

Universidade de Trás-os-Montes e Alto Douro

Técnicas de Tracking sem Marcas em Aplicações de Realidade Mista

– Versão Final –

Dissertação de Mestrado em Engenharia Informática

Por

David Gonçalves Narciso

Orientador: Doutor Luís Gonzaga Mendes Magalhães

Coorientador: Doutor Emanuel Soares Peres Correia



Vila Real, 2015

Universidade de Trás-os-Montes e Alto Douro

**Técnicas de Tracking sem Marcas em Aplicações de
Realidade Mista**

– Versão Final –

Dissertação de Mestrado em Engenharia Informática

Por

David Gonçalves Narciso

Orientador: Doutor Luís Gonzaga Mendes Magalhães

Coorientador: Doutor Emanuel Soares Peres Correia

Composição do Júri

Presidente:

Doutor Ramiro Manuel Ramos Moreira Gonçalves

Vogais:

Doutor António Fernando Vasconcelos Cunha Castro Coelho

Doutor Emanuel Soares Peres Correia

"A persistência é o caminho do êxito"

Charles Chaplin

Aos meus pais e irmãos.

Agradecimentos

À *mui* nobre Universidade de Trás-os-Montes e Alto Douro, pela disponibilização de todas as condições para a minha formação académica.

Aos meus orientadores, Professor Doutor Luís Gonzaga Mendes Magalhães e Professor Doutor Emanuel Soares Peres Correira, pela excelente orientação, sugestões e correções, disponibilidade, confiança e apoio prestado no desenvolvimento desta dissertação.

A todos os colegas da academia, pela amizade, disponibilidade, simpatia e espírito de entreajuda. De entre outros, um agradecimento especial à Carolina Simões, Cristiana Simões, Luís Pádua, Luís Costa, Luís Lobo, João Gonçalves, Jorge Pereira, Inês Domingues, David Ramalho e Márcia Guerra.

Aos colegas do *open space*, Telmo Adão, Miguel Melo, Jorge Freitas, César Meira, Tissiana Magalhães, Filipe Marques e João Barreira, pela sua companhia, boa disposição e ajuda no trabalho e na descontração.

Aos amigos escalabitanos de longa data, José Marcos, Rui Marcelino, Tiago Rodrigues, Eduardo Massano, João Massano e Gonçalo Reis, pelas suas amizades, apoio e compreensão ao longo destes cinco anos de percurso académico.

Aos meus pais, pelo carinho e confiança que depositaram em mim ao longo deste percurso e pela oportunidade de vida que me proporcionaram. Aos meus irmãos, pelo apoio e motivação ao longo desta etapa.

Por fim, a todos os que, direta ou indiretamente, contribuíram para a realização desta dissertação.

A todos, o meu muito obrigado!

David Narciso,

UTAD, Vila Real

2 de Setembro de 2015

RESUMO

A Realidade Mista (RM) tem como objetivo fornecer ao utilizador uma percepção melhorada da realidade. Para que tal aconteça, o conteúdo virtual é sobreposto numa vista do mundo real, ou vice-versa, dependendo do tipo de conteúdo que é predominante. Assim, designa-se por Realidade Aumentada (RA), quando existe uma maior predominância do conteúdo real, ou por Virtualidade Aumentada (VA), quando há uma maior predominância do conteúdo virtual.

O trabalho desenvolvido nesta dissertação insere-se no contexto do projeto de investigação e desenvolvimento (I&D) Mixed Augmented Reality (MixAR), o qual visou o desenvolvimento de um sistema de RM adaptativa para sítios arqueológicos. Este sistema tem como objetivo principal proporcionar aos seus utilizadores a observação do esplendor inicial dos edifícios que outrora existiram nos sítios arqueológicos, recorrendo para isso a técnicas de RM para a visualização de reconstruções virtuais, historicamente corretas, sobrepostas nas estruturas/ruínas atuais, caso subsistam, ou no local onde outrora existiram.

Esta dissertação propõe e implementa uma metodologia de *tracking* adequada aos requisitos definidos para o projeto MixAR. Desde o início, foi definido que seria usada uma técnica de *tracking* baseada em imagens e sem marcas, de modo a evitar a colocação de elementos intrusivos nos sítios arqueológicos. Tendo por base os requisitos definidos para o projeto e a plataforma de desenvolvimento selecionada, foi efetuada uma análise e comparação das diversas técnicas de *tracking* sem marcas, disponibilizadas por essa plataforma. Com base nesse estudo foram selecionadas e adaptadas algumas dessas técnicas de *tracking*. O trabalho aqui desenvolvido foi integrado na aplicação que é executada na unidade móvel, designada MixAR App. Face às alterações efetuadas ao longo do desenvolvimento da metodologia de *tracking*, foram produzidas duas versões desta aplicação (V1.0 e V2.0). A aplicação e a correspondente metodologia de *tracking* implementada foram testadas e avaliadas. Na primeira versão da aplicação foram apenas efetuados testes de desempenho. Já na segunda versão, além de testes de desempenho, foram efetuados testes com utilizadores. Uma análise dos resultados dos testes de desempenho e dos questionários (preenchidos pelos utilizadores após a realização dos testes) levou à conclusão de que as unidades móveis imersivas obtiveram, de uma forma consistente, melhores resultados que as unidades móveis não imersivas relativamente ao desempenho e ao sucesso na realização das tarefas relacionadas com o *tracking*.

Palavras-chave – Realidade Mista; Realidade Aumentada; Virtualidade Aumentada; Técnicas de *Tracking*; Metaio SDK; Unity3D.

ABSTRACT

Mixed Reality (MR) aims to provide the user with an improved perception of reality. For this to happen, the virtual content is superimposed on a view of the real world, or vice versa, depending on the type of content that is predominant. Therefore, it is called Augmented Reality (AR), when there is a predominance of the real content, or Augmented Virtuality (AV), when there is a greater prevalence of the virtual content.

The work developed in this dissertation inserts in the context of the project of research and development Mixed Augmented Reality (MixAR), which aimed at developing an adaptive MR system for archeological sites. The main goal of this system is to provide its users the observation of the initial splendor of buildings that once existed on archeological sites, relying on MR techniques to visualize virtual reconstructions, historically correct, overlapped in their current structures/ruins, in case they exist, or where they once existed.

This dissertation proposes and implements a tracking methodology appropriate for the requirements set for the MixAR project. From the beginning, it was defined that it would be used a tracking technique based on images and markerless, in order to avoid placing intrusive elements in archeological sites. Based upon the requirements set for the project and the selected development platform, an analysis and comparison of the different markerless tracking techniques available for this platform was made. This study resulted in the selection and adaptation of some of these tracking techniques. The work developed here has been integrated in the application running on the mobile unit, called Mix App. Given the changes made throughout the development of the tracking methodology, two versions of this application were produced (V1.0 and V2.0). The application and the corresponding tracking methodology implemented were tested and evaluated. In the first version of the application only performance tests were carried out. In the second version, as well as performance testing, tests with users were performed. An analysis of the results of performance tests and questionnaires (filled by the users after the tests) led to the conclusion that the immersive mobile units obtained, in a consistent way, better results than the non-immersive mobile units regarding the performance and success in carrying out the tasks related to the tracking.

Keywords – Mixed Reality; Augmented Reality; Augmented Virtuality; Tracking techniques;

Metaio SDK; Unity3D.

Índice

Agradecimentos	iii
RESUMO	v
ABSTRACT	vii
Índice de tabelas	xi
Índice de figuras.....	xiii
Glossário, acrónimos e abreviaturas	xvii
1 Introdução.....	1
2 Estado da Arte	7
2.1 Realidade Virtual	7
2.2 Realidade Mista	11
2.2.1 Realidade Aumentada	12
2.2.2 Virtualidade Aumentada	15
2.3 O Processo de Tracking na Realidade Mista: Conceitos Base.....	17
2.3.1 Detecção do Ambiente Real e do Movimento do Utilizador.....	18
2.3.2 Alinhamento entre Objetos Reais e Virtuais.....	21
2.4 Técnicas de Tracking na Realidade Mista.....	23
2.4.1 Baseadas em Sensores	24
2.4.2 Baseadas na Visão	25
2.4.3 Híbridas	34
2.5 Rendering	35
2.6 Modelação.....	38
2.7 SDK de RM.....	38
2.8 Aplicações de RM.....	42
3 Projeto MixAR	45
3.1 Enquadramento	45
3.2 Arquitetura e Funcionamento	46
3.2.1 Servidor de RM.....	47
3.2.2 Infraestrutura de Rede.....	49
3.2.3 Unidade Móvel.....	49

3.3	Seleção de Equipamento e Ferramentas de Desenvolvimento.....	50
3.4	Técnicas de Tracking Disponíveis no Metaio SDK	51
3.4.1	ID Marker.....	52
3.4.2	Picture Marker.....	52
3.4.3	Image Tracking.....	53
3.4.4	3D Map Tracking	54
3.4.5	Instant Tracking.....	54
3.4.6	Extended Image Tracking.....	55
3.4.7	Face Tracking.....	55
3.4.8	Latitude Longitude Altitude (LLA) Markers.....	55
3.4.9	CAD Model Tracking	56
3.4.10	GPS/sensores inerciais.....	57
3.5	Tracking no Metaio SDK + Unity3D	57
4	Metodologia de Tracking Proposta.....	63
4.1	Metodologia de Tracking: Proposta Inicial	63
4.2	MixAR App - Versão 1.0	68
4.3	Metodologia de Tracking: Proposta Final.....	71
4.4	MixAR App - Versão 2.0	72
5	Resultados e Discussão	77
5.1	Processo de Criação de Experiências de RM	77
5.2	MixAR App (Versão 1.0)	82
5.2.1	Testes de Desempenho	82
5.2.2	Resultados e discussão	84
5.3	MixAR App (Versão 2.0)	86
5.3.1	Testes de desempenho.....	88
5.3.2	Resultados e discussão	90
5.3.3	Testes com utilizadores.....	91
5.3.4	Resultados e discussão	96
6	Conclusões e Trabalho Futuro.....	103
	Referências Bibliográficas.....	106
	Anexos	111

Índice de tabelas

Tabela 1 - Comparação entre os diferentes métodos de <i>tracking</i> . Adaptado de Lima et al. (2010).....	32
Tabela 2 - Resumo das características principais entre os vários SDK apresentados.	42
Tabela 3 - Diferentes modos de execução suportados pela técnica Instant Tracking. Descrição das alterações efetuadas à técnica de <i>tracking</i> entre os diferentes modos de execução apresentados.	54
Tabela 4 - Propriedades dos diferentes modelos utilizados para aumentar as estruturas do mundo real.	83
Tabela 5 - Resultados dos testes da primeira versão do MixAR.	84
Tabela 6 - Propriedades dos modelos de ajuda visual, arestas e superfícies utilizados para efetuar o <i>tracking</i> a cada estrutura. O modelo de linhas não possui faces, daí a inexistência de tal coluna.	85
Tabela 7 - Características dos dispositivos que constituem cada configuração.	89
Tabela 8 - Resultados dos testes de desempenho efetuados à segunda versão da MixAR App.....	90

Índice de figuras

Figura 1 - Oculus Rift, um exemplo de um HMD do tipo <i>closed-view</i> . Imagem obtida de: http://www.techcult.com.br/wp-content/uploads/2015/05/oculus-rift.jpg	10
Figura 2 - Representação do contínuo de virtualidade, adaptado de Milgram and Kishino (1994).	11
Figura 3 - HMD <i>optical see-through</i> Vuzix STAR 1200XLD. A câmara presente no HMD é utilizada para técnicas de <i>tracking</i> baseadas na visão e não para fornecer ao utilizador uma visualização do mundo real. Esta é obtida diretamente pelo utilizador devido à transparência das lentes. Imagem obtida de: http://www.vuzix.com/augmented-reality/products_star1200xld/	12
Figura 4 - HMD <i>video see-through</i> Vuzix Wrap 1200DXAR. Neste caso, a imagem do mundo real é obtida utilizando uma ou duas câmaras e exibida nos <i>displays</i> embutidos. Imagem obtida de: http://www.vuzix.com/augmented-reality/products_wrap1200dxar/	12
Figura 5 - "The Sword of Damocles", o primeiro sistema de RA (Sutherland, 1968).	14
Figura 6 - Representação gráfica do problema do registo. Ao invés de a seta virtual ter sido apresentada em cima do parafuso indicado, foi apresentada próxima de um outro parafuso, possibilitando a realização de uma tarefa errada por parte do utilizador. Adaptado de Bajura and Neumann (1995).	21
Figura 7 - Sistema de RA <i>open-loop</i> , adaptado de Bajura and Neumann (1995).	22
Figura 8 - Sistema de RA <i>closed-loop</i> , adaptado de Bajura and Neumann (1995).	23
Figura 9 - (Esq.) Apresentação de diversas <i>tiles</i> com as devidas marcas fiduciais retangulares. (Dir.) Visualização de conteúdo virtual corretamente aumentado sobre algumas <i>tiles</i> (Poupyrev et al., 2001).	28
Figura 10 - Marcas fiduciais constituídas por vários anéis, adaptado de Cho and Neumann (2001).	28
Figura 11 - Taxonomia de técnicas de <i>tracking</i> sem marcas para aplicações de RA. Adaptado de Teichrieb et al. (2007).	29
Figura 12 - Técnicas de <i>tracking</i> baseadas em modelos. Imagem obtida de Lima et al. (2010).	31
Figura 13 - Ilustração das duas técnicas de <i>rendering</i> : foto realista e não foto realista (Finkelstein, 2005).	36
Figura 14 - Imagem foto realista renderizada através do <i>renderer</i> "LuxRender". Imagem obtida de: http://www.luxrender.net/forum/gallery2.php?g2_itemId=12426	37
Figura 15 - Aumento de um objeto através de uma marca fiducial utilizando o ARToolKit. Imagem obtida de: http://www.hitl.washington.edu/artoolkit/images/nakaohome.jpg	39
Figura 16 - Arquitetura geral do projeto MixAR. Descrição dos módulos funcionais presentes na arquitetura: unidade móvel, servidor de RM e infraestrutura de rede.	47
Figura 17 - Apresentação das diversas tabelas, e as suas propriedades, que formam o modelo de dados que suporta o SIG, desenvolvido no âmbito do projeto MixAR.....	48
Figura 18 - Diagrama de fluxo da interação entre os componentes que constituem a unidade móvel. Apresentação das tarefas executadas em cada componente da unidade móvel no âmbito do processo de deteção (<i>tracking</i>) e aumento das estruturas do mundo real.	50
Figura 19 - Marca fiducial como número de identificação 1 pré-definida no Metaio SDK.....	52
Figura 20 - Exemplo de uma imagem indicada para ser utilizada como marca quando se recorre à técnica de <i>tracking</i> Picture Marker. Imagem obtida de: http://dev.metaio.com/fileadmin/user_upload/documents/sdk/picturemarker.pdf	53
Figura 21 - Exemplo de uma imagem indicada para ser utilizada como marca quando se recorre à técnica Image Tracking. Imagem obtida de: http://dev.metaio.com/fileadmin/user_upload/documents/sdk/markerless.pdf	53
Figura 22 - Exemplo de uma aplicação que utiliza a técnica de <i>tracking</i> LLA markers. Imagem obtida de: https://dev.metaio.com/typo3temp/pics/4d59ef54f2.jpg	56
Figura 23 - Árvore de decisão para auxiliar os utilizadores a escolher a configuração mais adequada para a técnica CAD Model Tracking do Metaio SDK. Imagem obtida de: http://dev.metaio.com/typo3temp/pics/52168efcc0.png	57

Figura 24 - Instância de um <i>GameObject</i> criado através do <i>prefab</i> metaioSDK. À direita, na vista "Inspector", são visíveis os componentes, juntamente com as suas propriedades, que constituem o <i>GameObject</i> metaioSDK, nomeadamente os <i>scripts</i> "Metaio SDK" e "Metaio Callback".	58
Figura 25 - A figura apresenta os componentes e propriedades do <i>GameObject</i> selecionado. À direita, através da vista "Inspector", pode-se constatar que o sistema de coordenadas (Coordinate System ID) selecionado é o "1". No centro da figura é apresentado o modelo virtual (<i>GameObject</i> "Modelo para aumento"), que irá aumentar a cena após um <i>tracking</i> bem sucedido.	60
Figura 26 - A figura apresenta os componentes e propriedades do <i>GameObject</i> selecionado. À direita, através da vista "Inspector", pode-se constatar que o sistema de coordenadas (Coordinate System ID) selecionado é o "2". No centro da figura é apresentado o modelo virtual (<i>GameObject</i> "Modelo de ajuda visual ") que irá ser utilizado para auxiliar o utilizador a encontrar a pose inicial definida no ficheiro de configuração.	61
Figura 27 - Diagrama de fluxo da MixAR App. Descrição do processo de atualização constante que a aplicação executa de forma a fornecer informação contextualizada à unidade móvel.	64
Figura 28 - Ilustração de um modelo de arestas, superfície e ajuda visual.	66
Figura 29 - À esquerda a imagem apresenta o modelo de ajuda visual alinhado, pelo utilizador, com as arestas da estrutura do mundo real. À direita, é apresentado o modelo virtual, que aumenta a estrutura após a uma correspondência das arestas com sucesso.	67
Figura 30 - Diagrama de fluxo da técnica "Hybrid CAD Model Tracking", utilizada na versão 1.0 da MixAR App.	67
Figura 31 - <i>Screenshot</i> tirado pela unidade móvel na execução da primeira versão da MixAR App. Nesta pode-se constatar a visualização de um modelo virtual após o <i>tracking</i> de uma estrutura do mundo real.	70
Figura 32 - <i>Screenshot</i> tirado pelo <i>tablet</i> da interface gráfica na segunda versão do MixAR.	74
Figura 33 - <i>Screenshot</i> tirado pelo <i>smartphone</i> durante a execução do MixAR para configurações imersivas com apenas um ecrã.	75
Figura 34 - Imagem no canto superior direito fornece ao utilizador uma ajuda visual da pose inicial pretendida sendo utilizado neste caso a técnica "3D Map Tracking".	76
Figura 35 - Exemplo de uma estrutura do mundo real seguida do modelo virtual completo da mesma.	78
Figura 36 - Instruções fornecidas pela aplicação Metaio Toolbox para a criação de um mapa 3D. De uma forma contínua o utilizador move o dispositivo em redor da cena que pretende adquirir um mapa 3D, quando o utilizador entender que o mapa tem pontos 3D suficientes, para o processo de aprendizagem e armazena o mapa 3D.	80
Figura 37 - Interface gráfica para a configuração das técnicas "Image Tracking" e "Extended Image Tracking". Fornece a possibilidade de ativar ou desativar o <i>extended tracking</i> .	81
Figura 38 - Estruturas utilizadas nos testes da primeira versão da MixAR App. Da esquerda para a direita: contentor, entrada para o polo da Escola de Ciências e Tecnologia, entrada para a Biblioteca Central.	82
Figura 39 - As diferentes estruturas aumentadas com os modelos virtuais definidos na configuração de <i>tracking</i> .	85
Figura 40 - Vista aérea das estruturas do mundo real utilizadas para a realização de testes com a versão 2.0 da MixAR App. No centro da imagem encontra-se o museu da Vila Velha, ao seu redor são destacadas as estruturas utilizadas para as experiências, nomeadamente: a capela (A), as ruínas (B) e o descampado (C).	86
Figura 41 - Introdução do botão "Switch" para simular a posição do utilizador para a coordenada central das estruturas alternando assim entre configurações de <i>tracking</i> .	88
Figura 42 - Óculos de RM Neje Colorcross, <i>smartphone</i> BQ Aquaris E5 HD e <i>tablet</i> Samsung SM-T320. Imagens obtidas de: http://img.dxcdn.com/productimages/sku_332676_3.jpg , http://innovainformatica.com/images_url_amigables/1520/smartphone-bq-aquaris-e5-hd-ips-qc-16g-a4-4-b-n-c000060-grande.jpg e http://images.samsung.com/is/image/samsung/pt_SM-T320NZWATPH_000234416_Front_black?\$TM-Gallery\$, respetivamente.	92
Figura 43 - Estrutura A (capela).	92
Figura 44 - Modelo de arestas, superfície e ajuda visual criados para a estrutura A.	93

Figura 45 - Screenshot tirado pelo tablet na criação do mapa de pontos 3D para a estrutura B.	94
Figura 46 - <i>Screenshot</i> tirado pelo <i>tablet</i> na criação do mapa de pontos 3D para a estrutura C.	94
Figura 47 - Mapa de pontos 3D da estrutura C após a sua importação para o <i>Metaio Creator</i>	95
Figura 48 - Utilização das configurações imersiva (estrutura B) e não imersiva (estrutura A), respetivamente.	96
Figura 49 - Respostas dos utilizadores para ambas as configurações à pergunta "Foi difícil visualizar os edifícios durante a experiência".	97
Figura 50 - Respostas dos utilizadores para ambas as configurações à pergunta "Os conteúdos visualizados estavam sincronizados com o mundo real".	97
Figura 51 - Respostas dos utilizadores para ambas as configurações à pergunta "Os modelos eram apresentados de uma forma robusta (sem saltos) e fluída".	98
Figura 52 - Respostas dos utilizadores para ambas as configurações à pergunta "Fui capaz de aproximar-me dos edifícios para os observar".	99
Figura 53 - Visualização de um ambiente de virtualidade aumentada onde é apresentado o interior do edifício virtual que aumenta a estrutura A.	100
Figura 54 - Respostas dos utilizadores para ambas as configurações à pergunta "Consegui visualizar os edifícios virtuais de pontos de vista diferentes".	100

Glossário, acrónimos e abreviaturas

Lista de acrónimos

Sigla	Expansão
.NET	<i>.NET Framework</i>
AP	Access Point
API	<i>Aplication Programming Interface</i>
AREL	<i>Augmented Reality Experience Language</i>
ARM	<i>Advanced RISC Machine</i>
BD	Base de dados
CAD	<i>Computer Aided Design</i>
CAVE	<i>Cave Automatic Virtual Environment</i>
CPU	<i>Central Processing Unit</i>
DOF	<i>Degrees of Freedom</i>
FPS	<i>Frames Per Second</i>
GB	<i>Gigabyte</i>
GHz	<i>Gigahertz</i>
GPS	<i>Global Positioning System</i>
HD	<i>High Definition</i>
HMD	<i>Head-Mounted Display</i>
HTML	<i>HyperText Markup Language</i>
IDE	<i>Integrated Development Environment</i>
jQuery	<i>JavaScript Query</i>
JS	<i>JavaScript</i>
JSON	<i>JavaScript Object Notation</i>
LLA	<i>Latitude Longitude Altitude</i>
mAh	Miliamperes hora
MB	<i>Megabyte</i>
MixAR	<i>Mixed Augmented Reality</i>
MP	<i>Megapixel</i>
PHP	<i>PHP:Hypertext Preprocessor</i>
PTAM	<i>Parallel Tracking and Mapping</i>
QR-CODE	<i>Quick Response Code</i>
RA	Realidade Aumentada
RAM	<i>Random Access Memory</i>
RANSAC	<i>RANdom SAmple Consensus</i>
RM	Realidade Mista
RV	Realidade Virtual
SFM	<i>Structure from Motion</i>
SIG	Sistema de Informação Geográfico
SLAM	<i>Simultaneous Location And Mapping</i>
SO	Sistema Operativo
VA	Virtualidade Aumentada
WGS	<i>World Geodetic System</i>
XML	<i>eXtensible Markup Language</i>

Lista de abreviaturas

Abreviatura	Significado(s)
e.g.	Por exemplo
et al.	E outros (autores)
etc.	Etecetera, outros
i.e.	Isto é, por conseguinte

1 Introdução

Este capítulo introdutório pretende familiarizar o leitor com os conceitos tecnológicos nos quais se baseia a dissertação. A motivação por trás deste trabalho é apresentada, juntamente com os seus objetivos. O capítulo termina com uma explicação da organização da dissertação.

O presente trabalho está relacionado com o vasto conceito tecnológico da RM. Para entender o que é verdadeiramente a RM, uma breve descrição das suas origens será aqui apresentada de forma a elucidar o leitor. A primeira tecnologia deste género foi a RV, uma tecnologia cujo objetivo é simular a presença física de um utilizador em lugares do mundo real ou imaginários e permitir ao utilizador a sua interação com esse mundo. Pouco tempo depois, surgiu a RA, que ao invés de imergir o utilizador num mundo completamente sintético, sobrepõe conteúdo virtual em cima de uma vista do mundo real, complementando assim a realidade ao invés de a substituir. Outro conceito relevante, que surgiu na mesma altura e está fortemente relacionado com a RA, é o conceito de VA. Contrariamente à RA, que coloca conteúdo virtual sobre cenas do mundo real, a VA coloca conteúdo real sobre um mundo virtual. Uma maneira de verificar se estamos perante uma aplicação de RA ou VA é verificar qual é o tipo de conteúdo que compreende uma maior proporção da imagem final, onde se considera uma aplicação de RA no caso de haver uma predominância do conteúdo real, ou uma aplicação de VA no caso inverso. Relativamente ao conceito de RM, este consiste numa junção de ambos os conceitos de RA e VA. Em termos práticos, considera-se uma aplicação de RM uma que permita a visualização de conteúdos reais e virtuais com diferentes proporções, possibilitando uma predominância do real ou do virtual dependendo da situação. A RM, bem como os conceitos que esta engloba, possuem o mesmo objetivo: fornecer ao utilizador uma percepção melhorada da realidade.

Apesar do uso generalizado do termo RA na literatura, para descrever aplicações que combinem imagens do mundo real com conteúdo virtual, ao longo da dissertação frequentemente é utilizado o termo RM em detrimento do termo RA devido à sua generalidade, uma vez que este abrange ambos os termos RA e VA.

O presente trabalho possui uma forte ligação ao projeto de investigação e desenvolvimento MixAR. O MixAR é um projeto que visa o desenvolvimento de um sistema de RM adaptativa para sítios arqueológicos que pretende permitir, aos seus utilizadores, a observação do esplendor inicial de antigos edifícios presentes em sítios arqueológicos, através da visualização de reconstruções virtuais, historicamente corretas, sobrepostas nas suas estruturas/ruínas atuais, caso subsistam, ou no local onde outrora existiram.

O sistema de RM adaptativa pretende variar a mistura entre os conteúdos real e virtual consoante o tipo de cena, interior ou exterior, posição do utilizador e estado da estrutura onde será efetuado o *tracking*. Quando o utilizador se encontra numa cena exterior é utilizada uma aproximação de RA, onde o mundo real prevalece. Se o utilizador se mover para o interior de um edifício ou uma zona que seja considerada interior (devido à existência de uma antiga estrutura no local), é utilizada uma aproximação de VA, onde o mundo virtual prevalece. A VA é utilizada em cenas interiores pois esta, contrariamente à RA, tem controlo sobre as condições de iluminação, tendo assim possibilidade de criar um ambiente mais realista em interiores. Quanto ao fornecimento de informação contextualizada, o sistema irá fornecer ao utilizador informação utilizando como contexto a sua posição. A posição do utilizador será obtida pelo sistema através um sensor de Global Positioning System (GPS). O sistema irá tratar de calcular quais as estruturas nas proximidades do utilizador e devolver ao utilizador as informações necessárias (e.g. modelos virtuais que irão aumentar as estruturas) para que este consiga visualizar as reconstruções virtuais das estruturas do mundo real.

Para executar as tarefas propostas, o projeto MixAR segue uma arquitetura cliente-servidor, sendo os seus componentes principais: uma unidade móvel, um servidor de RM e uma infraestrutura de rede. A unidade móvel será utilizada para proporcionar aos seus utilizadores a experiência de RM. Esta será transportada pelos utilizadores durante a visita ao sítio arqueológico. O servidor de RM será utilizado para agir como uma unidade de processamento remoto e armazenar/fornecer às unidades móveis o acesso a conteúdo digital. Por último, a infraestrutura de rede será utilizada para suportar a comunicação entre o servidor e uma ou mais unidades móveis.

Uma das operações cruciais em qualquer sistema de RM é a operação de *tracking*. De uma forma simplificada, esta operação consiste na obtenção da posição e rotação que define o ponto de vista do utilizador em relação a um determinado objeto ou cena. A razão pela qual a

obtenção do ponto de vista do utilizador é crucial é simples. A posição e rotação que o define serão aplicadas numa câmara virtual (utilizada pelo *software* de computação gráfica para visualizar o conteúdo virtual) com vista a obter a mesma perspetiva que o utilizador no mundo real, alinhando ambos os conteúdos real e virtual. Consequentemente, quão maior for a precisão na obtenção do ponto de vista do utilizador mais preciso será o alinhamento entre os conteúdos real e virtual. Um alinhamento preciso assegura que o objeto/cena virtual é alinhado na perfeição com a imagem do mundo real. Se isto não acontecer, o conteúdo virtual irá parecer flutuar sem sentido, afetando a ilusão de que os dois mundos, real e virtual, coexistem no mesmo espaço (R. T. Azuma, 1997). A combinação de ambos os conteúdos, real e virtual, resulta na visualização de uma cena de RM.

O trabalho desenvolvido ao longo da dissertação foi motivado pela necessidade de implementar uma metodologia para realizar a operação de *tracking* no projeto MixAR. A operação de *tracking* pode ser realizada com recurso a diferentes tipos de sensores, também denominados *trackers*. Uma descrição dos *trackers* existentes bem como os seus problemas relacionados são apresentados no subcapítulo 2.3.1. Entre os *trackers* existentes, os óticos (baseados em câmaras) são destacados como os que têm a maior potencialidade de criar soluções de *tracking* precisas e robustas. Como tal, no projeto MixAR optou-se pela utilização de *trackers* óticos para realizar a tarefa de *tracking*. Utilizando este tipo de *tracker*, as imagens dos sítios arqueológicos (obtidas através de uma câmara posicionada na frente dos olhos do utilizador) são processadas por algoritmos de visão por computador de forma a obter uma estimativa da pose da câmara (i.e. a sua posição e rotação) que define o ponto de vista do utilizador em relação a uma determinada estrutura/ruína presente no sítio arqueológico. A pose da câmara pode ser estimada com base em dois tipos de técnicas de *tracking*: baseadas na visão com marcas e sem marcas. As técnicas de *tracking* baseadas na visão com marcas baseiam-se em marcas fiduciais colocadas manualmente no mundo real para efetuar o *tracking*. As técnicas de *tracking* baseadas na visão sem marcas utilizam características naturalmente presentes no mundo real de forma a efetuar o *tracking*. Tendo em conta os requisitos do projeto MixAR, foram selecionadas as técnicas *tracking* baseadas na visão sem marcas para serem implementadas no sistema. Estas revelam-se como as técnicas de *tracking* mais indicadas por diversos fatores: têm um elevado potencial de obter um bom nível de robustez e precisão no alinhamento entre os conteúdos real e virtual (também conhecida como tarefa do registo); possibilitam a criação de soluções de *tracking* económicas devido ao baixo

custo de aquisição das câmaras atualmente; não necessitam de uma preparação prévia do ambiente; por fim, conseguem um bom intervalo de distância na deteção do alvo.

Tal como referido anteriormente, o objetivo deste trabalho foi o desenvolvimento de uma metodologia de *tracking* para o projeto MixAR. Diversas tarefas foram realizadas neste contexto. Foi efetuado um estado da arte de forma a identificar as técnicas de *tracking* existentes na literatura bem como as suas vantagens e desvantagens a fim de determinar qual ou quais as mais apropriadas para serem utilizadas no projeto MixAR. Após a identificação das técnicas de *tracking* adequadas para o projeto MixAR (baseadas na visão sem marcas), foi efetuada uma comparação entre diversos Software Development Kit (SDK) de RM disponíveis no mercado que implementassem este tipo de técnicas, resultando na seleção de um SDK de RM denominado Metaio. A ideia por detrás do SDK foi acelerar o processo de desenvolvimento de aplicações de RM e possibilitar a experimentação de diferentes técnicas de *tracking* em estruturas do mundo real. Diversas técnicas de *tracking* disponíveis no Metaio SDK foram testadas de forma a detetar quais as que conseguiam os melhores resultados mantendo um bom nível de desempenho por parte da unidade móvel. Por fim, o trabalho consistiu numa configuração mais cuidada destas técnicas com o objetivo de criar um produto final para ser testado por utilizadores, para, entre outras coisas, obter uma avaliação do desempenho da metodologia de *tracking* criada.

O processo de *tracking* em si ocorre na unidade móvel. Por este motivo, o trabalho aqui apresentado foi implementado numa aplicação que é executada pela unidade móvel, designada MixAR App. A primeira versão desta aplicação foi definida pela obtenção de um primeiro protótipo funcional. Nesta versão, a tarefa de *tracking* foi realizada com recurso a apenas uma técnica de *tracking* disponível no Metaio. De maneira a obter alguns dados relativamente ao seu desempenho foram realizados testes de desempenho compreendendo uma unidade móvel de configuração não-imersiva (*tablet*). Após a conclusão dos testes de desempenho, novas funcionalidades foram adicionadas ao processo de *tracking*, o que resultou numa segunda versão da MixAR App. Estas foram: a implementação de um sistema de *tracking* multimodal, permitindo assim escolher a técnica de *tracking* mais adequada para cada caso, alterando entre estas de forma dinâmica; a implementação de raiz uma técnica de *tracking* baseada em sensores para fazer *tracking* às estruturas mais distantes do utilizador. Por fim, na segunda versão, foram efetuados novos testes de desempenho utilizando cinco unidades móveis diferentes (duas imersivas e três não imersivas). Foram, ainda, realizados

testes com utilizadores (seguido do preenchimento de um questionário), utilizando duas unidades móveis distintas (uma imersiva e uma não-imersiva). Os testes com utilizadores tiveram como objetivo avaliar a sensação de imersão, presença e conforto dos utilizadores durante a utilização destas unidades móveis. Após uma análise dos resultados dos testes de desempenho e dos questionários preenchidos pelos utilizadores pode-se concluir que, de uma forma geral, em ambos os testes, as soluções imersivas conseguiram resultados mais satisfatórios que as soluções não imersivas.

Em relação à organização da dissertação, esta é composta por seis capítulos. O capítulo 1 apresenta os diversos conceitos em que se baseia a dissertação, assim como a motivação e os objetivos que levaram à sua realização.

O capítulo 2 é iniciado pela definição do conceito de RM juntamente com a definição dos conceitos relacionados: RV, RA e VA. São apresentados os conceitos base relacionados com a operação de *tracking* na RM bem como as tecnologias existentes para o efeito. É feito um levantamento do estado da arte das técnicas de *tracking* existentes para a RM, com foco nas técnicas de *tracking* baseadas na visão sem marcas, onde são analisadas as vantagens e desvantagens destas. Além disso, são abordadas duas tarefas relevantes em aplicações de RM: *rendering* e modelação. O capítulo termina com uma comparação entre alguns SDK de RM disponíveis no mercado e com a apresentação de diversos exemplos de aplicações de RM, na sua maioria relacionados com a área da arqueologia.

Devido à forte ligação do trabalho aqui desenvolvido ao projeto MixAR, o capítulo 3 começa por enquadrar o trabalho através da descrição deste projeto, do seu funcionamento e dos vários componentes que constituem a sua arquitetura. De seguida, é apresentado o equipamento e as ferramentas selecionadas para o desenvolvimento do MixAR. Por último, são apresentadas as técnicas de *tracking* disponibilizadas no SDK de RM selecionado juntamente com uma explicação de como estas são executadas.

O capítulo 4 foca-se na metodologia de *tracking* e nas alterações da mesma ao longo do seu desenvolvimento e integração na MixAR App.

Relativamente ao capítulo 5, inicialmente, é descrito o processo de criação das experiências de RM, que levou à realização dos testes. Depois, são apresentados e discutidos os resultados dos testes efetuados às duas versões da MixAR App, sendo estes: testes de

desempenho, na primeira versão, e testes de desempenho e testes com utilizadores, na segunda versão da aplicação.

O capítulo 6 apresenta conclusões relativamente ao desempenho das unidades móveis, assim como conclusões relacionadas com a imersão, presença e conforto por parte dos utilizadores na utilização destas unidades móveis. Por fim, são apresentadas algumas considerações para um trabalho futuro.

2 Estado da Arte

O capítulo começa por uma secção que descreve o conceito de RV. Este é um conceito relevante pois é derivado dele que surge o conceito chave no qual se baseia o trabalho apresentado nesta dissertação, o conceito de RM. A descrição do conceito de RM é efetuada na secção seguinte (2.2). Utilizando subsecções são, também, descritos os conceitos englobados na RM: RA e VA. A secção seguinte (2.3.1), explica como é realizado o alinhamento entre objetos reais e virtuais, em aplicações de RM. Dentro desta, subsecções são utilizadas para introduzir os conceitos base relacionados com o processo de *tracking* na RM e para descrever como é feita a deteção do ambiente real e do movimento do utilizador. A próxima secção (2.4), descreve os vários tipos de técnicas de *tracking* na RM, nomeadamente: baseadas em sensores, baseadas em imagem e híbridas. A secção 2.5, refere um processo fundamental na RM: o processo de *rendering*, responsável por apresentar a combinação dos conteúdos virtual e real. A secção seguinte (2.6), apresenta o processo de modelação, responsável pela criação do conteúdo virtual que, no âmbito da RM, tem como objetivo ser combinado com a imagem do mundo real. A secção 2.7, apresenta diversos SDK de RM. Além disso, refere as funcionalidades de cada um e efetua uma comparação entre os mesmos. Por último, na secção 2.8, são apresentados alguns exemplos de aplicações de RM relacionadas com as áreas da herança cultural/arqueologia e turismo. Uma descrição de cada aplicação é apresentada juntamente com um resumo das suas abordagens, onde são identificados os tipos de técnicas de *tracking* utilizadas.

2.1 Realidade Virtual

A Realidade Virtual (RV) teve o seu início na década de 60. Mais propriamente, em 1965, quando Ivan Sutherland, fundador da área científica da computação gráfica, apresentou o primeiro sistema de RV intitulado "The Ultimate Display": um sistema com imersão e interação multissensorial (Sutherland, 1965). Desde então, a RV tornou-se uma área de investigação na computação gráfica. O termo "Realidade Virtual" foi cunhado por Jaron Lanier, em 1989, para descrever um ambiente simulado por computador com o qual as pessoas podiam interagir (Machover & Tice, 1994).

Segundo Milgram (Milgram & Kishino, 1994), a vista convencional de um ambiente de RV é uma onde o participante/observador está totalmente imerso e é capaz de interagir

com um mundo completamente sintético. Tal mundo pode imitar as propriedades de alguns ambientes do mundo real. No entanto, também pode ultrapassar as barreiras da realidade física através da criação de um mundo no qual as leis físicas que governam o espaço, tempo, mecânica, etc. não se aplicam. Jayaram (Jayaram, Connacher, & Lyons, 1997) define RV como a utilização ambientes virtuais gerados por computador e hardware associado para fornecer ao utilizador uma ilusão de presença física nesse ambiente. O autor refere, ainda, a RV como uma tecnologia que é normalmente vista como uma extensão natural de gráficos 3D gerados por computador com dispositivos de entrada e saída avançados. Brooks (Brooks Jr, 1999) define uma experiência de RV como qualquer uma onde o utilizador esteja imerso num mundo virtual responsivo e possua um controlo dinâmico do ponto de vista. O autor define, ainda, um sistema de RV como sendo tipicamente constituído por:

- ferramentas de saída (visuais, auditivas e táteis) que imergem o utilizador no mundo virtual e que bloqueiam as impressões sensoriais contraditórias do mundo real;
- ferramentas de entrada (e.g. câmara, luvas de deteção de movimento, giroscópio) que detetam, de forma contínua, a posição e movimentos do utilizador;
- sistema de renderização de gráficos que gera, a 20 ou 30 imagens por segundo, imagens do mundo virtual;
- construção da base de dados (BD) e software de modelação de objetos virtuais para criar e gerir modelos detalhados e realistas do mundo virtual.

Uma vez que as ferramentas de entrada detetam o movimento do utilizador, este pode modificar o ambiente sintético, criando a ilusão de interação com o ambiente virtual e sensação de imersão no mesmo. Segundo Burdea (Burdea & Coiffet, 2003), a RV consiste numa simulação em que a computação gráfica é utilizada para criar um mundo de aparência realista. Este mundo sintético não é estático e responde à entrada do utilizador (e.g. gestos, comandos verbais), o que marca uma característica principal da RV: a interação em velocidade interativa.

Steuer (Steuer, Biocca, & Levy, 1995) refere a RV como algo que é tipicamente definido como uma coleção particular de hardware, incluindo computadores capazes de produzir animações em velocidade interativa, HMD, *headphones* e luvas de deteção de

movimento. O autor refere, ainda, que é possível definir RV em termos de experiência humana, utilizando como chave o conceito de presença, que pode ser definido como o sentido de estar num ambiente. Segundo Witmer (Witmer & Singer, 1998), o conceito de presença é definido como a experiência subjetiva de estar num sítio ou ambiente, mesmo quando a pessoa está fisicamente localizada em outro. Os conceitos de imersão e presença são frequentemente associados à RV. No entanto, embora sejam relacionados estes não têm o mesmo significado. Slater (Slater & Wilbur, 1997) apresenta uma distinção entre ambos conceitos: imersão é a descrição de uma tecnologia e descreve a medida em que os *displays* de computador são capazes de fornecer uma ilusão da realidade para os sentidos de um participante humano; presença é um estado de consciência, a sensação de estar no ambiente virtual.

Com vista a atingir a sensação de imersão, a visualização destes mundos virtuais é, normalmente, feita através de dispositivos especiais tais como HMD, projetores envolventes do tipo Cave Automatic Virtual Environment (CAVE), projetores panorâmicos, projetores de bancada e *desktop displays* (Brooks Jr, 1999). De acordo com Grajewski (Grajewski, Górski, Zawadzki, & Hamrol, 2013), a sensação de imersão é alcançada principalmente pela projeção estereoscópica, uma vez que fornece uma ilusão de profundidade espacial, uma característica chave nos ambientes virtuais. O autor refere, ainda, os HMD como a tecnologia mais utilizada para este tipo de projeção atualmente. Estes permitem ao utilizador ver imagens geradas por computador com as quais podem interagir. São compostos por um ou dois ecrãs, posicionados na frente dos olhos do utilizador, que apresentam imagens gráficas. O utilizador vê duas imagens separadas, uma em cada olho que o cérebro combina para formar uma imagem 3D. Além da projeção estereoscópica, a sensação de imersão é alcançada pela deteção dos movimentos da cabeça do utilizador e pela consequente alteração da visualização. Azuma (R. T. Azuma, 1997) distingue dois tipos de HMD: *closed-view* e *see-through*. Tal como o nome sugere, um HMD *closed-view* possui uma "visualização fechada", o que significa que não é possível obter uma visualização direta do mundo real. Esta é uma característica que torna os HMD deste tipo naturalmente indicados para ambientes de RV, visto que estes ambientes são completamente sintéticos e não necessitam de obter qualquer visualização do mundo real. Já os HMD do tipo *see-through*, permitem "ver através" do ecrã uma visualização direta do mundo real. Por sua vez, esta característica torna-os indicados para a RM, motivo pelo qual

serão descritos com maior pormenor na devida secção (2.2). A título de exemplo, a Figura 1 apresenta um exemplo de um HMD *closed-view*.



Figura 1 - Oculus Rift, um exemplo de um HMD do tipo *closed-view*. Imagem obtida de: <http://www.techcult.com.br/wp-content/uploads/2015/05/oculus-rift.jpg>

No início, a grande maioria das aplicações da RV eram direcionadas ao entretenimento e ao treino de simulação. Hoje em dia, após décadas de desenvolvimento, a RV expandiu destas áreas para aplicações nas áreas da investigação aeronáutica, arquitetura, medicina, defesa, educação e treino (N.-N. Zhou & Deng, 2009). Alguns exemplos de diferentes áreas serão aqui referidos. Na área do treino, Jayaram (Jayaram et al., 1997) apresenta os conceitos que suportam um sistema de montagem virtual baseado em RV. O autor refere que a utilização da RV num sistema deste tipo trás como vantagem a redução significativa do tempo de concepção, reconcepção e criação de protótipos. Raque (Raque, Goble, Jones, Waldman, & Sutton, 2015) apresenta uma aplicação de RV para efeitos de simulação na área da medicina. A aplicação, denominada CAE AccuTouch, consiste num simulador baseado em RV concebido para que os seus utilizadores possam treinar broncoscopias e endoscopias superiores e inferiores. O objetivo da aplicação é fornecer, a cirurgiões recém-formados, a possibilidade de adquirirem e praticarem as suas habilidades antes de lidarem com um paciente real. Um último exemplo, na área da arquitetura, Sacks (Sacks et al., 2015) apresenta um conjunto de estudos que examinam o potencial de ferramentas de RV para um diálogo colaborativo entre arquitetos e construtores com vista a aumentarem a segurança na construção de um projeto. O autor mostra como o diálogo entre ambas as partes, juntamente com a possibilidade de alterar o objeto de estudo de uma forma proativa pode levar ao aumento da sua segurança na fase de construção.

2.2 Realidade Mista

Milgram e Kishino (Milgram & Kishino, 1994) referem a RM de uma forma genérica como uma subclasse particular de tecnologias relacionadas com RV que envolvem a fusão dos mundos real e virtual. Os autores definem o conceito de *virtuality continuum*, ou contínuo de virtualidade (Figura 2), este é um conceito que se relaciona com a mistura de classes de objetos apresentados em qualquer situação de visualização, onde ambientes reais (i.e. ambientes constituídos apenas por objetos reais) são mostrados no extremo esquerdo do *continuum* e ambientes virtuais (i.e. ambientes constituídos apenas por objetos virtuais) no extremo direito. A maneira mais simples de visualizar um ambiente de RM é, portanto, um onde objetos do mundo real e do mundo virtual são apresentados juntos num único *display*, isto é, em qualquer lugar entre os extremos do *virtuality continuum*.

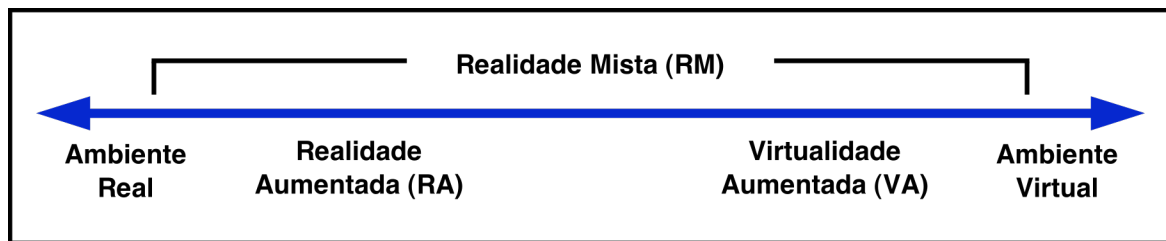


Figura 2 - Representação do contínuo de virtualidade, adaptado de Milgram and Kishino (1994).

A RA e VA são implementações de RM, cada uma compreendendo uma proporção diferente de objetos reais e virtuais dentro da imagem composta. De forma a diferenciar os conceitos de RA e VA, deve-se ter em conta qual o mundo predominante, sendo o mundo real predominante numa aplicação de RA e o mundo virtual numa aplicação de VA (Milgram & Kishino, 1994).

Semelhante à RV, a visualização destes ambientes de RM pode ser efetuada de uma forma imersiva ou não-imersiva. De uma forma não-imersiva são utilizados dispositivos tais como *tablets* e monitores comuns. Para se obter uma visualização imersiva recorre-se ao uso de dispositivos especiais. Tal como referido na secção 2.1, a tecnologia mais utilizada para este efeito são os HMD. No entanto, na RM, são utilizados HMD *see-through* ao invés dos HMD *closed-view* utilizados na RV. Os HMD *see-through* são indicados para a RM, este tipo de HMD permite que o utilizador veja o mundo real, com objetos virtuais sobrepostos utilizando uma tecnologia de ótica (HMD *Optical see-through*) ou de vídeo (HMD *Video see-through*). Segundo Rolland (Rolland, Holloway, & Fuchs, 1995), com um HMD *optical see-through*.

through, o mundo real é visualizado através de espelhos com um determinado nível de transparência posicionados à frente dos olhos do utilizador, tal como ilustra a Figura 3. Estes espelhos são também utilizados para refletir as imagens geradas por computador para os olhos do utilizador, combinando, de forma ótica, as visualizações dos mundos real e virtual. Com um HMD *video see-through*, as imagens do mundo real são capturadas por uma ou duas câmaras montadas no HMD (Figura 4) e as imagens geradas por computador são combinadas eletronicamente com a representação em vídeo do mundo real.



Figura 3 - HMD *optical see-through* Vuzix STAR 1200XLD. A câmara presente no HMD é utilizada para técnicas de *tracking* baseadas na visão e não para fornecer ao utilizador uma visualização do mundo real. Esta é obtida diretamente pelo utilizador devido à transparência das lentes. Imagem obtida de: http://www.vuzix.com/augmented-reality/products_star1200xld/



Figura 4 - HMD *video see-through* Vuzix Wrap 1200DXAR. Neste caso, a imagem do mundo real é obtida utilizando uma ou duas câmaras e exibida nos *displays* embutidos. Imagem obtida de: http://www.vuzix.com/augmented-reality/products_wrap1200dxar/

2.2.1 Realidade Aumentada

Azuma (R. T. Azuma, 1997) define RA como uma variação de RV. Na RV o utilizador é totalmente imerso num mundo sintético, sem a capacidade de ver o que se passa à sua volta no mundo real. A RA, pelo contrário, permite ao utilizador ver o mundo real, com objetos virtuais sobrepostos sobre o mesmo. Complementando, assim, a realidade, ao invés de

a substituir por completo. O autor define, ainda, RA como qualquer sistema que possua as seguintes características:

- fusão do real e virtual, num ambiente real;
- interativo em tempo real;
- registo em 3D.

Ao invés de tempo real, Rolland (Rolland, Davis, & Baillot, 2001) utiliza o termo velocidade interativa para descrever esta característica de um sistema. O autor justifica-se dizendo que a execução em tempo real é praticamente inatingível uma vez que o tempo de processamento não é zero. Já o termo velocidade interativa, refere-se a um sistema que reage "a tempo" de acordo com as ações tomadas por um utilizador, rápido o suficiente para permitir a um utilizador completar a sua tarefa de forma satisfatória. Milgram (Milgram & Kishino, 1994) refere o termo RA como qualquer caso onde um ambiente real é aumentado por objetos virtuais. Para Bajura (Bajura & Neumann, 1995), os sistemas de RA permitem aos utilizadores interagir com objetos gerados por computador e reais, ao exibir objetos virtuais 3D no ambiente natural do utilizador. Teichrieb (Teichrieb et al., 2007) define a RA como a integração de objetos virtuais 3D num ambiente real 3D em velocidade interativa, melhorando a percepção e interação do utilizador com o mundo. Segundo Afif (Afif, Basori, & Saari, 2013), a RA tenta combinar objetos gerados por computador num espaço do mundo real. A tecnologia consiste na sobreposição de objetos virtuais numa cena do mundo real capturada por uma câmara de vídeo. O utilizador pode ver estes objetos através de um *display*, resultando na percepção de que os objetos reais e virtuais estão no mesmo espaço. A RA pode exibir informação que pode não ser diretamente percebida pela visão ou audição, estendendo assim os sentidos humanos. Para Cai (Cai, Wang, & Chiang, 2014), a RA é uma extensão da RV. Ao contrário da tradicional RV, a RA fornece uma interface para utilizadores que combina ambos os mundos real e virtual. Os utilizadores podem interagir com objetos virtuais que são sobrepostos em cenas reais à sua volta e obter uma experiência de interação pessoa computador mais natural.

O primeiro sistema de RA foi criado em 1968, por Ivan Sutherland (Sutherland, 1968) com a ajuda do seu aluno Bob Sproull (Figura 5). O sistema, intitulado "The Sword of Damocles", utilizava um HMD *optical see-through* para visualizar o conteúdo virtual

sobreposto nas imagens do mundo real. No entanto, devido ao limitado poder computacional na altura, apenas desenhos muito simples podiam ser exibidos em velocidade interativa.

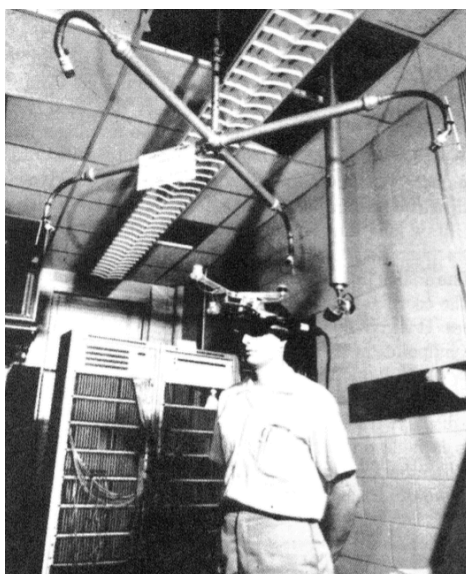


Figura 5 - "The Sword of Damocles", o primeiro sistema de RA (Sutherland, 1968).

O termo "Realidade Aumentada" foi cunhado em 1992, por Tom Caudell e David Mizell (Caudell & Mizell, 1992), para referir-se à sobreposição de material apresentado por computador na visualização do mundo real. Estes discutiram algumas vantagens da RA em relação à RV, como, por exemplo, uma menor necessidade de poder computacional dado que menos pixéis têm de ser renderizados. Além disso, reconheceram a dificuldade aumentada no registo de forma a alinhar o virtual e o real.

O objetivo básico de um sistema de RA é melhorar a percepção e interação do utilizador com o mundo real através da adição de objetos virtuais 3D que parecem coexistir no mesmo espaço que o mundo real (R. Azuma et al., 2001). De acordo com Krevelen (Krevelen & Poelman, 2010), os componentes principais para construir um sistema de RA mantiveram-se os mesmos desde o trabalho pioneiro de Sutherland na década de 1960. O autor refere-se a estes como os *displays*, que combinam os mundos virtual e real, os sensores (*trackers*) e as suas aproximações para detetar a posição e orientação do ponto de vista do utilizador, de forma a obter um registo correto do real e virtual, e os computadores e softwares gráficos.

Inicialmente, as aplicações de RA focaram-se nas áreas militares, médicas e industriais. Hoje em dia têm um amplo espectro de aplicações, em diversas áreas, tais como

(R. T. Azuma, 1997) (Krevelen & Poelman, 2010): assistência pessoal e publicidade, navegação, turismo, anotação, planeamento de trajetória de robôs, navegação de aeronaves militares, concepção de produtos, montagem, manutenção e reparação, combate e simulação, visualizações médicas, transmissões desportivas, jogos, colaboração, educação e treino.

Alguns exemplos de aplicações da RA em diferentes áreas serão aqui referenciados. Na área da manutenção/reparação, Platonov (Platonov, Heibel, Meier, & Grollmann, 2006) apresenta um sistema de RA que pretende guiar técnicos na reparação e manutenção em diferentes cenários industriais. Na indústria da arquitetura, engenharia e construção, Dunston (Dunston & Wang, 2005) descreve um protótipo de RA denominado AR CAD (Computer Aided Design), uma plataforma de investigação para estabelecer os princípios científicos fundamentais para a aplicação de conceitos e tecnologias de RA no planeamento e concepção de instalações. Na área da medicina, Londei (Londei et al., 2015) apresenta uma forma alternativa de assistir cirurgias na colocação de pregos intramedulares. Tradicionalmente, este processo é guiado através de raios X e requer uma quantidade significativa de tempo e exposição à radiação. Para absolver estas complicações, o autor propõe orientar o cirurgião através de um sistema de vídeo que utiliza a RA para indicar, de forma precisa, a posição pretendida para o prego intramedular e como realizar o seu subsequente bloqueio. Como último exemplo, nas áreas do treino e educação, três aplicações de RA distintas são apresentadas por Martín-Gutiérrez (Martín-Gutiérrez, Fabiani, Benesova, Meneses, & Mora, 2015). As aplicações têm em vista melhorar aspetos teóricos e práticos dos alunos no ramo da engenharia eletrotécnica, mais especificamente: ElectARmanual, utilizada para suportar a realização de tarefas de alunos num laboratório prático e para treinarem a utilização de máquina elétricas; ELECT3D, utilizada para os alunos compreenderem os planos elétricos e os seus símbolos; ElectAR_notes: utilizada para enriquecer notas teóricas sobre eletricidade com conteúdo virtual.

2.2.2 Virtualidade Aumentada

Tori (Tori, Kirner, & Siscoutto, 2006) define Virtualidade Aumentada (VA) como uma particularização da Realidade Mista (RM), onde o ambiente principal é o virtual ou há predominância do virtual. O autor define ainda a VA como o enriquecimento de um ambiente virtual com elementos reais, estáticos tais como uma cadeira, ou dinâmicos, como é o exemplo das mãos de um humano, pré-capturados ou capturados em tempo real. Segundo

Milgram (Milgram & Kishino, 1994), na VA o aumento não é feito sobre uma representação direta de uma cena real, mas sim sobre um mundo virtual, que é gerado primariamente por computador. O autor refere, ainda, que à medida que a tecnologia progride, pode, eventualmente, tornar-se menos evidente, perceber se o mundo primário a ser experimentado é, de facto, predominantemente real ou virtual, o que pode enfraquecer a utilização de ambos os termos RA e VA. Tal não deve, no entanto, afetar a validade do termo mais geral Realidade Mista (RM).

Ternier (Ternier, Klemke, Kalz, Van Ulzen, & Specht, 2012) refere o conceito de imersão como importante para aplicações de VA. Segundo o autor, na VA um participante navega através de um mundo sintético interagindo com objetos ficcionais ou reais. O autor apresenta, ainda, um caso de uso da VA direcionado para o ensino, intitulado StreetLearn, um aumento do cliente Google StreetView cujo objetivo é a introdução de elementos de gamificação interativos para aumentar a motivação do aprendiz e fornecer uma experiência de aprendizagem mais rica. Simsarian (Simsarian & Akesson, 1997) apresenta uma ferramenta e uma metodologia que podem ser usadas para criar mundos virtuais aumentados com texturas retiradas de objetos do mundo real. O objetivo é fornecer, através da VA, a capacidade de explorar um espaço remoto de forma interativa. Outra aplicação de VA é apresentada por Regenbrecht (Regenbrecht et al., 2004). Este apresenta a implementação de um sistema de videoconferência baseado em VA intitulado "cAR/PE!", o qual tenciona transportar os participantes para um cenário virtual com representações de vídeo e áudio espacialmente alinhadas. A solução apresentada permite que três participantes comuniquem através de uma rede num ambiente, simulando um encontro tradicional cara-a-cara. Na área da medicina, Paul (Paul, Fleig, & Jannin, 2005) apresenta um sistema de VA direcionado para neurocirurgias multimodais guiadas por imagem. O seu objetivo é efetuar uma tarefa crítica neste tipo de cirurgias: a visualização de informação anatómica e fisiológica, derivada de imagens médicas pré operativas, durante a realização de uma neurocirurgia. Para tal, o sistema apresentado pelo autor utiliza a VA para colocar imagens intraoperatórias do campo operatório sobre modelos virtuais 3D gerados através das imagens multimodais pré operativas. O resultado é mostrado num ecrã externo durante a realização da neurocirurgia. Sanguino (Sanguino, Márquez, Carlson, & Millán, 2014) apresenta uma aplicação da VA na área da robótica, mais especificamente, na navegação de veículos robóticos. A assistência na navegação destes veículos desempenha um papel fundamental na realização das suas tarefas.

Interfaces básicas baseadas em vídeo envolvem várias desvantagens consideráveis, resultando numa operação imprecisa do robô devido à limitada consciencialização do ambiente por parte do utilizador. Face a este problema, o autor propõe um sistema, baseado em VA, que constrói uma cena virtual 3D que combina vídeo em tempo real, informação proveniente de diversos sensores de infravermelhos, também em tempo real, e um mapa do ambiente, com vista a fornecer aos seus utilizadores uma percepção melhorada do ambiente remoto.

2.3 O Processo de Tracking na Realidade Mista: Conceitos Base

Tracking é definido como ato ou processo de seguir algo ou alguém. Na área da RM, o *tracking* consiste, fundamentalmente, numa obtenção contínua do ponto de vista do utilizador. Segundo Comport (Comport, Marchand, Pressigout, & Chaumette, 2006), o *tracking* de um objeto numa cena refere-se à obtenção de uma estimativa da pose da câmara relativamente a esse mesmo objeto. Este é um passo crucial nas aplicações de RM dado que os objetos virtuais são projetados na cena real com base nesta estimativa da pose da câmara. Para Lepetit (Lepetit & Fua, 2005), *tracking* significa identificar a posição de um objeto, de forma contínua, quando o objeto ou câmara se movimenta. Para tal existe uma grande variedade de aproximações dependendo do tipo de objeto, graus de liberdade do objeto e da câmara, e da aplicação alvo. O autor define dois métodos de *tracking*: *tracking* 2D, o qual segue a projeção de imagens de objetos ou partes de objetos cuja deslocação 3D resulte num movimento que possa ser modelado como uma transformação 2D (Este tipo de métodos não recolhe a posição do objeto no espaço); *tracking* 3D, recolhem os seis graus de liberdade que definem a pose da câmara relativa à cena, ou de forma equivalente, a deslocação 3D do objeto relativo à câmara. O *tracking* 3D é uma ferramenta útil para diversas áreas tais como a RA, servo visão e interfaces homem-máquina (Lepetit & Fua, 2005). No entanto, estas possuem diferentes finalidades:

- na RA pretende-se sobrepor conteúdo virtual sobre cenas do mundo real;
- na servo visão pretende-se controlar o movimento de um robot;
- nas interfaces homem-máquina pretende-se substituir interfaces tradicionais por interfaces mais intuitivas e naturais.

Segundo Krevelen (Krevelen & Poelman, 2010), a tarefa de *tracking* é, normalmente, mais simples quando se lida com ambientes interiores. A razão prende-se pelo facto de que,

em tais aproximações, os dispositivos de *tracking* não necessitam de ser completamente móveis, *wearables* ou capazes de lidar com as condições adversas que possam surgir no exterior. Um ambiente interior é mais simples de modelar e preparar. Além disso, condições tais como a temperatura e iluminação podem ser controladas.

As próximas subsecções focam-se em dois conceitos intrinsecamente relacionados e relevantes para a descrição do processo de *tracking* na RM: *tracking* (deteção do ambiente real e do movimento do utilizador) e registo (alinhamento entre objetos reais e virtuais). A primeira subsecção descreve o que se pretende obter com a tarefa de *tracking* e apresenta os diversos dispositivos/sensores, e problemas relacionados com estes, que podem ser utilizados para a sua realização. A segunda e última subsecção, descreve o conceito de registo e como este se relaciona com os diferentes tipos de dispositivos que podem ser utilizados para efetuar o *tracking*.

2.3.1 Deteção do Ambiente Real e do Movimento do Utilizador

Antes de um sistema de RM mostrar objetos virtuais num ambiente real, o sistema deve ser capaz de detetar o ambiente envolvente e fazer *tracking* do movimento relativo do espetador, preferivelmente com seis graus de liberdade (do anglo-saxónico Degrees of Freedom - DOF), sendo estes definidos por três variáveis (x , y , z) para a posição e três ângulos (*yaw*, *pitch* e *roll*) para a orientação (Van Krevelen & Poelman, 2010). O autor afirma que a obtenção de um registo preciso requer algum tipo de modelo do ambiente. Além disso, a maioria dos ambientes têm de ser preparados antes de um sistema de RM ser capaz de detetar o movimento com seis graus de liberdade. Contudo, nem todas as técnicas de *tracking* funcionam em qualquer ambiente. O autor refere, também, que determinar a posição e orientação de um utilizador continua a ser um problema complexo, sem nenhuma solução única ideal.

Segundo Azuma (R. T. Azuma, 1997), quando comparados com ambientes virtuais ou RV, os dispositivos de *tracking* de RA devem ter uma maior precisão, maior variedade de entradas de sistema e banda larga, e suportar maiores distâncias. Este foi o motivo pelo qual a RA demorou mais tempo a amadurecer do que a RV (Krevelen & Poelman, 2010).

Para que o ambiente seja detetado, os sistemas utilizam diversos dispositivos/sensores conhecidos por *trackers*. Um *tracker* é, fundamentalmente, um sensor cuja tarefa é detetar a

posição e/ou orientação de um objeto e tornar essa informação disponível para o resto do sistema (Holloway & Lastra, 1993). Tal como referido no início da subsecção, há seis tipos de movimentos que podem ser detetados:

- translação em x , y , z ;
- rotação sobre os eixos x , y , z : *roll*, *pitch* e *yaw*.

Estes são mutualmente ortogonais. Por este motivo, seis variáveis independentes (representando os graus de liberdade) são associadas a qualquer objeto 3D assimétrico. Estes seis números são o mínimo necessário para especificar completamente a posição e orientação de um objeto rígido. Um *tracker* pode monitorizar todos os movimentos ou apenas um subconjunto, dependendo da implementação. Além disso, alguns *trackers* monitorizam apenas um intervalo limitado de uma determinada variável (Holloway & Lastra, 1993).

O mesmo autor indica os diferentes tipos de *trackers* existentes:

- magnético: o campo magnético emitido por um transmissor induz corrente num receptor, de acordo com a sua distância e orientação;
- ótico: imagem obtida de uma ou mais câmaras;
- mecânico: mede a posição e orientação do dispositivo através de uma ligação física ao objeto;
- acústico: a transformação do objeto é determinada com base num som de frequência ultrassónica que é refletido do objeto;
- inercial: acelerómetros e giroscópios são utilizados para detetar alterações de velocidade linear e angular, respetivamente.

Além dos *trackers* mencionados acima, Krevelen menciona também os seguintes (Van Krevelen & Poelman, 2010):

- GPS: determina a posição do utilizador quando, pelo menos, quatro satélites estão diretamente visíveis, com uma precisão de 10-15m. dentro deste tipo de *tracker* está incluído o GPS assistido (Assisted Global Positioning System - A-GPS) que não apresenta a restrição anterior, a tecnologia Wide Area Augmentation System (WAAS), que aumenta a precisão para 3-4m, o GPS diferencial, com uma precisão de 1-3m, e,

por fim, o GPS RTK (Real Time-Kinematic) capaz de obter uma precisão de centímetros.

- radio: baseia-se em ondas de rádio de ultra banda larga para determinar a posição do utilizador. Dentro deste tipo de *tracker* estão incluídos os chips RFID (Radio Frequency IDentification) e, de forma complementar, redes sem fios (802.11b/g/n) para efetuar o *tracking*.

Holloway refere, ainda, os diversos problemas relacionados com os diferentes tipos de *trackers* (Holloway & Lastra, 1993):

- taxa de atualização: o número de vezes por segundo que é feita uma medição limita a frequência com que o *display* possa ser atualizado;
- latência: o tempo que decorre desde que uma alteração dos movimentos seja efetuada até que esta seja refletida;
- precisão: erro obtido na deteção dos movimentos;
- resolução: a menor quantidade que o *tracker* consegue detetar. Um movimento menor que a resolução de um *tracker* não será refletido na sua saída;
- interferência/distorção: todos os *trackers*, à exceção dos inerciais, estão sujeitos a interferência ou distorção;
- tipos de valores: um *tracker* pode reportar valores de posição/orientação absolutos ou relativos;
- intervalo: volume de trabalho e cobertura angular. *Trackers* absolutos têm limites no volume de trabalho. Muitos sistemas têm, também, um limite de intervalo angular;
- tamanho/peso: um tamanho ou peso excessivo pode condicionar a sua utilização;
- robustez: se o *tracker* está sujeito a erros grosseiros quando o ambiente sobre o qual opera está degradado;
- DOFs medidos: alguns *trackers* apenas monitorizam um subconjunto dos 6 movimentos possíveis;
- segurança: se a utilização do *tracker* apresenta algum risco de saúde a curto ou a longo prazo.

2.3.2 Alinhamento entre Objetos Reais e Virtuais

O alinhamento entre objetos reais e virtuais é um aspeto crucial em aplicações de RM. Sem um alinhamento preciso, a ilusão de que os objetos virtuais coexistem no mesmo espaço do que os reais é severamente comprometida. A obtenção de um alinhamento preciso é um problema comum na RM e na RV. Contudo, na RM é exigido um alinhamento mais preciso que na RV. A razão prende-se no facto de que é mais difícil o utilizador notar erros de alinhamento na RV (R. T. Azuma, 1997). Segundo Azuma (R. Azuma et al., 2001), o alinhamento entre os objetos reais e virtuais é definido como registo. Esta é uma definição que não se aplica apenas a determinadas tecnologias, como HMD, nem está limitada ao sentido da visão. Bajura (Bajura & Neumann, 1995) refere que, se uma aplicação de RA não mantém um registo preciso, os objetos virtuais irão parecer flutuar no ambiente natural do utilizador sem terem uma posição espacial 3D específica. Por este motivo, os objetos virtuais devem estar visualmente registados, de acordo com os objetos do mundo real, em todas as imagens que o utilizador vê. A Figura 6 ilustra o problema do registo. Neste caso a aplicação pretende indicar o parafuso a apertar através de uma seta virtual. A seta é mostrada perto do parafuso inferior direito, no entanto a aplicação pretendia indicar o parafuso esquerdo. O erro de registo é o deslocamento observado na imagem entre a posição atual e pretendida dos objetos virtuais (Bajura & Neumann, 1995).

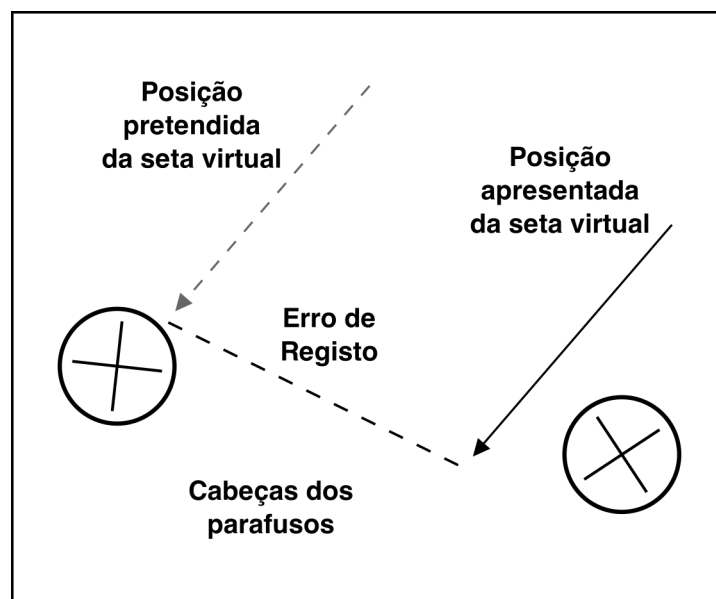


Figura 6 - Representação gráfica do problema do registo. Ao invés de a seta virtual ter sido apresentada em cima do parafuso indicado, foi apresentada próxima de um outro parafuso, possibilitando a realização de uma tarefa errada por parte do utilizador. Adaptado de Bajura and Neumann (1995).

Segundo Azuma (R. T. Azuma, 1997), erros de registo são difíceis de controlar devido aos seus requisitos de precisão elevados e diversas fontes de erros. O autor divide as fontes de erros em dois tipos: estáticos e dinâmicos. Consideram-se estáticos aqueles que causam erros no registo quando o ponto de vista do utilizador e os objetos no ambiente permanecem imóveis, e dinâmicos aqueles que surgem quando o ponto de vista ou os objetos se começam a mover.

O problema do registo está intrinsecamente relacionado com a tarefa de *tracking*. Isto é, um desalinhamento entre os conteúdos virtual e real apenas ocorre se houver uma estimativa incorreta do ponto de vista do utilizador, algo que obtido através do processo de *tracking*. Bajura (Bajura & Neumann, 1995) relaciona o problema do registo com dois tipos de sistemas: *open-loop* (Figura 7) e *closed-loop* (Figura 8). Numa abordagem *open-loop* o registo é baseado apenas na precisão dos componentes do sistema. Neste tipo de sistemas não se consegue medir o registo e como tal simplesmente assume-se que a sua saída contenha erro. A única forma de melhorar tais sistemas é aumentando a precisão dos seus componentes. Numa abordagem *closed-loop* a imagem combinada (i.e. imagem real sobreposta com conteúdo virtual ou vice-versa) é devolvida ao sistema para seja feita uma medição do erro de registo, através de técnicas de visão por computador, para que o sistema o possa corrigir de forma dinâmica. Esta é uma abordagem direccionada para sistemas que realizem a deteção do ponto de vista do utilizador com base em câmaras/vídeo uma vez que possuem imagens do real e virtual.

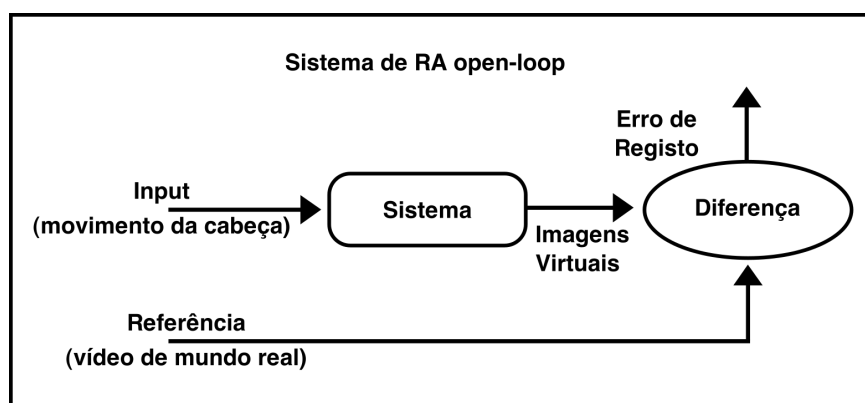


Figura 7 - Sistema de RA *open-loop*, adaptado de Bajura and Neumann (1995).

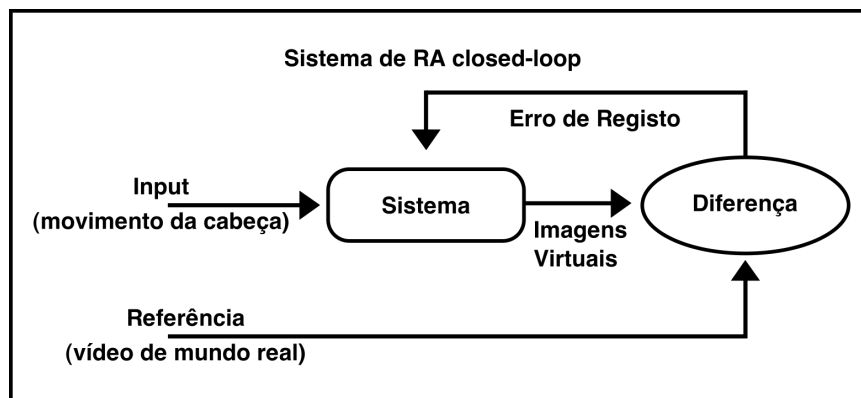


Figura 8 - Sistema de RA *closed-loop*, adaptado de Bajura and Neumann (1995).

De acordo com Lima (Lima, Simões, Figueiredo, & Kelner, 2010), o problema do registo é resolvido ao fazer *tracking* do ambiente para que os elementos sintéticos possam ser registados de forma adequada com a cena real. Utilizando tecnologias de *tracking* disponíveis são obtidas características do mundo real. Com base nessa informação, o sistema de RA determina quando, onde e como a cena virtual deve ser exibida.

A obtenção de um registo preciso está dependente, não só, do modelo geométrico utilizado, como também da distância a que os objetos a que se pretende fazer *tracking* se encontrem. Quanto mais longe um objeto se encontrar menos impacto terão os erros no *tracking* da posição e mais impacto terão os erros no *tracking* da orientação (R. T. Azuma, 1997).

2.4 Técnicas de Tracking na Realidade Mista

As técnicas de *tracking*, com a finalidade de registar o mundo virtual no real, são essenciais para o desenvolvimento de um sistema de RM. O seu objetivo é fazer com que os objetos virtuais permaneçam nos seus locais designados quando o utilizador muda o ponto de vista. Esta condição exige que o computador consiga detetar de forma contínua as mudanças no ambiente envolvente (Afif et al., 2013). Neste documento irão ser descritas técnicas de *tracking* para objetos ou cenas rígidas. Fora do seu âmbito estão as técnicas de *tracking* para objetos não-rígidos e superfícies deformáveis ou articuláveis, tais como t-shirts ou mãos humanas. Para o leitor interessado, Pilet (Pilet, 2008) mostra como tais superfícies podem ser aumentadas.

A detecção contínua das mudanças no meio envolvente é feita através de um sistema de *tracking*. Segundo Zhou (F. Zhou, Duh, & Billinghamurst, 2008), sistemas de *tracking* podem utilizar técnicas baseadas em sensores, baseadas na visão ou híbridas (detalhadas nas subsecções 2.4.1, 2.4.2 e 2.4.3, respetivamente). O autor relaciona, ainda, os sistemas de *tracking* com os sistemas *open-loop* e *closed-loop*, apresentados na subsecção 2.3.2. Referindo-se aos sistemas de *tracking*, que utilizam técnicas baseadas em sensores, como análogos a sistemas *open-loop*, cuja saída do sistema é entendida como tendo erro, e sistemas de *tracking*, que utilizam técnicas baseadas na visão, como análogos a sistemas *closed-loop*, uma vez que, através de algoritmos de visão por computador, têm a possibilidade de medir o erro de registo e devolver essa informação ao sistema para que este o reconheça e tente corrigir, fechando assim o ciclo.

2.4.1 Baseadas em Sensores

Técnicas de *tracking* baseadas em sensores foram as primeiras a serem implementadas. No entanto, devido à necessidade de posicionar equipamento especial à volta do utilizador, estas eram apenas possíveis em ambientes interiores. O primeiro HMD, desenvolvido por Ivan Sutherland (Sutherland, 1968) utilizava um *tracker* mecânico e um *tracker* ultrassónico para detetar a posição e orientação da cabeça do utilizador. As técnicas de *tracking* baseadas em sensores utilizam qualquer um dos tipos de *tracker* referidos na subsecção 2.3.1, à exceção dos *trackers* óticos uma vez que estes são incluídos na classe de técnicas baseadas na visão. Estas técnicas podem, portanto, basear-se em sensores GPS, magnéticos, mecânicos, inerciais, acústicos e rádio. Apesar de se poderem basear numa grande variedade de *trackers*, todos têm desvantagens: os *trackers* mecânicos e GPS não são precisos o suficiente, os *trackers* magnéticos sofrem de distorção causada por metal no ambiente, os ultrassónicos são imprecisos quando utilizados em longas distâncias e os magnéticos sofrem de arrasto (*drift*) ao longo do tempo (Krevelen & Poelman, 2010). Além dos problemas relacionados com a precisão do *tracking* muitos deles têm a necessidade de alterar o mundo real com dispositivos, algo que é considerado intrusivo ao ambiente. *Trackers* que possuam esta necessidade não são, portanto, indicados para serem utilizados no projeto MixAR pelo facto de que os alvos das experiências são sítios com um grande valor cultural e como tal não devem ser modificados.

Castro (Castro, Chiu, Kremenek, & Muntz, 2001) apresenta uma técnica de *tracking* baseada num *tracker* de rádio. Este apresenta o Nibble, um serviço que utiliza o protocolo IEEE 802.11 das redes sem fios para inferir a localização de um dispositivo através da medição da qualidade dos sinais. No entanto, este é um serviço que tenta determinar apenas a posição do utilizador. Para que este fosse utilizado numa aplicação de RM teria que ser combinado com dados de outro *tracker* tal como um giroscópio para obter a orientação do dispositivo.

Trackers inerciais como o acelerómetro e o giroscópio são comumente utilizados como parte de um sistema híbrido para ajudar a estimar a pose da câmara. O acelerómetro é capaz de detetar alterações de velocidade linear enquanto o giroscópio deteta alterações de velocidade angular, o que os torna indicados para o cálculo da orientação do utilizador. Além disso, podem fornecer um método prático de estimar a posição quando se combinam medições cronometradas destes com informações de direção precisas. Assim, as estimativas devem ser atualizadas periodicamente com medidas precisas a fim de minimizar os erros devido ao *drift* (Krevelen & Poelman, 2010).

2.4.2 Baseadas na Visão

Na sua essência, as técnicas de *tracking* baseadas na visão, também denominadas técnicas baseadas em sensores óticos, aplicam algoritmos de visão por computador. Tal como referido previamente, estas técnicas utilizam *trackers* óticos, têm como entrada imagens do mundo real obtidas por uma ou mais câmaras e, idealmente como saída, uma estimativa correta da pose da câmara. Segundo Doignon (Doignon, 2007), para determinar a pose da câmara é necessária a identificação de alguns parâmetros da câmara, designados parâmetros intrínsecos ou internos, num processo que é conhecido como calibração da câmara. Este é um processo que pode ser efetuado como um passo preliminar ou, em algumas situações, simultaneamente com a estimação da pose da câmara. Além dos parâmetros intrínsecos, devem ser identificados os parâmetros extrínsecos, responsáveis por definirem a posição e orientação da câmara em relação às coordenadas mundo. Segundo Lepetit (Lepetit & Fua, 2005), é comum em aplicações de RM assumir-se que os parâmetros intrínsecos são conhecidos e focarem-se em estimar apenas os parâmetros extrínsecos, ou seja, a pose da câmara. Ao assumirem-se os parâmetros intrínsecos como estáticos, a câmara não pode fazer

zoom porque é difícil distinguir uma alteração no comprimento focal a partir de uma translação no eixo Z da câmara.

Aproximações promissoras para uma estimação de pose com seis graus de liberdade de utilizadores e objetos no geral são baseadas na visão. Visto que seguem uma abordagem *closed-loop*, o campo de visão da câmara coincide com o do utilizador (e.g. num HMD *video see-through*) permitindo um registo preciso ao nível do pixel para os objetos virtuais (Krevelen & Poelman, 2010).

Segundo Afif (Afif et al., 2013), as técnicas de *tracking* baseadas na visão podem ser categorizadas em dois tipos: com marcas (*marker-based*) e sem marcas (*markerless-based*). A diferença principal entre as técnicas com marcas e as técnicas sem marcas é o método utilizado para colocar os objetos virtuais no mundo real. As técnicas com marcas levam o computador a realizar *tracking* sobre marcas fiduciais pré-definidas ou Light Emitting Diodes (LED) colocados em localizações arbitrárias. O facto de utilizarem marcas específicas pré-definidas torna estas técnicas benéficas em termos de redução de custos computacionais. No entanto, possuem várias desvantagens: o *tracking* é apenas efetuado quando as marcas estão visíveis e perto o suficiente. Além disso, não são apropriadas para alguns ambientes tais como grandes edifícios ou espaços de trabalho. As técnicas de *tracking* sem marcas surgem como uma solução para os diversos problemas apresentados pelas técnicas com marcas. Ao invés de utilizarem marcadores fiduciais, estas utilizam características naturalmente presentes no ambiente de forma a posicionar os objetos virtuais. Desta forma, não existem marcas intrusivas que não façam realmente parte do ambiente (Teichrieb et al., 2007). A mesma autora refere, ainda, que embora existam técnicas que não necessitem de marcadores fiduciais para efetuar o *tracking*, o uso de marcadores é adequado em sistemas que não se importem com a presença de elementos artificiais na cena. Infere-se que considera que a solução de *tracking* é definida pelo tipo de problema a ser enfrentado.

2.4.2.1 Marker-Based

Segundo Lepetit, o *tracking* 3D baseado na visão pode ser decomposto em dois passos principais, sendo o primeiro o processamento de imagens para extrair informação destas e o segundo passo a estimação da pose em si. A adição de marcas fiduciais à cena constitui uma grande ajuda em ambos os passos pois estas possuem características de imagem fáceis de

extrair e fornecem medidas fiáveis, fáceis de explorar, para estimar a pose (Lepetit & Fua, 2005).

Técnicas de *tracking* baseadas em marcas foram as primeiras implementações da classe de técnicas baseadas na visão. Estas podem-se basear em marcas fiduciais tais como as apresentadas por Kato (Kato & Billinghamurst, 1999) ou LED (Bajura & Neumann, 1995), posicionados de forma estratégica ao longo da cena do mundo real, de forma a recuperar a pose da câmara. As localizações ou padrões das marcas fiduciais são assumidas como conhecidas. O processamento da imagem deteta as posições destas e utiliza-as para fazer correções de modo a obter um registo preciso. Uma desvantagem deste tipo de técnicas é a necessidade de uma ou mais marcas estarem presentes na imagem do mundo real a todo o momento, pois sem estas deixa de ser possível determinar uma pose correta da câmara. Além disso, existem alguns problemas relacionados com a distância a que as marcas são visualizadas. No caso de uma pequena escala as marcas podem não ser detetadas corretamente e como tal não é realizado o *tracking*. Como vantagem, estas técnicas têm um cálculo relativamente baixo o que lhes garante um bom desempenho. Outra vantagem deste tipo de técnicas é a sua precisão. Quando as marcas estão visíveis, os resultados podem ser precisos ao nível de um pixel (R. T. Azuma, 1997).

Poupyrev (Poupyrev et al., 2001) apresenta uma aplicação denominada "Tiles" que implementa este tipo de técnicas de *tracking*. "Tiles" é uma interface de RM tangível e colaborativa que permite a vários participantes disporem e organizarem dinamicamente objetos virtuais num ambiente de RM. Neste sistema, o utilizador utiliza um HMD com uma pequena câmara acoplada em que ambos os quais são ligados a um computador. A saída da câmara é capturada pelo computador, o qual se encarrega de sobrepor imagens virtuais no vídeo do mundo real e termina por mostrar o resultado ao utilizador através do HMD. A posição e orientação 3D dos objetos virtuais é determinada através de técnicas de *tracking* baseadas em marcas utilizando marcadores fiduciais retangulares que são acoplados a cartas de 15x15cm, denominadas *tiles* (Figura 9). Os objetos virtuais são renderizados relativamente a estas marcas e são manipulados pelo utilizador através da interação física com a *tile*. O utilizador pode, assim, manipular os objetos virtuais sem ser necessário utilizar dispositivos de entrada adicionais.

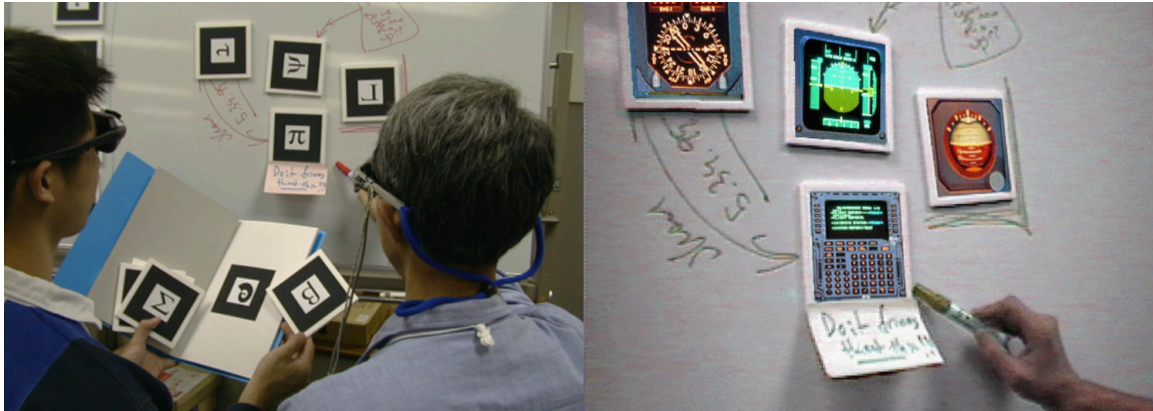


Figura 9 - (Esq.) Apresentação de diversas *tiles* com as devidas marcas fiduciais retangulares. (Dir.) Visualização de conteúdo virtual corretamente aumentado sobre algumas *tiles* (Poupyrev et al., 2001).

A utilização de marcas fiduciais retangulares, como as presentes na aplicação "Tiles", são uma abordagem bem conhecida e já com um bom nível de maturidade. Uma das *frameworks* mais populares na literatura (ARToolkit) utiliza este tipo de abordagem (Kato & Billinghurst, 1999). As marcas fiduciais mais populares são retangulares. No entanto existem outras abordagens tais como a apresentada em (Cho & Neumann, 2001) que utiliza, como marca fiducial, vários anéis de diferentes cores a fim de solucionar o problema da distância a que as marcas são visualizadas durante o *tracking*. A largura destes anéis pode ser proporcional ou constante (Figura 10), sendo uma largura proporcional adequada para sistemas de RA que devam ser capazes de trabalhar com diferentes distâncias da câmara ao alvo e uma largura constante mais indicada para sistemas que trabalhem com uma distância constante.

Anéis fiduciais com largura proporcional			
Anéis fiduciais com largura constante			
	Nível 1	Nível 2	Nível 3

Figura 10 - Marcas fiduciais constituídas por vários anéis, adaptado de Cho and Neumann (2001).

2.4.2.2 Markerless-Based

Técnicas de *tracking* baseadas na visão, sem marcas, têm a capacidade de produzir soluções não-invasivas que conseguem obter um registo preciso, desde que se esteja disposto

a investir o esforço necessário para desenvolver algoritmos robustos o suficiente (Lepetit & Fua, 2005).

Segundo Teichrieb (Teichrieb et al., 2007), a RM sem marcas conta com *trackers* especializados e robustos. Outra vantagem é a possibilidade de extrair características do ambiente que possam ser utilizadas mais tarde pelo sistema. A autora indica dois tipos pelos quais se podem dividir as técnicas de *tracking* baseadas na visão sem marcas: *model based* e *Structure from Motion* (SFM) *based* (Figura 11). Tal como o nome indica, as técnicas *model based* são baseadas em modelos, mais especificamente modelos virtuais 3D. Estas possuem duas fases: uma fase inicial *offline* onde informação do mundo real é obtida e posteriormente armazenada num modelo 3D, e uma segunda fase efetuada em velocidade interativa (*online*) que consiste na utilização desse mesmo modelo para estimar a pose da câmara. As técnicas *SFM based* baseiam-se nas *frames* provenientes da câmara para estimarem o movimento desta, resultando numa estimativa da sua pose. Este tipo de técnicas possui apenas uma fase *online* sem ser necessária qualquer informação prévia do mundo real: esta é adquirida durante o *tracking*.

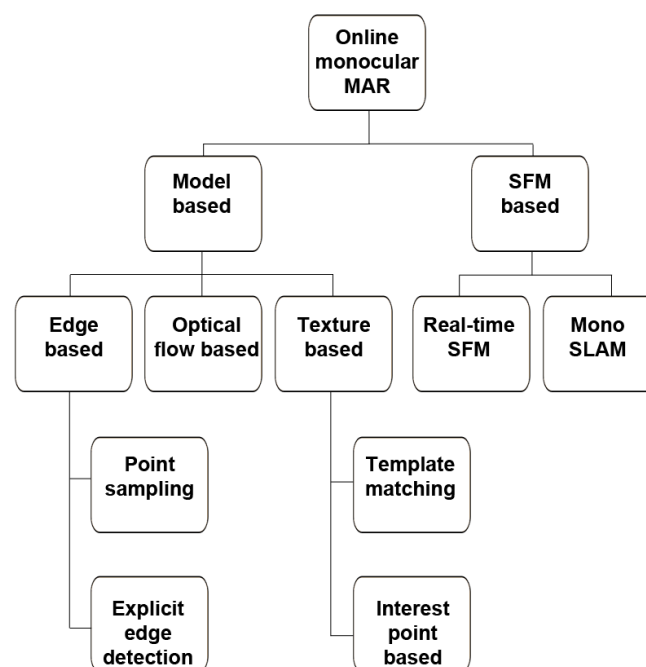


Figura 11 - Taxonomia de técnicas de *tracking* sem marcas para aplicações de RA. Adaptado de Teichrieb et al. (2007).

Em suma, existe um equilíbrio entre a inconveniência de construir um modelo 3D e o aumento de fiabilidade que este proporciona. A escolha de uma abordagem sobre a outra depende do objetivo da aplicação (Lepetit & Fua, 2005).

Model Based

A obtenção de informação geométrica 3D a partir de imagens 2D é um problema fundamental da área da visão por computador. De forma a calcular a pose de um corpo rígido, é necessário corresponder características de um modelo 3D virtual com parte das características visíveis na imagem 2D do mundo real, num processo referido como o problema da correspondência. Para tal, são extraídas características como pontos, linhas e elipses das imagens 2D do mundo real e explorado o modelo 3D assim como a pose do objeto (Doignon, 2007). Dentro do problema da correspondência existe o conceito de *inliers* e *outliers*. Segundo Lima (Lima et al., 2010), *inliers* representam as correspondências 2D-3D corretas quando se estima uma pose; *outliers* representam as correspondências incorretas. A existência de *outliers* pode arruinar a estimativa da pose mesmo que existam muitas *inliers*. Por este motivo é comum a utilização de métodos para reduzir a influência dos mesmos, sendo os mais comuns o RANdom SAMple Consensus (RANSAC) (Fischler & Bolles, 1981) e M-estimators (Lepetit & Fua, 2005). Segundo Comport (Comport et al., 2006), a principal vantagem de um método baseado em modelos é que o conhecimento sobre a cena (a informação 3D implícita) permite a melhoria da robustez e do desempenho, sendo capaz de prever o movimento escondido do objecto e atuar de modo a reduzir os efeitos de dados díspares (*outliers*) introduzidos no processo de *tracking*. No entanto, a informação 3D não está facilmente disponível em determinadas circunstâncias, por vezes é necessário fazer o cálculo da pose com menos informação sobre a cena. Nestes casos, o autor demonstra como se pode tirar vantagens das propriedades geométricas da cena e do sistema de visão para compensar a falta de um modelo 3D completo.

As técnicas de *tracking* baseadas em modelos (e.g. Marchand, Bouthemy, & Chaumette, 2001) requerem informação acerca do mundo real antes de iniciar o processo de *tracking*, tal como um modelo 3D que é depois usado para calcular a pose da câmara (Lima et al., 2010). Segundo Comport (Comport et al., 2006), uma vantagem das técnicas baseadas em modelos é a possibilidade de interação entre o mundo real e virtual, tal como a simulação de colisões entre objetos reais e virtuais e a oclusão de objetos virtuais por reais. O autor

enumera ainda algumas desvantagens das técnicas baseadas em modelos: não são totalmente *online*, é necessária uma fase prévia onde se obtêm os modelos; a inicialização do *tracker* é tipicamente manual ou requer um treino prévio para que seja automática; os objetos que são detetados com sucesso necessitam de estar sempre presentes.

Utilizando uma ou duas câmaras, aproximações baseadas em modelos podem reconhecer marcos (dado um modelo ambiental preciso) ou detetar movimento relativo dinamicamente entre *frames*. Há uma série de técnicas para detetar a geometria da cena (e.g. correspondência de *template*) e o movimento da câmara em 2D (e.g. fluxo ótico) e 3D (Krevelen & Poelman, 2010).

Além das técnicas de *tracking* do tipo baseadas em modelos apresentadas na Figura 11, Lima (Lima et al., 2010) refere adicionalmente as técnicas baseadas em pontos chave (keypoint based) como pertencentes a este tipo de técnicas (Figura 12). Na Figura 12 podemos constatar que o autor refere, ainda, dois tipos de categorias pelas quais se podem dividir as técnicas do tipo baseadas em modelos: recursiva, onde a última pose calculada é usada como uma estimativa para o cálculo da pose corrente; não recursiva/por deteção, onde a pose é estimada apenas com base na informação da *frame* atual, permitindo uma inicialização e recuperação automática.

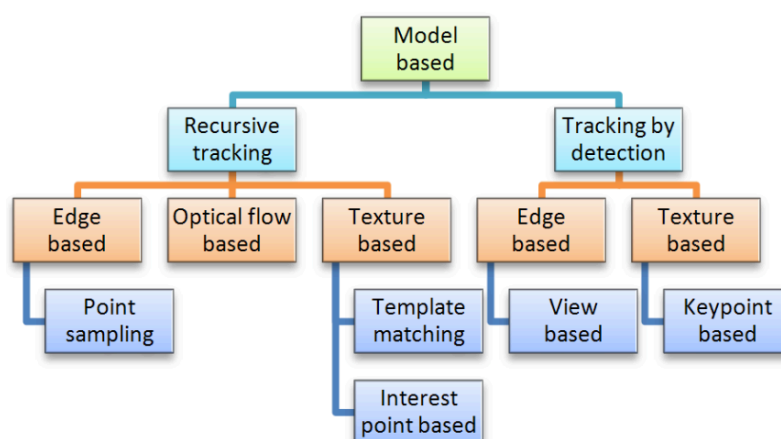


Figura 12 - Técnicas de *tracking* baseadas em modelos. Imagem obtida de Lima et al. (2010).

Tal como ilustram a Figura 11 e Figura 12, tendo em conta o tipo de característica utilizada para realizar o *tracking*, as técnicas baseadas em modelos podem ainda ser divididas em três categorias: baseadas em arestas, fluxo ótico e texturas. A primeira categoria consiste em métodos que apenas consideram arestas presentes na imagem do mundo real para efetuar o

tracking. A segunda categoria baseia-se no fluxo ótico da sequência de imagens provenientes da câmara. Por fim, a terceira categoria utiliza a informação de textura presente no objeto para realizar o *tracking*.

A Tabela 1 apresenta uma análise dos diferentes métodos apresentados por Lima (Figura 12) tendo em conta as seguintes métricas (Lima et al., 2010): detecção automática, que define se é, ou não, necessária uma interação do utilizador para obter uma estimativa da pose inicial; o peso computacional necessário para efetuar o *tracking*; precisão, que está relacionada com a exatidão das estimativas da pose da câmara ao longo das *frames* (nesta métrica, *jitter* refere-se ao tremer/instabilidade entre as estimativas da pose da câmara); e robustez, que define o quão resistente é o *tracker* às fontes de ruído.

Tabela 1 - Comparação entre os diferentes métodos de *tracking*. Adaptado de Lima et al. (2010).

Categoria	Método	Deteção	Processamento	Precisão	Robustez (Pontos fracos)
<i>Tracking</i> recursivo	Amostragem de pontos	Não	Baixo	<i>Jitter</i>	- Movimentos de câmara rápidos - Fundo confuso
	Baseado no fluxo ótico	Não	Baixo	Erros cumulativos	- Movimentos de câmara rápidos - Alterações na iluminação
	Correspondência de <i>template</i>	Não	Baixo	Elevada precisão	- Movimentos de câmara rápidos - Alterações na iluminação - Oclusões
	Baseado em pontos de interesse	Não	Elevado	Preciso	- Movimentos de câmara rápidos
<i>Tracking</i> por detecção	Baseado na visão	Sim	Elevado	Preciso	Intervalo de poses limitado
	Baseado em pontos chave	Sim	Elevado	<i>Jitter</i> e arrasto	Sem restrições

Técnicas baseadas em arestas foram a primeira implementação de técnicas de *tracking* baseadas em modelos devido à sua baixa complexidade e relativa facilidade de implementação. Como vantagens obtêm um bom desempenho e são robustas às diferenças na iluminação, sendo capazes de efetuar *tracking* em objetos especulares afetados pela iluminação ambiente (Teichrieb et al., 2007). As técnicas de *tracking* baseadas no fluxo ótico analisam as imagens capturadas pela câmara sequencialmente (tempo T e T+1). O algoritmo

determina quais os pontos do modelo projetado no tempo T permanecem na imagem no tempo $T+1$. O deslocamento destes pontos ao longo do tempo é utilizado para estimar o movimento da câmara. Uma vantagem é a obtenção de transições mais suaves entre as estimativas das poses. Outra vantagem é a sua necessidade de processamento moderada (Teichrieb et al., 2007). Por fim, nas técnicas baseadas na textura, a categoria *template matching* (correspondência de *template*) tem em conta uma região da imagem (informação global). Como vantagem consegue lidar com padrões complexos que seriam difíceis de detetar tendo em conta apenas informação local. A categoria *interest point based* (baseada em pontos de interesse) utiliza características locais. Por este motivo é menos pesada computacionalmente que a técnica *template matching* e lida melhor com as alterações na iluminação (Lima et al., 2010). Quanto à categoria *keypoint based* (baseada em pontos chave), esta sofre de arrasto e *jitter* ao calcular a pose da câmara. Ter em conta informação temporal reduz este problema. No entanto, o *tracking* por deteção tende a ser menos preciso que o *tracking* recursivo, devido à falta de precisão na correspondência (Lima et al., 2010).

Structure from Motion Based

Técnicas tradicionais SFM não têm em conta a restrição de velocidade interativa das aplicações de RM, estas eram utilizadas com o objetivo de obter reconstruções 3D a partir de sequências de imagens (Pollefeys, 1999). Técnicas de *tracking* baseadas em SFM são o resultado de diversas alterações ao processo do algoritmo tradicional de SFM (simplificações de algumas fases) para que este consiga ser executado em velocidade interativa, suportando assim a restrição de velocidade interativa das aplicações de RM (Teichrieb et al., 2007).

Contrariamente às técnicas baseadas em modelos, as técnicas baseadas em SFM não requerem informação prévia, dado que o movimento da câmara é estimado ao longo das *frames*. Isto permite que ambientes totalmente desconhecidos pelo sistema possam ser aumentados (Lepetit & Fua, 2005). Outra vantagem das técnicas baseadas em SFM, em relação às técnicas baseadas em modelos, é que estas não necessitam de ter o objeto/cena a que estão a fazer *tracking* sempre presente no campo de visão. Semelhante às técnicas baseadas em modelos, as técnicas baseadas em SFM podem fornecer informação acerca da cena para melhorar o sistema com características como oclusão de objetos virtuais pelos reais e interação física entre eles (Teichrieb et al., 2007). A autora refere, ainda, como desvantagem destas técnicas, o facto de que estas são frequentemente complexas de implementar. Segundo

Lepetit (Lepetit & Fua, 2005), outra desvantagem destas técnicas é a falta de controlo sobre o sítio onde a aumentação irá ocorrer, o que a torna impraticável em alguns sistemas. Uma vez que a restrição de velocidade interativa é suportada pelas técnicas baseadas em SFM, estas têm sido utilizadas em sistemas de RM (e.g. Lourakis and Argyros (2005)).

Outro algoritmo utilizado neste tipo de técnicas é o Simultaneous Location and Mapping (SLAM). O SLAM baseia-se no problema computacional de construir ou atualizar um mapa 3D de um ambiente desconhecido enquanto, em simultâneo, se mantém a par da localização do utilizador dentro deste. Uma destas implementações é o MonoSLAM, uma aproximação bem definida e conhecida pela comunidade robótica para construir uma representação instantânea do ambiente e estimar o movimento do robot (Davison, Reid, Molton, & Stasse, 2007). Na área da RM, Klein (Klein & Murray, 2007) propõe um sistema especialmente concebido para detetar a pose de uma câmara dentro de um pequeno espaço de trabalho. Dividindo as tarefas de *tracking* e de mapeamento em duas *threads* separadas numa metodologia que define como *Parallel Tracking and Mapping* (PTAM). O resultado é um sistema que produz mapas detalhados com milhares de pontos característicos que podem ser detetados em velocidade interativa, com uma precisão e robustez semelhantes à de sistemas com técnicas baseadas em modelos.

2.4.3 Híbridas

Há, também, a possibilidade de combinar diferentes técnicas de *tracking*, intituladas de aproximações híbridas, de forma a procurar colmatar os seus pontos fracos e combinar as suas forças (R. T. Azuma, 1997). Tais sistemas começaram a ser comercializados durante a década de 1990 e utilizavam maioritariamente técnicas baseadas em sensores para o *tracking* da posição e orientação (Van Krevelen & Poelman, 2010). Eventualmente, à medida que as técnicas de *tracking* baseadas em imagem se foram tornando mais robustas, foram combinadas com técnicas baseadas em sensores (e.g. Porzi, Ricci, Ciarfuglia, and Zanin (2012), Newcombe et al. (2011)) ou com outras técnicas baseadas em imagem (e.g. Vacchetti, Lepetit, and Fua (2004), Pressigout and Marchand (2006)). Segundo Höllerer (Höllerer & Feiner, 2004), as aproximações de *tracking* híbridas são as mais promissoras para lidar com os problemas apresentados por ambientes interiores e exteriores gerais de RM móvel.

Tal como apresentado na subsecção 2.3.1, existem sete aproximações tecnológicas para efetuar *tracking*. No entanto, a maioria delas possui limitações ao nível do intervalo,

precisão ou fluxo de dados. Por este motivo, é expectável o desenvolvimento de sistemas híbridos com vista a compensar as carências de uma única tecnologia, utilizando vários tipos de sensores para produzir resultados robustos.

Existem dois tipos de abordagens em sistemas que utilizem técnicas de *tracking* híbridas: podem ser utilizadas técnicas que façam parte da mesma tecnologia (e.g. duas técnicas baseadas na visão, baseada em arestas + baseado em texturas) ou combinar diferentes tecnologias (e.g. técnica baseada visão + técnica baseada em sensor), sendo utilizadas em simultâneo ou de forma alternada, onde uma delas é utilizada para inicialização, por exemplo.

No artigo de Newcombe (Newcombe et al., 2011) é apresentada uma solução que recorre ao Microsoft Kinect para efetuar o mapeamento e deteção (*tracking*) de superfícies densas. Esta é uma técnica de *tracking* híbrida com diferentes tecnologias uma vez que são utilizados, em simultâneo, uma câmara e um sensor de infravermelhos.

Vacchetti (Vacchetti et al., 2004) combina duas técnicas de *tracking* pertencentes à tecnologia de visão por computador, nomeadamente uma técnica baseada em arestas e técnica baseada em texturas de modo a conseguir fazer *tracking* sobre objetos com e sem textura. Segundo o autor, pelo facto desta abordagem explorar mais informação da imagem é mais estável e menos propensa ao arrasto que técnicas que apenas utilizem informação de arestas ou texturas.

2.5 Rendering

Numa aplicação de RM, a combinação e consequente apresentação dos modelos virtuais tridimensionais (3D) com imagens do mundo real, é efetuada através de um processo designado *rendering*.

Segundo Akenine-Möller, a função principal do *rendering* é gerar, ou renderizar, uma imagem bidimensional (2D), dada uma determinada câmara virtual, objetos 3D, fontes de luz, equações de sombras, texturas, entre outros componentes que definem uma cena (Akenine-Möller, Haines, & Hoffman, 2008). Existem duas técnicas de *rendering*: foto realista e não foto realista, do anglo-saxónico *photorealistic* e *non-photorealistic*, respetivamente (Figura 13) (Finkelstein, 2005).

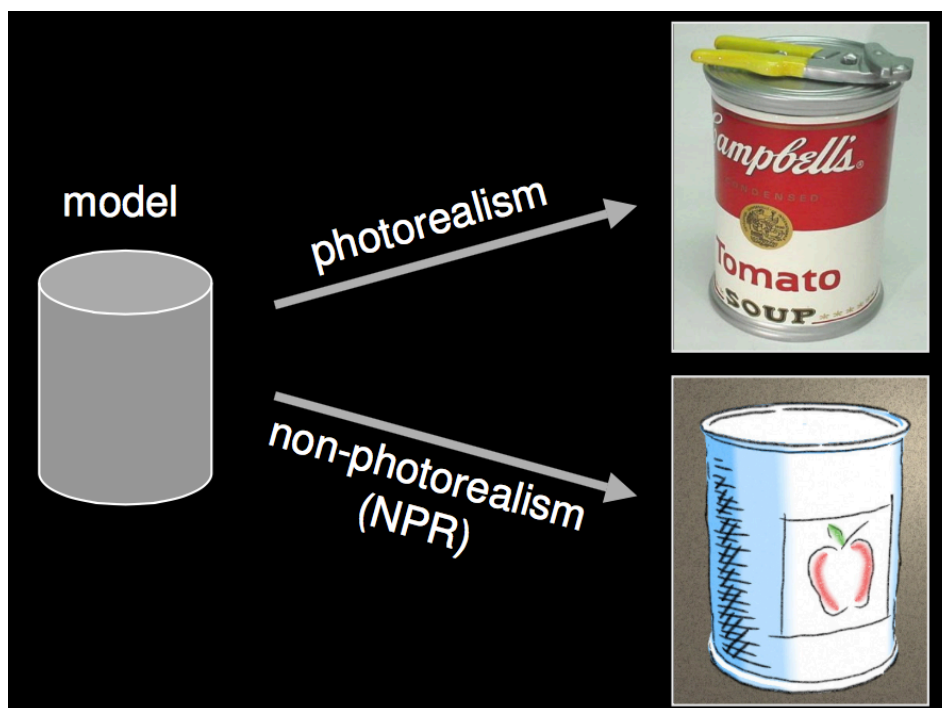


Figura 13 - Ilustração das duas técnicas de *rendering*: foto realista e não foto realista (Finkelstein, 2005).

O fotorrealismo é um género de arte que abrange pintura, desenho e outros meios gráficos, no qual um artista estuda uma fotografia e, em seguida, tenta reproduzir a imagem tão realista quanto possível sobre qualquer outra forma (Wikipedia, 2015a). Segundo Haller (Haller, 2004), ambas as técnicas de *rendering*, foto realistas e não foto realistas, são importantes e devem ser consideradas para a representação da RM. A escolha da técnica depende do tipo de aplicação e cada aplicação foca-se num determinado domínio. Como resultado, os utilizadores finais e os objetivos são diferentes, o que requerem diferentes técnicas de *rendering*. Na área da arqueologia virtual, Roussou (Roussou & Drettakis, 2003) refere as técnicas de *rendering* não foto realistas como mais apropriadas para instrumentos de investigação utilizados por arqueólogos e arquitetos, e técnicas de *rendering* foto realistas como mais apropriadas para um público mais amplo com propósitos de apresentação e educação.

O processo de *rendering* em si pode ser efetuado de duas formas (Akenine-Möller et al., 2008): *rendering* em velocidade não interativa, utilizado em animações de conteúdos multimédia não interativos (e.g. filmes). Permite a alavancagem de poder de processamento limitado a fim de obter uma imagem de maior qualidade; *rendering* em velocidade interativa, utilizado em conteúdos multimédia interativos (e.g. videojogos 3D). Baseia-se na utilização de placas gráficas com aceleradores de hardware 3D, para apresentar várias imagens por

segundo, a fim de a criar a ilusão de movimento ao olho humano. A velocidade do *rendering*, ou taxa de atualização de imagens, é medida em imagens por segundo (do anglo-saxônico Frames Per Second - FPS) ou Hertz (Hz), que é a notação 1/segundo (i.e. a frequência de atualização). Segundo o autor, a sensação de interação começa a crescer a partir dos 6 FPS. A partir dos 15 FPS, considera-se velocidade interativa, o utilizador foca-se na ação e reação. Como limite útil, é definido um valor de cerca de 72 FPS, onde as diferenças na taxa de visualização são efetivamente indetetáveis (Akenine-Möller et al., 2008).

A maioria do software de computação gráfica já tem o seu próprio *renderer*. No entanto, estes nem sempre são a melhor opção para cada projeto. Existe uma grande diversidade de *renderers* disponíveis no mercado atualmente, alguns exemplos destes são: Mental Ray, software proprietário desenvolvido pela Nvidia; Lumion, software comercial desenvolvido pela Act-3D; e LuxRender, software grátis *open-source* desenvolvido por uma equipa de cerca de 40 autores e contribuidores. A Figura 14 mostra um exemplo de uma imagem foto realista renderizada através do *renderer* LuxRender. A ligação entre os *renderers* e os softwares de computação gráfica, nomeadamente de modelação, é efetuada através de *plugins* desenvolvidos para o efeito.



Figura 14 - Imagem foto realista renderizada através do *renderer* "LuxRender". Imagem obtida de: http://www.luxrender.net/forum/gallery2.php?g2_itemId=12426

2.6 Modelação

A modelação 3D de um objeto pode ser vista como o processo completo que começa na aquisição de dados e termina com um modelo virtual 3D visualmente interativo num computador (Remondino & El - Hakim, 2006). De acordo com o autor, a modelação 3D de objetos e cenas é um problema de investigação intensiva e de longa duração nas comunidades gráfica, visão e fotogrametria. Embora seja fácil criar um modelo 3D simples, a geração de um modelo virtual preciso e foto realista de um objeto complexo requer um esforço considerável. Modelos 3D são necessários em muitas aplicações tais como inspeção, navegação, identificação de objetos, visualização e animação. Recentemente tornou-se num passo fundamental, em particular para a área da herança digital e arquivamento digital, uma vez que a modelação 3D permite a criação de recursos para educação, a criação de turismo e museus virtuais, documentação no caso de perda ou danificação e uma interação sem riscos de danificação.

Alguns exemplos de software de modelação são: 3ds Max, um software comercial para modelação 3D, animação e *rendering*. É desenvolvido pela Autodesk e suporta apenas o SO Windows; Cinema 4D, um software comercial direccionado para a concepção de gráficos 3D avançados. É desenvolvido pela MAXON e está disponível para os SO Windows e Mac OS. Os componentes de destaque do Cinema 4D são um motor de física, ferramentas de personagens avançadas e a possibilidade de utilizar uma rede de computadores de tamanho ilimitado para efetuar a tarefa de *rendering*; Por fim, blender, um software gratuito *open-source*. É desenvolvido pela Blender Foundation, uma organização pública independente sem fins lucrativos e está disponível para os três SO: Windows, Mac OS e Linux. Além disso, suporta na totalidade o *pipeline* 3D – modelação, montagem, animação, simulação, *rendering*, composição, deteção de movimento – e ainda edição de vídeo e criação de jogos.

2.7 SDK de RM

Um SDK consiste num conjunto de ferramentas de desenvolvimento de software que permitem a criação de aplicações para uma determinada plataforma de desenvolvimento (Wikipedia, 2015b). No caso de um SDK de RM, a ideia por detrás é tornar o processo de desenvolvimento de aplicações de RM mais simples e rápido.

Uma das dificuldades principais em desenvolver aplicações de RM é o problema de detetar (*tracking*) o ponto de vista do utilizador, normalmente definido como o problema de estimar a pose da câmara. A aplicação precisa de saber para onde o utilizador está a olhar no mundo real de forma a saber o ponto de vista correto para desenhar as imagens virtuais. Os SDK de RM facilitam esta tarefa, permitindo aos utilizadores a utilização/configuração de algoritmos que processem informações do mundo real com vista a extrair uma estimativa correta do ponto de vista do utilizador. Outra tarefa relevante levada a cabo pelos SDK de RM é a de sobrepor conteúdo virtual na imagem do mundo real, a qual é efetuada através de um processo de *rendering*. Para tal, os SDK podem fornecer o seu próprio *renderer* ou fornecer *plugins* para determinados *renderers* ou softwares de computação gráfica que se encarreguem dessa tarefa.

Atualmente, diversos SDK estão disponíveis no mercado. Um SDK com um elevado nível de maturidade na literatura é o ARToolKit (Kato & Billinghurst, 1999). O ARToolKit é um SDK de RM grátis *open-source*. Utiliza bibliotecas de *tracking* para calcular a posição e orientação real da câmara relativa a marcas fiduciais físicas em velocidade interativa. As suas características principais são: *tracking* da orientação e posição de uma câmara; *tracking* de quadrados pretos simples; possibilidade de utilizar qualquer padrão retangular; fácil calibração da câmara; disponibilidade para os SO: Windows, Linux, Mac OS e IRIX. A título de exemplo, a Figura 15 ilustra o funcionamento do mesmo.

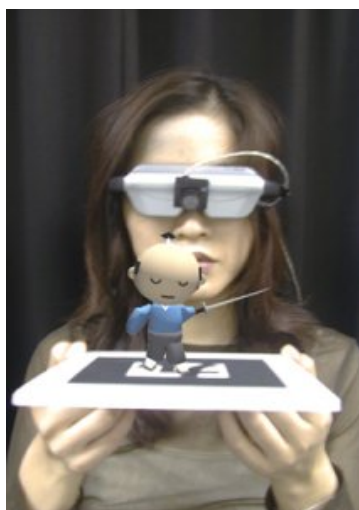


Figura 15 - Aumento de um objeto através de uma marca fiducial utilizando o ARToolKit. Imagem obtida de: <http://www.hitl.washington.edu/artoolkit/images/nakaohome.jpg>

Outro SDK disponível no mercado é o Wikitude, um SDK de RM comercial. Pode ser utilizado com uma licença comercial ou com uma licença de avaliação, sendo o único senão da versão de avaliação, a adição de uma marca de água em todas as aplicações que são produzidas. As características deste SDK são: *tracking* de imagens 2D; "Extended Image Tracking", uma técnica de *tracking* utiliza características do mundo real para manter o *tracking* mesmo quando o alvo já não se encontra no campo de visão da câmara; implementação de técnicas de *tracking* baseadas em sensores para produzir aplicações de RM baseadas na localização (e.g. exibição de pontos de interesse perto do utilizador); *rendering* de texto, vídeo, imagens 2D (estáticas ou animadas), modelos 3D (estáticos ou animados) ou trechos de HyperText Markup Language (HTML) sobre imagens do mundo real; suporte nativo para as plataformas Android, iOS e um número limitado de *smart glasses* de RM; possibilidade de desenvolver aplicações através de uma API de *JavaScript* fornecida pela Wikitude ou através das API nativas das plataformas; por fim, fornece dois *plugins*, um módulo e um componente para Unity3D, Cordova, Appcelerator Titanium e Xamarin, respetivamente (softwares que têm como objetivo permitir aos seus utilizadores desenvolverem o código necessário para uma aplicação apenas uma vez e difundir para diversas plataformas). Relativamente a *tracking* 3D, a página *web* oficial do SDK anuncia uma futura implementação de um algoritmo de SLAM para o efeito, não estando ainda disponível para o público.

Qualcomm Vuforia é um produto comercial desenvolvido pela Qualcomm Connected Experiences, Inc. Semelhante ao SDK Wikitude, pode ser utilizado com uma licença comercial ou com uma licença de avaliação, o que implica a presença de uma marca de água em todas as aplicações. Vuforia é um SDK de RM que tem como características: *tracking* de marcas fiduciais específicas do software; *tracking* de uma ou mais imagens 2D em simultâneo (na utilização de múltiplas imagens, é definido um arranjo geométrico entre estas); *tracking* de objetos 3D através de uma técnica de *tracking* baseada em modelos, direcionado para produtos de consumidor (e.g. brinquedos); "Smart Terrain", uma característica que permite a reconstrução 3D e posterior aumento de objetos através de uma técnica de *tracking* baseada em SFM; *tracking* de objetos com uma estrutura cilíndrica (e.g. canecas); reconhecimento de texto; "Extended tracking", ao contrário do Wikitude, que apenas suporta esta funcionalidade quando o alvo é uma imagem, o Vuforia consegue criar um mapa do ambiente com qualquer um dos alvos suportados, excepto marcas fiduciais ou reconhecimento de texto; por fim,

fornece o suporte para o desenvolvimento de aplicações nativas para Android e iOS e fornece um *plugin* para Unity3D, de forma a possibilitar a exportação das aplicações para mais plataformas.

O último SDK apresentado nesta secção é o Metaio. O Metaio é um SDK de RM comercial, pode ser utilizado com uma licença comercial ou de avaliação, tendo a mesma condição da marca de água que os SDK comerciais apresentados anteriormente. As suas características são: suporte nativo para Android, iOS, Windows e um número limitado de *smart glasses* de RM; fornece um *plugin* para que este seja utilizado com Unity3D, possibilitando assim a exportação das aplicações para um maior número de plataformas; *tracking* de marcas fiduciais específicas do software; *tracking* de uma ou mais imagens 2D (Ao contrário do Qualcomm Vuforia, este não precisa de definir uma relação geométrica. O *tracking* é feito de forma independente para cada imagem); "Extended Image Tracking", para continuar a efetuar o *tracking* do ambiente quando a imagem deixa de estar no campo de visão; leitura de códigos Quick Response Code (QR-CODE); implementação de uma técnica de *tracking* para superfícies deformáveis a fim de detetar a face de um ser humano; *tracking* de objetos e cenas 3D através de técnicas baseadas em modelos, baseadas em SFM ou híbridas (combinação dos dois tipos de técnicas); técnicas de *tracking* baseadas em sensores para produzir aplicações de RM baseadas na localização; *rendering* de texto, vídeo, imagens 2D (estáticas ou animadas), modelos 3D (estáticos ou animados) ou de um elemento *canvas* do HTML5 sobre imagens do mundo real; suporte para câmaras que obtêm dados de profundidade (e.g. Microsoft Kinect) a fim de obter um *tracking* mais preciso e gerar reconstruções 3D (funcionalidade limitada ao Windows); por último, define a Augmented Reality Experience Language (AREL), uma linguagem de *scripting* que permite a criação de experiências de RM, independente da plataforma, com base em tecnologias *web* comuns tais como HTML5, JavaScript e XML. Uma vantagem do Metaio em relação aos outros SDK é o seu nível de personalização, este permite que o processo de *tracking* se adeque da melhor forma ao problema em questão. A título de exemplo, as técnicas de *tracking* baseadas em modelos, para além de poderem ser combinadas com técnicas baseadas em SFM, têm a possibilidade de especificar quais os sensores, que pretendem utilizar para calcular a pose inicial do objeto/cena que a aplicação pretende detetar. Sendo possível escolher entre apenas acelerómetro; acelerómetro e compasso; acelerómetro, compasso e GPS.

Tabela 2 - Resumo das características principais entre os vários SDK apresentados.

SDK	Licença	<i>Tracking</i> baseado em sensores	<i>Tracking</i> baseado em imagens com marcas fiduciais	<i>Tracking</i> baseado em imagens sem marcas	
				Técnicas de <i>tracking</i> baseadas em modelos	Técnicas de <i>tracking</i> baseadas em SFM
ARToolKit	<i>Open-source</i>	Não	Sim	Não	Não
Wikitude	Comercial ou avaliação	Sim	Sim	Não	Não
Vuforia	Comercial ou avaliação	Sim	Sim	Sim	Sim
Metaio	Comercial ou avaliação	Sim	Sim	Sim	Sim

2.8 Aplicações de RM

Esta secção apresenta diversas aplicações de RM com foco em aplicações relacionadas com as áreas da arqueologia/herança cultural e turismo. Os sistemas que suportam estas aplicações são apresentados juntamente com um resumo das suas aproximações para efetuarem a tarefa de *tracking*.

Kretschmer (Kretschmer et al., 2001) descreve o sistema GEIST (termo alemão para fantasma). Um sistema que envolve o utilizador numa história interativa enquanto o mesmo explora sítios de herança cultural. Para tal utiliza diversas tecnologias, entre elas, a RM móvel (em ambiente exterior). O *tracking* é efetuado com recurso a uma técnica de *tracking* híbrida, que se baseia numa combinação de sensores e informação contida num modelo 3D do ambiente.

LIFEPLUS é um projeto que propõe a reconstrução 3D de pinturas *frescos* históricas. Um dos seus objetivos é recriar a fauna e flora (humanos, animais e plantas), presente nas pinturas, através de conteúdos 3D animados em velocidade interativa, num ambiente de RM imersivo. Para situar o utilizador no espaço e apresentar informação multimédia de acordo, pretendem utilizar um GPS diferencial e um compasso digital. Quando o utilizador se aproxima de um ponto de interesse, a pose da câmara é determinada com uma maior precisão através de técnicas de *tracking* baseadas na visão, podendo opcionalmente utilizar como suporte o GPS diferencial e compasso digital (Papagiannakis et al., 2002).

Vlahakis (Vlahakis et al., 2002) apresenta o ARCHEOGUIDE, um sistema móvel que tem como objetivo atuar como um guia eletrónico em visitas exteriores a sítios arqueológicos. O sistema utiliza a RM para melhorar a apresentação da informação, reconstruir ruínas, e simular vida antiga nestes sítios. Quanto ao processo de *tracking*, o ARCHEOGUIDE utiliza, inicialmente, um GPS diferencial, através do qual é obtida uma estimativa inicial da pose da câmara e qual a ruína que o utilizador está a visualizar. A estimativa é depois refinada através de uma técnica de *tracking* baseada na visão sem marcas que tenta corresponder imagens 2D do mundo real com imagens 2D das ruínas, obtidas anteriormente numa fase de preparação.

Mourkoussis (Mourkoussis et al., 2002) apresenta o projeto Augmented Reality In Cultural Heritage (ARICH). Este é um sistema concebido para ambientes interiores que pretende possibilitar a visualização de modelos virtuais de artefactos arqueológicos no mundo real através da RM. A integração dos modelos 3D no vídeo do mundo real é efetuada com base numa técnica de *tracking* baseada em imagens com marcas, fornecida pela biblioteca de software ARToolKit. O mesmo autor apresenta o projeto Augmented Representation of Cultural Objects (ARCO), um projeto cujo objetivo principal é a criação automatizada de objetos culturais virtuais através de fotogrametria. A ideia é permitir que pessoas sem conhecimentos técnicos na modelação 3D consigam obter uma representação virtual de um objeto cultural para que possa, dessa forma, ser partilhado para o público geral e investigadores da área da arqueologia.

Honkamaa (Honkamaa, Siltanen, Jäppinen, Woodward, & Korkalo, 2007) implementa um sistema que permite aumentar ambientes exteriores com qualquer modelo 3D, tornando-o genérico de forma a poder ser aplicado a um maior número de áreas. Para obter o ponto de vista do utilizador, o sistema utiliza uma técnica de *tracking* baseada em imagens sem marcas. Apesar deste sistema conseguir uma precisão ao nível do subpíxel em termos de registo durante o *tracking*, diversas simplificações foram efetuadas para que tal ocorresse, nomeadamente:

- o posicionamento inicial do objeto é selecionado de forma interativa pelo utilizador e não de forma automática pela técnica de *tracking*;
- é feita a assunção de que o utilizador permanece quieto no mesmo sítio, desta forma apenas 3 graus de liberdade (rotações da câmara ao longo de cada eixo)

precisam de ser detetados, o que implica a reinicialização do sistema sempre que o utilizador altera a sua posição de visualização;

- é também feita a assunção de que está a visualizar algo relativamente distante à lente focal da câmara.

Miyashita (Miyashita et al., 2008) apresenta um sistema de RM que serve como um guia para visitantes do museu do Louvre. As suas funções principais são guiar o visitante através de uma determinada exposição numa ordem específica assim como fornecimento de informação sobre a obra de arte em visualização. O sistema baseia-se em técnicas de *tracking* baseadas em imagens sem marcas, no caso da visualização da obra de arte, e numa técnica de *tracking* híbrida, onde informação da câmara é utilizada em conjunto com sensores inerciais, para guiar o utilizador pelas várias obras presentes na exposição.

Zöllner (Zöllner, Keil, Wüst, & Pletinckx, 2009) implementa um sistema de RM que tem como objetivo apresentar informação sobre sítios de herança cultural em locais remotos, sendo museus a principal aplicação. O seu objetivo é fornecer uma contextualização dos sítios de origem dos artefactos aos visitantes através de informação contextual tal como texto, imagens, vídeos, desenhos históricos e reconstruções 3D. Para que os utilizadores possam visualizar estes conteúdos são utilizados ecrãs amovíveis (360° em volta de um eixo vertical) ou computadores ultra móveis (do anglo-saxónico Ultra Mobile PC). Quanto à obtenção da pose da câmara, é utilizada uma técnica de *tracking* baseada em imagens sem marcas.

Neste capítulo foram abordados os diversos conceitos relevantes para a realização desta dissertação: RV, RM, RA, VA, *tracking*, *rendering* e modelação. Foram, também, apresentados os diversos tipos de técnicas de *tracking* existentes juntamente com as suas vantagens e desvantagens. Foi efetuada uma apresentação e comparação de alguns SDK de RM disponíveis no mercado. Por último, foram apresentadas diversas aplicações de RM presentes na literatura.

3 Projeto MixAR

Este capítulo descreve o projeto de investigação e desenvolvimento, em copromoção entre a UTAD e a empresa GEMA DIGITAL, onde o trabalho desenvolvido e apresentado nesta dissertação se enquadra. O MixAR é um projeto que visa o desenvolvimento de um sistema de RM adaptativa. O seu objetivo principal é a visualização de reconstruções virtuais, historicamente corretas, de sítios arqueológicos, devidamente sobrepostas e alinhadas nas suas estruturas atuais, caso subsistam, ou no local onde outrora existiram. Ao longo do tempo, os sítios arqueológicos foram sendo danificados ou totalmente destruídos. Este projeto pretende possibilitar a qualquer utilizador a observação do esplendor inicial de estruturas presentes, em tais sítios, através da sobreposição das suas reconstruções virtuais. Para tal, Magalhães (Magalhães et al., 2014) definiu uma nova metodologia para adaptar o nível de mistura, entre o real e virtual, enquanto o utilizador se movimenta, livremente, pelo sítio arqueológico. Esta adaptação é guiada pelo tipo de cena, interior ou exterior, pela posição do utilizador e pelo estado da estrutura que se pretende detetar (*tracking*) e posteriormente aumentar com conteúdo virtual. Quando o utilizador se encontra numa cena exterior é utilizada uma aproximação de RA, onde o mundo real prevalece. Quando se encontra numa cena interior é utilizada uma aproximação de VA, onde o mundo virtual prevalece. A utilização de uma aproximação de VA para os interiores prende-se com o facto de que esta, contrariamente à RA, permite que as condições de iluminação sejam controladas, visto que o que prevalece é a cena virtual e, portanto, a iluminação definida para a cena. Trata-se de uma capacidade relevante para o sistema pois a exposição direta ao sol ou a presença de luzes elétricas nos interiores das estruturas contribui para um decréscimo da experiência visual, não permitindo que os visitantes tenham uma verdadeira perceção da aparência dos interiores dos edifícios na altura em que eram habitados. A mudança entre os dois tipos de cena deve ser executada de uma forma suave e em velocidade interativa, garantindo que o utilizador a percecione como natural.

3.1 Enquadramento

Uma das tarefas mais importantes num sistema de RM é o registo geométrico entre a cena real e virtual, o qual assegura que a cena virtual é corretamente alinhada com a vista do mundo real correspondente. Se isto não acontecer, pode-se perder a ilusão de que os dois

mundos, virtual e real, coexistem no mesmo espaço (R. T. Azuma, 1997). Para tal, é necessário obter a posição e orientação do ponto de vista do utilizador relativo ao mundo real, ou vice-versa, o que implica uma operação de *tracking* (Magalhães et al., 2014).

O trabalho desenvolvido ao longo desta dissertação teve como objetivo o desenvolvimento da componente de *tracking* do MixAR. Após uma análise, apresentada no capítulo 2, dos diversos tipos de técnicas de *tracking* existentes, consideraram-se como sendo as mais adequadas, para o projeto MixAR, as técnicas de *tracking* baseadas na visão, pois ao invés das técnicas de *tracking* baseadas em sensores, que seguem uma abordagem *open-loop*, as técnicas de *tracking* baseadas na visão seguem uma abordagem *closed-loop*, o que lhes permite obter um alto nível de precisão na tarefa do registo. Acresce, ainda, que devido ao baixo custo de câmaras, as técnicas de *tracking* baseadas na visão permitem suportar soluções de *tracking* económicas. De entre as técnicas de *tracking* baseadas na visão estudadas, consideraram-se como mais adequadas aos requisitos e condicionantes do projeto MixAR, as técnicas baseadas na visão sem marcas. Com efeito e contrariamente às técnicas de *tracking* baseadas na visão sem marcas, as técnicas de *tracking* baseadas na visão com marcas são intrusivas pois necessitam de uma preparação prévia do ambiente onde se pretende realizar a operação de *tracking*, através da adição de marcas fiduciais ou LED's. A utilização deste tipo de técnicas de *tracking* com marcas é aceitável em casos onde a alteração do ambiente não seja um problema. Contudo, no caso do projeto MixAR, o ambiente deve permanecer inalterado devido ao valor cultural e histórico dos sítios arqueológicos. Além disso e tendo em consideração que a ocultação de uma marca pode levar a que deixe de ser possível estimar a pose da câmara, a maior robustez das técnicas de *tracking* sem marcas é outro fator importante na seleção das técnicas. Por último, utilizando técnicas baseadas em marcas, o *tracking* é, apenas, efetuado, quando as marcas estão perto o suficiente para serem detetadas. Isto torna-as não apropriadas para ambientes com grandes edifícios, como é comumente o caso das estruturas presentes em sítios arqueológicos.

3.2 Arquitetura e Funcionamento

A arquitetura do projeto MixAR obedece ao modelo cliente-servidor (Figura 16) e é constituída por vários módulos funcionais, nomeadamente: uma ou mais unidades móveis, constituídas por um componente de visualização, um de contextualização, um de processamento e um repositório local. São transportadas pelos utilizadores e têm como

objetivo gerir e proporcionar a experiência de RM ao utilizador; um servidor de alto desempenho – servidor de RM – constituído por um componente de processamento, uma ferramenta de autoria para o Sistema de Informação Geográfico (SIG) e um repositório global. A sua finalidade é armazenar, gerir e fornecer dados relevantes à experiência, assim como agir como unidade de processamento remoto; por fim, a infraestrutura de rede que suporta a comunicação entre o servidor e o(s) cliente(s).

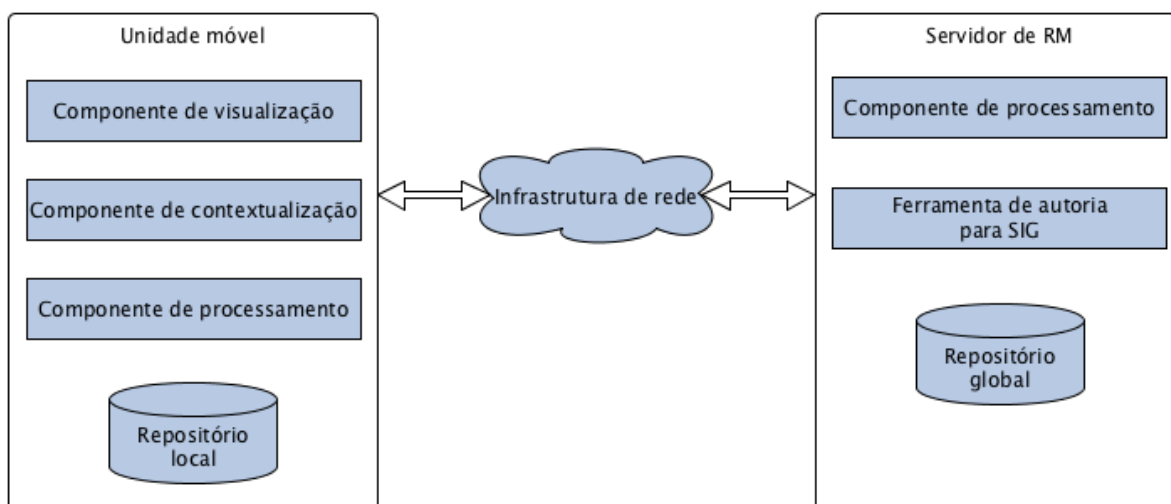


Figura 16 - Arquitetura geral do projeto MixAR. Descrição dos módulos funcionais presentes na arquitetura: unidade móvel, servidor de RM e infraestrutura de rede.

3.2.1 Servidor de RM

O servidor de RM é composto por um componente de processamento, um repositório de dados global e uma ferramenta de autoria para SIG. O componente de processamento é invocado, pela unidade móvel, em três situações distintas:

- para determinar se a unidade móvel possui o conteúdo necessário para a experiência de RM e se este está na última versão;
- para calcular o edifício mais próximo do utilizador com base na sua posição;
- para devolver uma lista com as identificações dos modelos virtuais que irão aumentar os edifícios nos arredores do utilizador, segmentados por nível de detalhe.

O repositório de dados é utilizado para armazenar e suportar o acesso a diversos conteúdos necessários para a experiência de RM, nomeadamente os modelos virtuais e

respetivas configurações de *tracking*. Por último, a ferramenta de autoria, irá permitir, ao gestor da experiência, a gestão das experiências de RM, através da criação e configuração de um SIG para o sítio arqueológico. O SIG irá conter informação acerca de todas as estruturas utilizadas para as experiências de RM, assim como as suas localizações georreferenciadas. Esta informação irá ser utilizada para identificar se o utilizador está num tipo de cena interior ou exterior e para determinar os modelos virtuais a apresentar-lhe. O modelo de dados que suporta o SIG é apresentado na Figura 17.

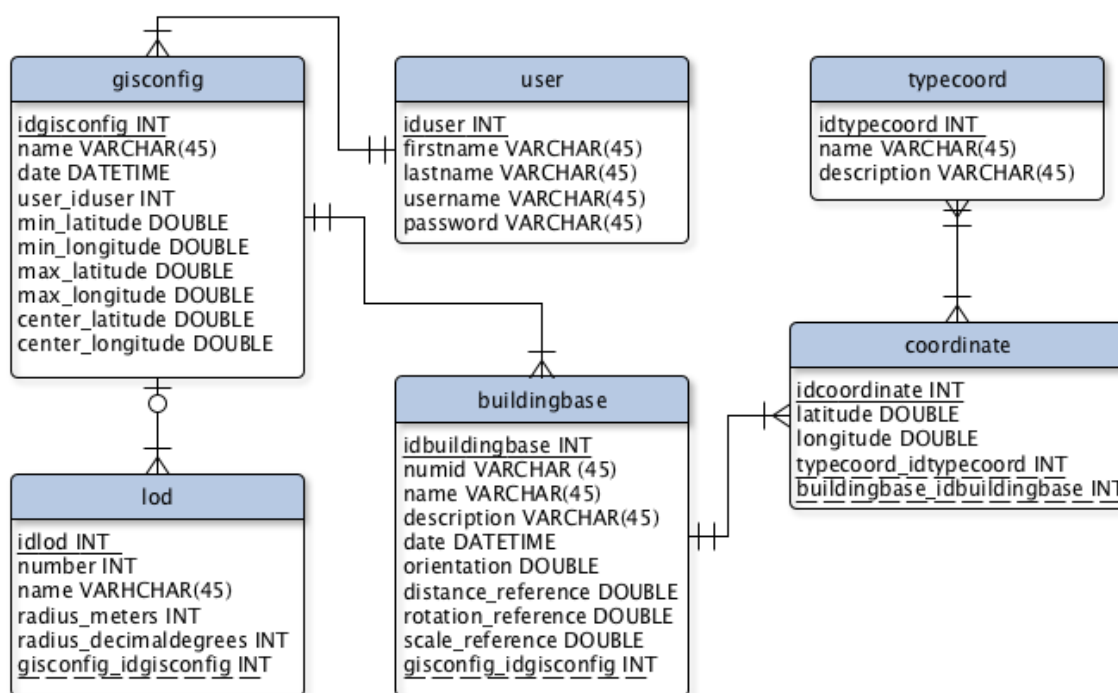


Figura 17 - Apresentação das diversas tabelas, e as suas propriedades, que formam o modelo de dados que suporta o SIG, desenvolvido no âmbito do projeto MixAR.

Cada experiência de RM é armazenada pela tabela "*gisconfig*". Esta serve, principalmente, para armazenar os limites, em termos de porção de terreno, onde irá ocorrer uma determinada experiência de RM. Os limites são armazenados na forma de um retângulo, em que os campos "min_latitude" e "min_longitude" definem o canto superior esquerdo do retângulo, os campos "max_latitude" e "max_longitude" o canto inferior direito e, por fim, os campos "center_latitude" e "center_longitude" definem o seu centro. A tabela "*user*" possui alguns dados pessoais sobre os utilizadores do SIG (i.e. gestores de experiências) e as suas credenciais, utilizadas para gerirem as experiências de RM. A tabela "*buildingbase*" contém informação acerca dos edifícios utilizados na experiência. A tabela "*coordinate*" armazena, através dos campos "latitude" e "longitude", todas as coordenadas que constituem as bases

dos edifícios, assim como uma coordenada central para cada um destes. O tipo da coordenada, central ou de contorno, é definido na tabela "*typecoord*". Por fim, a tabela "*lod*" armazena, em metros e em ângulos decimais, o raio de cada nível de detalhe. Utilizando a posição do utilizador como ponto de partida, os raios, definidos na tabela *lod*, são utilizados para determinar o nível de detalhe com que cada edifício deve ser renderizado. Os níveis de detalhe (do anglo-saxónico Level of Detail, LOD) são uma estratégia comum na computação gráfica, que envolve a diminuição da complexidade da representação de um objeto 3D à medida que este está mais afastado do espetador, aumentando, assim, a eficiência do *rendering*.

3.2.2 Infraestrutura de Rede

Relativamente à infraestrutura de rede, esta irá seguir, tipicamente, uma topologia em malha sem fios, para permitir uma comunicação entre os Access Points (AP) e a criação de *hotspots* locais, tentando obter uma cobertura eficaz de todo o sítio arqueológico. Para obter uma baixa latência e bom desempenho da rede, será utilizado o protocolo IEEE 802.11n (65 Mbps - 450 Mbps) ou, se possível, outro com melhores características tal como o IEEE 802.11ac (1.3Gbps).

3.2.3 Unidade Móvel

São consideradas duas configurações para a unidade móvel: imersiva e não-imersiva. A imersiva irá consistir num Head-Mounted Display (HMD), com câmara, utilizando uma tecnologia *optical* ou *video see-through*, assim com num conjunto de sensores ligados a um dispositivo de processamento. A configuração não-imersiva irá ser composta por um *tablet* com câmara e diversos sensores (acelerómetro, giroscópio, magnetómetro e GPS). Cada unidade móvel, independentemente da sua configuração, será composta por um repositório de dados local, e três componentes: de visualização, de contextualização e de processamento (Figura 16). O repositório de dados local consiste numa cópia completa do repositório de dados global, presente no servidor de RM. O objetivo é acelerar o acesso à informação e reduzir a carga da rede de comunicação, devido a acessos e transferências de ficheiros constantes do servidor. O componente de visualização irá permitir ao utilizador ver o conteúdo virtual sobreposto à imagem do mundo real. Será composto por um ecrã, no caso do HMD *video see-through* e do *tablet*, e por espelhos, com uma determinado nível de

transparência, no caso do HMD *optical see-through*. O componente de contextualização será constituído por uma câmara e sensores inerciais, magnéticos e de GPS, ligados ao HMD ou *tablet*. Por último, o componente de processamento será formado por um *Single Board Computer* (SBC), no caso da configuração imersiva, ou pelo próprio *tablet*, no caso da configuração não-imersiva.

A interação entre os componentes que constituem a unidade móvel é apresentada através de um diagrama de fluxo, na Figura 18. Inicialmente, o componente de contextualização captura, através da câmara e sensores, informação acerca do mundo real. Esta informação é enviada para o componente de processamento, o qual é responsável por executar os algoritmos de *tracking* sobre esta informação o que, em caso de sucesso, produz uma estimativa da pose da câmara. Com base na estimativa da pose da câmara, o componente de visualização procede no sentido de apresentar a cena de RM, onde o conteúdo virtual é aumentado na imagem do mundo real. No caso de não haver uma estimativa da pose da câmara, por falta de informação por parte do componente de contextualização ou por qualquer outro motivo, o componente de visualização apresenta apenas a imagem do mundo real.

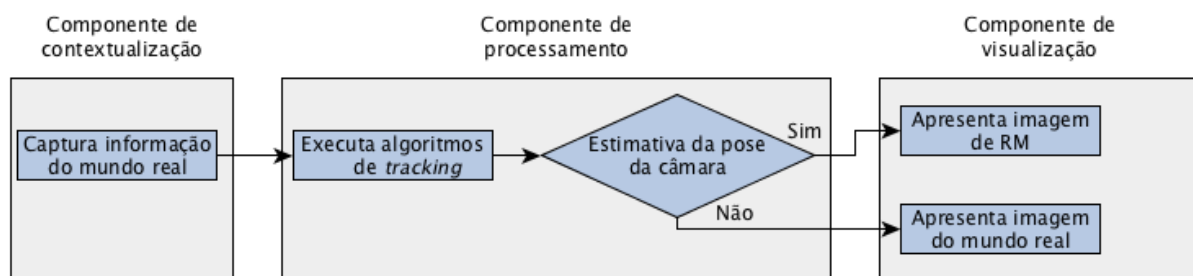


Figura 18 - Diagrama de fluxo da interação entre os componentes que constituem a unidade móvel. Apresentação das tarefas executadas em cada componente da unidade móvel no âmbito do processo de deteção (*tracking*) e aumento das estruturas do mundo real.

As tarefas, da unidade móvel serão realizadas através da instalação de uma aplicação denominada MixAR App, desenvolvida no âmbito do projeto MixAR. A descrição das tarefas juntamente com o diagrama de fluxo da MixAR App são apresentadas na secção 4.1.

3.3 Seleção de Equipamento e Ferramentas de Desenvolvimento

Tendo em conta os requisitos das unidades móveis, apresentados na subsecção 3.2.3, foi selecionado e adquirido o seguinte material: HMD Vuzix STAR 1200XLD; HMD Vuzix 1200DXAR; GPS Navicom NAV-GP01S; SBC Eurotech ANTARES; *tablet* Samsung SM-

T320. Isto possibilitou a criação de três configurações para a unidade móvel, ao invés das duas definidas na especificação inicial do projeto: uma não-imersiva, constituída pelo *tablet*, e duas imersivas, uma constituída por um HMD com uma tecnologia *optical see-through* (Vuzix STAR 1200XLD) e uma com um HMD *video see-through* (Vuzix 1200DXAR), pelo GPS NAV-GP01S e pelo SBC Eurotech ANTARES.

Relativamente à seleção das ferramentas de desenvolvimento foram tidos em consideração os requisitos do projeto MixAR, a saber: suporte para os Sistemas Operativos (SO) Windows, Android e iOS; implementação de técnicas de *tracking* baseadas em imagens sem marcas baseadas em modelos e baseadas em SFM; suporte para sensores inerciais e GPS. Com base nestes requisitos, as funcionalidades de diversos SDK de RM disponíveis foram analisadas, para seleccionar qual deles seria o mais indicado para o projeto MixAR. Os SDK considerados mais relevantes foram já apresentados na secção 2.7. Foi seleccionado o Metaio SDK, pois preenche todos os requisitos do projeto MixAR, tem um maior número técnicas baseadas em imagens sem marcas (baseadas em modelos e baseadas em SFM) implementadas e permite um maior nível de personalização das técnicas de *tracking*. Além disso, o Metaio SDK implementa uma técnica híbrida, onde duas técnicas de *tracking*, de tipos diferentes (baseadas em modelos e baseadas em SFM), são combinadas, para obter um processo de *tracking* mais preciso e robusto. Outro fator que levou à escolha do Metaio SDK foi o facto de este possuir um *plugin* para o Unity3D, uma plataforma de desenvolvimento robusta e flexível, usada para criar jogos e experiências interativas multiplataforma em 2D ou 3D. O Unity3D tornou-se popular pela facilidade em compilar projetos para diversas plataformas, suportando, até à presente data, 21 plataformas, na sua última versão (5.1.2) (Unity, 2015). Acresce, ainda, que, o Unity3D é disponibilizado para desenvolvimento nos três SO: Windows, OS X e Linux. Assim, no âmbito do projeto MixAR, escolheu-se utilizar o Metaio SDK juntamente com o Unity3D, permitindo que a aplicação desenvolvida possa ser facilmente exportada para os três SO exigidos na especificação do projeto MixAR: Windows, Android e iOS.

3.4 Técnicas de Tracking Disponíveis no Metaio SDK

O Metaio SDK, na sua última versão (6.0.2), disponibiliza um total de 10 técnicas de *tracking*. Esta secção tem como objetivo apresentar cada uma destas técnicas bem como algumas vantagens e desvantagens das mesmas.

3.4.1 ID Marker

Esta técnica de *tracking* baseia-se em marcas fiduciais pré-definidas pelo Metaio SDK sendo direcionada para aplicações de RM simples. Como vantagem, suporta o reconhecimento de, até, 512 marcas (número total de marcas fiduciais pré-definidas existentes), mantendo um bom nível de desempenho. Cada marca tem um número de identificação associado. Como exemplo, a marca com o número de identificação 1 é apresentada na Figura 19.

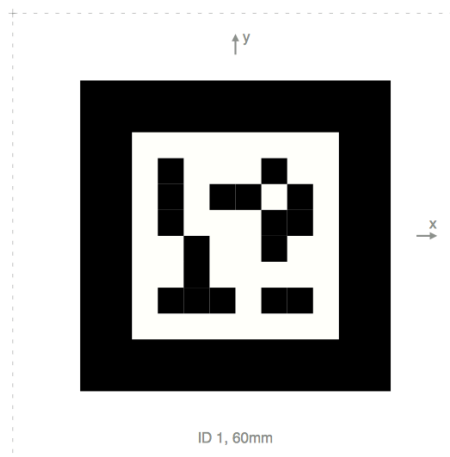


Figura 19 - Marca fiducial como número de identificação 1 pré-definida no Metaio SDK.

A técnica tem dois modos de execução distintos: "Fast" e "Robust". O modo de execução "Fast" é orientado ao desempenho tendo, como único requisito, a utilização em ambientes cujas condições de iluminação permaneçam estáticas ou não variem muito. No que respeita ao modo de execução "Robust", adapta-se constantemente aos níveis de iluminação, sendo, por isso, mais robusto, mas apresenta, como desvantagem, um desempenho inferior ao do modo de execução "Fast".

3.4.2 Picture Marker

Esta técnica de *tracking* utiliza uma imagem como marca e pode ser definida como uma extensão da técnica ID Marker. Qualquer imagem pode ser utilizada como referência para o *tracking* desde que algumas restrições sejam observadas: deve ser suficientemente texturizada, o seu contorno deve ser constituído por uma cor escura e a imagem deve ser apresentada sobre um fundo claro, de forma a criar um grande contraste e assim facilitar a sua deteção. Um exemplo de uma imagem que respeita estas restrições é apresentada Figura 20.

Tal como a técnica ID Marker, a Picture Marker possui os dois modos de *tracking* "Fast" e "Robust", já descritos na subsecção anterior.



Figura 20 - Exemplo de uma imagem indicada para ser utilizada como marca quando se recorre à técnica de *tracking* Picture Marker. Imagem obtida de:
http://dev.metaio.com/fileadmin/user_upload/documents/sdk/picturemarker.pdf

3.4.3 Image Tracking

Image Tracking é um técnica de *tracking* baseada em imagens sem marcas. De forma semelhante às duas técnicas já apresentadas, dispõe de dois modos de *tracking*: "Fast" e "Robust". O modo "Fast" é utilizado quando é pretendido um desempenho fluído e a imagem de referência a detetar é moderadamente texturizada. Quando os resultados com o modo "Fast" não são satisfatórios, pode recorrer-se ao modo "Robust", indicado para imagens com elevado nível de textura. Na Figura 21 é apresentado um exemplo de uma imagem indicada para esta técnica de *tracking*.



Figura 21 - Exemplo de uma imagem indicada para ser utilizada como marca quando se recorre à técnica Image Tracking. Imagem obtida de:
http://dev.metaio.com/fileadmin/user_upload/documents/sdk/markerless.pdf

3.4.4 3D Map Tracking

Esta técnica permite que qualquer objeto/cena do mundo real seja utilizado como referência para efetuar o *tracking*. O único requisito é a criação de um mapa de pontos 3D do objeto/cena numa fase prévia ao *tracking*. Um mapa 3D é criado através da aplicação Toolbox, desenvolvida pelo Metaio, disponível para iOS e Android. O que esta aplicação faz é recolher determinados pontos característicos, no espaço 3D, com base na textura da imagem do mundo real. A junção de todos os pontos representa o mapa 3D, utilizado na fase *online* do *tracking* para determinar a pose do objeto/cena.

3.4.5 Instant Tracking

As operações de *tracking* desta técnica são realizadas sem informação prévia sobre o ambiente. Essa informação é adquirida, em velocidade interativa, quando a técnica começa a ser executada. Ao contrário das outras técnicas, que são inicializadas/configuradas através de um ficheiro .XML, esta é inicializada através da função "startInstantTracking", presente na API do Metaio SDK. A função é parametrizável de forma a que o utilizador possa escolher um dos vários modos de execução da técnica, descritos na Tabela 3.

Tabela 3 - Diferentes modos de execução suportados pela técnica Instant Tracking. Descrição das alterações efetuadas à técnica de *tracking* entre os diferentes modos de execução apresentados.

Modo	Descrição
INSTANT_2D	Cria uma configuração de <i>tracking</i> instantânea para um alvo 2D plano.
INSTANT_2D_GRAVITY	Semelhante ao modo INSTANT_2D, adiciona o uso de um acelerómetro para ajudar a determinar a pose.
INSTANT_2D_GRAVITY_EXTRAPOLATED	Semelhante ao modo INSTANT_2D_GRAVITY, adiciona o uso de um giroscópio para que a pose possa ser determinada mesmo que o alvo não esteja a ser detetado visualmente.
INSTANT_2D_GRAVITY_SLAM	Semelhante ao modo INSTANT_2D_GRAVITY, adicionalmente utiliza 3D <i>tracking</i> (SLAM) para continuar a fazer <i>tracking</i> quando a imagem de referência sai do campo de visão.
INSTANT_2D_GRAVITY_SLAM_EXTRAPOLATED	Semelhante ao modo INSTANT_2D_GRAVITY_SLAM, utiliza adicionalmente o giroscópio para determinar a pose, mesmo quando o alvo não esteja a ser detetado visualmente.
INSTANT_3D	Também denominada SLAM, cria uma configuração de <i>tracking</i> instantânea para um alvo 3D.

A técnica INSTANT_3D, ou SLAM, tem como objetivo a construção/atualização de um mapa 3D enquanto, em simultâneo, localiza o utilizador dentro do mesmo. A implementação desta técnica, pelo Metaio, explora características específicas, baseadas em textura, do ambiente envolvente, para criar um mapa 3D em velocidade interativa, enquanto obtém a pose da câmara dentro do mesmo.

3.4.6 Extended Image Tracking

A implementação da técnica de *tracking* SLAM pelo Metaio SDK apresenta alguns problemas causados pelo facto de não existir informação acerca do objeto/cena *a priori*, a saber: a falta de controlo sobre a localização e a falta de precisão sobre a escala do conteúdo virtual que irá aumentar o objeto/cena. A técnica Extended Image Tracking (combinação das técnicas Image Tracking e SLAM) surge como uma solução para estes problemas. Esta técnica utiliza uma imagem-alvo para iniciar a criação do mapa 3D. Dado que o seu tamanho é conhecido, basta detetar a posição e orientação da imagem para que o mapa 3D comece a ser construído e o conteúdo virtual seja corretamente escalado e posicionado sobre a imagem do mundo real.

3.4.7 Face Tracking

O Metaio suporta, também, técnicas de *tracking* para objetos não rígidos, tal como a face de um ser humano. No entanto, estas não são tão exploradas quando comparadas com as restantes técnicas e, como tal, as suas funcionalidades são muito limitadas. A técnica Face Tracking efetua uma estimativa da posição de uma única face. Fora das suas capacidades está a deteção da orientação ou a deteção de várias faces.

3.4.8 Latitude Longitude Altitude (LLA) Markers

Esta técnica de *tracking* é semelhante à técnica ID Marker. A diferença está na adição de informação codificada acerca da localização do alvo. Após a deteção da localização, a técnica tem como objetivo guiar o utilizador ao seu alvo, conforme apresenta a Figura 22. LLA é um formato utilizado para definir posições no globo, onde a latitude e longitude são medidas em graus e a altitude medida em metros acima do nível do mar. O Metaio SDK utiliza, como sistema geodésico, o World Geodetic System 84 (WGS84).

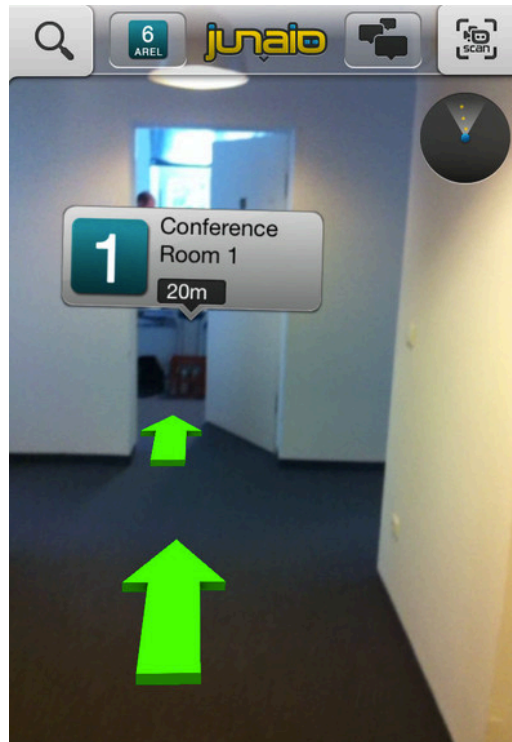


Figura 22 - Exemplo de uma aplicação que utiliza a técnica de *tracking* LLA markers. Imagem obtida de: <https://dev.metaio.com/typo3temp/pics/4d59ef54f2.jpg>

3.4.9 CAD Model Tracking

Esta técnica de *tracking* procura correspondências entre imagens 2D do mundo real e um modelo 3D – produzido numa fase *offline* – recorrendo a informação proveniente de texturas, arestas, ou ambas, dependendo do tipo de configuração. A versão atual do Metaio SDK (6.0.2) suporta três tipos de configurações: Feature Focused CAD Model Tracking, Pure Edge CAD Model Tracking e Hybrid CAD Model Tracking. Todas elas têm, em comum, o método de inicialização (baseado em arestas), devido à sua elevada precisão em obter uma estimativa inicial da pose da câmara. Após a inicialização, a técnica Feature Focused CAD Model Tracking utiliza pontos 3D baseados em textura (algoritmo SLAM) para continuar a efetuar o *tracking*, a técnica Pure Edge CAD Model Tracking continua a utilizar apenas as arestas, e a técnica Hybrid CAD Model Tracking utiliza, em simultâneo, informação proveniente das arestas e das texturas, para continuar o processo de *tracking*.

Para auxiliar os utilizadores a escolher a configuração adequada para esta técnica de *tracking*, o Metaio disponibiliza uma árvore de decisão onde os requisitos e as propriedades de cada configuração são apresentados (Figura 23).

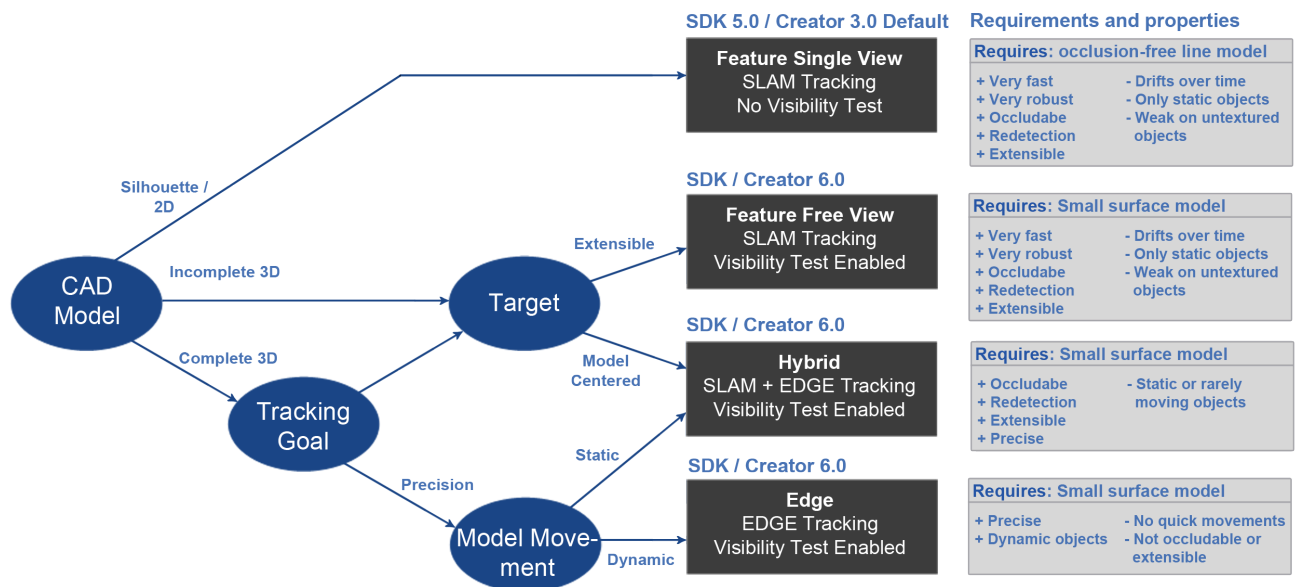


Figura 23 - Árvore de decisão para auxiliar os utilizadores a escolher a configuração mais adequada para a técnica CAD Model Tracking do Metaio SDK. Imagem obtida de:
<http://dev.metaio.com/typo3temp/pics/52168efcc0.png>

3.4.10 GPS/sensores inerciais

O Metaio SDK permite a criação de configurações de *tracking* baseadas em sensores. Estas suportam a utilização de GPS, giroscópio, acelerómetro ou compasso (magnético) para calcular a pose da câmara. Oferece, também, a possibilidade de se efetuar um *tracking* com apenas 3 DOF, direcionado para panoramas 360°, utilizando, nesse caso, apenas sensores inerciais.

A próxima secção descreve o processo de execução genérico das técnicas de *tracking*, disponibilizadas pelo Metaio SDK, na plataforma Unity3D.

3.5 Tracking no Metaio SDK + Unity3D

Conforme já foi referido na secção 3.3, o Metaio SDK disponibiliza um *plugin* para o Unity3D. Dentro do conteúdo disponibilizado estão alguns *prefabs*¹ relevantes: o *metaioSDK* e o *metaioTracker*. Estes *prefabs* atuam como *templates* a partir dos quais se podem criar novas instâncias de *GameObjects* na cena. A Figura 24 apresenta parte da interface gráfica do Unity3D. No canto esquerdo, a vista "Hierarchy" apresenta todos os *GameObject* presentes na

¹ *prefabs* são *GameObjects* (i.e. objetos utilizados nas cenas do Unity3D) armazenados com um determinado conjunto de componentes e propriedades.

cena, juntamente com as suas posições na hierarquia. Constata-se que alguns *GameObject* estão numa hierarquia inferior (e.g. *GameObject* DeviceCamera). Isto acontece porque o Unity3D usa o conceito de *parenting* (parentalidade), o qual permite que um *GameObject* possa estabelecer uma relação de pai/filho com um outro *GameObject*. Como resultado, o filho irá herdar o movimento e rotação do seu pai. No canto direito, a vista "Inspector" apresenta os componentes, e as suas propriedades, que constituem o *GameObject* selecionado no momento.

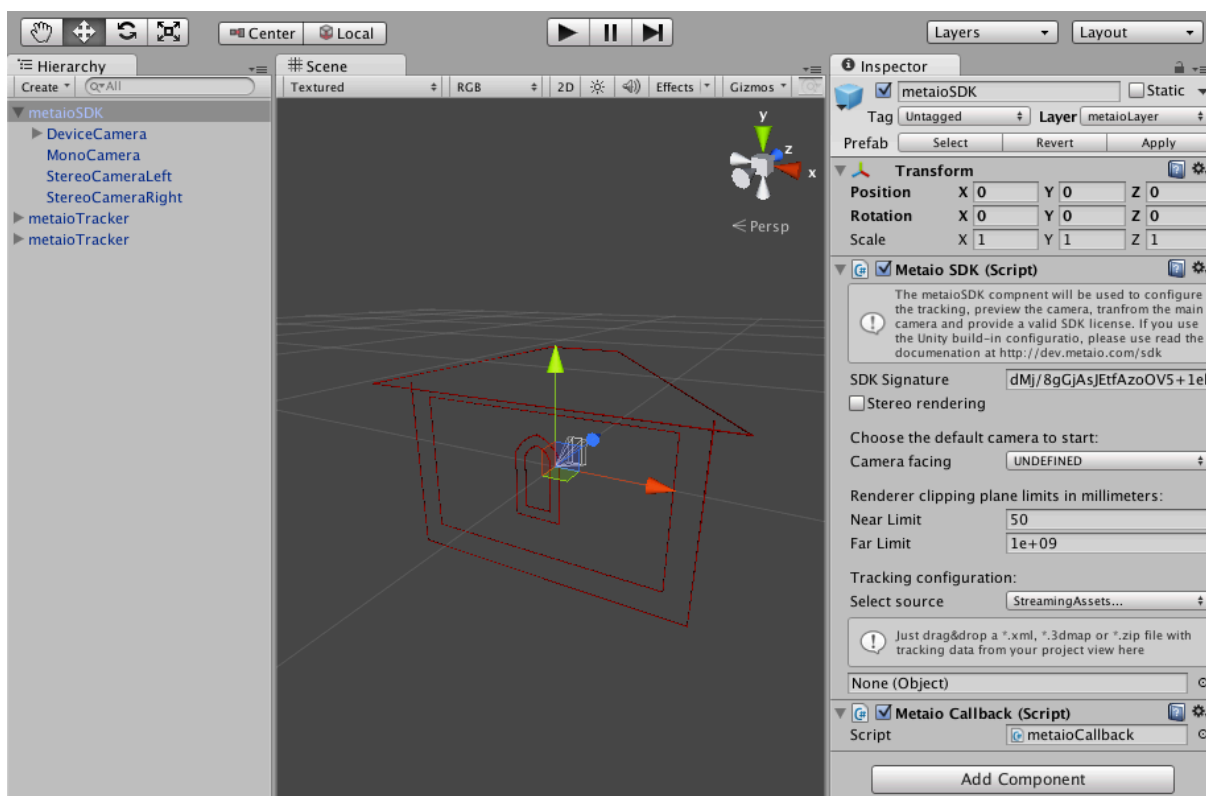


Figura 24 - Instância de um *GameObject* criado através do *prefab* metaioSDK. À direita, na vista "Inspector", são visíveis os componentes, juntamente com as suas propriedades, que constituem o *GameObject* metaioSDK, nomeadamente os *scripts* "Metaio SDK" e "Metaio Callback".

A Figura 24 apresenta, ainda, entre outros *GameObject*, um *GameObject* criado através do *prefab* metaioSDK. Uma vez que este se encontra selecionado, na vista "Inspector", são visíveis os componentes que constituem este *GameObject*, nomeadamente os *scripts* "Metaio SDK" e "Metaio Callback". O *script* "Metaio SDK" permite definir qual a câmara a utilizar, para o caso de ser detetada mais do que uma câmara, qual a configuração de *tracking* e se o *stereo rendering* deve ser ativo ou não. O *stereo rendering* consiste em apresentar o conteúdo virtual de duas perspetivas diferentes, num modo *split-screen*, com o objetivo de este aparecer em 3D. Devido a diferenças inerentes às múltiplas técnicas de

tracking, são disponibilizadas várias formas de definir uma configuração. No caso das técnicas de *tracking* que não necessitam de informação prévia, estas podem ser selecionadas diretamente pelo nome (e.g. QR-CODE). Caso possuam algum parâmetro de inicialização, este pode ser introduzido manualmente, através de uma *string* (e.g. "INSTANT_2D"). Nas técnicas de *tracking* onde é necessário fornecer algum tipo informação, através de ficheiros .XML, .3DMAP ou .ZIP, esta operação pode ser feita de duas maneiras: indicando o caminho completo para a localização do ficheiro de *tracking* (e.g. C:\tracking.zip), ou fazendo *drag-and-drop* do próprio ficheiro de *tracking* para o *script*, sendo o único requisito que a origem do ficheiro seja a pasta StreamingAssets, presente na solução Unity3D. O script "Metaio Callback" é utilizado para reportar eventuais mensagens de *debug*, aviso ou erro do *plugin* do Metaio SDK. Além disso, fornece informação acerca do estado do SDK e dos eventos de *tracking* que ocorrem. Relativamente à hierarquia do *GameObject* metaioSDK, este estabelece uma relação de pai com os *GameObject* (Figura 24): DeviceCamera, cuja tarefa é apresentar a imagem do mundo real, tal como obtida pelo câmara do dispositivo; MonoCamera, responsável por apresentar o conteúdo virtual, após haver uma estimativa de pose da câmara do dispositivo e essa ser aplicada a esta câmara virtual; StereoCameraLeft e StereoCameraRight, que realizam a mesma tarefa que MonoCamera, embora apenas utilizem duas câmaras virtuais para produzir duas perspetivas diferentes. São utilizadas quando o modo *stereo rendering* é ativo.

O *prefab* metaioTracker possui, apenas, um componente do tipo *script*, com o nome "Metaio Tracker" (Figura 25). Este é utilizado para fazer a ligação entre *GameObjects* do Unity3D e configurações de *tracking* do Metaio. Uma instância deste *prefab* atua como uma referência no projeto Unity3D para um determinado sistema de coordenadas – identificado por ID (número de identificação) – no ficheiro de configuração de *tracking*. Uma vez que, em cada instância do *prefab* metaioTracker, se pode apenas associar um sistema de coordenadas específico, o número de instâncias metaioTracker deve ser igual ao número de sistemas de coordenadas presentes na configuração de *tracking*. Além disso, o ID do sistema de coordenadas deve corresponder a um sistema de coordenadas válido na configuração de *tracking* definida. Por norma, o sistema de coordenadas com ID 1 é utilizado para o modelo virtual que irá aumentar o objeto/cena do mundo real, após um *tracking* bem sucedido. A utilização de outros ID varia conforme a técnica de *tracking* em si e o ficheiro de configuração .XML. A título de exemplo, na técnica *image-based* podem-se usar diversas

instâncias de metaioTracker com diferentes ID, uma vez que esta suporta *tracking* para várias imagens (cada imagem é associada a um determinado sistema de coordenadas), dentro da mesma configuração. No caso das técnicas baseadas em modelos são utilizados, obrigatoriamente, dois sistemas de coordenadas, onde o ID 1 é utilizado para o modelo virtual, que irá aumentar o objeto/cena (Figura 25) e o ID 2 é utilizado para o modelo de ajuda visual, que irá auxiliar o utilizador na tarefa de encontrar a pose inicial do objeto/cena, conforme definido no ficheiro de configuração (Figura 26).

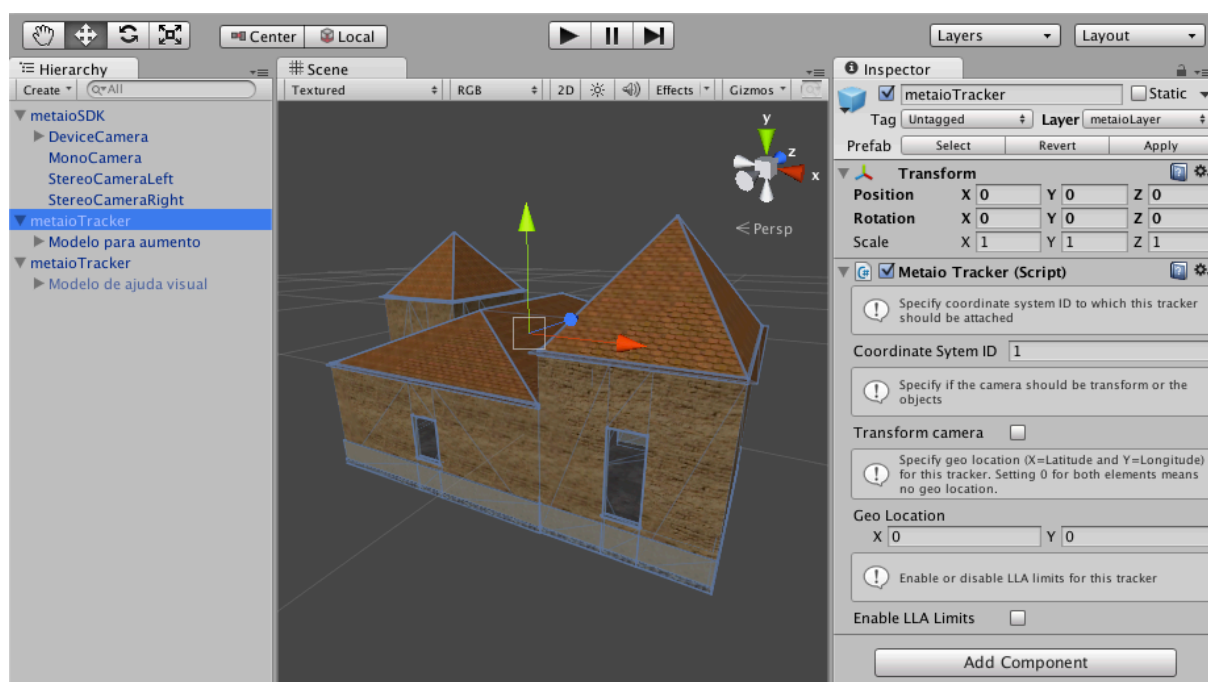


Figura 25 - A figura apresenta os componentes e propriedades do *GameObject* selecionado. À direita, através da vista "Inspector", pode-se constatar que o sistema de coordenadas (Coordinate System ID) selecionado é o "1". No centro da figura é apresentado o modelo virtual (*GameObject* "Modelo para aumento"), que irá aumentar a cena após um *tracking* bem sucedido.

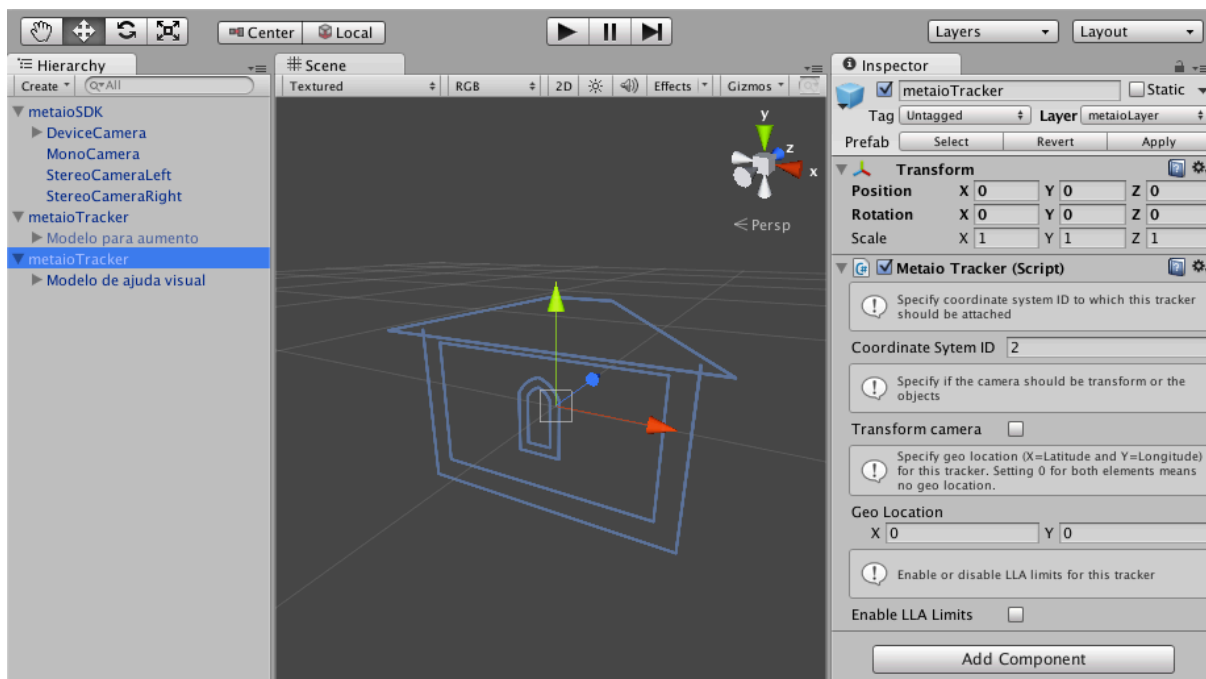


Figura 26 - A figura apresenta os componentes e propriedades do *GameObject* selecionado. À direita, através da vista "Inspector", pode-se constatar que o sistema de coordenadas (Coordinate System ID) selecionado é o "2". No centro da figura é apresentado o modelo virtual (*GameObject* "Modelo de ajuda visual ") que irá ser utilizado para auxiliar o utilizador a encontrar a pose inicial definida no ficheiro de configuração.

O *script* "Metaio Tracker" permite, ainda, especificar se as transformações – translações e rotações – devem ser aplicadas à câmara (virtual) ou ao objeto em si, qual a localização georreferenciada do objeto no mundo real (no caso da técnica de *tracking* utilizar geolocalização) e, por fim, se os limites LLA devem ser utilizados, fazendo com que o objeto seja renderizado apenas se estiver dentro dos limites definidos.

O próximo capítulo é focado na apresentação da metodologia de *tracking* implementada juntamente com as diversas alterações que ocorreram ao longo do seu desenvolvimento, tendo ultimamente contribuído para a criação de duas versões da MixAR App.

4 Metodologia de Tracking Proposta

O presente capítulo centra-se naquilo que é a motivação desta dissertação: a implementação de uma metodologia de *tracking* no projeto de investigação e desenvolvimento MixAR. São apresentadas duas metodologias de *tracking*, uma inicial e uma final que foram propostas durante a realização desta dissertação. As suas implementações são, também, apresentadas. A apresentação das metodologias e as suas implementações são devidamente separadas por subsecções ao longo do capítulo. Dado que a metodologia de *tracking* é implementada na aplicação que é executada na unidade móvel (MixAR App), o facto de existirem duas propostas para a metodologia de *tracking* resultou na criação de duas versões da MixAR App (Versões 1.0 e 2.0). A secção 4.1 refere-se à metodologia de *tracking* inicial proposta, a mesma é explicada através da apresentação do diagrama de fluxo da MixAR App e descrição das suas tarefas. Além disso, é efetuada uma comparação das técnicas de *tracking* fornecidas pelo Metaio SDK de forma a justificar a técnica escolhida (técnica "Hybrid CAD Model Tracking"). A secção 4.2 detalha a implementação da metodologia de *tracking* inicial proposta, sendo o resultado a obtenção do primeiro protótipo funcional da aplicação (MixAR App versão 1.0). A secção seguinte apresenta a metodologia de *tracking* final. Após a implementação da metodologia de *tracking* inicial proposta, alguns problemas e possíveis melhorias foram detetadas, esta segunda metodologia de *tracking* (proposta final), teve como objetivo corrigir os erros e incorporar as melhorias detetadas para ultimamente contribuir para uma melhoria das experiências de RM. Por fim, a secção 4.4 detalha a implementação da metodologia de *tracking* final proposta.

4.1 Metodologia de Tracking: Proposta Inicial

A metodologia de *tracking* é implementada através da MixAR App, como tal, esta secção baseia-se na apresentação do diagrama de fluxo da MixAR App, e descrição das suas tarefas, para explicar a metodologia de *tracking* proposta. O diagrama de fluxo da MixAR App é apresentado na Figura 27, neste pode-se constatar que a aplicação começa por verificar se a unidade móvel possui ou não o conteúdo necessário para a experiência de RM e, em caso afirmativo, se este se encontra na sua última versão. No caso de não o possuir ou de não estar na última versão, o conteúdo é transferido do repositório global, a partir do servidor de RM, para o repositório local da unidade móvel. Posteriormente, através dos dados obtidos pelo

sensor de GPS, é determinada a posição atual do utilizador. Nesta altura, o valor da posição é atribuído a uma variável "Posicao_atual" e copiado para a variável "Ultima_posicao". De seguida, o valor da posição é enviado do cliente para o servidor de RM. Com base nela, o servidor calcula a estrutura mais próxima do cliente e cria uma lista com as restantes estruturas da experiência, devidamente separadas pelos níveis de detalhe, conforme definido no SIG. Após este passo, a lista das estruturas é devolvida à unidade móvel, para que possa proceder ao carregamento da configuração de *tracking*, de uma forma dinâmica, da estrutura mais perto de si e ao carregamento dos respetivos modelos virtuais associados às restantes estruturas existentes na experiência, com o nível de detalhe adequado. A aplicação entra, depois, num ciclo, onde o processo de *tracking* é executado e o valor da variável "Posicao_atual" atualizado, novamente com base no dispositivo de GPS, até que a distância entre "Ultima_posicao" e "Posicao_atual" seja maior que metade do raio definido para o nível de detalhe mínimo (1), conforme definido na especificação do projeto MixAR. Quando esta última condição for verificada, o valor de "Posicao_atual" é atribuído a "Ultima_posicao" e os modelos e respetivas configurações de *tracking* são recalculados e carregados. Este ciclo repete-se até que o utilizador encerre a Mixar App.

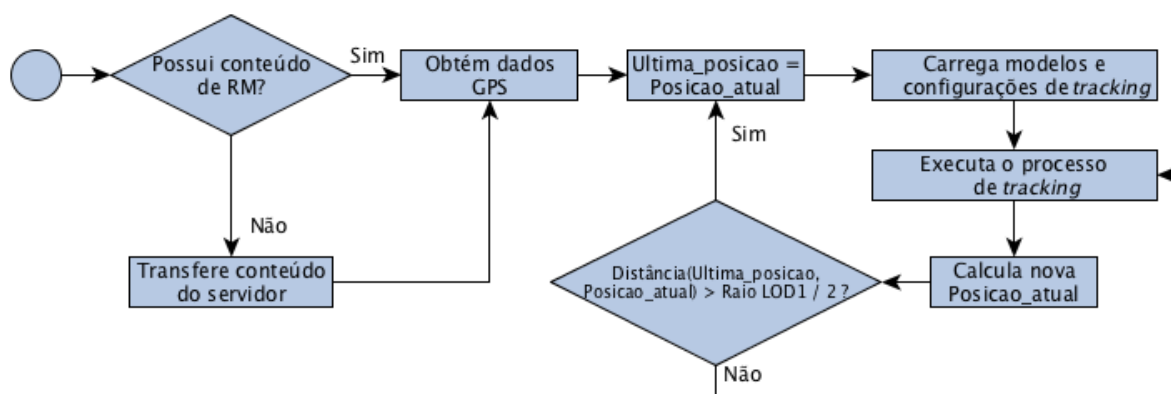


Figura 27 - Diagrama de fluxo da MixAR App. Descrição do processo de atualização constante que a aplicação executa de forma a fornecer informação contextualizada à unidade móvel.

A seleção da técnica de *tracking* para esta versão 1.0 da MixAR App foi antecedida de uma avaliação das diversas técnicas, disponibilizadas pelo Metaio SDK, com o objetivo de selecionar a mais indicada para desempenhar a tarefa pretendida. O Metaio SDK disponibiliza um total de 10 técnicas de *tracking* (descritas na secção 3.4). No entanto, nem todas são relevantes para serem utilizadas na MixAR App, nomeadamente: as técnicas "ID Marker" e "LLA Markers", por utilizarem marcas fiduciais; "Picture Marker", pelas restrições que

apresenta, causadas pelo facto de ser fundamentalmente uma extensão da técnica "ID Marker"; "Face tracking", por ser direccionada apenas para a deteção de faces humanas; "GPS/sensores inerciais", por ser análoga a uma abordagem *open-loop*, não permite que o sistema obtenha um feedback do registo entre os conteúdos virtual e real; finalmente, a técnica "Instant Tracking". Esta é uma técnica baseada em SFM e, como tal, tem a vantagem de conseguir aumentar cenas/objetos desconhecidos. No entanto, devido à falta de informação sobre o ambiente não é possível controlar o sítio que será aumentado, algo que é fundamental na MixAR App. Apesar do Metaio SDK possuir uma alteração da técnica "Instant Tracking" que permite o reconhecimento de um ponto de partida, para que tal aconteça a técnica exige a colocação de uma marca fiducial no terreno, algo que vai contra os requisitos da técnica de *tracking* pretendida. Desta maneira, sobram quatro técnicas de *tracking* com potencial para serem utilizadas: "Image Tracking", "3D Map Tracking", "Extended Image Tracking" e "CAD Model Tracking". De entre as quatro, foi selecionada a técnica "CAD Model Tracking" pelos seguintes motivos: tem a capacidade de produzir uma estimativa da pose de uma estrutura de qualquer ângulo, pelo facto de conhecer toda a geometria da mesma, as técnicas "Image Tracking" e "Extended Image Tracking" possuem um ângulo de deteção limitado porque estimam a pose da estrutura com base numa imagem 2D; a criação dos modelos virtuais é feita manualmente o que permite uma maior precisão, ao contrário da técnica "3D Map Tracking", que gera modelos virtuais através de um processo semiautomático (descrito na secção 5.1); por último, quando comparada com as restantes técnicas, a técnica "CAD Model Tracking" possui um maior número de parâmetros configuráveis, o que permite um maior nível de adaptação para cada caso de uso. Entre as possíveis configurações da técnica "CAD Model Tracking" (apresentadas na secção 3.4.9), optou-se pela configuração Hybrid, denominada de agora em diante como "Hybrid CAD Model Tracking". Esta é uma técnica híbrida que combina dois tipos de técnicas de *tracking*: baseada em modelos (técnica "Edge CAD Model Tracking") e baseada em SFM (técnica "Feature Focused Model Tracking"). A técnica "Edge CAD Model Tracking" deteta arestas, presentes nas imagens do mundo real, e tenta correspondê-las com as arestas presentes num determinado modelo 3D. A técnica "Feature Focused CAD Model Tracking" é uma implementação de um algoritmo SLAM, uma que se baseia em pontos característicos, de textura, para criar um mapa 3D do ambiente envolvente enquanto, em simultâneo, localiza a câmara. Através da combinação de ambas, a técnica "Hybrid CAD Model Tracking" tem como objetivo combinar as forças destas técnicas e suprimir as suas fraquezas na tentativa de obter um *tracking* mais robusto e preciso. Devido

a uma das técnicas ("Edge CAD Model Tracking"), ser baseada em modelos, a execução da técnica "Hybrid CAD Model Tracking" exige que seja criado um conjunto de modelos virtuais 3D com informação sobre a estrutura onde irá ser feito o *tracking*. Este conjunto de modelos virtuais é, mais especificamente, constituído por um modelo de arestas, superfície e ajuda visual (Figura 28).

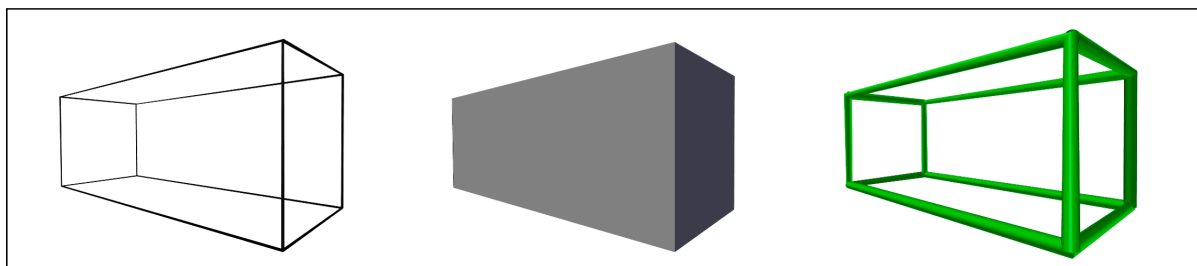


Figura 28 - Ilustração de um modelo de arestas, superfície e ajuda visual.

O modelo de arestas é utilizado para encontrar a pose de câmara ao tentar alinhar as arestas do modelo virtual com as arestas detetadas na imagem do mundo real. O modelo de superfície serve para calcular a localização 3D de pontos característicos, utilizados pela técnica "Feature Focused CAD Model Tracking", e para determinar as linhas visíveis do modelo de arestas, devido à possível auto-occlusão das mesmas pelo próprio modelo. Por fim, o modelo de ajuda visual atua como um guia para o utilizador corresponder a pose da estrutura do mundo real com uma determinada pose (denominada pose inicial) definida na configuração do *tracking*. A técnica "Hybrid CAD Model Tracking" possui três estados: "*Initialize*", "*Tracking*" e "*Lost*". O primeiro estado a ser assumido é o estado "*Initialize*", ou seja, de inicialização. Neste, apenas a técnica baseada em modelos ("Edge CAD Model Tracking") é executada, devido à sua capacidade em obter uma estimativa precisa da pose inicial da estrutura (e consequentemente da câmara). No entanto, como ponto fraco, só consegue obter uma estimativa da pose inicial quando as arestas detetadas no mundo real correspondem com as do modelo de arestas da estrutura, algo que requer a intervenção do utilizador. Para que o utilizador saiba qual a pose a procurar, é-lhe apresentado o modelo de ajuda visual, após a aplicação de uma pose inicial definida no ficheiro de configuração da mesma. Uma correspondência das arestas produz uma estimativa da pose da câmara e despoleta a apresentação do modelo virtual, definido na configuração, sobre a vista do mundo real. Esta ação faz, também, com que o estado seja alterado de "*Initialize*" para "*Tracking*". Esta transição é ilustrada pela Figura 29.



Figura 29 - À esquerda a imagem apresenta o modelo de ajuda visual alinhado, pelo utilizador, com as arestas da estrutura do mundo real. À direita, é apresentado o modelo virtual, que aumenta a estrutura após a uma correspondência das arestas com sucesso.

O estado de "*Tracking*" é mantido enquanto forem produzidas estimativas da pose. Para que tal aconteça, os algoritmos de deteção de arestas e SLAM são executados em simultâneo através de *multi-threading*, num processo que deteta arestas e extrai pontos característicos da textura de forma contínua, criando e estendendo um mapa de pontos 3D. O estado é alterado de "*Tracking*" para "*Lost*" quando os algoritmos deixam de conseguir estimar uma pose de câmara. Uma situação que pode ocorrer por diversos motivos, tais como o desaparecimento total do alvo de *tracking* do campo de visão da câmara ou a realização de movimentos bruscos com a câmara, fazendo, por exemplo, com que a pose calculada anteriormente seja completamente distinta da atual. Quando o estado é alterado para "*Lost*", é inicializado um processo de recuperação com base no algoritmo SLAM que, com base no mapa de pontos 3D criado até então, tenta por um determinado período de tempo reinicializar o *tracking* a partir da última pose calculada. No caso de insucesso, o mapa de pontos 3D é reinicializado e o estado alterado para "*Initialize*", repetindo o processo aqui descrito. O diagrama de fluxo da técnica de *tracking* aqui descrita é apresentado na Figura 30.

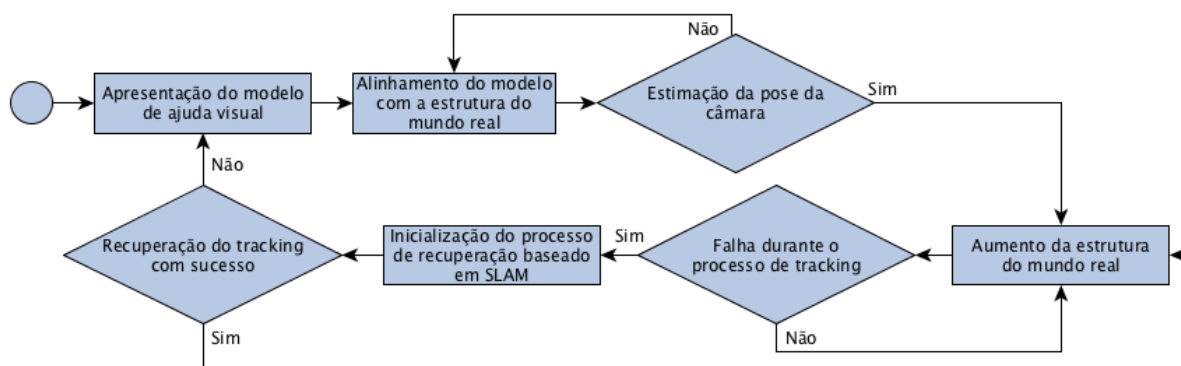


Figura 30 - Diagrama de fluxo da técnica "Hybrid CAD Model Tracking", utilizada na versão 1.0 da MixAR App.

4.2 MixAR App - Versão 1.0

Conforme especificado na secção 3.2, foi implementada uma arquitetura cliente-servidor que permite que uma ou mais unidades móveis comuniquem com um servidor de RM. Como servidor de RM foi utilizada uma solução de alojamento *web* comum. Como unidade móvel, um *tablet* Samsung SM-T320 e como infraestrutura de rede, recorreu-se à utilização da rede 802.11 com o Service Set Identifier (SSID) "eduroam" presente no *campus* da Universidade de Trás-os-Montes e Alto Douro (UTAD). Relativamente ao modelo de dados, este foi implementado do lado do servidor em MySQL. A Base de Dados (BD) resultante mapeia o modelo de dados explicado em detalhe na secção 3.2.1.

Quanto à implementação da metodologia de *tracking* na MixAR App, esta foi efetuada com recurso ao GameEngine previamente referido, Unity3D, juntamente com o Metaio SDK. Como Integrated Development Environment (IDE) optou-se pela utilização do MonoDevelop, configurado, por omissão, no Unity3D, e pela linguagem de programação C# com a framework .NET versão 2.0. Quanto ao fluxo da aplicação em si, este foi implementado exatamente como ilustrado anteriormente na Figura 27. De uma forma contínua, a aplicação deteta a localização do utilizador e procede ao carregamento dinâmico da configuração de *tracking* da estrutura mais próxima, na eventualidade de o utilizador ficar mais próximo de uma outra estrutura presente na experiência, a devida configuração é carregada e o processo de *tracking* reiniciado. Quanto ao fluxo da técnica de *tracking* escolhida ("Hybrid CAD Model Tracking"), este é executado tal como ilustrado na Figura 30. Após a escolha da técnica "Hybrid CAD Model Tracking", procedeu-se à configuração da mesma para algumas estruturas do mundo real a fim de realizar alguns testes de desempenho. Para cada estrutura utilizada foi necessária a criação de uma configuração de *tracking* distinta. Isto acontece porque, cada estrutura possui propriedades de geometria/estrutura e de textura diferentes. Como consequência, os modelos de arestas, superfície e ajuda visual são diferentes de estrutura em estrutura. Além disso, também os parâmetros de *tracking*, são alterados, de configuração para configuração, a fim de obter o melhor desempenho possível em cada uma das estruturas utilizadas para *tracking*. Estas configurações, da técnica "Hybrid CAD Model Tracking", são compostas por três ficheiros:

- *tracking.zip*: Resultado da compressão dos diversos ficheiros necessários para o *tracking*, sendo estes: configuração dos parâmetros da técnica de *tracking* (.XML),

pose inicial definida para a estrutura (.XML), modelo de arestas (.OBJ) e modelo de superfície (.OBJ);

- visualAid.obj: Modelo de ajuda visual;
- augmentationContent.obj: Modelo virtual utilizado para aumentar a estrutura. Além deste ficheiro são também colocadas as imagens que constituem a textura do modelo no formato .PNG.

Quanto à interface gráfica da MixAR App, o objetivo foi torná-la simples e intuitiva para o utilizador. A Figura 31 apresenta a interface gráfica desta primeira versão da aplicação. Esta é constituída pela apresentação da imagem do mundo real em *fullscreen*, podendo esta ser combinada com conteúdo virtual, nomeadamente um modelo de ajuda visual ou um modelo virtual que irá aumentar uma dada estrutura. Um mapa 2D no canto inferior esquerdo que tem como objetivo ajudar o utilizador a localizar os pontos de interesse, ou seja, as estruturas que poderão ser aumentadas após um *tracking* efetuado com sucesso. Por fim, quatro botões no canto inferior direito. Os botões "Translation" e "Rotation" servem para aplicar estas transformações, através do deslocamento do dedo do utilizador no ecrã, ao modelo de ajuda visual de forma a que o *tracking* possa ser inicializado de um ponto de vista diferente do definido no ficheiro de configuração. O botão "Reset" serve para reinicializar o *tracking*, sendo este útil em situações onde seja detetada uma pose inicial errada de uma estrutura. Por último, o botão "Quit" termina a execução da aplicação. Foi adicionado um botão lógico para terminar a aplicação pois após a realização de alguns testes verificou-se que os utilizadores carregavam diversas vezes, de forma involuntária, com o polegar, no botão físico de retroceder, interrompendo assim o funcionamento da aplicação. Relativamente à orientação da aplicação, nesta versão foi apenas utilizado o modo "*landscape left*".

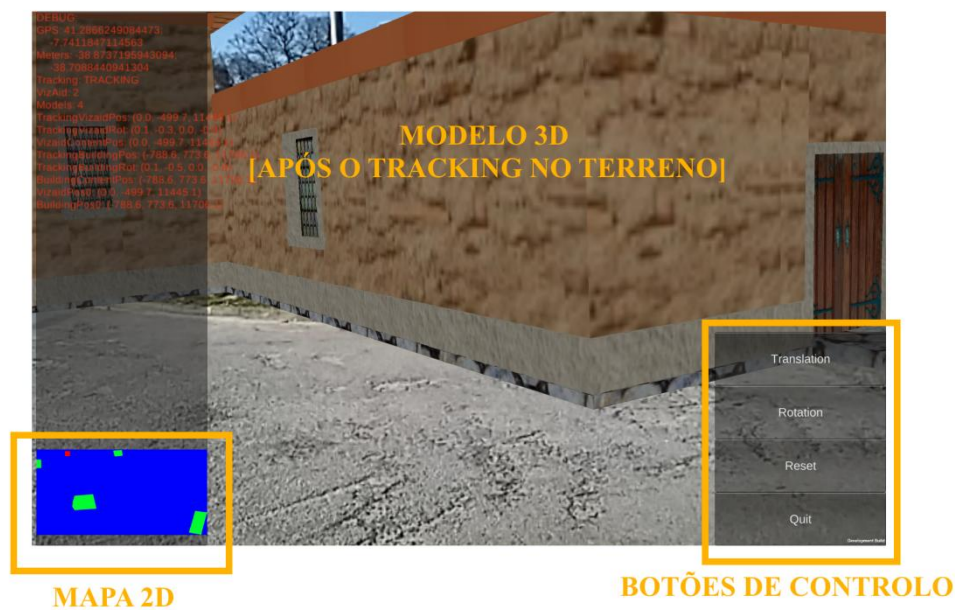


Figura 31 - Screenshot tirado pela unidade móvel na execução da primeira versão da MixAR App. Nesta pode-se constatar a visualização de um modelo virtual após o *tracking* de uma estrutura do mundo real.

Tal como referido anteriormente nesta secção, foram efetuadas configurações da técnica "Hybrid CAD Model Tracking" para algumas estruturas do mundo real com vista a realizar testes de desempenho. Isto exigiu a criação de três modelos virtuais (modelos de linhas, superfície e ajuda visual) para cada estrutura, a criação destes modelos foi efetuada através da ferramenta de modelação Blender, apresentada na secção 2.6. Já os modelos virtuais que aumentam as estruturas, foram gerados através de uma ferramenta de geração procedimental apresentada por Adão (Adão, Magalhães, Peres, & Pereira, 2014). Os testes de desempenho realizados com esta versão 1.0 da MixAR App tiveram como objetivo obter uma avaliação do desempenho da unidade móvel. Foram utilizadas três estruturas do mundo real, presentes no *campus* da UTAD, e como unidade móvel, foi igualmente utilizado o *tablet* Samsung SM-T320. As limitações principais dos testes de desempenho estiveram relacionadas com a necessidade de criar modelos virtuais e a distância a que as estruturas se encontravam. Dado que havia a necessidade de criar modelos virtuais, foram escolhidas estruturas com uma geometria simples a fim de facilitar a tarefa de modelação. Relativamente à distância, foram escolhidos edifícios dentro do *campus* da UTAD de forma a acelerar o processo de criar/alterar uma configuração de *tracking* e a testar no terreno. A realização dos testes, juntamente com os seus resultados e discussão, são apresentados na secção 5.2.

4.3 Metodologia de Tracking: Proposta Final

Após a realização dos testes de desempenho com a versão 1.0 da MixAR App (apresentados na secção 5.2.1), foram detetadas algumas melhorias à metodologia de *tracking* inicial proposta. Como tal, uma segunda metodologia de *tracking* é proposta com o objetivo de implementar estas melhorias e consequentemente contribuir para uma melhor experiência de RM aos utilizadores das unidades móveis. Estas melhorias são mais concretamente: a adição do suporte a mais técnicas de *tracking* disponíveis no Metaio SDK e a realização do *tracking* e posterior aumento de todas as estruturas nas proximidades da unidade móvel ao invés de apenas a mais próxima. Na metodologia de *tracking* inicial, para efetuar o *tracking* da estrutura mais próxima da unidade móvel optou-se pela utilização da técnica "Hybrid CAD Model Tracking", disponível no Metaio SDK. Apesar de ser uma técnica com uma grande abrangência em termos de deteção de estruturas, a necessidade de criar modelos 3D das mesmas pode-se tornar complexa quando se lida com ruínas num avançado estado de deterioração. Além disso, ruínas com muita textura podem provocar falsas deteções de arestas utilizando esta técnica. De uma forma geral, todas as técnicas de *tracking* têm pontos fortes e fracos, a melhor abordagem depende da situação em concreto. Aliás, nem todas as técnicas de *tracking* funcionam em qualquer ambiente e o cálculo da posição e orientação de um utilizador continua a ser um problema complexo sem nenhuma solução ideal (Krevelen & Poelman, 2010). Como melhoria foi planeada a implementação de um sistema de *tracking* multimodal. Um sistema que suporte a utilização de diversas técnicas de *tracking*, permitindo, assim, a escolha da técnica de *tracking* mais adequada para cada caso de uso. Assim como referido no início deste capítulo, a outra melhoria à metodologia de *tracking* consiste a realização da operação de *tracking* e posterior aumento das restantes estruturas da experiência de RM. Para implementar esta melhoria planeou-se o desenvolvimento de raiz de um técnica de *tracking* baseada em sensores, por dois motivos: o Metaio SDK é limitado no sentido em que apenas uma técnica de *tracking* pode ser executada de cada vez. Uma vez que já é utilizada uma técnica do Metaio SDK para a estrutura mais próxima da unidade móvel, não é possível, ao mesmo tempo, utilizar outra(s) técnica(s) do Metaio SDK para as restantes estruturas, daí a implementação de uma técnica de raiz; as técnicas de *tracking* baseadas na visão são geralmente mais exigentes em termos computacionais que as técnicas baseadas em sensores, dado que já é utilizada uma técnica baseada na visão, optou-se pela implementação de uma técnica baseada em sensores de modo a não provocar um grande decréscimo de

desempenho na unidade móvel. Com vista a tornar a técnica de *tracking* mais eficiente, todos os modelos virtuais que aumentam as estruturas nas proximidades da unidade móvel (à exceção do modelo virtual associado à estrutura mais próxima), são agrupados e controlados como se de um único modelo se tratassem. Ao agrupamento destes modelos virtuais denomino daqui em diante como ambiente extensivo. O ambiente extensivo é, mais concretamente, um *gameObject* do Unity3D que é constituído por todos os modelos virtuais pertencentes à experiência de RM à exceção do modelo mais próximo da unidade móvel. O ambiente extensivo torna o *tracking* mais eficiente pois as operações de rotação e translação (resultantes da estimativa da pose da unidade móvel pela técnica de *tracking*) são aplicadas no ambiente como um todo e não individualmente a cada modelo virtual. Estas operações são aplicadas ao modelo extensivo de forma a serem isoladas das transformações aplicadas pela técnica de *tracking* do Metaio SDK, cujas transformações são aplicadas diretamente na câmara virtual presente na cena do Unity3D. Além de contornar a limitação do Metaio SDK, o ambiente extensivo fornece um ambiente mais imersivo uma vez que os modelos virtuais que o constituem nunca desaparecem do ambiente. Resumindo, a MixAR App executa uma técnica de *tracking* do Metaio SDK para a estrutura mais próxima do utilizador enquanto, em simultâneo, a técnica baseada em sensores aumenta as restantes estruturas presentes na experiência com os edifícios virtuais conforme definido nas configurações das mesmas. Para o funcionamento da técnica de *tracking* baseada em sensores foi planeada a utilização de um total de três sensores com as seguintes funções: magnetómetro, para alinhar os modelos virtuais com o norte magnético; giroscópio, para obter a orientação da unidade móvel; e GPS, para obter a posição em coordenadas (Latitude e longitude) da unidade móvel.

4.4 MixAR App - Versão 2.0

O sistema de *tracking* multimodal foi implementado com sucesso através da adição do suporte a mais três técnicas de *tracking*, nomeadamente: "3D Map Tracking", "Extended Image Tracking" e "Image Tracking", permitindo assim escolher a técnica de *tracking* mais adequada para cada caso de uso. A técnica de *tracking* baseada em sensores foi, também, implementada com sucesso. Para obter a orientação correta da unidade móvel, a técnica funciona da seguinte forma: quando a MixAR App é inicializada, o magnetómetro recolhe o grau atual da unidade móvel em relação ao norte magnético e armazena-o numa variável. Este valor é então aplicado, de forma contínua, como *offset* ao valor de *yaw* do giroscópio, atuando como um grau de correção para alinhar este eixo do giroscópio para norte. Relativamente à

obtenção da posição da unidade móvel, esta é efetuada com base apenas no dispositivo de GPS através dos valores da latitude e longitude. Apesar da tecnologia de GPS ter algum erro associado este revelou-se insignificante durante as experiências de RM devido à distância física entre estruturas. Um fator que contribuiu para o sucesso da técnica foi o facto de que os erros de registo são mais evidentes nos eixos de rotação do que nos de translação. Uma vez que a rotação é feita através do giroscópio, que tem uma boa precisão, não foram notórios erros nos eixos de rotação, em relação aos eixos de translação, o dispositivo de GPS ainda que com algum erro, devido à distância das estruturas produziu resultados satisfatórios fazendo com que a técnica obtivesse um bom registo. Relativamente aos níveis de detalhe, estes não foram totalmente implementados. Apesar da ferramenta do SIG permitir a criação e edição destes, a versão 2.0 da MixAR App atualmente suporta apenas dois níveis de detalhe, um mais elevado, para o modelo virtual que irá aumentar a estrutura mais próxima do utilizador na experiência e um mais reduzido, para os modelos virtuais que aumentam as restantes estruturas presentes na experiência, ou seja, o ambiente extensivo.

Nesta versão 2.0 da MixAR App foram, ainda, efetuadas diversas alterações à interface gráfica da mesma. Tal como se pode constatar pela Figura 32, foi adicionada uma *label* no canto superior esquerdo com o nome da estrutura que é o atual alvo de *tracking* e foram removidos os botões "Rotation" e "Translation", utilizados para a manipulação do modelo de ajuda visual da técnica "Hybrid CAD Model". A aplicação foi alterada para que, na utilização desta técnica, a operação de rotação (no eixo do X, right, e Z, front) seja realizada de forma mais intuitiva ao utilizador através do acelerómetro do dispositivo. Ou seja, as operações passam a ser aplicadas ao modelo de ajuda visual através da rotação da própria unidade móvel ao invés do deslocamento do dedo do utilizador no ecrã. Quanto à operação de translação, esta foi removida da aplicação pois revelou-se ineficaz durante os testes preliminares, devido ao tamanho das estruturas o deslocamento era muito pequeno, o que tornava as alterações praticamente invisíveis.



Figura 32 - Screenshot tirado pelo tablet da interface gráfica na segunda versão do MixAR.

Foram ainda aplicadas outras alterações à interface gráfica, guiadas pelo tipo de imersão, pela plataforma utilizada e pelo número de ecrãs para visualizar o conteúdo. Para as configurações imersivas, os botões "Reset" e "Sair" (previamente "Quit") foram retirados visto que o utilizador não consegue clicar nestes utilizando um HMD. Nas configurações não imersivas, estes botões mantêm-se e além disso é possibilitado ao utilizador, através do toque, a manipulação dos modelos de ajuda visual no eixo do Y. Relativamente ao número de ecrãs, (apenas relevante nas configurações imersivas por haver algumas com dois ecrãs) no caso da configuração ter apenas um ecrã, todos os elementos que constituem a interface gráfica são duplicados e separados horizontalmente (Figura 33), devido aos dois pontos de visão do ser humano esta é compreendida como apenas uma imagem. No caso de ter dois ecrãs, o comportamento seguido é o mesmo que o apresentado na Figura 32 à exceção dos botões que são removidos por ser uma configuração imersiva.



Figura 33 - Screenshot tirado pelo *smartphone* durante a execução do MixAR para configurações imersivas com apenas um ecrã.

A adição de novas técnicas de *tracking* também contribuiu para a adição de novas funcionalidades na interface gráfica da aplicação. Sempre que o gestor de experiência queira, uma imagem é apresentada no canto superior direito bastando para tal adicionar a mesma no ficheiro .ZIP que alberga as configurações de *tracking*. O objetivo é fornecer ao utilizador uma ajuda visual da pose inicial correta da estrutura para que esta seja detetada e o *tracking* despoletado (Figura 34).



Figura 34 - Imagem no canto superior direito fornece ao utilizador uma ajuda visual da pose inicial pretendida sendo utilizado neste caso a técnica "3D Map Tracking".

Por fim, a orientação da aplicação é guiada pelo tipo de plataforma, Android e iOS utilizam a orientação "*landscape left*", enquanto que a plataforma Windows utiliza a orientação "*portrait*".

5 Resultados e Discussão

Este capítulo tem como foco principal a apresentação e discussão dos resultados obtidos nos testes de desempenho, realizados nas versões 1.0 e 2.0 da MixAR App, e nos testes com utilizadores, realizados na versão 2.0 da MixAR App. A secção 5.1 começa por descrever o processo que antecede qualquer teste da MixAR App, a geração de experiências de RM. As seguintes secções, 5.2 e 5.3, consistem na apresentação e discussão dos resultados obtidos nas versões 1.0 e 2.0 da MixAR App, respetivamente. Os testes de desempenho realizados com a versão 1.0 tiveram como objetivo obter uma primeira avaliação do desempenho da unidade móvel. Para tal, foi concebida uma experiência de RM que incorporava três estruturas do mundo real, existentes no campus da UTAD e foi utilizado, como unidade móvel, um *tablet* Samsung SM-T320. Na versão 2.0 da MixAR App, foram efetuados testes de desempenho e testes com utilizadores. Os testes de desempenho tiveram como objetivo conhecer o custo computacional associado a diversas combinações de unidades móveis e técnicas de *tracking*. Já os testes com utilizadores, visaram obter dados acerca da satisfação dos utilizadores e recolher opiniões sobre a sensação de imersividade, presença e conforto na utilização das unidades móveis. Relativamente às estruturas utilizadas na experiência de RM, na versão 2.0, foram utilizadas três estruturas do mundo real localizadas nos arredores do museu da Vila Velha em Vila Real.

5.1 Processo de Criação de Experiências de RM

Para a geração de experiências de RM foi alinhado um processo que combina a ferramenta desenvolvida para o SIG, a ferramenta de modelação e animação 3D Blender e duas ferramentas do Metaio direcionadas para a criação de configurações de *tracking*: o Metaio Creator e o Metaio Toolbox. A primeira etapa desse processo consiste em utilizar a ferramenta de SIG na seleção das bases das estruturas que irão ser utilizadas na experiência de RM, assim como na definição do raio de cada nível de detalhe. Após esta etapa, as porções de terreno que constituem as bases das estruturas são convertidas para um ficheiro *.OBJ*, respeitando a escala recomendada na documentação do Metaio (1 unidade para cada mm), sendo depois exportado do servidor, para que possa ser trabalhado em ferramentas *offline*. Este ficheiro é, então, importado para o Blender com o intuito de escalar e rodar os modelos virtuais, previamente gerados pela ferramenta de geração procedimental apresentada por

(Adão et al., 2014), de forma a que estes sejam corretamente posicionados sobre as estruturas aquando da sua utilização no ambiente extensivo. Nesta fase, estão concluídas as alterações aos modelos virtuais que irão ser utilizados para aumentar as estruturas do mundo real. A próxima etapa está relacionada com as técnicas de *tracking* que irão ser utilizadas na experiência de RM. Numa etapa preliminar, é efetuada uma avaliação visual das estruturas que irão ser utilizadas na experiência de RM com vista a decidir qual a técnica de *tracking* mais adequada para cada caso. De seguida, procede-se à criação das configurações de *tracking* para as técnicas escolhidas.

A MixAR App, na sua última versão, 2.0, suporta as técnicas: "Hybrid CAD Model Tracking", "3D Map Tracking", "Image Tracking" e "Extended Image Tracking". Na eventualidade de a técnica de *tracking* mais indicada ser a "Hybrid CAD Model Tracking", o primeiro passo consiste na criação de um modelo virtual completo – contendo arestas e superfícies – da estrutura do mundo real que se pretende detetar. Este modelo é criado através do Blender. A título de exemplo, a Figura 35 ilustra uma estrutura do mundo real seguida do seu modelo virtual completo.

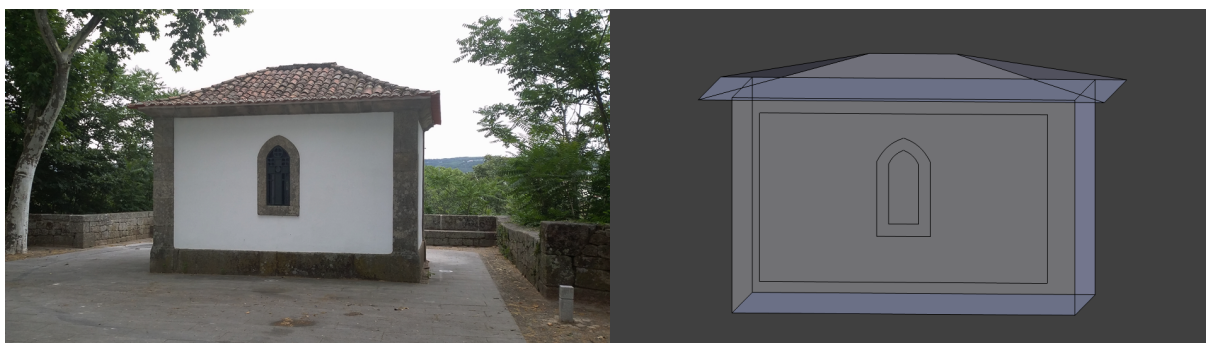


Figura 35 - Exemplo de uma estrutura do mundo real seguida do modelo virtual completo da mesma.

Tal como é descrito na secção 4.1, a configuração da técnica "Hybrid CAD Model Tracking" necessita de três modelos virtuais (modelo de arestas, superfície e ajuda visual) e de dois ficheiros .XML (configuração dos parâmetros da técnica e pose inicial). Apesar de existir um quarto modelo virtual na configuração desta técnica de *tracking* (o modelo que aumenta a estrutura), este não será aqui abordado uma vez que é gerado automaticamente por uma ferramenta procedimental, proposta por (Adão et al., 2014). Existem duas formas de criar uma configuração desta técnica: utilizando apenas o Blender ou utilizando o Blender juntamente com o Metaio Creator. No caso de se utilizar apenas o Blender, os modelos virtuais necessários são criados a partir do modelo virtual completo, relativamente aos

ficheiros .XML, são utilizados ficheiros *template* (disponibilizados no Metaio SDK) que contêm uma configuração base da técnica, sendo este alterado e testado até obter a pose inicial pretendida e um nível satisfatório de precisão e robustez no *tracking*. Utilizando o Blender em junção com o Metaio Creator, apenas é exigida a criação do modelo de ajuda visual, os restantes modelos (de arestas e de superfície) e ambos os ficheiros .XML são gerados automaticamente após a importação do modelo virtual completo da estrutura para o Metaio Creator. No entanto, nem sempre deve ser utilizado o Metaio Creator para gerar estes modelos, após a realização dos testes de desempenho com a versão 1.0 da MixAR App concluiu-se que o modelo de superfície é o que causa a maior carga computacional à aplicação. Este motivo leva a que os modelos virtuais necessários sejam frequentemente criados no Blender e não no Creator, onde apesar deste último suportar a redução da complexidade dos modelos, esta é feita de forma automática onde o utilizador não tem controlo sobre as arestas/superfícies que pretende remover.

Se a técnica de *tracking* adequada for a "3D Map Tracking", é necessária a utilização de um dispositivo móvel, Android ou iOS, com a aplicação Metaio Toolbox instalada. Através desta cria-se o mapa de pontos 3D, bastando para tal apontar para a estrutura e ir deslocando o dispositivo numa direção horizontal de forma contínua para que este possa ir estimando a posição 3D dos pontos (Figura 36). Após a recolha do mapa 3D procede-se à configuração do ficheiro de *tracking*. Igual à técnica de *tracking* anterior, esta utiliza um ficheiro .XML fornecido pelo Metaio SDK que serve como um *template* para começar a configurar a técnica para o caso de uso.

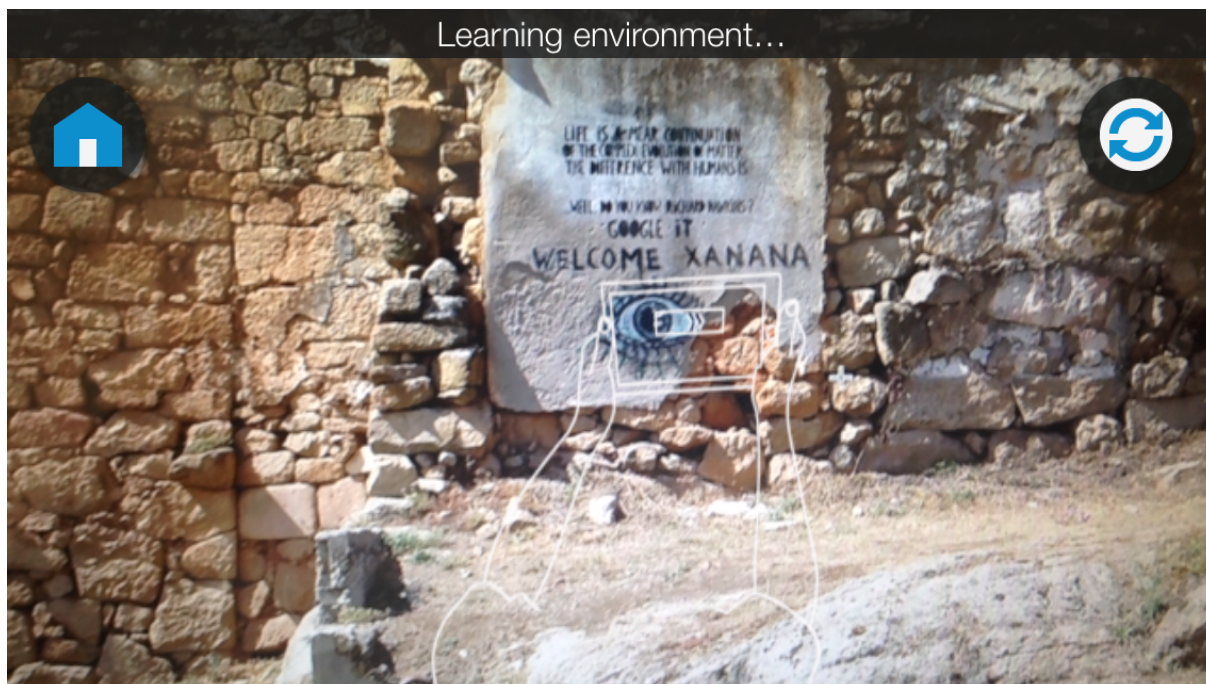


Figura 36 - Instruções fornecidas pela aplicação Metaio Toolbox para a criação de um mapa 3D. De uma forma contínua o utilizador move o dispositivo em redor da cena que pretende adquirir um mapa 3D, quando o utilizador entender que o mapa tem pontos 3D suficientes, para o processo de aprendizagem e armazena o mapa 3D.

Por último, no caso de se optar pela técnica "Image Tracking" ou "Extended Image Tracking", uma imagem da estrutura que se pretende detetar é suficiente para criar uma configuração de *tracking*. Esta é criada de forma automática pelo Metaio Creator após seleccionar uma das técnicas de *tracking* e importar a imagem da estrutura do mundo real. De maneira a alternar entre as técnicas "Image Tracking" ou "Extended Image Tracking", basta adicionar ou remover o suporte para o algoritmo SLAM através da interface gráfica do Metaio Creator (Figura 37). Esta interface permite ainda a configuração de outros parâmetros das técnicas de *tracking* referidas.

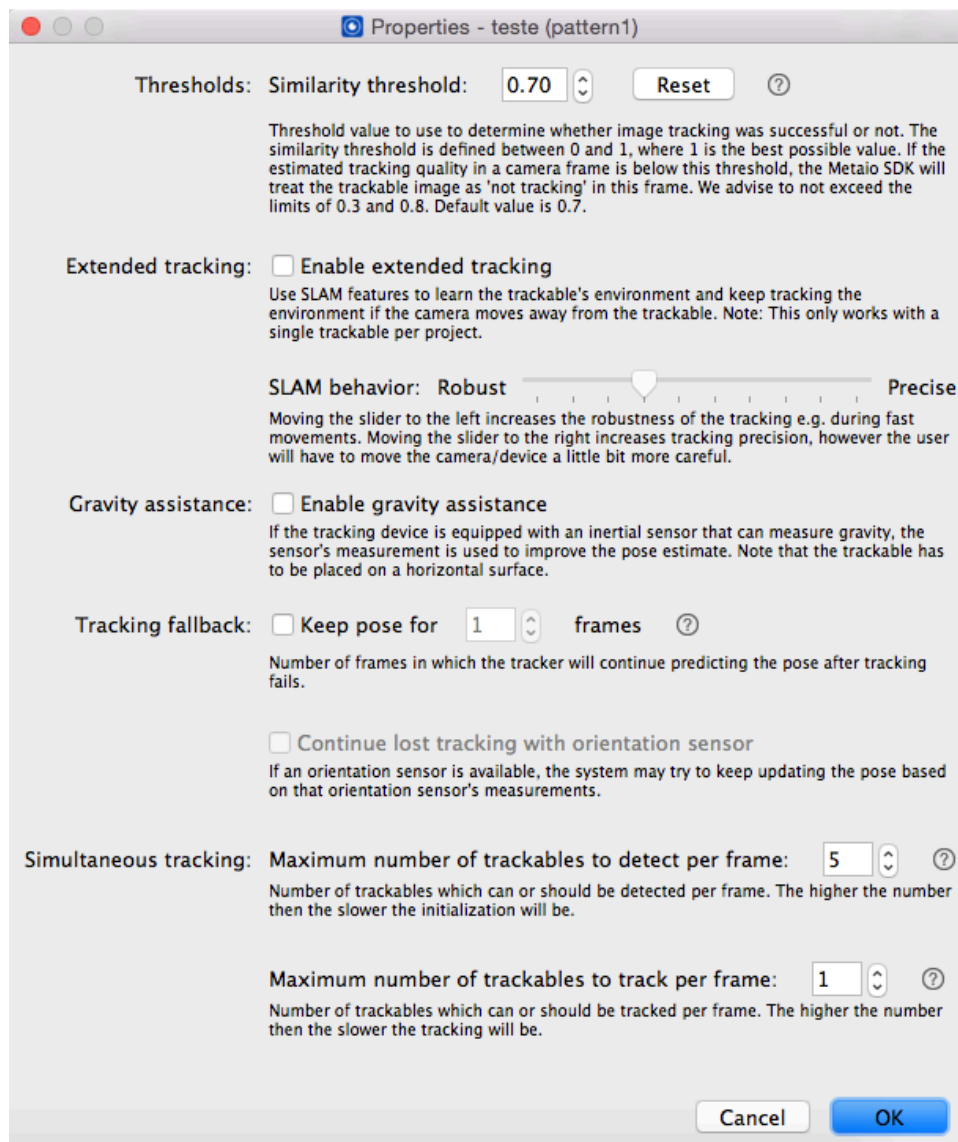


Figura 37 - Interface gráfica para a configuração das técnicas "Image Tracking" e "Extended Image Tracking". Fornece a possibilidade de ativar ou desativar o *extended tracking*.

Durante esta fase ocorrem diversos testes no campo a fim de obter configurações de *tracking* com bons níveis de robustez e precisão. Para concluir a configuração das técnicas de *tracking*, e caso seja necessário, pode-se definir um *offset* de rotação e translação adicional no modelo virtual que aumenta uma determinada estrutura para que seja corretamente posicionado sobre a mesma após um *tracking* efetuado com sucesso. Concluídas as configurações de *tracking*, a última tarefa consiste em colocar todo o conteúdo, modelos virtuais e configurações de *tracking*, no servidor para que este possa posteriormente ser transferido pelas unidades móveis na realização de experiências de RM.

5.2 MixAR App (Versão 1.0)

Tal como referido previamente, a primeira versão da MixAR App foi caracterizada pela obtenção de um primeiro protótipo funcional. Esta secção apresenta os testes de desempenho realizados a esse mesmo protótipo juntamente com os resultados e discussão dos mesmos.

5.2.1 Testes de Desempenho

Os testes de desempenho realizados nesta versão 1.0 da MixAR App tiveram como objetivo obter uma primeira avaliação do desempenho de uma unidade móvel, mais especificamente, um *tablet* Samsung SM-T320. As principais limitações dos testes de desempenho estiveram relacionadas com a necessidade de criar modelos virtuais e à distância física a que as estruturas se encontravam. Dado que havia a necessidade de criar modelos virtuais, foram escolhidas estruturas com uma geometria simples a fim de facilitar a tarefa de modelação. Relativamente à distância, foram escolhidas três estruturas dentro do campus da UTAD de forma a acelerar o processo de criar/alterar uma configuração de *tracking* e testá-la no terreno. Para a tarefa de *tracking*, nesta versão da MixAR App, foi utilizada apenas a técnica "Hybrid CAD Model Tracking". Quanto às estruturas escolhidas para efetuar o *tracking*, estas foram: um contentor e as entradas para a Biblioteca Central e para o polo da Escola de Ciências e Tecnologia (Figura 38).



Figura 38 - Estruturas utilizadas nos testes da primeira versão da MixAR App. Da esquerda para a direita: contentor, entrada para o polo da Escola de Ciências e Tecnologia, entrada para a Biblioteca Central.

Antes da realização dos testes *in situ*, foi efetuada uma tarefa *offline* para a preparação e configuração da experiência de RM. Para cada uma das estruturas referidas foi concebida uma configuração da técnica de *tracking* "Hybrid CAD Model Tracking", composta por três ficheiros, conforme descrito na secção 4.1. A criação dos modelos virtuais que aumentam as estruturas (um dos ficheiros presentes na configuração de *tracking*) foi realizada através de

uma ferramenta procedimental proposta por (Adão et al., 2014). A Tabela 4 apresenta as propriedades relacionadas com a complexidade de cada um destes modelos, sendo estas o número de vértices, texturas e faces, as quais serão consideradas para uma posterior análise do desempenho.

Tabela 4 - Propriedades dos diferentes modelos utilizados para aumentar as estruturas do mundo real.

Estrutura a aumentar	Número de vértices	Número de texturas	Número de faces
Contentor	6.504	10	3.552
Biblioteca Central	16.170	17	8.780
Polo da Escola de Ciências e Tecnologia	30.989	19	17.349

Após a preparação *offline* da experiência, a primeira versão da MixAR App foi testada com vista a determinar a sua eficácia e medir o desempenho da unidade móvel. O dispositivo de testes foi um *tablet* Samsung SM-T320 com sistema operativo Android, com um processador de arquitetura ARM de quatro núcleos e uma velocidade de 2.4GHz, 2GB de memória RAM, 16GB de capacidade de armazenamento interno, um ecrã embutido de 8.4" com uma resolução de 2560x1600, uma câmara traseira de 8MP e outros sensores relevantes tais como GPS, giroscópio, acelerómetro, magnetómetro e adaptador de WLAN. Como servidor de RM foi utilizada uma solução de *web hosting* comum e como infraestrutura de rede foi utilizada a rede sem fios 802.11 disponível no *campus* da UTAD. Relativamente às condições de teste, a imagem proveniente da câmara foi configurada para utilizar uma resolução de 800x600 enquanto os algoritmos internos de visão por computador, geridos pelo Metaio SDK, foram configurados para trabalharem com uma resolução de 400x300. Desta forma, reduziu-se o peso computacional causado pela utilização de uma aproximação de *tracking* híbrida. Quanto ao comportamento da aplicação, foi implementado o fluxograma da aplicação tal como descrito na Figura 27. A aplicação começa por transferir do servidor todo o conteúdo necessário para a experiência: a configuração do SIG, as configurações de *tracking* para cada estrutura e os modelos virtuais que irão aumentar as estruturas. Após a transferência do conteúdo a aplicação determina, através do sensor de GPS, a posição do utilizador. Esta é enviada para o servidor para que, com base na posição recebida, calcule o edifício mais próximo do utilizador e devolva o número de identificação deste para a unidade móvel. Através do número de identificação a unidade móvel carrega a configuração de *tracking* adequada e inicializa esta com o estado "Initialize", apresentando o modelo de ajuda

visual ao utilizador. A partir deste momento é assumido o comportamento normal da técnica de *tracking* "Hybrid CAD Model Tracking". O modelo de ajuda visual é apresentado até que se consiga corresponder as arestas detetadas na imagem do mundo real com as presentes no modelo de arestas, após uma correspondência efetuada com sucesso pelo algoritmo de *tracking* o estado é alterado para "*Tracking*" e é consequentemente apresentado um modelo virtual que irá aumentar a estrutura em causa. Quando uma pose de câmara deixar de ser estimada a técnica passa a assumir o estado "*Lost*". Se não for possível a reinicialização do *tracking* a partir da última pose conhecida, o estado é novamente alterado para "*Initialize*" e o processo de *tracking* começa do primeiro passo. Independentemente do estado da técnica de *tracking*, sempre que houver uma deslocação superior a metade do raio do nível de detalhe 1, é despoletada a alteração dinâmica da configuração de *tracking* e a reaplicação do processo acima descrito a partir do estado "*Initialize*".

5.2.2 Resultados e discussão

Para propósitos de testes, a aplicação desenvolvida foi modificada de forma a ignorar as funções de *tracking* durante um curto período de tempo no arranque da mesma. A ideia foi determinar máximo de desempenho da aplicação quando esta apenas obtém as imagens do mundo real e realiza o processo de *rendering* destas no ecrã da unidade móvel. Além disso, o valor pode ser utilizado como base de comparação para o peso computacional que o processo de *tracking* produz. Como resultado foi obtida uma média de 26.6 FPS. Após este período de inicialização, a aplicação volta ao normal ativando as funções de *tracking* e medindo os valores de FPS em cada estado de *tracking* para cada uma das estruturas presentes na experiência. Para efetuar a análise de desempenho, a aplicação registou de forma contínua o número de FPS da aplicação num ficheiro .CSV durante a execução da mesma. A Tabela 5 apresenta a média de FPS obtida para cada estrutura em cada estado da técnica de *tracking* "Hybrid CAD Model".

Tabela 5 - Resultados dos testes da primeira versão do MixAR.

Estrutura\Estado	Initialization (FPS)	Tracking (FPS)	Lost (FPS)
Contentor	17.3	16.4	16.3
Biblioteca	7.5	14.2	7.6
Polo da Escola de Ciências e Tecnologia	11.5	15.5	12.5

Conforme mostra a Tabela 5, o melhor desempenho foi obtido com a configuração do Contentor, seguido do edifício "Engenharias I" e terminando com a configuração da Biblioteca. Apesar de possuir o modelo de aumento mais complexo (Tabela 4), a configuração do "Engenharias I" obteve um desempenho superior à da Biblioteca. A razão para tal poderá estar na Tabela 6, que apresenta informação acerca dos modelos virtuais utilizados para o *tracking*. Esta mostra que o modelo de superfície utilizado para a configuração da Biblioteca possui um número de vértices e faces substancialmente superior aos do modelo utilizado na configuração do "Engenharias I". Por este motivo suponho que os cálculos baseados nos modelos de arestas/superfícies tenham um maior impacto em termos de peso computacional do que a complexidade dos modelos virtuais que aumentam a estrutura.

Tabela 6 - Propriedades dos modelos de ajuda visual, arestas e superfícies utilizados para efetuar o *tracking* a cada estrutura. O modelo de linhas não possui faces, daí a inexistência de tal coluna.

Estrutura	Modelo de ajuda visual		Modelo de arestas	Modelo de superfícies	
	Vértices	Faces	Vértices	Vértices	Faces
Contentor	816	564	8	8	12
Biblioteca	468	542	24	40	60
Polo da Escola de Ciências e Tecnologia	567	622	24	24	36

A Figura 39 apresenta através de *screenshots* os modelos virtuais sobrepostos às suas estruturas correspondentes após um *tracking* efetuado com sucesso.

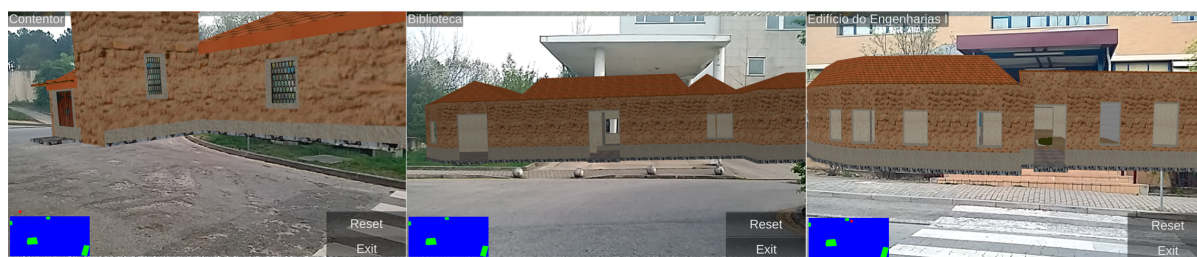


Figura 39 - As diferentes estruturas aumentadas com os modelos virtuais definidos na configuração de *tracking*.

Nesta versão da aplicação a visualização dos modelos por níveis de detalhe não tinha sido implementada. Como resultado apenas um modelo virtual era visualizado a cada momento, sendo este sempre visualizado no seu maior nível de detalhe.

5.3 MixAR App (Versão 2.0)

Nesta segunda versão do software foram efetuados dois tipos de testes, testes de desempenho e testes com utilizadores. Os testes de desempenho tiveram como objetivo conhecer o custo computacional entre as diferentes configurações de unidades móveis e técnicas de *tracking*, através do cálculo de FPS nos vários estados do *tracking* para cada estrutura. Já os testes com utilizadores visaram obter dados acerca da satisfação dos mesmos e também recolher opiniões sobre a sensação de imersividade, presença e conforto na utilização dos dispositivos. Relativamente aos alvos para *tracking*, nesta versão foram utilizadas novas estruturas do mundo real localizadas nos arredores do museu da Vila Velha em Vila Real. Mais especificamente, uma capela, as ruínas de uma habitação antiga e um descampado, daqui em diante denominadas de estrutura A, B e C, respetivamente (Figura 40).



Figura 40 - Vista aérea das estruturas do mundo real utilizadas para a realização de testes com a versão 2.0 da MixAR App. No centro da imagem encontra-se o museu da Vila Velha, ao seu redor são destacadas as estruturas utilizadas para as experiências, nomeadamente: a capela (A), as ruínas (B) e o descampado (C).

Em ambos os testes (desempenho e com utilizadores) foram utilizados os mesmos alvos para efetuar o *tracking*. No entanto, os testes em si foram bastante diferentes pois tinham objetivos distintos. Os testes com utilizadores foram realizados no local da experiência (ambiente exterior) e testaram duas configurações: uma imersiva e outra não imersiva. Os testes de desempenho foram efetuados num local remoto (ambiente interior), onde os alvos de

tracking foram simulados através de fotografias das estruturas, e utilizaram cinco configurações: três imersivas e duas não imersivas.

Relativamente às técnicas de *tracking* utilizadas, apesar dos alvos de *tracking* serem comuns em ambos os tipos de testes, devido novamente aos diferentes objetivos por parte de cada teste nem sempre foram utilizadas as mesmas técnicas nas estruturas. Nos testes com utilizadores foram utilizadas as técnicas mais adequadas para cada situação, nos testes de desempenho a ideia foi obter um índice de desempenho do sistema nas mais variadas situações. Por este motivo estes utilizaram o maior número de técnicas de *tracking*, sendo utilizada uma técnica diferente para cada estrutura presente na experiência.

Nos testes de desempenho, um ambiente interior foi utilizado para a realização das experiências ao invés do ambiente real (exterior) das estruturas para que as condições de iluminação fossem controladas e dessa forma garantir condições o mais semelhantes possível entre as diferentes configurações. Pelo mesmo motivo, os parâmetros de câmara utilizados foram idênticos entre as configurações, tendo sido utilizada uma resolução de 640x480 para visualizar as imagens do mundo real no ecrã e um *downsample* de valor 2, resultando numa imagem com uma resolução de 320x240 para os algoritmos de visão por computador. Outro factor que levou à escolha de um ambiente interior para a realização dos testes de desempenho deveu-se à incapacidade de alguns dispositivos em conseguirem desempenhar as suas funções corretamente no local da experiência. Nomeadamente a câmara do HMD Vuzix *video see-through* que, por não possuir um controlo da luminosidade, apresentava a imagem do mundo real como um clarão branco, o HMD Vuzix *optical see-through* que, devido à luz exterior, não permitia ver claramente os conteúdos virtuais apresentados e finalmente, o SBC que por não possuir uma bateria não podia ser utilizado de forma portátil. Quanto aos testes com utilizadores, a ideia foi fornecer a melhor experiência possível para cada configuração, o que resultou na utilização de diferentes parâmetros de câmara devido às diferentes capacidades de cada um. Na configuração não imersiva (*tablet*) foi utilizada uma resolução de câmara 640x480 com um *downsample* de 2 e no não imersivo (*smartphone*) foi utilizada uma resolução de câmara de 320x240 com um *downsample* de 1. O *downsample* é um parâmetro da câmara que define um valor inteiro pelo qual se divide a resolução da câmara para os algoritmos internos de visão por computador, sendo o seu objetivo conseguir apresentar uma boa imagem ao utilizador enquanto os algoritmos internos trabalham com uma resolução mais reduzida para que se consiga obter um bom nível de desempenho em todos os dispositivos.

5.3.1 Testes de desempenho

Estes testes tiveram como objetivo permitir uma comparação de desempenho entre as diferentes configurações utilizadas e permitir uma comparação de desempenho entre as diferentes técnicas de *tracking* utilizadas. Os testes foram realizados num ambiente interior, uma vez que foi utilizado um ambiente remoto as estruturas do mundo real foram simuladas através de fotografias das mesmas. Pelo mesmo motivo a aplicação foi ligeiramente alterada para que se pudesse simular a posição do utilizador. Esta simulação é feita através do botão "Switch" (Figura 41), o qual injeta de forma alternada as coordenadas centrais de cada estrutura presente na experiência.



Figura 41 - Introdução do botão "Switch" para simular a posição do utilizador para a coordenada central das estruturas alternando assim entre configurações de *tracking*.

Foram testados um total de cinco configurações, entre elas três imersivas e duas não imersivas. As características dos dispositivos que constituem cada solução são apresentadas na Tabela 7.

Tabela 7 - Características dos dispositivos que constituem cada configuração.

Unidade móvel Características		Computador portátil	SBC	Tablet	Smartphone	Smartphone com óculos 3D
Processamento	CPU	Intel Core i7 2630QM 2GHz	Intel Core i7 610E 2.53GHz	Krait 400 Quad-core 2.3GHz	MediaTek MT6592 2.0GHz	MediaTek MT6592 2.0GHz
	GPU	Nvidia GeForce 540M 1GB	Intel HD Graphics	Adreno 330	Mali 450	Mali 450
	Armazenamento /Memória	250GB e 8GB RAM	200GB e 4GB RAM	16GB e 2GB RAM	16GB e 2GB RAM	16GB e 2GB RAM
Visualização		Vuzix STAR 1200XLD	Vuzix 1200DXAR	Monitor de 8.4"	Monitor de 5"	Monitor 5" + Óculos Neje Colorcross
Contexto	GPS	Navicom NAV-GP01S	Navicom NAV-GP01S	Recetor GPS integrado	Recetor GPS integrado	Recetor GPS integrado
	Sensores Inerciais	Vuzix Wrap Tracker	Vuzix Wrap Tracker	Sensores integrados	Sensores integrados	Sensores integrados
	Câmara	Câmara Full HD	Câmaras VGA	Câmara traseira de 8MP	Câmara traseira de 13MP	Câmara de 13MP
Comunicação		Placa integrada WLAN	Placa USB WLAN	Placa integrada WLAN	Placa integrada WLAN	Placa integrada WLAN
Alimentação		Bateria integrada	Ligação AC	Bateria integrada de 4800mAh	Bateria integrada de 2500 mAh	Bateria integrada de 2500 mAh

Para efetuar a análise de desempenho, a aplicação regista num ficheiro .CSV o número de FPS de forma constante durante a execução da aplicação. O número de FPS é posteriormente separado pela estrutura alvo de *tracking* e pelo estado do *tracking*.

De referir que os estados do *tracking* considerados são "*Tracking*", quando uma pose de câmara é estimada e por consequente a estrutura aumentada, e "*Not tracking*" pelo caso inverso, quando não existe uma estimativa de pose de câmara. Os estados são diferentes daqueles registados para a primeira versão devido ao suporte para diversas técnicas, apenas foram utilizados os estados que todas as técnicas de *tracking* têm em comum.

De modo a testar o desempenho das diferentes técnicas de *tracking* as configurações utilizadas para cada estrutura foram alteradas de forma a que fossem utilizadas três técnicas

de *tracking* distintas, sendo estas "Hybrid CAD Model Tracking" para a estrutura A, "3D Map Tracking" para a estrutura B e "Image Tracking" para a estrutura C. A próxima secção apresenta os resultados e discussão dos testes de desempenho efetuados na segunda versão da MixAR App.

5.3.2 Resultados e discussão

Os resultados são apresentados na Tabela 8, através da sua análise podemos retirar algumas conclusões. De um modo geral entre todas as configurações podemos ver que a técnica "Hybrid CAD Model" é a única que possui valores de FPS inferiores quando se encontra num estado de "*Not tracking*", a razão para tal poderá estar relacionada com o processo de reinicialização da técnica. Durante esta fase a última pose de câmara estimada é utilizada pela técnica tentar reconhecer essa pose ao mesmo tempo que tenta corresponder pontos 3D do mapa de pontos 3D criado pelo algoritmo SLAM até então.

Tabela 8 - Resultados dos testes de desempenho efetuados à segunda versão da MixAR App.

Estrutura e técnica	A (Hybrid CAD Model Tracking)		B (3D Map Tracking)		C (Image Tracking)	
	Tracking	Not tracking	Tracking	Not tracking	Tracking	Not tracking
Tablet	18,03	9,12	20,46	28,92	17,54	28,34
Smartphone Normal	30,25	25,79	28,83	30,10	28,53	30,15
Smartphone Stereo	26,81	23,14	28,04	30,13	21,96	26,27
SBC + vídeo HMD	31,00	30,66	31,00	30,89	30,89	30,89
Portátil + optical HMD	30,26	30,23	30,24	30,33	30,45	30,23

A solução que obteve o melhor desempenho foi a "SBC + video HMD", obtendo valores perto dos 31 FPS. Com um desempenho ligeiramente inferior, apresenta-se a solução "Portátil + optical HMD", com valores à volta dos 30 FPS. Tendo em conta as características dos dispositivos que constituem ambas as configurações (Tabela 7) este era um resultado expectável. Estes valores permitiram também apurar aquilo que parece ser o limite máximo de desempenho da aplicação, ultrapassando a média de 26.6 FPS obtida na versão 1.0 da MixAR

App. A diferença entre o limite de FPS obtidos em cada versão é facilmente explicado pelo facto de que foram utilizados parâmetros de câmara diferentes. A versão 1.0 utilizou uma resolução de câmara de 800x600 com um *downsample* de 2 e a versão 2.0 utilizou uma resolução um pouco inferior, 640x480 igualmente com um *downsample* de 2.

Comparando as configurações do *smartphone* podemos ver que a versão normal foi praticamente sempre superior à *stereo*. A razão de a versão normal ter um desempenho superior em relação à versão *stereo* é facilmente explicada pelo facto de que a última duplica o conteúdo que é apresentado no ecrã da unidade móvel. Por fim, em relação ao *tablet*, este mostrou ter o pior desempenho entre todas as configurações chegando a obter um mínimo de 9 FPS num estado de "*Not tracking*" na estrutura A. Além disso, foi a solução com a maior quebra de desempenho entre estados, obtendo quebras entre os 8 e os 11 FPS (em média) na transição do estado "*Tracking*" para "*Not tracking*".

Uma das desvantagens de utilizar um software proprietário como o Metaio é o desconhecimento público do código fonte por detrás da implementação das técnicas, o que torna impossível para mim dizer ao certo onde está o peso computacional, por este motivo algumas das análises efetuadas aos resultados são feitas com base em suposições de como funcionam as técnicas internamente.

5.3.3 Testes com utilizadores

Os testes com utilizadores foram efetuados com base em duas configurações com diferentes níveis de imersão. Foi utilizado um *tablet* Samsung SM-T320 como solução não-imersiva e um *smartphone* BQ Aquaris E5 HD, combinado com uns óculos de RM Neje Colorcross, como solução imersiva (Figura 42).



Figura 42 - Óculos de RM Neje Colorcross, *smartphone* BQ Aquaris E5 HD e *tablet* Samsung SM-T320.
 Imagens obtidas de: http://img.dxcn.com/productimages/sku_332676_3.jpg,
http://innovainformatica.com/images_url_amigables/1520/smartphone-bq-aquaris-e5-hd-ips-qc-16g-a4-4-b-n-c000060-grande.jpg e [http://images.samsung.com/is/image/samsung/pt_SM-T320NZWATPH_000234416_Front_black?\\$TM-Gallery\\$](http://images.samsung.com/is/image/samsung/pt_SM-T320NZWATPH_000234416_Front_black?$TM-Gallery$), respetivamente.

Para a tarefa de *tracking*, duas técnicas distintas foram utilizadas. Para a estrutura A (Figura 40) utilizou-se a técnica de *tracking* "Hybrid CAD Model Tracking" e para as estruturas B e C (Figura 40) a técnica "3D Map Tracking". Tal como ilustra a Figura 43, a estrutura A possui arestas bem definidas assim como superfícies com pouca textura. Estes foram os fatores que levaram à escolha da técnica "Hybrid CAD Model Tracking", uma vez que é indicada para estes casos.



Figura 43 - Estrutura A (capela).

Dada a natureza baseada em modelos da técnica "Hybrid CAD Model Tracking", para que o *tracking* fosse efetuado na estrutura A, foi necessária a criação do conjunto de modelos 3D que acompanha qualquer cena/objeto onde seja utilizada esta técnica de *tracking*. Sendo estes os modelos de arestas, superfície e ajuda visual, visíveis na Figura 44.

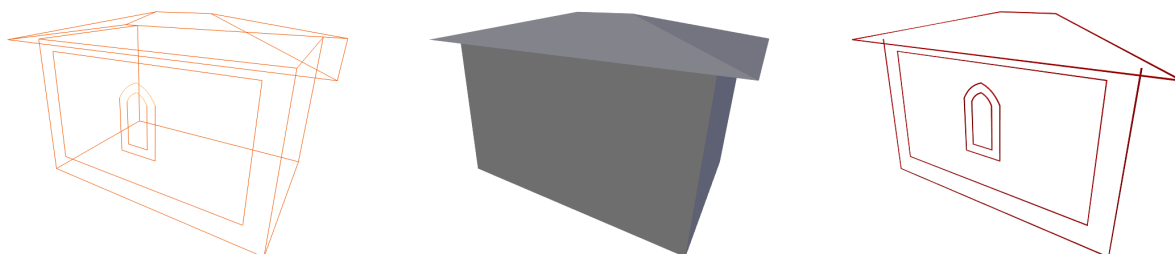


Figura 44 - Modelo de arestas, superfície e ajuda visual criados para a estrutura A.

Para as estruturas B e C recorreu-se à utilização da técnica de *tracking* "3D Map Tracking". O motivo deveu-se ao facto de ambas as estruturas possuírem um elevado número de arestas assim como um alto nível de textura, o que pode causar falsas deteções de arestas e por consequente uma não deteção de pose ou deteção de pose errada quando utilizada a mesma técnica que na estrutura A ("Hybrid CAD Model Tracking"). Semelhante à técnica de *tracking* utilizada na estrutura A, a técnica "3D Map Tracking" requer que se adquira informação sobre o objeto/cena numa fase prévia ao *tracking*, no entanto ao invés de modelos 3D, esta requer que se adquira um mapa de pontos 3D. Tal é feito através da aplicação *Metaio Toolbox*, disponível para *Android* e *iOS*. A diferença entre estas técnicas é que, na fase *online* do *tracking*, ao invés de tentar corresponder arestas presente num modelo de arestas ("Hybrid CAD Model Tracking"), a técnica "3D Map Tracking" tenta corresponder os pontos característicos presentes no mapa 3D com os pontos característicos detetados em velocidade interativa na imagem captada pela câmara.

A Figura 45 e a Figura 46 apresentam *screenshots* tirados pelo *tablet* durante a criação dos mapas de pontos 3D para as estruturas B e C através do *Metaio Toolbox*. Nestes pode-se constatar que são registados 278 pontos para a estrutura B e 252 pontos para a estrutura C, ilustrados como pixéis de cor amarela. Na Figura 47, o mapa da estrutura C é apresentado após a sua importação para o *Metaio Creator*. O *Metaio Creator* permite a edição dos mapas criados com o *Metaio Toolbox*, podendo-se remover pontos que sejam considerados *outliers* de modo a melhorar a precisão do *tracking*. Para que a edição seja mais simples para o

utilizador os pontos que constituem o mapa 3D são aumentados e a sua cor alterada para vermelho.



Figura 45 - Screenshot tirado pelo tablet na criação do mapa de pontos 3D para a estrutura B.

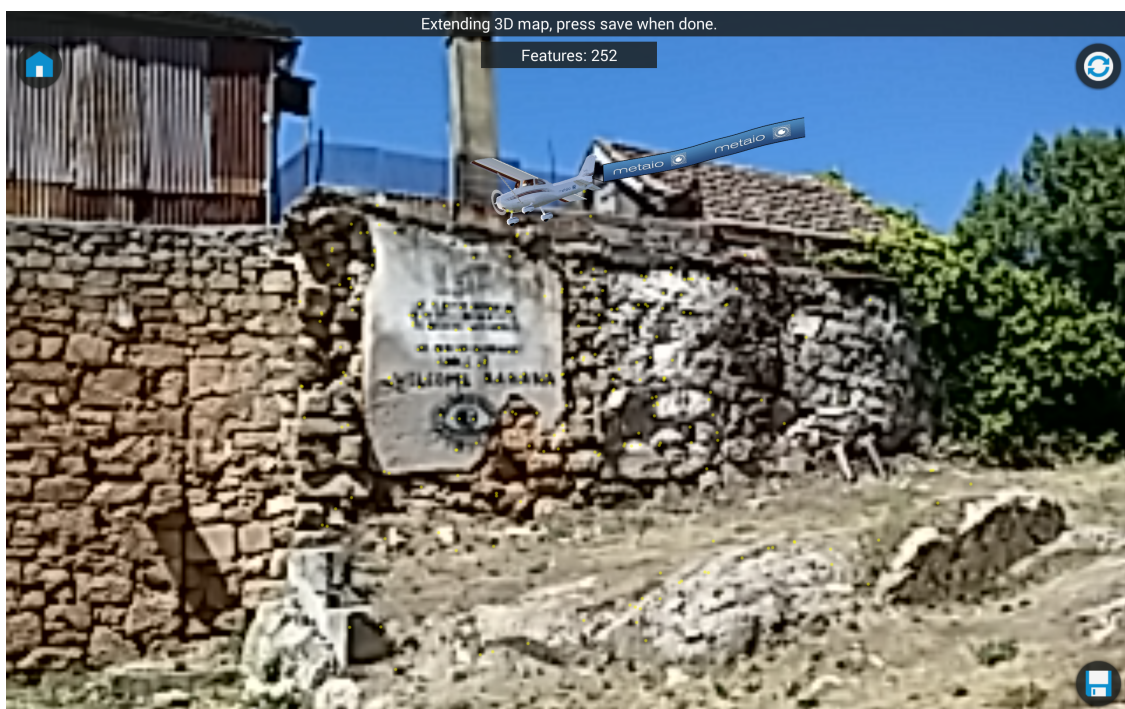


Figura 46 - Screenshot tirado pelo tablet na criação do mapa de pontos 3D para a estrutura C.

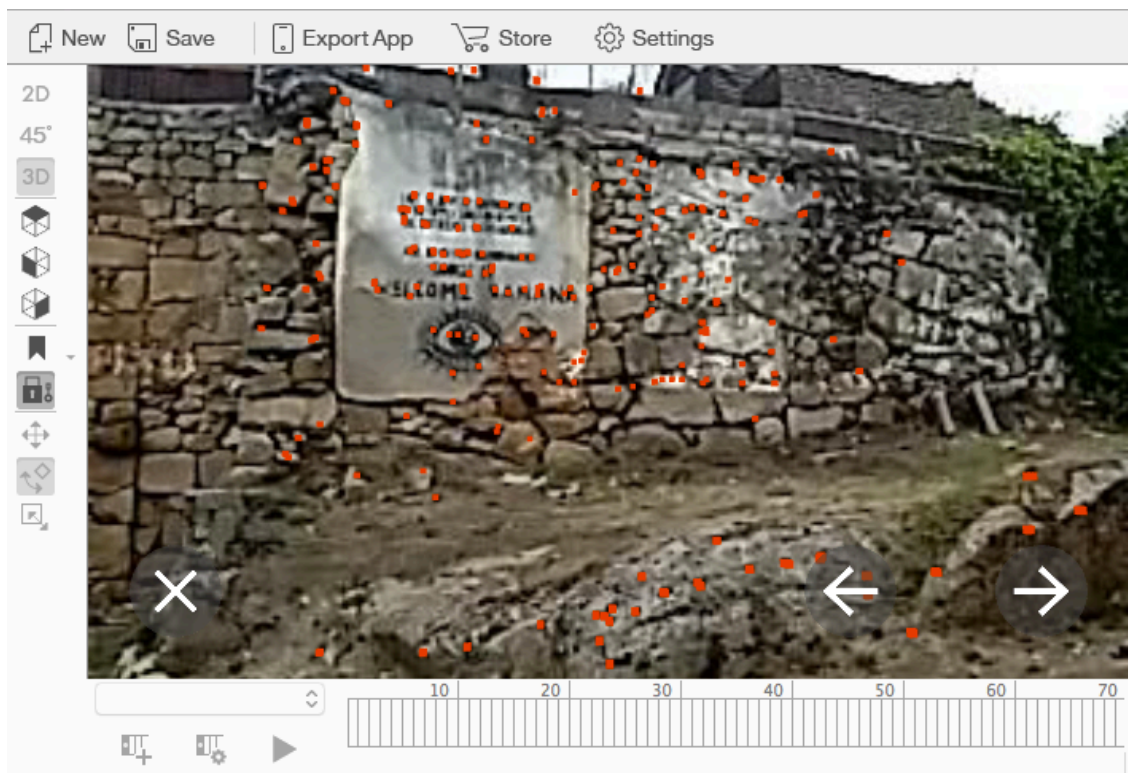


Figura 47 - Mapa de pontos 3D da estrutura C após a sua importação para o *Metaio Creator*.

Durante os testes no terreno, os utilizadores que iam passando na rua eram convidados a realizar os testes, ora com uma solução, ora com outra (Figura 48). Aos que aceitaram realizar a experiência, foi-lhes explicado o propósito da mesma, bem como a ordem como as estruturas teriam que ser visitadas: primeiro a capela (estrutura A), depois a ruína (estrutura B) e por último o descampado (estrutura C). Durante as experiências com a solução imersiva, os utilizadores eram devidamente acompanhados pelos membros da equipa de desenvolvimento para prevenir eventuais acidentes no percurso. No final foi-lhes pedido para responder ao questionário que se encontra em anexo neste documento (Anexo A). Este questionário foi elaborado com o apoio dos trabalhos cientificamente relevantes: (Kaufmann & Dünser, 2007), (Jennett et al., 2008), (Olsson, Kärkkäinen, Lagerstam, & Ventä-Olkkonen, 2012), (Olsson & Salo, 2011), (Witmer & Singer, 1998).



Figura 48 - Utilização das configurações imersiva (estrutura B) e não imersiva (estrutura A), respetivamente.

5.3.4 Resultados e discussão

No total 25 pessoas com idades compreendidas entre os 17 e os 32 anos participaram na experiência e preencheram um ou dois questionários, dependendo dos testes que realizaram, resultando num total de 42 testes com utilizadores, onde 21 destes testes foram realizados com o *tablet* (solução não-imersiva) e os outros 21 realizados com os óculos de RM juntamente com o *smartphone* (solução imersiva). Em termos de utilizadores, 68% são do género masculino e 32% do feminino.

Do questionário são analisadas apenas as perguntas com relevância para o âmbito desta dissertação, nomeadamente perguntas relacionadas com o *tracking*. Em termos de satisfação, na pergunta "Foi difícil visualizar os edifícios durante a experiência", a solução imersiva foi a que obteve os melhores resultados. No caso da solução imersiva, um total de 66% respondeu discordo ou discordo totalmente, 10% respondeu neutro e 24% selecionaram concordo ou concordo totalmente. Já na solução não imersiva, 48% selecionaram discordo como resposta, 28% neutro e 24% discordo ou discordo totalmente (Figura 49). Isto revela que houve uma maior facilidade na solução imersiva por parte dos utilizadores em visualizar os edifícios virtuais após um *tracking* efetuado com sucesso. Na nossa convicção, esta diferença nos resultados é explicada pela própria configuração em si, após observar alguns testes reparei que assim que a inicialização era efetuada com sucesso os utilizadores demonstravam uma vontade de visualizar de imediato todos os cantos do modelo virtual antes de proceder a uma análise mais cuidada do modelo, realizando várias vezes movimentos bruscos e perdendo o *tracking*. No entanto, ao utilizar a cabeça (HMD) para realizar a rotação da câmara ao invés das duas mãos estes movimentos eram suavizados o que permitia que uma

pose de câmara continuasse a ser detetada pelos algoritmos de *tracking*, resultando numa melhor experiência.

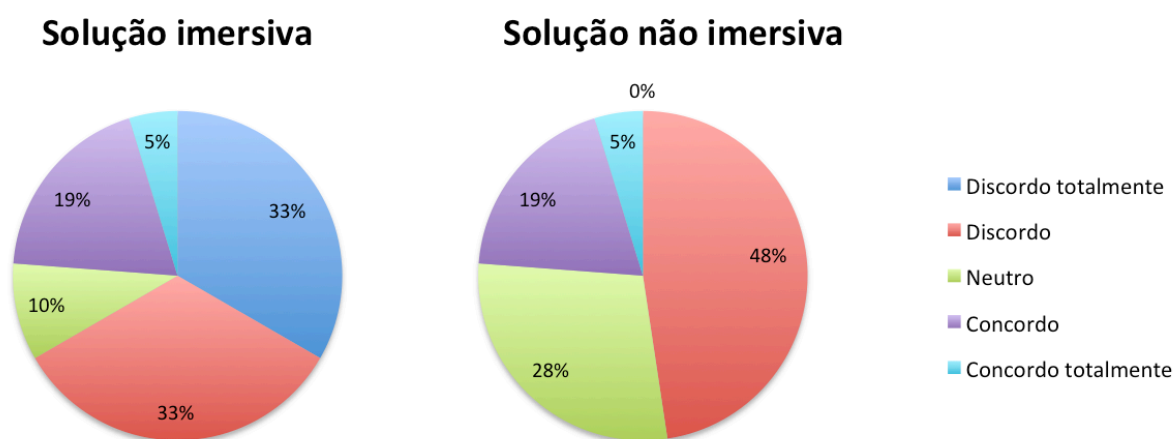


Figura 49 - Respostas dos utilizadores para ambas as configurações à pergunta "Foi difícil visualizar os edifícios durante a experiência".

De modo a obter um *feedback* acerca do registo no processo de *tracking* foi realizada a pergunta "Os conteúdos visualizados estavam sincronizados com o mundo real". Esta obteve resultados idênticos e satisfatórios em ambas as soluções com 71% dos questionários a seleccionarem a opção concordo e concordo totalmente, 19% neutro, apenas 10% discordo e 0% discordo totalmente (Figura 50). Face aos resultados obtidos, podemos constatar que a tarefa de registo foi realizada com sucesso pela MixAR App.

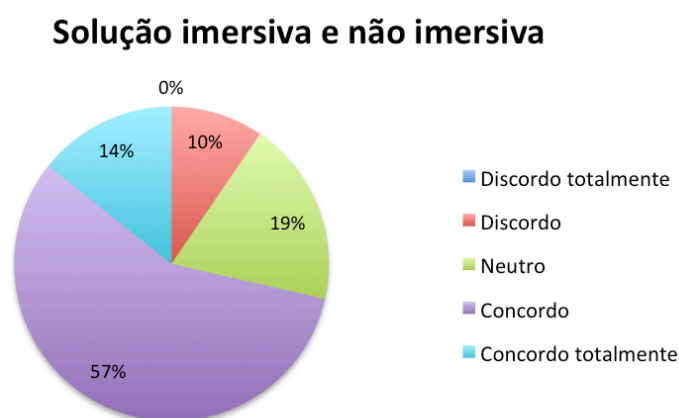


Figura 50 - Respostas dos utilizadores para ambas as configurações à pergunta "Os conteúdos visualizados estavam sincronizados com o mundo real".

Para avaliar o comportamento dos modelos virtuais que aumentam as estruturas durante o *tracking* foi apresentada a pergunta "Os modelos eram apresentados de uma forma robusta (sem saltos) e fluída", onde foram obtidos resultados díspares entre as duas configurações. Na solução não-imersiva, aproximadamente 48% selecionaram discordo, 24% neutro, 19% concordo e 9% concordo totalmente. Já na solução imersiva, apenas 10% selecionaram discordo totalmente ou discordo, 19% selecionaram neutro, 62% concordo e 9% concordo totalmente (Figura 51).

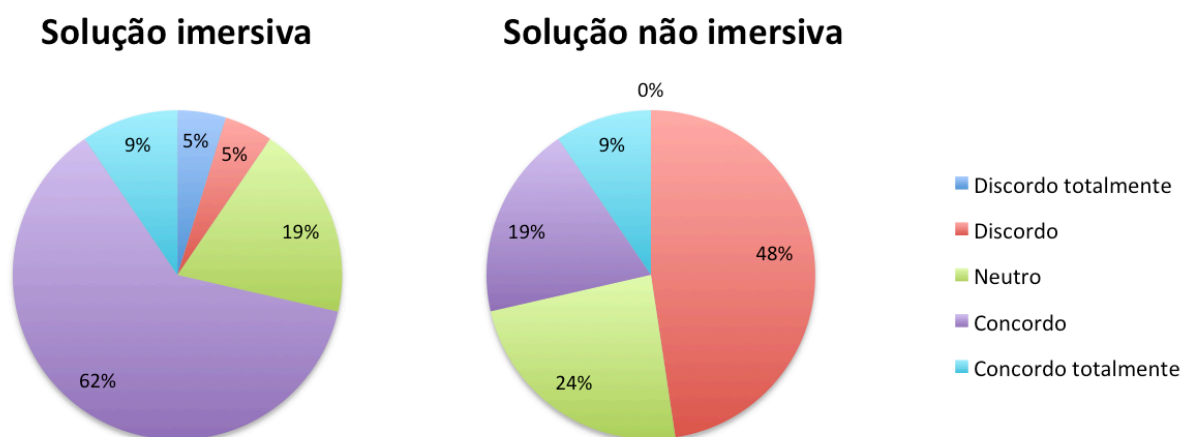


Figura 51 - Respostas dos utilizadores para ambas as configurações à pergunta "Os modelos eram apresentados de uma forma robusta (sem saltos) e fluída".

Como podemos ver pelos resultados, os utilizadores obtiveram uma melhor experiência na visualização dos modelos com a solução imersiva. O motivo prende-se com o próprio desempenho das soluções, a solução não-imersiva tem um desempenho inferior como foi visto na secção 5.3.1 - Testes de desempenho. Como resultado a aplicação obtém menos FPS, o que afeta a fluidez com que se visualizam os modelos. Em relação aos saltos, estes são conhecidos como *jitter*, onde os modelos parecem tremer/saltar durante o *tracking*. Estes são provocados pelo próprio algoritmo de *tracking* pois aplica transformações ao modelo (de arestas) com vista a obter correspondências com imagens do mundo real. Um desempenho inferior produz mais *jitter* pois menos transformações são aplicadas e testadas até que a pose seja descartada e o estado assumido como "Not tracking". Outro fator que beneficiou a solução imersiva foi a utilização de uma resolução de ecrã mais baixa (320x240), o que lhe reduziu a carga computacional quando comparado com a solução não-imersiva (640x480). A solução não-imersiva utilizou uma resolução um pouco superior pois ao utilizar a mesma que a solução imersiva tornava-se visível uma boa perda na qualidade visual.

Em termos de presença, à pergunta "Fui capaz de aproximar-me dos edifícios para os observar", ambas as configurações obtiveram 0% dos questionários com a opção discordo totalmente ou discordo selecionada. Ambas tiveram resultados positivos, a solução não-imersiva teve aproximadamente 19% neutro, 71% concordo e 10% concordo totalmente enquanto a solução imersiva aproximadamente 5% neutro, 67% concordo e 28% concordo totalmente (Figura 52). Através dos resultados, podemos constatar que ambas as soluções tiveram desempenhos satisfatórios na realização desta tarefa, permitindo que os utilizadores se chegassem perto dos modelos virtuais para os poderem realizar uma observação mais cuidada destes.

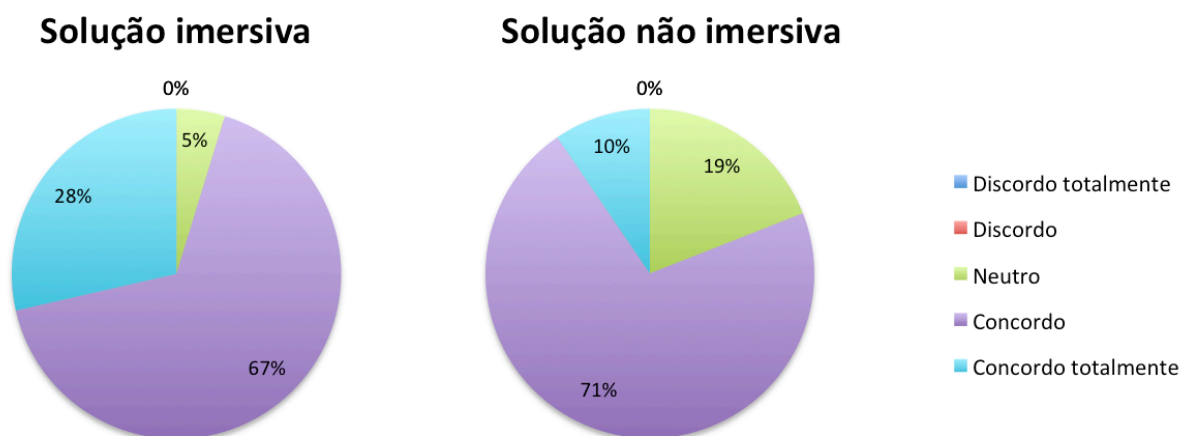


Figura 52 - Respostas dos utilizadores para ambas as configurações à pergunta "Fui capaz de aproximar-me dos edifícios para os observar".

Por fim, a pergunta "Conseguir visualizar os edifícios virtuais de pontos de vista diferentes" relaciona-se com a robustez do *tracking* em conseguir continuar a detetar uma determinada estrutura enquanto o utilizador muda o ponto de vista, por vezes entrando dentro dos edifícios virtuais para visualizar o seu interior num ambiente que pode ser definido como VA (Figura 53). Ambas as configurações obtiveram 0% de respostas discordo totalmente e 10% discordo. A solução não-imersiva obteve aproximadamente 34% neutro, 48% concordo e 10% concordo totalmente. A solução imersiva obteve aproximadamente 14% neutro, 67% concordo, 10% concordo totalmente (Figura 54).



Figura 53 - Visualização de um ambiente de virtualidade aumentada onde é apresentado o interior do edifício virtual que aumenta a estrutura A.



Figura 54 - Respostas dos utilizadores para ambas as configurações à pergunta "Conseguir visualizar os edifícios virtuais de pontos de vista diferentes".

Com base nos testes e resultados apresentados nas secções 5.2 e 5.3, pode-se concluir que, de uma forma geral, as soluções imersivas conseguiram resultados mais satisfatórios que as soluções não imersivas. Em termos de desempenho este seria um resultado expectável devido às diferenças de hardware entre as unidades móveis imersivas e não imersivas. Nos testes com utilizadores, os mesmos sentiram-se mais satisfeitos ao desempenharem as tarefas

num ambiente mais imersivo. Por este motivo pode-se constatar uma preferência na utilização da unidade móvel imersiva nos resultados dos questionários.

6 Conclusões e Trabalho Futuro

O presente trabalho consistiu no desenvolvimento, e consequente integração, de uma metodologia de *tracking* num sistema de RM adaptativa proposto por (Magalhães et al., 2014). O sistema de RM, denominado MixAR, pretende permitir, aos seus utilizadores, a visualização de reconstruções virtuais, de edifícios presentes em sítios arqueológicos, sobrepostas nas ruínas/estruturas atuais, caso subsistam, ou no local onde outrora existiram. A necessidade de implementar, uma metodologia de *tracking* para detetar as estruturas do mundo real, e posteriormente, aumentá-las com determinados modelos virtuais foi o que motivou a realização desta dissertação.

O objetivo do MixAR, do ponto de vista do utilizador, é permitir que estes ingressem numa experiência de RM, onde podem movimentar-se livremente por um sítio arqueológico enquanto visualizam reconstruções virtuais de edifícios, utilizando apenas uma unidade móvel. Para que tal aconteça, o sistema MixAR define uma arquitetura cliente-servidor, sendo os seus componentes principais: uma unidade móvel (cliente), utilizada para proporcionar aos seus utilizadores a experiência de RM, através da visualização de reconstruções virtuais; um servidor de RM, utilizado para armazenar e fornecer o acesso a conteúdo digital quando requisitado pelos clientes, assim como agir como uma unidade de processamento remoto; uma infraestrutura de rede, para suportar a comunicação entre o servidor e uma ou mais unidades móveis.

O trabalho aqui desenvolvido, foi integrado na aplicação que é executada na unidade móvel. Designada de MixAR App, a aplicação é responsável por executar o processo de *tracking*, além disso, estabelece a comunicação com o servidor de RM através de *web services*. A implementação da aplicação foi dividida em duas versões: uma primeira versão (1.0), caracterizada pela obtenção de um primeiro protótipo funcional, seguido da realização de testes de desempenho. Nesta versão, o *tracking* era efetuado com recurso a apenas uma técnica de *tracking*; uma segunda e última versão (2.0), caracterizada pela adição de um sistema de *tracking* multimodal e implementação de um técnica de *tracking* baseada em sensores, designada de ambiente estendido, para o processo de *tracking*, seguido de novos testes de desempenho e a realização de testes com utilizadores. Até à data, a tarefa de *tracking* continua a ser um problema sem uma única solução perfeita. Existem diversos tipos e implementações de técnicas de *tracking*, no entanto, estas possuem sempre vantagens e

desvantagens, o que as torna mais adequadas para determinados casos do que outros. Por este motivo a versão 2.0 da MixAR App implementou um sistema de *tracking* multimodal. Isto significa que suporta várias técnicas de *tracking*, mais especificamente suporta as técnicas: "Hybrid CAD Model Tracking", "3D Map Tracking", "Extended Image Tracking" e "Image Tracking". Todas estas têm em comum o facto de que exigem algum tipo de informação prévia ao *tracking*. Esta informação, no formato de modelos 3D, mapa de pontos 3D, ou imagem, é essencial pois é através dela que se obtém controlo sobre o sítio onde será feito o aumento.

Os testes com utilizadores foram realizados nos arredores do museu da Vila Velha, em Vila Real, tendo participado um total de 25 pessoas com idades entre os 17 e os 32 anos de idade. Como unidades móveis foram utilizadas duas soluções distintas, uma imersiva, constituída por uns óculos de RM (Neje Colorcross) e um *smartphone* (BQ Aquaris E5 HD), e uma não-imersiva, constituída por um *tablet* (Samsung SM-T320). As pessoas realizavam a experiência e preenchiam um ou dois questionários, conforme as unidades móveis que haviam experimentado. A experiência consistiu no aumento de três estruturas do mundo real com três modelos virtuais distintos. Destas experiências resultaram 42 testes, onde 21 foram realizados com a solução imersiva e 21 com a solução não-imersiva. Uma análise dos resultados levou à conclusão de que a solução imersiva conseguiu melhores resultados de uma forma constante, dando aos utilizadores uma melhor experiência na visualização dos modelos e uma maior facilidade em efetuar e manter o *tracking*.

No decorrer do trabalho, foram publicados dois artigos de conferência internacional indexados no Scopus: (Narciso, Pádua, Adão, Peres, & Magalhães, 2015), onde é apresentada a versão protótipo (1.0) do sistema de RM MixAR, com foco na unidade móvel. São apresentados os resultados dos testes de desempenho, efetuados com o protótipo, bem como uma discussão dos mesmos; (Pádua, Narciso, Cunha, Peres, & Magalhães, 2015), que propõe e compara quatro configurações distintas de unidades móveis, com diferentes níveis de imersão, para serem utilizados no sistema MixAR. A comparação é feita com base no custo, peso e desempenho das unidades móveis.

Pode-se concluir que os objetivos a que esta dissertação se propôs foram alcançados com sucesso, uma vez que todas as metas propostas foram atingidas e a MixAR App se encontra funcional. Dada a utilização de uma ferramenta de geração procedimental para gerar

automaticamente os modelos virtuais, para trabalho futuro, pode-se destacar a realização de novos testes com utilizadores utilizando reconstruções virtuais de ruínas/estruturas presentes num determinado sítio arqueológico. Juntamente com a realização de novos testes com utilizadores, poderiam ser utilizadas as unidades móveis que obtiveram os melhores resultados nos testes de desempenho, apresentados na secção 5.3.1. Dado o desempenho mais elevado, a qualidade de imagem poderia ser superior à utilizada nos testes com utilizadores, possibilitando a obtenção de melhores resultados nos questionários. Ainda como trabalho futuro, poderia ser estudada/testada uma forma de automatizar o processo de modelação das estruturas, atualmente efetuado manualmente através da ferramenta Blender, recorrendo a câmaras de profundidade tais como Microsoft Kinect ou ASUS Xtion. A obtenção automatizada de um modelo virtual 3D da estrutura iria facilitar, assim como acelerar, o processo de criação das configurações de *tracking* que requerem modelos das estruturas *a priori*.

Referências Bibliográficas

- Adão, T., Magalhães, L., Peres, E., & Pereira, F. (2014). *Procedural Generation of Traversable Buildings Outlined by Arbitrary Convex Shapes*. Paper presented at the Conference on ENTERprise Information Systems.
- Afif, F. N., Basori, A. H., & Saari, N. (2013). Vision-based Tracking Technology for Augmented Reality: A Survey. *International Journal of Interactive Digital Media*, 1(1), 46-49.
- Akenine-Möller, T., Haines, E., & Hoffman, N. (2008). *Real-Time Rendering, Third Edition*: CRC Press.
- Azuma, R., Bailiot, Y., Behringer, R., Feiner, S., Julier, S., & MacIntyre, B. (2001). Recent Advances in Augmented Reality. *Computer Graphics and Applications, IEEE*, 21(6), 34-47.
- Azuma, R. T. (1997). A survey of augmented reality. *Presence*, 6(4), 355-385.
- Bajura, M., & Neumann, U. (1995). Dynamic Registration Correction in Video-Based Augmented Reality Systems. *Computer Graphics and Applications, IEEE*, 15(5), 52-60.
- Brooks Jr, F. P. (1999). What's real about virtual reality? *Computer Graphics and Applications, IEEE*, 19(6), 16-27.
- Burdea, G. C., & Coiffet, P. (2003). *Virtual Reality Technology* (Vol. 1): John Wiley & Sons.
- Cai, S., Wang, X., & Chiang, F.-K. (2014). A case study of Augmented Reality simulation system application in a chemistry course. *Computers in Human Behavior*, 37, 31-40.
- Castro, P., Chiu, P., Kremenek, T., & Muntz, R. (2001). *A probabilistic room location service for wireless networked environments*. Paper presented at the Ubicomp 2001: Ubiquitous Computing.
- Caudell, T. P., & Mizell, D. W. (1992). *Augmented reality: An application of heads-up display technology to manual manufacturing processes*. Paper presented at the System Sciences, 1992. Proceedings of the Twenty-Fifth Hawaii International Conference on.
- Cho, Y., & Neumann, U. (2001). Multi-ring Color Fiducial Systems for Scalable Fiducial Tracking Augmented Reality. *Presence*, 10(6), 599-612.
- Comport, A. I., Marchand, E., Pressigout, M., & Chaumette, F. (2006). Real-Time Markerless Tracking for Augmented Reality: The Virtual Visual Servoing Framework. *IEEE Transactions on Visualization and Computer Graphics*, 12(4), 615-628.
- Davison, A. J., Reid, I. D., Molton, N. D., & Stasse, O. (2007). MonoSLAM: Real-Time Single Camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6), 1052-1067.
- Doignon, C. (2007). An Introduction to Model-Based Pose Estimation and 3-D Tracking Techniques *LSIIT - University of Strasbourg* (pp. 530-555).
- Dunston, P. S., & Wang, X. (2005). Mixed Reality-Based Visualization Interfaces for Architecture, Engineering, and Construction Industry. *Journal of Construction Engineering and Management*, 131(12), 1301-1309.
- Finkelstein, A. (2005). *Part I: Non-photorealistic Rendering*. Paper presented at the Line Drawings from 3D Models SIGGRAPH 2005 Course Notes.
- Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381-395.

- Grajewski, D., Górski, F., Zawadzki, P., & Hamrol, A. (2013). *Application of Virtual Reality Techniques in Design of Ergonomic Manufacturing Workplaces*. Paper presented at the International Conference on Virtual and Augmented Reality in Education.
- Haller, M. (2004). *Photorealism or/and non-photorealism in augmented reality*. Paper presented at the Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry.
- Höllerer, T., & Feiner, S. (2004). Mobile Augmented Reality. *Telegeoinformatics: Location-Based Computing and Services*, 21.
- Holloway, R., & Lastra, A. (1993). Virtual environments: A survey of the technology. *Eurographics*.
- Honkamaa, P., Siltanen, S., Jäppinen, J., Woodward, C., & Korkalo, O. (2007). *Interactive outdoor mobile augmentation using markerless tracking and GPS*. Paper presented at the Proceedings Virtual Reality International Conference (VRIC).
- Jayaram, S., Connacher, H. I., & Lyons, K. W. (1997). Virtual assembly using virtual reality techniques. *Computer-Aided Design*, 29(8), 575-584.
- Jennett, C., Cox, A. L., Cairns, P., Dhoparee, S., Epps, A., Tijs, T., & Walton, A. (2008). Measuring and defining the experience of immersion in games. *International journal of human-computer studies*, 66(9), 641-661.
- Kato, H., & Billinghurst, M. (1999). *Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System*. Paper presented at the 2nd IEEE and ACM International Workshop on Augmented Reality.
- Kaufmann, H., & Dünser, A. (2007). *Summary of usability evaluations of an educational augmented reality application*. Springer.
- Klein, G., & Murray, D. (2007). *Parallel Tracking and Mapping for Small AR Workspaces*. Paper presented at the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality.
- Kretschmer, U., Coors, V., Spierling, U., Grasbon, D., Schneider, K., Rojas, I., & Malaka, R. (2001). *Meeting the Spirit of History*. Paper presented at the Proceedings of the 2001 conference on Virtual reality, archeology, and cultural heritage.
- Krevelen, D. W. F. v., & Poelman, R. (2010). A Survey of Augmented Reality Technologies, Applications and Limitations. *The International Journal of Virtual Reality*, 9(2), 1-20.
- Lepetit, V., & Fua, P. (2005). Monocular Model-Based 3D Tracking of Rigid Objects: A Survey. *Computer Graphics and Vision*, 1(1), 1-89.
- Lima, J. o. P., Simões, F., Figueiredo, L., & Kelner, J. (2010). Model Based Markerless 3D Tracking applied to Augmented Reality. *SBC Journal on 3D Interactive Systems*, 1.
- Londei, R., Esposito, M., Diotte, B., Weidert, S., Euler, E., Thaller, P., . . . Fallavollita, P. (2015). Intra-operative augmented reality in distal locking. *International journal of computer assisted radiology and surgery*, 10, 1395-1403.
- Lourakis, M. I., & Argyros, A. A. (2005). Efficient, causal camera tracking in unprepared environments. *Computer Vision and Image Understanding*, 99(2), 259-290.
- Machover, C., & Tice, S. E. (1994). Virtual reality. *Computer Graphics and Applications, IEEE*, 14(1), 15-16.
- Magalhães, L. G., Sousa, J. J., Bento, R., Adão, T., Pereira, F., Filipe, V., & Peres, E. (2014). Proposal of an Information System for an Adaptive Mixed Reality System for Archaeological Sites. *Procedia Technology*, 16, 499-507.
- Marchand, É., Bouthemy, P., & Chaumette, F. (2001). A 2D-3D model-based approach to real-time visual tracking. *Image and Vision Computing*, 19, 941-955.

- Martín-Gutiérrez, J., Fabiani, P., Benesova, W., Meneses, M. D., & Mora, C. E. (2015). Augmented reality to promote collaborative and autonomous learning in higher education. *Computers in Human Behavior*, 51, 752–761.
- Milgram, P., & Kishino, F. (1994). A Taxonomy of Mixed Reality Visual Displays. *IEICE Transactions on Information Systems*, E77(D).
- Miyashita, T., Meier, P., Tachikawa, T., Orlic, S., Eble, T., Scholz, V., . . . Lieberknecht, S. (2008). *An Augmented Reality Museum Guide*. Paper presented at the Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality.
- Mourkoussis, N., Liarokapis, F., Darcy, J., Pettersson, M., Petridis, P., Lister, P., & White, M. (2002). *Virtual and Augmented Reality Applied to Educational and Cultural Heritage Domains*. Paper presented at the proceedings of Business Applications of Virtual Reality.
- Narciso, D., Pádua, L., Adão, T., Peres, E., & Magalhães, L. (2015). *MixAR Mobile Prototype: Visualizing Virtually Reconstructed Ancient Structures In Situ*. Paper presented at the Conference on ENTERprise Information Systems.
- Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., . . . Fitzgibbon, A. (2011). *KinectFusion: Real-Time Dense Surface Mapping and Tracking*. Paper presented at the 10th IEEE International Symposium on Mixed and Augmented Reality.
- Olsson, T., Kärkkäinen, T., Lagerstam, E., & Ventä-Olkkonen, L. (2012). User evaluation of mobile augmented reality scenarios. *Journal of Ambient Intelligence and Smart Environments*, 4(1), 29-47.
- Olsson, T., & Salo, M. (2011). *Online user survey on current mobile augmented reality applications*. Paper presented at the Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on.
- Pádua, L., Narciso, D., Cunha, A., Peres, E., & Magalhães, L. (2015). *Cost-effective and lightweight mobile units for MixAR: a comparative trial among different setups*. Paper presented at the Conference on ENTERprise Information Systems.
- Papagiannakis, G., Ponder, M., Molet, T., Kshirsagar, S., Cordier, F., Magnenat-Thalmann, N., & Thalmann, D. (2002). *LIFEPLUS: Revival of life in ancient Pompeii*. Paper presented at the Proceedings of Virtual Systems and Multimedia.
- Paul, P., Fleig, O., & Jannin, P. (2005). Augmented virtuality based on stereoscopic reconstruction in multimodal image-guided neurosurgery: Methods and performance evaluation. *IEEE Transactions on Medical Imaging*, 24(11), 1500-1511.
- Pilet, J. (2008). *Augmented Reality for Non-Rigid Surfaces*. École Polytechnique Fédérale de Lausanne.
- Platonov, J., Heibel, H., Meier, P., & Grollmann, B. (2006). *A mobile markerless AR system for maintenance and repair*. Paper presented at the Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality.
- Pollefeys, M. (1999). *Self-calibration and metric 3D reconstruction from uncalibrated image sequences*. PhD thesis, ESAT-PSI, KU Leuven.
- Porzi, L., Ricci, E., Ciarfuglia, T. A., & Zanin, M. (2012). *Visual-inertial Tracking on Android for Augmented Reality Applications*. Paper presented at the Environmental Energy and Structural Monitoring Systems (EESMS).
- Poupyrev, I., Tan, D., Billingham, M., Kato, H., Regenbrecht, H., & Tetsutani, N. (2001). *Tiles: A mixed reality authoring interface*. Paper presented at the INTERACT 2001 Conference on Human Computer Interaction.

- Pressigout, M., & Marchand, É. (2006). *Hybrid tracking algorithms for planar and non-planar structures subject to illumination changes*. Paper presented at the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality.
- Raque, J., Goble, A., Jones, V. M., Waldman, L. E., & Sutton, E. (2015). The Relationship of Endoscopic Proficiency to Educational Expense for Virtual Reality Simulator Training Amongst Surgical Trainees. *The American Surgeon*, 81(7), 747-752.
- Regenbrecht, H., Lum, T., Kohler, P., Ott, C., Wagner, M., Wilke, W., & Mueller, E. (2004). Using augmented virtuality for remote collaboration. *Presence: Teleoperators and virtual environments*, 13(3), 338-354.
- Remondino, F., & El - Hakim, S. (2006). Image - based 3D Modelling: A Review. *The Photogrammetric Record*, 21(115), 269-291.
- Rolland, J. P., Davis, L., & Baillet, Y. (2001). A survey of tracking technology for virtual environments. *Fundamentals of wearable computers and augmented reality*, 1, 67-112.
- Rolland, J. P., Holloway, R. L., & Fuchs, H. (1995). *A comparison of optical and video see-through, head-mounted displays*. Paper presented at the Photonics for Industrial Applications.
- Roussou, M., & Drettakis, G. (2003). *Photorealism and non-photorealism in virtual heritage representation*. Paper presented at the First Eurographics Workshop on Graphics and Cultural Heritage (2003).
- Sacks, R., Whyte, J., Swissa, D., Raviv, G., Zhou, W., & Shapira, A. (2015). Safety by design: dialogues between designers and builders using virtual reality. *Construction Management and Economics*, 33(1), 55-72. doi: <http://dx.doi.org/10.1080/01446193.2015.1029504>
- Sanguino, T. M., Márquez, J. A., Carlson, T., & Millán, J. d. R. (2014). Improving Skills and Perception in Robot Navigation by an Augmented Virtuality Assistance System. *Journal of Intelligent & Robotic Systems*, 76(2), 255-266.
- Simsarian, K. T., & Akesson, K.-P. (1997). Windows on the world: An example of augmented virtuality.
- Slater, M., & Wilbur, S. (1997). A framework for immersive virtual environments (FIVE): Speculations on the role of presence in virtual environments. *Presence: Teleoperators and virtual environments*, 6(6), 603-616.
- Steuer, J., Biocca, F., & Levy, M. R. (1995). Defining virtual reality: Dimensions determining telepresence *Communication in the age of virtual reality* (pp. 33-56).
- Sutherland, I. E. (1965). The ultimate display. *Multimedia: From Wagner to virtual reality*.
- Sutherland, I. E. (1968). *A head-mounted three dimensional display*. Paper presented at the Proceedings of the December 9-11, 1968, fall joint computer conference, part I.
- Teichrieb, V., Lima, J. P. S. d. M., Apolinário, E. L., Farias, T. S. M. C. d., Bueno, M. A. S., Kelner, J., & Santos, I. H. (2007). A Survey of Online Monocular Markerless Augmented Reality. *International Journal of modeling and Simulation for the Petroleum Industry*, 1(1), 1-7.
- Ternier, S., Klemke, R., Kalz, M., Van Ulzen, P., & Specht, M. (2012). ARLearn: Augmented Reality Meets Augmented Virtuality. *J. UCS*, 18(15), 2143-2164.
- Tori, R., Kirner, C., & Siscoutto, R. A. (2006). *Fundamentos e tecnologia de realidade virtual e aumentada*. Editora SBC.
- Unity. (2015). Game engine, tools and multiplatform. Consultado em 20 de Agosto, 2015. Disponível em: <http://unity3d.com/>

- Vacchetti, L., Lepetit, V., & Fua, P. (2004). Combining Edge and Texture Information for Real-Time Accurate 3D Camera Tracking. *ISMAR '04 Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality*, 48-56.
- Van Krevelen, D. W. F., & Poelman, R. (2010). A Survey of Augmented Reality Technologies, Applications and Limitations. *The International Journal of Virtual Reality*, 9(2), 1-20.
- Vlahakis, V., Ioannidis, N., Karigiannis, J., Tsotros, M., Gounaris, M., Stricker, D., . . . Almeida, L. (2002). Archeoguide: An Augmented Reality Guide for Archaeological Sites. *Computer Graphics in Art History and Archaeology*, 22(5), 52-60.
- Wikipedia. (2015a). Photorealism. Consultado em 15 de Agosto, 2015. Disponível em: <https://en.wikipedia.org/wiki/Photorealism>
- Wikipedia. (2015b). Software Development Kit. Consultado em 14 de Agosto, 2015. Disponível em: https://en.wikipedia.org/wiki/Software_development_kit
- Witmer, B. G., & Singer, M. J. (1998). Measuring presence in virtual environments: A presence questionnaire. *Presence: Teleoperators and virtual environments*, 7(3), 225-240.
- Zhou, F., Duh, H. B.-L., & Billinghurst, M. (2008). *Trends in Augmented Reality Tracking, Interaction and Display: A Review of Ten Years of ISMAR*. Paper presented at the IEEE International Symposium on Mixed and Augmented Reality.
- Zhou, N.-N., & Deng, Y.-L. (2009). Virtual reality: A state-of-the-art survey. *International Journal of Automation and Computing*, 6(4), 319-325.
- Zöllner, M., Keil, J., Wüst, H., & Pletinckx, D. (2009). *An Augmented Reality Presentation System for Remote Cultural Heritage Sites*. Paper presented at the The 10th International Symposium on Virtual Reality, Archaeology and Cultural Heritage VAST.

Anexos

Anexo A – Questionário das experiências com utilizadores

Folha de consentimento

Âmbito: Projecto MixAR (BI/MIXAR/38803/UTAD/2014): um sistema de realidade mista para a reconstrução virtual de sítios de herança cultural.

Termos da realização da experiência

Esta experiência tem como propósito a recolha de opinião sobre o conforto, presença, imersividade e satisfação, no âmbito do projeto supramencionado. A experiência inclui a visualização de conteúdos de realidade aumentada.

Os participantes são voluntários e não recebem quaisquer benefícios pela realização da experiência. A qualquer momento podem desistir da experiência, não estando estes vinculados a nenhuma obrigação.

Toda a informação recolhida será anonimizada e utilizada para fins científicos.

A experiência pressupõe a realização de um conjunto de tarefas (procedimento). No decorrer desta serão dadas instruções para a sua realização, sendo que poderão variar de teste para teste.

Esta experiência foi previamente testada, pelo que não se prevê que da sua realização resulte qualquer tipo de lesão para os participantes.

O investigador está à disposição para esclarecer quaisquer dúvidas suplementares sobre a experiência, antes da realização da mesma.

Declaração

Declaro que li e concordo com os termos realização da experiência. Reservando-me ao direito de abandonar a experiência em qualquer fase da mesma. Dando o meu consentimento a que os dados recolhidos durante a realização da experiência sejam utilizados para fins científicos.

Assinatura: _____

Data: ____ / ____ / ____

Questionário Genérico

Idade: _____ **Género:** Masculino ☐ Feminino ☐

Usa óculos ou lentes de contato? Sim ☐ Não ☐

Profissão: _____

Habilitações académicas

Ensino Básico ☐ Ensino Secundário ☐ Ensino Superior ☐

Qual a sua área de estudos / formação?

Como classifica a sua experiência com computadores

Nenhuma ☐ Básica ☐ Intermédia ☐ Boa ☐ Avançada ☐

Possui um bom nível de experiência na utilização de smartphones?

Sim ☐ Não ☐

Em que sistema(s) operativo(s)? Android ☐ iOS ☐ Windows Phone ☐

Possui um bom nível de experiência na utilização de tablets?

Sim ☐ Não ☐

Em que sistema(s) operativo(s)? Android ☐ iOS ☐ Windows ☐

Já teve contacto com alguma experiência de realidade aumentada?

Sim ☐ Não ☐

Já teve contacto com alguma experiência de realidade virtual?

Sim ☐ Não ☐

Dispositivo da experiência: _____

ID: _____

Por favor, assinale para cada um dos grupos de questões, a resposta que mais se adequa à experiência que realizou, entre o “Discordo Totalmente” e o “Concordo Totalmente”.

Questionário de satisfação

	Questão	Discordo Totalmente	Discordo	Neutro	Concordo	Concordo Totalmente
1.	Repetiria a experiência novamente	•	•	•	•	•
2.	Gostei dos conteúdos que me foram apresentados	•	•	•	•	•
3.	Foi difícil visualizar os edifícios durante a experiência	•	•	•	•	•
4.	O(s) dispositivo(s) da experiência era(m) pesado(s) e difícil(eis) de transportar	•	•	•	•	•
5.	No geral, fiquei agradado com a qualidade dos modelos virtuais exibidos	•	•	•	•	•
6.	Os conteúdos visualizados estavam sincronizados com o mundo real	•	•	•	•	•
7.	Os modelos eram apresentados de uma forma robusta (sem saltos) e fluída	•	•	•	•	•
8.	Tive a sensação de estar a visitar o passado	•	•	•	•	•
9.	Utilizaria este sistema em espaços arqueológicos para visualização de reconstrução de ruínas	•	•	•	•	•
10.	Gostaria que os museus e espaços arqueológicos disponibilizassem sistemas deste género para visitas interativas	•	•	•	•	•
11.	No geral, gostei da experiência que acabei de realizar	•	•	•	•	•

Questionário de presença

	Questão	Discordo Totalmente	Discordo	Neutro	Concordo	Concordo Totalmente
1.	Senti-me envolvido pela experiência	•	•	•	•	•
2.	Estive consciente do que se passava no mundo real (fora da experiência)	•	•	•	•	•
3.	Consegui abstrair-me dos dispositivos que estava a usar na experiência	•	•	•	•	•
4.	Fui capaz de aproximar-me dos edifícios para os observar	•	•	•	•	•
5.	Consegui visualizar os edifícios virtuais de pontos de vista diferentes	•	•	•	•	•
6.	Senti-me confuso ou desorientado durante a experiência	•	•	•	•	•
7.	Senti que havia um atraso entre as minhas ações e o conteúdo apresentado	•	•	•	•	•
8.	Adaptei-me com facilidade à experiência	•	•	•	•	•
9.	Senti facilidade a deslocar-me no ambiente	•	•	•	•	•
10.	Senti que a qualidade visual penalizou a execução das tarefas na experiência	•	•	•	•	•
11.	Consegui concentrar-me mais nas tarefas do que nos dispositivos	•	•	•	•	•
12.	Estive tão envolvido que perdi a noção do tempo	•	•	•	•	•

Questionário de imersividade

	Questão	Discordo Totalmente	Discordo	Neutro	Concordo	Concordo Totalmente
1.	Estive interessado em saber o que iria acontecer ao longo da experiência	•	•	•	•	•
2.	Fiquei intrigado com o que poderia acontecer em cada tarefa da experiência	•	•	•	•	•
3.	A qualidade dos conteúdos apresentados pareceu-me boa	•	•	•	•	•
4.	Senti que conseguia mover-me à vontade no ambiente	•	•	•	•	•
5.	Senti que os conteúdos virtuais apresentados faziam parte do mundo real	•	•	•	•	•
6.	Enquanto durou, a experiência foi a minha única preocupação	•	•	•	•	•
7.	Pareceu-me que passou pouco tempo desde o início ao fim da experiência	•	•	•	•	•
8.	A experiência prendeu a minha atenção	•	•	•	•	•
9.	Esqueci-me das minhas preocupações quotidianas durante a experiência	•	•	•	•	•
10.	Senti que estive a experienciar uma situação nova, mais do que apenas realizar tarefas	•	•	•	•	•
11.	Senti-me motivado com o decorrer da experiência	•	•	•	•	•
12.	Senti que realizei a experiência com sucesso	•	•	•	•	•
13.	Senti-me dececionado com o final da experiência	•	•	•	•	•

Questionário de conforto

Assinale as zonas corporais em que sentiu cansaço ou exaustão:

Cansaço generalizado: ☐

Pernas: ☐

Ombros: ☐

Braços: ☐

Costas: ☐

Pernas: ☐

Dorso nasal: ☐

Outro: _____

Assinale os tipos de indisposição que sentiu:

Tonturas ☐

Desorientação ☐

Vertigens ☐

Sensação de desmaio ☐

Enjoos ☐

Dores de cabeça ☐

Olhos cansados ☐

Visão turva ☐

Problemas em focar a visão ☐

Problemas de concentração ☐

Obrigado pela sua colaboração ☺