: Hierarquia de blocos, Grupos, Grupos de dimensão dinâmica e Regras.

O sistema implementado permite a serialização da informação anterior em formato XML para possível partilha e reutilização.

Numa primeira abordagem à aplicação desenvolvida, esta pode ser vista como um editor de níveis para o domínio definido. Os níveis são representados para um determinado modelo descrito na *framework* anterior, contendo uma matriz bidimensional de células com referência aos blocos que as descrevem e eventuais pares de parâmetros e respectivos valores. Os níveis podem também ser guardados e lidos em formato XML.

3.2 Integração com diversos videojogos

A aplicação possui uma estrutura modular para potenciar a representação de níveis de quaisquer jogos de plataformas. É possível implementar *plugins* específicos para jogos em particular desde que os níveis estejam de alguma forma acessíveis. Um *plugin* para um determinado jogo deverá conter, quando aplicável:

- O modelo de representação dos níveis desse jogo, descrito na *framework* apresentada anteriormente.
- A implementação de um método capaz de ler os dados a partir do jogo original e traduzi-los no modelo definido.

- A implementação de um método para exportar os dados de um nível criado na aplicação para substituir os dados originais desse jogo.
- Uma implementação de um método para lançar o jogo com o nível em visualização na aplicação.

À data, foram realizados testes de integração com os videojogos *Prince of Persia*, *Rick Dangerous*, *Infinite Mario Bros* e *Open Sonic*. A escolha destes títulos deveu-se principalmente ao facto de serem versões livres, de código aberto ou com existência de um formato conhecido e passível de edição. De facto, o maior entrave à integração com outros jogos está no facto de estes possuírem níveis tipicamente em formato proprietário e cujas especificações não são do domínio público.

O principal intuito do estudo desta capacidade de integração é, acima de tudo, o reconhecimento da abrangência da *framework* e a identificação de aspectos que possam ser melhorados.

3.3 Arquitectura para Geração Automática

A aplicação permite a definição de um *pipeline* de técnicas sucessivas para geração automática de níveis. Para o efeito, considera-se que uma técnica de geração automática é definida pelas seguintes características:

- Possui um conjunto predefinido de parâmetros de configuração.
- Indica o número de níveis que devem ser facultados ao algoritmo para uma determinada parametrização (dimensão do *input*).
- Indica o número de níveis que o algoritmo produz para uma determinada parametrização (dimensão do *output*).

Por exemplo, considere-se um gerador simples para o jogo *Infinite Mario Bros*. que gere sequências de blocos de terreno com a mesma altura, separados entre si. Os valores de espaçamento médio, mínimo e máximo são exemplos de possíveis parâmetros de configuração. Neste caso particular, independentemente da configuração, o número de níveis que o algoritmo recebe será 0 e o número de níveis gerados será, tipicamente, 1. Eventualmente, poderá ser considerada uma variante que permita a geração de múltiplos níveis em simultâneo, sendo o número de níveis definido como parâmetro.

É possível encadear um qualquer número de algoritmos de geração, preferencialmente se as dimensões de *output* e *input* forem equivalentes em algoritmos sucessivos. Nos casos em que esta condição não se verifica, é possível descartar resultados de um fase anterior e/ou forçar múltiplas invocações dessa mesma fase, impondo um determinado número de níveis.

Considere-se um segundo exemplo de um algoritmo de decoração que complemente um nível com elementos estéticos, ou seja, o algoritmo recebe 1 nível e produz como *output* também 1 nível. Assim, estes dois algoritmos podem funcionar de forma sucessiva, sendo que o segundo processa o *output* do primeiro, produzindo o

resultado final. Na Figura 7 é apresentado um exemplo da utilização destes dois algoritmos em sucessão. Na Figura 8 é apresentada a interface para o utilizador definir a sequência de algoritmos que pretende utilizar.

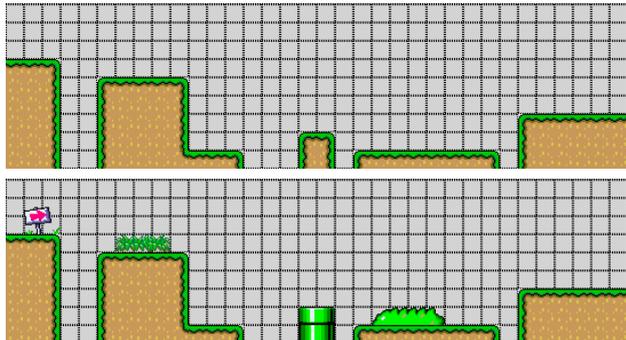


Figura 7 – Exemplo de construção automática de níveis com sequência de algoritmos. Em cima, o resultado de um primeiro algoritmo que cria a estrutura base do nível. Em baixo, o resultado de um algoritmo de decoração aplicado ao primeiro nível.

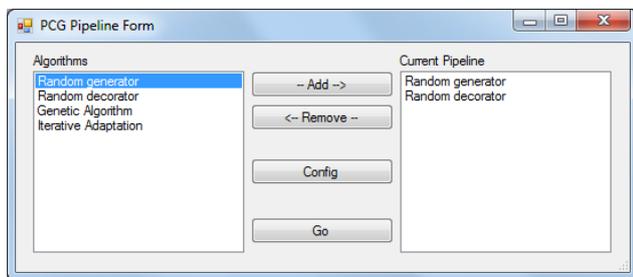


Figura 8 – Janela de configuração do pipeline de geração automática de conteúdo

3.4 Extracção automática de grafo de trajecto

Como foi explicado na Secção 3.1, é possível definir regras para associar um padrão de blocos a um sub-grafo. Uma das ferramentas incluídas no sistema permite o processamento de todas as regras para criação de um grafo geral dos trajectos de um nível. Na Figura 9 é possível observar o resultado da aplicação desta ferramenta num nível do jogo *Prince of Persia*.

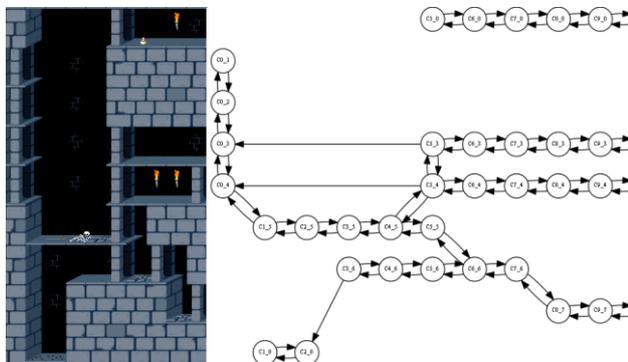


Figura 9 – Exemplo de um grafo de trajectos no primeiro nível do jogo *Prince of Persia*

Para tornar o processamento do grafo mais eficiente, é possível aplicar um passo de compactação ao mesmo,

removendo os nós que representem somente transições obrigatórias entre dois outros nós. Na Figura 10 é apresentado o grafo do exemplo anterior após compactação.

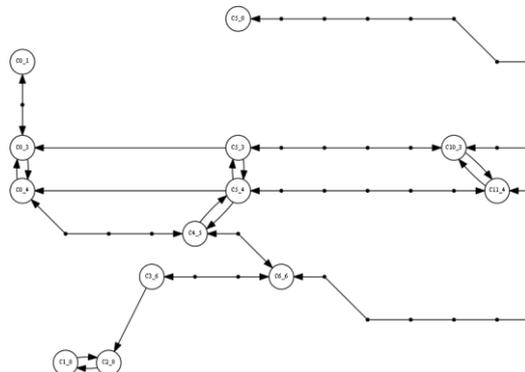


Figura 10 – Exemplo de um grafo compactado

Uma das funcionalidades expectáveis após a extracção do grafo é o cálculo do caminho mais curto entre dois pontos, tendo esta sido implementada recorrendo ao algoritmo de Dijkstra. Contudo, há limitações naturais na análise do grafo. Por um lado, existe a limitação prática associada à complexidade computacional, caso sejam representadas situações menos triviais, como caminhos de uso único, recolha de recursos ou interruptores. Um estudo detalhado sobre este assunto pode ser consultado no artigo intitulado *Gaming is a hard job, but someone has to do it* [Viglietta12]. Por outro lado, o nível a analisar pode ser uma representação incompleta num passo intermédio de uma sequência de algoritmos. Assim, um dos intuítos da criação e análise do grafo pode ser o auxílio à construção de um nível mais completo.

Deste modo, a análise ao grafo contempla, adicionalmente, o cálculo de todos os caminhos possíveis entre os dois pontos definidos como início e final do nível, excluindo todos os trajectos que repetem nós. Posteriormente, os nós do grafo são classificados consoante o seu papel no trajecto do nível, existindo as seguintes classificações:

- **Obrigatório**, para os nós que estão presentes em todos os trajectos entre o ponto inicial e o ponto final.
- **Opcional**, para os nós que fazem parte somente de alguns dos trajectos entre o ponto inicial e o ponto final.
- **Fim de caminho**, para os nós onde a única saída é um nó de entrada.
- **Caminho sem saída**, para os nós que não fazem parte de nenhum dos trajectos mas fazem parte de uma ligação entre um ou mais desses trajectos e um nó classificado como *Fim de Caminho*.
- **Inacessível**, para os nós que correspondem a posições que a personagem não consegue alcançar.
- **Sem retorno**, para os nós que correspondem a posições alcançáveis pela personagem mas que não permitem o retorno a nós dos caminhos válidos para finalizar o nível.

Esta classificação faculta um conhecimento adicional sobre a estrutura do nível. Na Secção 3.5.2 é apresentado um algoritmo para adaptação do conteúdo de um nível que tira proveito desta classificação.

3.5 Algoritmos de geração automática

Neste ponto abordamos duas técnicas implementadas e testadas sob o sistema apresentado.

3.5.1 Geração com Algoritmos Genéticos

Uma das técnicas implementadas permite a geração da estrutura base para níveis do jogo *Prince of Persia*, muito embora a mesma abordagem possa ser aplicada a jogos similares, em cenários tipicamente fechados com salas interligadas por corredores. Esta consiste na definição de um algoritmo genético onde os níveis são abordados como indivíduos de uma população, e passam por um processo de evolução mediante um conjunto de regras de avaliação, mutação e combinação. Estas foram estabelecidas a partir de heurísticas de desenho, tendo em conta os seguintes aspectos:

- Estrutura do caminho principal, nomeadamente em relação à linearidade e ramificação do mesmo.
- Análise individual de cada célula em relação ao seu contexto (vizinhança).
- Posicionamento da célula final do nível e dimensão do caminho principal criado.
- Equilíbrio do uso dos diversos tipos de bloco.
- Uso adequado do espaço disponível.

Esta técnica permite a geração de estruturas base como a apresentada na Figura 11. Uma descrição mais detalhada do funcionamento do algoritmo e dos testes realizados pode ser consultada em [Mourato11].

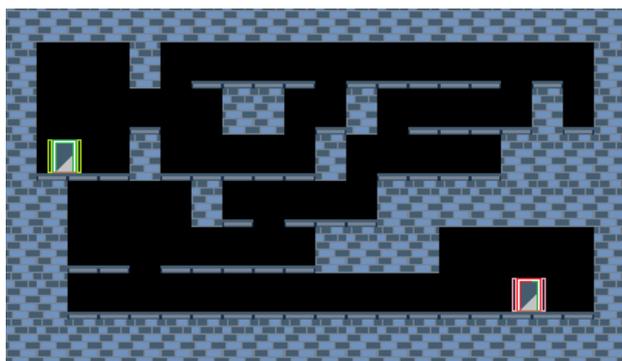


Figura 11 – Exemplo de um nível gerado automaticamente dentro da aplicação a partir de um algoritmo genético

3.5.2 Algoritmo Iterativo para adaptação de conteúdo e ajuste de dificuldade

O algoritmo de adaptação implementado visa a personalização de um nível existente. Este pode ter sido manualmente criado ou gerado automaticamente por um algoritmo como o apresentado no ponto anterior. Tipicamente, deverá recorrer a um conjunto limitado de blocos, descrevendo a geometria base, uma vez que será este algo-

ritmo o responsável pela colocação de outro tipo de blocos que tornam o conteúdo do nível mais completo.

O ponto de partida para este processo é a análise do grafo apresentada anteriormente na Secção 3.4, nomeadamente no que respeita à classificação dos nós desse mesmo grafo. Adicionalmente, o caminho é decomposto em segmentos sendo identificadas as diversas alternativas de caminho para cada segmento. A ideia principal é a de alterar alguns dos blocos existentes para atingir um determinado valor de comprimento no caminho a realizar, dentro de uma determinada estimativa de dificuldade. As alterações são realizadas iterativamente em diversas passagens em cada segmento do nível.

Com base na informação recolhida, o algoritmo procede ao seguinte tipo de alterações:

- Alteração da dificuldade de um segmento, pela inclusão ou remoção de entidades adicionais nesse mesmo segmento, como por exemplo adversários ou armadilhas. As probabilidades de acrescentar ou remover entidades são definidas a partir da estimativa geral de dificuldade.
- Criação de desvios ao trajecto, pela colocação de interruptores em caminhos sem saída e portões ou entidades semelhantes no caminho principal, associados a esses interruptores. Este tipo de alteração permite controlar o comprimento do trajecto a realizar.
- Ajustes para jogabilidade com dois jogadores de perícia distinta, adaptando segmentos paralelos, tornando um deles particularmente mais difícil que o outro e forçando a utilização de caminhos distintos entre os jogadores.
- Inclusão de entidades de bónus em caminhos sem saída para conferir significado a essas mesmas regiões do nível.

Na Figura 12 é apresentado um exemplo de nível criado com este algoritmo, novamente aplicado ao jogo *Prince of Persia*, tendo como base o nível gerado inicialmente com a técnica anterior.

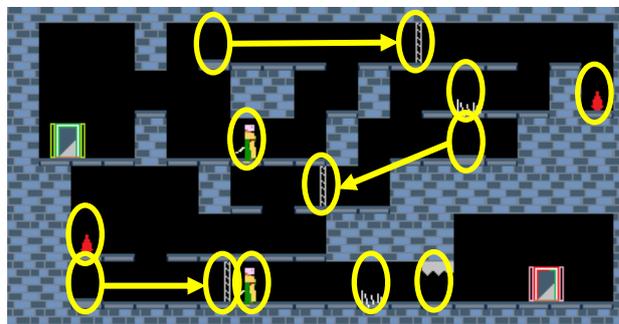


Figura 12 – Exemplo de um nível ajustado pelo algoritmo iterativo de adaptação de conteúdo. As modificações efectuadas estão marcadas com uma elipse. A associação entre botões e portões encontra-se marcada com setas.

Todas as entidades presentes no nível para além da geometria base foram colocadas pelo algoritmo de adaptação. É possível observar a existência de guardas e armadilhas no caminho, sendo o seu número dependente da dificuldade requerida pelo utilizador. É também possível observar a colocação de duas poções em troços sem saída. Finalmente, o sistema colocou também três portões fechados aos quais associou botões para os abrir, tendo estes sido colocados fora do trajecto principal, forçando o utilizador a realizar desvios a esse mesmo trajecto.

Uma descrição mais detalhada desta abordagem pode ser consultada em [Mourato12].

4. EXEMPLOS DE UTILIZAÇÃO

Nesta secção é descrito o funcionamento da aplicação, clarificando-se as suas potencialidades.

Como ponto de partida, na Figura 13 é apresentada uma visão geral da janela principal do programa. Nesta também é possível observar a aplicação de um dos *plugins* desenvolvidos, neste caso para importar níveis do videojogo *Rick Dangerous*.

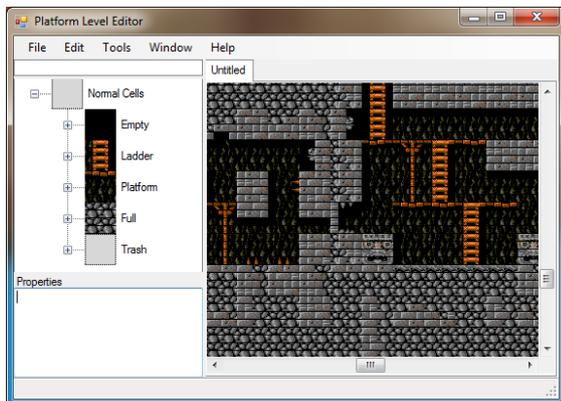


Figura 13 – Janela principal da aplicação desenvolvida, visualizando um nível importado do videojogo *Rick Dangerous*

Na Figura 14 é possível visualizar a capacidade de geração de níveis e utilização dentro de um jogo, aqui concretamente para uma versão adaptada do *Infinite Mario Bros*. Como foi referido anteriormente, este programa tem sido utilizado em contexto académico, no qual têm sido implementados alguns algoritmos de geração automática. Neste caso, o gerador criado acede aos dados do nosso sistema, lendo um nível no formato XML e interpreta-o como se o gerador fosse interno ao próprio jogo.

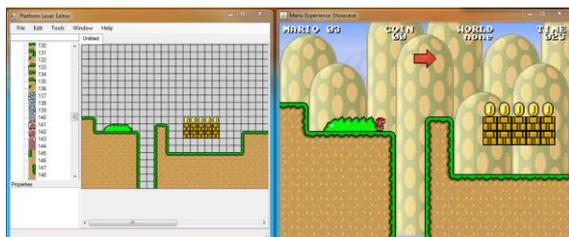


Figura 14 – Visualização lado a lado da janela de criação e edição de um nível e a janela do jogo *Infinite Mario Bros*. executando esse mesmo nível

Conforme indicado, a aplicação permite a definição de todo o modelo de representação dos níveis. Na descrição do sistema apresentou-se um exemplo de hierarquia, tendo sido considerado um caso simples. De uma forma geral, a definição do modelo não é particularmente complexa. Por exemplo, considerando o videojogo *Prince of Persia*, é possível uma adequação ao jogo original recorrendo à definição de 6 categorias para agrupar os diversos tipos de blocos e 16 regras para definir o cálculo dos diversos trajectos possíveis para a personagem. Na Figura 15 é possível observar a janela de configuração das categorias e, na Figura 16, é apresentada a janela de definição das regras para a trajectória da personagem. Nessas figuras são apresentados os dados de teste utilizados no jogo *Prince of Persia*, conforme indicado acima.

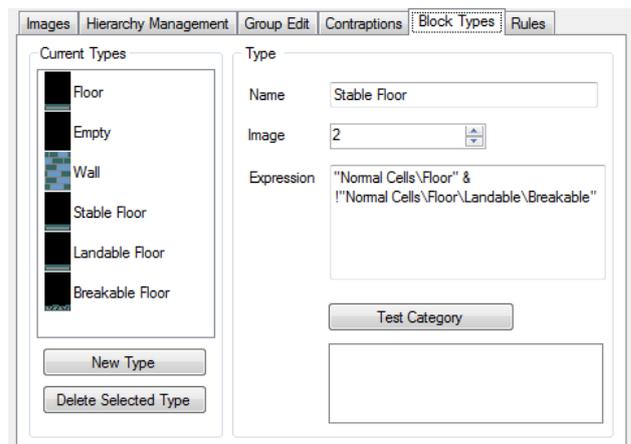


Figura 15 – Janela de edição do modelo de representação de níveis, separador de definição de categorias.

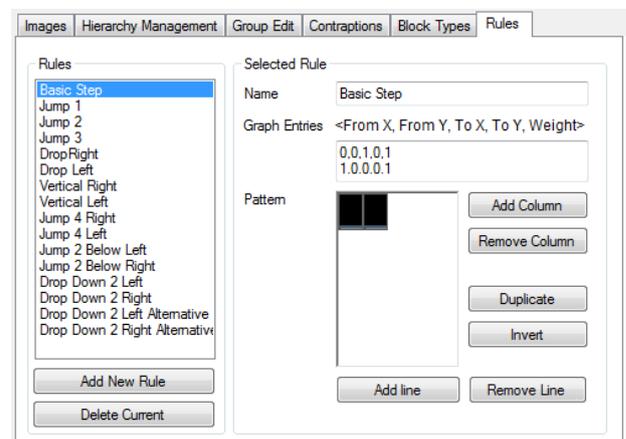


Figura 16 - Janela de edição do modelo de representação de níveis, separador de definição de regras.

5. CONCLUSÕES E TRABALHO FUTURO

No presente artigo descreveu-se um sistema integrado de suporte à criação de níveis para videojogos de plataformas, no qual foram implementadas diversas ferramentas de apoio à geração automática de conteúdo. Este poderá servir de base para a construção de um qualquer videojogo de plataformas, permitindo a definição de um modelo para os níveis e a respectiva edição dos mesmos, de forma manual ou automática.

O principal foco incidiu sobre a *framework* desenvolvida no âmbito deste projecto. Esta permite a representação do conteúdo de diversos jogos, tal como foi possível observar nos exemplos facultados. A maior limitação é, de facto, a acessibilidade ao conteúdo dos diversos jogos, o qual é tipicamente guardado em formatos proprietários. Para compreender melhor a capacidade de representação da referida *framework*, está prevista a representação do conteúdo de outros videogames, nomeadamente dos títulos *Super Tux* e *Secret Maryo Chronicles*, ambos projectos livres e em que existe acesso à estrutura dos níveis. Um dos aspectos interessante a abordar como trabalho futuro é o mapeamento nesta *framework* do conteúdo de jogos que não possuam qualquer restrição a uma grelha, identificando possíveis perdas de informação e eventuais alterações possíveis para que estas não aconteçam.

A aplicação desenvolvida assenta numa arquitectura fortemente modular, pelo que é possível expandir os resultados aqui apresentados. Como foi referido, pretende-se expandir o conjunto de jogos suportados pela aplicação, sendo este processo facilitado pela referida modularidade da aplicação. Do mesmo modo, está prevista a expansão do conjunto actual de mecanismos de geração automática, nomeadamente com suporte a recolha de dados sobre utilizações de jogadores. Deste modo, pretende-se potenciar o desenvolvimento de algoritmos mais orientados à personalização. Novamente, a arquitectura modular do programa facilita estes potenciais desenvolvimentos.

Um dos *plugins* mais completos actualmente implementados assiste a integração com o videogame *Prince of Persia*. Um aspecto interessante a constatar é que, para além de se conseguir representar os níveis na sua totalidade, existe uma maior expressividade da própria representação em relação ao jogo original. Por exemplo, a versão original do jogo divide o mapa em ecrãs isolados de 10 por 3 células, com um limite de 24 ecrãs e não permite a existência de mais do que um guarda por ecrã. Este tipo de limitações é comum em jogos antigos que, por escassez de memória, compactavam ao máximo a informação. Este tipo de limitação não existe dentro do sistema, pelo que, por exemplo, seria possível considerar mais do que um guarda por ecrã.

Finalmente, há a referir que se pretende lançar um projecto paralelo de desenvolvimento de um videogame original de plataformas que tire proveito da *framework* e das ferramentas aqui apresentadas.

6. AGRADECIMENTOS

O trabalho apresentado tem sido parcialmente subsidiado pelo Instituto Politécnico de Setúbal sob a bolsa FCT/MCTES com a designação SFRH/PROTEC/67497/2010 e pelo CITI sob a bolsa FCT/MCTES com a designação PEst-OE/EEI/UI0527/2011.

7. REFERÊNCIAS

[Compton06] Compton, K., Mateas, M. 2006. Procedural Level Design for Platform Games. *Proceedings of the 2nd Artificial Intelligence and Interactive Digital Entertainment Conference*.

[Jennings-Teats10] Jennings-Teats M., Smith G., Wardrip-Fruin, N. 2010. Polymorph: Dynamic Difficulty Adjustment Through Level Generation. *Proceedings of the Workshop on Procedural Content Generation in Games*.

<<http://doi.acm.org/10.1145/1814256.1814267>>

[Mawhorter10] Mawhorter, P., Mateas, M. 2010. Procedural Level Generation Using Occupancy-Regulated Extension. *CIG-2010 - IEEE Conference on Computational Intelligence and Games*.

[Mourato11] Mourato, F., Próspero dos Santos, M., Birra, F. 2011. Automatic level generation for platform videogames using Genetic Algorithms. *ACE 2011, 8th international conference on advances in computer entertainment technology*.

[Mourato12] Mourato, F., Birra, F., Próspero dos Santos, M. 2012. Enhancing level difficulty and additional content in platform videogames through graph analysis. *ACE 2012, 9th international conference on advances in computer entertainment technology*.

[Pederson09] Pedersen, C., Togelius, J., Yannakakis, G. 2009. Modeling player experience in super marios. In *Proceedings of the 5th international conference on Computational Intelligence and Games (CIG'09)*. IEEE Press.

[Shaker10] Shaker N., Togelius J., Yannakakis N., Weber B., Shimizu T., Hashiyama T., Soreson N., Pasquier P., Mawhorter P., Takahashi G., Smith G., Baumgarten R. 2010. The 2010 Mario AI Championship: Level Generation Track, *Special Issue of IEEE Transactions on Procedural Content Generation*.

[Smith08] Smith, G., Cha, M., Whitehead, J. 2008. A framework for analysis of 2D platformer levels. *Proceedings of the 2008 ACM SIGGRAPH Symposium on Video Games*. Sandbox '08. ACM
<<http://doi.acm.org/10.1145/1401843.1401858>>

[Smith09] Smith, G., Treanor, M., Whitehead, J., Mateas, M. 2009. Rhythm-based level generation for 2D platformers. *Proceedings of the 4th international Conference on Foundations of Digital Games*
<<http://doi.acm.org/10.1145/1536513.1536548>>

[Smith10a] Smith, G., Whitehead, J. 2010. Analyzing the Expressive Range of a Level Generator. In *Proceedings of the Workshop on PCG in Games*.

[Smith10b] Smith, G., Whitehead, J., Mateas, M. 2010. Tanagra: A Mixed-Initiative Level Design Tool. *Proceedings of the 2010 International Conference on the Foundations of Digital Games*.

[Smith11] Smith G., Whitehead J., Mateas M., Treanor M., March J., Cha M.. 2011. Launchpad: A Rhythm-Based Level Generator for 2D Platformers. *IEEE Transactions on Computational Intelligence and AI in Games* vol. 3, issue 1.

[Viglietta12] Viglietta G. 2012. Gaming is a hard job, but someone has to do it. *6th International Conference on Fun with Algorithms*

Identificação de Objetos em Imagens tomográficas através de GPGPUs

Bruno Preto Fernando Birra Pedro Medeiros
CITI, Departamento de Informática
Faculdade de Ciências e Tecnologia
Universidade Nova de Lisboa
bpreto@gmail.com, {fpb,pdm}@fct.unl.pt

Resumo

Neste artigo encontra-se a implementação e a avaliação de um novo algoritmo híbrido (CPU-GPU) para a identificação de estruturas conexas em volumes de dados tridimensionais. Este algoritmo explora o paralelismo tanto ao nível do CPU como dos GPGPUs. Contudo, o processamento é maioritariamente realizado em GPUs.

A motivação para este algoritmo resulta da sua utilização no contexto dum sistema mais vasto e orientado para a resolução de problemas de caracterização estrutural de materiais através de tomografia. Através da solução presente no artigo, será possível analisar a localização e a morfologia dos objetos presentes nos volumes tridimensionais.

A grande vantagem da utilização deste algoritmo deve-se a permitir tempos de execução bastante baixos, bem como à capacidade de processar grandes volumes de dados. Neste caso, as soluções parciais são calculadas de forma independente numa partição dos dados iniciais, sendo posteriormente integradas pelo CPU, usando uma abordagem que permite, ainda assim, explorar o paralelismo oferecido pelos múltiplos cores CPU.

Palavras-Chave

Algoritmos de processamento de imagens a três dimensões; Paralelização de aplicações; OpenCL; Tomografia; GPGPU.

1 INTRODUÇÃO

Os algoritmos do tipo *union—find* lidam com o problema de encontrar conjuntos disjuntos de objetos e as suas aplicações abrangem muitas áreas, tais como as redes de computadores, visão por computador e tomografia computadorizada. Os objetos a identificar consistem em elementos conexas e as operações elementares neste tipo de algoritmos estão na base da sua nomenclatura. A operação *union* funde dois conjuntos disjuntos de elementos, interligando-os, enquanto a operação *find* descobre se dois dados elementos estão interligados.

Quando os objetos a identificar correspondem a pixéis de uma imagem ou amostras de um volume, sendo os dados de entrada substituídos pela etiqueta do objeto a que pertencem, os algoritmos designam-se por *Connected-component labeling*. No caso das imagens 2D ou 3D, a conectividade de cada elemento (pixel ou voxel) pode ser definida ao longo das faces, das arestas e dos vértices.

A Figura 1 apresenta um possível resultado aplicado a uma imagem 2D, sendo a conectividade dos pixéis definida ao longo das arestas.

Numa fase inicial, os dados provenientes das amostras são filtrados, sendo submetidos, na maior parte dos casos a processos de remoção de ruído, erosão, dilatação, etc. Um

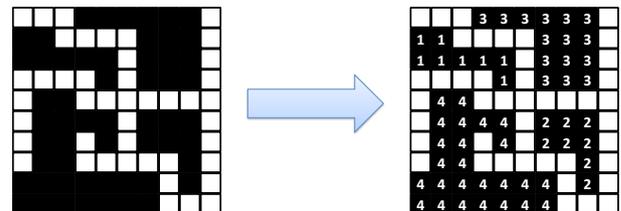


Figura 1: Algoritmo *Connected-component labeling*.

dos processos mais importantes nesse pré-processamento, é o da segmentação. É após a segmentação que se efetua a identificação de aglomerados conexas, para os quais cada amostra possui o mesmo valor após a segmentação.

Após a etiquetagem de cada objeto com um identificador único torna-se possível obter todos os pixéis ou voxéis do mesmo, para análise posterior, permitindo assim, não só visualizar esse mesmo objeto de forma isolada, mas também obter informação que permita ajudar à sua caracterização (área, volume, massa, fatores de forma, eixos principais, etc.).

Este trabalho incide num módulo dum sistema mais vasto do tipo PSE (*Problem Solver Environment*), desenvolvido no topo do sistema de visualização de dados ci-

entíficos SCIRun [SCI12]. A componente em causa efetua a identificação de objetos em volumes de dados respeitantes a amostras de materiais compósitos obtidos por processos de microtomografia, contribuindo assim para uma posterior caracterização das amostras analisadas.

2 TRABALHO RELACIONADO

Atualmente, existem diversas soluções para a identificação de objetos, as quais podem ser agrupadas em dois grupos, as soluções puramente sequenciais, descritas em [Wu 09], e as soluções que exploram o paralelismo, utilizando *clusters* [Harrison 11], e GPGPUs [Hawick 10] [OS11].

2.1 Soluções sequenciais

As soluções que processam sequencialmente os volumes de dados, apresentadas em [Wu 09], consistem em soluções lineares no tempo de resposta. Estas podem ser agrupadas em três grupos de algoritmos, *multi-pass*, *two-pass* e *one-pass*, segundo o número de passagens que efetuam ao volume de dados.

Os algoritmos *multi-pass* realizam diversas passagens ao volume de dados até se obter a solução. O algoritmo mais conhecido desta classe é descrito em Suzuki et al. [Suzuki 03], e efetua até quatro passagens no volume de dados, recorrendo a uma tabela de conectividade entre identificadores para reduzir o número de passagens.

Relativamente à classe de algoritmos *two-pass*, estes efetuam duas passagens ao volume de dados, acedendo apenas a posições contíguas de memória. Este algoritmo utiliza o endereçamento do volume de dados para atribuir inicialmente identificadores únicos aos voxéis. Para efetuar a junção de identificadores, é utilizada a estrutura de dados *Disjoint-Set Forests*.

Estas soluções têm a grande desvantagem de proporcionarem tempos de resposta bastante elevados, e de apenas poderem ser aplicadas a conjuntos de dados limitados.

2.2 Soluções paralelas

Relativamente às soluções que exploram o paralelismo, estas permitem reduzir o tempo de resposta, decompondo o processamento em diversas tarefas, que podem ser processadas em paralelo por múltiplas unidades de processamento. O grande problema desta abordagem, reside na complexidade da divisão do processamento. Este problema não é passível de ser resolvido sem a necessidade de sincronismo, dado que os valores a atribuir a uma dada região, necessitam de estar coerentes com a sua vizinhança. No que se refere às soluções utilizando *clusters*, destaca-se a solução descrita em [Harrison 11], que consiste em dividir o volume de dados por diversos nós, que processam um subconjunto do volume de dados original. Durante esse processamento existem pontos de sincronização de forma a obter informações e a conectividade entre os subconjuntos processados em nós distintos. Esta solução tem a vantagem de suportar grandes volumes de dados, sendo estes posteriormente decompostos e atribuídos aos nós disponíveis. O problema desta abordagem reside na relação custo/escalabilidade, dada a neces-

sidade de adicionar novos nós ao *cluster* para aumentar a capacidade computacional. Embora a escalabilidade seja um problema, a principal limitação da solução reside essencialmente nas latências da comunicação entre os nós, que tornam a interatividade limitada.

No que diz respeito à solução CCL [OS11], esta consiste em efetuar todo o processamento no GPGPU, tirando partido da memória e da localização da informação.

Este algoritmo apresenta enormes vantagens face aos restantes, no que concerne à redução dos tempos de execução. Contudo, este apenas efetua o processamento de imagens bidimensionais, não permitindo assim a deteção dos objetos tridimensionais. Além disso, apenas processa imagens de pequenas dimensões.

De seguida, apresenta-se a solução desenvolvida, que combina as duas soluções já existentes de forma a realizar o processamento de imagens tridimensionais através de GPGPUs com tempos de resposta adequados a um ambiente interativo.

3 DESCRIÇÃO DA SOLUÇÃO

O presente algoritmo foi desenvolvido para ser executado em máquinas com um CPU, contendo um ou vários núcleos, auxiliados por um conjunto de um ou mais GPGPUs, sendo nestes últimos que se realiza a maioria do processamento.

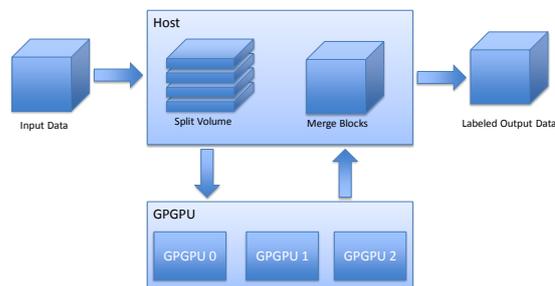


Figura 2: Arquitetura da solução.

Tal como se pode observar na Figura 2, esta solução recebe um volume de dados, o qual vai sendo dividido em vários blocos, posteriormente enviados e processados por um GPGPU numa pool de GPGPUs. Após o seu processamento, são enviados para o CPU, onde os resultados são consolidados no volume final de saída.

Este algoritmo explora o paralelismo oferecido pelos GPGPUs, visando a identificação de objetos em volumes de grande dimensão (aprox. 1 Giga amostras), em tempos de execução que permitam alguma interatividade.

Tal como é possível observar na Figura 3, este algoritmo encontra-se dividido nas seguintes quatro fases:

- decomposição do volume de dados em blocos;
- atribuição de identificadores iniciais aos voxéis de cada bloco (*kernel 1*);
- fusão de identificadores em subobjetos (*kernel 2*);

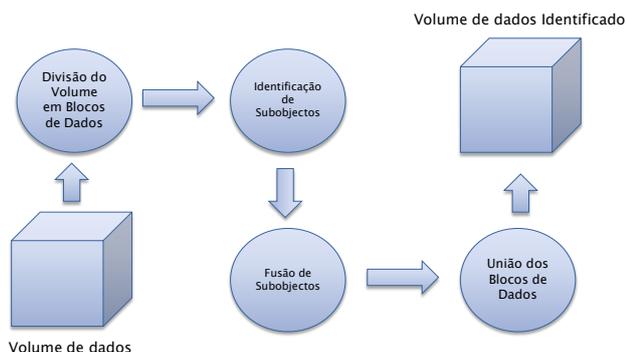


Figura 3: Fases do Algoritmo.

- fusão dos subobjetos entre blocos distintos,

as quais serão analisadas de seguida.

3.1 Decomposição do volume em blocos

Dada a limitação no tamanho da memória dos GPGPUs, o algoritmo começa por dividir o conjunto de dados em blocos, que serão posteriormente processados isoladamente pelos GPGPUs. Cada GPGPU vai requerendo blocos assincronamente, numa fila de trabalho, à medida que termina o processamento do bloco a ele anteriormente atribuído. Esta estratégia permite maximizar a utilização dos dispositivos, visto que estes não ficam inativos, à espera que outras unidades terminem o seu processamento.

3.2 Atribuição inicial de identificadores

O objetivo desta fase é o de proceder a uma identificação prévia de possíveis objetos, focando a atenção em extremos dos mesmos. Por exemplo, efectuando uma atribuição inicial de identificadores aos voxéis que se situam nas extremidades dos objetos e propagando de seguida esses identificadores pelos voxéis vizinhos, enquanto tal for possível.

Após a decomposição do volume de dados em blocos, cada voxel é mapeado num *thread*, que começa por poder atribuir identificadores únicos (recorrendo a contadores implementados com instruções atómicas), a um conjunto reduzido de voxéis, segundo a sua posição relativa nos objetos a que pertencem (p.ex. nos seus cantos). Esse conjunto de voxéis é calculado através da sua posição, caso um voxel possua o vizinho acima e à esquerda com a cor branca (vazio), então pede-se um identificador único novo para a sua cor, e o *thread* termina.

Quanto aos restantes *threads*, estes entram em estado de espera ativa, no qual vão validando os identificadores dos vizinhos. Assim que um vizinho possui um identificador válido, ou seja, diferente de branco ou preto, o voxel adota esse valor e termina. Esta operação tem como vantagem não precisar de memória auxiliar para a propagação dos identificadores, como os algoritmos sequencias, bem como permitir um elevado nível de paralelismo, visto que é gerado um *thread* por cada voxel.

Para minimizar o número de acessos à memória global,

a propagação de identificadores é apenas realizada em memória local, tornando o processamento mais rápido.

O facto de se utilizar a memória local para efetuar a propagação dos identificadores, implica que sejam necessárias menos iterações, visto que esta tem dimensões bastante menores (usualmente utilizamos um layout de 8x8x8), que a memória global que contém as dimensões do bloco a processar. Como se pode observar na Figura 4, a propagação de um dado identificador é apenas realizada dentro de cada região local, ao invés de todo o bloco, encurtando assim o tempo da sua propagação.

Sendo o endereçamento dos *threads* igual ao dos voxéis na matriz, os acessos realizados à memória são coalescentes, tirando assim um maior partido da arquitetura dos dispositivos, no que à largura de banda da memória diz respeito.

3.3 Fusão de subobjetos

Como ilustrado na Figura 4, após a execução do primeiro *kernel*, todos os voxéis do volume de dados possuem regiões com identificadores atribuídos garantindo que, objetos disjuntos não partilham identificadores comuns e, para cada objeto há um excesso de identificadores, ainda a fundir num só. A tarefa dessa fusão, ao nível de cada bloco dos dados iniciais é da responsabilidade do *kernel 2*.

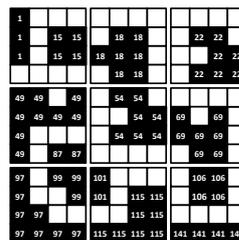


Figura 4: Volume de dados após a execução do primeiro *kernel*.

Este *kernel* é composto por três etapas. A primeira etapa consiste em mapear cada voxel num *thread*, validando de seguida todos os seus vizinhos. Caso algum possua um identificador inferior, este associa o seu identificador antigo ao identificador novo encontrado numa estrutura auxiliar (vetor de alterações), para que assim todos os voxéis com o valor antigo sejam alterados posteriormente.

A segunda etapa diz respeito à otimização do vetor de alterações, a qual analisa a transitividade entre identificadores, de forma a encontrar o identificador raiz. Esta operação consiste em cada *thread* consultar a posição do vetor de alterações correspondente, e caso esta esteja marcada para ser modificada por um novo identificador, o *thread* consulta a posição correspondente a essa identificação. Caso essa nova posição também se encontre marcada, o *thread* volta a repetir a operação, até encontrar a raiz da transitividade, correspondendo a uma entrada no vetor apontando para ela própria. Assim, após esta operação, todos os voxéis são modificados a sua raiz da árvore de equivalências.

A aplicação desta técnica pode ser observada na Figura 5,

onde o identificador 14 está marcado para ser substituído pelo 13, encontrando-se este marcado para ser substituído pelo 9. Assim, a otimização realizada modifica para 9 o valor pelo qual o identificador 14 irá ser substituído. Caso contrário, seria necessária uma nova iteração para atribuir o identificador 9 a esta região.

Na terceira etapa todos os voxéis são novamente mapeados num *thread*, utilizando o GPGPU, o qual altera o identificador corrente de cada voxel para o correspondente no vetor de alterações.

As fases supracitadas são repetidas até que todos os subobjetos do bloco sejam unidos, ou seja, até que não exista nenhuma modificação a efetuar no vetor das alterações na primeira fase, tal como consta na Figura 5.

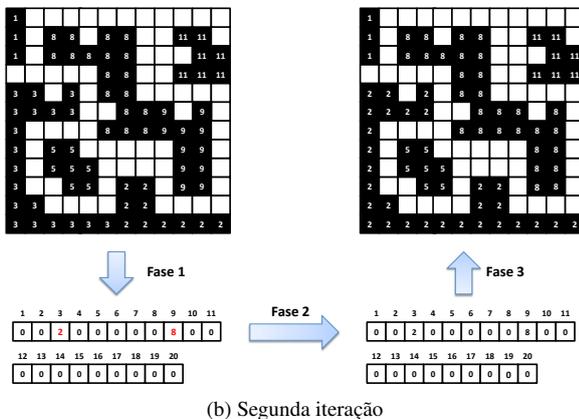
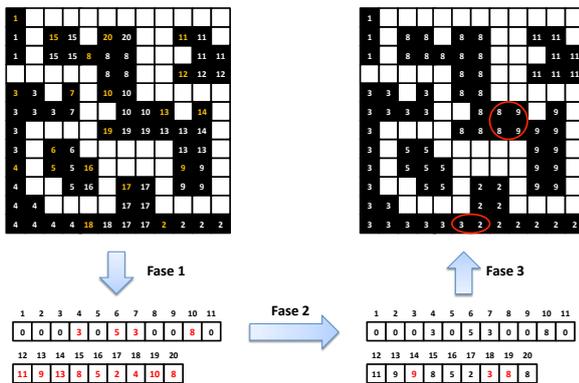


Figura 5: Exemplo de fusão de objetos em GPGPU.

4 Fusão de identificadores de blocos distintos

Uma vez que o volume de dados foi inicialmente decomposto em blocos, é necessário obter a conectividade entre os objetos presentes em blocos distintos. Para realizar esta operação é utilizado um grafo, onde cada objeto representa um vértice, e a conectividade uma aresta. Devido à elevada dimensão do volume, não é possível realizar esta operação de forma eficiente no GPU.

Esta operação é realizada em CPU, utilizando um grafo, onde um nó representa um identificador e as arestas a relação entre identificadores. Para que a geração do grafo seja realizada de forma eficiente, foram criados diversos

threads que assincronamente obtêm blocos e processam as suas fronteiras. Assim, dado que os identificadores são únicos globalmente em cada bloco, não existem acessos concorrentes ao mesmo vértice do grafo, visto serem apenas criadas arestas com origem em identificadores do bloco a processar.

Para finalizar, para se obterem os identificadores que pertencem a um mesmo objeto, é realizada uma pesquisa em profundidade no grafo, que permite obter, de forma eficiente, listas de identificadores que relacionam os subobjetos do mesmo objeto.

5 OTIMIZAÇÕES

De forma a otimizar a solução foram implementadas algumas técnicas, adequadas à arquitetura utilizada [Nvidia 11], as quais visam maximizar a utilização dos GPGPUs, proporcionando tempos de resposta mais reduzidos.

Uma das otimizações realizadas diz respeito às transferências dos blocos de dados entre CPU e GPU. Estas, para que sejam realizadas de forma mais rápida, requerem que os *buffers* em memória RAM sejam criados com o tipo *page-locked*, permanecendo fixos em RAM durante a sua utilização. Esta característica permite que se consigam atingir taxas de transferência na ordem dos 5 GBps.

Uma outra otimização bastante relevante, diz respeito ao paralelismo entre transferências e execuções para o mesmo dispositivo. Para que esta situação seja possível, para além dos *buffers* em RAM terem que ser do tipo *page-locked*, devem ainda ser utilizadas duas filas de comandos (*queues*) por dispositivo.

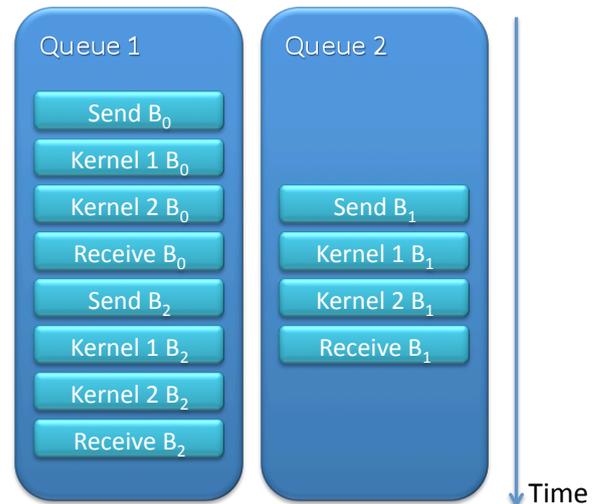


Figura 6: Ordem de envio dos comandos para o GPGPU.

Tal como se pode observar na Figura 6, a solução desenvolvida atribui duas filas de comandos para um mesmo dispositivo, sendo as operações de transferência e execução distribuídas pelas filas de modo a permitir a execução de *kernels* em simultâneo com as transferências de dados. A partição dos comandos pelas duas filas depende do problema em concreto mas, através desta técnica, foi possível

reduzir avultadamente o tempo de execução, conseguindo ocultar a maior parte das transferências de memória com o dispositivo, ficando estas escondidas pelas execuções dos *kernels*.

Além das técnicas supracitadas, foram realizadas as seguintes otimizações, já naturais: os *threads* encontram-se agrupados em múltiplos dos *warps*, de forma a maximizar o número de *threads* ativos; os dados são mantidos na memória do GPU durante todas as fases, de forma a reduzir o número de transferências; os *buffers* são reutilizados, de forma a evitar operações de alocação de memória.

6 ANÁLISE DE RESULTADOS

Para analisar a solução foram realizados testes a diversos níveis, de forma a extrair as suas vantagens e limitações. A avaliação consistiu em analisar os tempos de resposta, assim como alguns aspetos inerentes à implementação. De seguida, encontra-se a descrição: do *hardware* e *software* utilizados; dos volumes de dados utilizados; dos resultados obtidos.

6.1 Hardware e Software utilizados

Para a realização dos testes, foi utilizada uma estação de trabalho que possui como *hardware* base: um processador Xeon E5504 (4-core); 12 Gbytes RAM; um GPU nVidia Quadro FX 3800 para a visualização e um nVidia c2050 (Fermi) para o processamento. Este acelerador, que consiste num GPGPU com 448 cores CUDA, com 3 Gbytes de memória, sendo a sua capacidade de processamento de 1 Tflop em precisão simples e 515 Gflops em precisão dupla, efetuando um consumo de 238 W.

Relativamente ao software utilizado, este consistiu na plataforma OpenCL sobre o sistema operativo Linux na distribuição Ubuntu.

6.2 Volumes de dados utilizados

Os volumes de dados utilizados, ilustrados na Figura 7, consistem em dados fictícios, que têm com principal função simular cenários extremos, mais exigentes do que os volumes reais, obtidos através de micro tomografia. O primeiro volume de dados, designado de *chess* consiste num xadrez tridimensional, que tem como finalidade criar um elevado número de objetos de pequenas dimensões. O segundo volume de dados, designado de *blocks*, consiste em diversos paralelepípedos que se estendem em toda a dimensão z, analisando assim o comportamento dos algoritmos em volumes com bastantes objetos de grandes dimensões. O terceiro volume, designado de *spiral*, consiste num único objeto que o ocupa por completo, sendo a sua forma em espiral, de modo a maximizar a sua dimensão. Relativamente ao quarto volume de dados, designado de *snake*, consiste num único objeto, que percorre todo o volume de dados, possuindo apenas a espessura de um voxel.

6.3 Resultados obtidos

No gráfico ilustrado na Figura 8 é possível visualizar o tempo de transferência dos blocos de dados em função do volume de dados e do tipo de transferência. Nesta análise

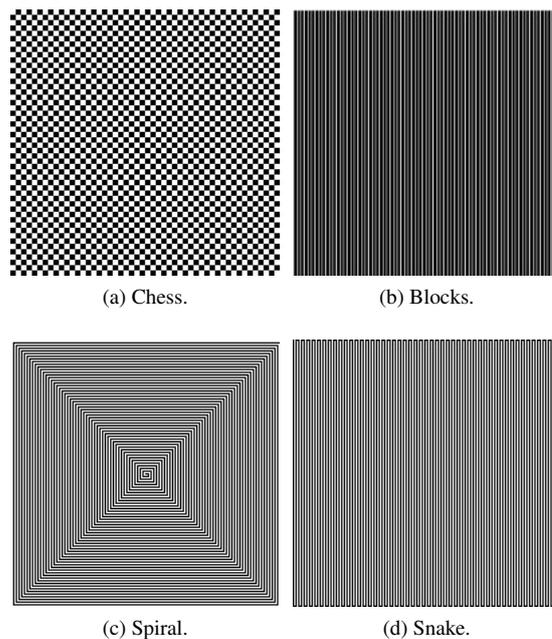


Figura 7: Volumes de dados.

quantitativa pode-se observar o tempo despendido no envio e na receção dos dados, podendo ver-se a sua variação em função da dimensão dos dados.

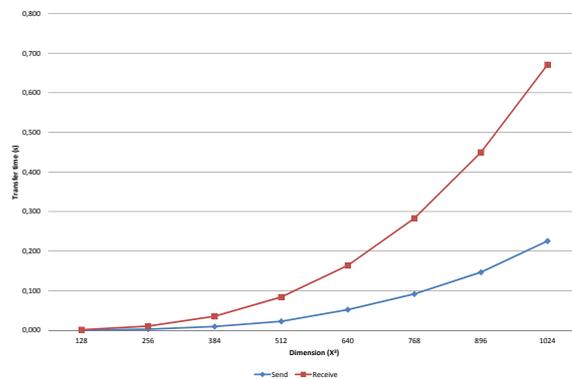


Figura 8: Tempos de transferência entre CPU e GPGPU.

Os tempos associados à emissão dos blocos para o GPGPU são inferiores à receção, visto que o volume de dados enviado é 4x menor. Este utiliza, para cada voxel, um byte, enquanto que o volume de dados recebido utiliza quatro bytes para representar cada identificador.

Relativamente ao processamento em GPGPU, através do gráfico da Figura 9, é possível observar o tempo de execução dos dois *kernels* em função da dimensão do volume de dados *chess*, ilustrado na figura 7a. Através deste gráfico é possível visualizar diversos aspetos. O primeiro diz respeito às diferenças entre o primeiro e o segundo *kernel*. O primeiro *kernel* começa por apresentar tempos superiores aos do segundo *kernel*, devido à complexidade do mesmo. Este primeiro *kernel*, como já foi referido, coloca diversos *threads* em espera ativa até que os identificadores

sejam propagados pelos vizinhos, ao contrário do segundo *kernel*, em que cada thread executa um processamento de complexidade constante.

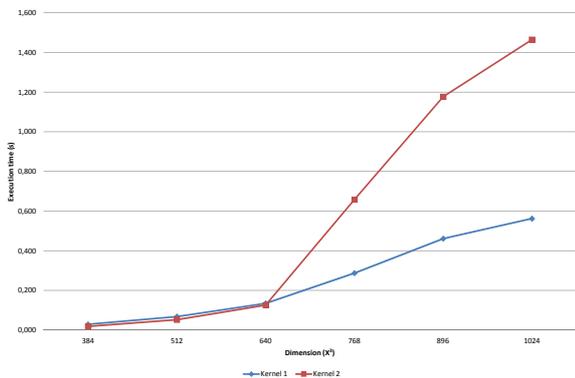


Figura 9: Tempos de processamento em GPGPU.

Através deste gráfico é possível verificar que quando a dimensão do volume de dados é superior a 640x640x640, o segundo *kernel* começa a exibir tempos superiores aos do primeiro. Esse facto deve-se à quantidade de iterações que é necessário executar para o segundo *kernel*, de modo a que os subobjetos sejam todos fundidos.

Um outro aspecto importante de analisar, é a fusão de blocos em CPU. Esta operação apenas é necessária quando o volume de dados é superior à memória do GPGPU. Contudo, o seu tempo de processamento é bastante baixo, visto que é uma operação realizada em paralelo, e apenas são validadas as fronteiras de cada bloco.

Dada a limitação da memória do GPGPU utilizado, bem como a limitação do OpenCL, relativa ao montante máximo de memória alocável para um *buffer*, foi necessário efetuar uma divisão do volume de dados em blocos de dados com a dimensão máxima de 128 MB. Assim sendo, um volume de 1024x1024x1024, é dividido em oito blocos.

Através da Tabela 1, é possível verificar os tempos de execução obtidos para cada um dos volumes de dados, com dimensão de 1024x1024x1024. Na referida tabela pode-se observar o tempo despendido na criação e na pesquisa em profundidade

Volume	Criação		Pesquisa		
	\bar{x} (ms)	σ (ms)	%	\bar{x} (ms)	σ (ms)
Chess	556.17	3.21	11,35	239.78	0.50
Blocks	111.29	3.09	1,88	11.00	0.26
Spiral	67.14	0.99	0,91	0.07	0.00
Snake	8.49	0.04	0,13	0.06	0.00

Tabela 1: Tempo de processamento na união dos blocos.

A diferença nos tempos de execução dos diferentes volumes de dados, diz respeito à quantidade de identificadores presentes no volume, que fazem com que o grafo resultante do processamento em CPU seja maior ou menor. Esse facto é bastante importante para determinar o tempo

de processamento necessário para a pesquisa em profundidade, realizada sequencialmente.

No que se refere ao tempo total despendido pela solução, este pode ser visualizado no gráfico da Figura 10. Estes valores foram obtidos através da média dos tempos de execução de cada um dos volumes. Como se pode verificar, esta solução possui tempos de execução baixos mesmo quando se aumenta o volume de dados a processar.

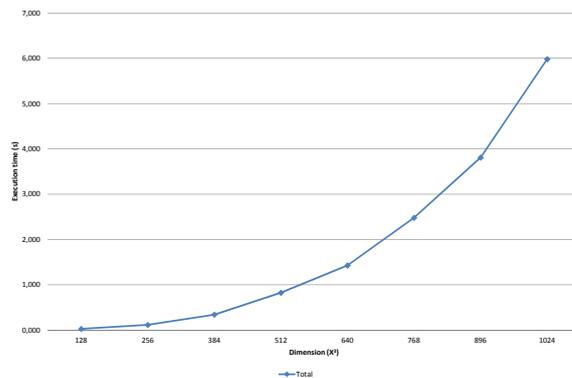


Figura 10: Tempos de execução da solução.

Para descrever cada componente da solução, o gráfico da Figura 11 apresenta a percentagem do tempo total de execução de cada componente. Uma vez mais, refere-se aos valores médios para os quatro volumes de teste. Como se pode verificar, o processamento predomina. Esse facto deve-se à permanência dos dados na memória do GPU enquanto são necessários, bem como às transferências do tipo *pinned*. Neste gráfico pode-se constatar que a componente realizada pelo CPU é bastante baixa.

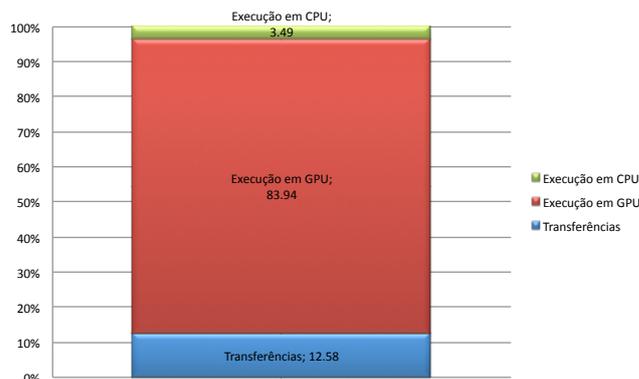


Figura 11: Decomposição do tempo total gasto pela solução.

Em suma, a solução permite o processamento de grandes volumes de dados com pequenos tempos de execução, permitindo a sua utilização em ambientes interativos.

7 CONCLUSÕES

Tal como já foi referido no presente artigo, a solução desenvolvida tem a capacidade de processar volumes de dados de dimensões elevadas, em curtos espaços de

tempo. Sendo assim uma mais-valia no que concerne à identificação de objetos. Através dos tempos de execução obtidos nesta solução, é possível concluir que esta se adequa a ambientes interativos, visto que até ao momento, a mesma era realizada *offline*.

A solução desenvolvida, além da importante aplicação que tem, no âmbito do projeto na qual foi desenvolvida, apresenta também diversas outras aplicações, como na área da saúde, para a deteção de massas em determinadas áreas do corpo humano, bem como na identificação de padrões em imagens.

Esta solução contém algumas limitações, as quais se encontram em estudo, para melhorar o desempenho do algoritmo. Uma dessas limitações diz respeito às interações necessárias realizar pelo *kernel 2*, as quais acrescem *overheads* no tempo total de execução. Uma outra limitação diz respeito à fusão dos blocos realizada em CPU. Embora este usualmente não cause grande impacto no tempo total da solução, em volumes com uma grande quantidade de objetos, os tempos de processamento desta fase começam a ser significativos.

Para além das limitações supracitadas, existem ainda alguns aspetos que se encontram em estudo para melhorar o algoritmo. Um desses aspetos diz respeito à utilização de múltiplos GPGPUs. Atualmente a solução já tem suporte para distribuir os blocos pelos múltiplos dispositivos, contudo, dada a falta de *hardware* com características semelhantes ao nVidia c2050, não foi possível analisar o desempenho da solução corretamente, bem como desenvolver otimizações para a combinação dos mesmos. Uma outra possível melhoria ao algoritmo, seria a possibilidade de identificar objetos provenientes de diversos intervalos de segmentação, os quais são representados com diferentes tons de cinzento.

8. AGRADECIMENTOS

Este trabalho foi parcialmente suportado por FCT/MCTES, projecto Ambiente de Resolução de Problemas para Caracterização Estrutural de Materiais por Tomografia - PTDC/EIA-EIA/102579/2008 e pelo CITI - PEst-OE/EEI/UI0527/2011.

9. REFERÊNCIAS

- [Harrison 11] Cyrus Harrison, Hank Childs, and Kelly P. Gaither. Data-parallel mesh connected components labelling and analysis. In Torsten Kuhlen, Renato Pajarola, and Kun Zhou, editors, *EGPGV*, pages 131–140. Eurographics Association, 2011.
- [Hawick 10] K. A. Hawick, A. Leist, and D. P. Playne. Parallel graph component labelling with gpus and cuda. *Parallel Comput.*, 36(12):655–678, December 2010.
- [Nvidia 11] C. Nvidia. Nvidia opencl programming guide, 2011.
- [OS11] Bedrich Benes Ondrej Stava. Connected component labelling in CUDA. In W.W. Hwu, editor, *GPU Computing Gems Emerald Edition*, pages 569–581. Morgan Kaufmann, 2011.
- [SCI12] Scirun: A scientific computing problem solving environment, scientific computing and imaging institute (sci). <http://www.scirun.org>, 2012.

[Suzuki 03] K. Suzuki, I. Horiba, and N. Sugie. Linear-time connected-component labeling based on sequential local operations. *Computer Vision and Image Understanding*, 89(1):1–23, 2003.

[Wu 09] K. Wu, E. Otoo, and K. Suzuki. Optimizing two-pass connected-component labeling algorithms. *Pattern Analysis & Applications*, 12(2):117–135, 2009.

A Apêndice

Algoritmo 1 Kernel 1.

```

__global unsigned char imageIn3D[]; // Imagem 3D original.
__global unsigned int imageOut3D[]; // Imagem 3D etiquetada.
__private Point3D globalID ← globalAddress(threadId);
__private Point3D localID ← localAddress(threadId);
__local unsigned int sharedMem[]; // Memória partilhada.
if imageIn3D[globalID.index] = BLACK then
    sharedMem[localID.index] ← globalID.index + 1;
else
    sharedMem[localID.index] ← 0;
end if
barrier(CLK_LOCAL_MEM_FENCE);
__local unsigned char isModify;
__private unsigned int currValue ← sharedMem[localID.index];
__private unsigned int newValue ← currValue;
while true do
    if localID.x > 0 then
        newValue ← sharedMem[localAddress(localID.x - 1, localID.y, localID.z)];
        if newValue > currValue then
            isModify ← true;
            currValue ← newValue;
        end if
    end if
    (...) // Expansão para  $x + 1, y - 1, y + 1, z - 1$  e  $z + 1$ .
    barrier(CLK_LOCAL_MEM_FENCE);
    if isModify = true then
        newValue ← currValue;
        while newValue ≠ sharedMem[newValue - 1] do
            newValue ← sharedMem[newValue - 1];
        end while
        if newValue ≠ 0 then
            sharedMem[localID.index][newValue - 1] ← newValue;
        else
            sharedMem[localID.index][newValue - 1] ← currValue;
        end if
    end if
    else
        break;
    end if
    barrier(CLK_LOCAL_MEM_FENCE);
    isModify ← false;
end while
imageOut3D[globalID.index] ← sharedMem[localID.index];

```

Algoritmo 2 Kernel 2

```

__global boolean isDone ← false; // Afectada a true caso ocorra alterações.
__global unsigned int image3D[]; // Imagem 3D etiquetada.
__global unsigned int changes[]; // Vector de alterações.

__private Point3D globalID ← globalAddress(threadId);
__private Point3D localID ← localAddress(threadId);
__private unsigned int oldValue ← sharedMem[localID.index];
__private unsigned int newValue ← oldValue;
__private unsigned int newId ← BLACK;
if (globalID.x > 0) and (localID.x = 0) then
    newId ← image3D[globalAddress(globalID.x - 1, globalID.y, globalID.z)];
    if (newId ≠ WHITE) and (newValue > newId) then
        newValue ← newId;
    end if
end if
    (...) // Expansão para  $x + 1, y - 1, y + 1, z - 1$  e  $z + 1$ .
if newValue ≠ oldValue then
    isDone ← false;
    changes[oldValue] ← newValue;
end if
barrier(CLK_LOCAL_MEM_FENCE);
__private unsigned int newId ← changes[globalID.index + 1];
while (changes[newID] ≠ WHITE) and (change[newID] < newID) do
    newID ← change[newID];
end while
if (newId ≠ WHITE) and (currId < newId) then
    image3D[globalID.index] ← newId;
end if

```

Visualização em Tempo Real de um Modelo Esparso de Mistura Paramétrica para síntese da BTF

Nuno Silva *

Centro de Computação Gráfica
Campus Azurém, Guimarães
nuno.silva@ccg.pt

Luís Paulo Santos

Dep. Informática, Universidade do Minho
Campus Gualtar, Braga
psantos@di.uminho.pt

Resumo

As funções de textura bidirecionais (*Bidirectional Texture Functions* - BTF) permitem visualizações de alta qualidade de materiais reais, que exibem detalhes complexos na sua aparência, e que não podem ser fielmente representados por funções paramétricas mais simples. Representações fiéis deste tipo de materiais requerem grandes quantidades de dados, dificultando a sua visualização em tempo real. A compressão de BTFs constitui um compromisso entre qualidade visual e tempo de síntese. Este artigo apresenta um visualizador a correr integralmente no GPU, usando um motor de *ray tracing*, de uma representação recente para BTFs, o Modelo Esparso de Mistura Paramétrica (*Sparse Parametric Mixture Model* - SPMM). A escalabilidade com o número de BTFs e o número de luzes é também estudado.

Palavras-Chave

Materiais complexos, BTF, Visualização em tempo real, GPU, Ray tracing

1 INTRODUÇÃO

Representações digitais de materiais complexos têm um grande impacto na indústria têxtil e do calçado, auxiliando *designers* e artistas na prototipagem de novos produtos, e fornecendo visualizações realistas destes produtos ao consumidor final. Para o *designer*, a utilidade destas ferramentas depende tanto da qualidade da representação do material, como da sua visualização interativa.

A aparência do material depende da forma como o fluxo radiante interage com a superfície, e varia de acordo com as direções de incidência e de reflexão, entre outros fatores [Weyrich08].

É comum usar a *Bidirectional Reflectance Distribution Function* (BRDF), para modelar a aparência de um material, mas estas funções não permitem simular vários fenómenos complexos, tais como auto-oclusão, translucência ou inter-reflexões. Para esse efeito, a *Bidirectional Texture Function* (BTF) [Dana99] está a tornar-se popular devido ao elevado realismo permitido, melhorias nos sistemas de captura e aumento do poder computacional dos GPUs [Weyrich08, Müller05, Filip09].

A BTF é uma função 6D que modela a aparência de um material capturando várias imagens deste para várias direções de iluminação e de observação. As imagens são armazenadas em grandes tabelas, e o processo de síntese

envolve simples acessos às mesmas. Uma única BTF pode consumir vários *gigabytes* de armazenamento, inviabilizando a visualização em tempo real. A BTF tem que ser transformada numa representação mais compacta, que permita a respetiva visualização eficiente, sem grande perda de fidelidade para com a aparência real do material [Müller03, Sattler03, Ma04].

Wu et al. [Wu11] apresentaram uma nova representação para a BTF, o modelo esparso de mistura paramétrica (SPMM). Esta abordagem foi demonstrada pelos autores usando um *ray tracer* paralelo implementado em CPU que, embora tenham obtido visualizações de alta qualidade, não permite taxas de visualização interativas.

Este artigo apresenta uma implementação da SPMM, usando o GPU com o *ray tracer* Optix [Parker10], que obtém taxas de visualização interativas, sem comprometer a qualidade visual. O tempo de síntese depende do número de pontos intersetados que mapeiam numa SPMM, o número de fontes de luz e o número de SPMMs; a escalabilidade é também estudada conforme estes parâmetros.

O visualizador permite mapear SPMMs nos materiais definidos em modelos geométricos, e visualizá-los sob iluminação direta e especular, com uma ou mais luzes pontuais. Adicionalmente a contribuição da iluminação difusa indireta é aproximada por oclusão de ambiente. Este programa está atualmente a ser usado por parceiros na indústria do calçado Português, envolvidos num projeto de financiamento nacional.

*Trabalho parcialmente financiado pelo projeto QREN n° 13114 TOPIC Shoe e por Fundos Nacionais através da FCT - Fundação para a Ciência e a Tecnologia no projeto PEst-OE/EEI/UI0752/2011

O artigo está organizado da seguinte maneira: primeiro é apresentado o estado da arte da representação de materiais complexos, juntamente com o formato SPMM; seguidamente é descrita a nossa abordagem para obter taxas de visualização interativas no GPU usando Optix, juntamente com os resultados obtidos e respetiva análise. O artigo termina com conclusões e trabalho futuro.

2 MODELOS DE APARÊNCIA E VISUALIZAÇÃO

A maioria dos materiais do mundo real têm uma aparência complexa que pode ser descrita em três níveis [Suykens03, Müller05]: ao nível macroscópico está a geometria do objeto, tradicionalmente modelada com representações explícitas, tais como malhas poligonais. O nível microscópico engloba as interações luminosas num determinado ponto da superfície do objeto, que podem ser representadas com BRDFs. Finalmente, o nível mesoscópico apresenta-se entre estes dois níveis, e é composto pelos vários fenómenos de iluminação mais subtis, como a auto-occlusão, inter-reflexões ou translucência. Para estes fenómenos a BRDF é insuficiente, sendo geralmente substituída por técnicas baseadas em imagens, como o mapeamento de texturas.

Spatially-Varying BRDFs (SVBRDF) [McAllister02], podem ser vistas como uma combinação da BRDF com o mapeamento de texturas, permitindo representar materiais com diferentes BRDFs ao longo da sua superfície. Embora representem alguns dos fenómenos ao nível mesoscópico, estas não conseguem capturar a auto-occlusão. A BTF apresenta-se assim como uma alternativa viável para capturar todos estes efeitos. A *Bidirectional Subsurface Scattering Reflectance Distribution Function* (BSSRDF) consegue modelar todos estes fenómenos de transporte de luz, mas esta função é demasiado complexa para ser usada no contexto de visualização interativa, e os sistemas de captura atuais apenas conseguem medir sub-conjuntos desta função [Weyrich08, Müller05].

Note-se que a aparência de um material também pode ser reproduzida seguindo uma abordagem procedimental, isto é, definindo algoritmos e ajustando os respetivos parâmetros até que o efeito desejado seja obtido [Ebert02]. No entanto, esta é uma tarefa morosa que necessita de mão de obra qualificada, e não consegue, por vezes, representar fielmente todos os materiais do mundo real. Abordagens baseadas em imagens são cada vez mais populares devido ao uso de câmeras para capturar a aparência, sem exigir mão de obra qualificada, constituindo portanto, uma alternativa adequada para projetos na indústria do calçado Português.

2.1 Bidirectional Texture Function

A BTF emergiu como uma alternativa que representa simultaneamente a escala mesoscópica e a microscópica. Esta função modela, para cada comprimento de onda, a aparência do material num ponto da superfície (x), para um par de direções de incidência e observação (ω_i, ω_o). Esta informação é obtida capturando várias imagens do material sob diversas direções de iluminação e de observação.

A BTF pode ser vista como um caso particular da SVBRDF, dado que se assume que a superfície capturada é planar [Lawrence06]. Conceitualmente, não existem grandes diferenças entre a BTF e a textura 2D, requerendo ambas uma fase de aquisição, representação e síntese.

Uma BTF de boa qualidade, tais como as existentes na base de dados de Bonn [Sattler03], contém 81×81 imagens para as direções de iluminação e de observação, cada uma com 256^2 *texels* para cada canal de cor RGB. Isto corresponde a cerca de 1.2GB de dados brutos para uma única BTF, sem incluir efeitos com alta gama dinâmica (HDR). De modo a obter taxas de visualização interativas em objetos mapeados com várias BTFs, é necessário comprimir estes dados, preservando as características mais relevantes da aparência do material. Os métodos de compressão devem também aproveitar a redundância existente nos dados de uma forma eficiente, e permitir uma rápida descompressão dos dados. Mais detalhes sobre o estado da arte na captura, representação e síntese de BTFs podem ser consultados em [Müller05] e [Filip09].

2.2 O Modelo Esparsos de Mistura Paramétrica

Na representação SPMM proposta por Wu et al. [Wu11], os dados capturados são analisados e ajustados a uma combinação linear de funções paramétricas, cada uma definida como uma BRDF rodada e pesada pelo coseno do ângulo definido pela normal e a direção de incidência. A equação 1 descreve tais funções, onde $f_j(k_j, \cdot)$ é um modelo analítico da BRDF, com parâmetros k_j . R é uma rotação que transforma um vetor para o sistema de coordenadas locais definido pela normal n_j .

$$\rho_j(\omega_i, \omega_o) = f_j(k_j, R(\omega_i), R(\omega_o))(n_j \cdot \omega_i). \quad (1)$$

Os dados originais no ponto x podem ser aproximados por uma soma pesada de m funções, cada uma com peso α_j , como ilustrado na equação 2 e na figura 1. Os detalhes de aparência mais subtis que não possam ser capturados pelas funções paramétricas, são armazenados numa função residual, ϵ_x , que é obtida subtraindo a combinação linear de funções paramétricas da BTF original.

$$\text{BTF}_x(\omega_i, \omega_o)(n_x \cdot \omega_i) = \sum_{j=1}^m \alpha_j \rho_j(\omega_i, \omega_o) + \epsilon_x(\omega_i, \omega_o). \quad (2)$$

Dado que ajustar cada *texel* da BTF é computacionalmente muito exigente, a coerência espacial dos dados é aproveitada com o agrupamento multi-nível dos *texels* (k-means clustering). O algoritmo completo de ajuste é aplicado apenas a alguns *texels* de cada grupo, sendo que as funções paramétricas resultantes são usadas como dicionário para acelerar o ajuste dos restantes *texels*.

A função residual ϵ_x é também calculada por grupo. É sugerido que ϵ_x possa ser melhorada por análise local dos componentes principais (LPCA), armazenando as funções base resultantes e respetivos coeficientes [Müller03]. O au-

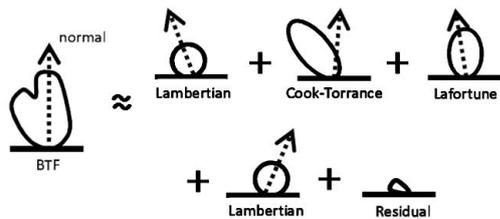


Figura 1. Ilustração da representação SPMM. A BTf é aproximada por uma soma pesada de modelos analíticos e uma parte residual. Cada modelo analítico tem o seu próprio sistema de coordenadas local. Adaptado de [Wu11].

mento do número de funções base permite melhorias marginais em troca de um maior consumo de memória. O visualizador apresentado permite a adaptação da qualidade de imagem limitando a parte residual a um sub-conjunto das suas funções base; assim, o tempo de síntese é adaptado às capacidades computacionais da placa gráfica.

A aproximação da BTf por uma soma de funções paramétricas conhecidas, fornece uma representação geral da aparência dos materiais, que reduz significativamente o volume de dados, permitindo uma síntese eficiente e, adicionalmente, a edição intuitiva dos seus parâmetros. Wu et al. testaram a SPMM com BTfs da base de dados de Bonn [Sattler03]; a nossa implementação é baseada nestas representações.

3 VISUALIZAÇÃO INTERATIVA DE SPMMs

O visualizador interativo usa o motor de *ray tracing* Optix da NVIDIA [Parker10], versão 2.5.1, e foi implementado em duas partes: a primeira corresponde à implementação base, simulando apenas a iluminação direta e raios especulares; a segunda aproxima fenómenos de iluminação global por oclusão de ambiente.

As experiências foram realizadas numa máquina equipada com um processador Intel 2.4GHz de quatro núcleos, 4GB de RAM e uma NVIDIA GeForce GTX 580, apresentando o resultado numa janela de 512×512 píxeis, com um raio primário por píxel. Todos os testes usam as representações SPMM das BTfs da base de dados de Bonn [Sattler03], que apresentam uma resolução espacial de 256×256 *texels* e uma resolução angular de 81×81 direções. Todos os testes apresentados usam o SPMM *Wool*, exceto quando indicado o contrário, e todos os coeficientes da função residual são avaliados, para maximizar a qualidade de imagem. Os valores indicados são obtidos da média de uma execução do programa durante sessenta segundos.

3.1 Implementação Base

O modelo de iluminação utilizado inclui apenas as componentes de iluminação direta e reflexões especulares. São simuladas fontes de luz pontuais, sendo a visibilidade de cada uma aferida com um raio sombra. As reflexões es-

peculares são calculadas disparando um raio ao longo da direção de reflexão ideal.

O uso eficiente do GPU requer que os dados do SPMM estejam armazenados na memória de textura da placa, como um array 1D, 2D ou 3D [Fernando04, Pharr05], permitindo assim acessos aleatórios rápidos. Desta forma, a memória de textura armazena, para cada texel, o identificador do grupo, os índices das funções paramétricas, os seus pesos α_j e os coeficientes da função residual. Para cada grupo armazenam-se as funções paramétricas ρ_j , os parâmetros correspondentes k_j e a função residual ϵ_x .

O formato SPMM não é apropriado para uma implementação no GPU pois consiste numa combinação linear de diferentes modelos analíticos, e traduzindo diretamente o código CPU para GPU resultaria em bastante divergência e num elevado número de ciclos. Consequentemente, a transformação das estruturas de dados, por forma a adaptar o código de síntese ao modelo de programação do GPU é uma tarefa fundamental.

As estruturas de dados utilizadas são um aspeto crucial pois determinam o consumo de memória e a largura de banda necessária; é importante minimizar estes dois fatores. Dado que as SPMMs geradas da base de dados de Bonn agrupam os *texels* em trinta e dois grupos, o identificador do grupo pode ser representado com um único *byte*. Outras estruturas que usam valores inteiros representam índices ou identificadores, e podem ser representados com dois *bytes*. As estruturas para os pesos das funções paramétricas, os seus parâmetros e a função residual são armazenadas no formato vírgula flutuante de precisão simples, com quatro *bytes* - o menor permitido em Optix.

O segundo aspeto a considerar é que o número de funções paramétricas m não é o mesmo para cada *texel*, e enviar esta informação dinamicamente para o GPU poderia comprometer o tempo de síntese. Uma abordagem possível seria calcular o número máximo de funções, uniformizando o número de funções avaliadas para cada *texel*, mas isto resultou num desempenho fraco. A nossa implementação pré-calcula m para cada *texel* e armazena o resultado numa estrutura adicional. Isto resulta em taxas de síntese mais elevadas, especialmente quando m varia muito de um *texel* para outro, isto é, quando a BTf apresenta variações locais mais complexas.

O terceiro problema advém da discretização esparsa das direções de captura da BTf. De modo a sintetizar a BTf para outras direções é necessário interpolar os dados das direções capturadas mais próximas. As funções paramétricas não são afetadas por este problema, dado que estas estão definidas para toda a semiesfera superior. No entanto, a função residual necessita de interpolação para evitar o aparecimento de artefactos quando mudam as direções capturadas mais próximas. Estas direções podem ser interpretadas como pontos 3D na superfície da semiesfera e projetadas no plano XY, ignorando a componente Z, resultando, assim, num conjunto de pontos 2D. Aplicando uma triangulação de Delaunay a esse conjunto de pontos, e armazenando os triângulos resultantes noutra es-

estrutura de dados, é possível interpolar intersecando um raio com um triângulo e usando as respetivas coordenadas baricêntricas [Pharr04]. Os pesos da interpolação, no caso do raio atravessar esse triângulo, são estas coordenadas. Isto é realizado separadamente para a direção de iluminação e de observação, para um total de nove pesos de interpolação.

Finalmente, as matrizes 2D cujas dimensões excedam o tamanho suportado pela placa são transformadas em fatias de uma matriz 3D, com as dimensões máximas suportadas pela placa gráfica. A figura 2 ilustra as estruturas de dados usadas (algumas das quais são apresentadas na seção seguinte), e o fluxo de dados necessário para sintetizar a SPMM para um ponto da superfície (x) e um par de direções (ω_i, ω_o).

3.2 Otimizações

A análise dos perfis de execução indicou que o maior limitador do desempenho era a interpolação da função residual. Dado que esta é baseada na interseção de um raio com todos os triângulos, e estes estão bem distribuídos ao longo de um círculo, usamos uma grelha regular para rapidamente descartar triângulos, limitando assim os testes de interseção a um subconjunto bastante reduzido dos mesmos. A grelha implementada é compacta, com requisitos de memória mínimos [Lagae08], construída no CPU e enviada para o GPU usando dois arrays: as estruturas *Grid Cells* e *Grid Data* no diagrama da figura 2.

A equação 3 é usada para o teste de interseção do raio com um triângulo, onde λ é o vetor de coordenadas baricêntricas, r é a direção do raio, e T é a matriz formada pelas coordenadas cartesianas dos vértices do triângulo. Este formato permite a precomputação da matriz inversa T^{-1} para cada triângulo, reduzindo o teste de interseção no GPU a uma subtração e uma multiplicação vetorial.

$$\begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{pmatrix} = T^{-1}(r - v_3); \quad (3)$$

$$\lambda_3 = 1 - \lambda_1 - \lambda_2.$$

Adicionalmente, de modo a maximizar a localidade dos dados, a matriz inversa, as coordenadas cartesianas de cada vértice e os correspondentes identificadores são armazenados numa mesma estrutura de dados, para um total de dezasseis valores de vírgula flutuante, por triângulo. Com estas duas otimizações o custo computacional da avaliação da função residual é aproximadamente o mesmo que da avaliação da combinação linear de funções paramétricas, enquanto que anteriormente, o primeiro era cerca de três vezes mais caro que o segundo.

De modo a tirar o máximo partido da largura de banda do GPU, são utilizadas as instruções SIMD para cálculo vetorial, e o acesso aos dados nas texturas é feito nos quatro canais disponíveis, RGBA, sempre que possível. Finalmente, durante o desenho da cena, o número de componentes resultantes da LPCA podem ser limitados pelo utilizador, de modo a adaptar o tempo de síntese às capacidades computacionais da máquina. Nas nossas experiências, esta opção

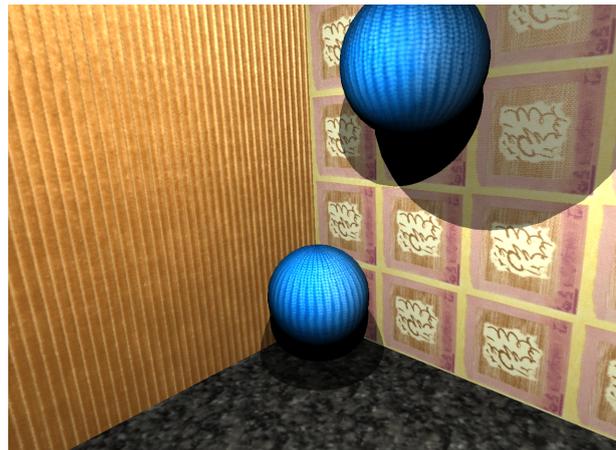


Figura 3. O interior de uma caixa (cena *Cornell Box*) mapeado com várias SPMMs. O material das esferas é *Wool*, e das paredes é *Corduroy*, *Wallpaper* e *Floor tile*. Com duas luzes pontuais obtém-se taxas de visualização de ≈ 50 FPS.

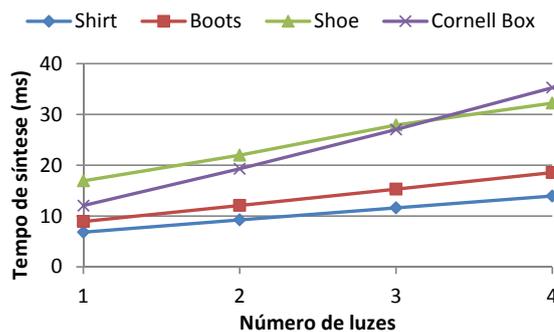


Figura 4. Escalabilidade com o número de luzes pontuais, em quatro modelos geométricos diferentes.

reduz bastante o tempo de síntese, tendo um impacto marginal na qualidade de imagem. A imagem 3 demonstra o resultado visual com iluminação direta, usando duas fontes de luz.

A escalabilidade do visualizador com a complexidade da cena é fulcral para o sucesso em ambiente de produção industrial. As medições de desempenho, tendo em conta a escalabilidade com o número de luzes e com o número de SPMMs são apresentadas nas figuras 4 e 5, respetivamente. Dado que o Optix fornece estruturas de aceleração eficientes, as variações de desempenho conforme a complexidade geométrica do modelo não se revelaram significativas.

O tempo de síntese aumenta linearmente com o número de luzes. Isto acontece devido ao cálculo da SPMM conforme o fluxo de dados no diagrama 2. No entanto, tal como indicado nesse diagrama, nem todas as estruturas de dados

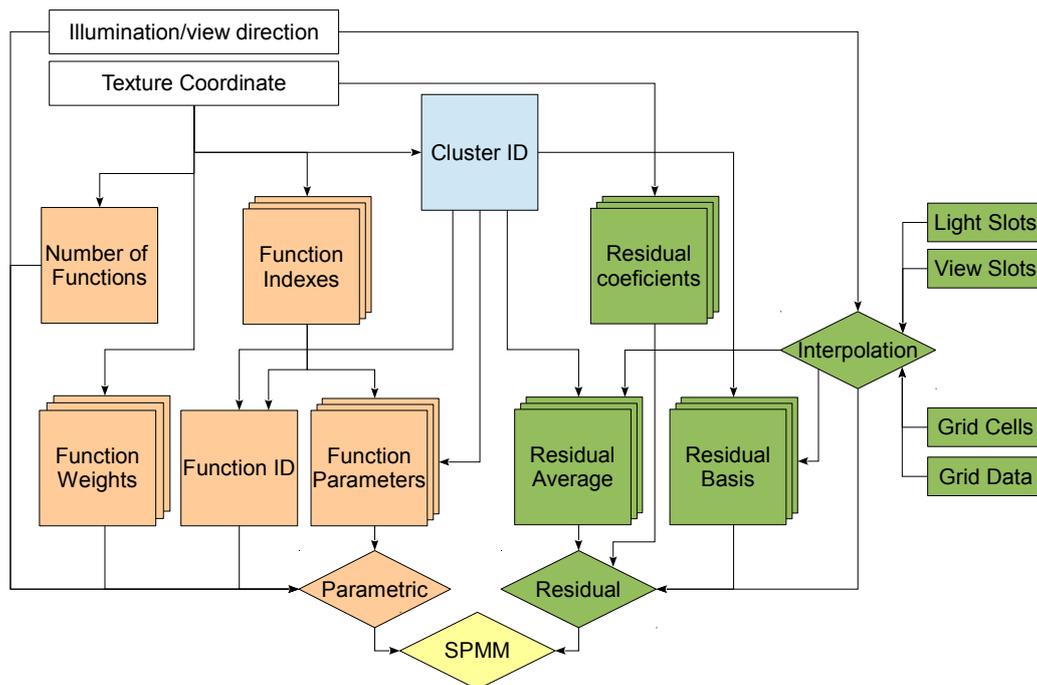


Figura 2. O fluxo de dados no programa de síntese. Os pequenos retângulos, os quadrados e os quadrados empilhados representam, respetivamente, texturas 1D, 2D, e 3D. Os diamantes representam a combinação dos dados de entrada.

dependem da direção de iluminação; implementações futuras poderão explorar este facto de modo a atingir melhor escalabilidade com o número de luzes pontuais.

Tal como indicado na figura 5, o aumento do número de SPMMs não afeta significativamente o número de FPS obtidos. De facto, o maior limitador do desempenho é o número de pontos intersetados que mapeiam numa SPMM, dado que a estes está associada uma carga computacional muito maior que a pontos com um modelo de iluminação mais simples. No entanto, mesmo que todos os pontos visíveis mapeiem em SPMMs, ou seja, 512×512 pontos (modelo *Cornell Box* no gráfico 5(a)), obtém-se uma taxa de visualização de cerca de 62 FPS.

3.3 Interreflexões Difusas

A qualidade das imagens sintetizadas depende dos fenómenos de transporte de luz simulados. As interreflexões difusas são especialmente relevantes, pelo que decidimos acrescentar a simulação deste fenómeno no nosso visualizador. Uma vez que este é um processo computacional muito exigente optamos por utilizar a técnica de oclusão ambiente; esta resulta numa aproximação pouco precisa às interreflexões difusas, mas com qualidade suficiente para esta aplicação.

A oclusão ambiente determina qual a percentagem de semiesfera superior que não está ocluída por geometria a uma dada distância do ponto a iluminar; a iluminação ambiente é depois multiplicada por esta percentagem antes de ser adicionada à iluminação incidente neste ponto. Para determinar a oclusão ambiente são disparados N raios ao

longo da semiesfera, distribuídos estocasticamente com estratificação. A percentagem destes raios que encontram uma interseção até à distância pré-determinada contribui para a oclusão ambiente.

O método descrito atrás tem um custo significativo, proporcional a N , que inviabiliza taxas de visualização interativas. Desenhamos então uma versão progressiva que dispara apenas um raio de oclusão por ciclo de desenho. O processo é reiniciado sempre que há movimentos da câmara. As imagens iniciais apresentam um ruído elevado, mas como as taxas de visualização se mantêm acima dos trinta FPS, a solução rapidamente converge para um nível de ruído perceptualmente irrelevante. A figura 6 apresentada em apêndice contém os resultados obtidos e compara com as imagens sem interreflexões difusas; o vídeo submetido como material suplementar inclui uma demonstração desta técnica.

4 CONCLUSÃO E TRABALHO FUTURO

Apresentamos um visualizador implementado em GPU usando o motor de *ray tracing* Optix, que combina os benefícios do formato compacto da SPMM, uma representação estado-da-arte para BTFs, com o poder computacional permitido pelos GPUs. A escalabilidade com o número de luzes e o número de pontos intersetados que mapeiam numa SPMM é linear; com o número de SPMMs presentes na cena é aproximadamente constante. O visualizador permite mapear várias SPMMs em modelos geométricos, e, combinando técnicas para aproximar fenómenos de iluminação direta e indireta no GPU, obtém

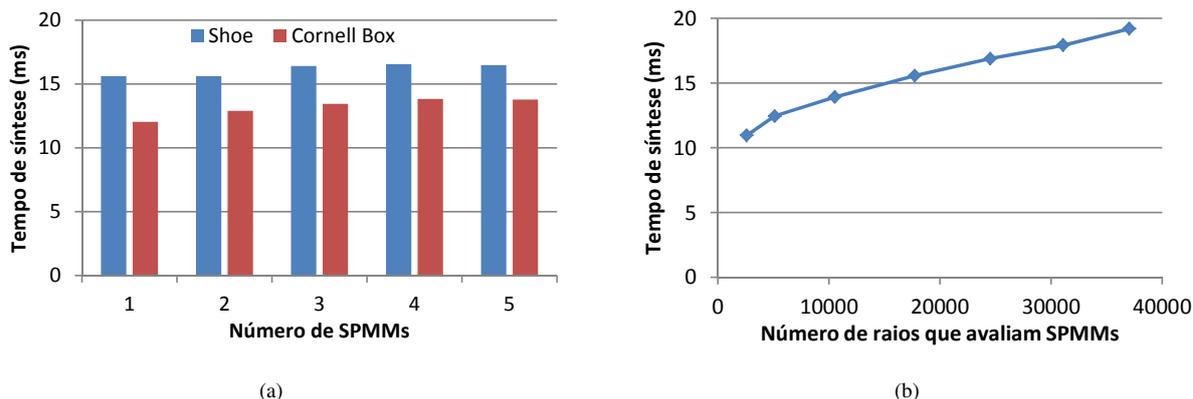


Figura 5. (a) Escalabilidade com o número de SPMMs em dois modelos geométricos, o número de raios disparados para SPMMs é constante. (b) Escalabilidade com o número de raios que avaliam uma SPMM, para o modelo do sapato com três SPMMs.

visualizações de alta qualidade e de alta fidelidade a taxas interativas.

Como trabalho futuro gostaríamos de melhorar a escalabilidade e expandir o visualizador para permitir edição interativa dos parâmetros da SPMM. Acreditamos que isto poderá ser útil para os *designers* e artistas da indústria, permitindo a prototipagem virtual rápida de novos produtos. Adicionalmente, técnicas de *displacement mapping* ([Wang03]) podem melhorar a aparência do material e permitir uma melhor percepção da profundidade e até da silhueta do material.

5. REFERÊNCIAS

- [Dana99] K. Dana, B. van Ginneken, S. K. Nayar, e J. J. Koenderink. Reflectance and texture of real-world surfaces. *ACM Trans. Graph.*, 18:1–34, January 1999.
- [Ebert02] D. S. Ebert, F. K. Musgrave, D. Peachey, K. Perlin, e S. Worley. *Texturing and Modeling: A Procedural Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edição, 2002.
- [Fernando04] R. Fernando. *GPU Gems: Programming Techniques, Tips and Tricks for Real-Time Graphics*. Pearson Higher Education, 2004.
- [Filip09] J. Filip e M. Haindl. Bidirectional texture function modeling: A state of the art survey. *Pattern Analysis and Machine Intelligence, IEEE Trans. on*, 31(11), nov. 2009.
- [Lagae08] A. Lagae e P. Dutré. Compact, fast and robust grids for ray tracing. *Computer Graphics Forum (Proc. of the 19th Eurographics Symp. on Rendering)*, 27(4), June 2008.
- [Lawrence06] J. Lawrence. *Acquisition and representation of material appearance for editing and rendering*. Tese de Doutorado, Princeton, NJ, USA, 2006. AAI3214568.
- [Ma04] W. Ma, S. Chao, B. Chen, C. Chang, M. Ouhyoung, e T. Nishita. An efficient representation of complex materials for real-time rendering. Em *Proc. of the ACM symp. on Virtual reality software and technology, VRST '04*, páginas 150–153, New York, NY, USA, 2004. ACM.
- [McAllister02] D. McAllister, A. Lastra, e W. Heidrich. Efficient rendering of spatial bi-directional reflectance distribution functions. Em *Proc. of the ACM SIGGRAPH/EUROGRAPHICS conf. on Graphics hardware, HWWS '02*, Aire-la-Ville, Switzerland, Switzerland, 2002. Eurographics Association.
- [Müller03] G. Müller, J. Meseth, e R. Klein. Compression and real-time rendering of measured btfs using local pca. Em T. Ertl, B. Girod, G. Greiner, H. Niemann, H.-P. Seidel, E. Steinbach, e R. Westermann, editores, *Vision, Modeling and Visualisation 2003*. Akademische Verlagsgesellschaft Aka GmbH, Berlin, Novembro 2003.
- [Müller05] G. Müller, J. Meseth, M. Sattler, R. Sarlette, e R. Klein. Acquisition, synthesis, and rendering of bidirectional texture functions. *Computer Graphics Forum*, 24(1):83–109, 2005.
- [Parker10] S. Parker, J. Bigler, A. Dietrich, H. Friedrich, J. Hoberock, D. Luebke, D. McAllister, M. McGuire, K. Morley, A. Robison, e M. Stich. Optix: a general purpose ray tracing engine. *ACM Trans. Graph.*, 29(4), Julho 2010.
- [Pharr04] M. Pharr e G. Humphreys. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.
- [Pharr05] M. Pharr e R. Fernando. *GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation (Gpu Gems)*. Addison-Wesley Professional, 2005.
- [Sattler03] M. Sattler, R. Sarlette, e R. Klein. Efficient and realistic visualization of cloth. Em *Eurographics Symp. on Rendering 2003*, Junho 2003.
- [Suykens03] F. Suykens, K. V. Berge, A. Lagae, e P. Dutré. Interactive rendering with bidirectional texture functions. *Computer Graphics Forum*, 22:463–472, 2003.
- [Wang03] L. Wang, X. Wang, X. Tong, S. Lin, S. Hu, B. Guo, e H. Shum. View-dependent displacement mapping. *ACM Trans. Graph.*, 22(3):334–339, Julho 2003.
- [Weyrich08] T. Weyrich, J. Lawrence, H. Lensch, S. Rusinkiewicz, e T. Zickler. Principles of appearance acquisition and representation. Em *ACM SIGGRAPH 2008 classes, SIGGRAPH '08*, New York, NY, USA, 2008. ACM.
- [Wu11] H. Wu, J. Dorsey, e H. Rushmeier. A sparse parametric mixture model for btf compression, editing and rendering. *Computer Graphics Forum*, 30(2):465–473, 2011.



(a) Um sapato mapeado com três SPMMs, sob iluminação direta e especu-
lar. Taxas de visualização de ≈ 52 FPS.



(b) O mesmo sapato mapeado sob iluminação direta, especular e difusa
indireta (oclusão ambiente). Taxas de visualização de ≈ 37 FPS.



(c) Um par de botas, com a SPMM *Wool*, sob iluminação direta e especu-
lar. Taxas de visualização de ≈ 84 FPS.



(d) O mesmo par de botas, sob iluminação direta, especular e difusa
indireta (oclusão ambiente). Taxas de visualização de ≈ 50 FPS.



(e) Uma camisola, com a SPMM *Pulli*, sob iluminação direta e especu-
lar. Taxas de visualização de ≈ 68 FPS.



(f) A mesma camisola, sob iluminação direta, especular e difusa indireta
(oclusão ambiente). Taxas de visualização de ≈ 32 FPS.

Figura 6. Vários modelos geométricos mapeados com SPMMs, sob iluminação direta e especular (coluna da esquerda) e sob iluminação difusa indireta por oclusão de ambiente (coluna da direita).

Realidade Virtual Aplicada ao Tratamento da Ansiedade Social

Tânia Pinheiro Ana Paula Cláudio Maria Beatriz Carmo
LabMAg, Faculdade de Ciências, Universidade de Lisboa

fc28181@alunos.fc.ul.pt, {apc, bc}@fc.ul.pt

Francisco Esteves
Centro de Investigação e Intervenção Social
Instituto Universitário de Lisboa/ISCTE

francisco.esteves@iscte.pt

Resumo

A ansiedade, cada vez mais frequente na nossa sociedade, pode ser muito debilitante em situações extremas e até mesmo conduzir a problemas de saúde sérios, se não for tratada de forma adequada. Este artigo descreve uma aplicação para apoio ao tratamento da ansiedade social, mais especificamente a ansiedade de falar em público. A aplicação envolve: i) uma simulação que se desenrola num cenário virtual de um auditório contendo um conjunto de humanos virtuais animados com comportamentos controláveis e ii) uma interface através da qual o terapeuta controla estes comportamentos e um conjunto de características e de eventos da simulação. O uso de ambientes de Realidade Virtual no tratamento de fobias surgiu na década de 90 e implica, habitualmente custos elevados. O desafio que tentámos abraçar com este trabalho foi desenvolver uma aplicação de baixo custo. Para desenvolver a aplicação escolhemos o Blender, uma ferramenta de uso livre para a criação de animações 3D e videojogos. A simulação é projectada numa simples tela de projecção, em perspectiva e de uma maneira tal que o paciente veja os elementos projectados com um tamanho idêntico ao dos elementos correspondentes do mundo real.

Palavras-Chave

realidade virtual, humanos virtuais, ansiedade social, medo de falar em público

1. INTRODUÇÃO

A ansiedade afecta cada vez um maior número de pessoas em todo o mundo, prejudicando e diminuindo a sua qualidade de vida. A ansiedade social (AS) é descrita no DSM-IV [DSM00] como um medo acentuado e persistente de embaraço em situações que exigem algum tipo de desempenho social, p.ex., falar em público. As pessoas com AS preocupam-se com o que os outros possam pensar delas, evitam muitas situações sociais, e recorrem frequentemente ao uso de drogas e/ou álcool a fim de serem capazes de as enfrentar.

Embora um certo nível de ansiedade seja natural e possa ter um valor motivacional positivo, em geral a ansiedade é difícil de tratar e em alguns casos pode ser altamente debilitante. Nestes casos, e se não for devidamente tratada, pode evoluir para problemas mais graves como, por exemplo, transtornos de ansiedade, comportamentos fóbicos ou ataques de pânico. A fobia social tem sido considerada a terceira desordem mental mais comum, com níveis de prevalência que variam de 5 a 13% [Gouveia00].

Durante as últimas décadas, o tratamento não-farmacológico mais aplicado aos transtornos de ansiedade e em particular na AS, tem sido a terapia cognitivo-

comportamental (TCC). Apesar de coexistirem diferentes protocolos e técnicas, a exposição às situações temidas é um ingrediente comum essencial. Acompanhando a evolução tecnológica, uma modalidade importante utilizada desde meados dos anos noventa, é a terapia de exposição recorrendo à Realidade Virtual (*Virtual Reality Exposure Therapy* - VRET). Esta técnica permite expor os pacientes a simulações das situações temidas, mesmo em casos em que a exposição real não é possível ou é difícil (p.ex., por ser muito cara).

A VRET apresenta vantagens, tais como a possibilidade de reproduzir e comandar o que ocorre nos ambientes virtuais, o controlo do risco e da situação enquanto a confidencialidade do paciente é mantida, factores que podem contribuir positivamente para a vontade do paciente fazer terapia. Segundo alguns estudos [Klinger04] [Herbelin05], a eficácia da terapia de exposição com ambientes virtuais é similar à da terapia tradicional. No entanto, também existem desvantagens: o alto custo de desenvolvimento dos ambientes virtuais, os equipamentos de RV dispendiosos e os efeitos secundários sentidos por alguns dos seus utilizadores.

O objectivo da aplicação que desenvolvemos é suportar simulações que possam ser usados na terapia de exposição

para o tratamento da AS e que se desenrolam num cenário de realidade virtual interactiva. Neste cenário existem humanos virtuais (HV) que se comportam de modo realista, exibindo um comportamento que tem como objectivo oferecer ao paciente uma experiência credível. O terapeuta faz uso da interface da aplicação para interferir no decurso da simulação, quer alterando o comportamento dos humanos virtuais, quer desencadeando eventos específico.

O desafio que tentámos abraçar com este trabalho foi desenvolver uma aplicação de baixo custo, um aspecto que facilitará certamente a sua disseminação e o seu uso.

Tirámos partido das potencialidades do Blender, um software público vocacionado para o desenvolvimento animações 3D e de videojogos, que já disponibiliza parte das funcionalidades que nos são necessárias. Os nossos cenários virtuais são exibidos numa tela com os HV projectados em tamanho real. O som também está incluído nas simulações, para que ambos os sentidos da visão e da audição sejam estimulados.

O desenvolvimento da aplicação tem sido acompanhado de perto por um psicólogo da equipa de trabalho que tem experiência neste tipo de terapias.

Este documento está organizado da seguinte forma: a secção 2 refere trabalhos nesta área; a secção 3 descreve a nossa aplicação e o processo de desenvolvimento do cenário e da interface; a secção 4 é dedicada às conclusões e ao trabalho futuro.

2. TRABALHO RELACIONADO

Em 2004, Klinger realizou um estudo em que examinou as mudanças no medo de falar em público em 36 participantes ao longo de 12 sessões [Klinger04]. Os participantes foram divididos em 2 grupos, um foi tratado com terapia cognitivo-comportamental tradicional e o outro com VRET. As exposições de VRET foram conduzidas em 4 ambientes virtuais e estes ambientes simulavam situações sociais que envolviam desempenho (por exemplo, falar em público), interacção inter-pessoal (por exemplo, uma conversa ao jantar), assertividade (por exemplo, ter de defender uma ideia) e avaliação (por exemplo, conversar enquanto se está a ser observado). De acordo com o estudo, os pacientes do grupo VRET mostraram uma redução mais significativa da sua ansiedade social do que os pacientes do grupo de TCC. Do ponto de vista clínico, a diminuição dos sintomas nos dois grupos foi similar tendo em conta critérios psicométricos.

Em 2005, Herbelin [Herbelin05] publicou um estudo de validação com 200 pacientes, confirmando que a sua plataforma de VRET cumpria os requisitos terapêuticos de exposição para a fobia social. Além disso também provou que é possível melhorar a avaliação clínica com ferramentas de monitorização integradas na aplicação, tais como o *eye-tracking*.

No entanto, este tipo de publicações apenas apresentam algumas das facetas da VRET. Estas técnicas tornaram-se a especialidade de clínicas e hospitais tais como a Virtually Better [VB], o Virtual Reality Medical Center [VRP] e o

Duke University Medical Center [DH] e são o resultado da investigação realizada ao longo de vários anos. Também existem soluções de uso livre tal como o projecto VRTherapy [VRT] que oferece diversas ferramentas e cenários para a *design* de ambientes.

3. TRABALHO DESENVOLVIDO

3.1. Introdução

A aplicação desenvolvida, designada por *Virtual Spectators*, tem por objectivo apoiar o tratamento da ansiedade social, mais especificamente a ansiedade de falar em público. A aplicação envolve: i) uma simulação que se desenrola num cenário virtual de um auditório contendo um conjunto de HV animados com comportamentos controláveis e ii) uma interface através da qual é efectuado o controlo destes comportamentos e de um conjunto de eventos que podem ser desencadeados na simulação. A aplicação tem dois tipos de utilizadores: o terapeuta e o paciente. Durante uma sessão de terapia, o cenário virtual é palco de uma simulação que é controlada pelo terapeuta e observada pelo paciente. O paciente, enquanto efectua o seu discurso perante a assembleia virtual, 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 afectar o decurso da simulação em conformidade, quer alterando o comportamento dos HV, quer desencadeando eventos específicos no cenário (por exemplo, acender ou apagar luzes no auditório). O terapeuta também configura o cenário inicial da simulação.

A instalação que vamos adoptar para a aplicação, e que está ilustrada na figura 1 é composta por um computador que executa a aplicação, um projector e uma tela.



Figura 1: Diagrama de utilização da aplicação

A simulação (figura 2) é projectada na tela e a janela de interacção no computador do terapeuta (figura 3), um vez que a aplicação cria duas janelas disjuntas.

Assim, temos as vantagens de o equipamento ser relativamente barato e fácil de instalar, de o tamanho da imagem projectada 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,



Figura 2: Janela de Simulação

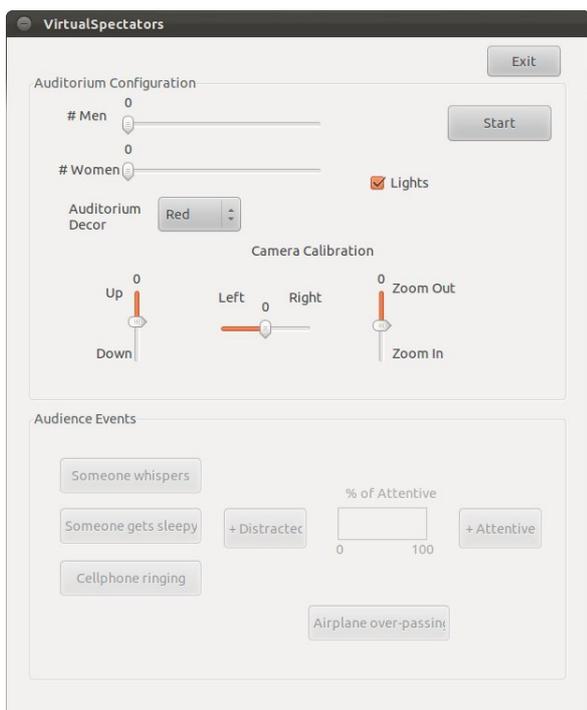


Figura 3: Janela de Interação

em ambientes de ensino. Além disto, eliminam-se os eventuais efeitos secundários desagradáveis que podem surgir da utilização de equipamentos de RV, como vertigens e náuseas.

Na janela de interação foram incluídas três barras de deslocamento que permitem ajustar, dentro de certos limites, a posição inicial da câmara dentro do cenário. É possível efectuar translações da câmara segundo três direcções ortogonais: aproximar e afastar (zoom), deslocar para a direita e para a esquerda, e deslocar para cima e para baixo. Isto permite calibrar a posição da câmara, adequando-a à posição e altura do paciente relativamente à tela, por forma a que os elementos na projecção tenham, como se pretende, um tamanho próximo do real.

O conceito de *serious game* foi a principal inspiração na arquitectura e *design* do sistema. Um *serious game* é um videojogo desenhado com um propósito diferente do puro entretenimento. Estas variantes de videojogos são cria-

das tendo em vista um desafio específico para o utilizador [Hertzschuch08]. Sendo a aprendizagem um processo activo, construtivo, pessoal e emocional, este tipo de jogos tem o potencial de ajudar no processo de aprendizagem e ser utilizado em diversas áreas tais como treino, educação e saúde. A aplicação *Virtual Spectators* foi implementada seguindo a arquitectura de um jogo. O motor de jogo do Blender [Ble] gere todo o conteúdo gráfico enquanto a componente lógica do sistema que implementámos em Python é executada em paralelo.

3.2. Implementação

3.2.1. O cenário da simulação

Para desenvolver a aplicação escolhemos o Blender [Ble], um software de uso livre. A principal vantagem de utilizarmos esta ferramenta é que, além de suportar toda a criação do conteúdo 3D, possui um motor de jogo integrado, minimizando a necessidade de conversões de formatos entre softwares distintos. Também podemos integrar módulos em Python estendendo as capacidades do motor de jogo.

A implementação de toda a parte gráfica está esquematizada na figura 4.

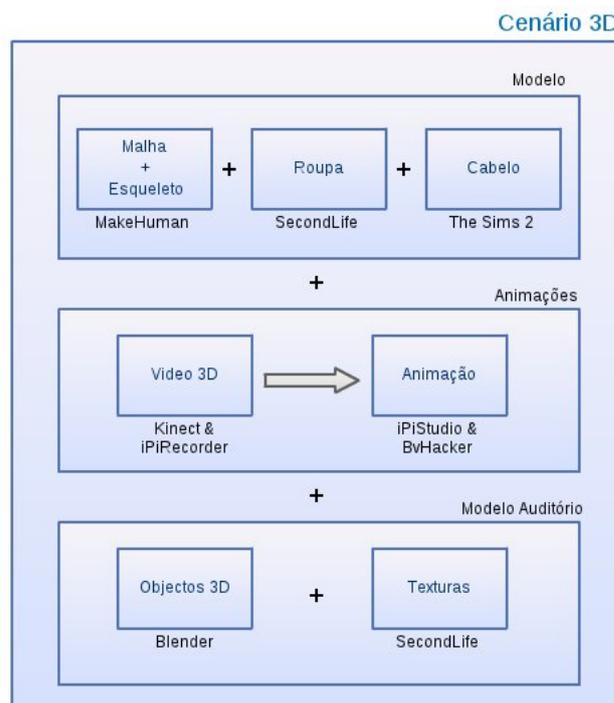


Figura 4: Esquema da implementação do cenário 3D

Criação dos humanos virtuais

Para a criação dos HV foi utilizado o software MakeHuman [MH], uma aplicação que permite criar modelos de personagens humanas 3D realistas. A ferramenta suporta a exportação destes modelos para o formato específico do Blender e permite configurar um conjunto de opções específicas em relação aos personagens, tais como o tipo de esqueleto e a complexidade da malha poligonal. Apesar de já estarem implementadas no MakeHuman funcionalidades que permitem a criação de roupa e cabelo, estas

possuem um nível de complexidade muito superior ao suportado pelo motor de jogo do Blender. Deste modo, foi necessário importar para o Blender HV sem roupa nem cabelo. Para cada personagem foi criada então a roupa e o cabelo. As roupas são texturas de uso livre que foram criadas inicialmente para outras aplicações (*Second Life* [SL] *OpenSim* [OS], *The Sims 2* [TS2]) que aplicámos através da técnica *UV Mapping*. Recorrendo a diversos websites com conteúdo livre para *SecondLife/OpenSim* [OSC] [SLS] [LK] foi possível obter texturas de vestuário credíveis, eliminando assim a necessidade da sua criação de raiz. Com o editor de imagens Gimp [Gim] obteve-se uma maior variedade de vestuário alterando simplesmente a cor base das peças de roupa. Os cabelos utilizados foram obtidos recorrendo ao website de conteúdo livre para o "The Sims 2", ModTheSims [MTS], sendo extraídas as malhas poligonais e as texturas através do software SimPE [Sim]. Também foi utilizada a mesma técnica de variação de cores para obter vários tipos de cabelo. Finalmente foi necessário criar poses chave (*shape keys*) no Blender para cada um dos HV para definir o comportamento de "pesta-nejar".

Animação dos humanos virtuais

As animações corporais foram criadas através da técnica de *motion capture*, recorrendo a uma câmara Kinect [Kin] e a software da iPiSoft [iPi]: o iPiRecorder e o iPiStudio. O iPiRecorder aliado à Kinect possibilita a gravação de um vídeo 3D. Posteriormente este vídeo é processado com o software iPiStudio que permite eliminar o ambiente de fundo e mapear um esqueleto que segue os movimentos do actor. A animação do esqueleto é depois exportada para o formato *bvh* e editada com o software BvHacker [BVH], uma aplicação originalmente criada para o *Second Life* e que permite o corte de secções da animação, criação de animações em ciclo, alteração dos limites de rotação e translação, entre outras opções. Além da criação de conteúdos em 3D, o Blender também possibilita a adição de *addons* criados por utilizadores. O MakeHuman fornece vários *addons* para Blender, entre os quais o *mocap tools*. Esta ferramenta importa as animações directamente para os personagens realizando o mapeamento dos esqueletos de forma automática e agilizando o processo de importação de animações. Devido a restrições dos *drivers* da câmara Kinect as animações resultantes não contêm as rotações das mãos e da cabeça, tendo sido necessário alterar as animações manualmente. Para isso recorreremos ao *addon mocap tools* para reposicionar o esqueleto e adicionámos manualmente as animações da cabeça e das mãos no *Action Editor* no menu *Dopesheet* do Blender. Criámos um conjunto de animações para os HV: "estar atento", "estar distraído", "receber um sms", "dormir", "cochichar", "ouvir o cochicho", "pesta-nejar" e "adormecer/acordar". Todas as animações, em especial as mais complexas tais como "estar distraído" e "receber um sms", foram alvo de um elevado número de alterações. Estas alterações foram realizadas manualmente recorrendo ao editor de animações do Blender que permitiu a adição dos movimentos das mãos e da cabeça.

Modelação do cenário

O auditório do cenário foi inteiramente modelado e texturizado em Blender utilizando as técnicas de modelação 3D habituais, tais como *mesh modeling*, *extrude*, *sculpt*, entre outras. Foram usadas texturas provenientes do *SecondLife/OpenSim*.

Todo o software utilizado é de uso livre à excepção do iPiStudio do qual foi usada uma versão de teste.

3.2.2. A interface

Para poder controlar os HV e desencadear eventos na simulação foi necessário criar uma forma de comunicação entre a simulação e a interface. Como os métodos de *input* do Blender (teclado e *joystick*) não eram viáveis para o tipo e quantidade de eventos pretendidos, foi criada uma janela de interacção que utiliza a comunicação entre processos para passar informações à janela de simulação e vice-versa. Inicialmente a comunicação era realizada em apenas um dos sentidos utilizando o módulo *Subprocess* [Sub], mas, mais tarde, passámos a usar o módulo *pexpect* [pex] que suporta troca de mensagens nos dois sentidos e usa um processo mais simples de implementação. Para controlar o fluxo de mensagens foi necessário implementar *threads* que gerem o *input/output* de forma a não bloquear as janelas.

A figura 5 esquematiza a articulação da nossa aplicação com o motor de jogo do Blender. O cenário 3D é actualizado pelo motor de jogo e é controlado através do nosso script em Python designado por Main Controller. O motor de jogo é responsável por criar a janela de simulação ao mesmo tempo que a janela de interacção comunica com o Main Controller. Este *script* que recebe o *input* da janela de interacção, além de accionar os eventos também gere as animações que estão a decorrer recorrendo ao módulo *bge* [BGE] do Blender. Este módulo possibilita o acesso aos objectos e a modificação das suas propriedades, tais como visibilidade, texturas, animação, entre outros, o que é fundamental para configurar o cenário.

A janela de interacção foi implementada recorrendo ao módulo *wxPython*[wxP] que utiliza o *design* de janelas nativo do sistema operativo e foi concebida para ser usada pelo terapeuta em dois instantes de tempo distintos: antes e durante a simulação. Assim, a interface está dividida em duas zonas, estando apenas uma delas activa em cada um destes instantes.

Antes da simulação, a interface permite configurar o auditório (figura 6): definir o número de HV masculinos e femininos que estarão presentes na simulação, escolher as cores dos elementos físicos da sala (paredes, cadeiras), definir o estado das luzes do auditório (ligadas, desligadas) e a posição da câmara dentro do cenário 3D.

Depois de configurar o cenário, o terapeuta dá início à simulação fazendo uso do botão Start; este, depois de accionado, passa a Pause e permite suspender a simulação. O botão Exit da interface permite sair da aplicação em qualquer altura. Durante a simulação, é permitido (figura 7): alterar o comportamento dos HV para que haja

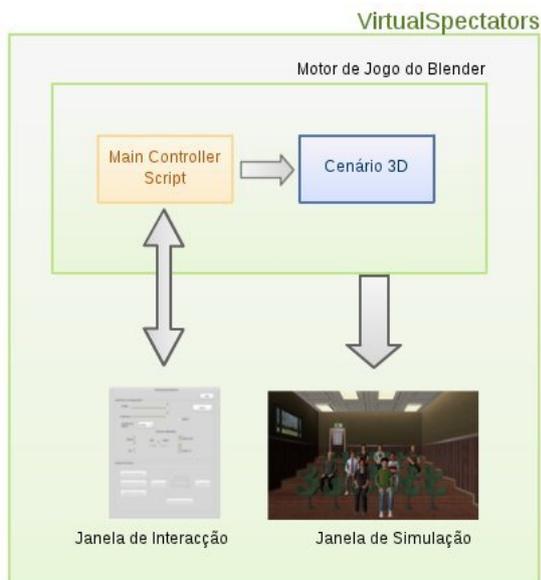


Figura 5: Esquema da arquitectura da aplicação

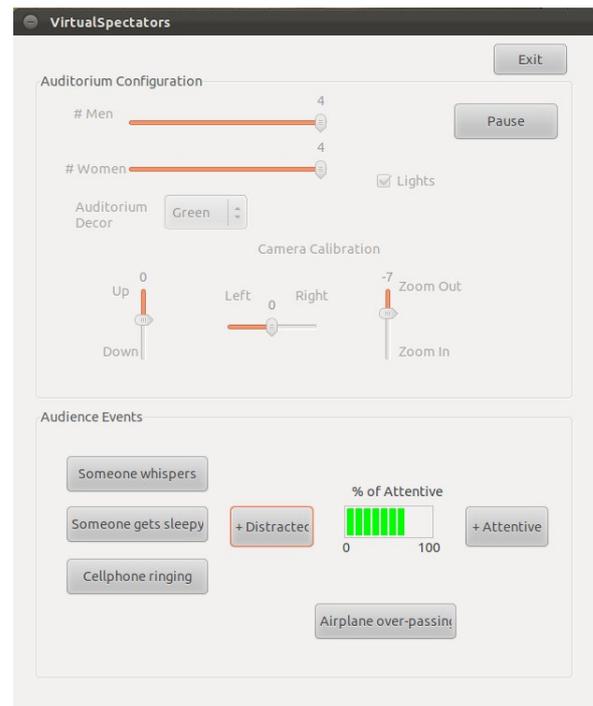


Figura 7: Janela de Interação (Modo Simulação)

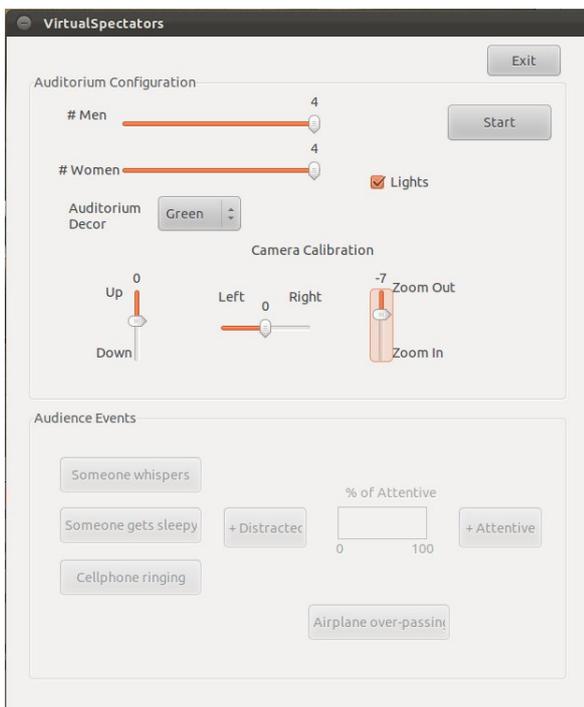


Figura 6: Janela de Interação (Configuração do auditório)

mais atentos ou distraídos, colocar alguns a dormir, desencadear a ocorrência de um cochicho no fundo da sala ou a recepção de um sms no telemóvel seguida da correspondente acção de escrever um sms de resposta. Na figura 8 podem observar-se vários destes eventos a ocorrer na simulação. Existe ainda um botão que desencadeia a geração do som de um avião a sobrevoar o auditório e sempre que ocorre um cochichar e a chegada de um sms ouvem-se os sons correspondentes associados. Foi utilizado o módulo *audaspace* [aud] para adicionar sons aos eventos: "cochichar", "receber um sms" e "avião a sobre-

voar o auditório". Estes sons foram obtidos na biblioteca online freesound.org [FS] de domínio público.



Figura 8: Vários eventos a decorrer em simultâneo na janela de simulação

Um dos pormenores mais importantes da simulação é que os HV estejam a olhar para o paciente quando estão atentos. A versão actual do Blender, 2.62, não permite alterar a posição da cabeça no decorrer da animação. A forma que adoptámos para conseguir colocar os HV a olharem para a câmara consiste em alterar o ângulo relativo do osso da cabeça tendo em conta a posição desta. Como o ângulo do osso é alterado, impossibilita a sua utilização nas animações posteriores e o efeito de transição entre animações é perdido. Ainda estamos a tentar solucionar este problema.

Finalmente foi adicionado ao cenário um personagem com um certo grau de autonomia. Os comportamentos deste

HV (à direita na fila da frente nas figuras 9 e 10) não são controlados pelo terapeuta mas sim pelo nível de atenção existente no auditório. Trata-se de um teste que permitiu verificar a possibilidade de estender a nossa aplicação incorporando um módulo de inteligência artificial.



Figura 9: O HV à direita na fila da frente está atento se o resto da audiência também estiver



Figura 10: O HV à direita na fila da frente está distraído porque mais de metade da audiência está distraída

4. CONCLUSÃO E TRABALHO FUTURO

A condição mais importante para realizar uma terapia de exposição bem sucedida é provocar no paciente estímulos análogos aos que surgiriam em situações reais, ou seja, provocar no paciente a sensação de presença [Herbelin05].

É assim importante que a aplicação permita criar uma exposição controlada interactivamente pelo terapeuta que é o responsável por gerir os níveis de estímulo provocados sobre o paciente.

O interface que desenvolvemos pretende ser simples e intuitivo, fornecendo ao terapeuta um leque de opções de configuração do auditório e de eventos na simulação.

A simulação tenta representar o mais realisticamente possível uma audiência que cause "impacto" no paciente recorrendo a HV realistas e a animações de *motion capture*. Neste tipo de aplicações é preciso sempre encontrar o equilíbrio entre o realismo do ambiente 3D e a necessidade de que este seja gerado em tempo-real.

O facto de a solução adoptada ser de baixo custo contribuirá para disseminar e facilitar o seu uso.

O desenvolvimento da aplicação tem vindo a ser seguido de perto por um psicólogo da equipa que tem avaliado informalmente o estado corrente e dá pistas para as etapas seguintes e funcionalidades a implementar.

Actualmente está a ser levada a cabo uma primeira etapa de avaliação com um grupo de psicólogos. A avaliação será realizada em 2 fases, em que na primeira é avaliada a facilidade de interacção com a aplicação e na segunda é simulada uma sessão de terapia de exposição com um paciente voluntário. No final de cada fase será preenchido um questionário de avaliação.

Depois da incorporação dos necessários melhoramentos e eventuais correcções identificados nesta primeira etapa, seguir-se-á um ensaio clínico com pacientes, que nos permitirá tirar conclusões sobre a adequação efectiva da nossa abordagem no tratamento da ansiedade de falar em público.

Está já em curso um trabalho que visa estender a aplicação de modo a suportar mais cenários, começando por um cenário de entrevista de emprego no qual se pretende explorar a expressão de emoções faciais.

Nesta linha será desenvolvido um módulo de inteligência artificial ligado à simulação de emoções. Consideramos que este tipo de cenários pode vir a ser interessante de usar, por exemplo, em contexto universitário, para ajudar alguns jovens que sofram de ansiedade de falar para um público ou para um conjunto mais restrito de pessoas numa situação de avaliação. Temos conhecimento que existe um número significativo de estudantes que recorre aos gabinetes de psicologia existentes nas faculdades para solicitar apoio nestas situações concretas.

5. AGRADECIMENTOS

Agradecemos à Fundação para a Ciência e Tecnologia (FCT) e ao Laboratório de Modelação de Agentes (Lab-MAg) o apoio financeiro para a realização deste trabalho.

6. REFERÊNCIAS

- [aud] Blender audaspace module website. <http://wiki.blender.org/index.php/User:NeXyon/GSoC2010/Audaspace>. [Last access 29 September 2012].
- [BGE] Blender game engine documentation website. <http://www.blender.org/documentation/blender/python/api/2.63/11/>. [Last access 29 September 2012].
- [Ble] Blender website. <http://www.blender.org>. [Last access 29 September 2012].
- [BVH] Bvhacker website. <http://davedub.co.uk/bvhacker>. [Last access 29 September 2012].
- [DH] Dukehealth phobia treatment website. <http://www.dukehealth.org/services/psychiatry/programs/adult/services/phobia/treatment>. [Last access 29 September 2012].
- [DSM00] *Diagnostic and statistical manual of mental disorders (Revised 4th ed.)*. Washington DC: American Psychiatric Press, 2000.

- [FS] Free sound database website. <http://www.freesound.org/>. [Last access 29 September 2012].
- [Gim] Gimp website. <http://www.gimp.org/>. [Last access 29 September 2012].
- [Gouveia00] J. Pinto Gouveia. *Ansiedade Social: da timidez à Fobia Social*. Quarteto, 2000.
- [Herbelin05] B. Herbelin. *Virtual reality exposure therapy for social phobia*. Tese de Doutoramento, Université Louis Pasteur, Strasbourg, 2005.
- [Hertzschuch08] K. Hertzschuch. Serious games. Em *Hauptseminar "E-Learning – Sommersemester, Ludwig Maximilians Universität München*, 2008.
- [iPi] ipisoft website. <http://www.ipisoft.com/>. [Last access 29 September 2012].
- [Kin] Kinect website. <http://www.xbox.com/en-US/kinect>. [Last access 29 September 2012].
- [Klinger04] E. Klinger, P. Légeron, S. Roy, I. Chemin., F. Lauer, e P. Nugues. Virtual reality exposure in the treatment of social phobia. *Studies in Health Technology and Informatics*, 99:91–119, 2004.
- [LK] Linda kellie website. <http://www.lindakellie.com>. [Last access 29 September 2012].
- [MH] Makehuman website. <http://www.makehuman.org>. [Last access 29 September 2012].
- [MTS] Mod the sims website. <http://www.modthesims.info>. [Last access 29 September 2012].
- [OS] Opensim website. <http://opensimulator.org>. [Last access 29 September 2012].
- [OSC] Opensim creations website. <http://opensim-creations.com>. [Last access 29 September 2012].
- [pex] Python pexpect module website. <http://www.noah.org/wiki/pexpect>. [Last access 29 September 2012].
- [Sim] Simpe website. <http://sims.ambertation.de>. [Last access 29 September 2012].
- [SL] Second life website. <http://secondlife.com/>. [Last access 29 September 2012].
- [SLS] Second life shirts website. <http://www.secondlife-shirts.com>. [Last access 29 September 2012].
- [Sub] Python subprocess module website. <http://docs.python.org/library/subprocess.html>. [Last access 29 September 2012].
- [TS2] The sims 2 website. <http://thesims2.ea.com/>. [Last access 29 September 2012].
- [VB] Virtuallybetter website. <http://www.virtuallybetter.com>. [Last access 29 September 2012].
- [VRP] Vrphobia website. <http://www.vrphobia.com>. [Last access 29 September 2012].
- [VRT] Vr therapy website. <http://www.vrtherapy.net>. [Last access 29 September 2012].
- [wxP] wxpython website. <http://www.wxpython.org>. [Last access 29 September 2012].

Modelação Procedimental de Ambientes 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

A modelação de ambientes rodoviários para simulação de condução pode-se tornar uma tarefa morosa e dispendiosa de recursos, se não se recorrer a métodos de modelação automática.

Os ambientes rodoviários utilizados na simulação de condução têm que obedecer a elevados requisitos de especificação e de realismo. A simulação de condução para fins científicos requer ainda a coexistência de uma descrição semântica do ambiente, que permitirá a colocação de atores durante a simulação e a produção de relatórios relativos à simulação.

Este artigo apresenta uma proposta de uma metodologia para a modelação procedimental de ambiente rodoviários adequados para simulação em tempo real. A metodologia proposta integra a modelação procedimental de ambiente rodoviários num único processo desde a determinação dos nós de ligação até à obtenção do modelo geométrico e da descrição semântica de todos os elementos envolvidos. No desenho desta metodologia pretendeu-se mapear os conceitos e práticas utilizadas em engenharia de vias, produzindo assim ambientes rodoviários semelhantes aos encontrados em traçado real. Será possível a interação, por utilizadores não especialista em engenharia informática, em qualquer fase do processo de especificação do ambiente permitindo cumprir as especificações particulares de cada ensaio experimental.

Estão previstas tarefas de avaliação dos modelos obtidos, quer a nível do seu traçado, da rede rodoviária gerada, assim como avaliação de todo o processo automático de geração de modelos.

Prevê-se que a metodologia proposta permita obter de forma eficiente modelos 3D de excelente qualidade e a adequados à utilização em sistemas de simulação visual para fins científicos.

Palavras Chave

Ambientes Rodoviários, Modelação 3D, Modelação Procedimental, Simulação de Condução.

1. INTRODUÇÃO

A criação de ambientes rodoviários de grandes dimensões, adequados para simulação de condução para fins científicos é uma tarefa morosa e dispendiosa de recursos. A criação de ambientes rodoviários virtuais requer a correta modelação de redes rodoviárias e a manipulação de objetos 3D tradicionais à região em que se insere, que é necessário integrar. Esta tarefa pode-se tornar demasiado trabalhosa e dispendiosa em termos de recursos se não se possuir uma metodologia automática de geração de ambientes virtuais.

Uma alternativa à realização de experiências de condução em ambiente real é a simulação de condução, que para além de um evidente aumento das condições de segurança, permite o controlo e monitorização de diferentes variáveis que seria inatingível em condições de tráfego real. A simulação de condução permite a realização de estudos experimentais impraticáveis em condições reais, uma vez

que reflete condições de total segurança para os participantes em eventos de tráfegos de alto risco, como por exemplo: estudo de manobras de ultrapassagem com risco de colisão frontal. Os simuladores de condução para fins científicos são requeridos por utilizadores nas mais diversas áreas, como por exemplo: engenharia de vias, psicologia e ergonomia.

Para a implementação de experiências de simulação de condução, além da geração do modelo visual do ambiente rodoviário deverá ser criada a correspondente definição semântica. A definição semântica é uma descrição de alto nível, de caracterização do ambiente rodoviário, que vai permitir parametrizar o modelo dinâmico do ambiente virtual. No caso específico de simulação de condução permite parametrizar: o comportamento dos atores, os eventos de tráfego específicos de cada experiência, a produção de relatórios, entre outros. Sempre que se deseje incluir participantes virtuais ou avaliar desempenhos de

participantes humanos, é fundamental que outros níveis de informação coexistam com a informação geométrica [Thomas00]. Adicionalmente, a definição semântica é também utilizada para geração de relatórios relativos à experiência de simulação de condução.

As ferramentas tradicionais de modelação de ambientes virtuais não permitem a criação conjunta destas duas representações.

A conceção estratégica de uma rede rodoviária, implica definir que nós (cidades, cruzamentos, rotundas, etc.) são interligados e qual o tipo de estrada utilizado. Esta definição envolve normalmente especialidades muito diversas, para além da Engenharia de vias e requer a análise da rede como um todo. Deste processo resulta essencialmente a definição topológica da rede que especifica as ligações entre nós e os respetivos tipos de via.

A partir da definição topológica da rede, é possível passar ao planeamento individual de cada estrada. Recorrendo a métodos de engenharia de vias, é criado um corredor entre os dois pontos de ligação, a uma escala que permita tratar a morfologia do terreno e respetivas condicionantes. Sobre esse corredor são depois traçados um conjunto de alinhamentos retos, que posteriormente irão permitir obter o traçado da via.

Nesta fase, são normalmente utilizadas aplicações de projeto de vias de comunicação, tais como o Civil3D. Estas aplicações exigem uma exaustiva interação com o utilizador e requerem deste um grande nível de conhecimento específico da área de vias de comunicação [Autodesk12]. Não são portanto adequadas à obtenção rápida de ambientes virtuais para simuladores por preparadores de outras áreas. Por outro lado, o Civil3D, não gera informação semântica suficiente para ser utilizada em simulação em tempo real.

As metodologias mais recentes de geração de ambientes rodoviários para a criação da rede obrigam à prévia definição do terreno, definição dos nós ou cidades de ligação e pontos de interesse a ligar pela rede rodoviária [Galín11].

Atualmente existem ferramentas que permitem a geração automática de ambientes rodoviários adequados para simulação de condução para fins científicos, mas requerem a prévia especificação do eixo da via, normalmente a partir de dados de projeto. Estas ferramentas possibilitam um elevado nível de controlo por parte do utilizador, permitindo cumprir as especificações particulares de cada projeto [Leitão00] [Campos07].

Não é conhecida nenhuma metodologia que gere automaticamente um ambiente rodoviário que cumpra os requisitos para uso em simulação de condução para fins científicos. Por outro lado, os ambientes rodoviários gerados deverão apresentar um traçado semelhante ao encontrado em estradas reais. Adicionalmente deverá existir a possibilidade de interação no processo de geração do ambiente rodoviário, de forma a permitir a parametrização de parâmetros específicos de cada projeto.

Neste artigo, apresenta-se uma metodologia inovadora para obtenção de redes rodoviárias de grandes dimensões de forma rápida e eficiente. A rede rodoviária será obtida a partir da definição de um conjunto de nós de interligação. Posteriormente com base na definição topológica da rede será determinado o traçado para cada via. No cálculo do traçado da via serão mapeados os métodos utilizados em engenharia de vias, para além de outros contributos identificados em trabalhos semelhantes [França11] [EP94] [Walton05] [Smelik08] [Galín11].

O utilizador deverá poder interagir em qualquer fase do processo de geração do ambiente rodoviário, permitindo definir critérios específicos de cada modelo, produzindo ambientes totalmente controláveis.

No caso de não existir nenhuma especificação inicial, a metodologia proposta será capaz de gerar um ambiente rodoviário aleatoriamente.

1.1 Organização do artigo

Na secção 2. É apresentado o estado da arte nas áreas relevantes para a definição da metodologia proposta e são apresentados os trabalhos relacionados mais significativos.

A secção 3. Descreve de forma detalhada a metodologia que se propõe para preparação de ambientes destinados a simuladores de condução.

Na secção 4. É apresentada a análise crítica da metodologia proposta e apontadas as direções para os próximos desenvolvimentos.

2. ESTADO DA ARTE

2.1 Redes rodoviárias

Uma rede, como por exemplo uma rede rodoviária pode ser definida por um conjunto de nós e ligações que os interligam (1).

$$R = (N, L) \quad (1)$$

onde R representa a rede, N os nós da rede e L as ligações entre os diferentes nós da rede.

Cada ligação faz a interligação entre dois nós distintos (2).

$$L = \{(N_i, N_j)\} \quad (2)$$

onde L representa a ligação, N_i e N_j os nós extremos da ligação.

Um percurso P (3), com extensão K através de uma rede, corresponde a uma sequência de ligações de nós, entre cada par de nós temos uma ligação L .

$$P = \langle N_0, N_1, \dots, N_k \rangle \quad (3)$$

onde P representa um percurso na rede, N_0 a N_k representam os nós interligados nesse percurso.

No caso de uma rede rodoviária, os nós correspondem a pontos de ligação e as ligações a estradas. Uma rede rodoviária pode ser definida por uma estrutura hierárquica de ligações em 3 níveis: rede de autoestradas, rede de

estradas secundárias, rede de estradas urbanas [Teoh08] [Weber09].

As redes rodoviárias são planeadas para ligar pontos de interesse, permitindo a deslocação de pessoas para os diversos pontos, tentando minimizar os custos de construção e os custos de operação. Os custos de construção estão associados à implantação em terreno real da via, em que engloba os encargos da própria via e também os custos com aterros, terraplanagens, expropriações, entre outras variáveis. Os custos de operação estão relacionados com a deslocação entre dois pontos e entra no seu cálculo vários fatores, como por exemplo: combustível, extensão do percurso, tempo de duração da viagem, entre outros.

O planeamento de uma rede rodoviária tem como objetivo a otimização dos custos de construção e operação da rede e também a minimização dos impactos desta rede no meio ambiente, procurando cumprir os níveis de segurança, satisfazer os padrões dos níveis de serviço e satisfazer a necessidades dos utentes

2.2 Conceção de vias de comunicação

A conceção de uma via parte inicialmente por uma decisão estratégica de que pontos, nós ou cidades, a nova estrada irão interligar. Essencialmente esta decisão é tomada tendo em consideração parâmetros como: crescimento demográfico, nível de serviço, estudos de tráfego e custo de construção.

Um projeto de uma estrada desenvolve-se de acordo com as seguintes fases: programa preliminar, estudo prévio e projeto de execução [Françal1]. O programa preliminar fornecido pelo dono da obra é basicamente um caderno de encargos constituído por disposições gerais, elaborado sobre carta militar à escala de 1:5000/1:10000, onde figuram as características específicas pretendidas e que condicionaram 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. O estudo de traçado nesta fase é realizado tipicamente à escala de 1:2500/1:5000 sobre carta militar ou levantamentos topográficos. É nesta fase que intervêm os estudos de tráfego, geológicos e geotécnicos, paisagísticos e de impacto ambiental.

Finalizado e aprovado o estudo prévio, 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. Um projeto de execução de uma estrada envolve várias especialidades de engenharia civil, tais como: traçado, terraplanagens, drenagem, pavimentação, sinalização e segurança, obras de arte e obras acessórias. O traçado geométrico da via é a primeira tarefa a ser realizada, onde se define a diretriz (traçado em planta), o perfil longitudinal (traçado em altimetria) e o perfil transversal.

Para obter o traçado da diretriz de uma estrada, em engenharia de vias, pode ser utilizado o conceito de linha dos zeros (figura 4).

Após estudo do traçado da via desenvolvem-se as restantes especialidades, sendo geralmente os estudos conducentes à seleção dos dispositivos de sinalização a adotar, um dos últimos a ser realizados. Na tarefa da sinalização, fazem sempre parte do projeto dois tipos de sinalização, uma referente à marcação horizontal e a outra referente à sinalização vertical. Na sinalização vertical existem ainda duas famílias, sendo uma referente à sinalização de código e a outra respeitante à sinalização de orientação, setas direcionais, pórticos e painéis laterais.

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].

2.3 Modelação de ambientes virtuais

A modelação de ambientes virtuais pode ser organizada em dois grupos, modelação de ambientes de cidade e modelação de ambientes rodoviários, à semelhança da classificação apresentada por Smelik et al. em 2009 [Smelik09]. Tipicamente estes trabalhos são orientados para a qualidade dos modelos obtidos, não existindo uma preocupação muito forte na criação de modelos de redes viárias, principalmente segundo as normas de traçado, de forma a produzir traçados semelhantes aos reais. Na criação da rede urbana podem adotar um ou mais padrões predefinidos de malha rodoviária. Tipicamente as redes de estradas urbanas seguem um determinado padrão: ortogonal, radial ou em ramificação [Sun02]. Mesmo os trabalhos mais relevantes de geração automática de redes viárias não abordam a possibilidade de utilização em simulação de condução em tempo real para fins científicos [Smelik08] [Galini11].

Paris et al. apresentam uma metodologia para a geração de cidades virtuais, baseada na modelação procedimental baseada em sistemas L [Paris01]. Neste trabalho é apresentada uma aplicação apelidada de *CityEngine*, que é capaz de modelar completamente uma cidade, com um conjunto reduzido de dados de entrada e completamente controlável pelo utilizador. O mapa de estradas é gerado recorrendo a uma extensão dos sistemas L, *Extended L-Systems*.

Che et al. apresentam um trabalho que aborda a problemática da modelação interativa de uma rede de estradas de uma grande área urbana [Chen08]. Os autores descrevem um método, em que o utilizador pode criar uma rede de estradas ou modificar uma já existente a partir de um esboço. Descrevem uma abordagem baseada em *Tensor Field* para guiar a criação da rede de estradas e gerar a aparência global do mapa rodoviário da área urbana a modelar.

Uma proposta para a modelação procedimental de um ambiente virtual, a partir de um esboço inicial, é apresentada por Smelik et al. em 2008 [Smelik08]. O autor descreve um processo em que o ambiente virtual é criado de

forma hierárquica. Partindo do modelo do terreno segue-se o modelo hidrográfico, o modelo da vegetação, o mapa rodoviário, e por último o modelo da cidade. A geração do modelo virtual global resulta da fusão das várias camadas não esquecendo a adequação do modelo de terreno às restantes definições [Latham06]. Para a geração do modelo geométrico do ambiente rodoviário são exploradas as técnicas apresentadas por Kelly et al. em 2008 [Kelly08].

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]. São contempladas modelações de curvas com a utilização de clotóides e é considerado o traçado em altimetria. Existe uma especial atenção na preparação dos modelos obtidos para a visualização em tempo real. Algumas destas metodologias foram já adotadas anteriormente por outros trabalhos no âmbito deste tema [Campos03] [Campos07]

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 ambiente urbano é criado recorrendo ao modelador interativo VUEMS [Donikian97]. Os modelos são criados sobre uma definição de terreno em mapa de elevação. Em simultâneo com a criação do modelo da cidade é gerado um modelo semântico. A criação do modelo semântico de acordo com o mapa rodoviário é também utilizada em outros trabalhos relacionados [Campos07] [Bayarri96] [Pareja99] [Thomas00]. O uso de um modelador interativo não permite a geração de um ambiente rodoviário de forma automática e eficiente.

Galin et al. apresentam um método para a criação de estradas [Galin10]. É utilizado o conceito de anisotropia para determinar o melhor percurso para a estrada a projetar. É descrito que o processo de gerar uma via é dividido em duas fases. Primeiro são determinados pequenos segmentos entre o ponto de origem e o ponto de destino segundo uma grelha com espaçamento regular. Na segunda fase, mediante os segmentos determinados anteriormente sobre a grelha regular, é calculado o traçado do eixo da via em planta. Na determinação do traçado do eixo da via utilizam técnicas de suavização de curvas apresentadas por Walton et al. em 2005 [Walton05]. A metodologia apresentada suaviza de forma aproximada o traçado em altimetria, utilizando modelos paramétricos genéricos. Para avaliar as diferentes direções do traçado da via, em cada ponto de análise, recorrem a uma função que avalia o custo de cada hipótese. Na função de custo é considerada a construção sobre água, a inclinação longitudinal da via e a curvatura do traçado, resultante do enquadramento da morfologia do terreno.

Galin et al. em 2011 apresentam um método de geração hierárquica de uma rede de estradas em 3 níveis: rede de autoestradas, rede primária e rede secundária. A geração da rede rodoviária global e iniciada pela criação da rede

de autoestradas, seguida da rede principal e por último a rede secundária. No final, recorrem a um método para efetuar a junção entre as diferentes redes viárias criadas, eliminando a existência de estradas muito próximas, com diferentes origens e partilham o mesmo destino [Alt92]. Na implementação apresentada o modelo rodoviário gerado é organizado em apenas 2 níveis: geração da rede topológica, e geração do modelo geométrico. Para a geração da rede, a metodologia apresentada, parte dos seguintes dados de entrada: terreno, cidades e pontos de referência, definidos pelo utilizador.

Em [OpenDrive10] é proposto um *standard* (linguagem declarativa) para especificação de uma rede viária utilizando uma sintaxe própria em XML. Para a criação do modelo da rede viária é utilizado o modelador interativo *Road Designer*.

Ferramentas tradicionais, como por exemplo CyberCity, Citygen, podem ser utilizadas para criar modelos virtuais, mas ainda continua a ser uma tarefa complexa e morosa [CyberCity12] [Kelly, 2008].

Um simulador de condução para fins científicos requer a preparação de bases de dados com a descrição de cenas de regiões geograficamente extensas, com estradas corretamente modeladas em função das normas e tradições do país que se pretende simular.

Para além do modelo visual é necessário garantir a coexistência de uma base de dados semântica, estruturada de forma hierárquica para consulta exaustiva em tempo real durante a simulação. É também necessário que os modelos de ambientes rodoviários sejam modelados segundo normas de traçado e adequados a simulação de condução para fins científicos.

Os trabalhos relacionados estudados são tipicamente desenvolvidos para a indústria do entretenimento e orientados para a qualidade de visualização.

Na maior parte destas implementações o modelo rodoviário gerado não verifica as normas de traçado sendo uma das lacunas que se pretende colmatar com a elaboração desta proposta. Para atingir esse objetivo, perspetiva-se que a integração dos métodos utilizados em engenharia de vias venham permitir obter traçados semelhantes aos encontrados em traçado real.

Por outro lado é necessário criar em paralelo com o modelo virtual, uma descrição semântica associada à rede rodoviária de modo a que o modelo virtual possa ser utilizado em simulação em tempo real. Esta descrição semântica é importante para a simulação, na medida em que permite a inclusão de atores, assim como também para a produção de relatórios relativos à tarefa de condução.

Parte dos trabalhos relacionados implementam modelos virtuais de cidades, mas não permitem a implementação ou reprodução em ambiente virtual de traçados reais, sendo uma das funcionalidades que também se pretende abranger.

Por isso torna-se necessário desenvolver uma nova abordagem na geração de ambientes rodoviários adequados

para simulação de condução com fins científicos de forma automática.

3. PROPOSTA

A metodologia que se apresenta, integra num único processo, a criação automática de um ambiente rodoviário, desde os nós de ligação até à geração do modelo geométrico. Esse ambiente rodoviário deverá ser gerado com base em normas de projeto de redes de estradas do País ou região que se pretende representar.

O preparador do ambiente virtual poderá interagir em qualquer fase do processo, de forma a permitir adequar o modelo de saída às especificidades concretas do projeto a implementar, obtendo modelos totalmente controláveis. No caso de não existir uma especificação inicial, será possível gerar de forma automática um ambiente rodoviário completo.

Com base em métodos de traçado de engenharia de vias, tal como em projeto de vias, será possível obter modelos rodoviários realistas, semelhantes aos encontrados em traçado real e adequados para simulação de condução com fins científicos.

Na proposta que se apresenta, a criação do ambiente rodoviário ao longo de todo o processo é realizada de forma hierárquica, como se ilustra na figura 1.

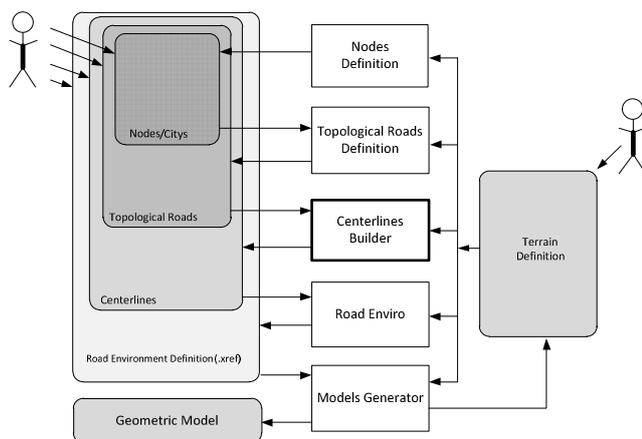


Figura 1. Diagrama funcional

No final do processo, obtém-se uma definição com toda a especificação do ambiente rodoviário, segundo uma ontologia, que corresponderá à definição semântica. Esta definição permitirá no final do processo de geração da definição do ambiente criar o respetivo modelo geométrico, representado pelo bloco da esquerda na figura 1 (*Road Environment Definition*).

Na figura 1 apresenta-se o diagrama funcional da metodologia concebida para a geração de um ambiente rodoviário de forma automática.

Na hierarquia do diagrama que se apresenta, a geração da definição de um ambiente rodoviário inicia-se pelo módulo *Nodes Definition*. Os módulos adjacentes dão continuidade ao processo de geração de um ambiente rodoviário.

Sendo que o módulo *Centerline Builder* é aquele que se destaca mais nesta proposta, pois será onde se centrará mais a inovação na criação de uma definição de uma via, mapeado os métodos utilizados em engenharia de vias. Como se pode ver no diagrama, na coluna da esquerda tem-se no final dos processos obtêm-se uma única definição de ambiente rodoviário que permitirá que o módulo *Models Generator* gere automaticamente o modelo visual. Também se pode observar no diagrama a possibilidade de interação do utilizador em qualquer fase do processo. Por último pode-se ver que em todo o processo a definição do terreno é considerada como um dos dados de entrada, podendo ser editada pelo módulo de geração do modelo visual, permitindo que a nova definição seja utilizada como modelo de entrada num novo processo.

O módulo *Nodes Definition* tem como função, no caso de não existir nenhuma definição, gerar automaticamente uma definição simplificada de nós sobre uma área de terreno. A definição de terreno poderá ter associado um conjunto de restrições que deverão ser consideradas. Cada nó associado à sua posição geográfica poderá ter um conjunto de parâmetros que possam ser úteis no processo de geração do modelo visual assim como posteriormente para a simulação. Um dos parâmetros a ter em consideração depois na geração da rede topológica será o fator de importância associado a cada nó [Kelly08] [Weber09].

O segundo bloco do diagrama *Topological Roads Definition* tem como objetivo criar uma definição simplificada de um mapa rodoviário de alto nível e associado a esse mapa, uma definição topológica da rede. Este módulo terá a capacidade de criar nós adicionais de interligação, neste caso com fator de importância zero. Na imagem da figura 2, apresenta-se um exemplo de estudo de uma definição de uma rede topológica gerada por este módulo.

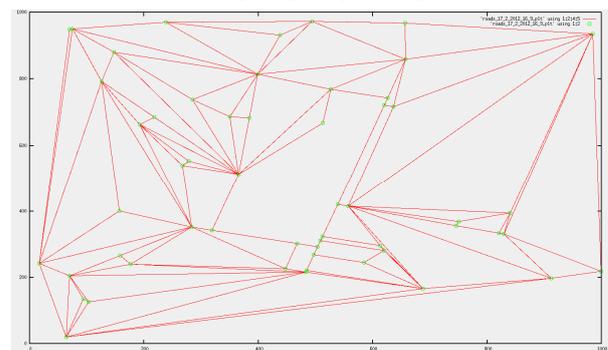


Figura 2. Rede Topológica

Além de definir que nós se interligam através de estradas, será necessário definir um conjunto de parâmetros, relativos ao perfil da via, necessários para a modelação e simulação. Esta análise começará por tratar as redes e os nós com maior importância à semelhança do que acontece noutros trabalhos [Teoh08] [Weber09] [Galin11]. Após a geração da rede topológica, será necessário proceder à sua otimização, simplificando a topologia da rede, podendo inclusive criar nós de ligação adicionais [Alt92].

O terceiro bloco, *Centerlines Builder* é o módulo irá determinar o traçado de cada estrada e o qual se tenciona dar mais ênfase nesta proposta. A partir da definição do mapa rodoviário gerado anteriormente, este módulo, tem como objetivo gerar o eixo da via de cada estrada. Para obter o traçado da via, neste processo serão mapeados os conceitos associados ao projeto de vias rodoviárias, assim como respeitar as normas de traçado. [EP94] [França11].

O processo de geração do eixo da via passará por criar um corredor a diferentes escalas entre o ponto de origem e o ponto de destino, sobre uma grelha regular. Sobre esse corredor com maior resolução, tipicamente à escala de 1:200, serão criados um conjunto de alinhamentos retos que depois permitirão obter o traçado da via em planta, como ilustrado na figura 3.

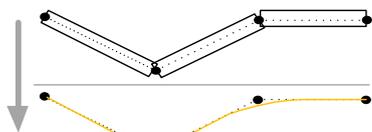


Figura 3. Eixo da via

Para calcular o traçado da via, neste processo serão utilizados métodos de engenharia de vias para obter o eixo da via, como por exemplo, poderá ser utilizado o conceito de linha dos zeros para determinar a diretriz de uma via sobre as diferentes curvas de nível do terreno [França11]. A linha dos zeros por definição é o lugar geométrico dos pontos de interseção da normal às curvas de nível do terreno com o eixo da via, como se ilustra na figura 4.

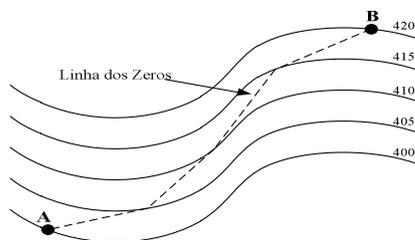


Figura 4. Linha dos Zeros

A linha dos zeros possibilita que num traçado de uma estrada entre dois pontos com cotas diferentes, A e B, o eixo da via mantenha sempre a mesma cota relativamente ao terreno. Para traçar a linha dos zeros é necessário inicialmente calcular a dimensão do segmento entre curvas de nível adjacentes, tendo posteriormente esses segmentos comprimento constante ao longo do traçado sobre as diferentes curvas de nível.

Posteriormente será calculado o traçado em altimetria de acordo com o modelo de terreno utilizado.

O bloco *Road Enviro* será responsável por gerar informação adicional relativa ao meio envolvente à rede rodoviária. Esta informação será útil para gerar o modelo visual e criar um ambiente rodoviário envolvente o mais real possível, com objetos que tradicionalmente se encontram no meio rodoviário que se pretende simular. O principal

objetivo deste módulo é tornar o meio rodoviário envolvente mais imersivo, de modo a ser utilizado em experiências de realidade virtual com fins científicos.

O último bloco, *Model Generator* a partir das definições geradas pelos módulos anteriores, irá gerar o modelo visual do ambiente rodoviário. Para a criação do modelo geométrico, pretende-se mapear métodos desenvolvidos em trabalhos anteriores [Campos07]. Toda a informação gerada na especificação por cada um dos módulos apresentados anteriormente constitui a definição global do ambiente, sendo depois utilizada por este módulo para criar o modelo geométrico, como se ilustra na figura 5.

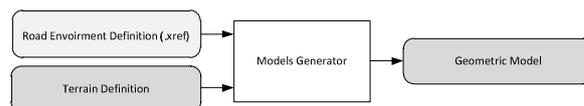


Figura 5. Geração do modelo

A definição semântica é descrita sobre uma ontologia no ficheiro de definição do ambiente rodoviário, sendo obtida como resultado da execução de todo o processo de geração do modelo virtual.

No processo de geração do modelo de ambiente rodoviário é considerada a especificação do modelo de terreno, que no final, estará em concordância com o respetivo modelo geométrico da rede rodoviária e todos os elementos 3D envolventes.

Os modelos de ambientes rodoviários serão gerados em pré processamento, permitindo exportar os modelos obtidos para outros sistemas de simulação.

Avaliação

A avaliação e validação da metodologia proposta terão como base a realização de estudos experimentais de geração de ambientes rodoviários para fins concretos. Isso implicará o recurso a métodos de avaliação experimental em que envolva o recurso à metodologia desenvolvida para geração de ambientes rodoviários. Tal como referido anteriormente, um dos objetivos da metodologia proposta é que seja passível de ser usada por utilizadores não especialistas em engenharia informática. Para tal ira-se recorrer a utilizadores das diferentes áreas científicas, que tipicamente realizam trabalho experimental em simuladores de condução. Para avaliar a usabilidade da metodologia desenvolvida poderão ser utilizadas métricas como: o tempo e o grau de conhecimento prévio necessário para gerar um ambiente rodoviário, assim como o nível de interação necessário para produzir um modelo de ambiente que cumpra os requisitos exigidos.

A avaliação de uma rede de dados ou rede rodoviária é uma tarefa que deverá ser realizada para permitir classificar a rede obtida. Um método a utilizar consistirá em calcular a distância mínima entre todos os pares de nós [Black06]. Para cada par de nós, essa distância permitirá associar um índice a cada estrada envolvida nesse percurso. Esse índice, associado a parâmetros de construção e utilização da rede, permitirá avaliar a eficiência da rede gerada.

O custo global da rede é calculado, considerado o custo de deslocação de veículos de todos os nós para todos os nós e assumindo que essa deslocação é realizada pelo percurso mais curto da rede rodoviária.

Para uma estrada (AB), como se pode ver pela equação (4), o valor adicionado ao segmento AB é determinado em função da sua extensão e do fator de importância dos nós extremos da ligação (nó de origem/ nó de destino). Sendo o segmento AB o percurso associado apenas a uma estrada.

$$\text{Peso(AB)} = \text{dist(AB)} * (\text{FIorig} + \text{FIdest}) \quad (4)$$

onde Peso(AB) corresponde ao valor a adicionar ao segmento AB, dist(AB) é a distância entre o ponto A e o ponto B na estrada, FIorig é o fator de importância do nó de origem e FIdest o fator de importância do nó de destino.

A avaliação do traçado de uma estrada poderá ser realizada, utilizando um conjunto de métricas como por exemplo: extensão das retas, média do raio de curvatura, integral da curvatura, concordância em altimetria, entre outras. Esta análise permitirá validar se a estrada gerada pela metodologia proposta possui um aspeto semelhante ao encontrado em traçado real.

Outra abordagem, consiste em recorrer à colaboração da engenharia de vias para avaliação do traçado obtido.

4. CONCLUSÕES

Um simulador de condução requer a preparação de modelos de ambientes rodoviários corretamente modelados, semelhantes aos encontrados em traçado real e representativos do País ou região que se pretende simular.

A metodologia apresentada permitirá gerar qualquer tipo de ambiente rodoviário de forma automática, com base em dados de projeto, de sistemas de informação geográfica ou de outras fontes. Será possível gerar qualquer tipo de ambiente rodoviário, como por exemplo: rural, urbano e extraurbano.

Para além destas funcionalidades, a metodologia apresentada permitirá gerar modelos com elevado realismo, de forma automática e de interação simples para o preparador da experiência. Pretende-se que o utilizador possa interagir no processo de modelação de modo a definir a aparência do modelo final para se adequar a cada caso específico, produzindo modelos totalmente controláveis.

A metodologia apresentada, na geração do ambiente rodoviário permite entrar em consideração com a definição do modelo do terreno. Está contemplada, na proposta apresentada, a adequação do modelo de terreno ao ambiente rodoviário gerado após a geração do modelo geométrico.

Estas características permitirão obter uma grande diversidade de modelos de redes viárias, com vias corretamente modeladas, com inclusão de sinalização de trânsito e mobiliário urbano envolvente. Possibilitarão obter modelos

de excelente qualidade, reduzindo drasticamente o trabalho e os custos envolvidos na preparação de experiências de simulação de condução, uma vez que evitam a modelação integral do ambiente por parte do utilizador, principalmente a prévia obtenção do traçado da via a partir de dados de projeto.

A avaliação a realizar incluirá o estudo do traçado das estradas criadas, do modelo de rede viária, assim como de todo o processo automático de geração do modelo rodoviário.

A realização deste trabalho vem trazer um contributo inovador na especificação e geração automática de ambientes rodoviários destinados a simulação em tempo real. No cálculo do traçado de uma estrada pretende-se desenvolver uma nova abordagem, recorrendo a métodos desenvolvidos em trabalhos semelhantes e principalmente a métodos utilizados em projeto de vias.

Uma das principais vertentes de aplicação desta metodologia será a geração de ambientes rodoviários adequados à simulação de condução, permitindo a realização de estudos científicos nas mais diversas áreas.

No futuro próximo será possível apresentar resultados concretos obtidos segundo a metodologia apresentada.

5. AGRADECIMENTOS

A realização desta proposta 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 Faculdade de Engenharia da Universidade do Porto.

6. REFERENCIAS

- [Alt92] ALT H., GODAU M.: Measuring the resemblance of polygonal curves. In Symposium on Computational Geometry, pp. 102–109. 1992.
- [Autodesk12] Autodesk, Civil3D, aplicação de projeto de vias de comunicação, informação disponível em: <<http://www.autodesk.pt>>, em Janeiro 2012.
- [Bayarri96] Bayarri, S; Fernadez, M; Perez, M; Virtual Reality for driving simulation-Bayarri, Vol. 39, n.º 5, Communications of the ACM, 1996.
- [Black06] Black, P; “Dijkstra's algorithm”, in Dictionary of Algorithms and Data Structures, documento disponível em: <<http://www.nist.gov/dads/HTML/dijkstraalgo.html>>
- [Campos03] Campos, C.; Cunha, V.; Leitão, J.; Geração de Ambientes Rodoviários para Simulação de Condução, 12º encontro Português de Computação Gráfica, p. 143-147, Outubro de 2003.
- [Campos07] Campos, C.; Leitão, J.; Rodrigues, C.; Modelação de Ambientes Rodoviários de Grandes Dimensões, 15.º Encontro Português de Computação Gráfica, Outubro de 2007.
- [Chen08] Chen,G.; Esch,G.; Wonka,P.; Muller,P.; Zhang,E.; Interactive Procedural Street Modeling, *ACM SIGGRAPH*, 2008.

- [CyberCity12] CyberCity 3DTM; CyberCity: modelador interativo de ambientes virtuais, <<http://www.cybercity3d.com/>>, Junho 2012.
- [Donikian97] Donikian, S.; VUEMS: A Virtual Urban Environment Modeling System, Computer Graphics International, pp. 84-92, 1997.
- [EP94] Normas de traçado, Junta Autónoma das Estradas, Atual Estradas de Portugal (E), ISBN-96379-6-2, 1994.
- [França11] França, A.; Apontamentos de vias de comunicação, Faculdade de Engenharia da Universidade do Porto, consultado em 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.
- [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 2006.
- [Leitão00] Leitão, J.; Agentes Autónomos Controláveis em Simuladores de Condução, tese para obtenção do grau de Doutor, pela Faculdade de Engenharia da Universidade do Porto, em Setembro de 2000.
- [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 Congresso de Engenharia Informática, 1999.
- [Paris01] 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.
- [Smelik09] Smelik, R.; Kraker, J.; Groenewegen, S.; Tutenel, T.; Bidarra, R.; A Survey of Procedural Methods for Terrain Modelling, Proceedings of the CASA Workshop on 3D Advanced Media In Gaming And Simulation, 2009.
- [Sun02] Sun, J.; Baciú, G.; Yu, X.; Green, M.; Template-Based Generation of Road Networks for Virtual City Modeling, VRST'02, pp 33-40, NY USA, 2002.
- [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.
- [Walton05] Walton D., Meek D.: A controlled clothoid spline, *Computer & Graphics*, vol. 29, 2005.
- [Weber09] Weber, B.; Müller, P.; Wonka, P.; Gross, M.; Interactive Geometric Simulation of 4D Cities, *Eurographics*, vol. 28, 2009.

CaS: Collection-aware Segmentation

Raquel Costa Manuel J. Fonseca Alfredo Ferreira
INESC-ID/IST/Technical University of Lisbon
Lisboa

rgcc@ist.utl.pt {mjf,alfredo.ferreira}@inesc-id.pt

Resumo

Ao longo dos tempos, a segmentação tem provado ser um desafio devido à sua subjectividade. A segmentação depende não apenas do domínio em causa mas acima de tudo da interpretação que os humanos fazem do objecto. Para cada contexto, diversas soluções específicas foram propostas com diferentes objectivos, limitações e vantagens. Neste trabalho propomos ultrapassar algumas dessas limitações usando o algoritmo de segmentação Collection-aware Segmentation (CaS). Este algoritmo identifica segmentos de objectos em colecções baseados na sua individualidade nessa colecção. Para esse efeito realizámos um conjunto de testes para compreender como as pessoas segmentam objectos numa colecção. A partir dos resultados destes testes desenvolvemos os algoritmos Adaped-CaS e Geons-augmented CaS. Avaliações experimentais com utilizadores mostraram que a abordagem proposta produz segmentações com significado para os humanos.

Abstract

Segmentation has always proven to be a challenge because of its subjectiveness. It depends not only of the application domain but also most on the human interpretation. To each context, several specific solutions were proposed with different goals, limitations and advantages. With this work we propose to overcome some of those limitations by improving the Collection-aware Segmentation algorithm (CaS). This algorithm identifies segments of objects in collections based on their individuality among the collection in which the objects belong. To that end we performed a set of tests to understand how humans segment a collection of objects. From the results of these tests we developed the Adaped-CaS and the Geons-augmented CaS algorithms. Experimental evaluation with users revealed that our approach produces a segmentation that is meaningful for humans.

Keywords

3D Object Segmentation, 3D Object Collections, Automatic Segmentation, Similarity Estimation

1. INTRODUCTION

Each human being interprets the environment from his own point of view. This generates an huge range of possible interpretations of the world, and his components. Thus, segmentation of objects may vary from individual to individual. Indeed, it and has been subject of studies in different areas, from mathematics to philosophy.

Over the past decades object segmentation has been also tackled within the computer graphics domain, as a result of the growing number of 3D objects in digital format and the widespread of applications that use them. Many computer graphics applications, as animation, collision detection, object indexing and retrieval use the segmentation approaches as a stage of the core process. However, most of the existing object segmentation techniques are domain or context dependent. These perform well in the domain and context for which they were designed for, but not so good in other domains or contexts.

To overcome this limitation we adopt a different approach

that extends the Collection Aware Segmentation (CaS) method. This was originally proposed in [Ferreira09] embedded in a solution for indexing and retrieval of 3D objects. In this paper we extend CaS to make it a stand-alone segmentation technique that produces meaningful results to the users, independently of the object domain.

With the present work we isolated the CaS approach of the indexing and retrieval application, thus creating a segmentation algorithm that is application independent. As the original CaS approach, the Adapted CaS is based on the Hierarchical Fitting Primitives [Falcidieno06] algorithm and uses Spherical Harmonics shape descriptor [Kazhdan03]. By adding geon analysis [Biederman87], we improved the algorithm, achieving better segmentation, closer to human perception.

To evaluate the proposed approach, we conducted an experimental evaluation where several tests were performed. The first test focusing on understanding how humans segment 3D objects. From the results of this test we devel-

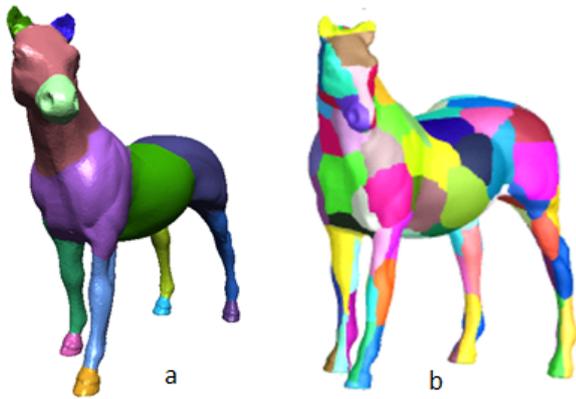


Figure 1: Two different types of segmentation: a) Part Based Segmentation; b) Surface Based Segmentation. (Figure taken from [Shamir08])

oped and refined our approach. Then, to evaluate the efficacy and effectiveness of the approach we executed performance tests to study execution time and memory requirements. Finally, to validate the quality of the segmentation, we organized a test where users were asked to compare the results of manual segmentation made by humans with results produced automatically by segmentation algorithms.

In the remaining of this document we start with a brief presentation of related work on three-dimensional object segmentation. Then we describe the original Collection Aware Segmentation algorithm, followed by its evolution and the explanation on detail of how these work. On section 4 we describe the evaluation tests and discuss the corresponding results. In the last section, we present the conclusions of this work and reflect on future research paths on this topic.

2 RELATED WORK

2.1 Categorization of segmentation approaches

Some authors [Agathos07, Shamir08] classify the segmentation techniques in two categories depending on the kind of segmentation accomplished, represented in Figure 1. The part based segmentation is closer to user perception and divides the object into sub-components, while the surface based segmentation are accomplished by analysing the surface shape features.

Among these two types, the 3D object segmentation has been widely studied and different solutions that uses distinct approaches have been proposed. Some of the existing and more relevant are presented next.

2.2 Region Growing and Clustering

The region growing methodology follows a exploratory approach that starts by visiting a seed (a face of the mesh), then agglomerates the adjacent faces by transversing the mesh and it stops when it reaches a stop condition. Such

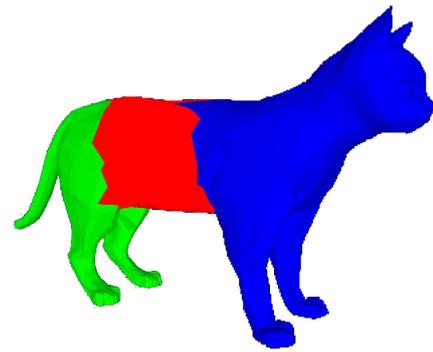


Figure 2: Hierarchical decomposition with fuzzy area represented in red. (Figure taken from [Katz03])

condition can be the segment convexity. Then the process is restarted using a non visited face as a new seed. The approach proposed by Zuckerberger on [Zuckerberger02], uses a depth first or breadth first search to transverse a graph that represents the mesh and the seed where it starts is a node of this graph. From this transverse is creates a segment and then, it restarts transversing on an unvisited node of the graph forming a new segment.

In a similar way, Attene et al. [Falcidieno06] approach also agglomerates faces, but instead of creating patch by patch, it creates all patches simultaneously. These are organized as an hierarchical tree in which the leafs are the mesh triangles and the root is the entire object. This tree is built bottom to top and clusters adjacent faces with the minimum merging cost that can be calculated on different ways. Attene et al. uses primitives by fitting them to the resultant cluster.

Previously, Katz and Tal [Katz03] had proposed to generate, in an iterative clustering, various results of segmentation given a number of clusters, and then chooses which is the best segmentation. It starts by creating a representative group of clusters and then each adjacent face is clustered until it reaches a ray r between the seed and the limit of the patch. This is used to agglomerate the faces that have more probability of belonging to a given segment, thus creating fuzzy areas, illustrated Figure 2. Lastly it is needed to transform the fuzzy decomposition in a final decomposition by refining the limits.

2.3 Skeleton Based

This approach uses the skeleton of the object to determine the segmentation. The most used method to extract the skeleton is the Reeb Graph. For instance, Tierny et al. [Tierny07] extract an enhanced topological skeleton by finding feature points located on object extremities using the geodesic distances. These are used to create a function that indicates the distance from a given point of the mesh to each feature point of the mesh and from there is built the Reeb graph. After having the skeleton, the object is seg-

mented in the areas where each skeleton node corresponds to a segment of the object.

2.4 Geometry and Structure-Based

The Taylor and Plumber algorithms, by Mortara et.al. [Mortara03] uses the object shape to perform segmentation based on geometry and structure. These algorithms detects tubular features by blowing bubbles that starts on a seed that is predefined and stops blowing when it finds an abrupt change on the object shape, such as a bifurcation. Later, Mortara et al. [Mortara06] used the Plumber approach to find the tubular zones in objects that represents the human body.

2.5 Feature Points and Core Extraction

In an attempt to overcome the pose invariant limitation a new approach arises called Feature Points and Core Extraction that was proposed by Katz et al. in [Katz05]. This initially creates a pose invariant representation of the object, then extracts the feature points and the core. After finding the core it is necessary to extract the rest of the segments which is done by matching each part to the feature points. In the end, it reverse the initial process so the object can come back to the same shape.

2.6 SDF

A different approach was proposed by Shapira et al. in [Shapira08]. They use the shape diameter function (SDF) for segmenting objects. In short, this gives the diameter of an object in a neighbour of a point and is used to merge points that have the same or close diameter values using a histogram.

2.7 Automatic Segmentation of 3D Collections

The above referred approaches, as most existing approaches, segment 3D objects individually. The segmentation is performed object by object individually, instead of segmenting various objects simultaneously. Indeed, this is a relatively recent concept: to accomplish automatic simultaneous segmentation of sets of objects. The approaches that use the automatic segmentation of 3D collections use the information of similar objects to improve the results. The group of Thomas Funkhouser in Princeton is one of the groups that is already studying this subject. They presented an algorithm [Golovinskiy09] that builds a graph whose nodes represents the mesh faces and the edges represents the edges of the mesh that connects adjacent faces of the same object. They also represent the correspondence between the faces of different meshes. In the next step the algorithm executes a hierarchical clustering of the graph, were the adjacent faces of the same model, and the correspondent faces of different models are going to probably belong to the same segment.

2.8 Discussion

Several segmentation approaches have emerged as decomposition of 3D objects as is used in many different applications in distinct domains that require different segmentations. This also makes the task of evaluating the ap-

proaches hard to perform, due due the large number of approaches. Nevertheless, Funckhouser and his team defined a benchmark for 3D mesh segmentation [Chen09], compared seven mesh segmentation algorithms and draw some interesting conclusions.

However, a common limitation is the need of inputs provided by the users in order to decompose the objects. Some need to predefine the seeds like in Region Growing and Iterative Clustering Approaches. Others to select the level of the hierarchy generated by the segmentation approach, these are all the approaches that produce a hierarchical segmentation. The result of the segmentation has more segments that it should have producing over segmented result. Some approaches try to overcome this limitation by using post-processing stage that removes the extra segments by merging them to others or by initially predefining a maximum number of segments.

The individual segmentation of objects can be considered a limitation if we consider segmenting a collection of objects. If the segmentation decomposes object by object as it is in the major approaches, then it can take more time to decompose the entire collection than if segmentation was performed simultaneously.

In order to overcome the limitations highlighted above, we propose a different solution based on the CaS algorithm, presented in the next section. This is a segmentation approach that will be application independent.

3 COLLECTION-AWARE SEGMENTATION

In this section we present a distinct approach for segmenting 3D objects whose main goal is to overcome some limitations found on existing approaches. We extracted the Collection-aware Segmentation (CaS) algorithm from the indexing and retrieval context where it was embedded [Ferreira09], as a step of a complete solution, and extended it to a fully fledged 3D object segmentation algorithm. This led to the original CaS and to the evolution for Geons-Augmented CaS.

3.1 Original CaS for Retrieval

The original CaS, integrated in a retrieval solution, did not indeed produced any object segmentation *per se*. Instead, it decomposes all objects in a collection and stores their sub-parts in a shape pool, used then for indexing the collection.

This approach, depicted in Figure 3 has two main stages: the initialization stage and the iteration stage. In the first stage the foundations of the segmentation are computed and loaded into memory, while in the following stage the objects of the collection are iteratively segmented into sub-parts.

The initialization starts by generating, for each object on the collection the respective hierarchical segmented mesh (HSM) using the Hierarchical Fitting Primitives (HFP) as proposed by Attene et al. [Falcidieno06]. This approach consists on merging clusters and then fitting them to primitives to create the final clusters. Next it saves the resulting HSM on the HSM Set, producing a set of seg-

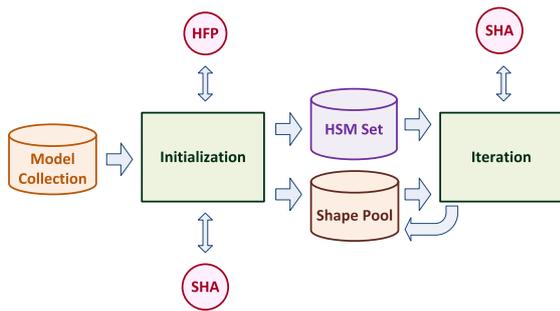


Figure 3: Overall architecture of the original CaS, with the two main stages: initialization and iteration.

mented meshes. Then is computed for each mesh, the object signature using the spherical harmonics shape descriptor (SHA) introduced by Kazhdan et al. [Kazhdan03]. The resulting signatures are stored in the shape pool for further processing.

In the iteration stage, each signature on the shape pool is verified for its uniqueness, thus being flagged for decomposition. This means that if the number of similar elements is below a pre-defined threshold, it is considered as unique, thus decomposable. This similarity is computed using the differences between object signatures.

If the object is decomposable, then it is decomposed by exploring the corresponding HSM on the HSM Set and getting its child nodes, computing their signatures and adding them to the Shape Pool. If the object is not decomposable, it passes to the next element on the shape pool. The iteration stage finishes when there are no more decomposable segments on the shape pool and the algorithm ends by returning the entire shape pool. Indeed, it does not produce segmented versions of the objects in the model collection.

3.2 Adapted CaS for Decomposition

To produce segmentations for objects in a collection, we adapted the original CaS for decomposition. As in the original, the Adapted CaS consists on two main stages: the initialization and the iteration. These stages have similar purpose to the those on the original CaS, but are slightly different and comprise new data structures, as depicted in Figure 4.

The Adapted CaS decomposer receives as input the entire collection of objects on the initialization stage. On this stage, for each object, it computes the corresponding HSM using HFP. This generates a binary tree that is built bottom to top. It starts on the leafs that represents the triangles of the mesh, then for each pair of neighbours it calculates the merging cost. This cost is calculated by fitting a primitive (cylinder, cone, sphere and plane) to the resulting cluster, then it compares the values and clusters the pairs that have less cost. This new cluster generated from the clustering are saved as a parent node of these clusters, in the tree.

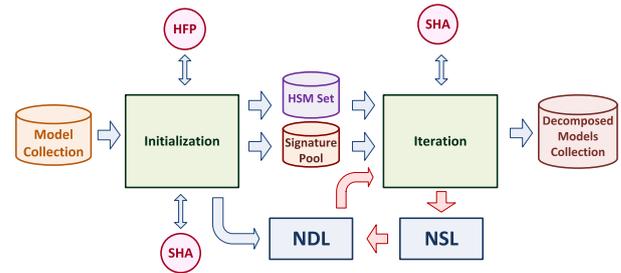


Figure 4: Overall architecture of Adapted CaS, with the new NDL and NSL structures.

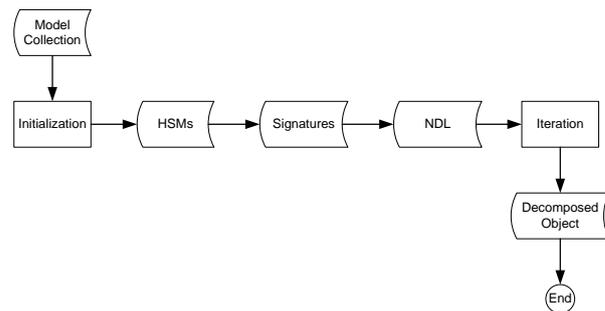


Figure 5: High level fluxogram of Adapted CaS.

It repeats the process until it reaches the top of the three where it is saved the entire object. These trees, stored in the HSM set, will be transversed during the iteration stage to produce the decomposed objects. Figure 5 presents an high level fluxogram of the decomposition process.

Since the decomposition algorithm is based on the singularity of an object, the second step of the initialization stage is to compute the SHA signatures of each object. These are shape descriptors, that is, a numerical representation of the object in a multidimensional space. Using these representations makes the comparison between segments easier and this comparison is used to label a segment as decomposable or not. Thus, each computed signature is stored on the signature pool for later use.

The main goal of this Adapted CaS technique is to present the collection of objects decomposed. Thus the shape poll is no longer necessary to and was replaced by the Signature Pool. It is only necessary to visit this set to label a segment as decomposable or not. Without the shape pool, no longer exists a second HSM build during the iteration stage. Instead two lists are used: the Non Segmented List (NDL), that contains the segments to be processed and the segments that are not labeled as decomposable and the Non Segmented List (NSL) that contains the segments that are going to be processed on the next iteration. So, at the end of the initialization stage, the NDL contains all the objects in the collection.

During its execution, the iteration stage traverses the

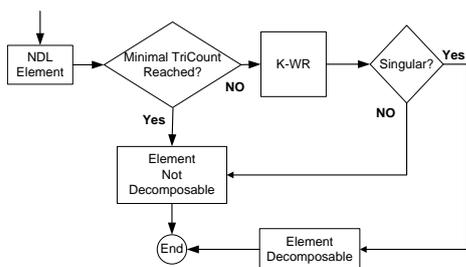


Figure 6: Decomposition fluxogram of Adapted CaS.

HSMs, level by level, that is, each new iteration corresponds to a level on the hierarchy. In order to support the segmentation, the iteration stage uses the NDL. In practice, each iteration of the stage is an iteration to the NDL. So, this stage starts on the first element of the NDL and verifies if it is decomposable or not.

To segment the objects based on the collection where they belong, it is necessary to compare each sub-part with all other sub-parts. Figure 6 illustrates the steps for labelling a segment as decomposable or not. It first verifies if it has reached the minimal triangle count by calculating the number of triangles between both child nodes and compares this with a previously defined value - the minimum triangle count. If the triangle count is below this threshold, the segment is flagged as non-decomposable. This step prevents from reaching the triangles of the mesh.

In case the minimum triangle count have not been reached, the element passes to the next step in the decomposition process. This step consists on the execution of a K-Within Range (K-WR) search. This algorithm returns the first K-elements whose signatures are within a predefined range and is used to verify if a segment is singular. For that end, two thresholds were previously defined: the similarity and similar count thresholds. The first threshold is used to verify if two objects are similar or not, to be similar the distance of two objects on the multidimensional space has to be less than the similar threshold, this is computed using the segments signatures and the distance between them. Then, the second threshold is used when comparing a segment with all the segments that are on the signature pool, it cannot have more similar segments than the similar count threshold. With this result it is verified if the segment is singular, to be singular the number of similar segments has to be above the similar count, if is that the case, then the segment is labelled as decomposable. If not, it labels as not decomposable.

If the result is the segment being decomposable, then it is going to decompose it, for that, it goes to the respective HSM and get his child nodes. After having the child nodes, it calculates their respective signatures, adds them to the signature pool and insert both nodes on the Non Segmented List (NSL), this list contains the segments that are going to be processed on the next iteration and removes the

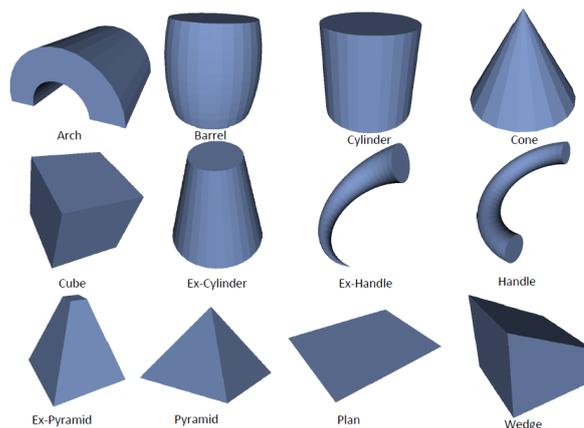


Figure 7: Group of geons used on the Geons-augmented CaS.

processed element from the NDL passing to the next element on the NDL that also will be processed. Also in the case of the result being not the decomposable it passes to the next element on the NDL.

An iteration ends when it reaches the end of the NDL but the iteration stage may ended or not. This depends on the NSL, if the NSL is not empty it means that are new segments to be processed. So, it appends all the elements on the NDL, remove them from the NSL and restarts a new iteration by visiting again the first element on the NDL.

The iteration stage and the entire algorithm ends when it has reached the end of the NDL and the NSL is empty, meaning that there are no more segments to be decomposed. So, the approach ends by returning the entire collection of decomposed objects.

3.3 Geons-augmented CaS Decomposer

During the execution of the manual segmentation test using HFP, one of the complains of the users was that some objects were over segment. This happens because it was used the HFP approach. After executing the adapted CaS approach with different similarity and similar count thresholds we notice that this problem still happens on this approach, so, to overcome this limitation it was introduced a new feature. The introduction of the geons to verify if an object is decomposable or not.

The geons are simple 3D objects, like cylinders, cones, cubes. The theory proposed by Biederman on [Biederman87] called "Recognition by Components" states that, like English words that are constituted by a number of phonetics, complex objects can also be composed by these simpler 3D objects, that is, by segmenting complex objects we get these simpler objects. Thus we added a set of geons, partially depicted in Figure 7, to improve decomposition results according to human perception.

Since the CaS approach uses the HFP that uses primitives to calculate the cost of merging clusters and there are some

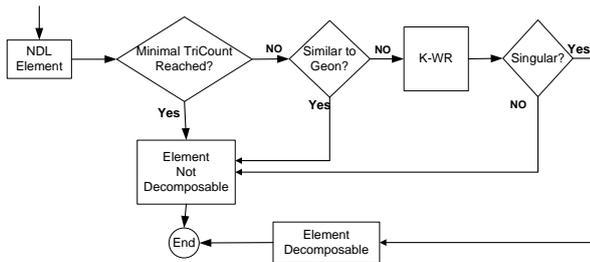


Figure 8: Decomposition fluxogram for Geons-Augmented CaS.

of the geons that are the same as these primitives, using these objects can help to stop the segmentation when a geon is found. Figure 8 represents the decomposition fluxogram for the Geons-augmented CaS. The process is similar to the one described in previous section and illustrated in Figure 6. The main difference lays after the first comparison. If the minimal triangle count of the child nodes is above the minimum triangle count predefined, then segments are compared with the set of geons. This is done by comparing the signature of the segment with each precomputed geon signature. Then, if it is not similar, it proceeds as the previous description, that is, it executes the K-WR search. In the case of being similar to any of the geons, then the segment is labelled as not decomposable.

4 EXPERIMENTAL EVALUATION

To validate and evaluate the proposed approach implementation, we verified the efficiency and effectiveness of the algorithm. Thus, several tests were organized, from performance tests to tests with users which involved a sample group of twenty people.

4.1 Manual Segmentation Tests

This test was used to understand the human interpretation of an object, more precisely, how humans segment the objects. To that end we used a small set of randomly chosen objects of the Engineering Shape Benchmark (ESB) [S.Jayanti06]. A preliminary conclusion we made from this test was that familiar objects are easier to decompose than those people see for the first time. Other conclusion was that users segment objects consistently. But draw such conclusions was not the primary goal of the tests and further studies should be made to validate such observations.

Based on the results obtained in these tests it was possible to refine the approach and also make it automatic by defining the similarity and similar count thresholds. In order to accomplish that, the results of the manual segmentation were compared to the results produced by the Geons-augmented CaS using different thresholds by comparing the number of segments and assigning a classification to the results.

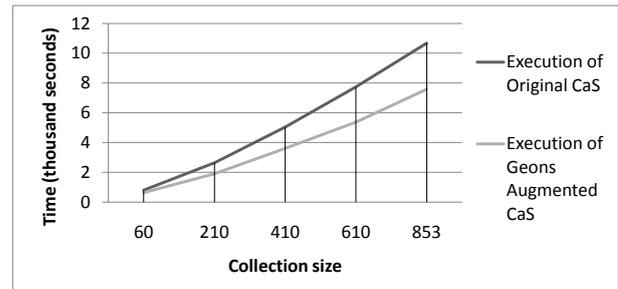


Figure 9: Execution times for both approaches regarding collection size

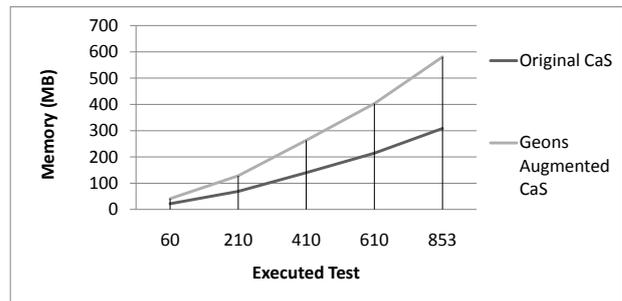


Figure 10: Memory requirements for both approaches with respect to collection size.

4.2 Execution time

The time of execution was obtained by executing the Geons-augmented CaS and the original CaS prototypes that were developed by implementing both approaches. It was used collections of objects with different amounts of objects. It is possible to conclude by observing Figure 9 that the time has a linear growing and comparatively, the geon-augmented CaS presents better results than the original CaS. It was also possible to observe that the signatures computation is time consuming and is where most of the time is spent.

4.3 Memory requirements

A similar study was performed to the memory requirements, as shown in Figure 10 it is used more memory on the geons augmented CaS than on the original CaS. This happens because it is used more lists in order to reduce the execution time. The memory is nowadays a cheap resource and has been increased over the years. So, having to spend more memory but as a result we get a approach faster and with better results to the users is a good trade off between these two important measures.

4.4 Evaluation with users

To prove the results quality, it was performed a test with users where they had to compare pairs of results.

The users were asked first to compare the results of the geons augmented with the ones produced by the original

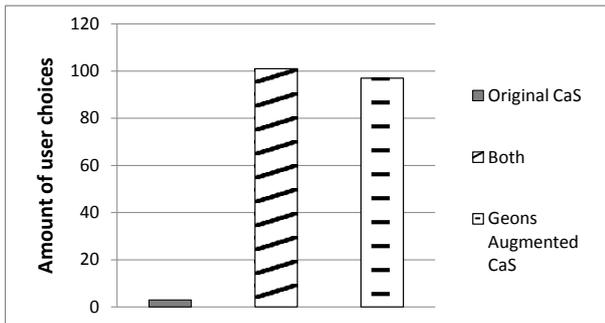


Figure 11: Users choice of best segmentation between original CaS and Geons-augmented CaS.



Figure 12: Segmentation of an object with the original CaS (left) and the geons-augmented CaS (right).

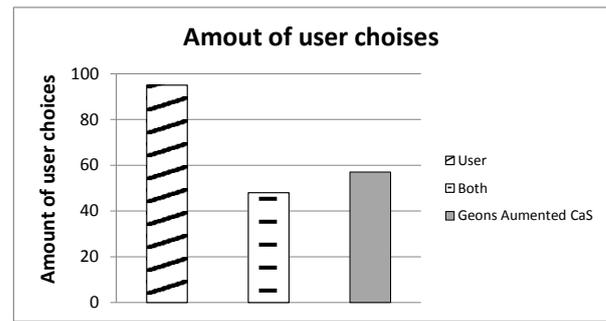


Figure 13: Users choice of best segmentation between men-made segmentation and Geons-augmented CaS.



Figure 14: Segmentation of an object made by humans (left) and through geons-augmented CaS (right).

CaS. On this test the results produced by Geons-augmented CaS approach proved to be more meaningful to the user as is shown on Figure 11 where most of user chosen both results or the geons augmented approaches but rarely only the original CaS. As shown in Figure 12 the segmentation result of an object produced by the original CaS approach has three more segments than the segmentation result using Geons-augmented CaS approach. These three more segments produce an over-segmentation according to user's perception.

From these results we conclude that the Geons-augmented CaS approach, according to the users, produces better results than the original CaS approach.

Additionally, we asked users to compare the results produced by the geons-augmented CaS with objects manually segmented by humans in a previous test. As shown in Figure 13, half of the users have chosen their results while the other half have chosen both results or the geons-augmented CaS. Indeed, as shown in Figure 14 the segmentation of an object produced by the the manual segmentation has very similar results when performed with geons-augmented CaS approach.

This analysis allow us to conclude that besides having some objects where the segmentation is not the best for the user, the majority of the collection objects results have been chosen by the user as the same or better than the one he chose. We can then conclude that manual segmentation prevails over the automatic segmentation algorithm, but often results are quite similar.

5 CONCLUSIONS and FUTURE WORK

We presented an extension of the CaS algorithm that segments a collection of 3D objects simultaneously in an automatic manner. The algorithm produces a meaningful segmentation regarding human perception, according to the results of the experimental evaluation. We also improved time complexity comparatively to the original CaS. Additionally, the proposed algorithm avoids over-segmentation by using geons as primitives.

Different tests were performed in order to test the efficacy and effectiveness of the approach and the quality of the produced results. From these tests we have concluded that the geons-augmented is faster than the original CaS but spends more memory. It was also clear that the presented solution produces results that are meaningful for humans. However, we consider that using HFP as a basis for the decomposition might be a limitation. Thus, it is necessary on future work to study the possibility of using other hierarchical based segmentation approaches.

It seems also promising to experiment other signature besides the rotation invariant spherical harmonics shape descriptor. Also, as one of the characteristics of these descriptors is that are not scale invariant, it is necessary to perform a deeper study on the geons to avoid having to use multiple geons on different scales.

In the future we believe that the proposed approach might be improved to develop into a robust, stable and scalable solution for the automatic decomposition of 3D object collections.

6. ACKNOWLEDGEMENTS

The work described in this paper was partially supported by the Portuguese Foundation for Science and Technology (FCT) through the project 3DORuS, reference PTDC/EIA-EIA/102930/2008 and by the INESC-ID multiannual funding, through the PIDDAC Program funds.

7. REFERENCES

- [Agathos07] A. Agathos, I. Pratikakis, S. Perantonis, N. Sapidis, and P. Azariadis. 3d mesh segmentation methodologies for cad applications. *Computer-Aided Design and Applications*, 4(6):827–841, 2007.
- [Biederman87] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–147, 1987.
- [Chen09] Xiaobai Chen, Aleksey Golovinskiy, and Thomas Funkhouser. A benchmark for 3D mesh segmentation. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28(3), August 2009.
- [Falcidieno06] M. Attene and B. Falcidieno and M. Spagnuolo. Hierarchical mesh segmentation based on fitting primitives. *Visual Computer*, 22(3):181–193, 2006.
- [Ferreira09] A. Ferreira, S. Marini, M. Fonseca M. Attene, M. Spagnuolo, J. Jorge, and B. Falcidieno. Thesaurus-based 3d object retrieval with part-in-whole matching. *International Journal of Computer Vision*, 2009.
- [Golovinskiy09] A. Golovinskiy and T. Funkhouser. Consistent segmentation of 3d models. *Computers & Graphics*, 33(3):262–269, 2009.
- [Katz03] S. Katz and A. Tal. Hierarchical mesh decomposition using fuzzy clustering and cuts. In *ACM SIGGRAPH 2003 papers*, pages 954–961, NY, USA, 2003. ACM.
- [Katz05] S. Katz, G. Leifman, and A. Tal. Mesh segmentation using feature point and core extraction. *The Visual Computer*, 21(8–10):649–658, 2005.
- [Kazhdan03] Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing, SGP '03*, pages 156–164, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.
- [Mortara03] M. Mortara, G. Patane, M. Spagnuolo, B. Falcidieno, and J. Rossignac. Blowing bubbles for multi-scale analysis and decomposition of triangle meshes. *Algorithmica*, 38(1):227–248, 2003.
- [Mortara06] M. Mortara, G. Patane, and M. Spagnuolo. From geometric to semantic human body models. *Computers & Graphics*, 30(2):185–196, 2006.
- [Shamir08] A. Shamir. A survey on mesh segmentation techniques. *Computer Graphics Forum*, 27(6):1539–1556, 2008.
- [Shapira08] L. Shapira, A. Shamir, and D. Cohen-Or. Consistent mesh partitioning and skeletonization using the shape diameter function. *The Visual Computer*, 24(4):249–259, 2008.
- [S.Jayanti06] S.Jayanti, Y. Kalyanaraman, N. Iyer, and K. Ramani. Developing an engineering shape benchmark for cad models. *Computer-Aided Design*, 38(9):939–953, 2006.
- [Tierny07] J. Tierny, J.-P. Vandeborre, and M. Daoudi. Topology driven 3d mesh hierarchical segmentation. In *International Conference on Shape Modeling and Applications*, pages 215–220, 2007.
- [Zuckerberger02] E. Zuckerberger. Polyhedral surface decomposition with applications. *Computers and Graphics*, 26(5):733–743, 2002.

Ray Tracing of Large Models on a Multi-Projection Display

Vasco Costa João M. Pereira Joaquim A. Jorge
INESC-ID/IST
Lisboa, Portugal
{vasc, jap, jaj}@vimmi.inesc-id.pt

Resumo

O uso de ecrãs de grande dimensão, no nosso caso uma 'display wall' multi-projector, tem vindo a aumentar. Quando vários utilizadores precisam de visualizar uma cena ou quando um único utilizador necessita de obter uma visão alargada do problema e requer uma resolução mais elevada que a disponível utilizando os outros tipos de ecrã à sua disposição. É natural que as cenas visualizadas neste tipo de ecrã de alta definição tenham também uma resolução elevada. Para efeitos de avaliação do desempenho do sistema de visualização utilizamos modelos 3D com uma complexidade na ordem de dezenas de milhões de triângulos. O sistema funciona através da subdivisão do ecrã de grande dimensão em M grupos de amostras em que M é o número de máquinas no aglomerado de computadores. Cada amostra é sintetizada traçando raios com recurso a um algoritmo de subdivisão espacial por nós concebido. O sistema permite a visualização a ritmos interactivos de cenas deste tipo.

Abstract

The use of large scale displays, in our case a multi-projector display wall, has been increasing in many applications where multiple users need to visualize a scene or when a single user needs to have a comprehensive view of things and requires a larger resolution than otherwise available. Hence it is natural that the scenes to be viewed should have high resolution as well. For benchmarking purposes we used 3D models with a complexity in the order of tens of millions of triangles. The system works by subdividing a large screen into M groups of samples where M is the number of machines in the rendering cluster. Each sample is ray traced using a spatial subdivision algorithm of our own design. The system is able to render such scenes at interactive rates.

Keywords

Raytracing, Parallel rendering, Distributed applications

1. INTRODUCTION

The use of large displays is increasingly more common for several kinds of applications due to the increased amount of information possible to present on such displays, they also make it possible for more than one person to use a display for visualization or other purposes. i.e. such displays are important for applications which are collaborative in nature or require work on minute detail in several areas including architectural, automotive, biomedical, heritage and others.

In our specific case we were interested in viewing large architectural or heritage pieces. For benchmarking purposes we selected a large scanned model from a standard dataset, namely the Thai Statue model, with 10 million triangles, from the Stanford archive.

To solve the specific issue of how to display the data we used the hardware resources available, namely a cluster of a dozen PCs, to render the model in a parallel fashion.

There are several standard solutions to visualize models on a cluster. These often use OpenGL [Shreiner09] based

libraries such as Chromium [Humphreys08], OpenSceneGraph [Wang10] or OpenSG [Reiners02]. These libraries provide a large amount of flexibility for someone who wishes to write an application. However they have difficulty handling large datasets. We were also interested in evaluating the performance of an alternative rendering algorithm, in this case ray tracing, for this scenario.

Ray tracing backed by an acceleration structure has a performance less sensitive to triangle count and large amounts of overdraw which will be common in future architectural scenes we wish to visualize. The currently benchmarked model has limited depth complexity.

We also decided early on the planning phase that the parallelization algorithm should be as independent of the particular ray tracing acceleration structure or method as possible. Hence we decided, contrary to other parallel ray tracing implementations, to focus on the task of assigning the task of rendering each sample. Since the display has a large resolution of 4096×2304 pixels there should be no issue in finding enough work to assign to computing nodes of

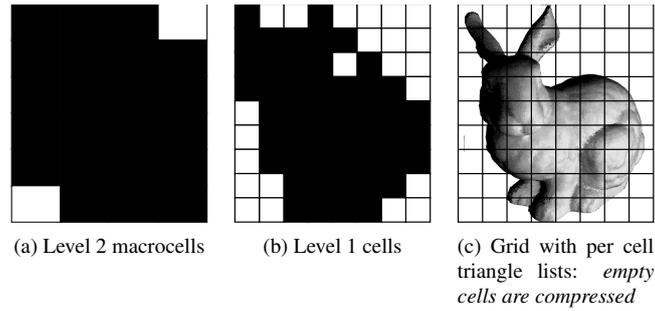


Figure 1: Multi-level static grid with macrocells for fast empty region traversal

the cluster.

Chapter 2 reviews some related work in the area focusing on grid acceleration structures for parallel ray tracing. In chapter 3 we present our multi-level static acceleration structure algorithm. Chapter 4 describes the system architecture: developed software and the hardware it runs on. We discuss the system test results in chapter 5. Finally we present our conclusions and pinpoint directions for future work.

2. PREVIOUS WORK

Ray tracing over a cluster of PCs [Wald02, Benthin06] has been an active area of research since the past decade. Ize et al worked on sort-middle cluster ray tracing using grids [Ize06] as well as how to optimize such a task using bounding volume hierarchies (BVH) on a cluster [Ize11] with an Infiniband interconnect. Their work divides acceleration structure (grid, BVH) cells among the cluster nodes so it is possible to view large models which would otherwise not fit in available memory. The issue with parallelizing an acceleration structure in such a fashion is the large performance penalty, up to $10\times$ slower, in their case compared to the case where it is simply replicated in full.

Grid acceleration structures [Fujimoto86] subdivide the space of the scene into cubically shaped cells. The 3D DDA cell traversal method for ray tracing introduced by Fujimoto et al. was later improved [Amanatides87]. This more recent traversal method is still commonly used today. Macrocells [Wald06] can be used to speed up empty space traversal during ray tracing.

Further improvements include efficient compression of the grid acceleration structure via the row displacement algorithm [Lagae08]. It reduces grid memory footprint by a factor of $20 : 1$ by compressing the empty grid cells which would otherwise use large amounts of memory (see Figure 2).

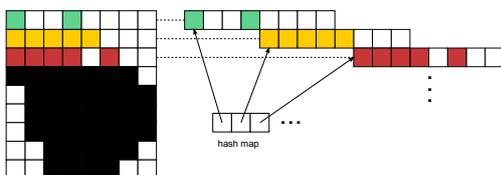


Figure 2: Row displacement compression.

3. MULTI-LEVEL STATIC GRIDS

In order to render complex scenes, with tens of millions of triangles, on current hardware it is necessary to use a spatial subdivision acceleration structure of some sort. In this way geometry which is either obscured or otherwise out of view is not processed resulting in interactive frame rates.

For this work we decided to use a multi-level static grid ray tracing acceleration structure of our own design. Since present hardware architectures have multiple levels of cache with varying bandwidth and latency it is advisable to reduce the working set to a minimum amount of memory. To further enhance rendering performance we speeded up cell traversal with a multi-level hierarchy of macrocells. The macrocells enable faster ray traversal ($2\times$ faster in scanned scenes) by skipping large empty regions of space (see Figure 1). In short the macrocell structure is a small 3D bit array which allows the trivial rejection of empty regions of space with minimum cache memory usage.

Improvements to the basic algorithm (macrocells, row displacement compression) enable the use of a finer grid resolution than would otherwise be possible with conventional algorithms which have severe memory bandwidth and cache thrashing issues [Costa10].

To pick the resolution of the rectangular axis aligned cells for a given scene in a grid we employ an heuristic. The heuristic usually involves the number of triangles in a scene, the scene's bounding box extents or volume, as well as a user defined density parameter. This parameter aims to make the number of cells M linear in regards to the number of triangles N in a scene, where:

$$M = \rho N \quad (1)$$

ρ is the grid density. The number of cells M is equal to the grid resolution $Mx \times My \times Mz$. Cubically shaped cells work best, and most heuristics take this into account. In this work a ρ value of 32 is used.

We use the following heuristic to pick the grid resolution:

$$M_i = S_i \sqrt[3]{\frac{\rho N}{V}} \quad (i \in \{x, y, z\}) \quad (2)$$

S_i is the scene bounding box size in dimension i , V is the bounding box volume.

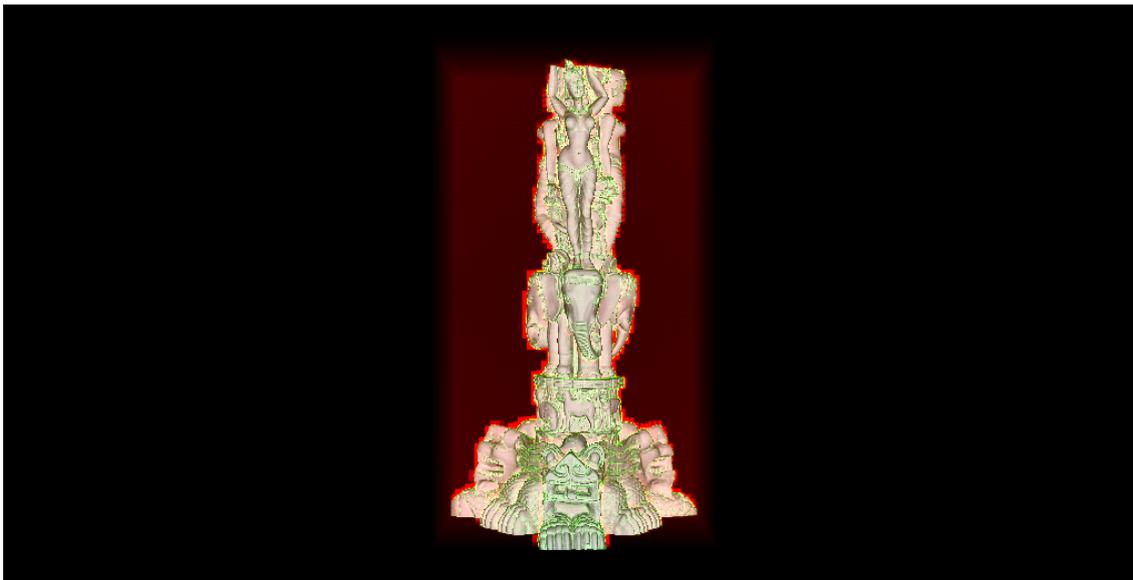


Figure 3: Visualization of the Thai Statue model (10 million triangles) which shows the ray tracing complexity for the given scene. Ray tracing was done using a multi-level static grid acceleration structure. Cell traversal costs are displayed in red, while triangle intersection costs are displayed in green.

The rendering time when using an acceleration structure of this type is approximately:

$$t_{rendering} = t_{cell\ traversals} + t_{triangle\ intersections} \quad (3)$$

Figure 3 shows an example of the ray tracing complexity of the scene.

4. SYSTEM ARCHITECTURE

We developed parallel ray tracing rendering system software which is run on a cluster of PCs.

We replicate the entire dataset on all cluster nodes. We are working on a Gigabit Ethernet cluster with much worse bandwidth and latency characteristics than the Infiniband used in [Ize11] so we chose not to use their kind of solution. This is done prior to visualization. Since the dataset is large but each node we have contains a limited amount of memory (1 GB DRAM) we needed to employ several forms of compression to enable the model to fit into the available memory space. We hope to upgrade the cluster powering the display soon which will enable us to further improve performance in the near future.

Below we describe the heuristics used and the software and hardware architectures of the implemented system in further detail.

4.1. WORKLOAD DIVISION ALGORITHMS

We divide the screen samples among the cluster nodes to split the workload. The cluster node assigned to work on that sample ray traces the full ray tree for the ray intersecting that point in the view area. We use a couple of heuristics to decide how to split the workload, namely the interleaved and tiled heuristics.

All of these heuristics are static in nature. They are easy to compute quickly but do not take the displayed scene char-

acteristics into account. If we had feedback on the rendering cost of each work unit of interest (sample, tile) we could attempt to do dynamic load balancing during the rendering process. This could be done with a parallel scheduler of some sort.

4.1.1. INTERLEAVED HEURISTIC

In the interleaved heuristic we farm out samples in the following fashion: sample i is assigned to machine:

$$i \bmod M \quad (4)$$

where M is the number of machines in the cluster.

For a cluster with a dozen machines: sample 0 is worked by machine 0, sample 1 is worked by machine 1, ..., sample 12 is worked by machine 0, sample 13 is worked by machine 1, ...

4.1.2. TILED HEURISTIC

In the tiled heuristic we subdivide the screen into tiles of 8×8 samples each. This should allow for increased memory coherency during ray tracing compared to the previous interleaved heuristic which has worse memory coherency in particular for clusters with large numbers of machines where samples are further apart.

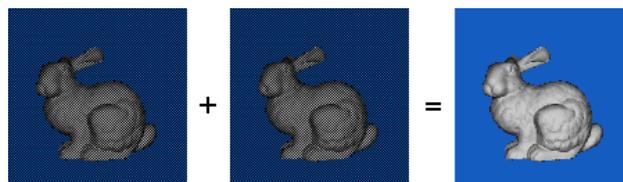


Figure 4: Interleaved heuristic sample distribution for a cluster with two nodes.

In the tiled heuristic we farm out tiles in the same fashion as we farm out samples in the interleaved heuristic: tile t is assigned to machine:

$$t \bmod M \quad (5)$$

where M is the number of machines in the cluster.

For example, for a cluster with a dozen machines: tile 0 is worked by machine 0, tile 1 is worked by machine 1, ..., tile 12 is worked by machine 0, tile 13 is worked by machine 1, ...

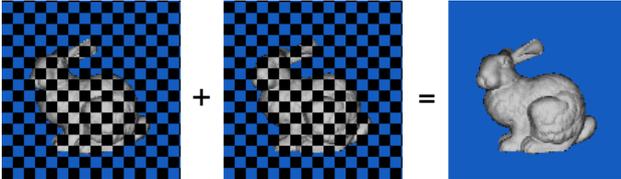


Figure 5: Tiled heuristic sample distribution for a cluster with two nodes.

4.2. SOFTWARE ARCHITECTURE

All task assignment work is performed by the Compositor Server (CS). The CS runs on a separate machine. Rendering machines connect to the CS as they are brought online. The CS is resilient to rendering machine breakdowns and allows dynamic joining and leaving of rendering processes.

Each rendering machine contains a process which does the rendering of the samples it is assigned to. Upon the beginning of each frame the CS sends a packet to each rendering machine with the instructions on which samples it needs to process. The results are then sent back to the CS as soon as work on the frame is finished. This ensures there is no visible tearing or other rendering glitches.

Once a frame is finished rendering the CS outputs the results to the screen, which may be the display wall, or any other OpenGL rendering surface.

The software was programmed in ANSI C++ without use of machine specific assembly instructions or intrinsics.

The scene memory usage can be computed as requiring 12 bytes per triangle for storing vertex index information (three machine words for each vertex index), plus 12 bytes for each vertex (three floating point numbers per coordinate). In this way we can reduce the storage requirements for those mesh triangles which have shared vertices. Ray/triangle intersections are computed using the Möller-Trumbore [Möller05] intersection algorithm because it does not require any additional memory.

4.3. COMPOSITOR PACKETS

The compositor server interacts with the rendering servers using two kinds of packets as shown in Figures 6, 7.

The output samples are sent as uncompressed RGB data using one byte per color component.

machine-id	: int
machines	: int
eye	: float[3]
ll	: float[3]
u	: float[3]
v	: float[3]
xres	: int
yres	: int

Figure 6: Packets written to each rendering process.

packet-size	: int
samples-rgb	: byte[packet-size]

Figure 7: Packets read from each rendering process.

4.4. HARDWARE ARCHITECTURE

The rendering cluster contains twelve PCs with a Pentium 4 CPU at 3.0 GHz each with 1 GB of RAM. The compositor server runs on a similarly specced machine. The cluster machines run the Linux operating system.

The cluster is connected via Gigabit Ethernet. The available network bandwidth from the rendering machines to the compositor server was benchmarked using *iperf* at 680 Mbits/sec.

The display wall is a canvas illuminated by twelve projectors with a 4×3 geometry. Each projector runs at 1024×768 resolution. The canvas surface has dimensions of $4.00m \times 2.25m$.

5. RESULTS AND DISCUSSION

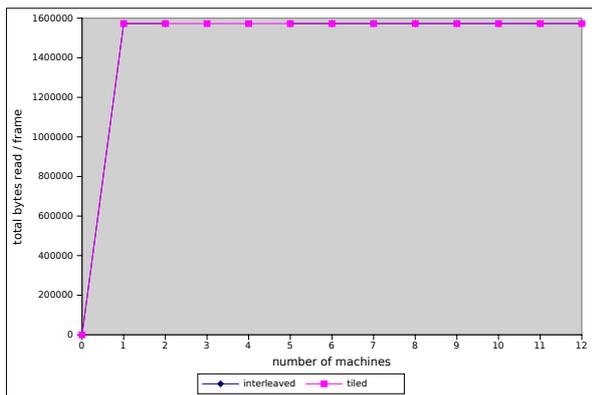
For each test statistics were gathered over an interval of five seconds of rendering in order to collect enough samples to reduce measuring error.

The test scene consisted of the ten million triangle Thai Statue from the Stanford 3D Scanning Repository. This scene is large enough to be representative of the kinds of scenes we wish to be able to visualize in the future.

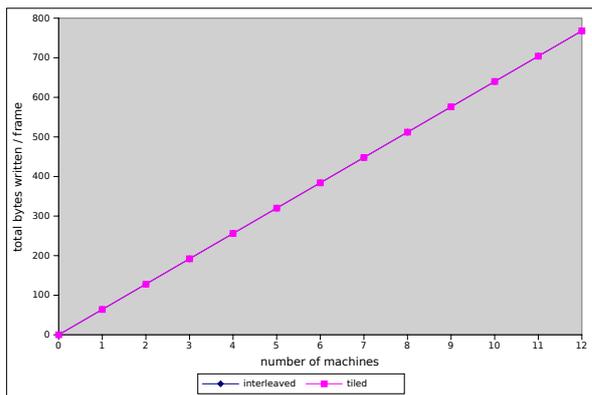
The tests were run using zero to twelve rendering machines in action to measure the speedup achieved from adding more machines to the system. Running the tests with zero ray tracing machines connected to the compositor server also allowed us to measure the speed at which it is able to send frames to the screen without any time being spent in ray tracing whatsoever.

Each frame has a resolution of 1024×512 with one sample per pixel. A frame is then scaled up to 4096×2304 screen resolution using the bilinear filtering facilities provided by OpenGL. This frame resolution was selected because our version of OpenGL does not support non power of two textures and this provided a similar aspect ratio to the actual screen resolution. This resolution also allows us to achieve interactive ray tracing frame rates for the scene in question.

As expected the bandwidth incoming to the compositor server remains constant since incoming data is composed of sampled pixels resulting from the ray tracing algorithm

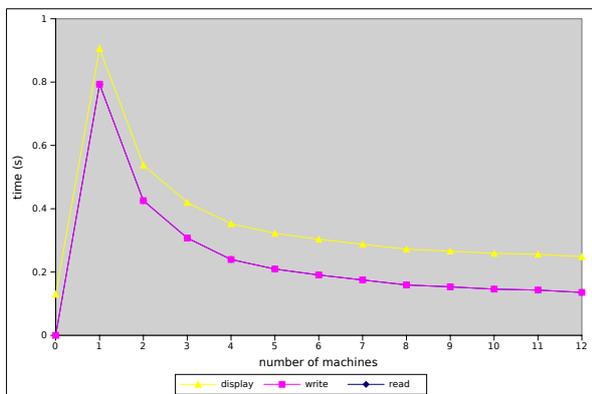


(a) Bandwidth incoming to compositor server

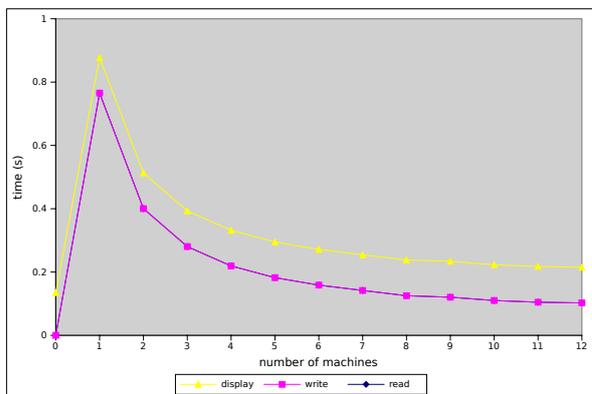


(b) Bandwidth outgoing from compositor server

Figure 8: The above charts show the bandwidth used by the compositor server while rendering a single frame.



(a) Interleaved rendering algorithm



(b) Tiled rendering algorithm

Figure 9: The above charts show the time required to render a single frame using each of the two heuristics.

which are $xres \times yres \times 3$ in size no matter what the number of machines in the cluster is. Outgoing bandwidth scales linearly with the number of machines in the cluster since the compositor server sends one packet to each machine telling it which rays it must render.

Running the tests showed we have a bottleneck sending each frame from the compositor server to the large screen display. This can clearly be seen in Figure 9. Just to display a blank screen takes a whole 114ms which results in a frame rate of 8.77 FPS.

This means the rendering time for one frame will always be inferior to this value. This requires more in depth investigation of our setup. We are using Chromium for doing the 2D frame scaling and screen blitting operations and this may be the bottleneck given the large size of the texture and/or the way we are displaying it. For the tiled rendering ray tracing algorithm it gets so ridiculous it takes more time to display the frame to the screen than to ray trace the whole scene.

The speedup is clearly sublinear. This may be due to a bottleneck in the compositor server which is not multi-threaded: the machine running it does not support the OpenMP parallelization primitives that we have in our implementation. Yet there is a clear performance improve-

ment even at the larger cluster sizes.

The goal of achieving speedup versus a single machine is clearly achieved since the ray tracing rendering operation is around 4x faster with the whole cluster rather than a single machine. However the performance speedup peters out as the number of machines in the cluster grows. Linear ray tracing speedup is observed with two machines but speedup starts decreasing as more machines are added to the configuration.

6. CONCLUSIONS AND FUTURE WORK

Clearly the cause of the bottleneck in displaying a frame to the multi-projector screen must be figured out and solved before any further improvements in ray tracing performance are to be attempted. It presently constitutes the majority of the time spent during rendering in our implementation.

Another thing which needs to be attempted is multi-threading the compositor server. The compositor server must handle multiple ray tracing machines at the same time and the current implementation may be the cause of the sublinear ray tracing performance speedup. The tiled rendering heuristic has proved superior but globally the results are not as impressive as expected in the current implementation.



Figure 10: Rendering of the Thai Statue (10 Mtri) on the $4.00m \times 2.25m$ display wall. The hardware powering this application consists of a ray tracing cluster of twelve machines with the screen compositor server running on a separate machine.

7. ACKNOWLEDGEMENTS

We would like to thank the Stanford 3D Scanning Repository for the Thai Statue model.

This work was supported by national funds through FCT - Fundação para a Ciência e a Tecnologia, under project PEst-OE/EEI/LA0021/2011.

8. REFERENCES

- [Amanatides87] J. Amanatides and A. Woo. A fast voxel traversal algorithm for ray tracing. In *Proceedings of EUROGRAPHICS*, volume 87, pages 3–10, 1987.
- [Benthin06] C. Benthin. *Realtime Ray Tracing on Current CPU Architectures*. PhD thesis, Saarland University, 2006.
- [Costa10] V. Costa, J. Pereira, and J. Jorge. Multi-level hashed grid construction methods. In *WSCG*, 2010.
- [Fujimoto86] A. Fujimoto, T. Tanaka, and K. Iwata. Arts: Accelerated ray-tracing system. *Computer Graphics and Applications, IEEE*, 6(4):16–26, 1986.
- [Humphreys08] G. Humphreys, M. Houston, R. Ng, R. Frank, S. Ahern, P.D. Kirchner, and J.T. Klosowski. Chromium: a stream-processing framework for interactive rendering on clusters. In *ACM SIGGRAPH ASIA 2008 courses*, page 43. ACM, 2008.
- [Ize06] T. Ize, I. Wald, C. Robertson, and S.G. Parker. An evaluation of parallel grid construction for ray tracing dynamic scenes. In *Proceedings of the IEEE Symposium on Interactive Ray Tracing*, pages 47–55, 2006.
- [Ize11] T. Ize, C. Brownlee, and C.D. Hansen. Real-time ray tracer for visualizing massive models on a cluster. In *Proceedings of the 2011 Eurographics Symposium on Parallel Graphics and Visualization*, 2011.
- [Lagae08] Ares Lagae and Philip Dutré. Compact, fast and robust grids for ray tracing. *Computer Graphics Forum (Proceedings of the 19th Eurographics Symposium on Rendering)*, 27(8), 2008.
- [Möller05] T. Möller and B. Trumbore. Fast, minimum storage ray/triangle intersection. In *International Conference on Computer Graphics and Interactive Techniques*. ACM Press New York, NY, USA, 2005.
- [Reiners02] D. Reiners. *OpenSG: A scene graph system for flexible and efficient realtime rendering for virtual and augmented reality applications*. PhD thesis, Darmstadt University, 2002.
- [Shreiner09] D. Shreiner. *OpenGL programming guide: the official guide to learning OpenGL, versions 3.0 and 3.1*. Addison-Wesley Professional, 2009.
- [Wald02] I. Wald, C. Benthin, and P. Slusallek. A flexible and scalable rendering engine for interactive 3d graphics. *Computer Graphics Group, Saarland University*, 2002.
- [Wald06] I. Wald, T. Ize, A. Kensler, A. Knoll, and S.G. Parker. Ray tracing animated scenes using coherent grid traversal. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 485–493. ACM, 2006.
- [Wang10] R. Wang and X. Qian. *OpenSceneGraph 3.0*. Packt Publishing, 2010.

Adaptação de Simbologia em Realidade Aumentada

Maria Beatriz Carmo, Ana Paula Cláudio, António Ferreira, Ana Paula Afonso,
Edgar Montez, Raúl Simplício

Departamento de Informática, Faculdade de Ciências, Universidade de Lisboa

Campo Grande 1749-016 Lisboa

{bc,apc,asfe,apa}@di.fc.ul.pt, fc35956@alunos.fc.ul.pt, rjsimplicio@fc.ul.pt

Resumo

Em aplicações de Realidade Aumentada, quando a cor dos elementos gráficos, que são desenhados sobre uma imagem real, é semelhante à cor do fundo circundante, torna-se difícil identificar esses elementos. O ajuste dinâmico do aspecto dos símbolos (objectos virtuais), em função da cor do fundo, deverá permitir melhorar a legibilidade da informação a transmitir ao utilizador. O objectivo deste trabalho é identificar um conjunto de adaptações a aplicar aos símbolos que permitam realçá-los mas sem alterar a sua semântica. Neste artigo apresentamos o resultado de dois estudos que efectuámos. No primeiro, averiguámos as preferências dos utilizadores relativamente às seguintes adaptações dos símbolos: adição de um rebordo, ajuste da luminosidade da cor, aumento do tamanho, e alteração da cor dos caracteres no interior do símbolo. Os utilizadores preferiram maioritariamente a adição de rebordo e em segundo lugar o ajuste da luminosidade da cor. Tendo em atenção estes resultados, realizámos um segundo estudo para averiguar qual a variação mínima da luminosidade para que um símbolo se destaque da cor do fundo, cujo resultado indica cerca de 0.25 unidades do modelo HSV.

Palavras Chave

Realidade Aumentada, Adaptação de Simbologia, Estudos com Utilizadores.

1. INTRODUÇÃO

As aplicações de Realidade Aumentada (RA) combinam símbolos gráficos com imagens do mundo real, para fornecer informação adicional ao utilizador. Como não há controlo sobre o conteúdo dessas imagens, os símbolos virtuais podem ficar imperceptíveis para o utilizador porque, por exemplo, a sua cor é semelhante à cor do fundo circundante da imagem. Este problema tem ainda maior relevo quando a aplicação é usada no exterior, onde, além de haver maior variabilidade de cenários, a iluminação é normalmente mais intensa e pode mudar rapidamente [Kalkofen09a].

Um exemplo deste problema é a visualização de pontos de interesse em dispositivos móveis, recorrendo a aplicações como o Layar [Layar] ou o Wikitude [Wikitude], em que, sobre a imagem de fundo captada pela câmara, são colocados símbolos gráficos alinhados com a localização dos objectos reais associados. Em situações em que os símbolos se confundem com o fundo, seria útil alterar a sua representação dinamicamente de modo a destacá-los. Contudo, uma alteração súbita que mude a cor dos símbolos poderá confundir e desorientar o utilizador. A visualização de dados recorrendo ao mapeamento de intervalos de variação em tabelas de cor (*color mapping*) é outro exemplo em que alterações súbitas de cor podem alterar a compreensão dos resultados [White09].

Interessa, portanto, identificar adaptações de simbologia que possam ser executadas automaticamente para realçar os símbolos, preservando, contudo, a semântica original.

Tendo por base características que guiam a atenção visual, seleccionámos um conjunto de adaptações dos símbolos que considerámos satisfazerem as condições pretendidas: adição de um rebordo, ajuste da cor em termos de luminosidade, aumento do tamanho, e alteração da cor dos caracteres presentes no interior do símbolo. Realizámos um estudo com utilizadores para averiguar se estas adaptações promoviam efectivamente a legibilidade dos símbolos.

Atendendo a que uma das conclusões deste estudo apontou para a utilização de ajustes na luminosidade da cor do símbolo, efectuámos um segundo estudo com utilizadores para determinar qual a variação mínima de luminosidade para garantir que um símbolo se destaque do fundo.

Apresentamos de seguida, na Secção 2, o trabalho relacionado, incluindo alguns estudos que abordam características que permitem realçar elementos gráficos. Nas Secções 3 e 4 descrevemos os dois estudos que efectuámos e, finalmente, na Secção 5, apresentamos as conclusões e o trabalho futuro.

2. TRABALHO RELACIONADO

Nas aplicações de RA a combinação de elementos virtuais sobre imagens do mundo real levanta alguns problemas. Kalkofen *et al.* propõem várias técnicas para melhorar esta combinação, nomeadamente, no que diz respeito a realçar representações com reduzido contraste com as regiões vizinhas, estes autores propõem o uso de cor artificial, ou seja, adicionar elementos gráficos co-

identes com elementos do objecto real, mas com uma cor diferente da destes [Kalkofen09b].

No trabalho de Gruber *et al.*, para harmonizar as cores na imagem, quer as cores dos objectos virtuais quer as do mundo real são ajustadas tendo por base directrizes estéticas para a combinação de cores [Gruber10]. Uma vez que a cor de alguns objectos do mundo real está relacionada com o seu significado, enquanto para outros tal não acontece, os objectos são classificados em diferentes categorias de acordo com as restrições que se colocam à mudança da sua cor. Desta forma, garante-se que a cor é mantida sempre que o seu significado seja importante. No nosso trabalho pretendemos ajustar os símbolos de forma a distingui-los do fundo sem entrar em consideração com a harmonização estética.

Para além de símbolos gráficos, também pode ser usado texto para juntar informação adicional em aplicações de RA. Em [Gabbard07] descreve-se um estudo com utilizadores para investigar as interações entre a imagem captada, a intensidade da luz exterior e a legibilidade do texto colocado sobre a imagem. Neste estudo foram usados algoritmos para melhorar, em tempo-real, o contraste entre o texto e o fundo.

Em [Leykin04] estuda-se também a legibilidade de texto em função das texturas presentes no ambiente real. Os autores fizeram experiências com utilizadores para criar uma aproximação baseada em reconhecimento de padrões para identificar as regiões de pouca legibilidade. Usaram imagens numa escala de cinzentos em que calcularam a intensidade do contraste entre o texto e a região envolvente. O objectivo era evitar aplicar texto em regiões com pouca legibilidade, alterando a sua posição sobre a imagem sempre que necessário.

No nosso trabalho não focamos a legibilidade de texto. O nosso objectivo é adaptar símbolos de modo a que fiquem perceptíveis, sem mudar a sua posição e sem modificar a imagem do mundo real. É por isso necessário identificar características que permitam dar realce aos símbolos.

Wolfe e Horowitz estudaram os atributos que são mais adequados para orientar a pesquisa visual [Wolfe04]. Baseados em estudos realizados por diferentes autores, concluíram que a cor, o movimento, a orientação e o tamanho são atributos que guiam a atenção do utilizador. Os mesmos atributos foram também estudados por Paley para distinguir o texto apresentado numa janela, com um fundo transparente, sobreposta a outra [Paley03]. Paley verificou que, neste caso, as seguintes adaptações melhoraram a legibilidade do texto: variar a cor dos caracteres, usar texto maior, mudar o estilo da fonte, por exemplo, usando negrito; colocar contornos em redor de cada carácter, e introduzir um movimento subtil com uma variação suave do deslocamento da janela de sobreposição.

O uso de um contorno também foi seguido por Nivala *et al.* em símbolos que representam pontos de interesse em mapas de dispositivos móveis [Nivala07]. Outro aspecto que consideraram foi a adaptação dos símbolos de acordo

com as características dos utilizadores; por exemplo, os símbolos para as pessoas mais velhas, além de serem um pouco maiores, tinham um fundo branco para os realçar.

Tendo em conta estes estudos, definimos um conjunto de adaptações a aplicar aos símbolos gráficos e realizámos um estudo com utilizadores para as avaliar. As adaptações são descritas na secção seguinte.

3. ESTUDO SOBRE ADAPTAÇÕES DE SÍMBOLOS

Este estudo consistiu em seleccionar um conjunto de adaptações e depois testar com utilizadores se essas adaptações realçavam os símbolos mantendo a sua semântica.

3.1 Adaptações consideradas

Considerámos como símbolo base (BA), a adaptar, um quadrado preenchido com uma cor uniforme, contendo um ou dois caracteres pretos (letras ou algarismos). Esta escolha é justificada por ser um tipo de símbolo frequente em aplicações de visualização de pontos de interesse e, além disso, por se tratar de um símbolo que *per si* não se destaca facilmente, permitindo criar situações em que é difícil distingui-lo do fundo.

Levando em linha de conta trabalhos realizados por outros autores sobre as características que podem guiar a atenção visual e realçar a representação, testámos as seguintes adaptações do símbolo base: adição de um rebordo (RE), que pode ser branco (RB) ou preto (RP); ajuste da luminosidade da cor (CO), com variantes C1 e C2, explicadas a seguir; ampliação do símbolo (AM); e alteração da cor dos caracteres dentro do símbolo (LE), que neste estudo se cingiu a letras brancas (LB). A Figura 1 mostra o símbolo base bem como exemplos de cada adaptação. Decidimos realizar as adaptações em separado e não realizar combinações de adaptações (por exemplo, adaptação da cor e rebordo) de modo a focar o estudo e reduzir eventuais interferências entre adaptações.

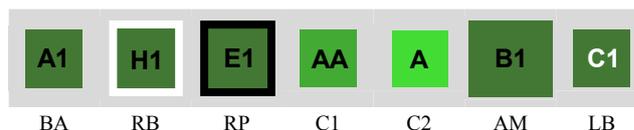


Figura 1: Exemplos de símbolo base e adaptações

Para decidir se um símbolo deve ser adaptado, comparámos a cor dominante da imagem na área que envolve o símbolo, isto é, a cor que ocorre com a maior frequência nessa área, com a cor do símbolo. Se o módulo da diferença entre cada uma das componentes RGB destas duas cores for inferior a um limiar pré-fixado, as duas cores são consideradas semelhantes, e quando isto acontece, o símbolo é adaptado.

3.1.1 Adição de rebordo

Na adaptação com adição de um rebordo estudámos o efeito de juntar um rebordo branco ou um rebordo preto tanto em situações em que o fundo e o símbolo são claros como quando o fundo e o símbolo são escuros. Apesar de considerarmos que neste tipo de adaptação seria mais adequado usar rebordo branco em símbolos escuros e rebordo preto em símbolos claros, testámos a adição dos

dois tipos de rebordo no mesmo símbolo para confirmar esta assunção.

3.1.2 Ajuste da luminosidade da cor

A alteração da cor de um símbolo pode ser interpretada como uma mudança no seu significado [Silva11]. Considerámos por isso apenas ajustes de luminosidade, para realçar os símbolos do fundo, mas sem comprometer o seu significado. Utilizámos a seguinte abordagem: quando o fundo é escuro, o símbolo é desenhado com uma luminosidade mais clara que a do símbolo base, e quando o fundo é claro, o símbolo é tornado mais escuro.

Utilizámos o modelo de cor RGB, com todas as componentes a variar no intervalo [0, 255]. Aplicámos um algoritmo simples para ajustar a luminosidade da cor com base nos valores da(s) componente(s) RGB dominantes na cor do símbolo, adicionando/subtraindo a essas componentes um valor constante pré-definido. Para assegurar que o ajuste seria perceptível, testámos duas constantes diferentes: 50 e 100. Estas variantes são identificadas na Figura 1 por C1 e C2, respectivamente.

3.1.3 Aumento do tamanho

Como mencionado anteriormente, Wolfe e Horowitz consideraram a variação de tamanho como uma característica que chama a atenção [Wolfe04]. Esta conclusão é também confirmada pelo nosso senso comum. Portanto, uma das adaptações que considerámos foi a ampliação do símbolo base. Após vários ensaios escolhemos o factor de escala de 50% para ampliar os símbolos.

3.1.4 Alteração da cor dos caracteres

Para aumentar o contraste dos caracteres presentes no interior do símbolo, quando o fundo e o símbolo têm pouca luminosidade, os caracteres no símbolo são representados em branco. Como usámos um símbolo base com letras pretas, não considerámos a adaptação que mudaria letras brancas para letras pretas sobre os símbolos com muita luminosidade.

3.2 Aplicação desenvolvida

De modo a afinar os parâmetros das adaptações, como a largura do rebordo, a variação da luminosidade da cor, a cor das letras, e o factor de escala para ampliar os símbolos, desenvolvemos uma aplicação de RA em Java usando o NyARToolkit [NyARToolkit]. A aplicação implementa os algoritmos que efectuam a adaptação dinâmica de um símbolo gráfico quando a sua cor dominante é semelhante à cor dominante do fundo em torno da marca.

3.3 Estudo com utilizadores

Realizámos um estudo com utilizadores para avaliar as adaptações descritas anteriormente, em cenários nos quais os símbolos base estão em condições adversas, isto é, têm cor semelhante à área circundante da imagem de fundo. Nestas circunstâncias assumimos que as adaptações iriam ajudar a realçar o símbolo, tomando como hipóteses que a adição de um rebordo, o ajuste da luminosidade da cor, a ampliação do símbolo, e a alteração de cor dos caracteres serão preferidos pelos utilizadores em comparação com o símbolo base.

3.3.1 Participantes

Um total de 55 pessoas, 22 homens e 33 mulheres, recrutados a partir de contactos sociais participaram voluntariamente e gratuitamente neste estudo. A idade média foi de 26 anos, com 24 participantes entre 20 e 24 anos, 10 entre 25 e 39 anos, e os restantes 21 entre 40 e 79 anos. 14 participantes eram alunos de doutoramento ou mestrado, 22 eram estudantes de licenciatura, e 19 não concluíram o ensino médio. Sobre a sua experiência em pesquisa de pontos de interesse em mapas, a maioria já o tinha feito, 19 frequentemente, 23 raramente, ao passo que 13 nunca o tinha feito.

3.3.2 Aparato

Realizámos este estudo dentro de portas para minimizar influências causadas por variações da luz do dia. Utilizámos um computador portátil com ecrã de 13.3 polegadas e uma resolução de 1280×800 *pixels*, colocado sobre a mesa em frente ao participante. Para cada pessoa a posição do ecrã foi ajustada para evitar reflexos da luz e permitir uma visão clara do seu conteúdo.

Usámos um pacote de *software* para fazer apresentações para mostrar imagens durante alguns segundos de uma maneira controlada.

3.3.3 Tarefas

Cada participante realizou tarefas de contagem e de indicação de preferência. A primeira serviu para forçar o participante a varrer toda a imagem para procurar os símbolos, para posteriormente indicar qual o preferido. A tarefa de contagem tinha um tempo limite de 8 segundos. Após o tempo estipulado, a imagem era retirada do ecrã e o participante dizia quantos símbolos tinha contado. A imagem era novamente mostrada para a tarefa de indicação de preferência, que não tinha limite de tempo. O participante indicava qual o símbolo que distinguia melhor.

3.3.4 Design

Desenhámos um estudo de medidas repetidas, ou seja, em cada teste o mesmo participante foi exposto a diferentes condições. Manipulámos duas variáveis independentes: o tipo de adaptação do símbolo e o tipo de fundo. Testámos todos os tipos de adaptação apresentados na Secção 3.1 e o símbolo base (sem adaptação). Em relação à variável de tipo de fundo, distinguimos os fundos predominantemente escuros dos claros, e, em qualquer dos casos, os símbolos sobrepostos tinham cores idênticas às do fundo. Mais precisamente, usámos símbolos cinzentos ou bege sobre fundos claros e símbolos verdes-escuros ou castanhos-escuros em fundos escuros.

Para garantir que todos os participantes estiveram expostos às mesmas imagens e em condições idênticas, efectuámos os testes com imagens estáticas (fotografias de ambientes urbanos) e dentro de portas. Desta forma, o estudo pôde ser realizado independentemente de influências externas, como a luminosidade intensa, mantendo ao mesmo tempo um grau razoável de realismo. Além disso, pudemos reproduzir mais facilmente situações adversas em que a cor de um símbolo era propositadamente semelhante à cor da área circundante na imagem de fundo.

Usámos várias fotografias para manter o interesse dos participantes e reduzir efeitos de aprendizagem, tendo a maior parte delas sido mostrada duas vezes: primeiro com o símbolo base e depois com símbolos base e adaptados.

Organizámos as fotografias em 5 blocos de testes: A, B, C, D e E. A Tabela 1 mostra o tipo de símbolo e adaptação associados a cada bloco, o tipo de fundo (claro ou escuro), e o identificador do teste. Apresentamos aqui 16 dos 21 testes efectuados, tendo os restantes sido omitidos por não serem relevantes para este estudo.

Bloco	Adaptação	Variantes	Escuro	Claro
A	Base	BA	T01	T03
	Rebordo	BA, RB, RP	T02	T04
B	Base	BA	T07	T09
	Ampliação	BA, AM	T08	T10
C	Base	BA	T12	T14
	Cor	BA, C1, C2	T13	T15
D	Base	BA	T18	—
	Letras	BA, LB	T19	—
E	Todos	BA, C1, C2, AM, RB, RP, LB	T20	T21

Tabela 1: Testes por adaptação e tipo de fundo (a adaptação LB não foi testada com fundo claro)

Em cada um dos blocos A, B, C e D comparámos um tipo de adaptação com o símbolo base. Por exemplo, no bloco A, no teste T01 apresentou-se uma imagem de fundo escuro com símbolos base. A mesma imagem foi mostrada posteriormente com símbolos base, símbolos com rebordo branco e símbolos com rebordo preto no teste T02. No bloco E foram mostrados todos os tipos de adaptação em simultâneo. Independentemente do teste, foram sempre colocados 13 símbolos em cada imagem.

A variável dependente do estudo foi a indicação do símbolo preferido nos testes em que mais do que um tipo de símbolo foi apresentado. Também anotámos comentários feitos pelos participantes.

3.3.5 Procedimento

Cada participante no estudo sentou-se em frente ao computador após o que um dos investigadores ajustou a posição do ecrã, tendo também feito uma breve apresentação acerca desta investigação e indicado uma estimativa da duração do estudo: entre 20 a 25 minutos.

O investigador preencheu um breve questionário sobre os dados pessoais do participante: idade, género, habilitações literárias e familiaridade com a pesquisa de pontos de interesse em mapas.

Os blocos de testes A, B, C e D foram realizados por ordem aleatória, seguidos do bloco E. Em cada um dos quatro primeiros blocos, mostrámos uma imagem de fundo escuro, primeiro com símbolos base e depois com símbolos base e símbolos com a adaptação correspondente ao bloco. A disposição dos símbolos foi mudada em imagens consecutivas. Em seguida repetimos o procedimento com uma imagem de fundo claro (excepto no bloco D). O bloco E incluiu duas imagens, uma escura e a

outra clara, cada uma delas com todas as variantes das adaptações.

3.3.6 Resultados

A Figura 2 resume os resultados dos testes efectuados. A proporção representada por cada coluna foi obtida dividindo o número total de opiniões favoráveis a uma adaptação pelo número de possibilidades de esta ser escolhida. Por exemplo, a adaptação rebordo (RE) poderia ser escolhida em 4 testes (ver testes T02, T04, T20, e T21 na Tabela 1), quer como rebordo branco (RB), quer como rebordo preto (RP). Como cada teste foi realizado por 55 participantes, houve 220 possibilidades de escolha dessa adaptação, que foi a preferida em 91% dos casos.

A Figura 3 apresenta as preferências em termos de cada um dos tipos de fundo e de cada uma das adaptações.

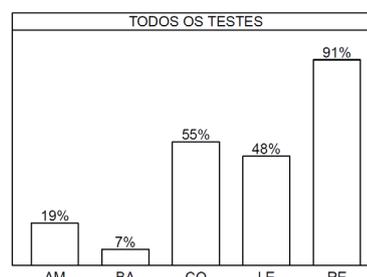


Figura 2: Preferências pelas principais adaptações

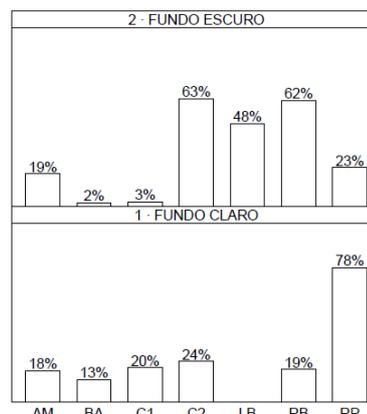


Figura 3: Preferências por tipo de fundo

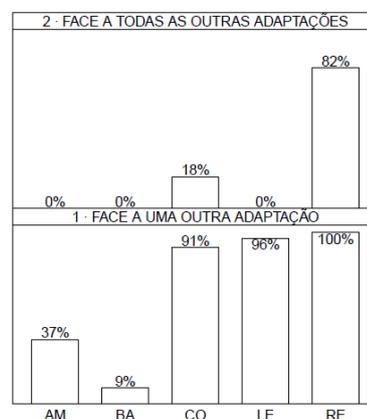


Figura 4: Preferências por contexto competitivo

Por fim, a Figura 4 mostra as preferências de cada adaptação em dois contextos distintos: quando em competição apenas com o símbolo base, correspondendo aos resultados dos blocos A, B, C e D; e face a todas as outras adaptações, correspondendo aos resultados do bloco E.

3.3.7 Discussão

A Figura 4.1 (gráfico de baixo) evidencia que a adição de rebordo (RE), o ajuste da luminosidade da cor (CO), e a alteração da cor da letra (LE), todos com preferências acima dos 90%, contribuíram para realçar os símbolos quando em competição com o símbolo base (BA), validando as hipóteses que colocámos no início sobre estas adaptações. Apenas a ampliação (AM) não favoreceu o realce do símbolo pois não atingiu os 50% quando em competição com o símbolo base e, quando em confronto com todas as outras adaptações, nunca foi a escolhida (Figura 4.2, gráfico de cima). A alteração da cor da letra, apesar de ter sido preferida em relação ao símbolo base também nunca foi seleccionada quando confrontada com todas as outras adaptações em simultâneo.

Na Figura 2, a adição de rebordo foi a adaptação favorita dos participantes, com 91% das preferências. Acresce que, em imagens com fundo claro foi preferida a adaptação com rebordo preto (Figura 3.1) e o rebordo branco foi o mais escolhido com fundos escuros (Figura 3.2).

A adaptação preferida em segundo lugar foi o ajuste da luminosidade da cor (Figura 2), que apesar de ter um resultado próximo do da alteração da cor da letra (55% contra 48%), e ao contrário desta última, foi escolhida por alguns participantes quando em competição com outras adaptações (Figura 4.2). Finalmente, da Figura 3.2 resulta que com fundos escuros, houve uma preferência para ajustes maiores da luminosidade da cor do símbolo (63% de C2 contra 3% de C1), situação que não se verificou com fundos claros, em que tanto C1 como C2 estiveram quase empatados (com 20% e 24% respectivamente). Este resultado motivou a realização de um segundo estudo que descrevemos a seguir.

4. ESTUDO SOBRE AJUSTE DA LUMINOSIDADE

O ajuste de luminosidade foi a segunda adaptação preferida pelos participantes do primeiro estudo. Em imagens escuras, a mudança de luminosidade mais escolhida correspondia a um maior afastamento da cor inicial, enquanto nas imagens claras havia um número maior de participantes a escolher uma mudança de luminosidade menos abrupta. Surgiu por isso a necessidade de esclarecer qual seria o ajuste mínimo de luminosidade adequado. Neste estudo, não foram colocados caracteres no símbolo para reduzir a possibilidade de identificação do símbolo através dos caracteres em vez da luminosidade.

Para preservar a semântica do símbolo é fundamental garantir que não haja uma alteração abrupta da cor original. Por este motivo optámos por modificar apenas a luminosidade. Contudo, o modelo RGB não contempla directamente esta componente. O modelo de cor HSV, ao invés do RGB, interpreta a cor segundo 3 componentes: a cor propriamente dita, a saturação e a luminosidade.

Além disso, separa a componente luminosa da restante informação cromática, consequentemente, o modelo HSV torna-se vantajoso no processamento de imagem [Asmare09].

Pretendemos então determinar qual a variação mínima de luminosidade para que um utilizador, sem limitações na percepção da cor, consiga distinguir os símbolos sobrepostos sobre imagens reais, nomeadamente fora de portas. De facto, em aplicações de Realidade Aumentada usadas em ambientes exteriores é difícil controlar as condições de luminosidade, podendo esta chegar a variar entre 1 e 100.000 lux [Gabbard06].

Tendo em conta este objectivo analisámos as variações de luminosidade descritas na secção seguinte e efectuámos um estudo com utilizadores para as avaliar.

4.1 Variações na luminosidade

De forma a estudar a situação mais adversa utilizámos um fundo de cor uniforme ao qual se sobrepuseram símbolos quadrados de cor uniforme de modo a minimizar interferências. De forma a validar qual a variação mínima de luminosidade necessária para distinguir elementos gráficos sobre um fundo liso, realizámos alguns testes preliminares com elementos da equipa. Dividimos o intervalo de variação da luminosidade em 10 intervalos de igual amplitude. Tendo em conta que a luminosidade no modelo HSV varia no intervalo [0,1], considerámos variações de luminosidade de 0.10. Verificámos que na maior parte dos casos variações inferiores a 0.20 eram suficientes para distinguir os elementos. Como no estudo descrito na secção anterior havíamos verificado que a cor do fundo pode influenciar a distinção dos símbolos sobre o fundo, optámos por testar 4 variações de luminosidade: 0.10, 0.15, 0.20 e 0.25.

4.2 Aplicação desenvolvida

Para construir os testes com utilizadores desenvolvemos uma aplicação usando a linguagem Java, com o SDK para Android.

A aplicação mostra imagens com fundo de uma única cor ao qual se sobrepõem 4 quadrados (30 pixels de lado), cada um deles mantendo as mesmas componentes de cor e saturação da cor do fundo, mas variando a luminosidade acrescentando ou diminuindo 0.1, 0.15, 0.2 ou 0.25 (Figura 5). Ou seja, se a cor de fundo tiver luminosidade acima de 0.5, cada um dos quadrados reduz a luminosidade com um dos valores acima. Caso contrário, os valores acima são somados à cor base.

De notar que, em imagens sucessivas, a posição de cada quadrado é alterada para evitar a sua identificação através da sua localização independentemente da cor.

4.3 Estudo com utilizadores

4.3.1 Participantes

Neste estudo participaram 18 pessoas, 7 do género feminino e 11 do masculino, que se voluntariaram sem receber qualquer tipo de compensação monetária. A média de idades foi de aproximadamente 28 anos. Houve 12 participantes com idades compreendidas entre os 17 e os 24

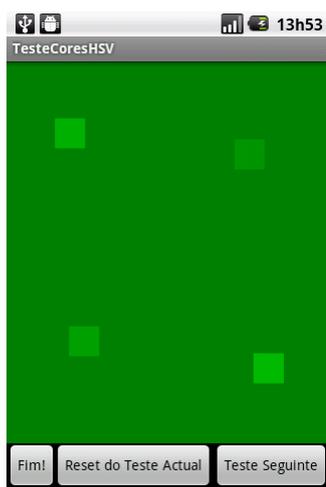


Figura 5. Exemplo de imagem da aplicação.

anos, 3 entre os 25 e os 39 anos e os restantes 3 entre os 40 e os 53 anos. A nível de habilitações literárias, 6 dos participantes não eram graduados, 9 eram licenciados e os restantes 3 doutorados. Nenhum dos participantes era daltónico.

4.3.2 Aparato

Os testes foram efectuados num *smartphone* LG P500 com o sistema operativo Android 2.2, com ecrã táctil de 3.2 polegadas e com uma resolução de 320×480 *pixels*.

O participante realizou os testes no exterior em dias de sol e às horas de maior intensidade luminosa, em local exposto directamente à luz do Sol ou numa zona de sombra adjacente. Ou seja, em condições de luminosidade adversas para a observação de imagens. Nos testes utilizámos a aplicação descrita na Secção 4.2.

4.3.3 Tarefas

O participante teve como tarefa seleccionar, em cada imagem, através de toques em símbolos, apenas aqueles que conseguia distinguir bem do fundo, ignorando os restantes. O participante seleccionou os símbolos por ordem, do menos para o mais perceptível.

4.3.4 Design

Neste estudo foram manipuladas duas variáveis independentes, a cor do fundo e a variação da luminosidade dessa mesma cor, e a variável dependente foi a luminosidade mais baixa que o participante conseguiu diferenciar.

Considerámos seis cores diferentes para o fundo e, para cada uma delas, quatro variantes diferentes de luminosidade: 0.25, 0.50, 0.75, 1.00. Deste modo, cada participante foi exposto a um total de 24 imagens.

A selecção das cores de fundo baseou-se, por um lado, em serem cores comuns em espaços exteriores e, por outro lado, em serem cores mais ou menos fáceis de distinguir de acordo com os três tipos de células (cones) que o olho humano possui para perceber a cor [Bruce03].

As seis cores escolhidas para fundo foram as seguintes:

1. Azul puro: normalmente de percepção mais difícil para o olho humano;

2. Amarelo: normalmente uma das mais fáceis de distinguir para o olho humano;

3. Verde puro: as variantes mais escuras desta cor são muito frequentes na natureza;

4. Roxo: requer a utilização dos três tipos de cones;

5. Azul claro: muito frequente na natureza;

6. Branco: muito comum nos ambientes exteriores.

Caso o fundo apresente uma luminosidade de 1.0 ou 0.75 os símbolos sobrepostos apresentarão uma intensidade igual à do fundo menos 0.10, 0.15, 0.20 e 0.25. Caso o fundo apresente uma luminosidade de 0.5 ou 0.25, os símbolos sobrepostos apresentarão uma intensidade igual à do fundo mais 0.10, 0.15, 0.20 e 0.25. Cada um destes símbolos foi distribuído aleatoriamente por cada um dos quadrantes para que o participante não adquirisse efeitos de habituação.

4.3.5 Procedimento

A cada participante apresentámos os objectivos do estudo, indicámos o tempo estimado de duração dos testes (entre os 10 e os 15 minutos) e fizemos um breve questionário sobre os seus dados pessoais.

Em seguida explicámos como deveria seleccionar os quadrados em cada imagem. As 24 imagens foram apresentadas pela mesma ordem a todos os participantes.

4.3.6 Resultados e discussão

Para cada participante foram registados num ficheiro os símbolos identificados e a ordem pela qual foram sendo escolhidos.

A Figura 6 apresenta 4 blocos de diagramas de extremos e quartis, cada um deles relativo a uma variação de luminosidade do fundo.

Verificámos que quando o fundo tem uma luminosidade que não é extrema (ou seja, blocos de 0.5 e 0.75), os diagramas mostram que os quartis e os extremos coincidem (sempre no 0.75 e quase sempre no 0.50) com a variação mais pequena da luminosidade (0.10). Ou seja, basta considerar uma variação de luminosidade de 0.10 para que o símbolo apresente um contraste suficiente para ser visível, qualquer que seja a cor de fundo analisada.

Caso o fundo tenha pouca luminosidade (bloco de 0.25, à esquerda), verificámos que, para todas as cores de fundo analisadas, os diagramas apresentam a mediana em 0.15 para 5 das 6 cores. Para a maioria das cores o terceiro quartil encontra-se em 0.20 e numa delas em 0.25. Ou seja, neste caso, a variação na luminosidade necessária para fazer contraste é maior e apenas com a variação de 0.25 todos os participantes distinguiram os símbolos.

No caso da luminosidade do fundo mais elevada (bloco 1.00, à direita), o comportamento das cores de fundo estudadas foi mais irregular. Enquanto para o azul puro o diagrama apresenta o mínimo e o máximo no valor 0.25, para o amarelo estes extremos situam-se no valor 0.10. Para as outras cores, os resultados mostram uma maior dispersão.

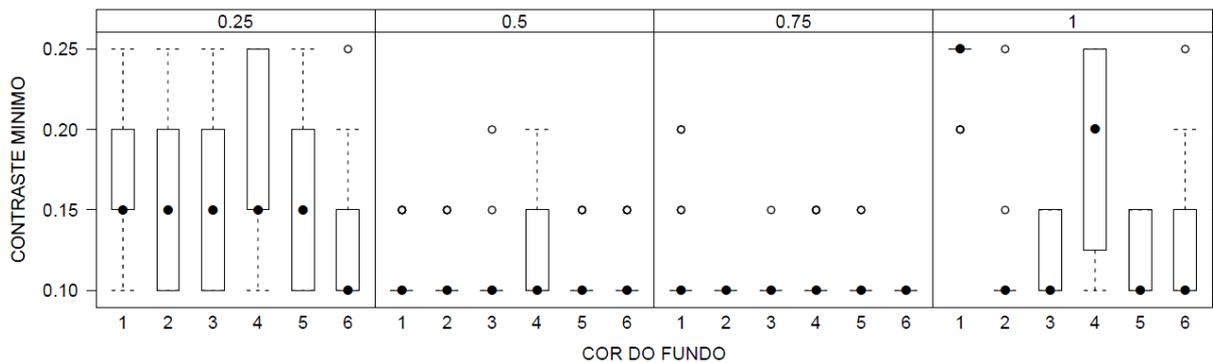


Figura 6. Contraste mínimo para cada cor em função da luminosidade do fundo (blocos 0.25, 0.5, 0.75 e 1).

5. CONCLUSÕES E TRABALHO FUTURO

Neste artigo apresentámos o resultado de um estudo sobre as preferências dos utilizadores sobre a adaptação de simbologia para realçar os elementos gráficos em aplicações de RA. Concluímos que os utilizadores preferiram a adição de um rebordo e o ajuste da cor em termos de luminosidade. Tendo em atenção estes resultados, realizámos um segundo estudo para averiguar qual a variação mínima da luminosidade para que um símbolo se destaque da cor do fundo.

Concluímos que se quisermos estabelecer um valor mínimo da variação de luminosidade para garantir que o utilizador distingue os símbolos do fundo, deverá ser considerado o valor 0.25. Contudo, podem existir soluções mais flexíveis, que tenham em conta variações inferiores para imagens com luminosidades intermédias.

Como trabalho futuro, pretendemos aplicar os resultados obtidos numa plataforma de desenvolvimento de aplicações de RA móvel, de modo que a simbologia seja adaptada em tempo real em função da imagem captada.

Estes resultados também podem ser aplicados à sobreposição de informação em vídeo, em particular, na colocação de legendas.

6. AGRADECIMENTOS

Agradecemos à FCT e ao LabMAg o apoio financeiro concedido aos bolsiros Raúl Simplicio e Edgar Montez. Agradecemos também a todos os que participaram na realização dos testes.

7. REFERÊNCIAS

- [Asmare09] M. H. Asmare, V.S. Asirvadani, L. Iznita. Color space selection for color image enhancement applications. *Proc. ICSAP'09*, pp. 208-212, 2009.
- [Bruce03] V. Bruce, P. R. Green, M. A. Georgeson. *Visual perception: Physiology, psychology and ecology*, 4ª edição, Psychology Press, Hove, 2003.
- [Gabbard06] J. L. Gabbard, J. E. Swan, D. Hix. The effects of text drawing styles, background textures, and natural lighting on text legibility in outdoor augmented reality. *Presence: Teleoperators and Virtual Environments*, 15(1), pp. 16-32, 2006.
- [Gabbard07] J. L. Gabbard, J. E. Swan, D. Hix, S. J. Kim, and G. Fitch. Active text drawing styles for outdoor augmented reality: A user-based study and design implications. *Proc. VR'07*, pp. 35-42, 2007.
- [Gruber10] L. Gruber, D. Kalkofen, D. Schmalstieg. Color harmonization for augmented reality. *Proc. ISMAR'10*, pp. 227-228, 2010.
- [Kalkofen09a] D. Kalkofen, S. Zollman, G. Schall, G. Reitmayr, D. Schmalstieg. Adaptive visualization in outdoor AR displays. Workshop on Outdoor Mixed and Augmented Reality, 2009.
- [Kalkofen09b] D. Kalkofen, E. Mendez, D. Schmalstieg. Comprehensible visualization for augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 15(2), pp. 193-204, 2009.
- [Layar] Layar. www.layar.com (acesso em Maio 2012)
- [Leykin04] A. Leykin, M. Tuceryan. Automatic determination of text readability over textured backgrounds for augmented reality systems. *Proc. ISMAR'04*, pp. 224-230, 2004.
- [Nivala07] A.-M. Nivala, T. L. Sarjakoski. User aspects of adaptive visualization for mobile maps. *Cartography and Geographic Information Science*, 34(4), pp. 275-284, 2007.
- [NyARToolkit] NyARToolkit Project Page, <http://nyatla.jp/nyartoolkit/wp/> (acesso em Maio 2012)
- [Paley03] W. B. Paley. Designing better transparent overlays by applying illustration techniques and vision findings. *Adjunct Proc. UIST'03*, pp. 57-58, 2003.
- [Silva11] S. Silva, B. S. Santos, J. Madeira. Using color in visualization: A survey. *Computers and Graphics*, 35(2), pp. 320-333, 2011.
- [White09] S. White, S. Feiner. SiteLens: Situated visualization techniques for urban site visits. *Proc. CHI'09*, pp. 1117-1120, 2009.
- [Wikitude] Wikitude World Browser. www.wikitude.org (acesso em Maio 2012)
- [Wolfe04] J. M. Wolfe, T. S. Horowitz. What attributes guide the deployment of visual attention and how do they do it? *Nature Reviews Neuroscience*, 5(6), pp. 495-501, 2004.

Interaction with large displays in a public space using the Kinect sensor

Igor Duarte Cardoso
DETI / IEETA
Universidade de Aveiro
idc@ua.pt

Paulo Dias
DETI / IEETA
Universidade de Aveiro
paulo.dias@ua.pt

Beatriz Sousa Santos
DETI / IEETA
Universidade de Aveiro
bss@ua.pt

Resumo

Com a evolução da tecnologia, monitores de computador e televisores têm-se tornado cada vez mais finos e baratos, o que permitiu a sua colocação em muitos espaços públicos tais como aeroportos, halls de entrada de edifícios, etc. para apresentação de informação a quem frequente esses espaços. No entanto, a utilização destes ecrãs apresenta alguns desafios sobretudo quanto à forma como a informação é apresentada e como se pode interagir com a mesma. Em trabalhos anteriores estudámos a possibilidade de utilizar telemóveis como dispositivos de interação com estes ecrãs; neste artigo, apresentamos uma evolução que utiliza o Kinect como dispositivo de entrada permitindo um controlo da informação apresentada no ecrã sem recurso a hardware adicional. São descritas as modificações ao protótipo original (desenhado para telemóveis) e duas propostas de interação diferentes: "Gesture-Based Interaction" e "Pointer-Based Interaction". Finalmente, apresentam-se os resultados dos testes efectuados para avaliar as escolhas realizadas e testar a viabilidade do sistema para interagir com ecrãs em espaços públicos.

Abstract

As technology evolves, computer and television screens have become increasingly thinner and cheaper. This has allowed these screens to be used in public locations such as airports, lobbies of buildings, etc., presenting relevant information to the people passing by. The use of these screens, however, presents a challenge in how information is displayed and how people can interact with it. Often, they do not provide any possible interaction and displayed information cannot be controlled at all by the users. In previous work, we studied the possibility to use mobile phones as interaction devices for these large public displays. In this paper we present an evolution using Kinect to allow a device free interaction avoiding the need to use any additional hardware. We present the modifications to the original prototype (for mobile phones) and two different interaction proposals: "Gesture-Based Interaction" and "Pointer-Based Interaction". Finally, we evaluated our choices and tested the viability of the system to interact with large displays in a public space.

Keywords

Kinect, large displays, gesture recognition, natural interaction, user interfaces

1. INTRODUCTION

With the reduction of size, weight and cost of computer and television screens, it is now common to find large displays available in public spaces, such as airports, train stations, waiting rooms, museums, etc. These displays often do not provide any interactivity and users cannot control in any way the displayed information. At the same time, major new products in the field of human-computer interaction have started emerging, allowing more natural user interfaces. Examples of such are innovative controllers, like the Wii Remote, Playstation Move, Kinect and other less known devices, that have been widely accepted by the community [Takala12]. Moreover, many types of sensors deeply integrated into smartphones, tablets and the like, enable richer user experiences and interaction.

In previous work, we studied the possibility of using mobile phones to allow user to interact with the content of large passive public displays using their own devices and with no need to buy additional hardware [Duarte11]. This solution, although interesting presents some limitations since it requires the user to download an application on their mobile devices and implies initial pairing steps for communication. A prototype called "DETI-Interact" was developed and is currently working in the lobby of the Department of Electronics, Telecommunications and Informatics of the University of Aveiro (see Figure 1). However, the number of users in this real scenario is reduced probably due to the initial configuration steps that most of them will not perform due to lack of trust (the application runs in their devices) and the time it takes to make the first

configuration.



Figure 1: User interacting with DETI-Interact

To overcome the initial configuration problem, we propose to use an alternative interaction device that does not belong to the user and allows natural gesture interaction. Thus, DETI-Interact has been re-imagined to work with the Kinect sensor. The user interface and the architecture have been adapted to this new device, maintaining compatibility with Android Bluetooth-enabled devices (though only one the two: Kinect or Android, can be active at a given time).

2. RELATED WORK

Kinect is a motion sensing input device developed by Microsoft and launched on November 2010. It allows tracking of physical objects and full body skeletons in a 3-dimensional space. The Kinect sensor includes: an RGB camera, an infrared laser projector, a monochrome CMOS sensor, a microphone array, a 3-axis accelerometer and other components less relevant for the interaction. Initially launched for the game industry on the Xbox 360 platform, some software development kits (SDK) and several toolkits appeared in the community, allowing development of Kinect-powered applications for the computer. Some SDKs are: OpenNI [Ope10b], OpenKinect [Ope10a] and the CL NUI Platform [CLN10]. More recently, Microsoft itself launched the final version of the Kinect for Windows SDK [Kin11a].

Besides being used the game industry and in some research projects, for instance in aiding the visually impaired [Zöllner11], modernization of user interaction classes [Villaroman11] and several hobbyist projects, Kinect has also spawned interest in important consumer electronics brands such as Samsung, which has recently released new solutions using similar sensors to control TV sets [Sam12].

Previous works, more aimed at interaction with large displays or surfaces, have tested how appropriate some gestures can be using Kinect [Hespanhol12] [Reis12]. Another work has joined surface-based interaction with space-based interaction into one, as a continuum of interactivity where one can start and stop gestures or movements in any part of the space including the touch surface of the tabletop system used [Marquardt11].

3. DETI-INTERACT

The basic functionalities of DETI-Interact are navigation on the Faculty list, department schedules, and the world

map. In the previous version [Duarte11], the only interaction method with the content was done by using a Bluetooth-enabled mobile device with touchscreen and accelerometer (running Android operating system).

The technologies used were based on .NET, with C# as the programming language, along with Windows Presentation Foundation (WPF) for a modern user interface, the Google Earth plugin to provide navigation in the world map (plus JavaScript for interacting with it) and Android SDK to develop the mobile application. The application also made use of internal web services from the Department (specifically, DSD) in order to obtain Faculty and schedules information as can be seen in Figure 2, depicting the general architecture.

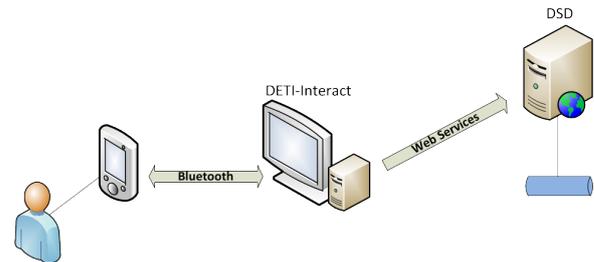


Figure 2: Previous general architecture

Now, DETI-Interact provides two new interaction methods that have been carefully developed, tweaked and finally submitted to usability tests. The goal is to find out which is the best interaction method and/or whether they can be merged into a single "best of both worlds" interaction method that will finally be deployed in our Department.

3.1. Adaptation to Kinect

To provide support for Kinect interaction, the Kinect for Windows SDK was chosen, integrating more properly with the DETI-Interact implementation (based on Microsoft technology) and being capable and generally recommended for full body tracking [Takala12]. Also, a library was forked from the Kinect Toolbox [Kin11b] gesture recognition and improved to provide more actions and better suit the needs of DETI-Interact.

Using the library to recognize gestures, a list of the last captured frames is stored in what we call the gesture window. It has a limited number of frames and is used to calculate the variation of user movements. The only gesture detected, quickly moving the hand in one direction, is identified by comparing every element in the gesture window. The element is configurable to be any of the 20 joints detected by Kinect for Windows SDK. The parameters used to compare these elements and decide if a gesture has been found are: the minimum distance of the gesture, whether it was between a predefined interval of time, was it stable in its path, i.e., the movement approximately followed a straight line, and whether it did not invert direction. The direction axes supported by this gesture are the horizontal, vertical and longitudinal ones, relatively to the Kinect

sensor.

Besides gesture detection, more types of actions have been added to this Kinect library, including one that periodically triggers when a user's joint is within a predefined range.

This adaptation to the Kinect sensor turned the architecture into what can be seen in Figure 3. Support for Kinect sensor was added without having to remove the Android one, made possible by developing the application in a way that generically supports any interaction device, having a middle layer to properly convert device-specific actions to UI-specific actions (enter, exit, zoom, and others), making the device abstract to DETI-Interact.

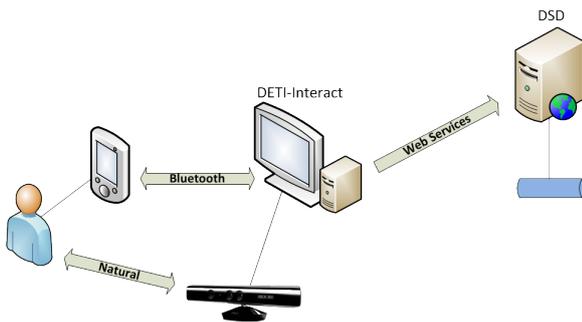


Figure 3: Current general architecture

3.2. User Interface

The overall appearance of DETI-Interact user interface has not changed much since the previous version, though some details have been tweaked specifically to improve interaction using the Kinect sensor. Overall, space has been optimized by removing padding and borders, font size has been increased on critical areas as well as the width of page tabs.

The user interface is based on pages that can be seen one at a time. They are arranged in an horizontal layout with tabs ("page tabs") at the right and left side which show the names of the pages they refer to. The major change from the previous version is the Gesture Tracker: a small "visor" that shows the relative position of the user's "navigation" hand (see Figure 4). This allows users to easily have feedback about their hand position while using the Gesture-Based Interaction without disturbing them too much. The last detected positions of the hand are shown in the Gesture Tracker as small circles.

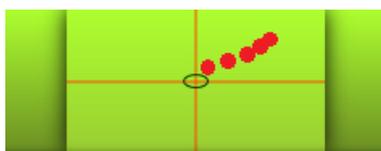


Figure 4: Gesture Tracker

The three basic functionalities are contained in three different pages. The title bar, at the top of every page, shows from left to right, the title, the Gesture Tracker, the current time and date. The user help information is shown at the bottom as in the previous version.

The first page, the Faculty viewer, presents a scrollable grid-based view on Faculty members, each shown in a business card fashion, containing the name, photo, office number, phone extension and webpage URL (Figure 5). For interaction and feedback purposes, four arrows are displayed around the grid.



Figure 5: Faculty viewer

The application allows the selection of a Faculty member to further open his/her webpage (Figure 6), using all the available space on the screen page. Depending on the interaction, an arrow is placed on the title bar (see Figure 9) providing the action to go back.



Figure 6: Faculty member webpage

Another different page is the Schedule viewer (see Figure 7), which allows navigating the Department schedule showing a full view of one timetable at a time.

The classes that are supposed to occur, as provided by the web services, are then arranged in the timetable with days of the week as rows and hours of the day as columns. At the top of the timetable are the Study Program and corresponding year selected.

As in the Faculty viewer, arrows are shown for interactivity and feedback purposes.

The final page is the world map (Figure 8). Although it has not been subject to formal usability tests because our main purpose was testing GBI against PBI in simple navigation contexts, some users actually interacted with it off the record. According to their feedback, interaction with the map seems to be an interesting experience. They have successfully found multiple locations on the map. In this version of DETI-Interact, the map has been max-

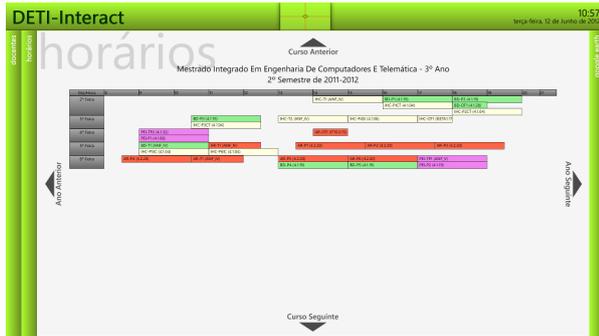


Figure 7: Schedule viewer

imized to fit all the space available on the screen page. The move around the map only one hand is required. Unlike [Stannus11], which provides a more intuitive yet complex way of interacting with the globe by having the users first grab it via closing their hands, we have a simplified implementation where latitude, longitude and altitude variation speeds are linearly controlled by the hand's position in the space. For instance, the farther away from the sensor, the faster the altitude increases (or the map zooms out). The same applies for latitude and longitude where moving the hand to the right rotates the earth towards east and moving the hand to the top rotates the earth towards north, and vice-versa. There is no need to first grab the globe nor distinguish between panning and zooming in an explicit way, in terms of individual gestures used.



Figure 8: World map

3.3. Interaction Methods

As mentioned earlier, there are two new interaction methods in the current DETI-Interact: Gesture-Based Interaction (GBI) and Pointer-Based Interaction (PBI), both using the Kinect sensor.

The most important operations that can be done in DETI-Interact and its two main pages (Faculty viewer and Schedule viewer) are the following:

[Because there is always not much to do in a DI page, the gestures can be reused better without being ambiguous, and cite Hespanhol here]

- Alternate between pages

- Find a Faculty member, i.e, navigate until selection is possible
- Select Faculty member to open his/her webpage
- Go back from the webpage, i.e, return to the Faculty member list
- Select a specific timetable

The actions detected by the developed Kinect library are:

- **Swipe**
Consists in the gesture of quickly moving the hand from one side to the other with enough amplitude. Vertically, when the hand moves from top to bottom, it is named a "Swipe Down". In the opposite direction, it is a "Swipe Up". Horizontally, when the hand moves from left to right, it is named a "Swipe Right". In the opposite direction, it is a "Swipe Left".
This action is highly configurable with a variety of attributes like: sleep time (duration of time the system will reject new gestures after a successful one), length of the gesture, maximum time of the gesture, tolerance for unstable gestures and minimum speed of the gesture (by the relative increase in distance).
- **Push/Pull**
Consists in the gesture of quickly moving the hand towards or away from the sensor. The concept of "Push" is the same as in [Hespanhol12] "Pushing" but we also provide the opposite, called "Pull". When the hand moves towards, it is named "Push". In the opposite direction (away from), it is "Pull". Push and Pull are exactly like a Swipe but in the longitudinal direction.
- **Stay**
This action just acquires the relative user's hand position with a great number of samples per second, very useful for showing the position in Gesture Tracker or for scrolling inside a page. Although it isn't exactly a gesture, but rather a constant refresh of the user's hand position, it is normalized and highly configurable. For example, it is possible to set the amount of samples per second and a virtual central point in the Z axis of the space, which is currently useful for zooming in Google Earth.
- **Timed Stay**
Partly based on the "dwelling" gesture by [Hespanhol12], "Timed Stay" works by hovering a zone for a specified amount of time to activate the corresponding action. It differs from "dwelling" because if the hand is within a specified range near each of the four edges of the field of view, the action will be periodically triggered (for instance, each second) and coupled with the actual direction, instead of triggering the action in any part of the screen.

These actions can be mapped to any hand and are the base for both methods of interaction.

3.3.1. Gesture-Based Interaction

The Gesture-Based Interaction (GBI) consists of simple hand gestures and positions, all done with only one hand. In the usability tests, the right hand was chosen only because the majority of users are right.

For the operation of alternating between pages, a Swipe action is done. When doing Swipe Right, the next page at the right expands and replaces the current one. In the opposite direction ("Swipe left"), the contrary occurs.

In order to find a Faculty member, users first need to use the Push action to activate the selection mode. In this mode, users navigate through the list using the Timed Stay action. This selects the next Faculty member in any of the directions (one per second, as configured for the usability tests) until the one sought is found. The Faculty member that is currently selected will have his/her "business card" color changed to provide feedback. Also, the correct arrow (there are four arrows around the list) will change color each time the action triggers. Users can also make use of the Stay action to quickly scroll the list, only doing a Push in the end.

For the operation of viewing the webpage of a Faculty member, first users should already have a Faculty member selected. Moreover, they just need to make another Push and the webpage will automatically be shown in the user interface.

To go back from a webpage, users do a Pull action (the opposite of Push). When users pull from the webpage, they will be taken back to the Faculty member's list. Another Pull and the Faculty member's list is unfocused, again allowing to switch among DETI-Interact pages.

Finally, to select a timetable, users must first use Push to initiate the selection mode (the same as the Faculty member's list), and then Swipe in any of the four directions until the wanted schedule is found. A Swipe Right will slide the timetable to the next on the right (next year). A Swipe Left will do the opposite (previous year). A Swipe Up will slide the timetable to the next above (previous Study Program). A Swipe Down will do the opposite (next Study Program). Each time a timetable is changed, the proper arrow around will change its color to further emphasize the direction taken. To exit selection and be able to change pages again, another Pull is required.

The Push/Pull actions are always used to enter and exit pages. Because the main pages of DETI-Interact can be swapped using the Swipe gesture, there needs to be a way of avoiding accidental changes of those pages. So, when users are interested in interacting with content, they should first activate that content using the Push action. When finished, they use the Pull action to go back to page selection.

3.3.2. Pointer-Based Interaction

The Pointer-Based Interaction (PBI) is a different approach that controls a screen cursor (pointer) with one hand and

selects/enters interface elements with the other hand. We chose to use different hands for selection and navigation because using a single hand for both has precision problems during the Push action. The right hand was chosen for controlling the cursor given that the majority of computer users is trained to use the mouse with this hand.

First, the user interface provides a noticeable cursor (a large green circle) so users know exactly where they are, even when a few meters away from the screen.

For the operation of alternating between pages, users need to hover the page tab desired (the tabs have the name of the page on it) and then select that tab using the other hand by quickly pressing it towards the sensor, through the Push action, simulating a mouse click.

In order to find a Faculty member in the list, users first scroll until the zone where the Faculty member is, made possible with the implicit Stay action. When users find out where the Faculty member is, they can hover it and the UI will provide feedback by changing the color of the Faculty member's "business card".

For the operation of viewing the webpage of a Faculty member, users simply select (Push) the Faculty member that is below the cursor. The webpage will automatically be displayed.

To go back from a webpage, users can use a specific interface element that only appears in Pointer-Based Interaction, which is the back button (see Figure 9). By selecting it with the hand, the Faculty viewer will be visible again. Alternatively, users can select another page if they are not interested in the Faculty viewer again, though when coming back to the viewer the webpage will still be visible.



Figure 9: Back button

Finally, to select a timetable, the arrow for the desired direction needs to be hovered and selected with a Push. The left arrow shows the timetable for the next year, while the right arrow shows the previous year. The up arrow shows the timetable for the previous Study Program, while the down arrow shows the next Study Program year.

4. USABILITY TESTS

With both interaction methods ready and sufficiently robust, usability tests were performed to evaluate the whole interaction, as can be seen in the Figure 10.

Unlike previous works that usually test their systems in isolated spaces in order to focus exclusively in the study of interaction methods, like [Hespanhol12], we have proceeded to usability tests in a public space. However, the tests occurred when the flow of people actually present in the public space, our department, were low but noticeable. Thus, we avoid having the comparison of the interaction methods efficiency compromised due to too many distractions.

Always less than half a dozen people passed by during the approximate 15 minutes of usability tests.

The tests consisted in four specific and realistic tasks to be carried out by the user:

1. Find the phone extension of a given Faculty member
2. Consult the classroom of a given class at a given time
3. Check the address of a given Faculty member
4. Find a specific timetable of a Study Program and finish the test



Figure 10: User interacting with the new DETI-Interact

The configuration parameters of the Kinect actions, especially for the gestures, were tweaked empirically by the developers to be at least acceptable in the usability tests.

Although every user tested both interaction methods, approximately half of them started with the Gesture-Based Interaction and the other half with the Pointer-Based Interaction to minimize learning effects.

The possible actions and operations were explained in general before the start of the test and some hints were given during the test in a way that users got to know what they were supposed to do without knowing the feeling of doing so.

In total, there were 17 participants in the usability tests, 8 of them started with the Gesture-Based Interaction while the other 9 started with the Pointer-based Interaction. These participants were all students of informatics-related courses so they were already trained with standard computer interaction. Of all participants, 15 had used touchscreen devices (like smartphones) before, while the other 2 didn't. Finally, 15 were male while 2 of them were female.

After they finished each task, they were asked to give a score for the difficulty, while their impression of the system was still fresh.

At the end of the test, each participant was given a questionnaire so they could evaluate the user interface and interaction methods in more detail, while also allowing them to write their own feedback. A copy can be accessed through the URL: <http://goo.gl/uq67M> (only available in portuguese).

The user questionnaire consisted in three types of questions: true or false, questions in the Likert scale and open ended questions.

In true or false questions, users were asked personal questions: if they had experience with touchscreen, accelerometer or digital compass powered devices, experience with devices like Kinect, Wii Remote or Playstation Move and what if they had already used the previous DETI-Interact. The results, as shown in Figure 11, indicate that nowadays almost every informatics student has experience with tactile or accelerometer-enabled devices. Also, it shows that more than half have had contact with the Kinect sensor or related devices.

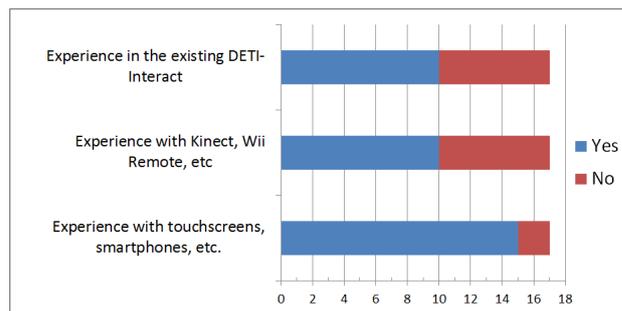


Figure 11: Personal true/false results

For the questions in Likert scale, they were asked general usability questions for each interaction method, for instance whether they felt the navigation was pleasant, intuitive or required much training. They were also asked about the difficulty of each action for each interaction method. Finally, they answered about the general satisfaction in each interaction method.

In the open ended questions, they were invited to comment on the set of gestures used, whether they liked them, and to propose new ideas.

It was observed that when a person is wearing a thick, cotton coats, the precision of skeleton detection decreases thus negatively affecting the interaction.

Some users commented that Pointer-Based Interaction was good, especially for selecting Faculty members, yet most of them took longer using this method of interaction and made negative comments due to limb occlusion from Kinect. When selecting something at the left side of the screen using the Pointer-Based Interaction, the right-arm would go in front of the left-arm, creating additional difficulty to the proper full skeleton detection necessary by the Microsoft Kinect SDK.

Another skeleton detection problem occurs when there are people passing by. As this project is supposed to be deployed in a public space, it is a real concern to improve the detection efficiency and robustness when more people are in front of the sensor.

Other common complaints and errors of interaction include: the non intentional input of gestures, when someone repositioned himself/herself more abruptly; the non-

responsive time right after a successful gesture, consequence of the sleep time implemented that prevents the movement of returning from a swipe to be detected as another swipe;

We also observed that 15 of the 17 users intuitively wanted the swipe to be in the opposite direction, i.e, instead of moving from left to right to show something that was at the right side, they wanted to virtually grab the page with their hand and slide it from right to left in order to view the page that was hidden at the right side. The direction we chose seemed appropriate due to the lack of a surface to grab the DETI-Interact pages. We think that the different direction users tended to take might be partly related to their previous experience with touch-devices, like smartphones. If our system was based on a continuum interaction space, as introduced by [Marquardt11], and the swipe started by touching the surface and swiping it to the desired direction (thus grabbing and dragging the page), we argue that we and every user would agree on the same direction.

In general, the Pointer-Based Interaction was worse, as indicated by the user satisfaction (see Figure 12), the median difficulty of actions (see Figure 13) and the median difficulty felt by the users in performing each task (see Figure 14); however, user performance was better with Pointer-Based Interaction in task 3: "Check the address of a given Faculty member" as can be seen in Figure 15. What we observed is in agreement with these charts. As it was expected, PBI is especially difficult when selecting areas of the screen that are close to the edges, especially at the edge opposite to the cursor hand (in our tests, the left edge of the screen) since the arms get interlaced and proper detection becomes difficult. However, when selecting something in the middle of the screen, PBI works well and is faster, the greatest difficulty being the successful detection of a push gesture that triggers a click. Such is the case of selecting a Faculty member to view the corresponding webpage and address, as in task 3. The feedback provided by the users also supports this. They presented some alternative ways of executing the clicks that do not have the problem of interlaced arms. One way would be to close the selection hand (left hand) to trigger a click, which cannot be done in Kinect for Windows SDK using skeleton frames due to its lack of precision. Another way would be to nod the head down for the same effect, thus only requiring one hand.

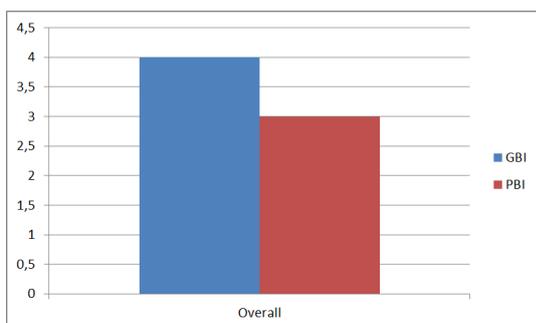


Figure 12: Overall satisfaction

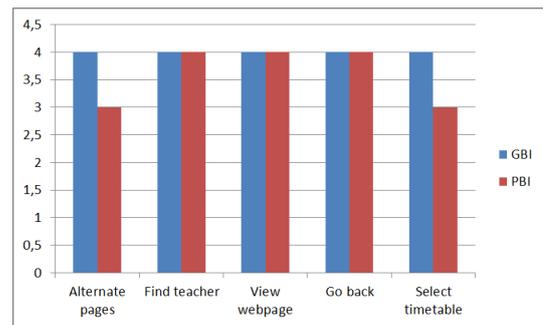


Figure 13: Median difficulty of actions

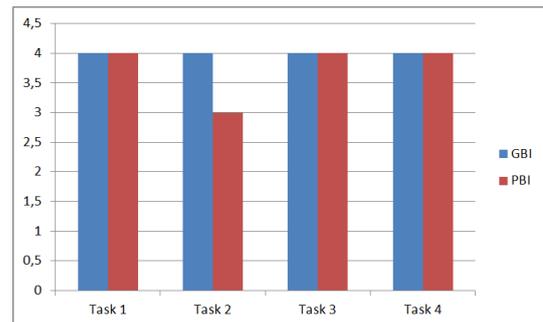


Figure 14: Median difficulty of tasks

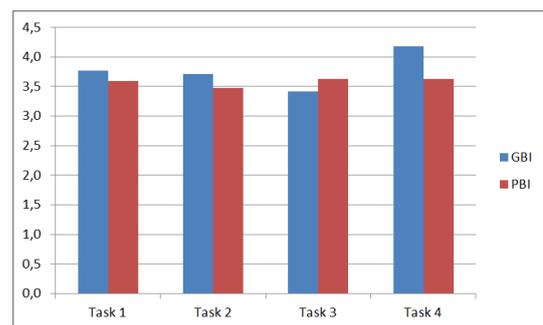


Figure 15: Average difficulty of tasks

5. CONCLUSIONS AND FUTURE WORK

Results have shown that for most of the UI elements, users preferred the Gesture-Based Interaction over the Pointer-Based Interaction. However, when users are presented with multiple selectable elements spread along the screen, they want to select just the one they intend to, without having to go through all the others. Therefore, Pointer-Based Interaction seems more appropriate in this case.

On the other hand, when presented with a simple screen with few selectable information, users prefer to swipe in order to select the next piece of information they are looking for, instead of "pushing" specific buttons to navigate. In this class, Gesture-Based Interaction simplifies and is overall better.

Given the mixed interaction advantages, a hybrid solution seems the most appropriate for a final implementation scenario, as we already expected.

During the usability tests, we noticed that most users needed some help in the beginning to interact with the system. However, after a short period of time, they were able to use the interface without major problems. Since normal users are not supposed to have any training, it is important to include help and other affordances.

Based on the results of this work, we plan to develop a new version of DETI-Interact with the hybrid solution, for deployment on the lobby of our Department and further logging/testing. Further improvements may include activation methods to ensure that the system will respond only to interested users. A possible solution may use a camera (already available in the Kinect sensor) and face recognition/tracking software to detect users staring at the monitor.

6. ACKNOWLEDGEMENTS

This Research was partly is funded by FEDER through the Operational Program Competitiveness Factors - COMPETE and by National Funds through FCT - Foundation for Science and Technology in the context of the project FCOMP-01-0124-FEDER-022682 (FCT reference PEst-C/EEI/UI0127/2011)

Our thanks also to all the students that participated in the evaluation providing valuable contributions.

7. REFERENCES

- [CLN10] CL NUI Platform. <http://codelaboratories.com/nui>, 2010. [Online; accessed 15/06/2012].
- [Duarte11] Filipe Duarte, Paulo Dias, and Beatriz Sousa Santos. Deti-interact: Interacting with public displays through mobile phones. In *6ª Conferência Ibérica de Sistemas e Tecnologias de Informação (CISTI 2011), Workshop on Information Systems for Interactive Spaces*, pages 787–792. IEEE, 15-18 June 2011.
- [Hespanhol12] Luke Hespanhol, Martin Tomitsch, Kazjon Grace, Anthony Collins, and Judy Kay. Investigating intuitiveness and effectiveness of gestures for free spatial interaction with large displays. In *Proceedings of the 2012 International Symposium on Pervasive Displays, PerDis '12*, pages 6:1–6:6, New York, NY, USA, 2012. ACM.
- [Kin11a] Kinect for Windows SDK. <http://www.microsoft.com/en-us/kinectforwindows/>, 2011. [Online; accessed 15/06/2012].
- [Kin11b] Kinect Toolbox. <http://kinecttoolbox.codeplex.com/>, 2011. [Online; accessed 15/06/2012].
- [Marquardt11] Nicolai Marquardt, Ricardo Jota, Saul Greenberg, and Joaquim A. Jorge. The continuous interaction space: interaction techniques unifying touch and gesture on and above a digital surface. In *Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction - Volume Part III, INTERACT'11*, pages 461–476, Berlin, Heidelberg, 2011. Springer-Verlag.
- [Ope10a] OpenKinect. http://openkinect.org/wiki/Main_Page, 2010. [Online; accessed 15/06/2012].
- [Ope10b] OpenNI. <http://openni.org/>, 2010. [Online; accessed 15/06/2012].
- [Reis12] Bernardo Reis, João Marcelo Teixeira, Felipe Breyer, Luis Arthur Vasconcelos, Aline Cavalcanti, André Ferreira, and Judith Kelner. Increasing kinect application development productivity by an enhanced hardware abstraction. In *Proceedings of the 4th ACM SIGCHI symposium on Engineering interactive computing systems, EICS '12*, pages 5–14, New York, NY, USA, 2012. ACM.
- [Sam12] Samsung ES8000 LED HDTV is Smarter Than Most Phones. <http://goo.gl/Wi497>, 2012. [Online; accessed 15/06/2012].
- [Stannus11] Simon Stannus, Daniel Rolf, Arko Lucieer, and Winyu Chinthammit. Gestural navigation in google earth. In *Proceedings of the 23rd Australian Computer-Human Interaction Conference, OzCHI '11*, pages 269–272, New York, NY, USA, 2011. ACM.
- [Takala12] Tuukka M. Takala, Paivi Rauhamaa, and Tapio Takala. Survey of 3DUI applications and development challenges. In Mark Billinghurst, Joseph J. LaViola Jr., and Anatole Lécuyer, editors, *3DUI*, pages 89–96. IEEE, 2012.
- [Villaroman11] Norman Villaroman, Dale Rowe, and Bret Swan. Teaching natural user interaction using openni and the microsoft kinect sensor. In *Proceedings of the 2011 conference on Information technology education, SIGITE '11*, pages 227–232, New York, NY, USA, 2011. ACM.
- [Zöllner11] Michael Zöllner, Stephan Huber, Hans-Christian Jetter, and Harald Reiterer. Navi: a proof-of-concept of a mobile navigational aid for visually impaired based on the microsoft kinect. In *Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction - Volume Part IV, INTERACT'11*, pages 584–587, Berlin, Heidelberg, 2011. Springer-Verlag.

ERAS – An Ontology-Based Tool for the Expeditious Reconstruction of Virtual Cultural Heritage Sites

Telmo Adão, Luís Magalhães,
Maximino Bessa
ECT-UTAD / INESC Porto
Vila Real / Porto
telmo.adao@gmail.com,
{lmagalha,maxbessa}@utad.pt

António Coelho, Augusto Sousa
DEI - FEUP / INESC Porto
Porto
{acoelho,augusto.sousa}
@fe.up.pt

Nuno Rodrigues,
Alexandrino Gonçalves
ESTG / CIIC / Polytechnic
Institute of Leiria
Leiria
{nunorod,alex}@ipleiria.pt

Roberto Rodrigues
DEI - FEUP
Porto
rmfmrodrigues@gmail.com

Francisco Pereira
ECT-UTAD
Vila Real
fsp@utad.pt

João Moura
ECT-UTAD / Engineering
and Decision Support
Research Group
Vila Real / Porto
jpmoura@utad.pt

Luís Paulo Reis
University of Minho /
LIACC
Guimarães / Porto
lpreis@fe.up.pt

Resumo

Nos últimos anos, vários investigadores têm proposto métodos semiautomáticos para a geração expedita de modelos virtuais de edifícios e até cidades, em alguns casos direcionados à reconstrução do património cultural. No entanto, a maioria das abordagens propostas foca a geração de exteriores. Apesar da investigação realizada em torno da geração de interiores não ser inexistente, denota-se ainda uma falha de regras reais de suporte à geração dos referidos interiores. Para além disso, a especificação das regras de geração envolve extensas análises a documentos como livros, documentos históricos, plantas e outras representações gráficas, bem como o conhecimento de linguagens de gramática. Pretende-se seguir estas linhas de investigação a fim de obter uma ferramenta de modelação expedita que produza modelos de alta-fidelidade, empregando modelação procedimental. Este artigo tem como objetivo a apresentação da arquitetura geral de um sistema para a reconstrução virtual de sítios arqueológicos, incluindo exteriores e interiores de edifícios, aplicando um novo método de extração de informação a partir de dados geográficos e descrições textuais. Também se pretende apresentar a ontologia abstrata que regula a extração de dados e que define o esquema semântico de relação entre os diferentes elementos que integram uma cidade genérica, de forma a permitir a derivação por parte de outros estilos arquitetónicos.

Abstract

In the last years several researchers have proposed semi-automatic methods for the expeditious modelling of virtual buildings or even entire cities, in some cases applied to the cultural heritage reconstruction. However, most of the approaches focus the generation of the building exteriors. Despite the existing research work on interiors generation, there is still a lack of support from real rules. Besides, the specification of generation rules involves an extensive analysis of documents such as books, historical documents, floor plans and other graphical representations, and knowledge about grammar languages as well. We intend to follow these lines of research in order to get an expeditious modelling tool that will produce high fidelity models, employing procedural modelling techniques. This paper aims to present the global architecture of a system for the reconstruction of virtual heritage sites, applying a new method for extracting information from geographical data and textual descriptions. Moreover, it is intended to present an abstract ontology to rule the information extraction and defines a semantic relationship between the different elements that compounds a generic city in order to allow derivation by other architectonic styles.

Keywords

Grammars and Other Rewriting Systems, Computational Geometry and Object Modeling, Ontology, Rule-based generation model, Procedural Modelling, Three-Dimensional Graphics and Realism.

1. INTRODUCTION

Accurate 3D reconstruction and realistic visualization of both archaeological sites and ancient monuments allow experts to fine-tune their theories about the lost links in the history of civilization. Although the 3D reconstruction is a major challenge, precisely because of the lack of visual information, it constitutes a crucial task for experts who are interested in studying and interacting with long disappeared settlements and structures. Furthermore, the public in general will be provided with the tools to explore these archaeological sites within virtual environments, thus fostering cultural, social and scientific participation.

To achieve these reconstructions, the textual descriptions and geospatial data collected by archaeologists on site may be used to overcome the absence of visual information. Still, this data will not suffice, in which case procedural modelling turns out to be essential in order to avoid a great deal of time and labour consuming modelling processes.

Several researchers have proposed semi-automatic methods, using procedural modelling, to generate virtual models of buildings or even entire cities. Generally these approaches focus only on buildings exteriors or on buildings interiors generation. In this work the centre of attention, of the proposed methodology, goes to complete buildings, with interior and exterior, and to the heritage architectural rules by which they were built to generate complete 3D traversable models.

Far from being an easy process, the specification of these rules does involve an extensive analysis of the information contained in different sources (documents, ancient books, drawings, floor plans, etc.) and knowledge about grammar languages. Therefore, we want to contribute to the automation of some stages of this process, by developing methods to automatically extract and convert the information found on textual descriptions into an unified data model.

In this paper it is presented the architecture proposed for ERAS, an expeditious modelling tool for the reconstruction of archaeological sites with traversable buildings. The architecture section will expose the main constituent processes: the information extraction process and the procedural modeller process. In the other hand, it will be presented a first approach of the semantic schema that will provide architectural knowledge to ERAS: the ontology. Based on it, this tool will be capable to extract informations from textual descriptions in order to mount a guiding data schema, endowed of architectonic awareness.

2. RELATED WORK

Most techniques for procedural modelling of buildings mainly concentrate their efforts on the generation of modern structures. In the past few years several methods have been presented (e.g. [Parish01, Greuter03, Wwsr03, Finkenzeller05, Martin05, Müller06b, Weber09]) which address different aspects for procedural modelling of ur-

ban environments. Most of them are discussed by Watson et al. in [Watson08] where several other aspects, advantages and practical applications of this promising area are also discussed. One of these applications is on the modelling of cultural heritage structures. A cited example is the modelling of ancient Roman Pompeii [Müller05] (Italy) and Mayan Xkipché [Müller06a] (Mexico). The XL3D modelling system [Coelho07] operates automatically based on a modelling specification and geospatial L-systems. The results obtained with the modeller have proved the potential to model virtual urban environments.

A common feature among these works is that the main effort is on the generation of the exterior of the structures. In most of the approaches the interior of the structures is not generated, thereby they are not traversable. In [Martin05], the author addresses this issue using graphs which represent the rooms of a house and the connections between them, to generate modern traversable houses. In [Hahn06], Hahn et al. also address interior generation, namely in real time, by randomly dividing rectangular floors, corresponding to building interiors, into rectangular rooms and hallways. Both these methods still prove to be unsuccessful if the goal coincides with the generation of structures which may represent real buildings. Rau Chaplin and co-workers [Rau-Chaplin96] applied shape grammars, usually used on exterior facades, to generate building divisions, which fit layout tiles from a library of room layouts. Each division is then fulfilled with furniture according to its functionality. Marson and Musse [Marson10] suggest a method for interiors generation based on squarified treemaps, which consists in a recursive subdivision of functional areas into rooms. The last step is the creation of the corridor to connect the rooms. Tutunel et al. [Tutunel10] developed a expansion-based floor plan generation technique. In this work the authors map the rooms into classes to define the adjacency between them. Next, the rooms are placed in the layout and expand until touching each other. Charman [Charman93] presents a technique that solves the generation of room layouts when the problem is stated as a planning issue. The author's planner works based on aligned-axis rectangles with variable position, orientation and dimension, which represent restrictions given by the user in order to express geometric constraints. Marrell and co-workers [Merrell10] also proposed a method to generate house interiors, using Bayesian networks, trained with real-world data. This networks provides floor plans stochastically, that are then transformed in 3D models including, doors, windows and roofs.

In [Rodrigues08a] the authors address this problem concerning modern houses, where a method for the automatic generation of traversable houses, using architectural legal rules and a L-system to generate the interior rooms, was presented. In [Rodrigues08b], the authors proposed a method for the automatic generation of heritage sites with particular focus on Roman civilization structures, where several rules were written based on the knowledge left by Vitruvius, mostly through the reading of the Portuguese

adaptation from M. Justino Maciel “Tratado de Arquitectura” [Maciel06]. This method extends the one already applied on modern houses.

As stated by Watson et al. [Watson08] the automated production of rules is one of the most challenging aspects on procedural modelling. Muller et al. in [Müller07] presented one initial work towards this goal where they devised algorithms to automatically derive shape grammar rules from single facade images that can be used on a procedural modelling technique. The automated generation of rules from textual descriptions, contained on documents or books, implies the processing of the written text and the extraction of the information relevant for the production of the rules.

The Gate platform - General Architecture for Text Engineering- and the NLTK Project - Natural Language Toolkit are some of the solutions developed which can help researchers and programmers to develop applications that need some kind of natural language processing. While NLTK presents itself as a set of libraries and programmes developed in Python [Garrette09], the Gate platform has an open-source infra-structure to develop components for the natural language processing [Bontcheva02].

The NooJ platform [Silberztein04] is a cooperative and free environment of linguistic development that includes a vast set of resources and modules in more than ten languages. With this platform it is possible to analyse the grammatical, syntactic or morphological structure of a text or group of texts, being able to process more than 100 file formats, including PDF and HTML.

In [Rodrigues10] the authors present a first attempt to generate buildings’ models from textual descriptions written in Portuguese language. The prototype developed uses Port4Nooj [Ribeiro08], a Nooj version for Portuguese language, to extract the relevant information from the textual description and build a cityGML [Kolbe05] representation of a simple 3D church’s model.

Liu and co-workers [Liu08] proposed an ontology-based method to guide the generation of Chinese cultural heritage buildings. In their system, the user can specify the rules of the building styles in a DTD format, and an urban map to be reconstructed. A style checker uses the rules and an ontology defining the semantics between the city elements to approve or disapprove the generated building models in a XML. The checking process aims to find a valid XML with a coherent city model to guide the virtual generation of the urban map submitted. Tutunel et al. [Tutunel11] developed a framework that also integrate ontology-based structures. This framework combines a set of procedural modelling techniques, which are used according to the phase of the generation. These techniques communicate with a semantic moderator to reach information about the buildings, using a semantic library with classes, attributes and constraints. This information is then combined by the moderator into a semantic model of the building, constituting an advisory base structure used to avoid conflicts, such as undesired buildings inter-

sections or wrong associations like applying a bathroom window into a bedroom.

Differently from the referred works, ERAS is an expeditious tool that will automatically extract the informations from textual descriptions and geographical data in order to produce an ontology-based data schema, which will be used to generate a consistent virtual model regulated by editable rules of production. The next sections will expose an architecture proposal for our system, explaining the processes to extract information and to generate the virtual models. Moreover, it is intended to present a new ontological data schema that will be integrated in the ERAS system to provide archaeological organization knowledge.

3. ERAS ARCHITECTURE

To reduce the amount of human intervention, this work aims, besides other objectives, to develop an expeditious 3D modelling system. Accordingly, our modelling system will be guided by heritage knowledge about the construction rules of heritage structures, encoded in a formal grammar by expert knowledge in historical architecture procedures. This system will also encompass the modelling of façades and interiors in a way that both modelling processes integrate seamlessly and complement missing knowledge of the structure.

The specification of the rules used on the procedural modelling process is not an easy process since it involves an extensive analysis of the information contained in different sources (documents, ancient books, drawings, photographs, floor plans, etc.) and knowledge about grammar languages. We intend to contribute for the automated generation of rules from textual descriptions contained on documents or books.

Therefore, we propose the development of a tool, ERAS, for archaeologists to produce enhanced virtual reconstructions of archaeological sites. The architecture proposed for this tool (Figure 1) is composed by two main processes as follow:

1. Information Extraction Process;
2. Procedure Modelling Process.

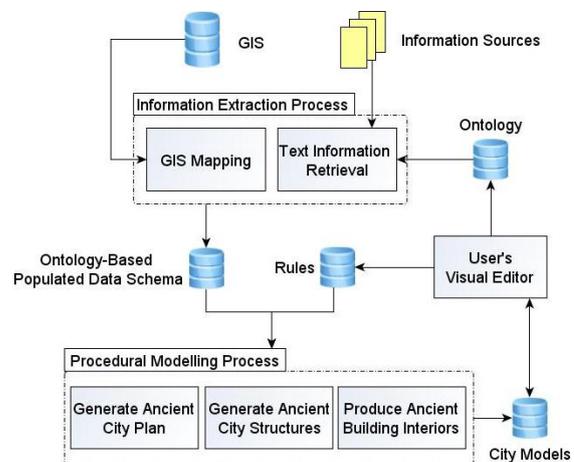


Figure 1. Architecture of ERAS, composed by two processes: the information extraction process that acquires the relevant

information relevant and determines the production guidelines and the procedural modelling process that constructs the 3D model of the ancient city based on architectonic editable rules.

The information extraction process will derive a structured data set from several information sources (e.g.: geospatial data, textual descriptions). Such information extraction process will be based on an Ontology created to this domain that will allow eliminating ambiguities and, alternatively, adding semantic information to our model. Thereby, this process results in a fulfilled data schema in which are set the city elements, including the buildings, and for each building is defined its divisional rooms.

The procedural modelling process is now ready to use the data schema to produce the virtual model. However, this process is regulated by a set of defined rules according to an architectonic style. The rules also have the crucial mission of avoiding inconsistency in generation, for instance, preventing the generation of buildings upon streets or putting windows on facades that are between adjacent houses. The rules, as the ontology, are editable thus a visual interface will let the user visualize and modify the rules to customize the generation constraints or guidelines, according to user's expectations or expert's point-of-view.

Taking rules in consideration, the procedural modelling process will be done in three stages. In the first one, it is generated a plan of the city in which is determined, at a very low level of detail, the placement of the basic components (e.g. streets, buildings, gardens, etc.). Then, the components are converted to tri-dimensional shapes with an increased detail (e.g. building mass models and street marks). At this point, the model does not contain the interior of the buildings, only an approximation to the exterior of the buildings. Hence, the final step is to generate the interiors where the exterior can act as a constraint, for all buildings in the model. The floor plans are generated with a defined adjacency for the divisions based on ontology and rules. Then, these divisions will be expanded until they fit the external contour. Finally, the buildings interiors are transformed into tri-dimensional representations with transitions between the rooms, such as windows and doors, were properly applied.

3.1 Information Extraction Process

We will define, in the section 4, an ancient urban ontology that best describes the semantic information of an archaeological environment. The ontology must deal with cultural and age difference, thus producing a general information model but that can encode different cultures and ages styles. This ontology constitutes a shared vocabulary that will allow modelling the type of objects and their properties and relations, including geospatial information. This vocabulary will be used by the information extraction module to encode structured data from the textual descriptions. In several documents one can find textual description of ancient structures and sometimes definitions of architectural styles. This textual information can be used to collect a set of guidelines which is a data schema containing the definitions of the existing elements

in the city. This data schema will guide the procedural modelling process. In a recent work [Rodrigues10] we presented a first approach for the expeditious building modelling from textual description. The prototype developed was used to automatically produce simple 3D models of monumental buildings based on a textual description written in Portuguese language. We intend to improve this methodology in order to generate more detailed 3D models of several types of structures, guided by the ontology defined and to extract information from generic descriptions, for instance, to define a specific building, with a position and a number of floors and room divisions. Other feature supported by this system is the capability of improve data models fidelity by obtaining the real positions of the elements present in the descriptions, whenever possible. This kind of information is provided by GIS (Geographic Information System) databases that retrieve a set of coordinates defining a street network or a building's base polygon. This information has to be converted by this process to an unified format used to map the virtual model, which was previously made by Coelho and co-workers[Coelho07].

The information extraction process will be responsible for translating the unstructured data presented on plain text to a more formal, ontology based, data that will be used for procedural modelling. This module will use the GATE ("General Architecture for Text Engineering"), which is bundled with a free Information Extraction system that will be configured to work for this domain and ontology. Moreover, the resulting data may contain geographical coordinates associated to the elements in the produced data schema, in order to provide location awareness.

The data schema produced by this module will feed the procedural modelling process while the rules will regulate it to ensure a reliable consistent model generation in order to avoid awkward situations such as buildings upon streets or even to amplify data to deal with the lacks of information.

3.2 Procedure Modelling Process

The second module is related to our recent work on expeditious modelling of urban [Coelho07] and archaeological [Rodrigues08b] environments. In the former it was defined a methodology for expeditious modelling of 3D virtual environments based on "Geospatial L-systems", a derivation of parametric L-systems [Prusinkiewicz90]. This methodology was mostly applied for the generation of urban environments (Figure 2). In the latter a framework for expeditious modelling of ancient structures was defined, based on a set of rules derived from constructions rules from the period of time in consideration and from expert knowledge. The buildings generated include the interiors and are traversable. This framework was applied to generate roman structures using Vitruvius' architectural rules (Figure 3). We intend to improve and merge these two works to develop an unified framework for the expeditious modelling of ancient cities with all their entities, where the buildings interiors are also generated depending on the active level of detail.



Figure 2. Urban environment procedurally generated [Coelho07]

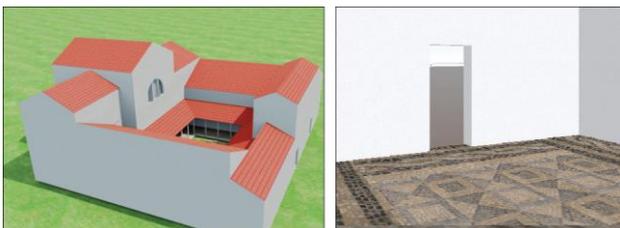


Figure 3. Rule-based generation of ancient roman houses; left: exterior appearance [Rodrigues11]; right: interior facade [Rodrigues08b].

The archaeological data collected on site will be stored in a geospatial database according to the defined ontology. This data will be amplified in order to generate the 3D virtual environment by using geospatial L-systems. This technology, already tested with good results in urban environments [Coelho07], uses a set of rules that encodes geospatial knowledge about the site such as the location, external polygon, type and height of each structure/building. The result of this first stage is a basic mapping of the entire archaeological site with a low level of detail, i.e., buildings, streets, gardens and other city elements will be, initially, represented by a two dimensional shape, creating a sort of a floor plan of the whole city. In a second stage, the two-dimensional shapes will be replaced by tri-dimensional shapes in order to improve detail and produce empty buildings mass models. At this point the city model should have the basic street definition and the buildings already assumed a 3D shape with a proposal for the outer facades. The last stage of the generation is reserved to the building interiors generation. The floor plans are generated for each produced building in the model and for each floor in all building. At this point, the floor plans have a two-dimensional representation and define the rooms of a given floor and the respective adjacencies. Next, the rooms are expanded until fit the exterior limits, fulfilling the inner space and respecting the defined adjacency. Then, it is marked the transition points that connect the division to each other and define the transitional passages between floors. Lastly, the tri-dimensional model of the interiors are generated, including the transitional points (windows, doors and staircases), regarding all the defined constraints. The whole process follows the guidelines previously extracted which are based on textual descriptions and ontology.

4. ERAS'S ONTOLOGY APPROACH

The ontology is the semantic structure that describes the relationships between the different elements on a system, as can be noticed in [Liu08]. This semantic definition represents by itself a guideline for the generation process by setting up a predefined structure in which every element should fit on. In order to define an ontology for ERAS system, the first approach was to develop a semantic structure that defines the set of abstract elements present on a abstract city. This ontology is based in the following principles:

- A city is composed by blocks and distribution networks, such as streets and water networks;
- The other components of the city are generated upon a block;
- Buildings can have gardens (private gardens) but gardens may also hold buildings (public gardens);
- A building is an abstract structure of a city that is constituted by one or more building parts;
- A building part is a vertical composition of a building and the horizontal compositions, floors, are seen as specializations of building parts;
- A building part is also a container of building divisions, such as main corridors and rooms, possibly containing other building parts;
- Building divisions have transitions or points of connection with the exterior of the building or passages for other divisions, such as windows or doors.

In Figure 4, it is shown the ontology used on ERAS according with the mentioned premises.

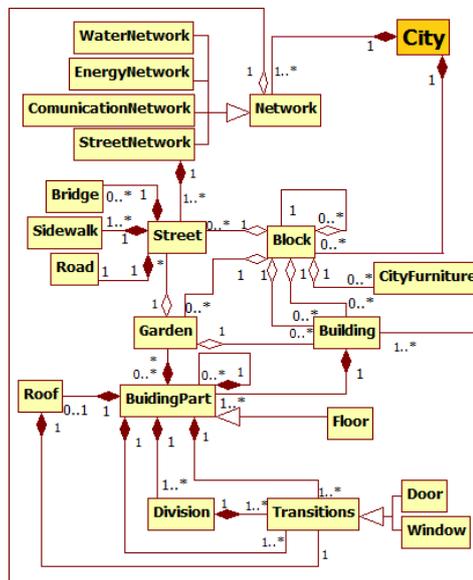


Figure 4. ERAS's abstract ontology, defining the relations between city elements.

Our ontology was planned to allow extensions to other architectural styles. The city elements of a particular style can inherit characteristics from the main elements present in the abstract ontology. For instance, a roman style is

constituted by its own building types that inherit from the building of our ontology (Figure 5). The mechanism will allow a reutilization of the elements and the expansibility of the ontology to other architectonic contexts.

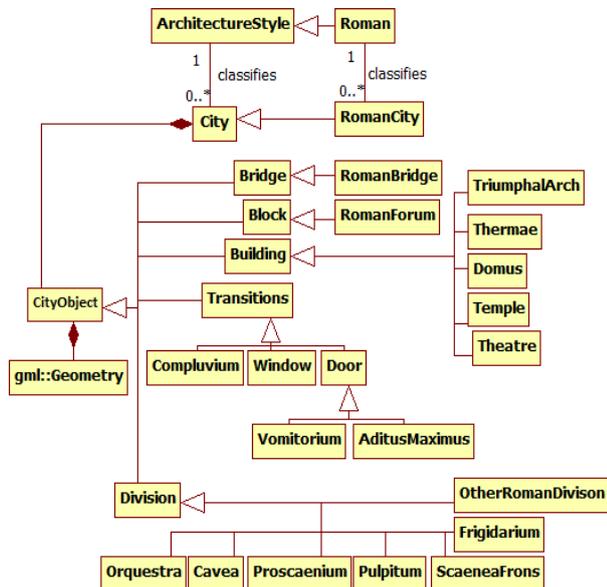


Figure 5. The inheritance of the roman style elements from the abstract ontology

5. CONCLUSION AND FUTURE WORK

In this paper we presented the system architecture of a tool that will enable archaeologists to produce a virtual reconstruction of an archaeological site in a short time period. The defined architecture exposes two main action processes: the extraction information process and the procedural modelling process. The first one proposes an innovative way of collecting and processing automatically the informations present in textual descriptions and GIS, based on our ontology. The output produced it is a structured data schema, with amplified data, that will be used by the procedural modeller, the second operating process. This generator applies the data schema given by the first process, and generates the 3D model of a city in a few steps. It is regulated by the rules that avoid inconsistencies in generation (e.g. generating building upon streets or intersecting each other) which can be edited by the user. Furthermore, it was presented the ontological data structure that guides the extraction of informations to form a coherent data schema. The designed ontology is abstract enough to be reused and expanded to a concrete architectonic style.

All the main technologies and methodologies needed to achieve the main goal of the project have been successfully tested in previous projects. Thus, we are confident that the hard work of integrating them in one single system capable of producing virtual reconstructions of archaeological sites will be successful.

In future work, we intend to develop each operating process described in ERAS architecture section based on our previous work. Our objective is to build a functional tool endowed of semantic capabilities to extract automatically

informations that will feed the procedural modeller in order to produce enhanced virtual representations. The ontology will be also refined and structured to be enabled for use by the extraction information process. It is also our intention to develop a graphical user interface to provide an interactive way to the user of editing the ontological data structure and the rules that will control and constraint the generation process. Finally, it is expected a successful development and integration of all the referred components to provide an innovative tool to aid archaeologists refining their theories, based on faithful and consistent virtual representations.

6. ACKNOWLEDGEMENTS

This work is partially supported by the Portuguese government, through the National Foundation for Science and Technology – FCT (Fundação para a Ciência e Tecnologia) and the European Union (COMPETE, FEDER) through the project PTDC/EIA-EIA/114868/2009 (FCOMP-01-0124-FEDER-015075) entitled “ERAS - Expeditious Reconstruction of Virtual Cultural Heritage Sites”.

7. REFERENCES

- [Bontcheva02] Bontcheva, K, Bunningham, H, Tablan, V, Maynard, D, Hamza, O. Using Gate As An Environment For Teaching Nlp. In Proceedings Of The Acl-02 Workshop On Effective Tools And Methodologies For Teaching Natural Language Processing And Computational Linguistics (2002), 54-62.
- [Charman93] Charman, P. (1993). *Solving space planning problems using constraint technology*. Paper presented at the NATO ASI Constraint Programming: Students Presentations.
- [Coelho07] Coelho, A, Bessa, M, Sousa, A, Ferreira, F. Expeditious Modeling Of Virtual Urban Environments With Geospatial L-Systems, *Computer Graphics Forum*, 26:4 (2007), 769-782.
- [Finkenzeller05] Finkenzeller, D, Bender, J, Schmitt, A. Feature Based Decomposition Of Façades. In Proceedings Of Virtual Concept (2005).
- [Garrette09] Garrette, D, Klein, E. An Extensible Toolkit For Computational Semantics, In Proceedings Of The Eighth International Conference On Computational Semantics (2009).
- [Greuter03] Greuter, S, Parker, J, Stewart, N, Leach, J. Undiscovered Worlds – Towards A Framework For Real-Time Procedural World Generation. In 5th International Digital Arts And Culture Conference (2003).
- [Hahn06] Hahn, E, Bose, P, Whitehead, A. Persistent Realtime Building Interior Generation. In Proceedings Of Acm Siggraph Symposium On Videogames (2006), 179–186.
- [Kolbe05] Kolbe, T, Groger, G, Plumer, L. Citygml - Interoperable Access To 3d City Models. In Proceedings Of The International Symposium On Geoinformation For Disaster Management (2005).

- [Liu08] Liu, Y, Xu, C, Zhang, Q E, Pan, Y. The Smart Architect: Scalable Ontology-Based Modeling Of Ancient Chinese Architectures. *Intelligent Systems, IEEE*, 23:1(2008), 49-56.
- [Maciel06] Maciel, M. *Vitrúvio – Tratado De Arquitetura*. Ist Press (2006).
- [Marson10] Marson, F, Musse, S R. Automatic Real-Time Generation Of Floor Plans Based On Squarified Treemaps Algorithm. *International Journal Of Computer Games Technology* (2010).
- [Martin05] Martin, J. *The Algorithmic Beauty Of Buildings: Methods For Procedural Building Generation* (2005). Honors Thesis, Trinity University.
- [Merrell10] Merrell, P., Schkufza, E., & Koltun, V. (2010). Computer-generated residential building layouts. *ACM Trans. Graph.*, 29(6), 1-12.
- [Müller05] Müller, P, Vereenoghe, T, Ulmer, A, Gool, L. Automatic Reconstruction Of Roman Housing Architecture. In *International Workshop On Recording, Modeling And Visualization Of Cultural Heritage* (2005), 287-297.
- [Müller06a] Müller, P, Vereenoghe, T, Wonka, P, Paap, I, Gool, L. Procedural 3d Reconstruction Of Puuc Buildings In Xkipche. In *Proceedings Of Eurographics Symposium On Virtual Reality, Archaeology And Cultural Heritage (Vast 06)* (2006a), 139–146.
- [Müller06b] Müller, P, Wonka, P, Haegler, S, Ulmer, A, Gool, L. Procedural Modeling Of Buildings. *Acm Transactions On Graphics (Tog)*, 25(3) (2006b), 614-623.
- [Müller07] Müller, P, Zeng, G, Wonka, P, Gool, L. Image-Based Procedural Modeling Of Facades. In *Proceedings Of Acm Siggraph 2007* (2007).
- [Parish01] Parish, H, Muller P. Procedural Modeling Of Cities. In *Proceedings Of Acm Siggraph 2001* (2001), 301-308.
- [Prusinkiewicz90] Prusinkiewicz, P, Lindenmayer, A. *The Algorithmic Beauty Of Plants* (1990). Springer-Verlag, New York, USA.
- [Rau-Chaplin96] Rau-Chaplin, A, Mackay-Lyons, B, Spierenburg, P. The Lahave house project: Towards an automated architectural design service, in *Proc. Int. Conf. Comput.-Aided Design* (1996), Hagenberg, Austria, 25–31.
- [Ribeiro08] Ribeiro A. Port4nooj: An Open Source, Ontology-Driven Portuguese Linguistic System With Applications In Machine Translation. In *Proceedings Of The 2008 International Nooj Conference (Nooj08)* (2008).
- [Rodrigues08a] Rodrigues, N, Dionísio, M, Gonçalves, A, Magalhães, L, Moura, J P, Chalmers, A. Rule-Based Generation Of Houses, *Computer Graphics & Geometry*, 10:2 (2008a).
- [Rodrigues08b] Rodrigues, N, Magalhães, L G, Moura, J P, Chalmers, A. Automatic Reconstruction Of Virtual Heritage Sites, In *The 9th International Symposium On Virtual Reality, Archaeology And Cultural Heritage* (2008b).
- [Rodrigues10] Rodrigues, R, Coelho, A, Reis, L P. k In *Proceedings Of Grapp 2010 - 5th International Conference In Computer Graphics Theory And Applications* (2010), 130-133.
- [Rodrigues11] Rodrigues, N, Magalhães, L, Moura, J, P, Chalmers, A, Santos, F, Morgado, L. Procedural Virtual Worlds. In *Virtual Worlds and Metaverse Platforms: New Communication and Identity Paradigms* (2011), 16 - 32.
- [Silberztein04] Silberztein, M. Nooj: A Cooperative, Object-Oriented Architecture For Nlp, Intex Pour La Linguistique Et Le Traitement Automatique Des Langues, *Cahiers De La Msh Ledoux* (2004).
- [Tutunel11] Tutunel, T, Smelik, R M, Lopes, R, De Kraker, K J, Bidarra, R. Generating Consistent Buildings: A Semantic Approach For Integrating Procedural Techniques. *Computational Intelligence And Ai In Games, IEEE Transactions On*, 3:3 (2011), 274-288.
- [Weber09] Weber, B, Muller, P, Wonka, P, Gross, M. Interactive Geometric Simulation Of 4d Cities. *Computer Graphics Forum*, 28:2 (2009), 481-492. [Watson08] Watson B, Müller, P, Veryovka, O, Fuller, A, Wonka, P, Sexton, C. Procedural Urban Modeling In Practice. *IEEE Computer Graphics And Applications*, 28:3 (2008), 18-26.

Artigos curtos

Adaptação de Cor para Dicromatas na Visualização de Imagens

Madalena Ribeiro

Escola Superior de Artes Aplicadas, Instituto Politécnico de Castelo Branco
Quinta da Sr.^a de Mércules, 6000-909 Castelo Branco
mribeiro@ipcb.pt

Abel Gomes

Departamento de Informática da Faculdade de Engenharia, Universidade da Beira Interior / Instituto de Telecomunicações
Rua Marquês d'Ávila e Bolama, 6201-001 Covilhã
agomes@di.ubi.pt

Resumo

O incremento de informação visual nos novos media veio conferir à cor um papel novo e preponderante. Entre outros, a cor é fundamental na destrição de conteúdos, possibilitando uma perceção integral das coisas. No entanto, algumas pessoas têm uma deficiência visual designada deficiente visão da cor, a qual limita a visão da cor de uma forma adequada, e que compromete a perceção correta dos conteúdos visuais.

De modo a ultrapassar este impedimento, neste artigo é proposto um algoritmo que pode ser usado para adaptar a cor para dicromatas, melhorando a perceção de imagens a cores. Consequentemente, as imagens tornam-se mais legíveis e é melhorado o discernimento dos respetivos conteúdos visuais.

Palavras-Chave

Daltonismo, dicromacia, protanopia, deuteranopia, deficiente visão da cor, adaptação de cor, mapeamento de cor, recoloração.

1. INTRODUÇÃO

A difusão de conteúdos visuais em ambientes computacionais veio conceder à cor uma preponderante posição como componente comunicacional. Além do seu inerente carácter de signo e da sua função no design, a cor é ainda uma componente fundamental na destrição dos conteúdos gráficos. No caso particular das imagens, sobre as quais recai a atenção no presente artigo, são as cores dos diversos elementos que constituem a imagem que possibilitam a distinção entre eles, permitindo perceber o que a imagem contém e entender a mensagem subjacente.

Embora a grande maioria da população seja portadora de uma normal visão da cor, a deficiente visão da cor (DVC), correntemente conhecida como daltonismo, atinge cerca de 5% da população. Sendo que esta insuficiência visual leva à redução da gama cromática visualizada, são várias as cores que, embora tendo matizes bastante distintos, sejam vistas de igual forma, levando à ambiguidade na interpretação de conteúdos nas imagens.

Neste contexto, neste artigo vem apresentar-se uma proposta de um algoritmo de recoloração para os casos mais frequentes de dicromacia (deuteranopia e protanopia, por esta ordem), mostrando os resultados obtidos na adaptação de cor em imagens.

2. VISÃO DA COR

Ao nível do olho humano, a visão da cor é possibilitada pela existência de células, designadas cones (L-cones, M-cones e S-cones), que existem ao nível da retina, sendo

que a cor percebida resulta da mistura da informação recolhida pelos 3 tipos de cones. Pelo facto da cor resultar da combinação de 3 canais distintos, a normal visão da cor designa-se tricromacia [Paterson'03]. Existem, no entanto, casos em que um (ou eventualmente mais que um) tipo de cones apresenta falha total ou parcial de sensibilidade, com consequente adulteração na perceção da cor. Quando um tipo de cones apresenta malfuncionamento, designa-se tricromacia anómala e quando um tipo de cones não funciona, de todo, designa-se dicromacia. Designa-se monocromacia (quando dois ou os três tipos de cones não funcionam, de todo) [Birch'01], [Case'03].

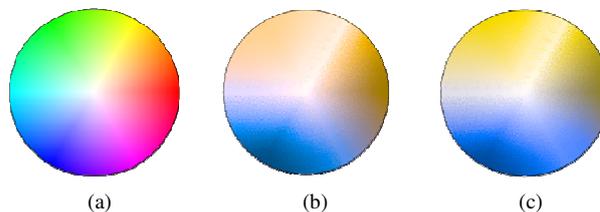


Figura 1: Círculo cromático¹ quando visto por uma pessoa com (a) normal visão da cor; (b) deuteranopia e (c) protanopia.

¹ A imagem em (a) foi gerada usando Processing 1.5.1 e as imagens em (c) e (d) foram obtidas a partir da imagem (a), por via de simulação cujo algoritmo vem referido em [Vienot et al.'99].

No caso do algoritmo proposto neste artigo, são considerados os casos de dicromacia em que os cones afetados são os L-cones (protanopia) ou os M-cones (deuteranopia), pelo facto de constituírem os casos com maior taxa de incidência para esse tipo de DVC e ainda pelo facto de sendo um nível de severidade média, representarem uma limitação na perceção muito mais incapacitante que a tricromacia anómala (que pode até ser ligeira).

3. ALGORITMOS DE RECOLORAÇÃO

Desde há cerca de uma década e meia, têm vindo a surgir tentativas de promover a melhorias da forma como os daltónicos veem as cores, particularmente algoritmos de recoloração para serem aplicados em imagens estáticas. A grande maioria dos algoritmos propostos na literatura tem por alvo a dicromacia (por ser mais incapacitante que a tricromacia anómala), especialmente a deuteranopia e protanopia, pela sua taxa de incidência.

Os algoritmos usam estratégias distintas e, em função disso, é adotado o espaço de cor que melhor se adequa ao processo de ajustamento da cor. Uma vez que as imagens se encontram armazenadas sob o formato RGB, na maioria dos casos, a cor é convertida de RGB para o espaço de cor, no qual a cor é alterada/adaptada e, depois reconverte novamente para o formato de cor RGB. São vários os espaços de cor usados e, nos quais é feita a adaptação da cor. Outros casos existem em que são usados os parâmetros de sensibilidade dos cones à luz, os seja, os valores LMS, nomeadamente em [Yang et al.'04], [Ma et al.'06], [Jefferson et al.'06] e [Jefferson et al.'07], assim como em [Lee et al.'11].

À semelhança do algoritmo apresentado neste artigo, em [Yang et al.'03], [Iaccarino et al.'06] e em [Huang'08] são usados espaços de cor (respetivamente, os espaço HSI, HSL e HSV) similares ao HSV, a seguir apresentado.

4. ALGORITMO PROPOSTO

O algoritmo proposto leva em conta a forma como os deuteranopes e tritanopes vêm as cores (que é deveras semelhante): apenas conseguem ver dois tons vivos, os amarelos e os azuis. Todas as outras cores são vistos como pardas. Por esse motivo, o objetivo principal do algoritmo é o de transformar os matizes vivos na área dos rosas, vermelhos e laranjas (que são confundidos pelos daltónicos com verdes e azuis descolorados), em outros matizes igualmente vivos (considerando a forma como estes dicromatas os veem).

4.1 Representação da Cor

O HSV (representado geometricamente por um cone invertido, conforme ilustra a Figura 2) foi o espaço de cor escolhido para representar a cor, dado o seu carácter intuitivo na forma de variação das cores em função da variação dos parâmetros, respetivamente *hue* (matiz), *saturation* (saturação) e *value* (valor/luz) [Smith'78], [Foley'91].

As cores são representadas por um triplo, i.e., $C = (H, S, V)$, onde $0 \leq H \leq 360$ e $0 \leq S, V \leq 1$. **H** representa o matiz, e permite distinguir o verde do amarelo e do azul (como ilustra a figura 2). A saturação **S** relaciona-se com a pure-

za versus adição de cinza e o **V** (valor) corresponde à luminosidade da cor.

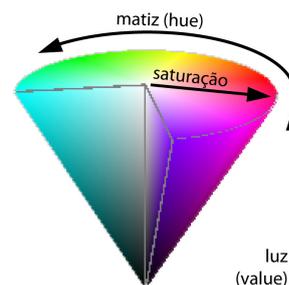


Figura 2: Representação geométrica² do espaço de cor HSV (hue, saturation, value).

4.2 Alteração da Cor

Com base na especificação da cor no espaço de cor HSV, $C = (H, S, V)$, sendo que a alteração dessa cor leva à obtenção da cor $C' = (H', S', V')$, onde:

$$H' = H + \Delta H \quad S' = S + \Delta S \quad V' = V + \Delta V \quad (1)$$

Neste contexto, o problema foca-se na forma como os fatores ΔH , ΔS e ΔV são encontrados, de forma a que a nova cor seja mais conveniente quando observada por um detentor de DVC.

4.3 Estratégia do Algoritmo

A estratégia adotada para a recoloração faz um tratamento distinto entre as cores muito escuras e as restantes. As primeiras são ainda mais escurecidas, através do decremento do parâmetro **V** (value), para metade do seu valor original. Para as outras cores, o procedimento é o seguinte:

- As cores cujo matiz se encontra na zona dos amarelos alaranjados, amarelos e verdes, permanecem inalteradas;
- Às cores cujo matiz se encontra na zona dos azuis é-lhes incrementada a saturação (para o dobro do seu valor original);
- Os matizes na zona dos magenta-vermelho-laranja são mapeadas em zuis e violetas, sendo-lhes ainda reduzida a saturação (para metade do seu valor original)

O mapeamento das cores vivas que se encontram na zona dos magentas, vermelhos e laranjas em azuis prende-se com o facto de os amarelos e os azuis serem as únicas cores vívidas que podem ser discernidas pelos deuteranopes e tritanopes. Sendo que os amarelos têm uma área de diversidade cromática mais reduzida (como pode ser visto pela figura 3), considerou-se o domínio de azuis como uma área destinatária mais conveniente no mapeamento.

No entanto, importa garantir a destrição entre as cores que são originalmente azuis e as que são mapeadas para azul. A garantia dessa distinção é proporcionada pela alteração da saturação: a saturação é incrementada para as cores originalmente azuis e decrementada para as cores

² Imagem gerada usando Processing 1.5.1.

mapeadas em azuis. Este esquema evita, quase totalmente, a sobreposição de cores.

4.4 Mapeamento dos Matizes da Área Magenta-Vermelho-Laranja

O mapeamento em azuis, das cores da zona magenta-vermelho-laranja é efetuado através de uma simetria de matizes, criada na linha dos 300º, onde se encontra o magenta.

Por meio desta simetria, à medida que os matizes se afastam do magenta (no sentido contrário aos ponteiros do relógio), são mapeados em violetas e azuis, cada vez mais distantes do magenta (mas no sentido dos ponteiros do relógio).

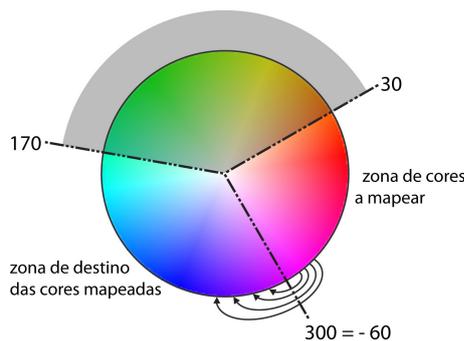


Figura 3: Mapeamento de matizes da zona magenta-vermelho-laranja em matizes da zona ciano-azul-violeta.

Sendo que a área de origem (magenta-vermelho-laranja) é menor que a área de destino (violeta-azul-ciano), é usado o teorema de Thales (eq. (2)) para possibilitar uma distribuição ajustada e obter uma correspondência uniforme.

$$\frac{\alpha - \alpha_{\min}}{\alpha_{\max} - \alpha_{\min}} = \frac{\beta - \beta_{\min}}{\beta_{\max} - \beta_{\min}} \quad (2)$$

α é um valor pertencente ao intervalo $[\alpha_{\min}, \alpha_{\max}]$ e β é o valor correspondente no intervalo $[\beta_{\min}, \beta_{\max}]$.

Aplicando a eq. (1) e considerando o domínio de matizes a transformar $[h_{D1_min}, h_{D1_max}]$ e o domínio de matizes de destino $[h_{D2_min}, h_{D2_max}]$, obtém-se a equação que possibilita o mapeamento do matiz original no matiz adaptado (respectivamente h_{orig} e h_{final} , na eq. (3)).

$$h_{final} = \frac{(h_{orig} - h_{D1_min})(h_{D2_max} - h_{D2_min})}{h_{D1_max} - h_{D1_min}} + h_{D2_min} \quad (3)$$

Mais especificamente, $[-60, 30]$ é o domínio de matizes a transformar e $[300, 170]$ é o conjunto de destino desses matizes, conforme ilustrado na figura 3.

Note-se que o domínio de destino do mapeamento é maior do que o domínio de origem. Este facto é vantajoso, na medida em que é possível dispersar os matizes mapeados pela área de destino, evitando a sua concentração, o que ajuda à sua destriça (como se pretende).

Além disso, o valor de início do domínio de destino é mais elevado que o valor de final, o que possibilita a

simetria (relativamente ao ponto de 300º). Assim, os matizes ligeiramente superiores a 300 são mapeados em matizes ligeiramente inferiores a 300, e à medida que os matizes se tornam gradualmente superiores a 300, são mapeados em matizes proporcionalmente inferiores a 300.

5. EXPERIÊNCIAS

O algoritmo de recoloração foi implementado com recurso ao HTML5 (e javascript), na medida em que o objetivo final é a aplicação do método na recoloração de páginas web, na perspetiva de melhor a acessibilidade de pessoas com DVC, fomentando a inclusão.

O novo elemento *canvas*, facultado pelo HTML5, permite alterar a imagem, pixel a pixel, à semelhança de outros sistemas gráficos existentes (e.g., o Processing ou OpenGL).

6. RESULTADOS

Quando um deuteranope ou um protanope vê a imagem original, não vê as flores rosas, lilases e vermelho corretamente, sendo vistos como cores pardas, por vezes, com a mesma tonalidade que a verdura de cor verde. No caso do vermelho intermédio, as flores são completamente confundidas com o verde (ver canto inferior esquerdo da figura 4).

Após a recoloração, as flores que são originalmente azuis assumem um tom forte conferido pela saturação, que se torna mais alta. Em contrapartida, as flores cujas cores foram mapeadas em azul, têm uma tonalidade algo moderada, controlada pela redução na saturação. O algoritmo possibilita a coerência percetual após a recoloração, para duas cores com o mesmo matiz original e com luminosidade diferente (e.g. vermelho claro e escuro).

Após a recoloração da imagem, a legibilidade aumentou, havendo uma clara destriça entre a verdura e as flores.

7. CONCLUSÃO

A deficiente visão da cor apresentada por deuteranopes e protanopes, reduz a sua visão da cor a apenas dois matizes de cor, o azul e o amarelo, sendo que todos os outros matizes vistos pelas pessoas com normal visão da cor se reduzem a um verde ou azul pardo. Assim, na adaptação que foi realizada foi considerada a impossibilidade de visualização de determinados tons e considerando o domínio cromático que as pessoas detentoras da condição de DVC conseguem ver.

Sendo que a maior parte dos verdes que existem se relacionam com a natureza, a forma descolorada como são vistos não interfere com a perceção. Por esse motivo, poderão permanecer inalterados. As cores vibrantes na área do magenta, laranja e são mapeadas em cianos/azuis e violetas, tendencialmente vibrantes, possibilitando que assim sejam percebidas.

No global, após a adaptação de cor, a legibilidade das imagens aumenta, promovendo uma perceção mais próxima do real, para os deuteranopes e para os protanopes, melhorando a sua condição como observador.

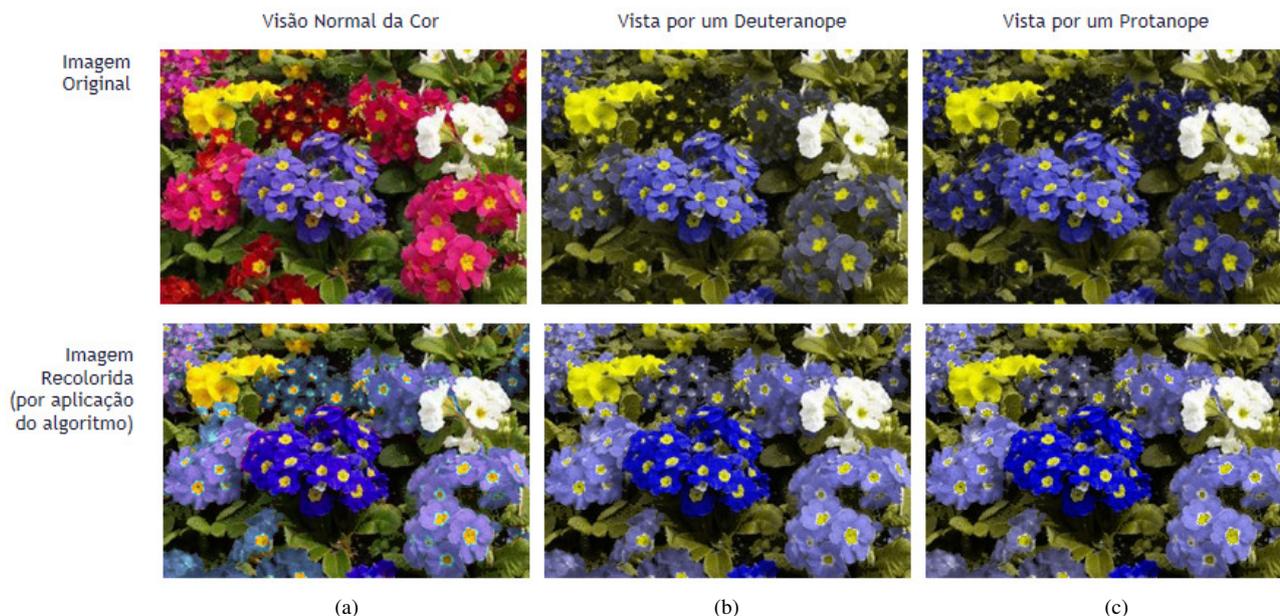


Figura 4: Na 1ª linha vê-se a imagem original quando vista por uma pessoa com (a) visão normal³; (b) deuteranopia e (c) protanopia. Na 2ª linha vê-se a imagem original recolorida pelo algoritmo proposto, quando vista por uma pessoa com (a) visão normal; (b) deuteranopia e (c) protanopia. A simulação foi implementada com recurso a [Vienot et al.'99].

³ Tirada de <http://www.flickr.com/photos/>, com cortesia de [Dave - aka Emptybelly](#) (licença da imagem: CC Attribution-NonCommercial 2.0).

A partir de <http://mribeiro.ipcb.pt> podem ser encontrados muitos outros exemplos de aplicação do algoritmo de recoloração.

8. REFERÊNCIAS

- [Birch'01] Birch, Jennifer. *Diagnosis of Defective Colour Vision*. Second edition. Edimburgh: Elsevier Science, 2001.
- [Case'03] Case, B. *Color Blindness*. 2003.
- [Foley'91] Foley, James, Van Dam, Andries, Feiner, Steven, Hughes, John. *Computer Graphics, Principles and Practice*. second edition. 1991.
- [Huang'08] ECCV Workshop on Computer Vision Applications for the Visually Impaired, 2008, Enhancing Color Representation for the Color Vision Impaired. City 2008.
- [Iaccarino et al.'06] Iaccarino, Gennaro [et al.]. Efficient edge-services for colorblind users: Proceedings of the 15th international conference on World Wide Web. Edinburgh, Scotland: ACM, 2006.
- [Jefferson et al.'06] Jefferson, Luke, Harvey, Richard. Accommodating color blind computer users: Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility. Portland, Oregon, USA: ACM, 2006.
- [Jefferson et al.'07] Jefferson, Luke, Harvey, Richard. An interface to support color blind computer users: Proceedings of the SIGCHI conference on Human factors in computing systems. San Jose, California, USA: ACM, 2007.
- [Lee et al.'11] Lee, Jinmi Lee, Santos, Wellington Pinheiro Dos. An adaptive fuzzy-based system to simulate, quantify and compensate color blindness. *Integrated Computer-Aided Engineering*. Vol. 18, n.º 1 (2011), 29-40.
- [Ma et al.'06] Ma, Y., Gu, X. D., Wang, Y. Y. - A new color blindness cure model based on BP neural network. In: Wang, J. [et al.]. *Advances in Neural Networks - Isnn 2006, Pt 3, Proceedings*. Berlin: Springer-Verlag Berlin, 2006. 740-745.
- [Paterson'03] Paterson, Ian. *A dictionary of Colour*. London: Thorogood, 2003.
- [Smith'78] Smith, Alvy Ray. *Color Gamut Transform Pairs: SIGGRAPH 78*. Atlanta, USA, 1978.
- [Vienot et al.'99] Vienot, F., Brettel, H., Mollon, J. D. Digital video colourmaps for checking the legibility of displays by dichromats. *Color Research and Application*. Vol. 24, n.º 4 (1999), 243-252.
- [Yang et al.'03] Image Processing, 2003. ICIIP 2003. Proceedings. 2003 International Conference on, 2003, Visual Contents Adaptation for Color Vision Deficiency. City 2003. 453-456.
- [Yang et al.'04] Yang, Seungji [et al.]. Improving visual accessibility for color vision deficiency based on MPEG-21. *Etri Journal*. Vol. 26, n.º 3 (2004), 195-202.

Inserção Dinâmica de Objectos Virtuais no Contexto de Fotografias Tiradas por Utilizadores

Rui Nóbrega Nuno Correia

CITI, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa
2829-516 Caparica, Portugal

rui.nobrega@campus.fct.unl.pt, nmc@fct.unl.pt

Resumo

A introdução de objectos virtuais em ambientes de Realidade Aumentada em tempo real requer normalmente a configuração de marcas pré-definidas e a sua posterior detecção no cenário. Neste artigo é proposto um sistema semi-automático de análise de estrutura tri-dimensional em fotografias, com o objectivo de introduzir interactivamente objectos virtuais no cenário. Estes objectos tri-dimensionais são representados com a perspectiva correcta e reagem a elementos da fotografia como se fossem entidades no mundo real. Utilizando estes elementos virtuais é possível criar aplicações de realidade aumentada em que o utilizador tira uma ou mais fotos de um determinado local e introduz interactivamente objectos ou elementos virtuais que se misturam com a foto em tempo real. Neste artigo são discutidas novas formas de adquirir imagens e informação sobre o cenário envolvendo o utilizador e também como esses dados podem ser processados para uma correcta visualização final do ambiente de realidade mista.

Palavras-Chave

Realidade Aumentada, Visão por Computador, Computação Gráfica, Multimédia

1. INTRODUÇÃO

A maior parte das aplicações de Realidade Aumentada (RA) utilizam imagens do mundo real como base mas com limitações e constrangimentos por forma a que os elementos e grafismos virtuais sejam integrados correctamente no cenário. Muitas usam cenários fixos com modelos pré-construídos à partida, enquanto outras utilizam referências no cenário como por exemplo marcas. Este artigo propõe um sistema para construir aplicações de RA em qualquer espaço sem um conhecimento *a priori* do mesmo. Mais concretamente, pretende-se criar ambientes de RA em espaços escolhidos pelo utilizador oferecendo a possibilidade de interacção com esse espaço de forma virtual mas tendo em conta propriedades do mundo físico.

Por detrás deste projecto está a motivação de construir aplicações interactivas que permitam ao utilizador refazer, redefinir e alterar virtualmente um espaço pessoal. Possíveis implementações deste sistema seriam importantes para aplicações de decoração de quartos, reorganização de interior e mobiliário, jogos e sistemas de modelação e animação 3D.

Tendo em conta as motivações descritas o principal problema considerado neste artigo será como adquirir virtualmente um espaço designado pelo utilizador e usar o modelo detectado para adicionar informação virtual tri-dimensional de forma geometricamente correcta. Adicio-

nalmente deverá ser possível interagir com o modelo final em tempo real, respeitando as propriedades da cena representada na imagem. Mais do que construir um sistema de reconstrução 3D foto-realisticamente perfeito, pretende-se a construção de um modelo suficientemente robusto construído interactivamente por um utilizador não especializado.

A solução proposta sugere um modelo de interacção com o utilizador para adquirir uma imagem do cenário. Seguidamente são utilizados vários algoritmos para detectar elementos visuais nessa imagem. Por fim, uma aplicação de Realidade Aumentada é proposta.

2. TRABALHO RELACIONADO

Existem vários tipos de aplicações de RA, sendo que os sistemas mais simples utilizam marcas físicas que servem de referência para a colocação de elementos virtuais em videos e imagens. Um destes sistemas é o ARToolkitPlus¹ que utiliza marcas quadradas. Outros sistemas utilizam um cenário pré-definido e fixo como é o caso do jogo Tennis Real Play [Lai11]. A utilização de câmaras com sensores de profundidade pode facilitar a recuperação da estrutura da imagem como é exemplo o projecto KinectFusion [Izadi11]. Alguns projectos como o PTAM [Klein07] dependem de uma inicialização com um pequeno video em translacção do cenário. Depois de processado o video,

¹ARToolkitPlus, <http://handheldar.icg.tugraz.at/artoolkitplus.php>

é possível gerar uma superfície virtual onde a aplicação de RA decorre. A análise de uma imagem por forma a compreender a sua estrutura foi efectuada em vários projectos. No Make3D [Saxena09] a imagem é analisada com algoritmos de segmentação para gerar um cenário tri-dimensional da figura. Em [Lee09] é apresentado um sistema que analisa linhas da imagem para reconhecer o espaço virtual da cena. Por fim, [Karsch11] apresenta uma aplicação semi-automática de reconhecimento 3D da cena com introdução virtual de objectos. Os objectos possuem texturas e iluminação baseadas na fotografia de entrada.

3. SISTEMA DE REALIDADE AUMENTADA

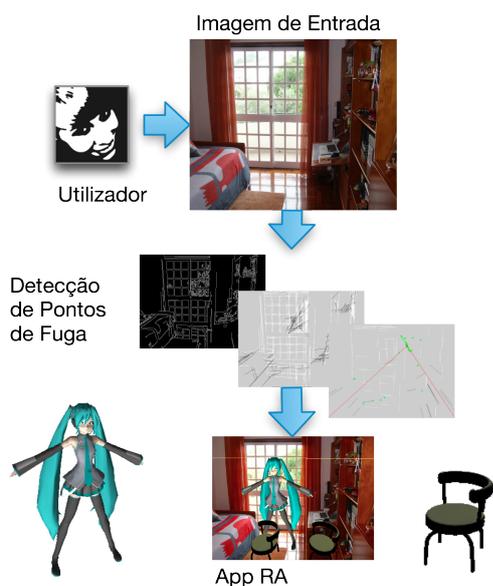


Figura 1: Sistema de RA, o utilizador tira uma fotografia que é analisada por forma a ser construído um modelo interno do mundo real. Os objectos virtuais são posteriormente introduzidos com a perspectiva correcta.

O sistema de Realidade Aumentada proposto tem como objectivo a criação de aplicações que tenham por base o espaço físico do utilizador. O objectivo principal é utilizar o espaço físico do utilizador sem usar marcas no mundo real e utilizando apenas sensores disponíveis em vários equipamentos. Para tal deverá ser criado um modelo interno do mundo real por forma a que a aplicação de RA possa ser executada sob as imagens capturadas. Os principais sensores utilizados serão câmaras e acelerómetros. As câmaras utilizadas terão no mínimo as características presentes em qualquer câmara disponível actualmente em qualquer telefone móvel ou computador portátil. Não incluídas nestes sensores estão as câmaras com sensores de profundidade.

Tendo em conta as ideias por detrás deste projecto realizou-se uma análise dos principais requisitos do sistema proposto. A aquisição do cenário da aplicação de

Realidade Aumentada deverá poder ser feita por um utilizador que não tenha grande experiência em fotografia ou manipulação 3D. O utilizador deverá conseguir iniciar o sistema com um mínimo de instruções básicas e pouco ou nenhum treino. O sistema deve criar o modelo espacial subjacente à aplicação que guarde todos os elementos necessários à introdução de elementos virtuais. A criação do modelo deverá ocorrer quase em tempo real. A interacção com a aplicação que use o modelo deverá ser natural, os elementos virtuais deverão revelar propriedades próprias do mundo físico (Ex: gravidade, colisões com chão).

O conceito de interacção proposto pode ser dividido em três momentos: (1) Aquisição, (2) Processamento de Informação e (3) Aplicação Interactiva. Numa primeira parte (1) temos a aquisição de imagens e informação sobre o cenário onde queremos implantar a aplicação de RA. Aqui a interface do sistema desempenha um papel fundamental pois deverá instruir o utilizador a tirar fotos com algum significado e úteis para o sistema. Adicionalmente, o utilizador poderá ser chamado a localizar certos elementos visuais como o chão ou o tecto por forma a auxiliar os algoritmos de detecção. Estes algoritmos de processamento de imagem e de visão por computador entram em cena numa segunda fase (2). Pegando na informação adquirida o objectivo é detectar visualmente elementos como chão, horizonte, pontos de fuga, orientação da cena e elementos tridimensionais presentes. Utilizando todas as propriedades que for possível detectar poderão ser criadas várias aplicações interactivas (3) que criam a possibilidade de interagir com elementos virtuais num espaço real escolhido pelo utilizador.

Um exemplo do organigrama de interacção proposto pode ser observado na figura 1. Estas três fases serão revisitadas em maior detalhe nas próximas secções onde será apresentado o protótipo que implementa este sistema.

3.1. Aquisição de Informação

A aquisição de imagens e da informação sobre o espaço onde a aplicação de Realidade Aumentada vai decorrer é um dos passos fundamentais na construção do sistema. Como já foi referido, a correcta detecção do modelo tridimensional do cenário depende de algoritmos de análise de imagem. Por este motivo é necessário que a fotografia tirada pelo utilizador tenha conteúdo suficiente para ser analisada. Isto exclui fotografias de paredes em branco, muito perto ou desfocadas. Idealmente as fotografias deverão conter uma grande quantidade de linhas rectas com bastantes superfícies lisas e texturas difusas tipo Lambert. Tipicamente este tipo de fotografias pode ser encontrado em edificações urbanas e em interiores de casas ontem tipicamente as paredes, o chão, o tecto e a mobília contêm bastantes ângulos rectos. Para além de uma boa fotografia é necessário detectar a que altura do chão a imagem foi tirada. Outro problema com as câmaras é que possuem diferentes parâmetros como ângulo de visão, profundidade de campo e distorção radial de imagem. Adicionalmente é necessário ter em conta a escala do cenário e que tamanho terão que ter os objectos virtuais.

O utilizador pode inserir uma foto tirada com uma câmara fotográfica ou capturar directamente a imagem usando uma webcam. Se usada a webcam o sistema pode usar um sistema de auxílio ao utilizador que avisa quando é que a imagem tem detalhe suficiente para ser capturada. Este sistema auxiliar é baseado no número de linhas rectas detectadas (através da transformação de Hough) e através do número de pontos SURF detectados (ver implementação OpenCV²). Para obter mais dados sobre o sistema poderão ser sugeridas várias perguntas por forma a obter informações simples mas úteis sobre o espaço capturado. A perguntas ou tarefas sugeridas são:

–”Desenhe uma linha à volta do chão.”– O utilizador desenha um esboço de uma linha à volta da zona que considera o chão da cena. A localização do chão é depois efectuada com exactidão utilizando um algoritmo de segmentação como o *Graph Cuts*.

–”Desenhe uma linha à volta de objectos que estão em cima ou à frente do chão.”– Similar à anterior tarefa o objectivo aqui é inferir que tipo de objectos terão de ser representados em frente aos objectos virtuais ocultando os mesmos.

–”A que altura foi retirada a fotografia? (1.70 m por omissão)”– Esta pergunta ajudará a inferir onde deverá ser colocado o plano virtual que representa o chão.

–”O horizonte foi correctamente detectado?”(dar duas ou três hipóteses) – O sistema de análise (explicado à frente) apresenta normalmente vários resultados com diferentes graus de probabilidade. Por vezes, nem sempre o horizonte com maior grau de probabilidade é o mais correcto, podendo assim o utilizador escolher outra possibilidade.

–”Desenhe um linha recta junto a uma aresta vertical de uma porta ou janela.”–Sabendo a altura média de uma porta ou de uma janela, poderá ser inferida alguma informação sobre a escala que deverão ter os objectos inseridos. Permite também inferir com maior correcção qual a direcção vertical da cena.

Toda a informação recolhida será incorporada no modelo do cenário da aplicação de RA. Na próxima secção é explicada sucintamente como é processada a imagem de entrada.

3.2. Processamento de Imagem

O protótipo actual analisa a imagem de entrada em busca de várias propriedades visuais como linhas principais, pontos de fuga principais e superfícies. A Figura 1 apresenta alguns dos passos intermédios necessários à detecção dos vários elementos. Todo o processo realizado é descrito em detalhe em [blind review]. De uma forma resumida o algoritmo analisa a imagem em três fases. Cada fase é repetida várias vezes com parâmetros diferentes para apanhar um máximo de situações possíveis (diferentes resoluções, diferentes iluminações, níveis diferentes de lixo visual).

Na Figura 2 podemos observar um esquema simplificado das várias estruturas de dados que suportam o sistema.

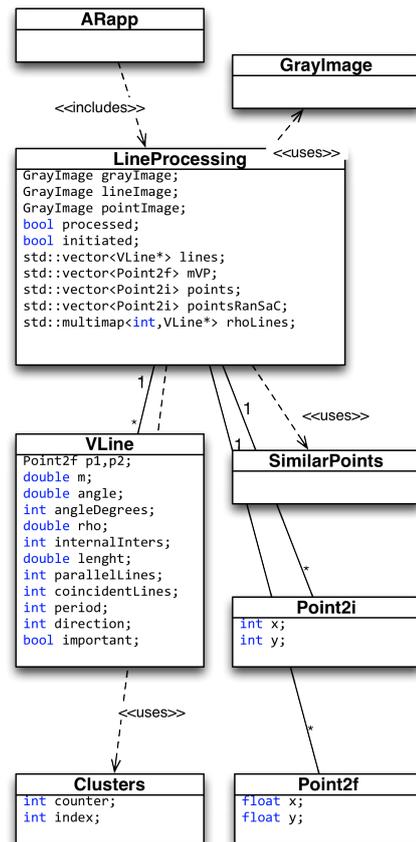


Figura 2: Diagrama simplificado das classes que implementam o sistema de detecção de pontos de fuga. LineProcessing é a classe que centraliza e serve de repositório para toda a informação. A classe VLine representa cada linha que é detectada.

Numa fase inicial os principais contornos da imagem são extraídos utilizando um filtro *Canny*. De seguida os principais segmentos de recta são extraídos usando a transformação de *Hough*. As linhas detectadas são guardadas em estruturas como a *VLine* presente na Figura 2. Um sistema de limpeza de lixo visual elimina as rectas que têm menos probabilidade de pertencer a estruturas e elementos sólidos da cena. Tendo em conta que são detectadas linhas na mesma imagem com parâmetros diferentes, existe uma grande quantidade de linhas que são coincidentes. Quanto mais linhas coincidentes, mais importante é uma linha. Quanto mais intersecções uma linha tiver, maior a probabilidade de essa linha ser gerada por ruído.

Usando as linhas consideradas importantes, é calculada a intersecção entre todas as linhas (ou uma amostra aleatória significativa no caso de serem muitas). Estes pontos são considerados pontos de fuga candidatos. Os vários pontos detectados são agrupados em grupos maiores. Usando algoritmos de *clustering* esse grupos de pontos são classificados em termos de probabilidade de representarem um

²OpenCV, <http://opencv.willowgarage.com/>



Figura 3: Capturas de ecrã da aplicação de posicionamento interactivo de objectos virtuais. A caixa virtual encontra-se pousada no chão e automaticamente alinhada com o cenário onde está inserida.

ponto de fuga válido.

Utilizando os pontos de fuga é possível encontrar o horizonte da imagem e o sitio para onde a maior parte dos elementos na fotografia estão orientados. Utilizando esta informação é definida uma rotação da cena no modelo tridimensional de forma a que quando representados os objectos virtuais tenham a mesma perspectiva que a cena original.

3.3. Aplicação Interactiva e Resultados

Por forma a validar os conceitos introduzidos nas secções anteriores, um protótipo foi desenvolvido que valida uma parte significativa das ideias propostas. A aplicação foi implementada recorrendo às bibliotecas OpenFrameworks³ e OpenGL para visualização gráfica e interacção. Para suporte dos algoritmos de análise de imagem foi utilizada a já referida biblioteca OpenCV. A aplicação pode ser observada nas Figuras 1 (em baixo) e 3 e permite a introdução de objectos que podem ser movidos interactivamente. Os objectos colidem com o chão e estão inseridos na perspectiva da cena. Ao posicionar o objecto o utilizador sente que o objecto se alinha automaticamente com a fotografia aparecendo duas linhas guia verdes. Da mesma forma quando o utilizador pousa o objecto no chão duas linhas guia amarelas aparecem indicando que atingiu o plano virtual. A interacção é feita usando os dedos ou o rato.

Testes realizados e apresentados em [BFRP] mostram que o sistema de criação do modelo a partir de imagens funciona em 70 a 80% das situações consoante o conjunto de fotografias que se testa. Isto quer dizer que quando são colocados, os objectos encontram-se visual e geometricamente correctos. Também foi possível concluir que ao utilizarem este sistema os utilizadores colocaram 32% melhor os objectos a nível tri-dimensional.

4. DISCUSSÃO E CONCLUSÕES

O protótipo actual implementa um sistema que permite introduzir objectos 3D pré-definidos dentro de imagens fornecidas pelo utilizador [Nóbrega12]. É possível mover es-

tes objectos mantendo estes a perspectiva correcta e interagindo com o chão e com a orientação natural do mundo fotografado. As actuais limitações prendem-se com a pouca quantidade de elementos virtuais que podem ser actualmente inseridos e com o facto de o factor de escala não ser neste momento automático. No futuro pretende-se testar este sistema em sistemas móveis que permitam tirar fotografias em qualquer lugar e imediatamente interagir com os objectos virtuais.

5. AGRADECIMENTOS

This work was funded by FCT grants SFRH/BD/47511/2008 and PEst-OE/EEI/UI0527/2011.

6. REFERÊNCIAS

- [Izadi11] Shahram Izadi, Richard A. Newcombe, David Kim, Otmar Hilliges, David Molyneaux, Steve Hodges, Pushmeet Kohli, Jamie Shotton, Andrew J. Davison, and Andrew Fitzgibbon. KinectFusion: real-time dynamic 3D surface reconstruction and interaction. In *ACM SIGGRAPH 2011 Talks*, UIST '11, page 23. ACM, ACM, 2011.
- [Karsch11] Kevin Karsch, V. Hedau, and David Forsyth. Rendering synthetic objects into legacy photographs. *ACM Trans. Graph*, 30(6):1–12, 2011.
- [Klein07] Georg Klein and David Murray. Parallel Tracking and Mapping for Small AR Workspaces. *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, 07:1–10, 2007.
- [Lai11] JH Lai, CL Chen, PC Wu, and CC Kao. Tennis real play: an interactive tennis game with models from real videos. In *Proceedings of the 19th ACM international conference on Multimedia - MM '11*, volume 3, pages 483–492, 2011.
- [Lee09] David C. Lee, Martial Hebert, and Takeo Kanade. Geometric reasoning for single image structure recovery. In *Computer Vision and Pattern Recognition (CVPR'09), 2009 IEEE Conference on*, volume 0, pages 2136–2143. IEEE Computer Society, 2009.
- [Nóbrega12] R Nóbrega and N Correia. Interactive Insertion of Virtual Objects in Photos and Videos. In *Eurographics 2012*, volume 0, pages 0–1. Eurographics Association, 2012.
- [Saxena09] Ashutosh Saxena, Min Sun, and Andrew Y. Ng. Make3D: Learning 3D Scene Structure from a Single Still Image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):824–40, 2009.

³OpenFrameworks, <http://openframeworks.cc/>.

Exploração Interactiva dos Hábitos de Audição de Música

Ricardo Dias, Manuel J. Fonseca, Daniel Gonçalves
Departamento de Engenharia Informática e de Computadores
INESC-ID/IST/Universidade Técnica de Lisboa

ricardo.dias@ist.utl.pt, mjf@inesc-id.pt, daniel.goncalves@inesc-id.pt

Resumo

Devido à expansão da Internet é comum as pessoas utilizarem serviços *online* para registarem detalhes acerca das suas vidas, tais como os hábitos de audição de músicas. Embora estes serviços disponibilizem alguns dados estatísticos e pequenos gráficos, estes foram concebidos apenas para gravar e permitir o acesso directo à informação. Neste artigo descrevemos uma solução para explorar e visualizar o histórico das músicas ouvidas pelos utilizadores. A solução desenvolvida combina uma técnica de navegação temporal com um conjunto de vistas sincronizadas e um mecanismo de filtragem interactiva, de forma a criar uma solução flexível e fácil de usar. Adicionalmente, recorremos a técnicas de *brushing* e *highlighting* para permitir aos utilizadores identificarem padrões e tendências nos seus hábitos. A avaliação experimental revelou que os utilizadores foram capazes de completar as tarefas propostas com uma taxa de erro baixa, bem como conseguiram identificar mudanças nos seus hábitos de audição e recordar eventos das suas vidas, resultados que indicam que a nossa combinação de técnicas de visualização é eficaz na transmissão de informação relevante sobre os hábitos de audição.

Palavras-Chave

Exploração Interactiva, Visualização, Histórico de Audição de Músicas

1 INTRODUÇÃO

Com a expansão da Internet, temos assistido nos últimos anos ao aparecimento de inúmeros serviços de *lifelogging*. Actualmente, é comum as pessoas usarem estes serviços para registarem detalhes das suas vidas pessoais, tais como as músicas que ouvem¹. Neste caso, esta informação é um subproduto do processo de audição, facto que leva a que grandes quantidades de dados sejam geradas e recolhidas de forma automática. Assim, visualizar e compreender esta quantidade de dados torna-se uma tarefa difícil de realizar.

Embora estes serviços enriqueçam a sua oferta apresentando resultados estatísticos, sob a forma de pequenos gráficos, os seus principais objectivos continuam a ser a recolha e o acesso directo aos dados. Cria-se assim a oportunidade para o desenvolvimento de visualizações para este tipo de informação pessoal. Recentemente têm aparecido algumas soluções, sendo as mais importantes as desenvolvidas por Byron e Wattenberg, *StreamGraph* [Byron08], e as criadas por Baur et al., *LastHistory* [Baur10] e as três visualizações complementares *Tangle*, *Strings and Knots* [Baur09]. Também por parte da comunidade tem havido um esforço para visualizar este tipo de informação, destacando-se o *Scrobbling Timeline*² e o *LastGraph*³. Porém, estas técnicas apresentam algumas limitações, tais



Figura 1: Protótipo da Interface Utilizador.

como: estão mais preocupadas em apresentar vistas gerais dos dados com o foco em aspectos estéticos e de entretenimento, em detrimento da interactividade; apresentam limitações na quantidade de dados de entrada; e finalmente, algumas necessitam de componentes externos ou são dependentes do sistema operativo.

Neste artigo descrevemos uma nova solução para explorar e filtrar interactivamente o histórico das músicas ouvidas pelos utilizadores, que combina um mecanismo de filtragem temporal e um conjunto de vistas sincronizadas (ver Figura 1). A solução desenvolvida disponibiliza não

¹<http://www.last.fm>

²<http://playground.last.fm/demo/timeline>

³<http://lastgraph.aeracode.org>

só uma visão geral de todo o histórico, como também um mecanismo dinâmico de filtragem que permite inspecionar um determinado período temporal ou um elemento dos dados (artista, álbum, música). Recorrendo a técnicas de *brushing* e *highlighting* permitimos aos utilizadores identificarem padrões e tendências nos seus hábitos de audição.

Os resultados da avaliação experimental mostram que os utilizadores conseguiram completar as tarefas com uma taxa de erro baixa (<3%), gostaram de utilizar a nossa solução e descreveram-na como fácil de utilizar. Adicionalmente, os utilizadores foram capazes de identificar mudanças nos seus comportamentos e hábitos de audição.

No resto do documento começamos por descrever a solução desenvolvida dando detalhes sobre a sua arquitectura e principais características. Em seguida, descrevemos a avaliação experimental e apresentamos os resultados obtidos. Finalmente, concluímos o artigo e apresentamos possíveis direcções para trabalho futuro.

2 SOLUÇÃO DESENVOLVIDA

A solução desenvolvida é uma nova abordagem para visualizar o histórico de músicas ouvidas, pois combina um mecanismo de exploração temporal ajustável (dia, semana, mês, ano), com um conjunto de vistas sincronizadas, de forma a permitir uma exploração interactiva e a detecção de padrões nos hábitos de audição. Uma vez que os dados do histórico contêm informação temporal, decidimos utilizar técnicas baseadas em mecanismos temporais como princípio orientador da visualização (ver [Aigner07, Muller03] para uma descrição genérica).

2.1 Arquitectura da Solução

De forma a permitir aos utilizadores explorarem os seus hábitos de audição a partir de qualquer lugar, a solução desenvolvida segue uma arquitectura do tipo *cliente-servidor web* típica (ver Figura 2). Do lado do cliente encontra-se o ponto de entrada na solução, através da introdução do nome de utilizador da conta *Last.fm*, bem como todos os mecanismos de visualização / interacção e filtragem. No servidor residem os mecanismos de indexação do histórico de músicas ouvidas, e de pré-processamento dos dados recolhidos. O servidor expõe um conjunto de serviços que permitem obter os dados processados e potenciar o desenvolvimento das diferentes visualizações.

2.2 Visualização e Exploração

A interface utilizador da solução desenvolvida (ver Figura 1), inclui três áreas de interacção: visualização principal (1), filtragem (2) e detalhes e dados estatísticos (3).

2.2.1 Visualização principal

A área da visualização principal (ver Figura 1-(1)) contém o mecanismo de filtragem temporal e a visualização por *pilhas de pontos*, que constitui a principal representação dos hábitos de audição. Nesta técnica, tal como nos trabalhos desenvolvidos por Viégas em [Viégas04, Viégas06], cada pilha de pontos codifica um intervalo temporal do período seleccionado, e cada ponto representa um elemento dos dados (artista, álbum ou música) ouvido nesse

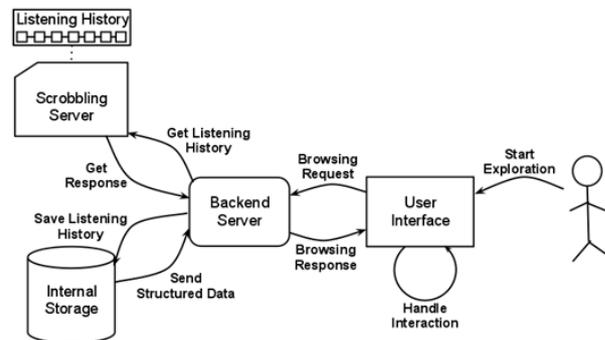


Figura 2: Arquitectura da solução.

intervalo. Na nossa solução introduzimos um novo algoritmo de ajuste temporal, que altera o modo de agregação de dados com base na duração do intervalo de tempo seleccionado, permitindo assim ao utilizador visualizar os dados de forma adequada. Por exemplo, se o utilizador ao visualizar a informação agregada por *dia* aumentar a duração do período seleccionado, isso reduz a dimensão dos pontos. Caso o seu tamanho fique abaixo de um determinado limite, o algoritmo muda o modo de agregação para o nível seguinte da hierarquia temporal (para *semana*), tornando a visualização perceptível novamente.

O painel de fundo nesta visualização contém linhas verticais delimitadoras dos intervalos de tempo mais significativos, etiquetas dos elementos mais tocados e uma representação gráfica da frequência dos hábitos de audição do utilizador nesses intervalos temporais. Embora esta decisão de desenho tenha sido guiada por motivos estéticos e para contextualizar os intervalos temporais, a avaliação experimental mostrou que esta informação ajuda os utilizadores a explorarem a sua informação.

O tamanho e a ordenação dos pontos são codificados usando a frequência de audição dos elementos dos dados e sua relevância no histórico. Ao contrário de outras soluções, usámos a cor dos pontos para introduzir um novo conceito sobre os hábitos dos utilizadores: a "idade das músicas", ou seja, quão recentes ou antigas são em relação ao ano em que foram lançadas e o ano em que foram ouvidas. Para codificar esta propriedade, usámos cores fortes e quentes (tons de vermelho) para representar músicas recentes e cores frias para representar elementos mais antigos (tons de azul). Esta representação permite aos utilizadores obterem detalhes sobre os seus hábitos de audição, podendo observar a sua preferência por músicas mais recentes em detrimento de músicas mais antigas, ou mudanças na audição de músicas/álbums do mesmo artista, mas de décadas diferentes, algo que não pode ser visualizado recorrendo a soluções do estado da arte actual.

Para permitir aos utilizadores obterem informação adicional e detectarem tendências nos seus hábitos, recorremos a técnicas de *brushing* e *highlighting* (ver Figura 3). Sempre que o utilizador passa por cima de um ponto, são apresentadas informações adicionais, e uma linha aparece ligando todos os pontos que representam o mesmo elemento de da-

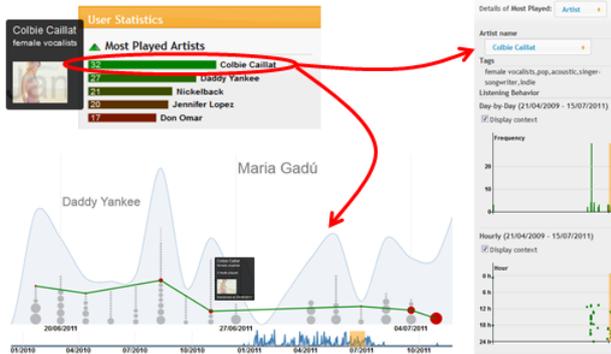


Figura 3: Técnicas de Brushing e Highlighting.

dos. Esta característica torna possível seguir a evolução na audição das músicas/artistas. As cores das linhas correspondem às cores usadas no painel de estatísticas (descrito abaixo).

2.2.2 Filtragem interactiva

A área de filtragem (ver Figura 1-(2)) contém a interface para o mecanismo interativo de filtragem. Os utilizadores podem filtrar a visualização principal usando três filtros: género musical, nome do artista e texto livre. Embora os filtros possam ser combinados e os resultados dinamicamente destacados na visualização, eles apresentam comportamentos diferentes: enquanto o filtro de género e texto livre fazem com que os pontos que codificam elementos desse género mantenham a sua cor e os restantes se desvançam (técnicas de foco+contexto), o filtro do artista faz com que os pontos que codificam o mesmo artista (ou álbuns/músicas desse artista) se liguem para que seja possível se seguir a evolução temporal.

2.2.3 Detalhes e estatísticas

Este painel (ver Figura 1-(3)) exhibe estatísticas gerais e detalhes sobre os artistas, álbuns, músicas e géneros musicais mais tocados, no período seleccionado. Por omissão, os primeiros 5 de cada categoria são mostrados, no entanto, o utilizador pode modificar o número de elementos que pretende ver. Esta informação é representada utilizando gráficos de barras codificados com as frequências de audição, com o esquema de cores a variar entre *verde* e *vermelho escuro*, que permitem visualmente ligar os diferentes painéis de informação, bem como as linhas que ligam os pontos na visualização principal. Adicionalmente, duas vistas, uma por dia e outra por hora, mostram detalhes sobre a evolução temporal dos elementos mais ouvidos. Estas visualizações fornecem detalhes sobre como o utilizador ouviu um determinado artista, álbum ou música, permitindo responder a questões como, "Em que dias ouvi mais músicas do Rui Veloso (e a que horas)? Sempre de manhã?".

2.3 Implementação

Para validar a solução desenvolvida, implementámos um protótipo sob a forma de aplicação web, usando como

principais tecnologias *HTML*, *CSS* e *Javascript*. As visualizações foram criadas através da biblioteca gráfica *Protovis*⁴. O servidor de backend foi implementado sobre a plataforma *Play*⁵ de forma a guardar e processar os dados do histórico dos utilizadores. Como formato de transferência de dados, utilizámos *JSON*, dado que este é usado nativamente tanto no *Javascript* como no *Protovis*.

3 AVALIAÇÃO EXPERIMENTAL

De forma a avaliar a solução desenvolvida conduzimos duas avaliações experimentais: na primeira medimos a capacidade dos utilizadores realizarem um conjunto de tarefas para obter informação e recolhemos a sua satisfação na utilização da nossa abordagem; na segunda determinámos se os utilizadores conseguiram obter detalhes sobre padrões e tendências nos seus hábitos de audição.

3.1 Utilizadores

Na primeira experiência realizámos testes com 10 utilizadores (8 homens e 2 mulheres), de idades compreendidas entre os 20 e os 50 anos, que costumam ouvir música diariamente, e têm profissões diferentes, nomeadamente, 4 eram estudantes de doutoramento, 2 eram engenheiros de software (um deles com formação musical), um trabalhava como jornalista amador, outro era estudante de ciências sociais e os restantes dois eram estudantes de mestrado na área de ciências da computação. Para a segunda experiência conduzimos testes com 4 utilizadores, de idades compreendidas entre os 20 e os 40 anos, que costumam ouvir música diariamente e possuem conta no *Last.fm* (3.000 a 30.000 músicas ouvidas). Todos eram profissionais da área de Engenharia Informática.

3.2 Procedimentos e Tarefas

Cada teste teve a duração média de 45 minutos e foi realizado nos computadores dos utilizadores. Enquanto na primeira experiência foi utilizado um registo com o histórico de um dos autores do trabalho, na segunda experiência foram utilizados os históricos pessoais. Em cada sessão começámos com uma descrição do teste, agradecendo a participação e explicando os objectivos da experiência. Em seguida introduzimos a aplicação explicando as principais funcionalidades, e demos um período de treino para os utilizadores se familiarizarem com a solução. Em seguida os utilizadores realizaram um conjunto de tarefas, cobrindo diferentes objectivos. Finalmente, os utilizadores preencheram um questionário de satisfação, seguido de uma entrevista informal, com o objectivo de recolher opiniões sobre a experiência conduzida.

As 10 tarefas utilizadas nos testes foram criadas com base no trabalho de [Baur10] e agrupadas em dois grupos de acordo com os nossos objectivos. Com as tarefas do primeiro grupo, procurámos avaliar se os utilizadores conseguiram utilizar a nossa solução para explorar de forma eficaz o histórico de audição. As tarefas variavam entre

⁴<http://mbostock.github.com/protovis/>

⁵<http://www.playframework.org/>

a utilização directa da interface para obter informação sobre os artistas mais ouvidos, ou o período temporal com maior frequência de audição, para outras onde a exploração interactiva através da aplicação de filtros era necessária. No segundo grupo, procurámos verificar se os utilizadores conseguiam encontrar e usar a informação disponível para descrever comportamentos e tendências nos hábitos de audição. Os utilizadores realizaram tarefas tais como, descrever a evolução na audição de uma determinada música, ou inferir factos acerca dos perfis de audição (p.ex., a preferência em ouvir álbuns completos ou músicas soltas).

3.3 Resultados

3.3.1 Completude, Taxa de Sucesso e Satisfação

Todos os utilizadores, conseguiram completar as tarefas indicadas, não sentindo dificuldades na sua realização. Em média, 13 dos 14 utilizadores consideraram as tarefas fáceis de realizar (usando uma escala de Likert de 5 pontos, 1-*Muito fácil* e 5-*Muito difícil*), e apenas um considerou a tarefa 5 (sobre exploração) um pouco traiçoeira, devido a uma confusão temporária com o conceito das linhas que descrevem a evolução na audição. A taxa de sucesso foi bastante alta (97%). Uma tarefa era considerada correcta quando os utilizadores conseguiam terminá-la e dar uma resposta correcta (quando pedida). A análise aos questionários de satisfação mostrou que os utilizadores consideraram a nossa solução fácil de usar, e classificaram a experiência como cativante e recompensadora, uma vez que conseguiram aprender algo sobre outros utilizadores e sobre eles próprios.

3.3.2 Detecção de Padrões

Na primeira experiência, embora os utilizadores explorassem o histórico de outra pessoa, todos foram capazes de compreender quatro facetas acerca dos seus hábitos, nomeadamente: recentemente ouviu mais músicas de outros artistas que não os seus favoritos; prefere ouvir álbuns completos durante os dias de trabalho, mas à noite costuma ouvir mais músicas soltas; teve uma mudança sazonal de hábitos muito significativa, passando a ouvir mais músicas de *Reggaeton* em detrimento de música *Pop/Rock*; costuma ouvir mais músicas recentes do que antigas, embora em alguns dias ouça mais músicas antigas (facto difícil de compreender usando as soluções actuais do estado da arte).

Na segunda experiência, os utilizadores tiveram a possibilidade de encontrar padrões pessoalmente relevantes nos seus hábitos de audição e relacioná-los com o seu quotidiano. Por exemplo, um utilizador identificou algumas tarefas que costumava desempenhar no trabalho com base na exploração temporal e nas diferentes vistas disponibilizadas, relatando: "Aqui estava a escrever um artigo para uma conferência, porque estava a ouvir principalmente música clássica, e gosto de o fazer quando estou a escrever". Outro utilizador conseguiu confirmar a sua preferência por música mais antiga, apenas pela inspecção directa da cor dos pontos na visualização; no entanto, também conseguiu perceber que nas últimas semanas tinha ouvido música mais recente, porque um dos seus artistas favoritos aca-

bara de lançar um álbum novo após anos de ausência. Em relação a tendências na audição de músicas, usando as pistas visuais dadas pela informação de contexto, um utilizador conseguiu identificar alguns comportamentos, como por exemplo, não só descobriu músicas que apenas ouvia no fim de semana, porque estava em casa nesse período, como também percebeu que se comportava de forma diferente durante a semana (e em relação aos locais onde estava), com base na frequência com que ouviu determinadas músicas e artistas.

4 CONCLUSÕES E TRABALHO FUTURO

Neste artigo descrevemos uma nova solução para explorar o histórico de audições dos utilizadores, usando uma visualização baseada em mecanismos temporais combinada com um conjunto de vistas sincronizadas. A avaliação experimental revelou que a nossa solução é fácil de usar e que permite aos utilizadores identificarem mudanças e tendências nos seus hábitos de audição.

Futuramente, pretendemos utilizar algoritmos de *data mining* para descobrir novos padrões nos hábitos dos utilizadores, e assim permitir-lhes a exploração dos seus dados de forma diferente, levando-os a aprender mais sobre si. Finalmente, esta informação irá ser utilizada no desenvolvimento de sistemas de recomendação de música, que tenham em conta o contexto temporal e a actividade que se está a realizar.

5. AGRADECIMENTOS

Este trabalho foi apoiado pela FCT (fundos multi-anuais do INESC-ID) através de fundos do programa PIDDAC, e através do projecto Crush PTDC/EIA-EIA/108077/2008. Ricardo Dias foi apoiado pela bolsa de doutoramento da FCT, com a referência SFRH/BD/70939/2010.

6. REFERÊNCIAS

- [Aigner07] Wolfgang Aigner, Silvia Miksch, Wolfgang Müller, Heidrun Schumann, e Christian Tominski. Visualizing time-oriented data - a systematic view. *C&G*, June 2007.
- [Baur09] Dominikus Baur e Andreas Butz. Pulling strings from a tangle: visualizing a personal music listening history. *IUI*, 2009.
- [Baur10] Dominikus Baur e et al. The streams of our lives: Visualizing listening histories in context. *IEEE TVCG*, 2010.
- [Byron08] L. Byron e M. Wattenberg. Stacked graphs geometry & aesthetics. *IEEE TVCG*, 2008.
- [Muller03] W. Muller e H. Schumann. Visualization methods for time-dependent data - an overview. Em *Simulation Conference*, 2003.
- [Viegas04] F.B. Viegas, D. Boyd, D.H. Nguyen, J. Potter, e J. Donath. Digital artifacts for remembering and storytelling: posthistory and social network fragments. Em *HICSS '04*, 2004.
- [Viégas06] Fernanda B. Viégas, Scott Golder, e Judith Donath. Visualizing email content: portraying relationships from conversational histories. *CHI*, 2006.

AUV Planning, An application for AUV mission planning

António Coelho Ricardo Gonçalves Rui Rodrigues
FEUP/DEI/INESCTEC

Rua Dr. Roberto Frias s/n 4200-465, Porto, Portugal

{acoelho@fe.up.pt, refg@inescporto.pt, rui.rodrigues@fe.up.pt}

Resumo

Veículos autónomos submarinos (Autonomous underwater vehicles - AUV), são veículos controlados por computador que operam de forma autónoma em ambientes submarinos. A inacessibilidade que resulta de operar a grandes profundidades complica o planeamento e controlo devido à dificuldade em monitorizar os veículos em ambientes subaquáticos. Neste artigo é apresentada uma aplicação 3D que simula a execução de missões por AUVs em ambientes subaquáticos, permitindo tanto o planeamento como a visualização de missões. A possibilidade de ter uma referência visual dos AUV's enquanto desempenham as suas tarefas pode assim ajudar a evitar problemas e a diminuir o risco de fracasso das missões.

Abstract

Autonomous underwater vehicles (AUV) are computer-controlled vehicles that operate autonomously in underwater environments. The inaccessibility that results of operating at such depths further complicates the mission planning and control because of the difficulty to observe and monitor the vehicles in underwater environments. This paper presents a 3D application that simulates mission execution by AUVs in underwater environments, allowing both the planning and visualization of missions. This possibility of having a visual reference of the AUV while these are performing their tasks may help avoid problems and diminish the risk of mission failure.

Keywords

AUV, Mission Planning, Virtual, 3D, Environment, Unity3D

1. INTRODUCTION

The ocean's floor is full of resources like oil, gas and minerals. Nevertheless, deep sea exploration is dangerous and despite of the great advances made in this area, most of this territory remains unexplored. One possible way to lower the costs and raise the chances of mission success is through the use of autonomous underwater vehicles (AUV). However, AUVs pose their own challenges, such as the inaccessibility to the vehicle [Kuroda96] and the fully autonomous and unattended navigation.

In this paper we present a software application created to help planing and visualizing missions in a 3D environment for the Mares and Trimares AUVs [CN08].

Running the missions virtually as a way of testing them may help reduce potential failures in at-sea trials [Song03, Brutzman95] by anticipating or identifying problems that could otherwise pass unnoticed until the actual live mission. This application is able to receive any amount of tasks and assign them to AUVs to perform, concurrently or sequentially, allowing a higher degree of mission customization. Finally, by providing a visual representation of

missions, people who are not familiar with the subject will be able to better grasp what missions entail, how AUV's perform and the benefits of this approach. Giving them a better understanding of the whole process, this application is being used both for marketing and educational purposes.

2. RELATED WORK

Computer simulation and planning of real life objects (or potential objects) has always been one of the best ways to evaluate scenarios and anticipate problems and flaws, before trying them out on the field. This approach not only allows the decreasing of resource costs, but also provides a platform for safer and faster testing. This is why simulation is present in most fields of engineering. Robotics is one of such fields, related with what we are presenting in this paper and therefore we will look at some simulation applications that currently exist.

The Webots [Michel04] is a commercial development tool that allows simulating mobile robots with custom attributes and shapes in a common environment, as well as the use of predefined robots such as Sony Aibo¹. These robots can

¹<http://www.sonyaibo.net/>

also be programmed through the use of a C/C++ API that uses TCP/IP to communicate with the simulator. Physics simulation is handled by ODE² (Open Dynamics Engine) library.

USARSim [Carpin07] is an open source and free to use multi-robot and environment simulator created with Unreal Engine to model arbitrary application scenarios. This simulator allows the customization of robots through its capabilities of modeling and adding sensors. It also uses Mobility Open Architecture Simulation and Tools framework (MOAST)³ which allows the use of modular modeling and design to the robots, making every module semi-independent and reprogrammable.

Another open source and free simulator is the Player Project⁴. This simulator was designed for research of robotic and sensor systems. Because of its extension capabilities, it allows simulation of both 2D and 3D environments, through the use of Stage and Gazebo backends respectively, and the use of third party plugins that can customize and add more features as third party utilities.

Unity3D⁵ is a full featured engine which can be used to create high quality 3D interactive real time applications. It combines authoring tools for creating content, as well as flexible development facilities for implementing complex behaviors, in an integrated manner. Another strong point of Unity3D is its capability to deploy to several systems such as Windows, Mac, Android, IOS and also as a browser application.

Because of all the positive aspects of Unity3D and also due to the fact that the research group already has experience working with this tool and therefore having a considerably lower learning curve in comparison to the other solutions, the AUV Planner was developed using this game engine.

3. AUV MISSION PLANNER

The AUV mission planner application aims at producing a 3D simulation of a mission, described as a set of tasks assigned to one or more AUV's.

As shown in figure 1, the task manager starts by receiving a XML file (at this point of development, this file is still being created by hand), containing all tasks to be executed, which is parsed by the task manager. The task manager is an object within the virtual world that simulates a console. At this moment the user cannot yet interact with it, but in the future it will allow the user to change simulation parameters of the active mission.

The XML File is divided in three parts, the Console tasks, the tasks to be assigned to the AUVs and finally, camera information. The first part is a list of tasks for the console to execute, and this happens as soon as the file has been parsed. These tasks may include the deployment of AUVs, or the management of the vertical AUV transport rail.

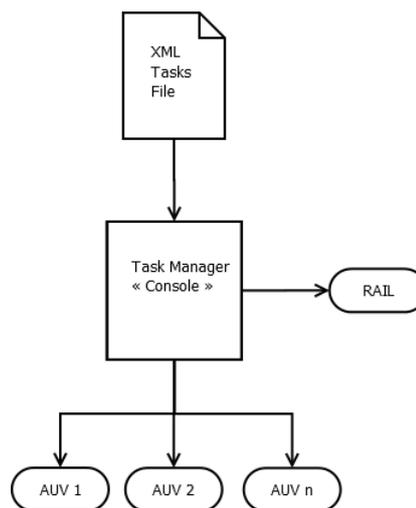


Figure 1. Mission Process

```

<TaskManager>
- <camera>
  <task type="LookAT" position="12.8;-8.0;2.9" direction="11.9;-7.6;2.7"
  id="0" />
  <task type="LookAUV" targetname="SURVEYOR1" position="8;27;2"
  id="1" />
</camera>
- <console>
  <task type="LowerRail" changecamid="2" raildisp="20" />
  <task type="OpenRail" />
  <task type="DeployAUV" targetname="SURVEYOR1" />
</console>
- <auv name="SURVEYOR1">
  <task type="Wait" duration="5" />
  <task type="FollowWP" description="Preparing to dive">
    <waypoint type="followmode" position="15.5;-15.5;-1"
    velocity="0.5;0.4;2" />
    <waypoint type="divemode" position="15.5;-2995;-1"
    velocity="2;0.1;0" description="Diving to damaged pipeline" />
  </task>
  <task type="Search" targetname="damagedpipe"
  description="Searching for pipeline" />
</auv>
</TaskManager>
  
```

Figure 2. XML file example

The AUV task part is divided in sections, one for each AUV. Each section contains the list of tasks to be associated to the corresponding AUV by the console. These tasks can be set in five modes:

1. Wait Mode - The AUV waits for a defined number of seconds until it goes to the next task;
2. Follow waypoints Mode - This task defines one or more waypoints to be followed by the AUV one by one until all waypoints have been visited;
3. Search Mode - The AUV begins a outwards spiral search, with its scanner active, searching for a matching object with the one defined in the task;
4. Pause tasks mode - The AUV enters an indefinite wait state until it receives a signal from another AUV or from the console to resume its tasks;
5. Resume tasks mode - Sends a resume task signal to the console or to an AUV.

²<http://www.ode.org>

³<http://moast.sourceforge.net>

⁴<http://playerstage.sourceforge.net>

⁵<http://unity3d.com/>

3.1. Waypoint following

Each task can also change a camera mode and define a description for the current task. The waypoint following mode is possibly one of the most used and important task mode because it is through this mode that the AUV knows where to move and how to move. Each waypoint has two main fields of information that are always required: the *target coordinates* to where the AUV must move and a *velocity* field that is composed by *top speed*, *slowdown speed* and *minimum distance* to the target coordinates to start slowing down. Besides these mandatory fields, the waypoint can also define a *jump to* field with coordinates to instantly move the AUV to that place, useful for skipping long dives, and also a *heading* field to be used in pair with the *jump to* so that when the AUV is moved, it will be facing the defined direction. As with tasks, waypoints can also change the camera mode and define or redefine the current task description. Camera functionality will be addressed on the next section.

The AUV navigation from waypoint to waypoint starts by slowly rotating towards the destination waypoint and then shifting its position vector, respecting the modeled AUVs top speed of 2m/s, towards its forward vector.

3.2. Camera control

Being able to add camera control information to tasks and waypoints is useful to create demos of the mission with a cinematic feel to it. Cameras can be set in one of three modes that focus on an AUV:

1. Fixed position mode, rotating to follow an AUV
2. AUV chase mode, moving behind an AUV at a defined distance
3. Onboard camera mode, displaying the area in front of the AUV

These camera modes can be triggered to change when an AUV starts a new task, but can only be activated if the camera is set on cinematic mode. However, if the camera is set in free mode, the user can jump between AUVs by pressing a keyboard key, or roam free through the scene.

4. SCENE ELEMENTS

Another of our aims is to show the simulation as close to reality as possible in order to create a compelling and immersive environment. That is, the creation of an underwater look. At first we started looking at real underwater videos in order to identify the main elements in such a scene. Some of these elements were: poor visibility at medium range; light reduction with increasing depth; water particles that flow with underwater currents; water transparency and water disturbance from the propellers.

The next step was mixing all these elements and tweaking them until we got the desired underwater effect. Another aspect of the underwater effect was that the simulation could start above water and therefore, the scene would have to work accordingly. To solve this problem we created two effects, one for above the surface of the water

and another for below. When the camera is above the ocean surface, the scene is composed by a skybox, the sun, and of course the ocean surface. On the other hand, when the camera moves below the ocean surface, the skybox is disabled and uses the camera default background color instead. To address the poor visibility a fog effect was added to occlude the vision. In order to create a seamless color effect in the scene the fog color and the camera's default background color are always the same. As the depth increases this color (cyan blue) darkens, however instead of going totally black as is expected in a real world deep underwater area, one of the goals of this application would be compromised because the whole scene would be darkened. Additionally the sun flares are also disabled under the water and the light source intensity is also decreased as the depth increases, due to the same reason stated above, the default scene ambient light is set to a dark grey that still allows a good visibility at very high depths.

To make the scene look more dynamic, and also enhance the underwater effect, a particle generator was added to the camera that simulates water particles being affected by currents. This particle generator produces particles with random energy in a cubic volume with a side length of 15 meters. Additionally to each particle a random velocity is added to create the illusion of different currents and movement.

Finally to add a sense of movement to the AUV a trail was added to its rear. This trail is a Unity3D specific component called Trail Renderer. Initially, the trail seemed to be dragged by the AUV. To solve this issue a script was added to the trail in order to animate the texture by shifting the texture UVs in the opposite direction of the AUV movement.

For the moment, the AUV Planner is using a fixed 3D scene. However we have plans to create two procedural approaches to scene creation. One and the simplest of the approaches is the use of a XML file with scene specifications, such as the start and end coordinates of a pipeline and with those values the scene generator would add as many pipes as needed. The other approach involves an algorithm that is fed with mission parameters and generates a random scene.

5. CASE STUDY

An initial version of this framework was tested with a mission created for a commercial demonstration of the capabilities of a specific set of AUVs of two types, Mares and Trimares. Because the actual mission planning and supervision mode are still being developed, these were not part of this test.

The created scene takes place in some part of an ocean where at the center of the scene and on the surface there is an offshore platform that has a vertical rail that can transport two Mares AUVs at once from a control room at the top of the platform, down to 15 meters below the surface of the water, as well as bring them back up.

Deep into the ocean, 3000 meters below the surface, a net-



Figure 3. Platform and Rail

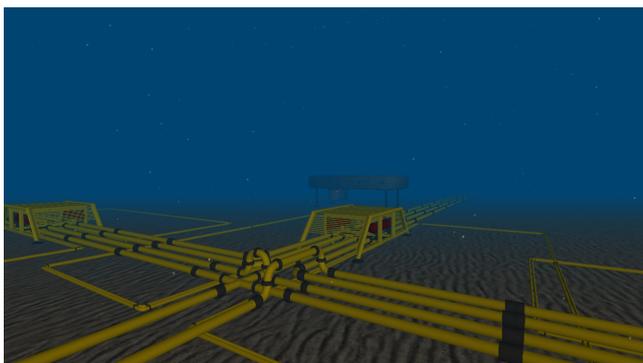


Figure 4. Pipeline network and submarine base

work of pipelines exists, and one of the pipes has suffered damage. Also, near the pipelines there is a small submarine base with a deployment area that can accommodate two Trimares AUVs. The test mission had as objectives the observation and analysis of the damaged area in the pipeline.

The test mission begins with the lowering of the rail, carrying a Mares AUV. When the AUV is finally released at 15 meters below the surface, it begins its descent path until it reaches the depth of 2990 meters. Once there, it begins a spiral search pattern, searching for the pipeline. As soon as it finds the correct pipeline it follows it, with its scanner activated until he finds his target. Once the target has been found it does some preliminary scanning and signals the console it has finished performing the scan. At this point, a Trimares AUV is deployed from the submarine base for a better analysis. Once all tasks are finished, both AUVs return to their respective bases.

6. CONCLUSIONS AND FUTURE WORK

Even though the AUV Planner is still under development, we have been able to visualize some missions for marketing purposes and the results were very satisfactory. The visual impact was tested at the trade fair "Fórum do Mar"¹ and the overall impression was extremely positive. We feel that the environment, especially in 3D, can provide a feeling of deep water exploration. On the other hand, the AUVs were able to perform the mission with success

and the XML-based mission process gives us a strong basis from which to work and evolve this system.

Nevertheless there is still much room for improvement and possibly branch into an educational game. The application is being upgraded to be used as a serious game for learning activities in introductory programming courses. In this new application each AUV will be controlled by agents through a client/server architecture and be able to communicate and cooperate with each other. Furthermore, this agent control plugin will also allow the use of the AI operating in the real AUV's.

Possible features for future development include an overhead map with details of all the objects in the scene and the mission, missions that allow the creation of multiple options for specific scenarios that can be chosen in real time through the console, also a mission designer for an easy way of creating the missions and finally transform AUVs into real AI agents, each with its own control script being run remotely in a client/server architecture.

Furthermore, the AUVs are being developed with dynamic models to accurately simulate their behavior. This simulation tools will be integrated in the application to provide accurate simulations of the AUV behavior under specific underwater conditions.

7. ACKNOWLEDGEMENTS

This work is partially supported by the Portuguese government, through the National Foundation for Science and Technology - FCT (Fundação para a Ciência e a Tecnologia) and the European Union (COMPETE, QREN and FEDER) through the project PTDC/EIA-EIA/108982/2008 entitled "3DWikiU - 3D Wiki for Urban Environments".

8. REFERENCES

- [Brutzman95] Don Brutzman. Virtual world visualization for an autonomous underwater vehicle. In *Proceedings of the IEEE Oceanic Engineering Society Conference OCEANS 95*, pages 1592–1600, 1995.
- [Carpin07] S. Carpin, M. Lewis, Jijun Wang, S. Balakirsky, and C. Scrapper. USARSim: a robot simulator for research and education. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 1400–1405, 2007.
- [CN08] Matos Aníbal Cruz Nuno. The mares auv, a modular autonomous robot for environment sampling. In *Proceedings of the MTS-IEEE Conference Oceans'2008*, pages 1–6, 2008.
- [Kuroda96] Yoji Kuroda, Koji Aramaki, and Tamaki Ura. Auv test using real/virtual synthetic world. pages 365–372, 1996.
- [Michel04] Olivier Michel. WebotsTM: Professional Mobile Robot Simulation. *International Journal of Advanced Robotic Systems, Volume 1, Number 1*, pages 39–42, December 2004.
- [Song03] Feijun Song, P.E. An, and A. Folleco. Modeling and simulation of autonomous underwater vehicles: design and implementation. *Oceanic Engineering, IEEE Journal of*, 28(2):283 – 296, april 2003.

¹<http://www.forumdomar.exponor.pt/>

COGNITO – Captura, reconhecimento e visualização de atividades manuais complexas

Gustavo Mações

Hugo Domingues
Centro de Computação Gráfica
Campus Azurém, Guimarães

Luis Almeida

{gustavo.macaes,hugo.domingues,luis.almeida}@ccg.pt

Luis Paulo Santos
Dep. Informática, Universidade do Minho
Campus Gualtar, Braga
psantos@di.uminho.pt

Resumo

Neste artigo curto apresenta-se um sistema capaz de automaticamente capturar, reconhecer e visualizar atividades motoras humanas, em diferentes contextos, mas com aplicação prática, por exemplo, na implementação de manuais virtuais 3D ou em vídeo-jogos de nova geração, passando pelos simuladores de treino. Este trabalho tem vindo a ser desenvolvido em consórcio internacional, no contexto de um projeto apoiado pela comissão europeia (COGNITO), e tem-se centrado na captura, análise, armazenamento e visualização 3D, com recurso a tecnologias de realidade virtual e aumentada, de tarefas manuais complexas, executadas em ambiente industrial. O sistema é composto por quatro módulos principais: uma rede de sensores colocados no corpo, uma unidade de captura dos movimentos e ferramentas utilizadas, uma componente de aprendizagem não supervisionada e uma componente gráfica capaz de fazer a apresentação de informação ao utilizador através de um módulo de realidade aumentada (RA). Este artigo apresenta o sistema global e a sua arquitetura, referindo com mais detalhe os desenvolvimentos efetuados para a componente gráfica.

Palavras-Chave

Realidade Aumentada, Realidade Virtual e Visão por computador.

1. INTRODUÇÃO

A captura, reconhecimento e representação gráfica da atividade humana tem potencial para gerar diferentes tipos de aplicações desde manuais virtuais, simuladores 3D, vídeo-jogos de nova geração ou automatização de robôs. Contudo, os sistemas de captura de atividade humana existentes no mercado focam-se essencialmente na captura dos dados alinhando os mesmos com um esqueleto predefinido sem preocupação com o tipo de atividade executada. Esta abordagem traz problemas na criação de manuais virtuais ou simuladores porque normalmente estes ficheiros são extensos e implicam pós processamento por parte de um técnico. Nos últimos anos têm sido obtidos avanços significativos no campo da RA, tanto a nível de software como do hardware [Teichrieb07]. No campo do hardware, estão disponíveis novos HMDs com ecrãs e câmaras com alta resolução, novos sistemas de visão por computador e sensores que permitem um *tracking* mais robusto e a implementação de aplicações mais apelativas. Em particular, os sistemas com *tracking* baseado em modelos [Drummond02][Pupilli06], combinam visão por computador e sensores inerciais [You99] e mais recentemente a técnica SLAM - Simultaneous Loca-

lization and Mapping [Davison07]. No campo do software para desenvolvimento aplicacional tem-se assistido ao aparecimento de múltiplas soluções, tais como, por exemplo, o NyARToolkit [NyARToolkit08], StudierStube [StudierStube08], SLARToolkit [SLAR10] e PTAM [Klein07].

Todas estas tecnologias permitem o surgimento de projetos para a implementação de sistemas avançados, que delas tirem partido, como é o caso do projeto MATRIS [Comport06], AMIRE [Grimm02], ARVIKA [Friedrich02] e DWARF [Bauer01].

O projeto COGNITO [COGNITO10] surge nesta sequência e propõe-se especificar e implementar o protótipo de um sistema que irá permitir a criação de um assistente pessoal capaz de dar suporte ao utilizador na realização de atividades e na manipulação de ferramentas para a realização de tarefas industriais complexas. Para isto ser possível é necessário começar por capturar um conjunto de informação relativa aos movimentos do indivíduo e dos objetos da cena (ferramentas que estão a ser utilizadas, etc), utilizando um conjunto de sensores de movimento e de imagem, devidamente colocados no corpo, ao que se segue o processamento necessário à identificação

e segmentação da atividade em ações atômicas. O resultado desta análise é exportado em formato XML, verificado e filtrado por um técnico experiente que identificará as ações atômicas reconhecidas pelo sistema com apoio de um editor desenvolvido para o efeito. Após este tratamento, a informação armazenada está preparada para ser visualizada na componente de RA, durante a execução de tarefas por qualquer utilizador. A secção seguinte apresenta com mais detalhe o sistema COGNITO. A secção três apresenta o estado atual dos elementos constituintes da componente gráfica. Na secção quatro são descritos os cenários de teste e por fim, na secção cinco, são apresentadas as principais conclusões e indicações relativas ao trabalho futuro.

2. ARQUITETURA COGNITO

O sistema COGNITO é complexo e inclui hardware integrado que vai gerar uma grande quantidade de dados, a uma velocidade elevada. O sistema terá de lidar com fluxos de dados heterogéneos da rede de sensores e ainda disponibilizar informação, na forma de RA, em tempo real, ao utilizador. Para além disso, o sistema deve permitir a mobilidade do utilizador, sendo por isso leve e portátil.

A figura 1 apresenta uma representação gráfica do sistema e respetivos módulos constituintes.

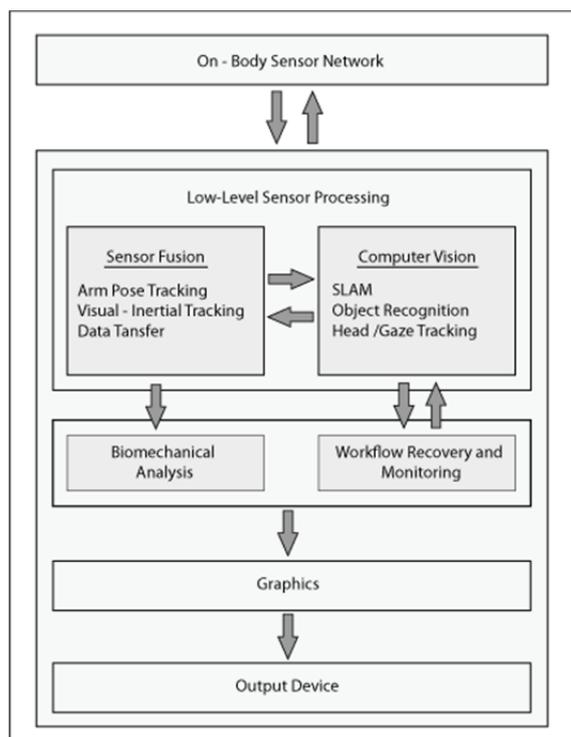


Figura 1: Arquitetura COGNITO

O sistema COGNITO pode ser dividido em 4 módulos principais:

- Rede de Sensores
- Sistema de Captura de Dados

- Componente de Aprendizagem
 - Análise biomecânica
 - Identificação de workflow
- Componente gráfica

A rede de sensores é composta por seis unidades de medição inercial (IMU), que detetam os movimentos do tronco e braços do utilizador. Inclui também uma câmara que fica colocada no peito do utilizador com uma lente olho de peixe que permite visualizar a tarefa a 180° e guardar as imagens captadas para posterior identificação e análise. A rede de sensores fica completa com uma câmara RGB-D que dá informação relativa a profundidade da área de trabalho que ajuda a identificar as ferramentas que o utilizador manipula (fig. 2).

O módulo de captura de dados é responsável por guardar e integrar os diferentes fluxos de dados. Este processamento é feito por uma unidade de visão por computador, que trata das imagens provenientes das câmaras, e por uma outra unidade responsável por adaptar todos os dados ao mesmo sistema de coordenadas (referencial).

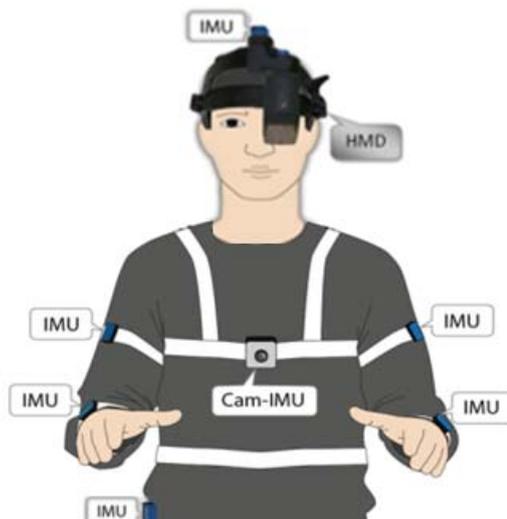


Figura 2: Rede de sensores

Em seguida os dados serão tratados pelo módulo de aprendizagem que analisará os movimentos e objetos manipulados, efetuando uma segmentação da atividade em tarefas atômicas e registando o tempo inicial e final das mesmas.

Por fim, é feito o registo da informação processada, em formato XML, utilizando para o efeito um modelo de ações definido especificamente para o COGNITO. A informação gerada neste processo, será utilizada num editor, com o qual técnicos experientes analisam e filtram as tarefas atômicas definidas e acrescentam informação adicional que posteriormente será usada pelo visualizador RA. Quando todas as ações se encontram identificadas o template de ações é armazenado para ser utilizado pelo assistente (visualizador de RA) que irá guiar o utilizador na execução das tarefas.

3. COMPONENTE GRÁFICA

Esta componente pode ser dividida em duas aplicações principais: um editor da informação capturada e processada na forma de um fluxo de ações elementares e um visualizador de RA para apresentação de toda a informação, utilizando um HMD.

A função principal do editor centra-se na identificação, por parte de um utilizador experiente, das tarefas atómicas que o sistema deteta e guarda no modelo, associando-lhes toda a informação que deverá ser apresentada na forma de RA (modelos virtuais 3D, designação e descrição das tarefas, etc). O Editor (fig. 3) é composto por três áreas principais: na parte lateral esquerda é feita uma representação gráfica do template de ações que contém as tarefas atómicas; na parte lateral direita é apresentado o vídeo ilustrativo da ação atómica selecionada (captado em tempo de recolha de dados); a parte inferior da aplicação é reservada à introdução da informação correspondente a cada ação atómica.

O processo de edição de tarefas atómicas consiste, além da atribuição da designação inteligível, na inclusão de informação adicional que serve para alimentar o visualizador de RA. Esses elementos podem ser objetos virtuais 3D, imagens, vídeos, sons e ainda descrição mais detalhada acerca da atividade que é depois fornecida ao utilizador final recorrendo a um conversor de texto para áudio (TTS).

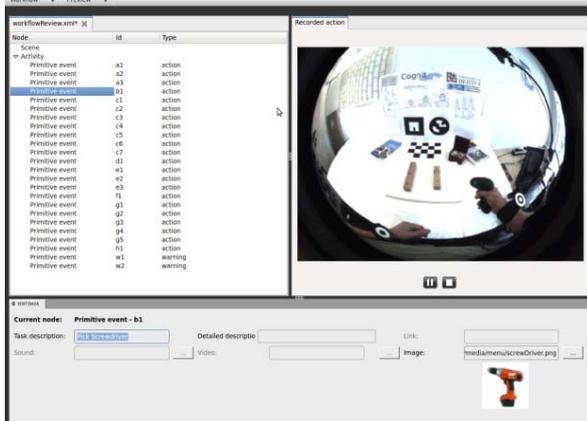


Figura 3: Componente Gráfica - Editor

Após o processo de edição, o modelo de ações fica atualizado para ser lido pelo visualizador RA. A visualização em RA destina-se a utilizadores finais que vão executar tarefas manuais e com necessidade de instruções/ilustrações de procedimento passo a passo (em termos de tarefas atómicas). Esta abordagem imersiva recorrendo a RA pretende manter o utilizador focado na sua tarefa.

O visualizador de informação RA pode ser utilizado de duas formas distintas: integrado no sistema ou de modo autónomo. Quando integrado no sistema, é controlado pelo componente de aprendizagem, a qual, recebendo a informação dos sensores e sabendo quais as ferramentas que estão a ser usadas, quando compara esses dados com as atividades atómicas previamente analisadas, consegue

transmitir instruções ao módulo de RA para apresentar ao utilizador a tarefa corrente e os objetos empregues (fig.4).

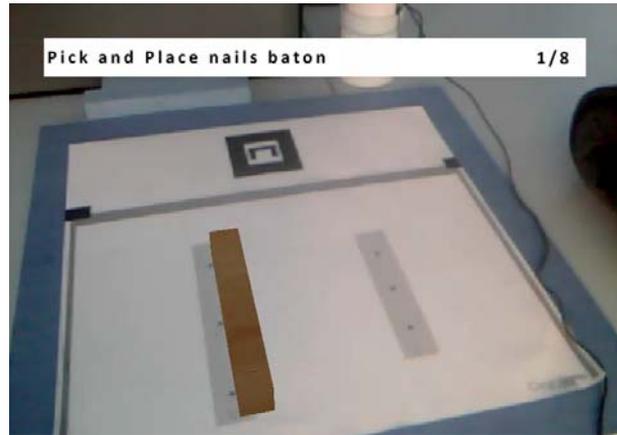


Figura 4: Componente Gráfica - Visualizador RA

No modo autónomo é o utilizador que controla a aplicação através de comandos de voz.

Atualmente, as combinações funcionais que o visualizador RA disponibiliza, em termos de informação apresentada e controlo, são as seguintes:

- Texto + Instrução áudio
- Texto + Vídeo Ilustrativo + Instrução áudio
- Texto + Modelos 3D + Instrução áudio

O utilizador pode navegar nas atividades atómicas podendo avançar, voltar atrás ou pedir para repetir a instrução. A informação de RA apresentada ao utilizador pode variar consoante as suas preferências ou o local onde se encontra a trabalhar. A informação textual e o progresso da atividade encontram-se sempre disponíveis. Além do texto, as descrições alargadas das tarefas atómicas podem ser apresentadas na forma de áudio através da funcionalidade de TTS. Associado à tarefa atómica existe também um pequeno vídeo representativo da tarefa, o qual ilustra o procedimento a executar. O recurso à visualização de modelos virtuais 3D em sobreposição contextualizada à imagem real, é, no entanto, a principal funcionalidade de RA disponível no visualizador, com os quais é possível exemplificar os movimentos a realizar, as ferramentas a empregar, indicar os pontos do espaço físico onde a ação deve ser executada, entre muitas outras representações gráficas possíveis.

4. CENÁRIOS DE TESTE

Atualmente o sistema COGNITO tem vindo a ser testado com sucesso em dois cenários distintos, os quais, não sendo o definitivo (complexo), permitem cumprir o objetivo de realizar testes que validem os vários módulos do sistema e a comunicação entre eles. O primeiro consiste numa tarefa aparentemente simples que instrui o utilizador para posicionar um taco de madeira numa região previamente marcada, fixá-la com três pregos usando um martelo; de seguida, o utilizador deve pegar noutra taco ligeiramente mais pequeno e aparafusar em três locais previamente estabelecidos, usando uma parafusadora. O

segundo cenário testado consiste no empacotamento de embalagens de sabão líquido. Nesta sequência de atividades o utilizador tem que pegar numa embalagem e etiquetá-la; posteriormente coloca-a numa caixa que é fechada e escreve a morada para onde a mesma deve ser enviada.

Estes cenários de teste, apesar de simplificados permitiram já validar todo o sistema COGNITO, com as suas várias componentes integradas e comunicantes, prevenindo-se que o último cenário a ser utilizado, no fim do projeto, consista num ambiente industrial real.

5. CONCLUSÕES E TRABALHO FUTURO

O sistema COGNITO, aqui apresentado, é capaz de aprender ações executadas por um utilizador, de forma autónoma, através de um conjunto de sensores de inércia colocados no corpo e recorrendo a tecnologias de visão por computador. A componente de RA apresenta instruções multimédia, devidamente contextualizadas no espaço físico, para utilizadores não experientes, instruindo a realização de tarefas passo a passo. O uso de realidade aumentada e virtual apresenta-se como uma solução interessante para ajudar pessoas na execução de tarefas, sendo mais eficazes que os tradicionais manuais em papel [Tang03]. Em termos de trabalho futuro, há ainda evoluções a realizar, como por exemplo o cálculo detalhado da posição dos elementos 3D, a partir do reconhecimento de ações em tempo real.

O estudo e melhoramento das representações (texto, elementos gráficos, elementos 3D, etc) e da funcionalidade de reconhecimento de voz no visualizador de RA serão efetuados em conjunto com o parceiro industrial.

A aplicação e teste do sistema em cenários mais complexos é objetivo final do projeto.

6. AGRADECIMENTOS

Este trabalho tem o apoio da Comissão Europeia através do sétimo programa quadro, sob o contracto nº COGNITO FP7-ICT248290. Um agradecimento à Doutora Elizabeth Carvalho pela colaboração no projeto.

7. REFERÊNCIAS

- [Bauer01] Martin Bauer, Bernd Bruegge, Gudrun Klinker, Asa Williams, Thomas Reicher e Martin Wagner Distributed Wearable Augmented Reality Framework (DWARF) Design and Implementation of a Module for the Dynamic Combination of Different Position Tracker, Augmented Reality IEEE, 2001.
- [COGNITO10] COGNITO Project, FP7, ICT-2009.2.1 Cognitive Systems and Robotics, ICT – Information and Communications Technologies, number ICT-24829, <<http://www.ict-cognito.org/index.html>>
- [Comport06] Andrew Comport, Eric Marchand, Muriel Pressigout e François Chaumettex. "Real-time markerless tracking for augmented reality: the virtual visual servoing framework," *Visualization and Computer Graphics, IEEE* 2006.
- [Davison07] Andrew Davison, Ian Reid, Nicholas Molton e Olivier Stasse, "MonoSLAM: Real-Time Single Camera SLAM", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007.
- [Drummond02] Tom Drummond e Roberto Cipolla, "Real-time visual tracking of complex structures", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.
- [Friedrich02] Wolfgang Friedrich "ARVIKA-augmented reality for development, production and service," *Mixed and Augmented Reality*, 2002. ISMAR 2002.
- [Grimm02] Paul Grimm, Michael Haller, Volker Paelke, Silvan Reinhold, Christian Reimann e Jurgen Zauner "AMIRE - authoring mixed reality" *IEEE* 2002.
- [Klein07] Georg Klein e David Murray "Parallel Tracking and Mapping for Small AR Workspaces," *Mixed and Augmented Reality*, 2007. ISMAR 2007.
- [NyARTToolkit08] NyARTToolkit (available at <<http://nyatla.jp/nyartoolkit/wiki/index.php?FrontPage.en>>, visited on 31-5-2012).
- [Pupilli06] Mark Pupilli e Andrew Calway, "Real-time Camera Tracking Using Known 3D Models and a Particle Filter" 18th International Conference on Pattern Recognition 2006.
- [SLAR10] SLARtoolkit (available at <<http://slartoolkit.codeplex.com/>>, visited on 31-5-2012).
- [StudierStube08] StudierStube Project (available at <<http://studierstube.icg.tugraz.at/main.php>> visited on 31-5-2012).
- [Tang03] Arthur Tang, Charles Owen, Frank Biocca e Weimin Mou, Comparative effectiveness of augmented reality in object assembly, SIGCHI 2003.
- [Teichrieb07] Veronica Teichrieb, João Lima, Eduardo Apolinário, Thiago Farias, Márcio Bueno, Judith Kellner e Ismael Santos, "A Survey of Online Monocular Markerless Augmented Reality", 2007.
- [You99] Suya You, Ulrich Neumann e Ronald Azuma, "Hybrid inertial and vision tracking for augmented reality registration", *Proceedings of Virtual Reality IEEE* 1999.

Figuras de acção como interfaces tangíveis para a criação de filmes de animação

Pedro Ribeiro
Centro de Computação Gráfica
Guimarães, Portugal
pedro.ribeiro@ccg.pt

Ido Iurgel
Rhine-Waal University of Applied Sciences
Kamp-Lintfort, Alemanha
EngageLab, Centro Algoritmi
Universidade do Minho, Guimarães, Portugal
Ido.Iurgel@hochschule-rhein-waal.de

Resumo

Figuras de acção utilizadas como interfaces tangíveis possibilitam uma interacção fácil, e intuitiva para crianças que pretendam animar personagens virtuais. Discutimos a experiência da concepção e desenvolvimento do sistema Voodoo, um sistema onde as crianças assumem o papel de um “realizador de cinema”, animando personagens virtuais através de figuras de acção. Este sistema transforma os movimentos e gestos executados sobre a figura de acção em animações, tendo por base padrões de movimentos e o contexto narrativo de uma história bem conhecida. Discutimos sobre uma avaliação preliminar do Voodoo, descrevemos algumas lições aprendidas, e reflectimos sobre o futuro deste tipo de sistemas.

Palavras-Chave

Personagens virtuais, Interfaces tangíveis, Figuras de acção, Brinquedo, Criança, Animação por Computador, Storytelling.

1. INTRODUÇÃO

Voodoo é um sistema criado no âmbito de um projecto interno do Centro de Computação Gráfica (CCG) [Centro12]. *Voodoo* alicerça-se no conceito de Tangible User Interfaces (TUI) [Fishkin04] que define a interacção com o mundo digital, através de objectos tangíveis que são familiares para o utilizador. São interfaces funcionalmente construídas para beneficiar da capacidade natural que os seres humanos têm para interagir com o mundo físico. Este artigo trata de bonecos como TUIs para a manipulação de personagens virtuais (PVs). O bem conhecido conceito de Interação Humano Computador (IHC) *Affordance* [Rogers10], representa a imediata percepção de como se deve naturalmente manusear um determinado objecto. Este conceito explica a percepção automática que uma criança tem quando interage com um boneco. Para a criança é óbvio qual é o correcto manuseamento.

Ao brincar com bonecos, a criança ganha a capacidade de construir histórias e mundos imaginários. Esta forma universal de criar histórias com bonecos, foi a principal motivação para a conceptualização do sistema *Voodoo* [Ribeiro11]. Neste, pretende-se que as crianças ao brincar com bonecos, possam de forma fácil e intuitiva criar histórias, gerando automaticamente a animação de (PVs), que estão inseridas num contexto narrativo escolhido pela criança. Na idealização do *Voodoo* [Ribeiro11] (Figura 1) considera-se a desambiguação da intenção e o aprendizagem do utilizador, os factores “chave” a explorar para

potenciar a utilização de bonecos, como TUI’s orientados para a criação de filmes de animação.

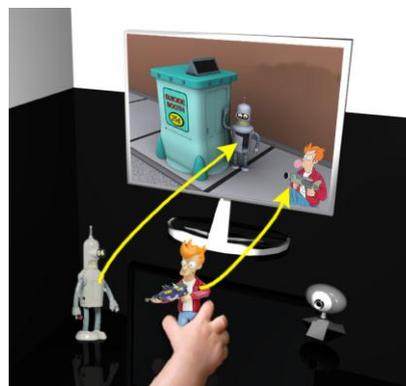


Figura 1: Esquema de funcionamento do *Voodoo*.

Como é que um boneco pode ser manipulado para criar de uma forma controlada, um conjunto vasto de animações? Num sistema capaz de animar o PV do *Capuchinho Vermelho* a falar amedrontado para o PV do *Lobo Mau*, é complexo deduzir a partir dos movimentos feitos sobre um boneco, que o PV do *Capuchinho Vermelho* está a “falar”, e ainda mais difícil perceber que está com “medo”. Os movimentos feitos sobre o boneco, são ambíguos e desambiguar a intenção da criança é um problema importante [Johnson99], [Ribeiro11].

Na conceptualização do sistema *Voodoo* [Ribeiro11], um sistema de animação baseado na interacção com bonecos,

não tem necessariamente que ter um poderoso sistema de adaptação para reconhecer e interpretar os movimentos do boneco. A criança deverá aprender e adaptar os seus movimentos para uma interpretação expectável, “ajudando” assim o sistema na tarefa de reconhecimento e interpretação de movimentos. É importante referir que esta abordagem é fortemente complementada pelo conceito de desambiguação da intenção.

Pretende-se neste artigo apresentar a construção do sistema *Voodoo* [Ribeiro11], discutir os problemas e soluções resultantes da avaliação feita sobre o *Voodoo* e reflectir sobre o futuro do uso de bonecos como interfaces para a criação de filmes de animação.

2. TRABALHO RELACIONADO

Outros trabalhos já exploraram TUIs para o controlo de PVs, direccionado para crianças [Johnson99], [Figueiredo05]. Através deste paradigma, a criança manipula um boneco, o sistema detecta o movimento e anima o PV em conformidade. O *Voodoo* apresenta uma nova abordagem para desambiguar a intenção da criança, e contextualiza este tipo de interfaces no campo da criação de histórias digitais. Para resolver o problema da ambiguidade, o *Voodoo* recorre a informação de contexto adicional, que inclui (i) o papel narrativo que cada boneco “incorpora”, (ii) relação entre personagens de uma narrativa e (iii) ambiente associado a narrativa.

3. CONTEXTO DA HISTÓRIA PARA A INTERPRETAÇÃO DA INTENÇÃO

A metáfora que o boneco representa é ainda mais efectiva no caso da figura de acção (FA), pois estas possuem um papel narrativo e aspecto que se baseia em personagens de um filme, jogo de computador ou banda desenhada. Como o contexto narrativo influencia a interpretação dos movimentos que uma criança faz sobre uma FA, o sistema pode de uma forma mais simples decidir qual a acção que deve ser feita pelo PV. Por exemplo uma criança que encena com uma FA do homem aranha, numa situação em que este personagem tem de abrir uma porta numa situação de combate, provavelmente o PV do homem aranha não rodará a maçaneta da porta, mas dará sim um pontapé na porta. Este último exemplo leva-nos ao conceito de IHC - *ecological constraints* [Rogers10]: as acções de um individuo são influenciadas pelo ambiente que o rodeia. Da mesma forma que a acção de um ser humano é determinada pelo meio, o comportamento do PV deve ser especificado pelo ambiente onde a FA está inserida.

A interpretação do meio envolvente de uma FA deve ser influenciada pela narrativa associada [Iurgel10]. Por exemplo na história de *Hulk* (Marvel), muita informação pode ser extraída do personagem *Bruce Banner*, a partir da sua interacção com outras personagens da história e com o seu meio envolvente: quando furioso, *Bruce Banner* transforma-se fisicamente e psicologicamente numa criatura poderosa e selvagem, sendo que esta transformação é mais provável quando o personagem *General Ross* está perto ou se *Bruce Banner* estiver enclausurado; *Bru-*

ce Banner pode ainda ter comportamentos mais afectuosos quando próximo de *Betty Ross*. Assim a interpretação das acções feitas sobre as FAs devem ter em conta um contexto narrativo.

4. VOODOO

No sentido de proporcionar este contexto narrativo à criança, recorremos a um mapa em papel DIN A4 onde as FAs podem ser colocadas (Figura 2).

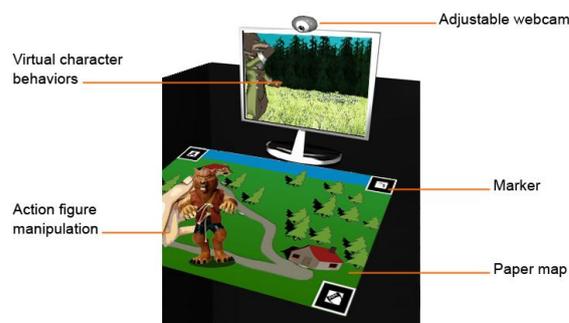


Figura 2: Elementos do sistema

Contrariamente às abordagens que recorrem ao mapeamento directo dos movimentos dos bonecos para animações sobre PV [Mazalek07], [Mazalek09], consideramos que o sistema pode ficar mais intuitivo e robusto, se o mapeamento for baseado nos movimentos feitos sobre a FA, e no contexto em que a FA está inserida.

O protótipo do *Voodoo* que desenvolvemos implementa o conceito anteriormente descrito. Para provar este conceito, recorremos a história bem conhecida do *capuchinho vermelho*¹. O protótipo está direccionado para uma única criança com idade compreendida entre os 6 e os 10 anos, que pode manipular duas FAs ao mesmo tempo, uma em cada mão.

Tecnologicamente este sistema possui um módulo de *input* baseado em visão por computador- *computer vision sensing module* (CVSM), este permite determinar em tempo real qual o ambiente que rodeia as FAs, qual a relação espacial entre FAs, bem como o tipo de movimento que as crianças exercem sobre as FAs. O CVSM baseia-se em duas técnicas: a detecção por cor das FAs; e a detecção de ambientes através da detecção de padrões visuais existentes no mapa. Estes padrões visuais permitem a definição de áreas de interacção no mapa; estas áreas correspondem a ambientes associados a uma determinada história, e.g. A casa do capuchinho vermelho, a floresta, etc. A localização das FAs no mapa é avaliada através da projecção do baricentro da FA, no plano de contextos correspondente ao mapa de papel.

¹ Numa fase prévia a implementação, conduziu-se um estudo informal de *Wizard of Oz* para o desenho do sistema, com duas crianças de oito anos.



Figura 3: Contexto narrativo influencia a animação

Os movimentos das FAs despoletam em tempo real as animações sobre os PVs correspondentes, animações que são visíveis no ecrã (figura 3). Para o reconhecimento destes movimentos, implementou-se uma simples binarização do movimento: movimentos verticais e movimentos horizontais; implementou-se também o cálculo da velocidade dos movimentos. Tendo por base estes parâmetros toda a especificação da animação é determinada através dos ambientes da narrativa: A FA escolhida, a área do mapa onde esta se encontra, e a relação entre os personagens da história que estão no mesmo ambiente narrativo. Assim, através de dois tipos de movimentos mais a velocidade é possível gerar uma grande quantidade de animações.



Figura 4: Escolher história base.

A adaptação deste sistema a outro contexto narrativo implica apenas a criação de um novo mapa de papel com os respectivos padrões visuais, especificar como as diferentes FAs, ambientes da narrativa, movimentos e velocidade dos movimentos afectam o resultado da animação. Actualmente o *Voodoo* inicia com base num ficheiro de especificação de histórias. No futuro um sistema de gestão de contextos permitirá a gestão em *back office* deste ficheiro de especificação de histórias, e conteúdos associados. Neste ficheiro estão descritas todas as histórias que poderão servir de base para a criação de animações. Para além da definição do nome da história e personagens, por cada especificação de história existe a definição do mapa (imagem, padrões visuais usados, nome dos contextos, associação de contexto aos padrões visuais, ambientes

virtuais a serem sintetizados por cada contexto) e a definição comportamental que cada PV da história terá (associar os comportamentos aos movimentos feitos sobre a FA, e PVs a serem sintetizados). Esta especificação das histórias está mapeada na interface com o utilizador, na fase de configuração do sistema. Aqui a criança pode escolher qual é a história (figura 4) que vai dar contexto a sua criação, aprender a associação que existe entre a animação do PV (corporal e facial) e os movimentos que deverão ser feitos sobre a FA, e fazer a associação entre uma FA e um PV (figura 5).



Figura 5: Associar FA ao PV.

5. RESULTADOS PRELIMINARES

Um teste informal do protótipo funcional foi feito com três crianças num ambiente familiar. Estas foram instruídas sobre a funcionalidade do *Voodoo* e convidadas a usar o sistema de forma livre.

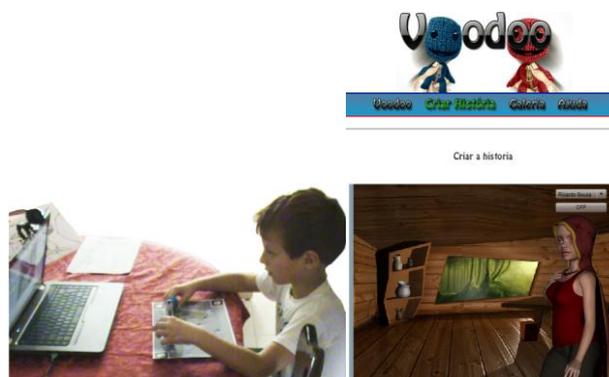


Figura 6: Criança utiliza o *Voodoo* para criar a animação

Avaliou-se a aceitação do sistema através da observação directa dos participantes (figura 6). As crianças interpretaram correctamente as metáforas de interacção implementadas. Perceberam que o posicionamento das FAs iria alterar o cenário sintetizado no ecrã, bem como alterar a aparência (Lobo mau transfigura-se) e comportamentos dos PVs. No que diz respeito a experiência de utilização, as crianças gostaram das animações geradas. No controlo emocional dos PVs, as crianças frequentemente expressam por palavras a emoção do PV. As crianças nunca mostraram intenção clara de criar ou transformar uma narrativa através do *Voodoo*. Os constrangimentos do

teste influenciaram a experiência de utilização, tipicamente brincar é um comportamento espontâneo, a criança brinca quando quer, onde quer e da forma que quer. Relativamente ao CVSM, constatamos que este ainda não é suficientemente robusto e eficiente, pois por vezes a criança oclui os padrões visuais que o CVSM procura.

6. CONCLUSÕES E TRABALHO FUTURO

Ferramentas de criação interactiva de histórias lineares para crianças, têm um enorme potencial pedagógico e lúdico. Um dos principais desafios relacionado com este tipo de ferramentas, é a geração de animações complexas, e de histórias a partir de simples *inputs* gerados através de dispositivos de input intuitivos. Apresentamos uma abordagem onde a interpretação do contexto narrativo permite a uma criança criar filmes de animação, através do manuseamento de FAs como dispositivos de *input*.

Da avaliação preliminar, concluímos que as crianças nunca se abstraíram do teste, influenciando negativamente pois as crianças nunca sentiram vontade de livremente brincar com o sistema e criarem narrativas. Em futuros testes, acreditamos que a solução poderá passar pelo uso de *cultural probes* [Gaver99], pois são eficazes quando é necessário obter informação das crianças com a mínima influência nas suas acções, ou quando o processo e eventos a explorar são de longa duração e esporádicos. Apesar de tudo, foi possível perceber o potencial do conceito de interacção proposto, as crianças facilmente entenderam a lógica de interacção, mesmo já possuindo experiência na utilização de interfaces IHC. Será positivo explorar no futuro os vários aspectos dos bonecos que podem permitir a criação de novas metáforas de utilização, como é o caso do tipo do boneco (FAs, marionetas, etc). O uso de *marionetas* poderia resolver o problema da oclusão do boneco, e facilitar a tarefa de detecção do boneco ao CVSM.

Antevemos potencial na introdução da pós-edição da animação criada. Consideramos que futuramente as emoções e falas das crianças poderão influenciar a decisão das escolhas de animações. O uso colaborativo do sistema também é possível: para isso, o tamanho do mapa de papel e ecrãs devem ser aumentados. Da mesma forma acreditamos que é imperativo que a criança possa configurar os gestos que pretende usar para controlar os PVs. O aumento das FAs com uma rede de sensores e actuadores poderá permitir um *tracking* mais eficiente, bem como transformar a FA numa interface de output (boneco a respirar). Uma tecnologia já existente que em parte pode concretizar esta visão é a tecnologia *QUMA* [quma12]. Por outro lado o sistema poderia ser direccionado para o mundo dos livros. Cada página do livro corresponde a um mapa, com vários ambientes narrativos que podem influenciar a actuação dos PVs. Esta contextualização ao mundo dos livros proporciona uma nova metáfora – a noção temporal representada no virar das páginas, que poderá ser muito útil na criação e edição de novas narrativas. Por fim apontamos para o uso de um mapa virtual, que possa ser visualizado numa mesa multitoque, assim a

detecção da posição das FAs seria facilitada, e o uso colaborativo do sistema seria possível.

7. AGRADECIMENTOS

A implementação do protótipo aqui apresentado, incluiu desenvolvimentos prévios efectuados pela equipa do [Centro12] no contexto do projecto europeu “GUIDE”, nomeadamente a tecnologia de PVs. O GUIDE está contratualizado pela comissão europeia sobe o nº 248893.

8. REFERÊNCIAS

- [Centro12] Centro de Computação Gráfica <http://ccg.pt> (2012).
- [Gaver99] B. Gaver, T. Dunne, E. Pacenti. “Design: Cultural probes”. Interactions: New Visions of Human-Computer Interaction, ACM Press, New York, (1999).
- [Figueiredo05] R. Figueiredo, A. Paiva. Watch and fell: An affective interface in a virtual storytelling environment. In: Affective Computing and Intelligent Interaction, Springer, London, (2005).
- [Fishkin04] K. A. Fishkin. A taxonomy for and analysis of tangible interfaces. In: Affective Computing and Intelligent Interaction, In: Journal of Personal and Ubiquitous Computing, pp. 347-358. Springer, London, (2004).
- [Iurgel10] I. A. Iurgel, R. E. Silva, M. F. Santos. Towards Virtual Actors for Acting Out Stories, In: Proceedings of the Entertainment for education, and 5th international conference on E-learning and games, pp. 570-581. Springer, Berlin, (2010).
- [Johnson99] M. P. Johnson, et al. Sympathetic interfaces: using a plush toy to direct synthetic characters. In: CHI 1999 Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit, pp.152-158. ACM, New York, (1999).
- [Mazalek07] A. Mazalek, M. Nitsche. Tangible interfaces for real-time 3D virtual environments, In: Proceedings of the international conference on Advances in computer entertainment technology, pp. 155-162. ACM, New York, (2007).
- [Mazalek09] A. Mazalek, et al. Giving yourself to the game: transferring a player's own movements to avatars using tangible interfaces, In: Proceedings of the 2009 ACM SIGGRAPH Symposium on Video Games, pp. 161-168. ACM, New York, (2009).
- [quma12] quma. <<http://quma.jp/en/quma/>> (2012).
- [Ribeiro11] P. Ribeiro, I. A. Iurgel, M. Ferreira. Voodoo: a system that allows children to create animated stories with action figures as interface. In: Proceedings of the international conference on Interactive Digital Storytelling, pp 354-357. Springer, Berlin, (2011).
- [Rogers10] Y. Rogers. New theoretical approaches for HCI, In: Journal of Information Science and Technology, pp. 87-143, (2004)

Posters

Human Motion Analysis Tools

João Ferreira Nunes^{1,2}

joao.nunes@estg.ipvc.pt

Pedro Miguel Moreira^{1,3}

pmoreira@estg.ipvc.pt

João Manuel R. S. Tavares⁴

tavares@fe.up.pt

¹ESTG-IPVC - Escola Superior de Tecnologia e Gestão, Instituto Politécnico de Viana do Castelo

²DEI - Departamento de Engenharia Informática, Faculdade de Engenharia, Universidade do Porto

³LIACC - Laboratório de Inteligência Artificial e Ciência de Computadores, Universidade do Porto

⁴INEGI - Instituto de Engenharia Mecânica e Gestão Industrial, Universidade do Porto

Resumo

Análise de movimento é actualmente um tópico de pesquisa bastante activo nas áreas da Visão por Computador, Computação Gráfica e Biomecânica, devido à sua aplicabilidade num vasto espectro de aplicações em diversas áreas. Com este trabalho pretendemos apresentar um detalhado, abrangente e atualizado estudo sobre aplicações de análise e/ou de simulação de movimento, que têm sido desenvolvidas tanto pela comunidade científica como por entidades comerciais. O principal contributo deste estudo, além da listagem abrangente de ferramentas de análise de movimento, é a apresentação de um esquema eficaz para classificar e comparar ferramentas de simulação e de análise de movimento.

Abstract

Motion analysis is currently an active research topic in Computational Vision, Computer Graphics, and Biomechanics mainly due to its applicability into a wide spectrum of relevant applications in many areas. This work intends to present a detailed, broad and up to date survey on motion and/or simulation analysis software packages that have been developed both by the scientific community and commercial entities, to be used in the field of biomechanics. The main contribution of this study, beyond the comprehensive listing of motion analysis tools, is the presentation of an effective framework to classify and compare motion simulation and analysis tools.

Keywords

Motion analysis, motion simulation.

1. INTRODUCTION

Motion analysis is a multidisciplinary research topic dedicated to the detection, tracking and understanding the physical behavior of moving entities. Although this is not a new topic of research, it is very active and receives considerable attention within the research communities of Computational Vision, Computer Graphics and Biomechanics [Gavrila99, Dariush03, Poppe07]. Over the latest years, due to the technological evolutions in personal computers, in video analysis, and in laboratory data acquisition hardware, major contributions have been addressed to develop more robust, more accurate and faster solutions. Many relevant engineering software applications have been developed with the purpose of analyzing and/or simulating human motion. The usage of these tools becomes fundamental in various areas of movement science.

This work presents ongoing work towards a detailed, broad, and up to date survey on human motion and/or simulation analysis software packages that have been developed both by the scientific community and commercial entities to be used in the field of biomechanics. The main contribution of this study, beyond the comprehensive list-

ing of motion analysis tools, is the presentation of an effective framework to classify and compare motion simulation and analysis tools. For the mentioned purpose a set of relevant features is identified and described. As the main outcome, the surveyed tools are classified in respect to the proposed framework and enabling a comparative overview of all the analyzed tools is summarized in a table.

2. HUMAN MOTION ANALYSIS AND SIMULATION

The analysis of human actions is an essential task for a wide range of applications which can be classified under three different groups: surveillance, control and analysis [Moeslund06]. Our work focuses on the class of applications used for analysis. In these type of applications the motion data is reduced to relevant features for comparison and classification purposes. In sports, for example, the biomechanical analysis of the movements of athletes can help them to understand and improve their performances or even facilitate the recovery process after injuries. In ergonomics, the motion data is used to study phenomena related to human postures during their occupation, or to design new office furniture, new products and new workstations. In medicine, the study of human motion can be

also extremely valuable. A good understanding of the human motion, taking into account the biomechanical foundations and the physiology of the elements involved, is extremely important. It can help to reveal the distinct patterns that classify a particular movement. For example, stroke victims exhibit movements that are distinctive from healthy people. With effective classifiers it's possible to have quicker diagnoses and more effective rehabilitation techniques that specifically address the needs of the individual patient.

Simulations using motion data can make predictions of movements based on changes to muscles, positions, and forces. In fact, simulations offer many advantages in respect to physical experiments. They can be used to evaluate and analyze the effects of a hypothetical surgery, to design new products and processes, helping to reduce the number of physical prototypes, as well as expenses and the associated risk [Garcia02].

3. PROPOSED CLASSIFICATION FRAMEWORK

Once selected the most cited motion analysis tools within the scientific literature, we defined a set of features to describe them. This include a set of generic features, a set of modeling capabilities features, a set of data acquisition features, and a set of analysis and simulation features.

3.1. Generic Features

The first set of features attempts to characterize the tool in a generic manner. It comprises: i. the tool latest version and its release date, giving a sense of maturity of the application; ii. if there is an active user community sharing knowledge and expertise using discussion forums, wikis or other collaborative tools; iii. the information specifying whether the tool began as an academic project or not, and in case of affirmative, indicate the seminal paper (whenever possible); iv. the software availability, i.e., whether it is a free or a commercial product; v. openness of the software architectures, in particular what are, if available, the mechanisms to extend the tool with custom functionalities, as for instance: open source, software development kits (SDK) and application programming interfaces (API), plugin developments or scripting; vi. the ability to export the resulting data from motion analysis, allowing further analysis using other tools; vii. the ability to store the patients profile as well as their trials and sessions for easy comparison of "before and after" scenarios.

3.2. Modeling Features

The second set of features describe the modeling capabilities of each tool, indicating: i. the possibility to create and/or edit models; ii. if the tool has its own collection of models that can be used. it is very important that a software solution enables users to develop and edit their own models, and that they have available a set of predefined whole-body and/or body-part models.

3.3. Data Acquisition Features

The third set describes how and what data can be acquired, mentioning: i. if the tool has a native, exter-

nal or absent motion capture system; ii. if the tool imports analogue data, identifying all the sources of the data (C3D/EMG/force plates). The synchronization of all signals allows technical personnel to evaluate simultaneously how the patient moves the articulations, uses his or her muscles and exchanges forces with the ground during the stride cycle.

3.4. Analysis Features

The analysis set of features refers to: i. the dimensionality of the motion analysis (2D, 3D or both); ii. the timing that the analysis occurs (real time, offline or both);

3.5. Simulation Features

The simulation set of features comprises: i. a flag indicating whether it is possible or not to simulate the human movement.; ii. the dimensionality of the motion simulation (2D, 3D or both);

4. CONCLUSIONS AND FUTURE WORK

Human Motion Analysis and Simulation (HMAS) is, since long time ago, an important and multidisciplinary research topic. Applications vary from diverse areas as medical, biomechanics, sports performance and human machine interaction. Recent technological developments allow to provide researchers with automated and semi automated tools in order to model, analyze or simulate human motion. Several open problems, such as marker-less tracking, interactive and real time operation, full 3D acquisition, uncontrolled environments (lighting, occlusions, etc.) still remain. This work presents the state of the art in respect to automated tools in HMAS and introduces our perspective on how HMAS tools can be classified and compared. A logical, structured and feature oriented classification framework is available at <http://bit.ly/HUMANTOOLS>.

As future work, and beyond keeping it up-to-date, we intend to extend the herein presented framework in respect to some particular technologies and approaches and refine it accordingly towards a full taxonomy of HMAS Tools.

5. REFERENCES

- [Dariush03] Behzad Dariush. Human motion analysis for biomechanics and biomedicine. *Machine Vision And Applications*, 14:202–205, 2003.
- [Garcia02] J. M. Garcia, M. Doblare, and J. Cegonino. Bone remodelling simulation: a tool for implant design. *Computational Materials Science*, 25:100–114, 2002.
- [Gavrila99] D. M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73:82–98, 1999.
- [Moeslund06] Thomas B. Moeslund and Adrian Hilton. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(90-126), 2006.
- [Poppe07] Ronald Poppe. Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding*, 108:4–18, 2007.

Outdoor Sports Performance Analysis Enhancement and Multimedia Documentation

J. M. Brisson Lopes
Dep. Computer Science,
Instituto Superior Técnico
Lisboa
brisson@ist.utl.pt

Resumo

Apresenta-se um demonstrador de conceito para o aumento da visualização integrada de dados registados por unidades GPS de pulso em provas desportivas e elementos de multimédia. Uma aplicação para análise de desempenho foi modificada para gerar descrições de percursos em KML, incluindo elementos multimédia.

Abstract

This paper presents a proof of concept for the multimedia visualization enhancement of data from sporting events recorded by wrist held GPS devices. An existing application for performance analysis was modified to generate KML representations that include multimedia.

Keywords

Visualization, animation, data analysis, sports, multimedia synchronization, GPS, KML.

1. INTRODUCTION

Outdoor sports data collecting devices that record GPS data and physiological data (e.g., heartbeat) are widespread now because these devices are sold at ever-cheaper prices with increased functionality, robustness, size and ease of use. Their use extends to a large number of outdoor sports such as mountaineering, tracking, cycling, BTT, ski and running. These devices process location data and produce, display and export data on location, height, speed, climbing speed, number of runs (or laps), etc.

Data can be extracted from these devices and uploaded to computers where it can be fully processed by vendor software or by software developed for extended capabilities that generic and limited vendor software does not support. Data can be exported to applications such as Google Earth but vendor software only use the most primitive features provided by KML (Keyhole Markup Language, [GOOGLE]).

1.1 The Need for Synchronization

Athletes started documenting events using media such as photography and video for publication (e.g., in social media) or practice evaluation, with digital cameras. It is natural then to bring together digital recordings and GPS data to obtain richer snapshots of sports and enable better documentation and analysis of training activities. However, users still have to use several applications to obtain consistent and synchronized multimedia documents.

1.2 Common Mistakes

But users are not experts and make mistakes when using GPS enabled devices. These keep correct time provided

that time zone and daylight saving are properly set. Moreover, most users set camera time only once (right after buying a camera) from the time on a watch that might not be precise, and may confuse 12-hour display mode with 24-hour mode. In case known to us, a 12 hours clock offset camera was taken to a different time zone and photos taken without clock readjustment.

2. RECORDED DATA PROCESSING

GPS data from devices is usually exported using the de facto GPX standard (GPS Exchange Format), [GARMIN]. This lightweight XML format supports most GPS related data (time, position, altitude, speed, heartbeat rate) and features like annotations and media reference. KML (Keyhole Markup Language) lacks the lossless and compactness of GPX but is the logical export format for visualization with Google Earth.

To synchronize photos and video with GPX data one needs to ensure that time and location of both fit together. Good hints are locations where speed is (almost) zero but, for media taken on the run, such fit is not always ensured since any two GPS devices may show different locations. This requires user interaction for confirmation.

3. SOFTWARE DEVELOPMENT FOR PROOF OF CONCEPT

An existing software application ([Brisson11]) imports GPS track data in many formats, including GPX 1.0 and 1.1. It displays GPS data on maps, graphs and summaries and exports data to formats such as GPX and KML.

The application was modified to enable media reference. This called for obtaining date and time information from each media element and, if available, location data.

Digital photo files follow the JPEG (ISO/IEC 10918-1) standard [CCITT92]. The structure of a JPEG file contains several segments. The Exif standard (EXchangeable Image File format for digital still cameras) from the Japan Electronics and Information Technologies Industries Association [JEITA11] extends the JPEG format by specifying a structure for the data to store in the APP1 and APP2 JPEG marker segments. This data is very complete and includes data on the camera and lens and a full set of data on the actual shooting conditions (aperture, shutter speed, metering mode, exposure time, exposure program, focal length, white balance, ISO speed and many other). Date and time information from digital photos can thus be correlated with location and time from GPS devices.

The existing software was upgraded to automatically detect idle times and pauses and suggest locations for media elements. A dialogue to set Google Earth visualization options when saving a track in KML format allows users to specify how multimedia, waypoints, route and track data are to be shown on Google Earth. By setting the appropriate options in the dialogue one can display a more rich and complete picture of the data with location altitude and name, as Figure 1 shows.

3.1 Track and Media Display

The KML options setting dialogue allows referencing media files with route, track and waypoints data. Media thumbnails of the photos are then displayed on Google Earth at the appropriate locations. Clicking on the thumbnails opens up the full media file. For this to happen, the dialogue requests the user to input a file name containing the URL of the media location, the media file names and an optional time synchronization gap enable time fine-tuning of time stamps from the Exif information stored with each photo.

Figure 2 shows a track with a few image thumbnail holding billboards. Clicking on one of the billboards downloads its photo (Figure 3). The track and photos were recorded by a group of bike raiders on a bike raid.

4. CONCLUDING NOTES

The results reported above of a test and demonstration application for enhancing track data presentation together with multimedia show that the concept behind this work was solid and can now be fully extended.

In the present work we detected several types of errors that occur when GPS data is exported to GPX format by different exporters and their impact on multimedia synchronization. Media shooting location and time error sources were also identified. These will be addressed in future work.

Work will progress on the above and on the extension to support media types other than photographs and links or references to media files that were not shot at the time of the GPS recordings that will be referencing them.

5. REFERENCES

[Brisson11] J. M. Brisson Lopes, Interacting with Outdoor Sports Performance Data, 5th SIACG, Faro, June 2011.

[CCITT92] CCITT, Information Technology – Digital Compression And Coding Of Continuous-Tone Still Images – Requirements And Guidelines, ISO/IEC 10918-1, September 1992,

<<http://www.w3.org/Graphics/JPEG/itu-t81.pdf>>

[GARMIN] GARMIN, GPX: the GPS Exchange Format, Version 1.1, Garmin, July 2007,

<<http://www.topografix.com/gpx.asp>>.

[GOOGLE] Google, Keyhole Markup Language, KML Documentation,

<<https://developers.google.com/kml/>>.

[JEITA10] JEITA, Exchangeable Image File Format for Digital Cameras: Exif Version 2.3, JEITA, April 2010.

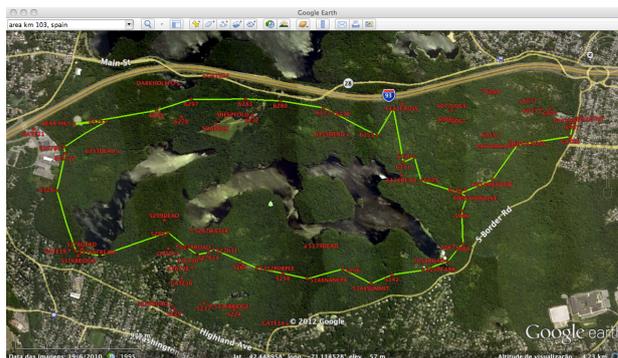


Figure 1: Route and waypoints showing altitude and name labels.

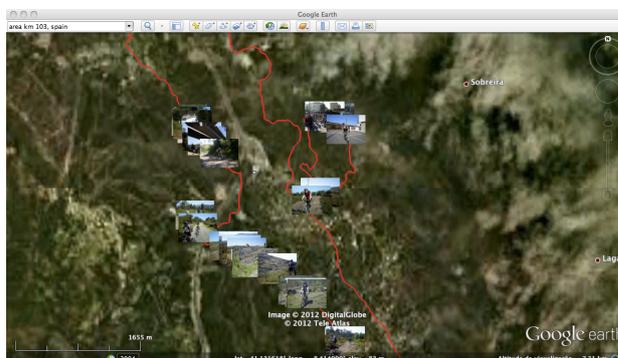


Figure 2: Track with thumbnail images of photos taken during a bike raid.

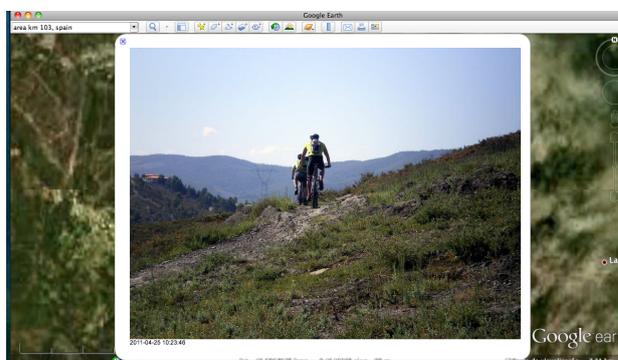


Figure 3: Image displayed by selecting its thumbnail on the track display of Figure 2.

Utilização de Objetos Reais em Aplicações de Realidade Aumentada para Dispositivos Móveis: Aplicação de Advertainment

Nuno Cardoso, Fábio Rocha¹
{nunocardoso83, fabioalvesrocha}@gmail.com

Pedro Miguel Moreira^{1,2,3}
pmoreira@estg.ipv.pt

¹UM - Universidade do Minho

²IPVC - Instituto Politécnico de Viana do Castelo

³LIACC – Lab. de Inteligência Artificial e Ciência de Computadores, Universidade do Porto

Resumo

A evolução das tecnologias de realidade aumentada e dos dispositivos móveis permitem que hoje em dia estes executem aplicações de realidade aumentada utilizando como padrões de reconhecimento uma qualquer imagem a cores, substituindo o tradicional padrão a preto e branco. Neste artigo é apresentado um conceito de aplicação de realidade aumentada, com foco no 'Advertainment', utilizando produtos comuns do dia a dia. Tirando partido da presença do logotipo e da forma geométrica de um determinado produto podemos criar aplicações que utilizam estes fatores de forma mais ativa misturando-os com os elementos virtuais de modo a criar aplicações que inserem um elemento real no mundo virtual, criando uma nova experiência. Este conceito permite que uma aplicação de entretenimento tenha um cariz orientado à publicidade beneficiando a relação do utilizador com a marca. Este conceito é demonstrado através da descrição de um protótipo sobre os quais se realizaram experiências preliminares. Os resultados são promissores quanto à sua viabilidade e interesse.

Palavras-chave

Realidade Aumentada, Realidade Virtual, Jogos, Mobile, Android, Advertainment.

1. INTRODUÇÃO

As tecnologias de Realidade Aumentada tem vindo a evoluir bastante nos últimos tempos permitindo hoje em dia a utilização de uma qualquer imagem como padrão de reconhecimento, substituindo o tradicional padrão a preto e branco.

Os recentes dispositivos móveis permitem a execução de aplicações 3D com um nível de detalhe gráfico bastante agradável. Esta evolução permitiu o suporte de tecnologias de Realidade Aumentada abrindo novos horizontes para a criação de aplicações [Preston11].

A maioria das aplicações de Realidade Aumentada baseiam-se em detetar um determinado padrão e sobre este desenhar elementos virtuais que vão desde conteúdos informativos, animações ou jogos.

Este artigo propõe um conceito de aplicação de Realidade Aumentada para dispositivos móveis, a funcionar sobre a plataforma Android, utilizando produtos comuns do dia a dia tirando partido do seu logotipo e da sua forma geométrica. Este conceito permite explorar novas áreas da publicidade e do entretenimento [Chehimi07] podendo causar nos utilizadores a necessidade de procurar um determinado produto para experimentar o resultado da sua utilização numa aplicação de Realidade Aumentada.

2. CONCEITO

Nos tempos de hoje a maior partes dos produtos comuns vem acompanhados de um logotipo identificativo da marca que o criou. Tirando partido da presença do logotipo num determinado produto podemos criar aplicações que fazem uso tanto do logotipo como da forma geométrica desse produto fazendo com que o produto se misture com os elementos virtuais de modo a criar aplicações que inserem um elemento real no num mundo virtual criando uma nova experiência.

Para a implementação deste conceito criamos um pequeno jogo com foco numa lata de refrigerante de uma marca conhecida. Neste jogo a lata funciona como um elemento presente no cenário interagindo com os objetos virtuais como se fosse também um objeto virtual.

3. IMPLEMENTAÇÃO

Para a criação do jogo utilizamos uma lata de Coca-Cola como objeto de teste e o seu logotipo como padrão de reconhecimento. A aplicação após reconhecer o padrão desenha uma ilha, inserindo a lata no cenário, um personagem virtual no topo da lata e um conjunto de caixas que circulam em volta da lata. O jogo passa por destruir as caixas e fazer o personagem saltar para a ilha. O papel da lata neste jogo, para além de elemento do cenário é

criar a oclusão dos elementos virtuais que circulam em sua volta e de partes da ilha, misturando o real e o virtual. As tecnologias utilizadas no desenvolvimento do jogo foram o Unity 3D [Unity] para a criação e programação dos conteúdos 3D e a extensão do Qualcomm Vuforia [Vuforia] para Unity como tecnologia de RA.

A integração da lata real no mundo virtual passa por colocar uma lata virtual com o tamanho aproximado da lata real tendo como referência o padrão de reconhecimento, que neste caso é o logotipo, posicionando-se na imagem na mesma zona que a lata real. O conhecimento á partida da relação do tamanho e disposição do logotipo com a forma geométrica da lata real permite esta abordagem. A lata virtual é utilizada para fazer a oclusão dos elementos virtuais e a lata real como elemento a ser mostrado no ecrã. O Unity permite aplicar aos objetos um tipo de material chamado “Depth Mask”, este tipo de material permite a um objeto não ser desenhado mas continuar a ser usado nos cálculos de visibilidade criando a oclusão dos elementos ou parte deles que se encontrem por trás deste objeto, este material é usado na lata virtual para criar este mesmo efeito.



Figura 1: Objeto (esquerda) e padrão (direita) utilizados

4. RESULTADOS

Com o resultado final da implementação do conceito num jogo, como vemos na Figura 2, verificamos que trabalhando mais sobre este conceito é possível criar aplicações que transmitem ao utilizador a sensação de que o objeto real se mistura convincentemente com o virtual.

Relativamente a tecnologia de Realidade Aumentada usada, o Qualcomm Vuforia, verificamos algumas limitações e ou problemas. O Vuforia permite a utilização de padrões de reconhecimento utilizando uma qualquer imagem impressa numa superfície plana ou com um pequeno grau de curvatura, como é o exemplo de uma folha de revista ou uma folha A4. O uso de uma lata de refrigerante, cuja forma geométrica é cilíndrica traz alguns problemas na utilização do logotipo como padrão. Verificamos que estando o logotipo disposto na horizontal resulta num ineficaz reconhecimento, pois o Vuforia assume que a imagem á partida esta impressa numa superfície plana. A utilização do logotipo disposto na vertical e com uma largura reduzida permite contornar esta limitação pois aí o grau de curvatura é menor facilitando o reconhecimento pelo Vuforia.

Ao escolher uma lata como objeto de teste estávamos cientes de que as reflexões criadas pelo material exterior deste objeto iriam causar problemas na deteção do padrão por parte do Vuforia.

As condições de iluminação continuam a ser um problema nas tecnologias de RA, incluindo o Vuforia. Concluímos que em situações de muito boa iluminação natural os resultados são bons, em espaços de iluminação reduzida a deteção oscila bastante provocando também um oscilar no desenho dos elementos virtuais.



Figura 2: Capturas de ecrã do jogo

5. CONCLUSÕES E TRABALHO FUTURO

Neste artigo demonstramos que o conceito usado como ação publicitária e entretenimento [Chehimi07] pode impulsionar a procura de um produto em conjunto com uma nova experiência. As tecnologias utilizadas foram adequadas ao desenvolvimento do protótipo como demonstração do conceito.

Como trabalho futuro pode ser explorado a utilização em conjunto do reconhecimento de imagens e das emergentes tecnologias de seguimento 3D.

6. REFERÊNCIAS

- [Preston11] Jon A Preston, Jeff Chastine. Mobile Technology for Games and Augmented Reality. *In Proceedings of ACM Shoutheast Conference, March 2011.*
- [Chehimi07] Fadi Chehimi, Paul Coulton, Reuben Edwards. Augmented Reality 3D Interactive Advertisements on Smartphones. *In Proceedings of Sixth International Conference on the Management of Mobile Business, July 2007.*
- [Unity] Site Oficial.
<<http://unity3d.com>>
- [Vuforia] Site Oficial.
<<http://www.qualcomm.com/solutions/augmented-reality>>

Realidade Aumentada para Caça ao Tesouro em Larga Escala

João Fonseca
DI-Universidade do Minho
Braga
pg18976@alunos.uminho.pt

Gilberto Felgueiras
DI-Universidade do Minho
Braga
pg21000@alunos.uminho.pt

Pedro Miguel Moreira
LIACC/IPVC/DSI-UM
Portugal
pmoreira@estg.ipvc.pt

Resumo

Este artigo apresenta e descreve a implementação de uma aplicação móvel que faz uso de dois sistemas de realidade aumentada para assistir o utilizador numa modalidade de caça ao tesouro. Esta actividade lúdica, conhecida por geocaching, consiste na exploração e descoberta de artefactos, escondidos por todo o mundo pelos vários milhões de praticantes desta modalidade. Os dois módulos que compõem a aplicação, fazem uso das crescentes capacidade dos smartphones actuais, necessitando assim que estes possuam uma antena GPS e/ou uma ligação de internet móvel para obtenção das coordenadas da sua posição; e uma câmara para a captura de códigos de barras, que estarão nos itens escondidos. Neste sentido, o uso de QR-Codes possibilita duas coisas, a obtenção de informação sobre o artefacto em si ou dos passos seguintes na caça ao tesouro, e também a captura e detecção do seu padrão para apresentar assim esta informação no ecrã do smartphone, num sistema de RA.

Palavras-Chave

Realidade Aumentada, Geocaching, QR Code, Caça ao Tesouro, Computação Móvel

1. INTRODUÇÃO

A caça ao tesouro pode ser considerada um desporto ou uma atividade lúdica. Um praticante guiado por um mapa (coordenadas, passos ou dicas) tem de tentar descobrir um artefacto escondido num determinado local. Dentro deste domínio de jogos de Caça ao Tesouro, existe o *Geocaching*, uma plataforma à escala mundial onde um praticante pode esconder um ou vários tesouros - compostos por um invólucro que segue os modelos aconselhados pela plataforma, um lápis/caneta, bloco de anotações [Geo] - num determinado local e disponibilizar as coordenadas GPS desse ponto na *web* para que outros praticantes o tenham encontrar. Além das posições geográficas, o utilizador poderá também disponibilizar uma descrição do lugar, entre outras informações que considere importantes. No universo *Geocaching*, o tesouro é chamado de *cache* e o praticante *geocacher*. Este artigo faz referência ao desafio de *Geocaching* de procura de vários tesouros - *multicache*.

Nesse contexto, este artigo foca-se na implementação de uma aplicação que une elementos reais de uma procura de *caches* escondidos à escala mundial com elementos sintetizados, de forma a que nesta busca possa ser adicionada informação geográfica e histórica dos locais onde as *caches* se encontram, e que ao mesmo tempo consiga criar um ambiente apelativo e divertido de caça ao tesouro e de partilha social.

2. OBJETIVOS DA APLICAÇÃO

Esta aplicação é direccionada para sistemas móveis detentores de GPS. Um utilizador deverá seguir as coordena-

das GPS de uma *cache* e quando esta é encontrada deverá possuir um padrão que dá assim acesso a uma camada de RA. Este padrão deverá ser criado e anexado à *cache* pelo seu responsável e deverá conter a informação que este considere necessária. Se for uma *multicache* então o padrão também deverá conter informação para a *cache* seguinte. Para este tipo de interação com os praticantes será necessário adotar um conjunto de padrões capazes de armazenar a sua própria informação, daí a utilização de *QR Codes*

3. QR CODES

Os QR Codes são códigos de barras bidimensionais e de dois tons, geralmente preto e branco e cujo padrão contém *bits* de informação codificada que posteriormente serão decodificados por uma aplicação. As mais comuns formas de utilização deste tipo de tecnologia são o acesso rápido a sítios na internet e a disponibilização de informação adicional sobre um objeto [Nikolaos10]. Este tipo de padrões foi desenvolvido pela Denso Wave Corporation em 1994. Ao contrário dos marcadores convencionais de RA, os QR Codes são de fácil geração; é possível gerar aproximadamente 10^{7089} padrões diferentes [Kan09] e estes códigos podem conter uma quantidade de caracteres considerável. É uma tecnologia acessível a quase todos os membros da população mundial e mesmo quando danificado, pode continuar a ser possível a sua decodificação.

Estas características tornam os QR-Codes como um conjunto de padrões ideais para um ambiente densamente povoado e de informação diferente como o *Geocaching*.

4. ARQUITETURA DA APLICAÇÃO

A aplicação faz uso de dois sistemas diferentes de RA. O primeiro tem por objectivo indicar ao utilizador as posições das *caches* que se encontram na sua vizinhança; o outro sistema de RA permite ao utilizador que depois de descoberta uma *cache* e estando na presença do QR Code correspondente, direciona a câmara para o respetivo *pattern*, obtendo assim a sua informação.

4.1. Nearest Caches

Dependendo do local em que o utilizador está, é possível ter acesso às posições GPS das *caches* que se encontrem nas suas redondezas. Fazendo uso da bússola, GPS e câmara de vídeo do smartphone, é criada uma camada de realidade virtual no ecrã do dispositivo, onde são desenhados ícones que representam as *caches*. Quando o utilizador direciona a câmara para o horizonte, dependendo da orientação obtida através da sua bússola, é possível visualizar estes ícones, tendo acesso no momento às localizações das *caches* [Wik]. Além das suas localizações, se o utilizador tocar num destes ícones, pode ter acesso a informação extra.

4.2. Capture QR Code

O segundo módulo consiste na deteção e descodificação de um QR-Code que o artefacto possui. Descodificando este código de barras, temos acesso a informação relevante para a caça ao tesouro. Este módulo segue um conjunto de passos que visa detetar este código de barras e apresentar a sua informação no ecrã.

No primeiro passo, usamos uma frame capturada pela câmara (que está em constante captura), e é analisada para determinar se estamos na presença de um QR-Code. Assim que os pontos que identificam um código bidimensional são detectados, é calculada a sua posição na frame capturada. Com a posição do QR-Code, desenhamos a nossa camada de RA sobre a imagem capturada. Estes passos iniciais são executados sempre que possível para aumentar o realismo da camada de RA. Paralelamente, a aplicação também tenta, sempre que possível, descodificar o padrão. Este processo é a operação mais pesada computacionalmente, por isso é executada pontualmente, ou seja, assim que detetamos a posição do padrão, desenhamos a camada de RA e apenas quando o conseguimos descodificar com sucesso, apresentamos a sua informação no ecrã. Esta informação contida no código de barras, tanto poderá ser informação adicional sobre a *cache* que acabou de *caçar*, como pistas para os próximos passos a seguir. Devido ao ruído que algumas frames capturadas poderão ter, assim como o facto de poderem estar desfocadas, algumas pequenas optimizações foram introduzidas para melhorar a experiência de RA. Existem frames onde não é detetado nenhum QR Code - por ruído, imagem desfocada ou inexistência de um - por isso, introduzimos um valor de tolerância, usado para minimizar os *saltos* gráficos durante a captura.

5. TECNOLOGIA UTILIZADA

Esta aplicação foi desenvolvida para a plataforma *Android*. Para a deteção e *decoding* de QR Codes, foi usada a biblioteca open-source ZXing [ZXi]. Algum do seu código teve de ser modificado para servir os nossos propósitos, uma vez que o objetivo principal desta biblioteca é o *decoding* de códigos de barras bidimensionais e não a componente de RA. Na componente *Nearest Caches* mencionada acima, usamos a API do Wikitude [Wik]. Com o recurso a esta API, podemos rapidamente implementar uma camada de RA que faz uso da localização GPS e da orientação do dispositivo com recurso à bússola.

6. CONCLUSÃO

Faltam ainda passos para tornar a experiência de RA mais funcional, interativa e mais fluída nesta aplicação. Para isso, e como trabalho futuro, o cálculo das transformações espaciais dos padrões terão de ser otimizados, para que o objeto virtual se oriente segundo eles; o design de aplicação terá de ser melhorado de forma a torna-la funcional e visualmente agradável, para que no futuro, se torne viável para a utilização e integração no ambiente *geocaching*. Para que os *geocacher* que desejem criar e colocar *caches* nalguma localização, poderá ser necessário a criação de uma plataforma *web* que promova a criação dos próprios modelos virtuais de apresentação de informação, gerando um código QR Code, que será eventualmente colocado na *cache*. Assim a aplicação deverá, para além de descodificar o padrão, fazer o parser dessa informação e gerar o objeto 3D criado anteriormente. Esta aplicação poderá tornar a atividade de *geocaching*, uma atividade mais rica educacionalmente e culturalmente.

7. REFERÊNCIAS

- [Geo] *Geocaching*. <http://www.geocaching.com/>.
- [Kan09] Tai-Wei Kan, Chin-Hung Teng, and Wen-Shou Chou. Applying qr code in augmented reality applications. In *Proceedings of the 8th International Conference on Virtual Reality Continuum and its Applications in Industry*, pages 253–257, NY, USA, 2009. ACM.
- [Nikolaos10] Tsouknidas Nikolaos and Tomimatsu Kiyoshi. Qr-code calibration for mobile augmented reality applications: linking a unique physical location to the digital world. In *ACM SIGGRAPH 2010 Posters*, pages 144:1–144:1, NY, USA, 2010. ACM.
- [Wik] *Wikitude*. <http://www.wikitude.com/>.
- [ZXi] *ZXing Zebra Crossing*. <http://code.google.com/p/zxing/>.

Extração de informação de texto formal para a modelação expedita de edifícios monumentais

Ricardo Baptista
DEI - FEUP/ LIACC
Rua Dr. Roberto Frias, s/n
4200-465 Porto
ricardjose@fe.up.pt

Luís Paulo Reis
Universidade do Minho/LIACC
Guimarães/Porto
lpreis@dsi.uminho.pt

Luís Magalhães
Universidade de Trás-os-Montes e
Alto Douro/INESC TEC
Vila Real/Porto
lmagalha@utad.pt

Roberto Rodrigues António Coelho Augusto Sousa
INESC TEC / DEI - Faculdade de Engenharia, Universidade do Porto
Rua Dr. Roberto Frias, s/n, 4200-465 Porto
{roberto.rodrigues, acoelho}@fe.up.pt

Resumo

A extração de informação a partir de descrições textuais para a modelação procedimental de ambientes urbanos é apresentada com solução para os edifícios antigos. No entanto, este tipo de edifício carece de maior cuidado com os detalhes de alto nível. Este artigo descreve uma plataforma para a geração expedita de modelos 3D de edifícios monumentais, cuja arquitetura é modular. O primeiro módulo permite a extração de informação a partir de textos formais, pela integração do NooJ num Web Service. No segundo módulo, toda a informação extraída é mapeada para uma ontologia que define os objetos a contemplar na modelação procedimental, processo esse realizado pelo módulo final que gera os modelos 3D em CityGML, também como um Web Service.

A partir desta plataforma, desenvolveu-se um protótipo Web para o caso de estudo da modelação das igrejas da cidade do Porto. Os resultados obtidos deram indicações positivas sobre o modelo de dados definidos e a flexibilidade de representação de estruturas diversificadas, como portas, janelas e outras características de igrejas.

Palavras-chaves

Procedural modelling, natural language processing, information extraction.

1. INTRODUÇÃO

A modelação expedita apresenta-se como uma solução para a criação de ambientes urbanos com redução de custos e de tempo de execução. Contudo, para manter uma fidelidade visual aceitável é preciso aceder a fontes de informação, que no caso dos edifícios antigos, existe em grande quantidade em formato textual. No trabalho apresentado, damos o enfoque à extração do texto formal para um formato intermédio (XML) e, posterior conversão para o formato CityGML [Kolbe05]. Esta tecnologia é aplicada na modelação das igrejas do Porto.

2. TRABALHO RELACIONADO

Este trabalho segue uma linha de investigação relacionada com a extração de informação para edifícios monumentais [Rodrigues10]. A automatização do processo de modelação de edifícios baseia-se nas ferramentas automáticas de tradução e geração de resumos [Baeza04]. A utilização do NooJ neste trabalho revelou-se importante pelas seguintes características: multilingue, análise gramatical, sintática e morfológica de múltiplos textos em vários formatos, inserção de anotações ao texto e aplicação de gramáticas personalizadas [Silberztein05].

Para a visualização dos modelos 3D gerados, utilizou-se o CityGML, [Kolbe05].

3. ARQUITETURA DO SISTEMA

O sistema proposto designa-se por MEXEM (Modelação EXpedita de Edifícios – Monumentais) e é composto por três blocos: modelo de dados, extração e modelação3D.

3.1 Modelo de Dados

Entende-se por edifícios monumentais as catedrais, palácios e castelos cujas estruturas são complexas, precisando dum elevado nível de detalhe quando modelados.

A Figura 1 apresenta uma representação simplificada do modelo de dados desenvolvido, que faz corresponder os nós do ficheiro XML com os atributos das classes criadas para a criação do modelo em cityGML.

3.2 Extração de Informação com NooJ

A extração de informação tem como ponto de partida a extração de palavras-chave ou padrões de informação do ficheiro de texto. Para garantir a interoperacionalidade entre os vários módulos assume-se que o formato XML é o garante necessário, obedecendo às especificações do *schema* que suporta o modelo de dados.

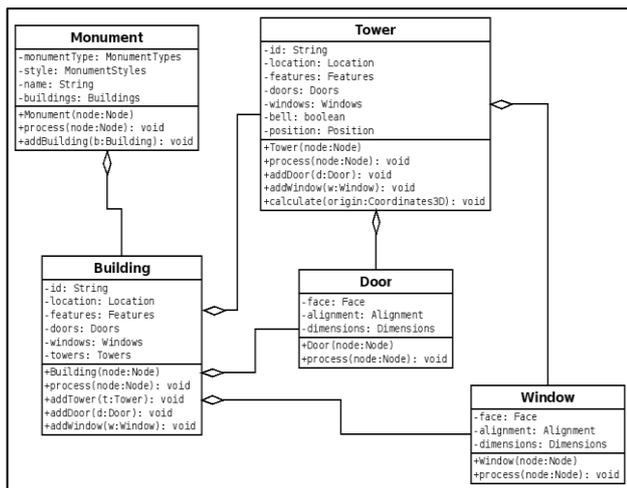


Figura 1: Modelo de dados para edifício monumental

Nas cerca de 109 gramáticas criadas, são definidos padrões conforme a sequência de texto pretendida e adicionadas anotações para enquadrar a informação extraída. Na Figura 2 apresenta-se um exemplo de gramática, com os padrões e anotações que se pretende no output.

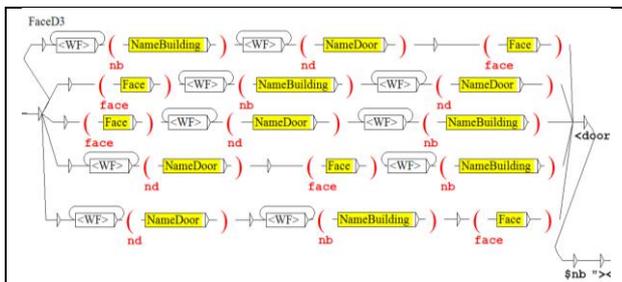


Figura 2: Exemplo de mapeamento da gramática

A partir da sintaxe própria da ferramenta conseguem-se inserir as anotações em formato de tags para delimitar a informação de forma coerente com o modelo de dados.

3.3 Modelação 3D em CityGML

A escolha do CityGML para converter para o modelo 3D deveu-se à possibilidade de combinação da informação semântica com a geométrica, os diversos tipos de detalhe e possuir uma biblioteca de criação de ficheiros gml.

O ficheiro XML é convertido para um modelo interno, baseado na ontologia, através de uma biblioteca gratuita fornecida pelo CityGML, cityGML4j.

4. DESENVOLVIMENTO

O desenvolvimento do protótipo Web teve como suporte a linguagem C# através do Visual Studio 2010. Com a utilização do Web Service, conseguiu-se que a extração de informação (Nooj) e a modelação 3D (cityGML4j) sejam independentes da plataforma, para potenciar outras utilizações e desenvolvimentos futuros.

5. RESULTADOS

Para os testes foi utilizado um conjunto de igrejas da zona do Porto, cujos textos estavam disponíveis em http://pt.wikipedia.org/wiki/Categoria:Igrejas_do_Porto.

Apresentamos na figura 3, o resultado da extração da informação relativa à igreja do Bonfim, onde a comparação com a imagem real permite verificar uma grande aproximação visual ao monumento arquitetónico.



Figura 3: Modelo 3D e imagem real da igreja do Bonfim

O resultado obtido com as 30 igrejas ficou um pouco aquém do esperado, pois só após a sua reescrita foi possível representar 11 delas com grande proximidade e realismo. Este facto deveu-se ao não alinhamento dos textos com os padrões das gramáticas, bem como a ausência de detalhe suficiente para o modelo de dados.

6. CONCLUSÕES

O objetivo de extrair informação de uma descrição textual para obter a representação 3D foi conseguido para diversas igrejas, bem como confirmada a adequação do modelo de dados à modelação pretendida a partir das gramáticas do NooJ. Desta forma, tornou-se possível a criação de modelos 3D com um nível aceitável de fidelidade visual. Também se constatou a dificuldade inerente à extração de informação quando as descrições não foram escritas com orientação às gramáticas personalizadas.

Como trabalho futuro, propomos a extensão da extração de informação para outros tipos de edifícios arquitetónicos e a apresentação de um exemplo do formato textual mais ajustado às gramáticas utilizadas. Pretende-se também promover a edição interativa de texto, bem como colmatar algumas lacunas de informação relativa às dimensões dos edifícios, integrando o sistema com outras fontes de dados geográficos (ex. GIS).

7. AGRADECIMENTOS

Este trabalho é financiado por Fundos FEDER através do Programa Operacional Fatores de Competitividade – COMPETE e por Fundos Nacionais através da FCT – Fundação para a Ciência e a Tecnologia no âmbito do projeto ERAS - Expeditious Reconstruction of Virtual Cultural Heritage Sites (PTDC/EIA-EIA/114868/2009).

8. REFERÊNCIAS

[Baeza04] Ricardo, Baeza-Yates. Challenges in the Interaction of Information Retrieval and Natural Language Processing. *5th Int. Conf. on Computational Linguistics and Intelligent Text Processing*, 2004.

[Kolbe05] T. H. Kolbe, Gröger G., et al. CityGML - Interoperable Access to 3D City Models. *Proc. Int. Symp. on Geoinformation for Disaster Management*, 2005.

[Rodrigues10] Roberto Rodrigues, António Coelho e Luís Paulo Reis. Procedural modelling of monumental buildings from textual descriptions. In *2010 IEEE Cong. on Evolutionary Computation (CEC)*, Barcelona, 18-23 July, pp. 1-8, 2010.

[Silberstein05] M. Silberstein. NooJ: A Linguistic Annotation System For Corpus Processing. *HLT/EMNLP on Interactive Demonstrations*, Stroudsburg, USA, 2005.

An Intuitive Navigation System: Augmented Reality on the Car Windshield

Waldir Pimenta
Universidade do Minho
Braga
wpimenta@di.uminho.pt

Resumo

Sistemas de assistência à condução e navegação em automóveis tornaram-se ubíquos nos últimos anos. No entanto, o modo de interacção mais comum destes sistemas foi avaliado por diversos estudos como tendo um considerável potencial de distração e de risco da segurança.

Modelos alternativos de interacção têm sido apresentados na academia e na indústria, melhorando aspectos de atenção e de usabilidade, mas continuam a apresentar insuficiências que podem prejudicar a sua eficácia e segurança. Este projecto avalia uma potencial abordagem a dispositivos de assistência a navegação, baseada num head-up display (HUD) projectado no pára-brisas do carro, associado a uma componente de realidade aumentada (AR) para apresentar informação de navegação directamente sobreposta à rede viária.

Abstract

In recent years, driving assistance devices have become commonplace both as separate instruments and as built-in components of car equipment. However, the most widespread model of interaction for these devices has been proved to cause distraction and potential safety hazards.

Alternative devices have been proposed and even deployed in commercial settings, improving some of these security and usability aspects, but still present some drawbacks. This project aims to study the feasibility of a navigation assistance system that improves over these problems, using a windshield-based integrated navigation guidance head-up display (HUD), which presents routing information directly superimposed to the road, in the style of Augmented-Reality (AR) applications.

Keywords

windshield display, augmented reality, route guidance, head-up display, driving assistance.

1. INTRODUCTION

Driving assistance systems can generally be categorized as Head-Up Displays (HUD) and Head-Down Displays (HDD). HDDs are by far the most common modes of display, encompassing instrument panel (dashboard) indicators, and side-displays such as portable GPS devices. However, while their advantages are well established through controlled evaluations of driving performance and validated by commercial adoption, a body of research has revealed that they are more likely to create safety hazards due to the loss of visual contact with the road, the cognitive processing delay caused by the context switch and delayed reaction times to critical warnings [Steinfeld95, Green93].

Audio feedback has become common in the form of both spoken routing instructions, and warning sounds for various potential dangers such as high speed, proximity of obstacles, etc. This alleviates the aforementioned problem, but doesn't entirely solve it, given the human dependency on visual input for situational awareness.

This research project focuses on Augmented Reality (AR) displayed onto windshield-based HUDs, superimposing the informational elements directly into the driver's field of view. HUDs have been proven to be more efficient and secure than HDDs, since they don't require the driver to shift their gaze away from the road. The main cited disadvantages of HUDs are the cost and the potential for visual clutter distracting the driver, but focused research should inform the properties that define a non-intrusive implementation of a HUD-based navigation assistance system.

Several projects (and even commercial systems) have implemented the display of iconographic or textual information on the windshield [Doshi09]. However, since the presentation of floating virtual elements directly superimposed on the scene can distract the driver, the current project targets integrated displays that fuse the virtual elements into the real scene, thus implementing an AR system. The specific application studied is navigation

assistance through display of the planned route directly over the driver's view of the road.

2. STATE OF THE ART

While HUDs are less common than HDDs in driving assistance systems, HUDs that present turn-by-turn navigation assistance (as opposed to icons or text) are even rarer. BMW presented such a system in 2011, but makes little attempt to visually integrate the routing icons into the real scene.

On the other hand, AR approaches have also surfaced in the industry, such as in the systems commercialized by Blaupunkt (2008) and Wikitude (2009). These are, however, HDDs: essentially, portable GPS devices with an integrated camera that displays live video from the road and superimposes a virtual route on top of it. Blaupunkt's system doesn't track the road and relies on the device being positioned in a central location so that the arrows will generally fit in. The Wikitude system attempts to do some visual tracking, but the lack of real 3D information makes the route float above and sometimes slightly off the actual road.

Finally, more recent systems have attempted to include both techniques (AR and HUD). To the author's knowledge, only two such commercial systems exist: MVS's Virtual Cable¹ (2011), and Pioneer's Cyber Navi² (2012). The AR component of these systems, however, is based on similar principles as the Blaupunkt system mentioned above: they rely on the imprecise GPS positioning and use a "virtual cable" metaphor to avoid having to follow the roads accurately.

3. SYSTEM OVERVIEW

The hypothesis to be evaluated by this project is that precise positioning of the virtual elements is crucial to reduce decision errors and delay, especially in complex intersections.

As such, one of the main components that needs to be implemented is a system for detecting the position and orientation of the car with high precision and responsiveness, so that the virtual overlay covers the correct places of the real world.

Secondly, such high-precision positioning of the virtual elements will need to adapt to the driver's position, so that the perspective is always correct. This will provide motion parallax, which is one of the most important features of visual spatial perception, especially at the medium-to-long distances usually involved in driving landscape. In windshield-based HUDs, motion parallax can be implemented either by head tracking [Doshi09], or by applying autostereoscopic properties to the windshield, such as lenticular screen technology [Takaki11], to convert it into a multi-view display.

Also, elements of 3D vision must be taken into account to produce an effective HUD. If the driver is looking at the road, their eyes will converge at a distance, and any ele-

ment located closer to the eye will appear blurred (unfocused) and in double vision. To merge these two images, the eyes must converge and focus at the windshield. This type of focus shift is undesirable and causes similar problems to HDDs. If a stereoscopic pair of images is displayed (each perspective properly directed at the appropriate eye) then the eyes can converge in the road without seeing a double image [Doshi09]. This needs to be addressed by a system that allows projecting different images to each eye, which drives up the cost and complexity of the system.

The proposal advanced by this project is that a workaround for this stereoscopy requirement could involve projecting the route not as a continuous line, but as a series of horizontal strips. Since eye separation is mostly horizontal, no vertical unalignment should be visible; and since no continuous vertical border exists, any of these horizontal lines, when seen in double due to the driver focusing on the road, will fuse easily into a single line, somewhat longer than the original and fading out at the ends.

Such a system can be implemented using off-the-shelf components, namely:

1. A standard **GPS receiver**, to obtain global positioning information and allow the calculation of the routing.
2. An **outer camera** to recognize salient features of the environment and make adjustments to the 3D model of the surroundings;
3. A **projector + mirror** setup to display the virtual elements on the windshield, similar to the system implemented in [Sato06].
4. A **processing unit** would naturally be included to perform the calculations and control the system
5. An **inner camera**, which will be mounted on the ceiling of the car, to track the head of the driver and adjust the perspective of the projected elements to produce correct head motion parallax.

This system is very similar to [Doshi09], who used a laser-based display, and [Sato06], who used, as mentioned above a projector + mirror setup. However, neither attempted to provide the AR environment, instead focusing on the display of informational icons.

4. REFERENCES

- [Steinfeld95] A. Steinfeld and P. Green. *U.Mich. Transportation Research Inst. Tech Report* (1995).
- [Green93] P. Green, M. Williams, E. Hoekstra, K. George and C. Wen. *Tech.rep. UMTRI-93-32* (1993).
- [Doshi09] A. Doshi, S.Y. Cheng and M.M. Trivedi. *IEEE Trans. on Systems, Man and Cybernetics, Part B: Cybernetics* (2009), pp. 85-93.
- [Takaki11] Y. Takaki, Y. Urano, S. Kashiwada, H. Ando and K. Nakamura. *Optics Express*, 19(2).
- [Sato06] A. Sato, I. Kitahara, Y. Kameda and Y. Ohta. *Proc. 13th World Congress on Intelligent Transport Systems* (2006)

¹ <http://www.mvs.net/technology.html>

² <http://pioneer.jp/press-e/2012/pdf/0511-1.pdf>

Índice de autores

- Adão, Telmo 89
Afonso, Ana Paula 73
Almeida, Luís 115
- Baptista, Ricardo 133
Bessa, Maximino 89
Birra, Fernando 19, 27
- Campos, Carlos 51
Cardoso, Igor Duarte 81
Cardoso, Nuno 129
Carmo, Maria Beatriz 43, 73
Castro, António 11
Cláudio, Ana Paula 43, 73
Coelho, António 51, 89, 111, 133
Correia, Nuno 103
Costa, Abílio 3
Costa, Raquel 59
Costa, Vasco 67
- Dias, Paulo 81
Dias, Ricardo 107
Domingues, Hugo 115
- Esteves, Francisco 43
- Felgueiras, Gilberto 131
Ferreira, Alfredo 59
Ferreira, António 73
Fonseca, João 131
Fonseca, Manuel J. 59, 107
- Gomes, Abel 99
Gonçalves, Alexandrino 89
Gonçalves, Daniel 107
Gonçalves, Ricardo 111
- Iurgel, Ido 119
- Jorge, Joaquim A. 67
- Leitão, João Miguel 51
Leitão, Pedro 11
Lopes, João Manuel Brisson 127
- Magalhães, Luís 89, 133
Maçães, Gustavo 115
Medeiros, Pedro 27
Montez, Edgar 73
Moreira, Pedro Miguel 125, 129, 131
Moura, João 89
Mourato, Fausto 19
- Nóbrega, Rui 103
Nunes, João Ferreira 125
- Pereira, Francisco 89
Pereira, João M. 67
Pereira, João P. J. 3, 11
Pimenta, Waldir 135
Pinheiro, Tânia 43
Preto, Bruno 27
- Reis, Luís Paulo 89, 133
Ribeiro, Madalena 99
Ribeiro, Pedro 119
Rocha, Fábio 129
Rodrigues, Nuno 89
Rodrigues, Roberto 89, 133
Rodrigues, Rui 111
- Santos, Beatriz Sousa 81
Santos, Luís Paulo 35, 115
Santos, Manuel Próspero dos 19
Silva, Nuno 35
Simplicio, Raúl 73
Sousa, Augusto 89, 133
- Tavares, João Manuel R. S. 125

