

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Sistema Híbrido Baseado em Geometria e Vídeo 360° 3D para Realidade Virtual em Dispositivos Móveis

Leonel Jorge Nogueira Peixoto



Mestrado Integrado em Engenharia Informática e Computação

Orientador: Rui Pedro Amaral Rodrigues

Co-orientador: Hugo Miguel Gonçalves Crespo Machado da Silva

Junho de 2017

Sistema Híbrido Baseado em Geometria e Vídeo 360° 3D para Realidade Virtual em Dispositivos Móveis

Leonel Jorge Nogueira Peixoto

Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo Júri:

Presidente: António Fernando Vasconcelos Cunha Castro Coelho

Vogal Externo: Paulo Miguel de Jesus Dias

Orientador: Rui Pedro Amaral Rodrigues

Co-orientador: Hugo Miguel Gonçalves Crespo Machado da Silva

24 de Julho de 2017

Resumo

A realidade virtual tem vindo a ganhar popularidade na sociedade devido ao facto de facultar uma experiência imersiva com menor custo e melhor usabilidade do que teve no passado. Porém, verifica-se que o desenvolvimento dos dispositivos utilizados para difundir essa tecnologia tende a divergir em dois caminhos diferentes. Por um lado, existem dispositivos que usam o poder computacional de computadores fixos permitindo uma experiência com alta qualidade mas que limitam a capacidade de movimento do utilizador pelo facto de usarem cabos. Por outro lado, assiste-se ao crescimento de dispositivos móveis que possibilitam uma maior liberdade de movimento mas que limitam a experiência pelo facto de possuírem menor capacidade de processamento.

Para possibilitar a realidade virtual em dispositivos móveis com alta qualidade e fidelidade visual sem quebras na taxa de refrescamento e sem prejuízo na experiência imersiva, é necessário simplificar os ambientes virtuais visualizados, reduzindo assim os requisitos computacionais.

Um possível compromisso passa pela combinação de objetos com geometria tridimensional sobrepostos à uma imagem ou vídeo 360° que cobre a área de visualização do utilizador. Essa imagem ou vídeo 360° gerados previamente num sistema com elevado poder de processamento, apresentam assim os detalhes necessários para oferecer uma boa qualidade visual. A imagem ou vídeo 360° não permitem a interação que normalmente é necessária em ambientes de realidade virtual, mas a sobreposição de geometria em tempo real pode permitir a interatividade com determinados objetos do cenário.

Porém, esta solução acarreta problemas de coerência entre os objetos 3D e o fundo pré-gerado como a manutenção de colisões, oclusões, iluminação, sombras e transparências juntamente com problemas relacionados com o fluxo de dados e compressão do vídeo de fundo. Por isso, deve ser preservado um conjunto de meta-dados bem definido provenientes da cena que originou o vídeo de fundo, e estes devem ser utilizados no dispositivo final com técnicas apropriadas para resolver os problemas de coerência.

Neste trabalho propõe-se uma tecnologia destinada a criar um sistema híbrido baseado em objetos 3D sobrepostos à imagem ou vídeo 360° para proporcionar ambientes de realidade virtual de alta qualidade em dispositivos com baixa capacidade de processamento.

A solução proposta foi devidamente testada e validada através de medidas quantitativas e qualitativas.

Espera-se que esta abordagem permita uma maior qualidade das experiências baseadas em dispositivos móveis, o que por sua vez, terá um impacto no custo dos sistemas e na progressão da nova tecnologia numa maior parcela da sociedade.

Abstract

Nowadays virtual reality is becoming popular in society because it provides an immersive experience at lower costs and with better usability than it had in the past. However, it turns out that the development of the devices used to spread this technology tends to diverge in two different ways. On the one hand, there are devices that use the computing power of fixed computers, allowing a high-quality experience but that limit the user's ability to move due to the necessary cables. On the other hand, there is the growth of mobile devices that allow greater freedom of movement but limit the experience due to the fact that they have less processing capacity.

To promote virtual reality in mobile devices with high quality and visual fidelity without significant frame rate losses and without prejudice in the immersive experience, it is necessary to simplify the virtual environments visualized, thus reducing the computational requirements.

A possible compromise is the combination of objects with three-dimensional geometry superimposed on a 360° image or video that covers the user's viewing area. This 360° image or video is previously generated on a high-processing power system, thus containing the details needed to provide good visual quality. The generated 360° image or video cannot provide the interaction that is typically needed in virtual reality environments, but overlapping real-time geometry allows for interactivity with specific objects of the scenario.

However, this solution may cause coherence problems between 3D objects and the pre-generated background, such as collisions, occlusions, lighting, shadows, and transparency maintenance along with problems with data streaming and background video compression. Therefore, a set of well-defined metadata must be preserved from the scene that originated the background video, and used in the final device with appropriate techniques to tackle the problems of coherence.

In this work we propose a technology aimed at creating a hybrid system based on 3D objects superimposed on the 360° image or video to provide virtual reality environments in devices with low processing capacity.

The proposed solution was duly tested and validated through quantitative and qualitative measures.

It is expected that such an approach will enable higher quality on mobile-based experiences, which in turn will have an impact on the cost of systems and the progression of the new technology in a larger part of society.

Agradecimentos

Ao Professor Rui Pedro Amaral Rodrigues, pelo seu apoio e orientação imprescindível na determinação dos objetivos prioritários deste trabalho bem como a sua colaboração na minha integração no grupo *Graphics, Interaction and Gaming* (GIG) da FEUP.

Ao Hugo Miguel Gonçalves Crespo Machado da Silva, por ter proposto este tema de trabalho que considerei muito motivante, pelos seus conselhos, orientação e confiança depositada em mim na elaboração deste trabalho.

À toda a equipa da Darwin & Warhol, Nuno, Amílcar, Pedro, Jorge e Sérgio, pela formidável simpatia com que me acolheram, pelo espaço e meios de trabalho disponibilizados.

Aos elementos do GIG pela ajuda, ideias partilhadas, conselhos e tempo disponibilizado nas reuniões de orientação de mestrandos que estiveram a concluir a tese.

Às pessoas que participaram nos testes de avaliação da solução desenvolvida neste trabalho, pelo seu tempo disponibilizado de forma completamente voluntária.

Aos meus colegas do MIEIC com os quais eu convivi e colaborei em trabalhos académicos, pela amizade, esforço e dedicação partilhada em momentos de convívio e trabalho mútuos.

Aos meus queridos pais, pelo apoio incondicional, afeto, carinho e paciência que tiveram comigo durante as horas incontáveis que passei em frente ao computador durante estes 5 anos do MIEIC.

Leonel Jorge Nogueira Peixoto

Conteúdo

Introdução.....	1
1.1 Contexto.....	1
1.2 Motivação e Objectivos	3
1.3 Estrutura da Dissertação	4
Técnicas de renderização com utilização de fundos pré-gerados	5
2.1 Técnicas de visualização de fundos 360°.....	6
2.1.1 Projeção cúbica.....	7
2.1.2 Projeção esférica.....	8
2.2 Coerência entre textura de fundo e objetos 3D.....	9
2.2.1 Oclusões e Transparências.....	10
2.2.2 Transparências.....	11
2.2.3 Iluminação	12
2.2.4 Sombras	13
2.2.5 Colisões	14
2.2.6 Artefactos de codificação da textura de fundo	16
2.3 Requisitos da Realidade Virtual para dispositivos Móveis	17
2.4 Trabalhos Relacionados.....	18
2.5 Resumo e Conclusões	20
Arquitetura e Implementação	21
3.1 Arquitetura do sistema.....	21
3.1.1 Sistema de pré-renderização	22
3.1.2 Sistema de visualização híbrido com fundo 360 e objetos 3D	24
3.2 Implementação da coerência entre a textura de fundo e objetos 3D interativos.....	26
3.2.1 Oclusões e Transparências.....	26
3.2.2 Colisões	28
3.2.3 Iluminação e Sombras	29
3.3 Geração dos meta-dados	30
3.3.1 Pré-renderização e geração do vídeo de 360 por 180 graus	30
3.3.2 Geometrias.....	34

3.3.3	Animações	35
3.3.4	Fontes de luz.....	36
3.4	Melhorias para aumento de desempenho.....	37
3.4.1	Simplificação da geometria	37
3.4.2	Sistema de <i>occlusion culling</i>	38
Testes e Resultados.....		39
4.1	Testes de desempenho	39
4.1.1	Descrição do protótipo	39
4.1.2	Descrição dos testes.....	41
4.1.3	Resultados.....	43
4.1.4	Análise dos resultados	46
4.2	Testes qualitativos	47
4.2.1	Descrição dos protótipos	47
4.2.1.1	Protótipo A – cenário simples	48
4.2.1.2	Protótipo B - Cenário complexo.....	49
4.2.1.3	Protótipo C - Cenário simples com interatividade	50
4.2.2	Descrição dos testes.....	51
4.2.3	Resultados.....	52
4.2.4	Análise dos resultados	54
Conclusões e Trabalho Futuro.....		56
5.1	Trabalho Futuro	57
Referências.....		58
Anexo A: Resultados de desempenho.....		65
7.1	Resultados de desempenho: Imagens/s com Vídeo: 1750×874 @ 30 imagens/s - Imagem fixa: 2048×2048.....	65
7.2	Resultados de desempenho: Imagens/s com Vídeo: 4096×2048 @ 60 imagens/s - Imagem fixa: 4096×4096.....	67
Anexo B: Declaração de consentimento informado apresentado aos participantes voluntários		68
8.1	Declaração de consentimento informado apresentado aos participantes.....	68
Anexo C: Questionário apresentado aos participantes voluntários.....		70
9.1	Questionário apresentado aos participantes voluntários.....	70
Anexo D: Resultados Qualitativos		84
10.1	Resposta dos participantes nas questões notadas de 0 a 5	84
10.2	Resposta dos participantes nas questões abertas.....	86

Lista de Figuras

Figura 5: Projeção cúbica.	7
Figura 6: Projeção esférica.	8
Figura 1: Exemplo de incoerências entre a textura de fundo e os objetos 3D em tempo real.	9
Figura 2: Cena pré-renderizada (à esquerda), oclusão do objeto 3D em tempo real inexistente (ao centro), objeto 3D em tempo real corretamente ocultado (à direita).	10
Figura 3: Cena pré-renderizada (à esquerda), iluminação incorreta com a adição de um objeto 3D em tempo real (ao centro), iluminação correta (à direita).	12
Figura 4: Projeção das sombras inexistentes (à esquerda), projeção correta das sombras (à direita).	14
Figura 7: Arquitetura geral do sistema	22
Figura 8: Estrutura do sistema de visualização híbrido.	24
Figura 9: Exemplo de uma imagem estereoscópica equiretangular gerada pelo sistema de pré-renderização. A parte superior corresponde ao olho esquerdo e a inferior ao olho direito.	31
Figura 10: As regiões coloridas permitem destacar o desalinhamento que ocorre entre os <i>proxies</i> e a textura de fundo na região polar superior (em cima) e na região equatorial (em baixo).	33
Figura 11: Distorção da textura de fundo existente no polo superior das esferas (à esquerda), em comparação com o original (à direita).	34
Figura 12: Exemplo de um ponto de vista do cenário usado para os testes quantitativos.	40
Figura 13: Cena original totalmente tridimensional (à esquerda), imagem 360° pré-renderizada (ao centro) e o sistema híbrido com um objeto 3D interativo em tempo real (à direita).	41
Figura 14: Número de imagens por segundo <i>versus</i> método de renderização para cada dispositivo. Vídeos 360° de 1750×874 <i>pixels</i> a 30 imagens/s e imagens fixas de 2048×2048.	45

Figura 15: Número de imagens por segundo <i>versus</i> método de renderização para cada dispositivo. Vídeo 360° de 4096×2048 <i>pixels</i> a 60 imagens/s e imagens fixas de 4096×4096.	45
Figura 16: Ponto de vista do cenário simples com trajetória do objeto interativo.	48
Figura 17: Ponto de vista do cenário complexo com trajetória do objeto interativo.	49
Figura 18: Ponto de vista do cenário simples com interação.	50

Lista de Tabelas

Tabela 1: Características técnicas dos dispositivos utilizados para testes de desempenho.	42
Tabela 2: Resultados de desempenho do sistema híbrido com vídeos 360° de 1750×874 <i>pixels</i> a 30 imagens/s e imagens fixas de 2048×2048 <i>versus</i> método totalmente 3D.	43
Tabela 3: Resultados de desempenho do sistema híbrido com vídeo 360° de 4096×2048 <i>pixels</i> a 60 imagens/s e imagens fixas de 4096×4096 <i>versus</i> método totalmente 3D.	44
Tabela 4: Denominação das várias versões dos protótipos usadas para os testes qualitativos.	47
Tabela 5: Resultados obtidos para as oclusões	52
Tabela 6: Resultados obtidos para as colisões	52
Tabela 7: Resultados obtidos para a qualidade visual	52
Tabela 8: Teste de Friedman para as oclusões	53
Tabela 9: Teste de Friedman para as colisões	54
Tabela 10: Teste de Friedman para a qualidade visual	54

Abreviaturas e Símbolos

2D	Bidimensional
3D	Tridimensional
HDR	<i>High Dynamic Range</i>
HMD	<i>Head-Mounted Display</i>
IBL	<i>Image-Based Lighting</i>
LOD	<i>Level Of Details</i>
XML	<i>eXtensible Markup Language</i>

Capítulo 1

Introdução

A realidade virtual é uma tecnologia cuja utilização despertou múltiplos interesses desde o seu aparecimento. Esta é regularmente usada para fins educativos e formativos, mas recentemente o aparecimento no mercado de múltiplos dispositivos de realidade virtual disponíveis para o consumidor comum tem vindo a alargar consideravelmente o uso desta tecnologia num maior segmento da sociedade.

1.1 Contexto

Para promover a sensação de imersão num ambiente de realidade virtual devem ser estimulados vários sentidos que o ser humano usa para perceber o seu meio ambiente, isto é, a visão, audição, tacto, olfacto e paladar. Tipicamente, os fabricantes de dispositivos imersivos em ambientes de realidade virtual, focam-se nos canais sensoriais da visão e audição por serem os mais fáceis de estimular através de um *Head-Mounted Display* (HMD) que combina um ou dois ecrãs e auscultadores sonoros. Estes dois sentidos são as principais fontes de informação utilizada pelo nosso cérebro para entender o ambiente. Mais de 80% da informação vem pela visão, e mais de 10% pela audição. A informação recebida pelos outros sentidos contribui percentualmente pouco.

Atualmente existe uma escolha variada de HMDs disponíveis no mercado. Contudo, estes podem ser classificados em dois grupos bem distintos.

O primeiro grupo de HMDs, para o qual o HTC Vive [1] pode ser utilizado como exemplo, possui dois ecrãs juntamente com um conjunto de lentes e sensores de orientação e posicionamento. A geração do ambiente virtual em três dimensões (3D) é efetuada num computador ligado fisicamente ao HMD através de cabos para poder enviar as imagens geradas ao ecrã do HMD. Ao ser usado um computador fixo com elevado poder computacional,

Introdução

juntamente com uma placa gráfica de alto desempenho, pode ser conseguida uma grande qualidade visual do ambiente virtual com taxa de refrescamento suficiente para promover ao utilizador uma boa sensação de imersão. Porém, ao serem usados cabos de ligação entre o HMD e o computador fixo, a capacidade de movimentação do utilizador encontra-se limitada e pode ser um ponto negativo em aplicações que exigem liberdade de movimento.

O segundo grupo de HMDs, em que o Samsung Gear VR [2] é um exemplo, à semelhança do primeiro grupo, possui todos os componentes de visualização e orientação. A grande diferença reside no facto que as unidades de processamento central e gráfico se encontram embutidos diretamente no HMD. A maioria dos dispositivos deste grupo tira partido do uso de um *smartphone* aproveitando todas as características de visualização e computação que este possui. Dentro deste grupo existem também HMDs que integram completamente de forma transparente o *hardware* necessário sem necessidade de qualquer dispositivo suplementar. Este grupo de HMDs apresenta a vantagem de oferecer uma grande capacidade de movimento ao utilizador pelo facto de utilizar dispositivos móveis libertando-se assim da necessidade de cabos. Contudo, o relativamente baixo desempenho computacional dos dispositivos móveis usados por este grupo de HMDs acarreta grandes limitações para a qualidade visual do ambiente virtual resultando numa experiência menos realista e imersiva.

Existe uma necessidade real de adaptar a concepção de ambientes de realidade virtual aos dispositivos móveis pois estes encontram-se limitados pelas suas capacidades de processamento em tempo real. Para promover uma boa sensação de imersão aos utilizadores destes tipos de dispositivos, a complexidade do ambiente virtual deve ser reduzida para atingir as capacidades de processamento disponíveis.

A redução da complexidade pode ser alcançada convertendo parte da cena 3D para um vídeo 360. Este pode representar a mesma cena, mas ter um custo de visualização final reduzido permitindo preservar toda a complexidade visual do ambiente virtual sem a necessidade de calcular em tempo real toda a geometria que à originou. Consequentemente, esta técnica permite a redução do número de polígonos apresentados e os cálculos necessários dentro do *camera frustum* possibilitando um melhor desempenho por parte do dispositivo móvel.

Numa primeira fase *offline*, é efectuada uma pré-renderização do ambiente virtual para obter um vídeo panorâmico sob a forma de uma textura em duas dimensões (2D). Ao ser *offline*, este passo pode ser efetuado numa plataforma diferente do dispositivo móvel final, como por exemplo num computador fixo, aproveitando toda a sua capacidade de processamento gráfico para obter uma qualidade visual elevada. Numa segunda etapa, no dispositivo móvel final, objetos 3D são sobrepostos à textura obtida no passo anterior permitindo ao utilizador interagir com estes em tempo real. Ao ser usada a combinação da textura do ambiente de fundo juntamente com objetos 3D interativos em tempo real, é possível promover ao utilizador a sensação de que este se encontra imerso no ambiente virtual original mantendo um certo grau de interação.

Introdução

Todavia, esta técnica revela-se também problemática em algumas situações. Ao ser gerada a textura de fundo, perde-se toda a informação de geometria, profundidade e iluminação associada aos objetos da cena virtual original. Esta falta de informação reflete-se em possíveis incoerências entre os objetos da cena original representada pela textura de fundo e os objetos 3D interativos em tempo real. Estas incoerências manifestam-se de várias formas:

- Oclusões: objetos da cena de fundo são sempre ocultados por objetos 3D interativos em tempo real.
- Colisões: objetos 3D interativos em tempo real não têm a possibilidade de colidir com objetos da cena de fundo e vice-versa.
- Iluminação: objetos 3D interativos em tempo real não têm a possibilidade de irradiar a sua iluminação incidente sobre os objetos da cena de fundo e vice-versa.
- Sombras: objetos 3D interativos em tempo real não têm a possibilidade de projetar sombras sobre objetos da cena de fundo e vice-versa.
- Transparências: objetos transparentes da cena de fundo não têm a possibilidade de deixar transparecer objetos 3D interativos em tempo real.
- Artefactos de compressão da textura de fundo permitem diferenciar os objetos da cena de fundo dos objetos 3D interativos em tempo real degradando a sensação de imersão do utilizador.

Uma outra limitação importante que esta técnica possui é a redução do número de graus de liberdade do utilizador no ambiente virtual e isso advém do facto de a pré-renderização ser efetuada num ponto de visualização fixo. Assim só são possíveis movimentos de rotação em torno de três eixos não sendo permitido movimentos de translação no mundo virtual.

1.2 Motivação e Objectivos

O aumento crescente do mercado dos dispositivos móveis tem facilitado a difusão desses dispositivos na sociedade. Por outro lado, o aparecimento de HMDs que utilizam dispositivos móveis está a possibilitar a utilização da realidade virtual de uma forma mais cómoda e transparente sem a utilização de computadores de alto desempenho e custo elevado. Contudo o baixo desempenho dos dispositivos móveis não permite visualizar ambientes virtuais complexos com qualidade e fluidez usando os métodos tradicionais de renderização.

Este trabalho tem como principal objetivo desenvolver uma solução que permita visualizar cenários de realidade virtual complexos e com qualidade visual elevada sem quebras do desempenho nos dispositivos móveis. Para tal, será utilizada a técnica descrita na secção 1.1 que utiliza vídeo 360° pré-renderizado juntamente com objetos 3D interativos em tempo real.

As questões de investigação deste estudo incidem sobre os pontos seguintes:

- Verificar se a solução híbrida desenvolvida consegue manter um bom compromisso entre o seu desempenho e a qualidade visual fornecida.

Introdução

- Descobrir metodologias adequadas para resolver as questões de oclusão e colisão associadas ao sistema híbrido.

1.3 Estrutura da Dissertação

Para além da introdução, este relatório contém mais 4 capítulos. No capítulo 2, é exposta uma revisão bibliográfica dos métodos usados para resolução dos problemas descritos e são apresentados trabalhos relacionados. No capítulo 3 apresenta-se a arquitetura do sistema desenvolvido bem como as técnicas e métodos adotados para solucionar os problemas presentes neste trabalho. No capítulo 4 são expostos os protótipos que foram desenvolvidos e os testes efetuados para obter resultados sobre a validade dos métodos usados para resolver os problemas de investigação deste trabalho. Estes resultados são igualmente analisados neste mesmo capítulo. Por último, no capítulo 5 são apresentadas as principais conclusões, bem com perspectivas de futuros trabalhos que podem trazer melhorias às soluções aqui propostas.

Capítulo 2

Técnicas de renderização com utilização de fundos pré-gerados

Neste capítulo é descrito o estado da arte na resolução dos problemas descritos no capítulo anterior, nomeadamente técnicas usadas para resolver as incoerências entre a textura de fundo e os objetos 3D interativos em tempo real no contexto proposto. Também serão abordadas estratégias de implementação, como por exemplo, o tipo de projeção usada para a textura de fundo. São igualmente apresentados vários trabalhos dos quais alguns não abordam diretamente o tema deste trabalho mas estão relacionados com ele e outros trabalhos que abordam diretamente o tema mas de forma diferente.

Muitas das técnicas referidas a seguir são baseadas em investigações efectuadas na área da realidade virtual. Também são referidas técnicas usadas nas áreas da realidade aumentada e realidade mista pois, nessas áreas, são tipicamente sobrepostos objetos 3D com interação em tempo real a uma imagem de fundo que representa o mundo real capturada por uma câmara e depois apresentados num monitor (*video see-through*) ou visível através de óculos transparentes (*optical see-through*). Pode-se então estabelecer uma semelhança em termos de desafios entre a sobreposição de objetos 3D sobre uma textura de fundo originada pela renderização de um ambiente virtual e sobre uma imagem de fundo obtida por uma câmara a capturar o mundo real. Os autores das investigações nessas áreas deparam-se com os mesmos problemas de incoerência entre a imagem ou vídeo de fundo e os objetos 3D interativos em tempo real.

2.1 Técnicas de visualização de fundos 360°

Um dos requisitos definidos nos objetivos deste trabalho passa por gerar a textura de fundo que cobre completamente a área de visão do utilizador. Isso implica que o ponto de visão do utilizador deve ser colocado no centro de um objeto sobre o qual é projetada a textura panorâmica. Para exemplificar, num planetário, é projetada a imagem da abóbada celeste sobre a superfície interna de uma semiesfera [32, 33].

A projeção mais comum e regularmente usada é a projeção em perspectiva que consiste em projetar parte de um ambiente virtual sobre um plano [34]. Tipicamente, é gerada uma textura em 2D de uma cena tridimensional e esta textura é depois mapeada sobre um plano com dimensões proporcionais. Este método foi regularmente utilizado em videojogos como *The 7th Guest* [35] e *Resident Evil* [36]. Porém, esta projeção não é adequada para visualizações panorâmicas em 360° [34].

Segundo J. Brosz e F. Samavati [37], uma projeção panorâmica é definida pela superfície geométrica utilizada para derivar a equação de projeção. Logo, o tipo de superfície usado vai influenciar a equação de projeção e consequentemente as propriedades da imagem projetada. Como tal deve ser escolhida a melhor projeção possível sem correr o risco de deformar demasiado a textura projetada.

Existem vários tipos de projeções, das quais se destacam a cúbica, cilíndrica, retangular equidistante e cónica [34, 37]. A projeção cilíndrica introduz áreas não-visíveis nas zonas superiores e inferiores de visualização. As projeções cónicas e semi-esféricas introduzem áreas não visíveis na zona inferior de visualização. Como tal só serão abordadas as projeções cúbicas e esféricas.

Técnicas de renderização com utilização de fundos pré-gerados

2.1.1 Projeção cúbica

A projeção cúbica consiste no mapeamento de seis imagens sobre as seis faces internas de um cubo. Estas imagens são normalmente obtidas através de uma câmara virtual com um campo de visão de $90^\circ \times 90^\circ$, colocada no centro do cubo, e sofrendo rotações de 0° , 90° , 180° e 270° no eixo y para obter as quatro faces laterais e sofrendo igualmente rotações de 90° e -90° no eixo x para obter as faces inferior e superior do cubo [37]. Na figura 5 é apresentado um exemplo de projeção cúbica.

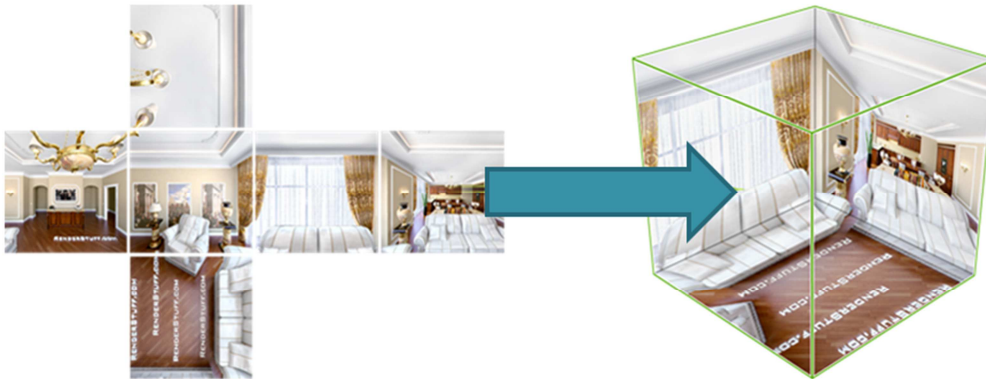


Figura 1: Projeção cúbica [38].

Esta projeção é usada por C. Cruz-Neira, D. J. Sandin, T. A. DeFanti, R. V. Kenyon, e J. C. Hart [39] na tecnologia intitulada “*The CAVE*” para visualizar um ambiente virtual com a ajuda de projetores comuns dentro de uma sala cúbica onde se encontra o utilizador. Três a seis projetores projetam uma imagem sobre as superfícies de um cubo com as dimensões adequadas para imergir os utilizadores no ambiente projetado. Apesar do tipo de aplicação ser diferente do presente trabalho, é bem ilustrativa do uso deste tipo de projeção.

Esta técnica de mapeamento é facilmente suportada por unidades de processamento gráfico modernas, pois estas estão optimizadas para efetuar uma técnica de iluminação conhecida por *environment mapping*, que usa a projeção cúbica como algoritmo base [34]. No entanto, por ser uma técnica baseada em seis projeções planares, esta induz distorções da imagem, nomeadamente nos pontos ao redor da zona central de visualização [37].

Técnicas de renderização com utilização de fundos pré-gerados

2.1.2 Projeção esférica

Esta projeção é igualmente referida como projeção equiretangular ou projeção cilíndrica equidistante. É obtida através do mapeamento de uma imagem sobre a superfície interna de uma esfera. Na figura 6 é apresentado um exemplo de projeção esférica.



Figura 2: Projeção esférica [38].

Segundo J. Ardouin, A. Lécuyer, M. Marchal, e E. Marchand [34], é a projeção mais usada para visualização de fotografias e vídeos panorâmicos, por permitir uma completa representação das áreas ao redor da zona central de visualização. Possui a propriedade de equidistância, ou seja, todos os pontos presentes na superfície de projeção encontram-se à mesma distância do ponto de visualização (considerando que o ponto de visualização se encontra no centro da esfera). Esta propriedade não existe na projeção cúbica. Como tal, a projeção esférica não sofre tanta distorção como a projeção cúbica. Segundo o estudo feito pelos mesmos autores, a projeção esférica é aquela que os utilizadores preferem para navegar em ambientes de realidade virtual [34].

Neste trabalho optou-se então por usar este tipo de projeção definindo o centro da esfera como sendo o ponto de visualização. Contudo, como já foi mencionado, e visto que o ponto de visualização é fixo, não é possível efetuar movimentos de translação no ambiente virtual sendo apenas permitido movimento de rotação em torno dos três eixos cartesianos.

2.2 Coerência entre textura de fundo e objetos 3D

Como já foi referido no capítulo 1, a rasterização de um ambiente virtual deve originar uma imagem ou vídeo que será usado como textura de fundo. Porém este processo tem como consequência a perda de informações relativas às propriedades dos objetos como, por exemplo, malhas geométricas tridimensionais, malhas de colisão e materiais de reflexão e transparência. São igualmente perdidas informações associadas às fontes de luz da cena original necessárias para os cálculos de iluminação. As informações globais da cena original, como a distância relativa entre os objetos e a iluminação global, são igualmente perdidas. Como é possível ver na figura 1, esta falta de informação reflete-se principalmente em incoerências notáveis entre a textura de fundo, que representa o ambiente, e os objetos 3D sobrepostos em tempo real. Os artefactos de compressão da textura de fundo induzem igualmente incoerências perante os objetos 3D sobrepostos em tempo real.

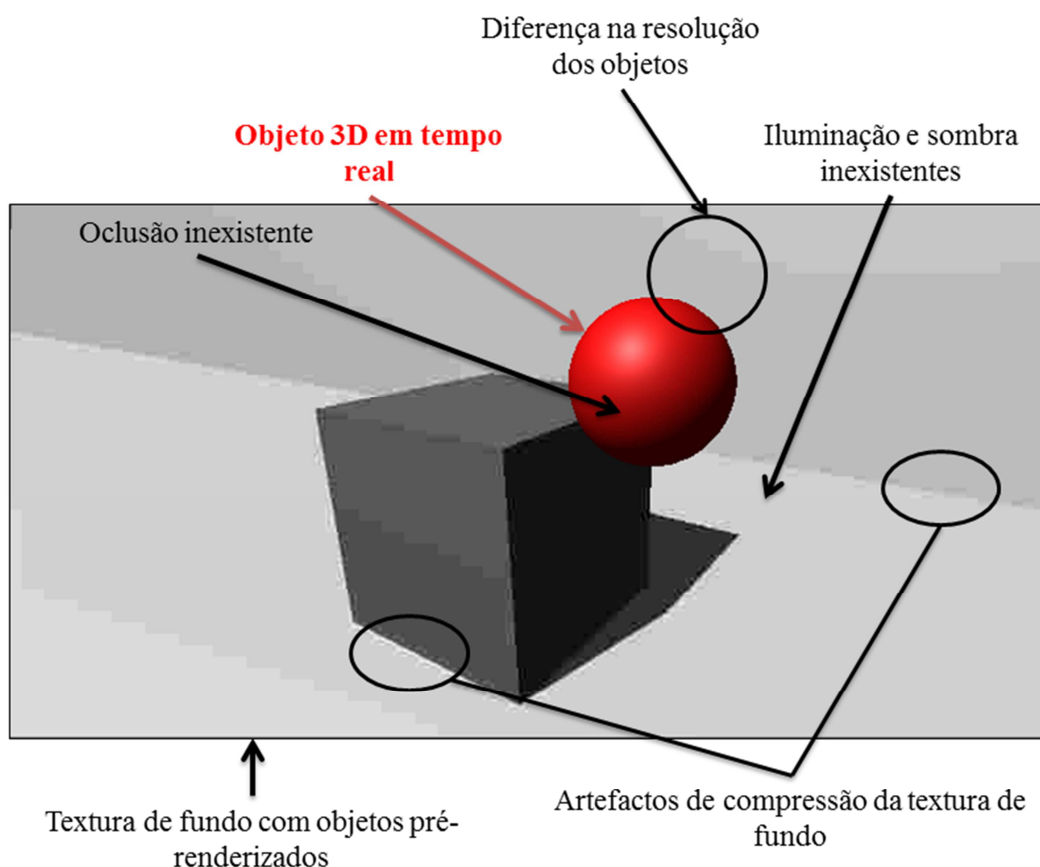


Figura 3: Exemplo de incoerências entre a textura de fundo e os objetos 3D em tempo real.

2.2.1 Oclusões e Transparências

No contexto proposto, as oclusões entre objetos da cena pré-renderizada são automaticamente resolvidas e podem ser visíveis na textura de fundo. As oclusões mútuas entre os objetos 3D sobrepostos em tempo real também são resolvidas usando os mecanismos normalmente usados no cálculo de visibilidade. Porém, como os objetos da cena original só existem representados na textura de fundo, estes são sempre ocultados pelos objetos 3D sobrepostos em tempo real. Isso é problemático pois, em algumas situações, se um objeto da cena original estiver mais próximo que um objeto 3D sobreposto em tempo real, este último pode ser ocultado pelo objeto da cena original. Na figura 2 é possível visualizar uma situação em que um objeto 3D em tempo real (esfera vermelha) não se encontra corretamente oculto por um objeto da cena pré-renderizada (cubo cinzento) e uma situação que deveria ser a correta.

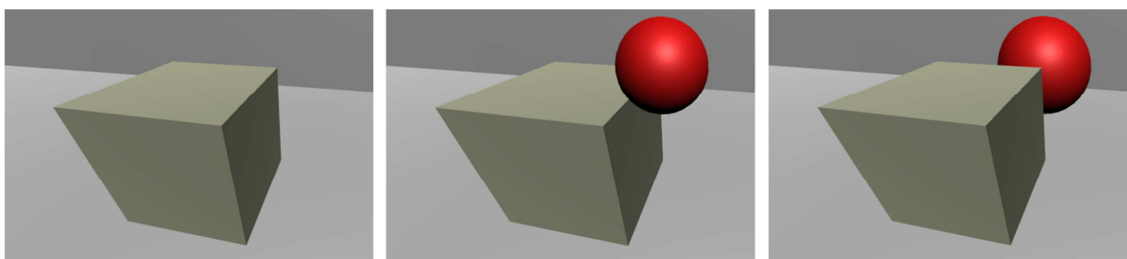


Figura 4: Cena pré-renderizada (à esquerda), oclusão do objeto 3D em tempo real inexistente (ao centro), objeto 3D em tempo real corretamente ocultado (à direita).

Este problema ocorre frequentemente na área da realidade aumentada em que objetos 3D virtuais devem ser ocultados por objetos do mundo real. Nesta área, são frequentemente destacadas duas técnicas diferentes para criar representações corretas das oclusões entre objetos virtuais e reais [3].

A primeira técnica consiste em extrair a profundidade dos objetos do mundo real. Para tal, o mundo real é capturado não através de uma única câmara, mas sim através de duas câmaras separadas por uma determinada distância para criar uma visão estereoscópica do mundo real. Assim é possível calcular a disparidade lateral entre as duas imagens obtidas [3-6]. Também pode ser usada a câmara de profundidade integrada no dispositivo Kinect [7] para poder reconstruir a profundidade entre os objetos reais [8, 9]. Nos dois casos é obtido um mapa de profundidades que associa uma profundidade a cada *pixel* da imagem do mundo real.

Em computação gráfica, para a renderização de cenas virtuais é efetuado o cálculo de visibilidade para determinar quais são os objetos visíveis ou ocultados por outros. Para tal, é frequentemente utilizada a técnica de *Z-buffering* que utiliza uma estrutura de dados chamada *Z-buffer*. Esta estrutura é uma imagem do campo de visão em que o valor de cada *pixel* corresponde à profundidade (valor da coordenada *Z*) dos objetos presentes na cena. Quanto mais próximo estiver um objeto menor será o valor dos *pixels* do *Z-buffer* [10].

Técnicas de renderização com utilização de fundos pré-gerados

Ao utilizar o mapa de profundidades do mundo real diretamente no *Z-buffer* da aplicação final antes de desenhar os objetos virtuais, é possível ocultar os objetos virtuais com os objetos do mundo real [6]. Contudo, esta técnica está muito limitada à resolução do mapa de profundidades obtido e uma baixa resolução pode resultar em oclusões incompletas ou incorretas dos objetos. Esta desvantagem é ainda mais notória para objetos muito distantes.

A segunda técnica consiste em usar objetos “fantasma” aqui por *proxies* que são representações simplificadas dos objetos reais no mundo virtual. Estes objetos “fantasmas” são sobrepostos aos objetos reais sem serem propriamente desenhados. Só são utilizados os valores de profundidade dos objetos “fantasmas” e para tal estes são escritos no *Z-buffer* antes dos objetos virtuais. Os objetos “fantasmas” tornam-se então transparentes para a imagem de fundo mas não para os objetos virtuais que passam por trás deles [11, 12]. A grande vantagem deste método é de poder ocultar de forma exata os objetos virtuais, sejam eles próximos ou muito distantes [11]. Esta técnica exige que os objetos “fantasmas” sejam previamente modelados.

2.2.2 Transparências

Na construção de um ambiente virtual, é frequente a utilização de materiais transparentes. Estes constituem um tipo de material com características diferentes dos materiais opacos pois os objetos transparentes devem ter a capacidade de deixar transparecer os objetos que estiverem por trás deles. Isto altera muito a forma como objetos transparentes devem ser desenhados bem como o cálculo de visibilidade dos objetos que estiverem por trás deles.

Tendo em consideração o sistema híbrido aqui proposto juntamente com as transparências, a pré-renderização de um ambiente virtual para obter a textura de fundo faz com que qualquer material transparente perca esta propriedade. Como tal, os objetos 3D em tempo real não vão poder transparecer quando passarem por trás de objetos transparentes da cena pré-renderizada.

Tipicamente os efeitos de transparência são conseguidos através da aplicação de um material com uma textura que possui um canal extra para além dos três canais de cor RGB. Este canal é conhecido pela designação de canal alfa. Usando técnicas como o *alpha blending* descrito por T. Porter e T. Duff [13] é possível tornar determinados objetos transparentes para outros objetos. O valor de alfa indica ao sistema de rasterização a proporção com que a cor da textura do objeto transparente deve ser adicionada à cor dos objetos que estiveram por trás. Quanto mais alto é o valor do canal alfa, mais opaco é o material do objeto e vice-versa.

Assim, continuando com o uso de *proxies* simplificados, estes podem possuir um material transparente de forma a poder transparecer os objetos 3D em tempo real que passarem por trás.

Outra possibilidade passa pelo uso de várias camadas sobrepostas de vídeo de fundo com transparência [14]. Se for colocada uma camada de vídeo transparente com objetos pré-renderizados transparentes em frente aos objetos 3D em tempo real, é assim possível fazer que estes últimos transpareçam por trás dos objetos pré-renderizados.

2.2.3 Iluminação

Em computação gráfica a iluminação envolve dois conceitos diferentes. O primeiro é o sombreamento (*shading*) dos objetos gerado a partir das fontes de luz presentes no ambiente virtual. O segundo conceito é o cálculo de sombras projetadas por objetos sobre outros objetos. Nesta secção é abordado o conceito de sombreamento dos objetos enquanto que o calculo de sombras só é abordado na secção 2.1.4.

Um ambiente virtual bem concebido deve permitir que os cálculos de iluminação sejam corretamente processados, pois a iluminação é um fator muito importante para promover uma boa sensação de imersão e tem um custo computacional elevado. Normalmente a iluminação incidente sobre qualquer objeto deve ser refletida e irradiada na proximidade direta do objeto. Considere-se, por exemplo, uma cena em que existe uma fonte de luz branca dentro de um quarto completamente fechado e composto só por chão, teto e paredes brancas. Logicamente, neste cenário, as paredes devem apresentar uma cor branca. Se for adicionado um objeto vermelho no centro do quarto, parte da luz incidente sobre ele deverá ser refletida e irradiada sobre as paredes do quarto, fazendo com que apresentem alguma cor vermelha. O mesmo ocorre no sentido inverso, isto é, o objeto vermelho deve apresentar cor vermelha ligeiramente mais clara devido à luz refletida pelas paredes brancas. Admitindo agora que o quarto é uma cena que foi pré-renderizada numa textura e que o objeto vermelho é um objeto 3D em tempo real, as paredes representadas na textura não vão poder receber de forma coerente a iluminação irradiada pelo objeto 3D em tempo real e vice-versa. Na figura 3 é apresentada a situação em que a iluminação incidente sobre o objeto 3D em tempo real não é irradiada sobre os objetos da cena pré-renderizada e a situação que seria de esperar.

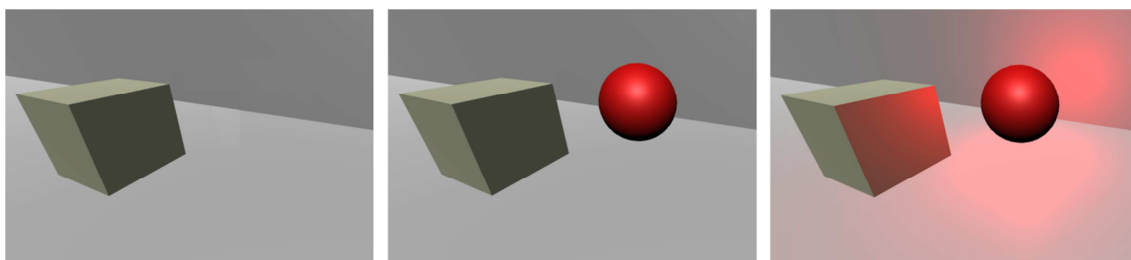


Figura 5: Cena pré-renderizada (à esquerda), iluminação incorreta com a adição de um objeto 3D em tempo real (ao centro), iluminação correta (à direita).

Este problema é frequentemente estudado em realidade aumentada pois os objetos do mundo real só existem sob a forma de uma textura.

Um método frequentemente usado para resolver uma parte deste problema é conhecido por *Image-based lighting* (IBL) [15, 16]. Resumidamente, este método consiste em capturar uma imagem do mundo real através de uma câmara panorâmica e mapeá-la num cubo ou esfera que envolve os objetos virtuais para poder simular a iluminação da cena real. Este método pode ser

Técnicas de renderização com utilização de fundos pré-gerados

melhorado usando imagens *High Dynamic Range* (HDR) que contêm uma ampla gama de luminosidade e proporcionam uma iluminação mais realista [17, 18]. Porém este método só considera a iluminação distante e só consegue refletir a iluminação dos objetos reais sobre os objetos virtuais, não permitindo o inverso.

Outro método frequentemente usado em realidade virtual é o *Light mapping* [19]. Este método consiste em calcular previamente a luminosidade das superfícies dos objetos estáticos existentes num ambiente virtual. Assim é obtido um conjunto de texturas que são usadas em fases posteriores que requerem processamento em tempo real. Como o cálculo do mapa de iluminação não necessita de ser em tempo real, este método é muito atrativo para acelerar a renderização final, pois basta utilizar as texturas previamente geradas para simular a iluminação do cenário. À semelhança do método anterior, este só consegue refletir a iluminação dos objetos reais sobre os objetos virtuais e não o inverso.

Um método relativamente recente utiliza uma *shading probe* baseada no conceito de *light probe*, que consiste num conjunto de várias sondas que medem o sombreamento de vários pontos do mundo real [20]. Ao reutilizar o sombreamento no mundo virtual é possível aplicar aos objetos virtuais o sombreamento e iluminação provenientes dos objetos reais.

Estes métodos podem resolver em grande parte os problemas de iluminação dos dois tipos de objetos do sistema híbrido, contudo, os problemas de iluminação estão frequentemente associados aos problemas de sombras pois ambos são problemas semelhantes mas abordados de um ponto de vista diferente.

2.2.4 Sombras

Nesta secção pretende-se expor técnicas que permitem o cálculo de sombras entre objetos de uma cena virtual pois as sombras constituem um fator muito importante para manter uma boa coerência entre os vários objetos presentes num determinado cenário.

As sombras existentes entre objetos da cena pré-renderizada não são problemáticas pois estas são calculadas durante a fase de pré-renderização. As sombras existentes entre objetos 3D interativos também não causam problemas ao serem calculadas pelos algoritmos habituais de projeção de sombras em tempo real. Porém, como já foi referido anteriormente, a fase de pré-renderização remove toda a informação tridimensional dos objetos da cena original, bem como a profundidade relativa entre eles. Como tal, os objetos da cena pré-renderizada não têm a possibilidade de projetar sombras sobre os objetos 3D interativos. Pelo mesmo motivo, os objetos 3D interativos não têm a possibilidade de projetar sombras coerentes sobre os objetos pré-renderizados. Na figura 4 é ilustrada uma situação em que um objeto pré-renderizado não projeta a sua sombra sobre um objeto 3D em tempo real e vice-versa. É igualmente apresentada a situação que se deveria esperar.

Técnicas de renderização com utilização de fundos pré-gerados

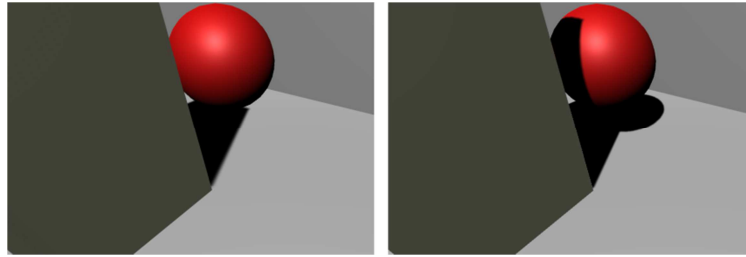


Figura 6: Projeção das sombras inexistentes (à esquerda), projeção correta das sombras (à direita).

Este problema existe igualmente na realidade aumentada e a solução mais utilizada baseia-se novamente na utilização de objetos 3D “fantasmas” que são representações dos objetos reais com a capacidade de projetar e receber sombras [21, 22]. Assim é possível calcular normalmente a projeção de sombras de objetos do mundo real sobre objetos virtuais e vice-versa.

2.2.5 Colisões

A pré-renderização de uma cena com objetos dinâmicos permite representar corretamente as colisões que ocorrem entre os objetos da cena original. Nesta situação, a textura de fundo obtida deve então ser um vídeo. Da mesma forma, colisões que ocorrem entre os objetos 3D em tempo real podem ser processadas pelos métodos tradicionais de detecção de colisões. No processo de pré-renderização, todos os objetos da cena original vão perder a informação sobre as suas propriedades físicas, mais particularmente, vão perder as malhas tridimensionais de colisões associadas. Consequentemente, a textura que representa a cena original, por si só, não é suficiente para determinar as colisões entre os objetos 3D interativos em tempo real e os objetos da cena original.

Novamente, para resolver este problema de colisões não processadas, a área da realidade aumentada fornece soluções pertinentes. Isso compreende-se pelo facto dos objetos do mundo real não terem forma de colidir com os objetos virtuais e vice-versa. Assim existem vários métodos propostos para solucionar este problema.

O primeiro método implica técnicas de segmentação da imagem capturada para extrair os contornos dos objetos móveis do mundo real. Um exemplo de objeto pode ser o corpo de um utilizador que está a interagir com um sistema de realidade aumentada. Após a extração dos objetos móveis do mundo real é possível efetuar a detecção de colisões com objetos virtuais [23, 24]. Este método não tem em consideração a profundidade dos objetos dos dois mundos, como tal, as colisões são processadas na superfície de um plano e não no espaço 3D.

Técnicas de renderização com utilização de fundos pré-gerados

O segundo método usado por D. E. Breen, R. T. Whitaker, E. Rose e M. Tuceryan, consiste na extração do mapa de profundidades do mundo real através de duas câmaras. Assim, para detetar colisões, basta comparar os *pixels* do mapa de profundidades do mundo real com os respetivos *pixels* do *Z-buffer* do mundo virtual. Se um valor do *Z-buffer* for superior ao respetivo valor do mapa de profundidades obtido, então uma colisão ocorreu entre um determinado objeto do mundo virtual e um objeto do mundo real [25]. Uma variante deste método utiliza um dispositivo Kinect para obter o mapa de profundidades do mundo real que é depois convertido num terreno virtual com a malha de colisão aproximada do mundo real. Esta malha é depois usada para detetar as colisões com os objetos virtuais [8]. Uma desvantagem muito relevante deste método baseado em mapa de profundidades, advém da sua incapacidade em detetar todas as colisões nomeadamente aquelas que ocorrem entre objetos ocultos por objetos [11].

O terceiro método utiliza conjuntos de *voxels*¹ para criar no mundo virtual uma aproximação dos objetos do mundo real. Os objetos do mundo real devem ser igualmente extraídos por técnicas de análise de imagem e é obtido um mapa de profundidades do mundo real através de duas câmaras que permitem inferir a distância dos objetos em relação à câmara. A quantidade e tamanho dos *voxels* podem ser alterados dinamicamente consoante o tamanho do objeto real detetado. Como os *voxels* fazem parte do mundo virtual, estes podem facilmente colidir com outros objetos virtuais. Logo, os objetos do mundo real podem colidir com objetos do mundo virtual [4, 27].

O quarto método utiliza modelos 3D “fantasmas” que são representações transparentes dos objetos do mundo real, previamente modelados. Os modelos possuem malhas de colisão com uma forma muito similar à dos objetos reais. Assim os objetos do mundo virtual podem facilmente colidir com objetos do mundo real. A grande vantagem deste método reside na possibilidade de detetar colisões que possam ocorrer em zonas ocultas por objetos reais [11]. Contudo, mantém-se a desvantagem resultante do uso direto de geometria como malhas de colisão que é computacionalmente exigente para a unidade de processamento central.

O último método aproxima um corpo humano detetado numa imagem do mundo real, por um conjunto de esferas, cápsulas, caixas e modelos 3D convexos criados em tempo de execução no mundo virtual. Cada parte do corpo é representada por uma forma específica e o tamanho de cada modelo geométrico pode ser alterado em função das proporções do corpo detetado. Como cada modelo possui malhas de colisão representativas do corpo real, este pode colidir com outros objetos do mundo virtual [28]. No contexto deste estudo, este tipo de técnica pode ser útil para a simplificação da geometria de colisão utilizada para os *proxies* pois o uso de primitivas

¹ Em computação gráfica, os *voxels* são frequentemente utilizados como alternativa à representação típica de objetos 3D por vértices e polígonos. Resumidamente, o termo *voxel* vem da contração das palavras “volume” e “pixel”. Um *voxel* permite representar objetos tridimensionais a partir uma unidade estrutural básica como por exemplo um cubo. Como tal, o espaço virtual deve estar subdividido numa grelha de unidades elementares sendo cada uma representada por um *voxel* tal como uma imagem se encontra subdividida em *pixels* [26].

Técnicas de renderização com utilização de fundos pré-gerados

de colisão que aproximam a forma dos objetos é computacionalmente menos exigente que o uso de malhas de colisão com geometria exata usada pela técnica anterior.

2.2.6 Artefactos de codificação da textura de fundo

A imagem ou vídeo gerado pela pré-renderização da cena original pode ser utilizada sem qualquer tipo de compressão para manter a melhor qualidade possível, pois a utilização de técnicas de compressão de imagem *lossy* provocam perdas na qualidade [29]. Porém isso pode provocar um grande consumo da banda de comunicação disponível entre os vários componentes do dispositivo móvel (como por exemplo entre a unidade de processamento central e a memória dinâmica) degradando consideravelmente o desempenho. Como tal, a textura obtida deve ser comprimida usando algoritmos comuns de compressão de imagem e vídeo. Mesmo assim, a maioria dos algoritmos introduzem artefactos visuais notórios e que tornam incoerente a fusão dos objetos 3D em tempo real com a textura de fundo. Os artefactos são provocados pelas técnicas de compressão que convertem os espaços de cor ou dividem a imagem em sub-blocos de *pixels* dando origem a artefactos como *blocking*, *blurring*, *ringing*, *mosquito noise* e efeitos de escada [29]. Para além dos artefactos, a diferença entre a resolução da textura e a resolução dos objetos 3D interativos pode igualmente provocar incoerências entre ambos. Se a resolução da textura não for suficientemente grande, o tamanho dos *pixels* da textura serão percebidos como sendo maiores que os *pixels* dos objetos 3D interativos criando incoerências visuais entre os dois tipos de objetos.

Em realidade virtual, um método tradicionalmente usado para resolver este problema é conhecido como *anti-aliasing*. Esta técnica permite suavizar linhas muito pronunciadas existentes numa imagem mitigando assim o fenómeno visual conhecido por “efeito de escada” [30].

Em realidade aumentada, J. Fischer [31] propõe igualmente o uso de uma variante do *anti-aliasing* que permite combinar um vídeo do mundo real com objetos virtuais. Para tal, num primeiro passo, a resolução do vídeo é aumentada. Num segundo passo é efetuada a média dos *pixels* existentes unicamente nas regiões de sobreposição entre os objetos virtuais e o vídeo de fundo, criando assim uma fusão credível entre os dois.

2.3 Requisitos da Realidade Virtual para dispositivos Móveis

Para facultar uma boa sensação de presença e imersividade em ambientes virtuais, um HMD e o dispositivo de geração de imagem devem apresentar requisitos específicos. Devido a estes requisitos serem exigentes, a realidade virtual necessita muito em termos de poder de processamento gráfico por parte do dispositivo de geração de imagem incluindo a unidade de processamento gráfico.

Segundo M. Abrash [40], os principais requisitos que um bom HMD e dispositivo de geração de imagem devem apresentar são:

- Um campo de visão de pelo menos 80°;
- Resolução de ecrã adequada;
- Persistência dos *pixels* do ecrã inferior a 3ms;
- Taxa de refrescamento de imagem suficientemente elevada;
- Iluminação uniforme dos *pixels* do ecrã;
- Lentes que introduzem o menor número possível de distorção e aberrações;
- Sensor de orientação com precisão de pelo menos um quarto de grau;
- Latência entre movimento e imagem gerada (*motion to last photon*) inferior a 25ms.

Em termos temporais, a taxa de refrescamento de imagem e a latência são os principais fatores a ter em consideração para evitar sensações de enjôo. Tal sintoma ocorre quando existe uma grande diferença entre a orientação percebida pelo sistema vestibular e a imagem percebida pelo sistema visual humano [41-43]. Logo é necessário ter latências baixas e taxas de refrescamento de imagem altas para manter a imagem percebida pelo utilizador coerente com os seus movimentos.

Neste trabalho, só é dada importância à taxa de refrescamento de imagem, pois é a única característica que não está diretamente dependente das propriedades físicas e técnicas inerentes ao tipo de HMD utilizado. Quanto maior for o número de imagens por segundo, melhor será a perceção da imagem por parte do sistema visual do utilizador. Porém, autores como M. Abrash [40], D. Chu, K. Boos e E. Cuervo [44] consideram que 60 imagens por segundo é um valor razoável para oferecer uma boa perceção de imagem sem provocar sensações de enjôo. Este é um valor mínimo recomendado que será usado como objetivo a atingir para os testes de desempenho, contudo, isso não impede a possibilidade de poder atingir taxas de refrescamento ainda mais altas, possibilitando a visualização de cenas virtuais mais complexas.

2.4 Trabalhos Relacionados

A técnica proposta que faz uso de cenários pré-renderizados não é novidade e foi muito usada na década de 90 principalmente em videogames que utilizavam cenários renderizados numa textura 2D planar e projetada em perspectiva sobre um plano. Nesta altura, as consolas e outras plataformas para videogames não possuíam capacidades gráficas suficientes para gerar um ambiente virtual completamente tridimensional em tempo real e com boa qualidade visual. Contudo, o aparecimento do CD-ROM permitiu o uso de conteúdos multimédia de grandes dimensões, como imagens e vídeos. Podem ser citados como exemplos *The 7th Guest* [35] e várias sequelas dos jogos *Resident Evil* [36] e *Final Fantasy* [45]. Nestes videogames foram pré-renderizados vários cenários de fundo com alta qualidade visual e nos quais o jogador podia progredir através de um avatar tridimensional controlado em tempo real. Nestes exemplos, cada cenário de fundo fixo é projetado num plano e a câmara virtual é igualmente fixa, permitindo assim um único ângulo de visão para cada cenário. Para situações em que o cenário de fundo incluísse objetos móveis, é igualmente usado um vídeo em vez de uma imagem fixa para recriar o aspeto dinâmico do cenário.

Mais recentemente, o Google Street View [46] utiliza fotografias panorâmicas do mundo real para permitir aos utilizadores navegarem pelas ruas de vários locais do mundo. A arquitetura do sistema é do tipo cliente-servidor, em que o servidor possui uma base de dados de imagens panorâmicas associadas a localizações (latitude e longitude) e o cliente é o navegador internet do utilizador. Aqui, as imagens panorâmicas são projetadas sobre o interior de uma esfera. O utilizador pode interagir ao mover o cursor e selecionar a próxima vista panorâmica desejada. O cursor é um objeto interativo sobreposto à textura panorâmica, que permite destacar a topologia do terreno visualizado. Tal é possível porque é igualmente utilizado um mapa de profundidades panorâmico do cenário. É importante salientar que são apenas permitidos 3 graus de liberdade como movimentos de rotação e o utilizador pode navegar através de pontos de teleporte que permitem visualizar o cenário em posições diferentes.

No contexto da realidade virtual mista, F. Okura, M. Kanbara, e N. Yokoya [47] propõem a navegação num monumento real agora inexistente. Para tal, o monumento é reconstruído num modelo 3D sobreposto a um conjunto real de imagens panorâmicas esféricas da zona atual do monumento. Para a navegação, o conceito de teleporte é aqui inexistente pois a navegação é completamente livre (rotação e translação) e tal é possível por técnicas de *Image-Based Rendering* (IBR).

Mais direcionados para a realidade virtual em dispositivos móveis, D. Chu, K. Boos e E. Cuervo [44] desenvolveram um conceito muito semelhante ao proposto no presente trabalho. Estes autores usam técnicas de pré-renderização em pontos específicos de um cenário virtual para obter uma organização hierárquica dos dados obtidos durante uma fase de pré-

Técnicas de renderização com utilização de fundos pré-gerados

renderização. Os dados são essencialmente imagens capturadas sob a forma de projeções cúbicas estereoscópicas que contêm informação de cor e profundidade da cena virtual. Devido ao facto desta informação estar pré-calculada e devidamente organizada, isso possibilita uma visualização em tempo real com baixa latência em dispositivos de baixo desempenho. Apesar de só existir informação pré-calculada em determinados pontos, é dada igualmente a hipótese de navegar no ambiente virtual com 6 graus de liberdade e tal é possível graças a técnicas de IBR aplicadas quando o utilizador se encontra fora de pontos pré-calculados. O método permite igualmente a inclusão de objetos animados pelas mesmas técnicas de pré-renderização e interpolação entre imagens por IBR aplicados unicamente a estes tipos de objetos e sobrepostos ao fundo através de métodos aditivos. A solução desenvolvida pelos autores permite assim atingir 60 imagens por segundo em dispositivos móveis modestos. Apesar de possuir vantagens evidentes, o método utilizado não permite cenários com muitos objetos animados devido a limitação dos fundos pré-renderizados serem estáticos e não permite igualmente adicionar objetos interativos pois não é contemplada a possibilidade de existirem colisões entre objetos.

No evento Google I/O 2017, a empresa Google apresentou uma nova tecnologia igualmente desenvolvida no sentido de poder visualizar em tempo real, ambientes virtuais em dispositivos móveis com a qualidade visual equivalente à obtida em unidades de processamento gráfico de alto desempenho [48]. A tecnologia intitulada “*Seurat*” foi utilizada como prova de conceito pela ILMxLab para demonstrar as suas potencialidades com conteúdos cinematográficos gerados por computador. A Google ainda não publicou informação detalhada sobre o modo de funcionamento da tecnologia, no entanto, foi revelado que é usada uma técnica designada por “*surface light-fields*”. Esta técnica consiste em definir no cenário original, uma área de visualização para o utilizador e capturar amostras das possíveis perspectivas existentes para determinar todos os objetos visíveis a partir dessa área. Os conteúdos do cenário original são simplificados no sentido de reduzir drasticamente o número de polígonos, até possibilitar o seu uso em dispositivos móveis mantendo a qualidade visual da cena original. Se o dispositivo móvel estiver equipado com sensores de localização espacial adequados, além dos de orientação, esta tecnologia permite igualmente a movimentação no cenário virtual com 6 graus de liberdade.

2.5 Resumo e Conclusões

Neste capítulo foram abordadas várias técnicas existentes nas áreas da computação gráfica, realidade aumentada e realidade mista para resolver problemas de oclusões e colisões entre objetos tridimensionais e objetos representados através de uma textura.

Foi possível identificar as principais características técnicas que um sistema de realidade virtual deve possuir para poder maximizar a imersão em cenários virtuais minimizando a sensação de enjoo, destacando-se um número de 60 imagens por segundo como objetivo a atingir neste trabalho.

Foram igualmente apresentados vários trabalhos relacionados com este estudo. Alguns diferem bastante, outros abordam claramente o tema proposto revelando a vantagem de permitirem uma movimentação com 6 graus de liberdade ao invés dos 3 graus de liberdade da solução proposta neste trabalho inerente ao uso de vídeos 360° por 180°. Contudo, propõe-se aqui a possibilidade de adicionar objetos 3D interativos em tempo real e com a possibilidade de colidirem com os restantes objetos estáticos do cenário virtual. Com isso, é possível verificar que ainda não existe uma solução universal, existindo assim, uma necessidade crescente de criar soluções para visualizar cenários de realidade virtual com interatividade e alta qualidade visual em dispositivos móveis com baixo poder de processamento gráfico.

Capítulo 3

Arquitetura e Implementação

Neste capítulo é apresentada a arquitetura do sistema incluindo os principais módulos estruturais que constituem o sistema desenvolvido. Este permite gerar um ambiente virtual com geometria simplificada e geometria interativa sobrepostas a vídeos estereoscópicos de 360 por 180 graus que permitem fornecer uma alta qualidade visual a cenários virtuais. São igualmente expostos os métodos utilizados para resolver os problemas de investigação.

3.1 Arquitetura do sistema

O sistema desenvolvido é constituído por dois subsistemas principais sendo o primeiro, o sistema de pré-renderização, e o segundo, o sistema de visualização híbrido com fundo 360° e objetos 3D interativos em tempo real. Na figura 7 é apresentada a arquitetura geral do sistema incluindo os dois principais subsistemas, fluxo de dados e processos existentes em cada um deles, e estes serão descritos em mais detalhe nas secções seguintes.

O sistema de pré-renderização inclui a cena virtual totalmente tridimensional e baseia-se na informação presente na cena para criar os meta-dados necessários à obtenção da solução final. Esta solução encontra-se implementada no sistema de visualização híbrido. Este segundo sistema utiliza os meta-dados gerados no sistema de pré-renderização ao qual são acrescentados objetos 3D interativos que permitem ao utilizador interagir com o cenário final. Cada um dos dois sistemas contém módulos de exportação e importação de dados. No sistema de pré-renderização existe igualmente um passo opcional que deve ser aplicado em situações em que a geometria da cena original tem que ser simplificada.

É importante salientar que toda a arquitetura do sistema foi desenvolvida com a ajuda do ambiente de edição do motor de jogo Unity 5 [49] que permite criar facilmente cenas de realidade virtual.

Arquitetura e Implementação

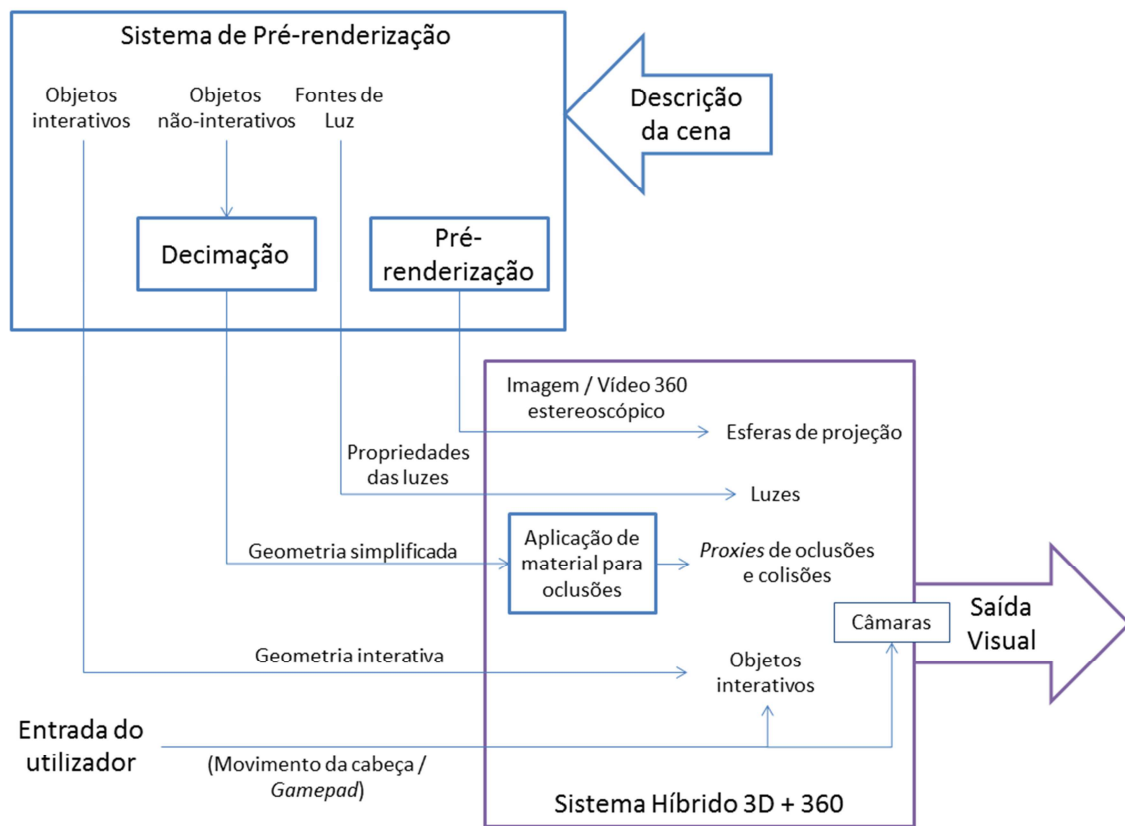


Figura 7: Arquitetura geral do sistema

3.1.1 Sistema de pré-renderização

Este subsistema é o primeiro elemento essencial na construção e processamento da cena virtual que se pretende visualizar em dispositivos móveis com baixo desempenho.

O primeiro passo a ser efetuado neste módulo reside na criação do grafo de cena do cenário virtual pelos métodos usuais de modelação tridimensional. Consoante as necessidades do cenário, podem ser criados objetos 3D fixos ou dinâmicos com qualquer tipo de material. Podem igualmente ser incluídos sistemas complexos de partículas animados. Relativamente às fontes de luz, estas são igualmente criadas livremente no grafo de cena para produzir a iluminação pretendida sobre os objetos. Quando a fase de modelação estiver concluída, pode ser efetuado o cálculo de iluminação global que, no caso do Unity, é feito a partir das fontes de luz e modelos geométricos que foram considerados como sendo objetos estáticos devido ao facto da iluminação global pré-calculada não ter contribuído sobre objetos dinâmicos. O cálculo da iluminação global, apesar de dar um maior realismo à cena virtual modelada, é um passo complementar e podem ser usadas fontes de luz em tempo real como alternativa.

Arquitetura e Implementação

Os objetos 3D interativos em tempo real podem ser adicionados ao grafo de cena aplicando novamente técnicas comuns de modelação ou por importação de objetos tridimensionais previamente modelados e texturizados.

Após a modelação e iluminação da cena virtual estarem concluídos, devem ser configurados três módulos essenciais para a criação dos meta-dados usados posteriormente. O primeiro módulo é responsável por capturar uma ou várias vistas panorâmicas estereoscópicas da cena virtual. O segundo módulo exporta toda a geometria presente na cena incluindo a posição da câmara de captura para o formato FBX da Autodesk [50] pelo facto de ser um formato facilmente reutilizável. O formato FBX permite igualmente guardar as animações de objetos dinâmicos que constituem uma informação importante que se deve igualmente ter em consideração, como será possível ver na secção 3.3.3. Os objetos que constituem a geometria exportada, excluindo os objetos interativos, serão utilizados para criar os *proxies* no sistema de visualização híbrido. O terceiro módulo exporta as fontes de luz com as suas propriedades em formato *eXtensible Markup Language* (XML). Os módulos de exportação e respectivos meta-dados gerados serão descritos na secção 3.3.

Para manter a portabilidade do sistema proposto, os dados são exportados em formatos abertos como FBX e XML de forma a poderem, se necessário, ser gerados ou consumidos por outras plataforma por serem formatos amplamente usado e bem definidos. Se assim não fosse, os meta-dados podiam ficar em *prefabs* do Unity, mas essa seria uma solução fechada a plataforma utilizada neste trabalho.

Arquitetura e Implementação

3.1.2 Sistema de visualização híbrido com fundo 360 e objetos 3D

Neste subsistema são utilizados os meta-dados gerados pelo subsistema anterior para reconstruir a cena virtual original com as simplificações necessárias para ser visualizada em dispositivos de baixo desempenho. Na figura 8 é apresentada a estrutura do sistema de visualização híbrido.

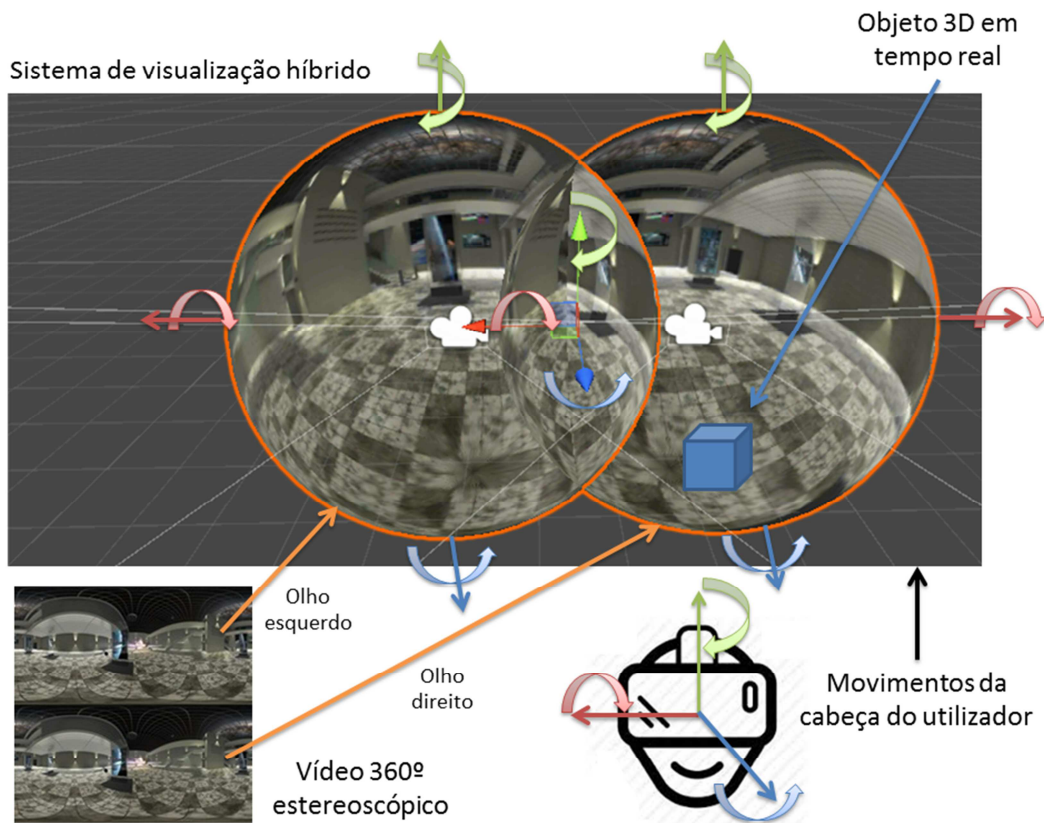


Figura 8: Estrutura do sistema de visualização híbrido.

A imagem ou vídeo panorâmico estereoscópico exportado pelo sistema de pré-renderização deve ser aqui utilizado para simular a informação de cor do cenário virtual original. Para tal, a imagem ou vídeo panorâmico deve ser convertido numa textura mapeada através de uma projeção equiretangular sobre o interior de duas esferas suficientemente grandes para conter todos os *proxies* correspondentes aos objetos presentes na vista panorâmica. De uma certa forma, as esferas simulam uma abóbada celeste ou *skybox* para cada olho do sistema de visão. Como tal, deve então existir no grafo de cena um par de câmaras dirigidas no mesmo sentido, com o objetivo de capturar a informação visual correspondendo a cada olho. Estas devem estar separadas por uma distância inter-pupilar idêntica à distância usada no módulo responsável por capturar as vistas panorâmicas da cena virtual no sistema de pré-renderização. Cada câmara deve então encontrar-se no centro da respetiva esfera e isso pode ser facilmente implementado tornando cada esfera filha da respetiva câmara na hierarquia do grafo de cena. As

Arquitetura e Implementação

câmaras e as esferas devem possuir três eixos de rotação comuns cujo ponto de intersecção está situado no meio do segmento que une as duas câmaras, isto é, situado à meia distância interpupilar. Estes eixos correspondem aos eixos de rotação da cabeça do utilizador, permitindo assim mudar facilmente a orientação do campo de visão na cena virtual consoante a orientação da cabeça do utilizador no mundo real. A rotação da cabeça é igualmente aplicada a cada esfera. Para sincronizar a orientação das câmaras virtuais com a orientação da cabeça do utilizador, são utilizados os sensores de orientação do dispositivo móvel cujas medições são adquiridas após processamento por um filtro de Kalman [51, 52]. Os dados de orientação são obtidos com a integração do *asset* da Google Cardboard para o Unity [53]. Este módulo, além de fornecer uma interface de programação para a aplicação da Google Cardboard, permite igualmente aplicar uma distorção de imagem adaptada ao sistema de lentes ópticas usadas pelo HMD utilizado.

A geometria exportada pelo sistema de pré-geração é adicionada à cena de trabalho através do método de importação de dados FBX nativo que o motor de jogo possui para este efeito. Antes do processo de importação, a geometria pode opcionalmente passar por um processo de decimação descrito na secção 3.4.1 e que visa a redução do número de triângulos que a constitui. A geometria importada para gerar os *proxies* não necessita de ter qualquer tipo de informação sobre normais, nem sobre coordenadas de mapeamento de texturas (UV). A geometria importada para gerar os *proxies* deve simplesmente conter a informação sobre os vértices e arestas que constituem a sua malha. Para gerar os *proxies*, é aplicado à geometria importada um material que torna estes objetos transparentes para o fundo e que permite igualmente a oclusão de objetos 3D interativos em tempo real que possam eventualmente passar por trás dos *proxies*. O sistema de visualização híbrido deve garantir o alinhamento correto dos *proxies* com as imagens de fundo. Para facilitar este processo optou-se por colocar os *proxies* e o par de câmaras na posição que se encontravam originalmente no sistema de pré-renderização. Salienta-se que a posição original do par de câmaras é importada juntamente com a geometria que origina os *proxies*.

As fontes de luz com as suas propriedades devem ser importadas através da leitura dos dados em XML gerados pelo sistema de pré-renderização. Assim é possível obter novamente a iluminação proveniente da cena original. Salienta-se que só é necessário importar fontes de luz cujo contributo para o cálculo de iluminação é feito em tempo real. Estas luzes devem ser importadas unicamente para a iluminar os objetos 3D interativos em tempo real.

Finalmente, os objetos 3D interativos em tempo real são igualmente importados da mesma forma que a geometria que origina os *proxies*, mas a geometria e os materiais originais dos objetos 3D interativos devem ser mantidos sem sofrer qualquer tipo de alteração. O sistema de visualização híbrido altera a ordem de desenho destes objetos relativamente à ordem de desenho dos *proxies*. Esta alteração descrita na secção 3.2.1 é essencial para permitir que os objetos interativos sejam ocultados quando passarem por trás dos *proxies*.

3.2 Implementação da coerência entre a textura de fundo e objetos 3D interativos

Nesta parte são apresentadas as soluções utilizadas para resolver as questões de investigação enunciadas no capítulo 1. Contudo, lembra-se que neste trabalho o foco é na resolução dos problemas de oclusões e colisões existentes entre objetos representados pelo vídeo de fundo 360° e objetos 3D interativos em tempo real. Os problemas relacionados com transparências, iluminação e sombra são abordados mas apenas com implementações simples ou sugestões de implementação que deverão ser aprofundadas em futuros trabalhos.

3.2.1 Oclusões e Transparências

Como foi verificado no capítulo 2, é possível dar solução ao problema de oclusão dos objetos interativos por parte dos objetos pré-renderizados através do uso de objetos “fantasma” aqui designados como *proxies*. Estes objetos são gerados no sistema de pré-renderização a partir da geometria presente no cenário virtual original.

É importante que os *proxies* sejam transparentes e estejam alinhados com o fundo pré-renderizado. Contudo, isso não é suficiente para ocultar eventuais objetos interativos que possam passar por trás dos *proxies*. Assim, a ordem de renderização deve ser alterada de tal forma que as esferas da imagem de fundo sejam desenhadas em primeiro, a seguir os *proxies* e finalmente os objetos 3D interativos em tempo real.

No Unity, a ordem de renderização de cada objeto é definida nas suas propriedades através de um número inteiro que por omissão tem um valor igual a 2000. Objetos que tiverem a mesma ordem são desenhados ao mesmo tempo. Quanto maior for a ordem de renderização, mais tarde um objeto será desenhado.

Para garantir então a ordem de renderização referida acima (esferas, *proxies*, objetos 3D), as esferas que suportam a imagem de fundo possuem uma ordem de renderização igual a 2000, a ordem de renderização dos *proxies* é alterada para 2001 e a dos objetos 3D interativos em tempo real é alterada para 2002.

Além disso, os *proxies* usam um material que desenha unicamente o que já foi renderizado no ecrã para tornar o objeto transparente, mas respeitando o cálculo de visibilidade efetuado através do Z-buffer.

Desta forma garante-se que os objetos 3D em tempo real possam ser efetivamente ocultados pelos *proxies*, caso passem por trás deles.

Imaginando que um objeto interativo se encontra por trás de um *proxy*, obtém-se os seguintes passos de renderização:

1. As esferas com as imagens de fundo são desenhadas;
2. O cálculo de visibilidade baseado no *Z-buffer* marca o objeto interativo como sendo invisível por estar atrás do *proxy*.

Arquitetura e Implementação

3. O *proxy* é desenhado com a informação de cor que já foi previamente desenhada no ecrã, ou seja, a imagem de fundo.
4. Seria agora a vez do objeto interativo ser desenhado mas como este foi marcado como invisível no *Z-buffer*, ele simplesmente não é desenhado pelo sistema de rasterização, sendo percebido como um objeto ocultado.

Neste trabalho optou-se por implementar a técnica descrita anteriormente, contudo, podem ser utilizados mapas de profundidade para realizar as oclusões dos objetos 3D interativos em tempo real. Os mapas de profundidades seriam capturados no sistema de pré-renderização em formato equiretangular juntamente com as imagens panorâmicas e utilizados no sistema híbrido. Para ocultar os objetos interativos, a parte do mapa de profundidade equiretangular correspondente a área de visualização do utilizador deveria ser escrita no *Z-buffer* depois da escrita dos objetos interativos. Como o *Z-buffer* iria conter a informação da profundidade, tanto dos objetos pré-renderizados como dos objetos interativos, o cálculo de visibilidade marcaria automaticamente os objetos interativos como sendo visíveis ou invisíveis conforme a sua posição perante os objetos pré-renderizados.

Seguindo esta possível metodologia o problema das oclusões pode ser resolvido, contudo, existiriam desvantagens importantes perante a metodologia que utiliza os *proxies*. A primeira desvantagem é consequência do formato utilizado para guardar o mapa de profundidades. Se for capturado um mapa com uma resolução de 8 bits, existiriam apenas 256 níveis de profundidade, fazendo com alguns objetos não fossem corretamente ocultados. Poder-se-ia então capturar mapas com 16 ou 32 bits para aumentar a precisão mas isso implicaria custos acrescidos no espaço de armazenamento necessário. Algoritmos de compressão podem ser úteis mas devem introduzir o menor número possível de artefactos para evitar erros de oclusões que sejam visualmente perceptíveis. Esta metodologia pode revelar-se eficaz em ambientes virtuais estáticos, sendo apenas necessário um único mapa de profundidades para cada ponto de visualização, porém, em ambientes dinâmicos já seria necessário capturar uma sequência de mapas de profundidades. Seria então necessário mais um fluxo de vídeo para a informação de profundidade para além do fluxo de vídeo 360° já existente. No entanto, a decodificação em simultâneo destes dois fluxos de vídeo em dispositivos móveis pode revelar-se problemática devido às exigências computacionais necessárias. Outra desvantagem importante desta metodologia reside no facto de não conseguir resolver os problemas de colisões, pois o mapa de profundidade não permite detetar colisões que ocorrem em zonas ocultas existentes entre objetos pré-renderizados levando à necessidade de duas soluções distintas.

Relativamente às transparências presentes nos objetos pré-renderizados, e apesar de não serem o foco deste trabalho, decidiu-se utilizar novamente os *proxies* para tentar mitigar este problema. Admitindo que um objeto pré-renderizado é completamente transparente e com o canal alfa igual a 100%, por exemplo uma janela totalmente transparente, então o método utilizado nesta situação foi de simplesmente não desenhar o respetivo *proxy*, desativando o seu

Arquitetura e Implementação

componente de renderização no motor de jogo. No caso em que o objeto pré-renderizado não seja completamente transparente, ou seja, é translúcido ou com partes opacas, como por exemplo um vitral, então, o *proxy* continua com o material original mas o *shader* de desenho é alterado para uma versão que é computacionalmente menos exigente. Salienta-se que os problemas de transparências não fazem parte dos problemas de investigação e como tal optou-se por abordagens simples que podem funcionar bem em determinadas situações mas noutras não. Por exemplo se no cenário virtual existirem muitos objetos com materiais transparentes diferentes, a abordagem proposta pode não ser escalável. Assim sendo esta problemática necessita uma investigação mais aprofundada em futuros trabalhos.

3.2.2 Colisões

Para que os objetos 3D interativos em tempo real possam colidir com os objetos pré-renderizados, é novamente aproveitada a informação da geometria dos *proxies* exportada pelo sistema de pré-renderização. Esta geometria pode ser utilizada para criar malhas de colisão idênticas às malhas de desenho dos objetos originais. As malhas de colisão podem ser geradas de forma cómoda no processo de importação dos *proxies* facultado pelo Unity, dando a instrução ao motor de jogo para criar malhas de colisão (*mesh colliders*) idênticas as malhas geométricas dos modelos importados. Com esta informação acrescentada ao sistema híbrido, o motor de física trata de processar normalmente as colisões entre os *proxies* e os objetos 3D interativos em tempo real.

Contudo, é importante salientar que o uso de malhas complexas de colisão pode ser muito custoso para o motor de física e pode tornar-se computacionalmente intensivo para a unidade de processamento central. Para mitigar este problema, pode ser usado uma malha geométrica simplificada em vez da malha geométrica original. Se for possível, a simplificação pode ser realizada utilizando primitivas de colisão disponibilizadas nativamente pelo motor de jogo. Por exemplo, se um objeto da cena original tiver uma forma que pode ser aproximada a um paralelepípedo cuboide, poderá ser usado uma caixa de colisão (*box collider*) que envolve o objeto evitando assim as necessidades computacionais exigidas pelo uso de malhas complexas de colisão.

3.2.3 Iluminação e Sombras

Os problemas relacionados com iluminação e sombras, apesar de bem definidos e destacados, não foram considerados como problemas prioritários de investigação. Contudo, tentou-se sugerir ou implementar possíveis soluções.

Como já foi referido no capítulo 2, ao serem pré-renderizados, os objetos da cena original não têm a possibilidade de dar o seu contributo de iluminação sobre os objetos 3D interativos em tempo real e vice-versa. Uma possível solução consiste em utilizar a imagem de fundo pré-renderizada como fonte de iluminação ambiente no sistema híbrido através da técnica de *Environment mapping* que, por sua vez, é baseada em *Image-based lighting*. Para tal, a parte da textura de fundo correspondente a um dos dois olhos (direito ou esquerdo) poderia ser convertida numa projeção cúbica (*cube map*) e ser usada como fonte de luz ambiente. Este é o método usado pelo Unity para efetuar o *Environment mapping*. Desta forma consegue-se facilmente iluminar os objetos interativos com a irradiação luminosa proveniente dos objetos pré-renderizados. A captura em HDR da imagem de fundo utilizada para efeito de iluminação pode ser uma boa opção para obter uma iluminação ainda mais realista dos objetos interativos. Aliás, como já foi previamente referido, esta técnica encontra-se frequentemente implementada nas unidades recentes de processamento gráfico e não deverá acarretar custos computacionais adicionais para o dispositivo móvel.

Uma outra sugestão baseia-se no uso de sondas de luz (*light probes*) colocadas em determinadas posições da cena virtual do sistema de pré-renderização. Assim, a informação de iluminação capturada por cada sonda seria exportada num formato específico e importada no sistema híbrido para ser aplicada localmente sobre os objetos 3D interativos em tempo real.

Os dois métodos sugeridos permitem iluminar os objetos interativos com a irradiação proveniente dos objetos pré-renderizados, mas não permite o processo inverso ficando esta lacuna como tema para futuro desenvolvimento.

Para resolver o problema da projeção das sombras dos objetos interativos sobre os objetos pré-renderizados utilizou-se uma funcionalidade existente no motor de jogo do Unity designada por *shadow projector*. Resumidamente, esta funcionalidade consiste em projetar um material característico de uma sombra sobre qualquer tipo de superfície geométrica. Se o projetor de sombra estiver colocado na mesma posição que uma fonte de luz e projetar na primeira superfície existente por trás de um objeto interativo, é possível simular de forma aproximada a sombra que seria criada pelo objeto interativo. Todavia, esta metodologia é aplicada de forma individual para cada objeto interativo e pode revelar-se incomportável nos casos em que existem muitos objetos interativos. Uma outra possibilidade passaria por modificar o *shader* usado pelo material aplicado aos *proxies* para que estes possam projetar e receber sombras pelo método de cálculo de sombras nativo do motor de jogo.

3.3 Geração dos meta-dados

Com foi possível verificar ao longo da descrição da arquitetura geral do sistema, existe um conjunto de meta-dados que deve ser gerado no sistema de pré-renderização para ser depois utilizado no sistema híbrido. Nesta secção pretende-se expor de uma forma mais detalhada as técnicas aplicadas para gerar os meta-dados e descrever alguns detalhes importantes dos formatos utilizados.

3.3.1 Pré-renderização e geração do vídeo de 360 por 180 graus

O conceito principal deste trabalho baseia-se na pré-renderização de cenas virtuais sob a forma de imagens estáticas no caso de o ambiente ser completamente estático ou sob a forma de um vídeo 360° no caso de o ambiente possuir elementos animados.

Seguindo a revisão bibliográfica apresentada no capítulo 2, decidiu-se gerar no sistema de pré-renderização, imagens panorâmicas do ambiente virtual através de capturas convertidas em imagens equiretangulares. O módulo responsável por capturar estas imagens foi criado utilizando o *asset* “360 Panorama Capture” disponível para o Unity que possui uma interface de programação simples de usar e permite especificar os parâmetros necessários à obtenção de uma ou várias imagens com as características pretendidas [54].

Para efetuar uma captura, é fundamental definir na cena virtual a posição a partir da qual se pretende obter a captura panorâmica. Tal pode ser realizado criando um objeto vazio na posição desejada. Assim, para preservar o máximo possível a qualidade visual original da cena capturada, especificou-se uma resolução de imagem de 8192 por 8192 *pixels* em formato PNG [55] e sem qualquer tipo de *anti-aliasing*. Para permitir uma captura estereoscópica, definiu-se uma distância inter-pupilar de 6,285 cm, sendo que este valor foi escolhido por ser a média exibida pela população norte americana determinada num estudo das forças armadas em 2012 [56].

A imagem obtida é a junção vertical de duas imagens distintas tendo cada uma a resolução de 8192 por 4098 *pixels*. A imagem superior corresponde ao olho esquerdo e a imagem inferior corresponde ao olho direito. Na figura 9 é possível ver um exemplo deste tipo de imagem.

Arquitetura e Implementação

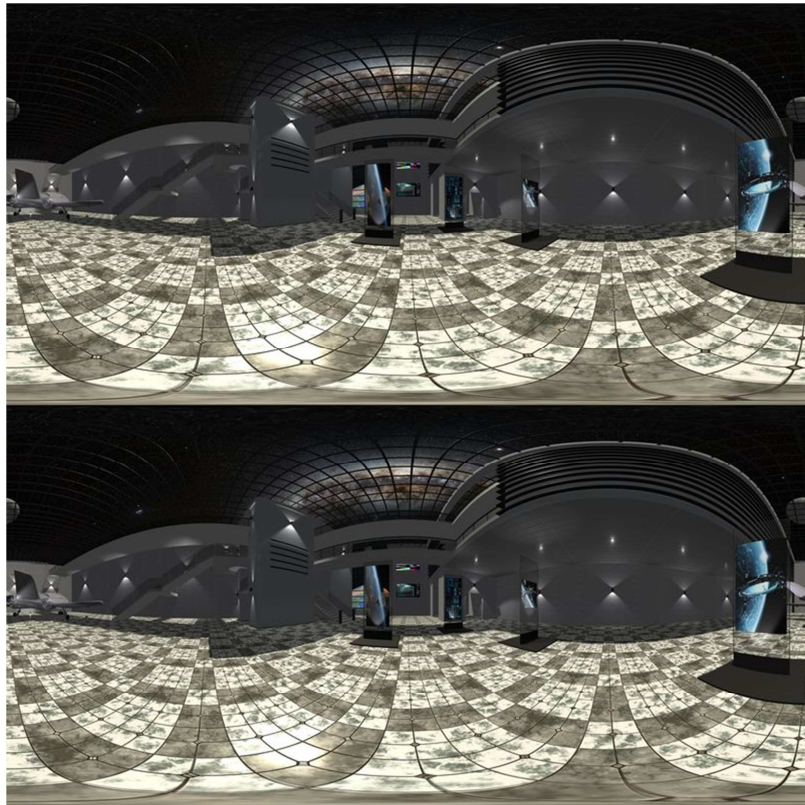


Figura 9: Exemplo de uma imagem estereoscópica equiretangular gerada pelo sistema de pré-renderização. A parte superior corresponde ao olho esquerdo e a inferior ao olho direito.

Se a cena virtual contiver elementos dinâmicos, é necessário gerar um vídeo panorâmico em vez de uma única imagem fixa. Neste caso deve-se possibilitar uma captura de imagens sequenciais e especificar o número de imagens a serem capturados por segundo. Estes parâmetros podem ser especificados no módulo de captura tendo-se escolhido uma velocidade de captura de 60 imagens por segundo. Após a captura da sequência de imagens em formato PNG é necessário convertê-las num ficheiro de vídeo MP4 (MPEG-4 Part 14) [57] e com compressão H.264 (MPEG-4 Part 10) [58, 59]. Para este efeito, foi usada uma instância do FFmpeg [60] numa interface de linha de comandos que permite esta conversão através da especificação dos parâmetros necessários para obter o ficheiro de vídeo com as características pretendidas.

No sistema híbrido, no caso de ser usado um vídeo pré-renderizado em vez de uma imagem fixa, foi adicionado um módulo responsável por decodificar cada imagem do vídeo e convertê-la numa textura mapeada na superfície interna das esferas de fundo. Para implementar o módulo foi usado o *asset* “Easy Movie Texture” desenvolvido para o Unity [61]. Apesar do Unity possuir um módulo de conversão de vídeo para textura (*Movie texture*) desde a versão 5.4, escolheu-se este *asset* por ser a única solução disponível durante a fase de desenvolvimento

Arquitetura e Implementação

que exibia bons resultados em dispositivos móveis. Contudo, a versão 2017 do Unity já oferece um leitor de vídeo (*Video Player*) funcional em dispositivos móveis.

Ao longo da fase de desenvolvimento foi possível verificar a existência de um desalinhamento visualmente perceptível entre os *proxies* e o vídeo de fundo. O desalinhamento é maior nas regiões polares das esferas de projeção do fundo e mantém-se mínimo na região equatorial. Além disso, o desalinhamento é inversamente proporcional à distância dos *proxies* relativamente ao ponto de visualização, isto é, quanto mais próximos os *proxies* estiverem do ponto de visualização, maior é o desalinhamento com o fundo.

A principal consequência do desalinhamento é a ocorrência de oclusões incompletas nas regiões polares de visualização para objetos relativamente próximos. Na figura 10 está exemplificado um exemplo de desalinhamento que ocorre entre os *proxies* e a textura de fundo para os dois olhos. Em cima é possível visualizar a região polar superior e em baixo, a zona equatorial. Os *proxies* estão representados em cinza e a textura de fundo nas restantes cores. As regiões coloridas visíveis representam os desalinhamentos. Considerando o exemplo da figura 10, a situação ideal seria visualizar apenas os *proxies* em cinza sobre o fundo branco que corresponderia a uma situação de alinhamentos exatos.

Os desalinhamentos devem-se provavelmente ao facto da metodologia usada para gerar a imagem panorâmica utilizar apenas um conjunto de câmaras situadas no plano equatorial, não permitindo manter uma disparidade binocular correta nas regiões polares [62, 63] e alterando a paralaxe exibida pelos objetos mais próximos.

Uma possível solução para este problema poderia passar por alterar a metodologia de obtenção da imagem panorâmica acrescentando câmaras de captura dirigidas para as regiões polares [63].

Apesar de este problema de desalinhamento ser visível com os *proxies* propositadamente desenhados (como estão na figura 10), relembra-se que os *proxies* são transparentes no sistema híbrido tornando este problema relativamente pouco perceptível.

Arquitetura e Implementação

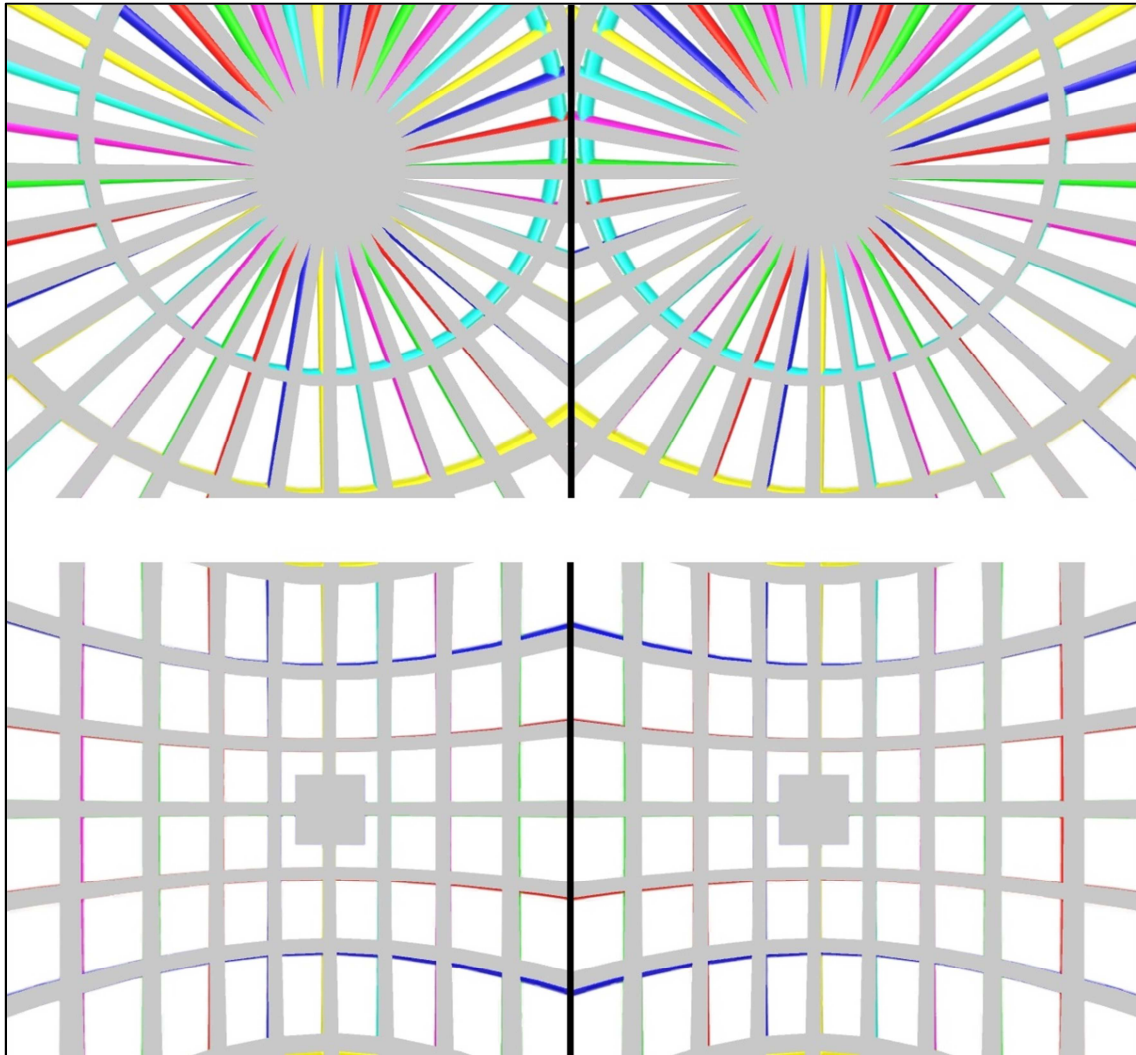


Figura 10: As regiões coloridas permitem destacar o desalinhamento que ocorre entre os *proxies* e a textura de fundo na região polar superior (em cima) e na região equatorial (em baixo).

Convém igualmente referir que as coordenadas usadas para mapear o vídeo panorâmico nas esferas de projeção podem originar uma distorção visualmente perceptível nos polos, devido à junção dos limites da imagem nestes pontos. Na figura 11 é possível visualizar a distorção da textura de fundo existente no ponto de junção do polo superior de uma das esferas.

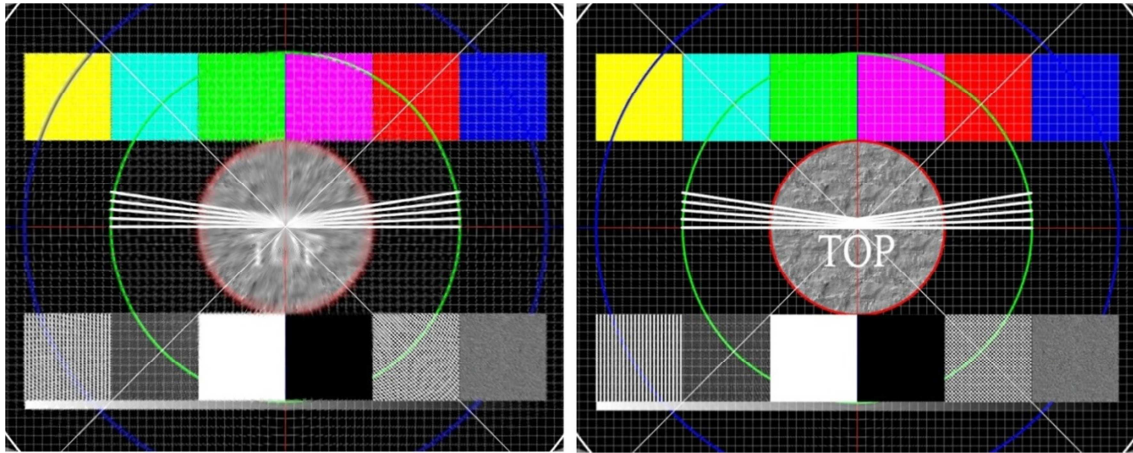


Figura 11: Distorção da textura de fundo existente no polo superior das esferas (à esquerda), em comparação com o original (à direita).

3.3.2 Geometrias

A geometria dos objetos presentes na cena virtual do sistema de pré-renderização é essencial para a geração dos *proxies* existentes no sistema híbrido. Esta informação deve ser preservada na migração entre os dois sistemas e isso só é possível exportando os vértices e arestas num formato adequado. Para tal, escolheu-se o formato FBX que permite guardar esta informação de uma forma estruturada e cuja importação é bem suportada por um amplo conjunto de editores de modelos tridimensionais. Para possibilitar a migração da geometria existente no sistema de pré-renderização para o sistema híbrido, foi incluído no primeiro sistema o módulo responsável pela exportação deste tipo de dados. Para desempenhar esta função usou-se a versão demo do *asset* “FBX Importer & Exporter for Unity β ” [62]. Com este módulo é possível gerar um ficheiro FBX com toda a geometria presente da cena virtual devidamente modelada e respeitando a estrutura hierárquica dos objetos. Não é necessário exportar os materiais dos objetos que originam os *proxies* nem as normais sendo apenas necessário as malhas geométricas.

A geometria exportada pode ser diretamente usada no sistema híbrido para criar os *proxies* no caso em que a cena virtual for relativamente simples no que respeita ao número total de triângulos. Contudo, para cenas complexas em que o número de triângulos for muito elevado, isso pode ser um fator limitativo para se conseguir atingir 60 imagens por segundo em dispositivos com desempenho menores. Como tal, e visto que não é necessária toda a geometria original para resolver os problemas de oclusões e colisões, poderá ser útil processar a geometria antes de ser importada no sistema híbrido, para reduzir o número de triângulos.

Arquitetura e Implementação

No sistema híbrido, o processo de importação de geometria em formato FBX é feito pelo método nativo incluído no ambiente de edição do Unity. Este gera novamente a geometria que é a base para a criação dos *proxies*. Os *proxies* utilizam as malhas geométricas importadas tanto para efeitos de oclusões como para efeitos de colisões. Contudo, as geometrias não necessitam de ser idênticas. As malhas usadas para colisões podem ser mais simples que as malhas usadas para as oclusões, ou o contrário, dependendo da precisão que se pretende obter em cada caso.

3.3.3 Animações

Para cenas virtuais em que se verifica a existência de objetos animados pré-renderizados, deve ser guardada a geometria bem como a animação associada a cada um desses objetos. No sistema de pré-renderização, o módulo de exportação de geometrias permite guardar as animações de cada objeto sob a forma de sequências das coordenadas de posições e rotações de cada objeto animado. No entanto, tal como na exportação da sequência de imagens do vídeo capturado, deve ser especificado o número de imagens por segundo a serem gravadas para as animações bem como a duração total da gravação. A versão demo do *asset* “FBX Importer & Exporter for Unity β” só permite exportar 150 imagens totalizando uma duração de 2,5 segundos, se for especificado 60 imagens por segundo, contudo, esta duração é suficiente para implementar os testes de avaliação deste trabalho. É importante referir que o número de imagens por segundo e a duração das animações devem ser idênticos ao das sequências de imagens do vídeo capturado para se conseguir uma boa sincronização entre os dois fluxos de dados.

Se o módulo de exportação da geometria for configurado para guardar animações, este grava as coordenadas dos objetos para cada instante capturado, mesmo para os objetos estáticos. Logo, se uma exportação contiver em simultâneo objetos dinâmicos e estáticos, o ficheiro gerado poderia ocupar muito espaço em disco devido a informação desnecessária ocupada pelos objetos estáticos. Para evitar essa situação, recomenda-se a exportação dos objetos estáticos sem animações separada da exportação dos objetos dinâmicos com animações.

Assim sendo, no sistema híbrido, o módulo de importação de modelos geométricos nativo do Unity trata de reconstruir a geometria dos objetos dinâmicos juntamente com a respetiva animação. Para reproduzir as animações, é adicionado ao grafo de cena um módulo de animação incluído no Unity e designado por “*animator*”. Estes objetos são depois convertidos em *proxies* dinâmicos, aplicando o material que possibilita as oclusões e adicionando as malhas ou primitivas de colisão. É essencial sincronizar as animações dos *proxies* animados com o fluxo de vídeo de fundo. Esta sincronização é feita em tempo de execução através de um script em C# que determina o número da imagem atual do vídeo fornecido pelo módulo de reprodução e aplica esse número de imagem ao módulo de reprodução das animações. Com esta metodologia evita-se erros acumulados de sincronização, todavia, as animações podem exibir um quadro de avanço ou de atraso em relação ao vídeo em determinados instantes, que dada a taxa de

Arquitetura e Implementação

refrescamento elevada, é muito pouco perceptível. Uma potencial causa destas flutuações será a descodificação assíncrona do vídeo na unidade de processamento gráfico que origina uma reprodução dessincronizada do vídeo em relação à animação.

3.3.4 Fontes de luz

Para que os objetos interativos presentes no sistema híbrido sejam corretamente iluminados conforme a iluminação dos objetos pré-renderizados, as fontes de luzes da cena virtual original devem ser exportadas pelo sistema de pré-renderização e reconstruídas no sistema híbrido. O módulo que trata da exportação das propriedades das fontes de luz existentes no sistema de pré-renderização está incluído no módulo de exportação de geometria fornecido pelo *asset* “FBX Importer & Exporter for Unity β”. O módulo transforma todas as propriedades das luzes em formato XML, das quais se destacam as mais importantes como o tipo de luz, cor, intensidade, posição e orientação. No sistema híbrido, um processador XML reconstrói as fontes de luzes com as propriedades e posições originais. Como os objetos interativos são objetos dinâmicos, estes só podem ser iluminados com fontes de luz cuja contribuição luminosa é calculada em tempo real. Como tal, durante o processo de importação, todas as fontes de luz devem ser marcadas como sendo luzes calculadas em tempo real (*real-time*). É muito importante salientar que o cálculo em tempo real da iluminação proveniente de fontes de luz é muito exigente para a unidade de processamento gráfico, sendo este facto agravado pelo número e tipo de luzes existentes no cenário. Como tal, recomenda-se uma utilização limitada de fontes de luz do tipo pontuais (*point lights*) e holofotes (*spot lights*), sendo preferível o uso de luzes direcionais.

3.4 Melhorias para aumento de desempenho

Como será possível ver nos resultados deste trabalho, a pré-renderização de um cenário virtual com a metodologia proposta permite um aumento significativo da qualidade visual e do desempenho em dispositivos com baixo poder de processamento. Contudo, o uso de *proxies* pode revelar-se uma desvantagem para situações em que o seu número ou complexidade geométrica for muito elevado. Para mitigar este problema são sugeridas duas metodologias descritas a seguir.

3.4.1 Simplificação da geometria

O número de triângulos que uma unidade de processamento gráfico pode processar por segundo é considerado como sendo uma métrica comum da capacidade de desempenho. Assim sendo, ao modelar um ambiente virtual deve-se ter um cuidado particular com o número de triângulos que constituem os objetos presentes na pirâmide de visualização da câmara (*view frustum*) em cada instante. Quanto maior for o número de triângulos, menor será o número de imagens por segundo e isso é particularmente notório em dispositivos móveis com baixo desempenho gráfico. Logo, se os *proxies* contiverem um número elevado de triângulos, isso pode-se refletir negativamente na fluidez de visualização do sistema híbrido.

Uma possibilidade para contornar a complexidade geométrica dos *proxies* consiste simplesmente em reduzir a quantidade de triângulos que a constitui mantendo a forma global de cada objeto. Isso pode ser conseguido através de um processo bem conhecido em modelação tridimensional designado por decimação [63]. Existem vários algoritmos utilizados para o processamento da malha e podem diferir entre si consoante a técnica utilizada. Uma das técnicas mais comuns consiste em colapsar vértices que se encontrem no mesmo plano ou planos cujo ângulo de intersecção seja suficientemente pequeno [64, 65]. Neste trabalho o processo de decimação foi aplicado sobre os objetos estáticos exportados pelo sistema de pré-renderização antes de serem importados no sistema híbrido usando a técnica do colapso de vértices coplanares. Contudo, poderia ser igualmente utilizado outro algoritmo de decimação ou até utilizar modelos geométricos que possuem vários níveis de detalhes (*level of details* - LOD) aproveitando o nível com a geometria mais simples.

Assim, é possível reduzir a complexidade das geometrias usadas para as oclusões e colisões. No entanto, este processo de simplificação pode reduzir a precisão das oclusões, fazendo com que objetos 3D interativos não sejam corretamente ocultados nas orlas dos *proxies*. O mesmo ocorre com a precisão das colisões, principalmente para objetos pequenos, fazendo com que um objeto 3D interativo possa seguir uma trajetória diferente da que deveria após a colisão com um *proxy*.

Arquitetura e Implementação

Com tal, devem ser tomadas em consideração estas limitações durante a especificação dos parâmetros do processo de decimação para se conseguir um justo equilíbrio entre a simplificação e o seu impacto na precisão das oclusões e colisões.

3.4.2 Sistema de *occlusion culling*

A técnica de *frustum culling* que consiste em não desenhar os objetos que estiverem fora da pirâmide de visualização da câmara permite um ganho significativo no desempenho de visualização de ambientes virtuais. Contudo, dentro do *view frustum* pode existir um número significativo de objetos que não necessitam de ser desenhados por serem ocultados por outros objetos resultando numa redução de desempenho proporcional a este número de objetos. O mesmo ocorre para os *proxies* existentes no sistema híbrido. Existe um método denominado *occlusion culling* que permite determinar objetos ocultados por outros fazendo com que só os objetos visíveis dentro do *view frustum* sejam realmente desenhados, melhorando assim aquela situação. O ambiente de edição do Unity permite aplicar nativamente este método através do algoritmo de “*cells and portals*” [67-69]. Contudo, este método não apresenta bons resultados em dispositivos móveis devido à exigência colocada na unidade de processamento central durante o tempo de execução.

Decidiu-se então implementar um sistema de *occlusion culling* que pré-determina os *proxies* visíveis aproveitando o facto de o utilizador não poder movimentar-se livremente com movimentos de translação no ambiente virtual. O utilizador só pode mudar o seu ponto de vista com a ajuda de pontos de teleporte previamente colocados na cena virtual. Assim sendo, a determinação dos *proxies* não ocultados por outros pode ser efetuada no sistema de pré-renderização para cada ponto de teleporte.

Em cada ponto de teleporte são lançados raios em várias direções. Quando um raio colide com um *proxy*, este último é guardado numa lista de *proxies* visíveis a partir do ponto de teleporte. Os *proxies* que não tiverem sido atingidos por um raio simplesmente não são guardados na lista. Quando o utilizador se encontrar num ponto de teleporte, só os *proxies* que estiverem na lista são desenhados ao invés dos restantes, fazendo com que só os *proxies* não ocultados por outros, isto é, só os *proxies* visíveis pelo utilizador, sejam realmente desenhados, resultando num aumento do desempenho do sistema híbrido para cenários com um número elevado de *proxies*.

Convém referir que existe a possibilidade de existir objetos que não sejam interceptados pelos raios lançados em cada ponto de teleporte, contudo, este facto pode ser mitigado diminuindo o ângulo entre cada raio aumentando assim o número de raios lançados.

Apesar dos *proxies* não-visíveis não serem desenhados, é importante realçar que a geometria de colisão destes é mantida ativa para poder processar colisões com objetos 3D interativos em tempo real.

Capítulo 4

Testes e Resultados

Neste capítulo pretende-se expor e analisar um conjunto de resultados obtidos através da submissão de um grupo de protótipos a várias avaliações quantitativas e qualitativas. Estes resultados e análises estão aqui apresentados para validar as questões de investigação destacadas, nomeadamente, as oclusões e colisões entre objetos pré-renderizados e objetos 3D interativos em tempo real bem como o desempenho do sistema.

4.1 Testes de desempenho

Para obter resultados e medidas concretas sobre a solução desenvolvida neste trabalho, recorreu-se à elaboração de um protótipo criado com o objetivo de medir o desempenho do sistema híbrido aqui desenvolvido. O protótipo descrito no ponto 4.1.1 foi produzido com base num ambiente virtual com uma complexidade geométrica e visual elevada. Os testes quantitativos descritos no ponto 4.1.2 pretendem medir o desempenho do sistema híbrido em quatro dispositivos móveis diferentes. Os resultados são expostos no ponto 4.1.3 e analisados no ponto 4.1.4.

4.1.1 Descrição do protótipo

O cenário desenvolvido para efetuar os testes quantitativos foi criado usando métodos normais de modelação tridimensional por adição de modelos geométricos já construídos e aplicação de materiais com boa qualidade visual. O cenário original é constituído por objetos que constituem uma geometria com um total de aproximadamente 642000 triângulos para renderização e aproximadamente 561000 triângulos para colisões. No que diz respeito à iluminação, existe uma luz direcional e duas luzes pontuais.

Testes e Resultados

É de referir igualmente a existência de oito pontos de teleporte a partir dos quais é possível ter pontos de vista diferentes sobre o cenário virtual. Na figura 12 é possível ter um ponto de vista sobre o ambiente virtual.

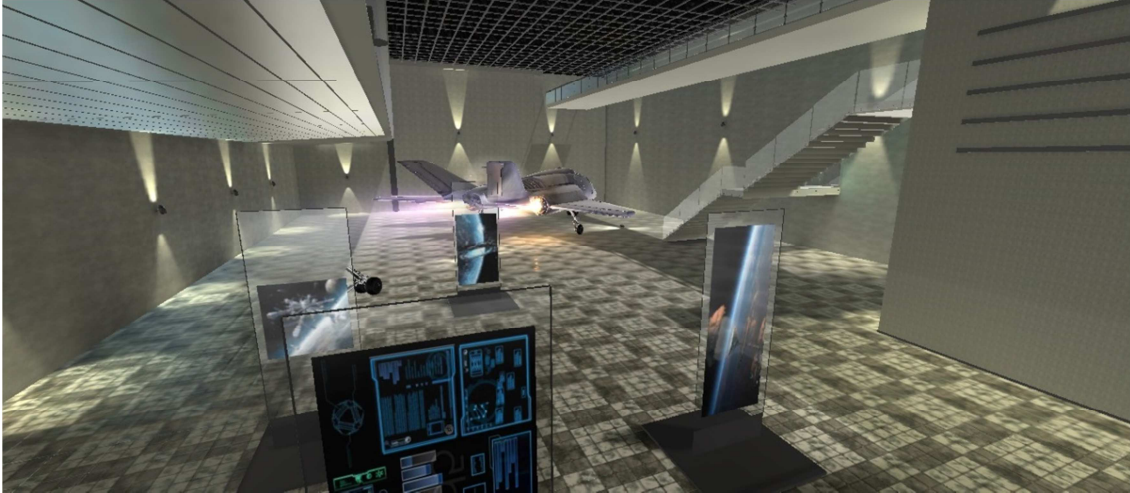


Figura 12: Exemplo de um ponto de vista do cenário usado para os testes quantitativos.

Para criar o equivalente deste cenário no sistema híbrido, foi utilizado o método descrito na secção 3.4.1 com as simplificações de geometria sugeridas, obtendo *proxies* com um total de aproximadamente 64000 triângulos para as oclusões e aproximadamente 55000 triângulos para as colisões. Assim, foi conseguida uma simplificação da geometria dos *proxies* com uma proporção de 10 para 1. Os objetos interativos em tempo real possuem 112 triângulos para desenho e colisão. As esferas usadas para desenhar a textura de fundo possuem 3968 triângulos. Salienta-se que estes valores de triângulos são duplicados quando visualizados em estereoscopia. Para a iluminação, foi reutilizada a luz direcional.

No sistema híbrido são igualmente importados os pontos de teleporte com os respetivos vídeos de fundo mas não é utilizado o sistema de *occlusion culling* descrito no ponto 3.4.2.

A título comparativo, é possível visualizar na figura 13 a cena original totalmente tridimensional, a imagem 360° pré-renderizada e o sistema híbrido com um objeto 3D interativo em tempo real.

Testes e Resultados

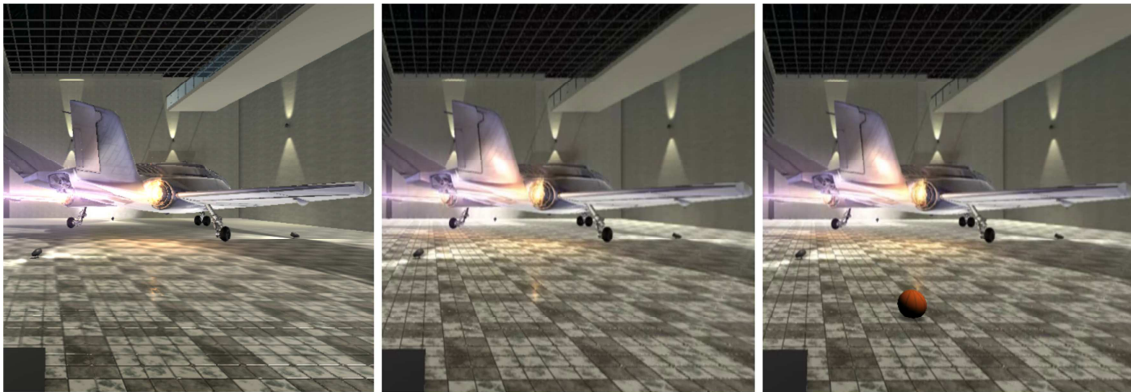


Figura 13: Cena original totalmente tridimensional (à esquerda), imagem 360° pré-renderizada (ao centro) e o sistema híbrido com um objeto 3D interativo em tempo real (à direita).

Ao analisar com cuidado as três imagens destacadas, é possível notar o efeito de escada (*aliasing*) na imagem da cena original 3D, particularmente na asa do avião e na textura do chão. Isso não acontece no sistema híbrido pelo facto da pré-renderização aplicar *anti-aliasing* à imagem gerada. Contudo, no sistema híbrido é possível notar a diferença de resolução entre o objeto 3D em tempo real (a bola) e a imagem de fundo.

4.1.2 Descrição dos testes

Para medir o desempenho da solução híbrida foram efetuadas três sequências de medições do número de imagens por segundo em quatro dispositivos. A primeira sequência foi dedicada à versão totalmente tridimensional do cenário. A segunda sequência foi para a solução híbrida que usa vídeo 360° como textura de fundo. A terceira sequência foi novamente para a solução híbrida mas com uma textura de fundo fixa.

Em cada versão é medido o número mínimo, máximo e médio de imagens por segundo para cada um dos oito pontos de teleporte presentes no ambiente virtual. O número de imagens médio é calculado dividindo as imagens acumuladas pelo tempo presente num determinado ponto de teleporte. Em cada ponto é efetuada uma rotação de 720° em torno do eixo vertical das câmaras de visão que corresponde a efetuar duas visualizações panorâmicas completas sucessivas. A rotação é feita de forma programática com uma velocidade angular constante de 12° por segundo para garantir que o movimento seja sempre igual.

Para testar a solução híbrida com várias resoluções de vídeo e imagem efetuou-se duas vezes a sequência de testes. A primeira vez foi com texturas panorâmicas fixas de 2048 por 2048 *pixels* e vídeos de 1750 por 874 *pixels* com 30 imagens por segundo por ser a configuração comum suportada por todos os dispositivos utilizados. A segunda vez foi com texturas panorâmicas fixas de 4096 por 4096 *pixels* e vídeos de 4096 por 2048 *pixels* com 60 imagens por segundo.

Testes e Resultados

Para as configurações da qualidade de renderização utilizadas, estas foram as definidas por omissão no modo *simple* do Unity.

Relativamente aos dispositivos móveis utilizados para efetuar estes testes, foram usados dois *smart phones* (Samsung Galaxy S4 e Lenovo ZUK Z2), um *tablet* (Cube I7 Remix) e uma consola de jogo portátil (GPD XD) com características diferentes para obter resultados comparativos entre os vários dispositivos. As características técnicas dos dispositivos encontram-se detalhadas na tabela 1.

Tabela 1: Características técnicas dos dispositivos utilizados para testes de desempenho.

Dispositivos utilizados para testes de desempenho						
Dispositivo	Circuito integrado (<i>System on Chip</i>)	Unidade de processamento central	Memoria dinâmica (<i>Giga bytes</i>)	Sistema operativo	Unidade de processamento gráfico	Resolução do ecrã (píxeis)
Samsung I9190 Galaxy S4 mini	Qualcom MSM8930AB Snapdragon 400	Dual-core 1,7 GHz Krait 300	1,5	Android 4.2.2	Adreno 305	540 × 960
Cube I7 Remix	Intel Atom Z3735F	Quad-core de 1,8 GHz	2	Remix OS 11.6	Intel HD Graphics	1080 × 1920
Gamepad GPD XD	Rockchip RK3288	Quad-core de 1,8 GHz	2	Android 4.4.4	Mali-T760	1280 × 720
Lenovo ZUK Z2	Qualcom MSM8996 Snapdragon 820	Quad-core (2 × 2,15 GHz e 2 × 1,6 GHz) Kryo	4	Android 6.0.1	Adreno 530	1080 × 1920

Testes e Resultados

4.1.3 Resultados

Nas tabelas 2 e 3 encontram-se os resultados obtidos através do método de teste descrito anteriormente. Os valores apresentados correspondem ao número médios de imagens sendo que os resultados completos estão apresentados no anexo A incluindo os números mínimos e máximos de imagens por segundo.

Tabela 2: Resultados de desempenho do sistema híbrido com vídeos 360° de 1750×874 *pixels* a 30 imagens/s e imagens fixas de 2048×2048 *versus* método totalmente 3D.

Pontos de teleporte	Imagens por segundo (valores médios)											
	Vídeo: 1750×874 @ 30 imagens/s - Imagem fixa: 2048×2048											
	Samsung S4 Mini			Cube i7 Remix			Gamepad GPD XD			Lenovo Zuk Z2		
	Híbrido 3D	Híbrido Vídeo	Híbrido Imagem fixa	Híbrido 3D	Híbrido Vídeo	Híbrido Imagem fixa	Híbrido 3D	Híbrido Vídeo	Híbrido Imagem fixa	Híbrido 3D	Híbrido Vídeo	Híbrido Imagem fixa
p01	11	40	52	17	45	53	15	42	60	56	60	60
p02	9	38	49	11	42	51	14	42	60	44	60	60
p03	8	37	50	10	44	49	13	42	60	41	60	60
p04	8	36	50	10	44	49	12	44	60	41	60	60
p05	10	38	52	11	41	44	15	43	60	53	60	60
p06	9	38	51	11	44	50	14	42	60	43	60	60
p07	9	39	50	11	42	51	13	40	60	46	60	60
p08	9	37	52	13	44	52	14	44	60	50	60	60
Média	9,13	37,88	50,75	11,75	43,25	49,88	13,75	42,38	60,00	46,75	60,00	60,00
Desvio Padrão	0,93	1,17	1,09	2,17	1,30	2,57	0,97	1,22	0,00	5,29	0,00	0,00
Aumento de imagens/s vs 3D		28,75	41,63		31,50	38,13		28,63	46,25		13,25	13,25
Aumento de desempenho vs 3D		4,15	5,56		3,68	4,24		3,08	4,36		1,28	1,28

Testes e Resultados

Tabela 3: Resultados de desempenho do sistema híbrido com vídeo 360° de 4096×2048 *pixels* a 60 imagens/s e imagens fixas de 4096×4096 *versus* método totalmente 3D.

Pontos de teleporte	Imagens por segundo (valores médios)											
	Vídeo: 4096×2048 @ 60 imagens/s - Imagem fixa: 4096×4096											
	Samsung S4 Mini			Cube i7 Remix			Gamepad GPD XD			Lenovo Zuk Z2		
	Híbrido 3D	Híbrido Vídeo	Híbrido Imagem fixa	Híbrido 3D	Híbrido Vídeo	Híbrido Imagem fixa	Híbrido 3D	Híbrido Vídeo	Híbrido Imagem fixa	Híbrido 3D	Híbrido Vídeo	Híbrido Imagem fixa
p01	11	-	50	17	17	54	15	44	60	56	60	60
p02	9	-	48	11	15	52	14	42	60	44	60	60
p03	8	-	52	10	14	50	13	43	60	41	60	60
p04	8	-	51	10	15	48	12	44	60	41	60	60
p05	10	-	47	11	15	43	15	42	60	53	60	60
p06	9	-	51	11	15	52	14	44	60	43	60	60
p07	9	-	50	11	15	51	13	42	60	46	60	60
p08	9	-	54	13	17	51	14	40	60	50	60	60
Média	9,13	-	50,38	11,75	15,38	50,13	13,75	42,63	60,00	46,75	60,00	60,00
Desvio Padrão	0,93	-	2,06	2,17	0,99	3,14	0,97	1,32	0,00	5,29	0,00	0,00
Aumento de imagens/s vs 3D		-	41,25		3,63	38,38		28,88	46,25		13,25	13,25
Aumento de desempenho vs 3D		-	5,52		1,31	4,27		3,10	4,36		1,28	1,28

O aumento de desempenho foi obtido pela fórmula de cálculo do *speedup*, que neste caso, consiste em dividir o valor obtido no sistema híbrido pelo valor obtido no método totalmente 3D. O resultado exprime um fator multiplicativo em relação a versão original. Por exemplo, para o Samsung S4 Mini, a solução híbrida com vídeo 360° de baixa resolução consegue um aumento de desempenho de 4,15 vezes relativamente ao original. Não foi possível obter resultados para a solução híbrida com vídeos 360° de 4096 por 2048 *pixels* a 60 imagens por segundo no Samsung S4 Mini devido ao facto de esta configuração de vídeo não ser suportada pelo decodificador dedicado do dispositivo.

Para uma melhor visualização, os resultados são apresentados nas figuras 14 e 15 sob a forma de gráficos.

Testes e Resultados

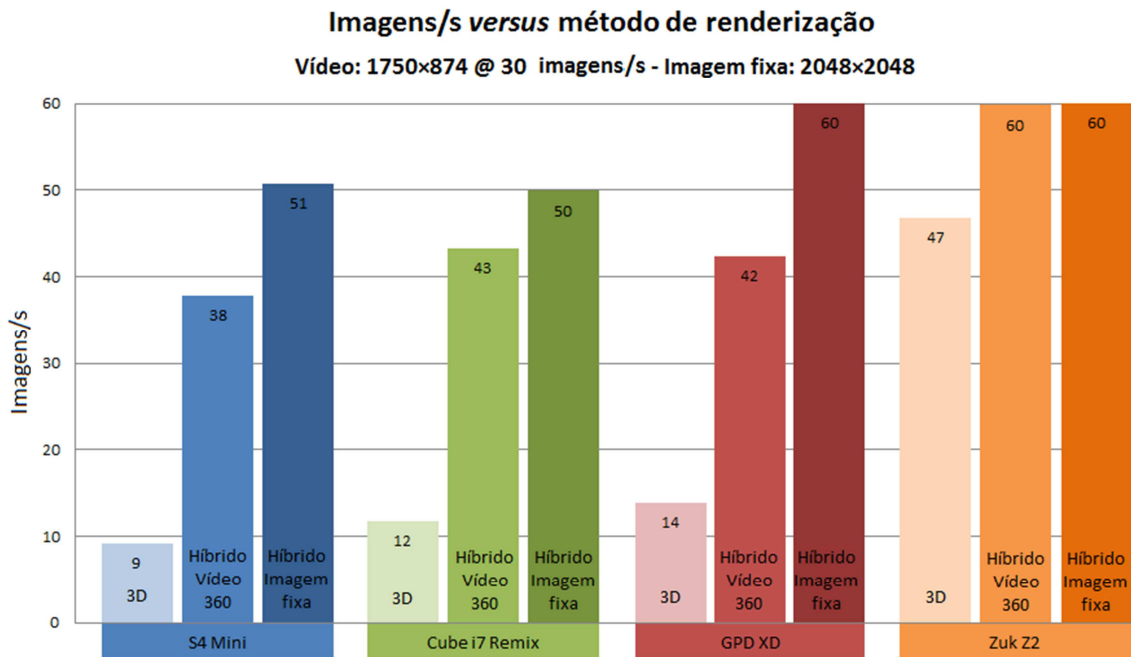


Figura 14: Número de imagens por segundo *versus* método de renderização para cada dispositivo. Vídeos 360° de 1750×874 *pixels* a 30 imagens/s e imagens fixas de 2048×2048.

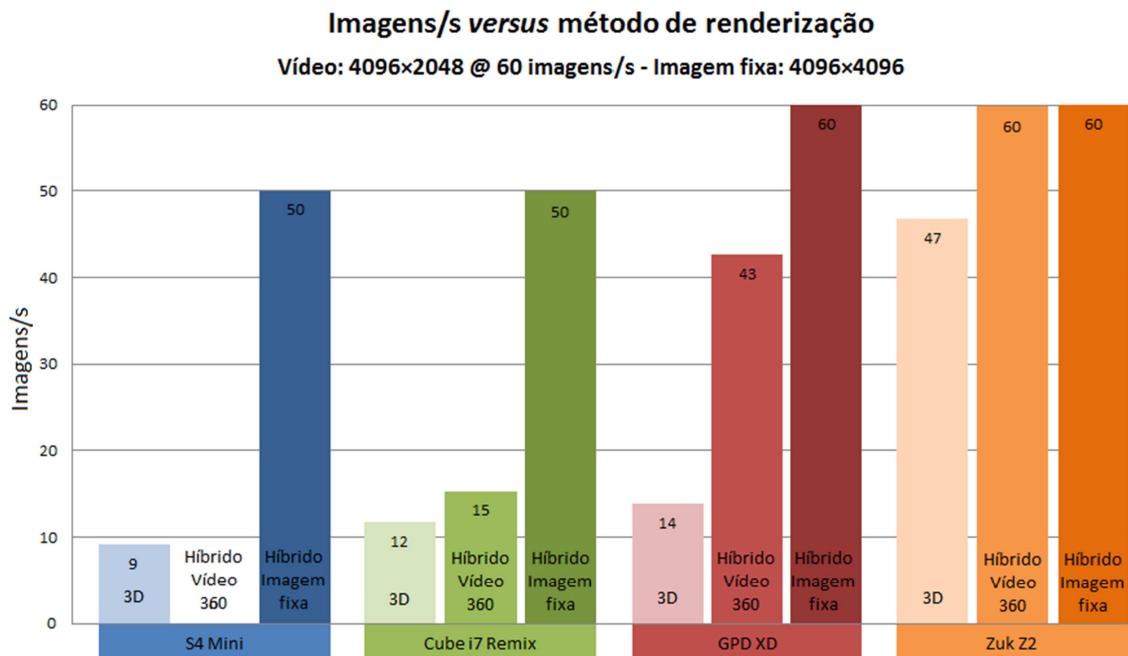


Figura 15: Número de imagens por segundo *versus* método de renderização para cada dispositivo. Vídeo 360° de 4096×2048 *pixels* a 60 imagens/s e imagens fixas de 4096×4096.

Testes e Resultados

4.1.4 Análise dos resultados

Observando as tabelas e os gráficos apresentados no ponto 4.1.3, verifica-se que o aumento de desempenho mínimo é de 1,28 e o máximo é de 4,15 para a versão híbrida com vídeo 360°. Para a versão com imagem fixa o mínimo é de 1,28 e máximo é de 5,56. A versão híbrida com vídeo 360° exibe menor desempenho do que na versão com imagem fixa e isso é explicado pelo facto do descodificador de vídeo da unidade de processamento gráfico ser requisitado e consumir portanto mais recursos do que a versão com imagem fixa.

É possível verificar, como seria de esperar, que aumentos da resolução e da frequência de imagens do vídeo 360° têm um impacto no desempenho do sistema híbrido. Esta observação é suportada pelos valores apresentados pelo Cube i7 Remix. No caso em que o vídeo possui uma alta resolução a 60 imagens/s, o desempenho é claramente menor do que com uma baixa resolução a 30 imagens/s. Isso é ainda mais notório para o Samsung S4 mini que simplesmente não suporta a maior resolução testada. Contudo, não se verificam diferenças no desempenho para dispositivos que suportam altas resoluções de vídeo como o GPD XD e o ZUK Z2.

Para imagens fixas é possível verificar que o aumento da resolução não provoca quebras de desempenho. Contudo, é importante referir que não foi possível desativar a sincronização vertical do refrescamento do ecrã dos dispositivos móveis que está usualmente limitado a 60 imagens por segundo. Como tal, os valores médios para a versão híbrida com imagem fixa no GPD XD e para as duas versões híbridas no ZUK Z2 ficaram limitados a 60 imagens por segundo. Se fosse possível desativar a sincronização vertical provavelmente verificar-se-ia um maior valor do aumento de desempenho nesses dispositivos.

É possível constatar que em qualquer um dos dispositivos utilizados, existe sempre um aumento do número de imagens por segundo para as duas versões do sistema híbrido sobre o método de renderização tradicional completamente 3D. Nomeadamente, a diferença entre o desempenho das versões híbridas e do método original é tanto maior quanto menor for o desempenho do dispositivo no método original 3D.

Resumindo, com a análise do resultados, é possível afirmar que o sistema de visualização híbrido apresenta, tanto com imagens de fundo fixas como com vídeo 360°, um aumento do desempenho perante o método completamente 3D.

Testes e Resultados

4.2 Testes qualitativos

Para obter resultados qualitativos, recorreu-se à elaboração de mais dois protótipos para além do apresentado nos testes de desempenho. Os dois novos protótipos foram produzidos com base num ambiente virtual que apresenta uma complexidade geométrica e visual simples. Os três protótipos foram submetidos a testes qualitativos em que 25 participantes voluntários foram convidados a imergir em versões diferentes dos três cenários. Os participantes responderam igualmente a um questionário elaborado com o intuito de determinar o grau de incoerências percebidas pelos utilizadores, pela utilização do sistema híbrido, mais particularmente, para verificar se os métodos de resolução dos problemas de oclusões e colisões introduzem inconsistências visuais negativas.

4.2.1 Descrição dos protótipos

A seguir estão apresentados os três protótipos elaborados para os testes qualitativos. Para simplificar a denominação, escolheu-se atribuir uma letra a cada um dos protótipos. Esta atribuição encontra-se na tabela 4. Cada protótipo existe em várias versões, nomeadamente, uma versão totalmente tridimensional (3D), uma versão para a solução híbrida descrita neste trabalho e uma versão só com vídeo 360° sem qualquer *proxy* ou objeto 3D em tempo real. Nota-se que o protótipo C não existe na versão só com vídeo 360° pois esta não permite interação com os participantes.

Tabela 4: Denominação das várias versões dos protótipos usadas para os testes qualitativos.

	Método de renderização		
	3D	Híbrido	360°
Cenário simples / Cena A	A1	A2	A3
Cenário complexo / Cena B	B1	B2	B3
Cenário interativo / Cena B	C1	C2	-

As versões totalmente tridimensional e só com vídeo 360° são utilizadas como padrões de referência para comparar com a solução híbrida.

Não existe nenhum ponto de teleporte nos cenários e como tal os participantes só podem visualizar os cenários a partir de um único ponto de vista.

Testes e Resultados

4.2.1.1 Protótipo A – cenário simples



Figura 16: Ponto de vista do cenário simples com trajetória do objeto interativo.

O cenário original deste protótipo (A1) possui objetos com aproximadamente 6900 triângulos para renderização e aproximadamente 5400 triângulos para colisões. Existem duas luzes pontuais para a iluminação. Na figura 16 é possível observar um ponto de vista sobre o ambiente virtual bem como a trajetória do objeto interativo (bola).

Os *shaders* utilizados são todos nativos do Unity e são do tipo *mobile* que permitem um melhor desempenho em dispositivos móveis.

Neste cenário os participantes devem seguir com atenção a trajetória de um objeto 3D interativo representado por uma bola que colide com um vaso de flores e acaba o seu percurso sobre uma cadeira. A bola possui uma primitiva de colisão esférica (*sphere collider*) e o vaso possui uma primitiva de colisão capsular (*capsule collider*). A cadeira é constituída por caixas de colisão (*box collider*) para produzir o assento e o encosto.

Para criar o equivalente deste cenário na versão híbrida (A2), os objetos foram importados sem simplificação da geometria originando *proxies* com o mesmo número de triângulos para as oclusões e colisões. Contudo, é importante referir que o vaso no qual colide a bola já não possui um *capsule collider* mas sim uma malha de colisão idêntica a malha original do modelo. Esta alteração foi voluntariamente introduzida para verificar se a alteração de uma primitiva de colisão numa malha de colisão introduz diferenças notórias. As restantes primitivas de colisão são mantidas. O objeto 3D interativo em tempo real possui 768 triângulos. As esferas usadas para desenhar a textura de fundo possuem 3968 triângulos. Para a iluminação, foi reutilizada uma única luz pontual.

Testes e Resultados

Relativamente a versão composta unicamente por vídeo 360° (A3), esta foi criada efetuando a pré-renderização da cena incluindo o objeto animado.

Este protótipo é utilizado para avaliar os métodos de resolução das oclusões e colisões sem aplicação do processo de simplificação da geometria da cena original.

4.2.1.2 Protótipo B - Cenário complexo

Este cenário (B1) já foi descrito nos testes de desempenho. Os participantes devem igualmente seguir com atenção a trajetória de um objeto interativo representado novamente por uma bola. O obstáculo onde a bola bate usa a sua malha geométrica como malha de colisão.

Na figura 17 é possível observar um ponto de vista sobre o ambiente virtual bem como a trajetória da bola.



Figura 17: Ponto de vista do cenário complexo com trajetória do objeto interativo.

A versão híbrida derivada deste cenário (B2) possui igualmente as características da versão equivalente dos testes de desempenho. Contudo foi acrescentada a geometria do objeto 3D interativo em tempo real com um total de 768 triângulos. O obstáculo onde a bola bate usa a sua malha geométrica devidamente simplificada como malha de colisão.

A versão composta unicamente por vídeo 360° (B3) foi criada efetuando a pré-renderização da cena incluindo o objeto animado.

Este protótipo é utilizado para avaliar os métodos de resolução das oclusões e colisões com aplicação do processo de simplificação da geometria da cena original.

Testes e Resultados

4.2.1.3 Protótipo C - Cenário simples com interatividade

Este ambiente é derivado do cenário simples descrito no ponto 4.2.1.1, contudo, uma grande parte dos objetos foram removidos para dar lugar a um tabuleiro de jogo inclinado sobre o qual os participantes largam uma bola no instante que quiserem pressionando um botão de um *gamepad* com *Bluetooth*. Na figura 18 é possível visualizar o tabuleiro de jogo presente neste cenário. Após o participante largar a bola, esta progride sobre o tabuleiro sob influência da gravidade. Durante o seu percurso a bola é forçada a colidir contra obstáculos dispostos a intervalos regulares sobre o tabuleiro. A bola acaba o seu percurso em locais com uma determinada pontuação. Visto que os locais onde colide a bola são de difícil determinação para os participantes, a bola adota uma trajetória imprevisível para os participantes não tendo forma de prever em que local a bola pode cair.

A cena completamente tridimensional (C1) possui 10000 triângulos para renderização, 3000 triângulos para colisões e duas fontes luz pontuais. Os obstáculos do tabuleiro possuem primitivas de colisão do tipo *capsule collider*.

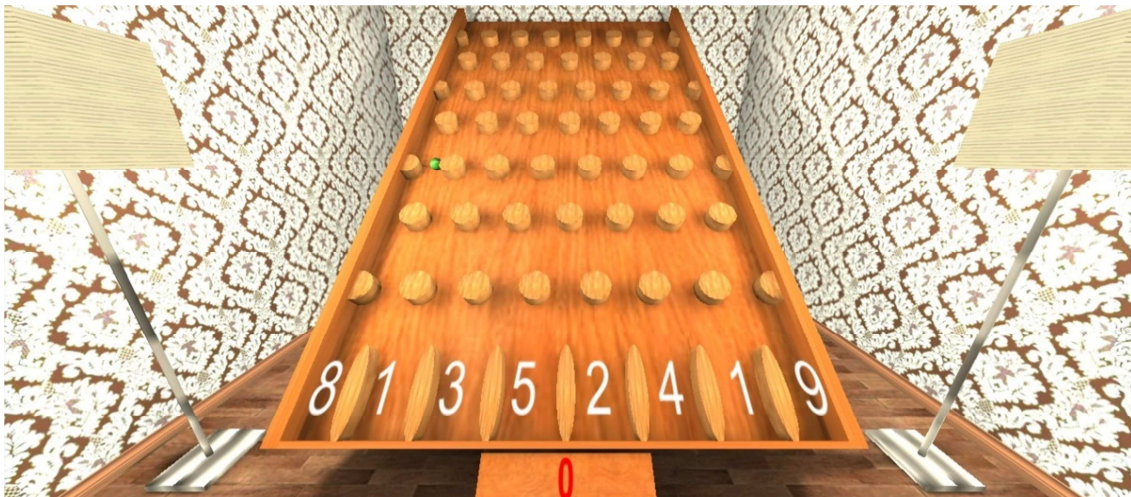


Figura 18: Ponto de vista do cenário simples com interação.

A versão híbrida foi construída aplicando uma simplificação da geometria originando *proxies* com 2600 triângulos de renderização, 1200 triângulos para colisões e duas fontes de luz pontuais. O objeto 3D interativo em tempo real possui 768 triângulos. As esferas usadas para desenhar a textura de fundo possuem 3968 triângulos. Nesta versão, apesar de os obstáculos poderem ser aproximados com *capsule colliders*, optou-se por usar *mesh colliders* a partir das malhas geométricas simplificadas para avaliar a diferença entre o uso de primitivas de colisão e malhas de colisão.

Realça-se o facto de o tabuleiro abranger a região próxima do polo inferior das esferas de projeção da textura de fundo. Nesta região o tabuleiro encontra-se bastante próximo do ponto de

Testes e Resultados

visualização. Como já foi referido no capítulo 3, nestas condições o efeito de oclusão por parte dos *proxies* é incompleto.

Assim, este protótipo pretende avaliar a validade dos métodos utilizados para resolver os problemas de oclusão e colisão tendo em consideração as limitações da tecnologia desenvolvida.

4.2.2 Descrição dos testes

Para obter resultados qualitativos sobre os métodos de resolução das oclusões e colisões, foram convidados 25 participantes voluntários. No início de cada sessão de testes, foi explicado a cada participante o contexto e objetivos dos testes. Cada participante assinou um acordo de consentimento informado cujo modelo se encontra no anexo B.

Foi pedido aos participantes que imergissem em cada uma das versões dos três cenários descritos anteriormente estando com particular atenção à trajetória do objeto interativo animado. Para tal, foi utilizado o *smartphone* Lenovo ZUK Z2 já descrito na tabela 1 e o HMD utilizado foi o Samsung Gear VR modelo SM-R322 com um campo de visão de 96° [2]. Apesar de o HMD utilizado só estar adaptado para *smartphones* do mesmo fabricante, verificou-se que o Lenovo ZUK Z2 possui dimensões adequadas possibilitando o seu uso neste HMD.

A sequência dos cenários bem como a sequência das versões apresentadas a cada participante foram previamente sorteadas por ordem aleatória. Assim, o primeiro cenário apresentado pode ser o protótipo A ou B. Se o primeiro for o A então o segundo será o B e vice-versa. Finalmente o último cenário apresentado aos participantes é sempre o protótipo C. Após cada versão de um cenário o participante responde a perguntas específicas para a versão em causa. Após todas as versões de um determinado cenário o voluntário responde a perguntas que pretendem comparar as versões de um mesmo cenário. Estas perguntas incidem principalmente sobre a perceção por parte dos participantes da oclusão do objeto 3D em tempo real, sua colisão com obstáculos e qualidade visual do cenário. Os utilizadores devem quantificar esses parâmetros numa escala de 0 a 5. Para as oclusões, o valor 0 corresponde ao objeto 3D não ter sido corretamente ocultado e 5 corresponde a uma ocultação correta. Para as colisões, o valor 0 corresponde ao objeto 3D não ter colidido corretamente com o obstáculo e 5 corresponde a uma colisão correta. Para a qualidade visual, o valor 0 corresponde ao cenário exibir uma fraca qualidade e 5 corresponde a uma muito boa qualidade visual. O questionário completo encontra-se apresentado no anexo C.

Todas as versões dos cenários foram visualizadas com 60 imagens por segundo à exceção da versão totalmente tridimensional do cenário complexo (B2) que foi visualizada com uma média de 50 imagens por segundo. Como textura de fundo foram utilizados vídeos 360° de 4096×2048 *pixels* a 60 imagens/s.

Testes e Resultados

4.2.3 Resultados

Para cada uma das versões dos três cenários foram obtidos resultados sobre a percepção dos utilizadores para as oclusões, colisões e qualidade visual de 0 a 5. Nas tabelas 5, 6 e 7 é possível ver um resumo dos resultados para cada uma das versões dos protótipos testados para oclusões, colisões e qualidade visual. Os resultados para as versões completamente tridimensionais estão destacados na coluna com o cabeçalho “3D”. Os resultados para as versões que utilizam a solução híbrida implementada neste trabalho estão destacados na coluna com o cabeçalho “Híbrido”. Os resultados para as versões que utilizam unicamente um vídeo 360° estão destacados na coluna com o cabeçalho “360°”.

Tabela 5: Resultados obtidos para as oclusões

	Oclusões								
	Cena A			Cena B			Cena C		
	3D	Híbrido	360°	3D	Híbrido	360°	3D	Híbrido	
Mediana	5	5	5	5	5	5	5	4	
Media	4,6	4,72	4,64	4,24	4,48	4,4	4,2	3,8	
Desvio Padrão	0,57	0,45	0,62	1,27	1,02	0,94	1,13	1,33	

Tabela 6: Resultados obtidos para as colisões

	Colisões								
	Cena A			Cena B			Cena C		
	3D	Híbrido	360°	3D	Híbrido	360°	3D	Híbrido	
Mediana	5	5	5	5	5	5	4	4	
Media	4,16	4,12	4,16	4,28	4,04	4,16	3,84	3,68	
Desvio Padrão	1,16	1,31	1,19	0,92	1,43	1,29	1,38	1,32	

Tabela 7: Resultados obtidos para a qualidade visual

	Qualidade visual								
	Cena A			Cena B			Cena C		
	3D	Híbrido	360°	3D	Híbrido	360°	3D	Híbrido	
Mediana	4	4	4	3	4	4	4	3	
Media	3,72	4,04	3,72	3,28	3,8	3,72	4,12	3,48	
Desvio Padrão	1	0,72	0,92	1,25	1,17	0,96	0,77	1,1	

Testes e Resultados

Analisando o valor das medianas apresentadas nas tabelas anteriores é possível constatar diferenças entre as versões de alguns cenários. Isso acontece para as oclusões na cena C e para a qualidade visual nas cenas B e C. Contudo, essas diferenças não são suficientes para afirmar que existe diferença significativa entre a versão híbrida e a versão completamente tridimensional.

Para verificar se existe realmente alguma diferença significativa entre os dois métodos foi realizada uma análise estatística aplicando o teste não-paramétrico de Friedman. A escolha deste teste é explicada pelo facto de se pretender comparar amostras relacionadas com escala ordinal. Neste caso concreto, compara-se os resultados de um mesmo parâmetro para versões diferentes. Por exemplo, comparar os valores obtidos para as oclusões na versão totalmente 3D e versão híbrida da cena A. É importante salientar que em cada teste não são emparelhadas as três versões porque se surgisse uma diferença significativa, não seria possível saber qual das versões deu origem a esta diferença. Assim, são comparadas apenas duas versões em cada teste de Friedman. Os resultados estatísticos foram calculados com a ajuda do SPSS e estão apresentados nas tabelas 8, 9 e 10. É importante referir que o teste de Friedman é baseado no teste do Qui-quadrado cujo valor foi igualmente calculado. O teste do Qui-quadrado necessita do número de graus de liberdade que é igual ao número de variáveis menos um. Como neste caso o número de variáveis é sempre igual a dois, o número de graus de liberdade é igual a um.

Tabela 8: Teste de Friedman para as oclusões

	Oclusões						
	Cena A			Cena B			Cena C
	Híbrido vs 3D	Híbrido vs 360°	360° vs 3D	Híbrido vs 3D	Híbrido vs 360°	360° vs 3D	Híbrido vs 3D
N	25	25	25	25	25	25	25
Graus de liberdade	1	1	1	1	1	1	1
Valor do Qui-Quadrado	3	0,333	1	1,286	1	0	1,6
<i>p</i>	0,083	0,564	0,317	0,257	0,317	1	0,206

Testes e Resultados

Tabela 9: Teste de Friedman para as colisões

	Colisões						
	Cena A			Cena B			Cena C
	Híbrido vs 3D	Híbrido vs 360°	360° vs 3D	Híbrido vs 3D	Híbrido vs 360°	360° vs 3D	Híbrido vs 3D
N	25	25	25	25	25	25	25
Graus de liberdade	1	1	1	1	1	1	1
Valor do Qui-Quadrado	0,5	0,111	0	0,11	0,333	0,4	0,286
<i>p</i>	0,48	0,739	1	0,739	0,564	0,527	0,593

Tabela 10: Teste de Friedman para a qualidade visual

	Qualidade Visual						
	Cena A			Cena B			Cena C
	Híbrido vs 3D	Híbrido vs 360°	360° vs 3D	Híbrido vs 3D	Híbrido vs 360°	360° vs 3D	Híbrido vs 3D
N	25	25	25	25	25	25	25
Graus de liberdade	1	1	1	1	1	1	1
Valor do Qui-Quadrado	1,923	1,923	0	3,267	0	1,333	4,263
<i>p</i>	0,166	0,166	1	0,71	1	0,248	0,039

4.2.4 Análise dos resultados

Para os testes estatísticos escolheu-se um nível de significância α igual a 0,05. Nas tabelas anteriores o valor de p corresponde à probabilidade das variáveis comparadas estarem diretamente relacionadas. Se o valor de p for igual ou superior ao valor de α , então é possível afirmar que não existe diferença significativa entre as duas variáveis comparadas. Se o valor de p for inferior ao valor de α então é possível afirmar que existe uma diferença significativa entre as duas variáveis e essa diferença não ser devida ao acaso.

Observando o valor de p para os testes de comparação das versões totalmente 3D e unicamente com vídeo 360° das cenas A e B, é possível notar que todos são superiores ao valor de α . Logo pode-se afirmar que não existem diferenças significativas entre as versões totalmente 3D e unicamente com vídeo 360°. Tal constatação é espectável pelo facto da versão só com vídeo 360° ter sido obtida pela pré-renderização da cena totalmente 3D incluindo o objeto interativo. Logo as oclusões e colisões são mantidas corretamente durante o processo de pré-renderização e percebidas como idênticas nas duas versões.

Testes e Resultados

As versões totalmente 3D e unicamente com vídeo 360° das cenas A e B são utilizadas como referência para comparação com as respectivas versões híbridas.

Nos testes de comparação das versões híbridas com as versões totalmente 3D e unicamente com vídeo 360° das cenas A e B, o valor de p é sempre superior ao valor de α . Logo para as cenas A e B não existem diferenças significativas entre a versão híbrida e as de referência.

Na cena A, na versão totalmente 3D é utilizado uma primitiva de colisão do tipo *capsule collider* enquanto que na versão híbrida é utilizado uma malha de colisão não simplificada. Logo é possível afirmar que o uso de malhas de colisão permite recriar colisões corretas na solução híbrida. Relativamente às oclusões, como seria de esperar, a utilização das malhas originais nos *proxies* permite a ocultação correta dos objetos 3D interativos em tempo real.

Para a cena B, é igualmente possível afirmar que o uso de malhas simplificadas nos *proxies* permite oclusões e colisões corretas.

Relativamente à qualidade visual da versão híbrida das cenas A e B não é possível afirmar que esta apresenta melhor qualidade em relação a versão original.

Analisando os resultados da cena C, verifica-se que não existem diferenças significativas tanto para as oclusões como para as colisões. Logo pode-se afirmar novamente que o uso de malhas simplificadas para os *proxies* permite criar oclusões e colisões corretas no sistema híbrido. Contudo existe uma diferença significativa na qualidade visual. Na tabela 6 verifica-se que a mediana da versão híbrida é inferior à versão 3D original. Assim, constata-se que os participantes atribuíram menor qualidade à versão híbrida. Não existem dados suficientes para dar uma explicação concreta para esta diferença. Contudo, no anexo D, analisando as respostas abertas dos participantes relativas às incoerências que acharam pertinentes em salientar para este cenário, é possível constatar que uma grande parte dos utilizadores reparou em incoerências existentes entre o objeto 3D em tempo real e o fundo quando o objeto se encontrava na região polar inferior. Como já foi destacado anteriormente, esta região não apresenta um alinhamento correto entre os *proxies* e o fundo. Pode então supor-se que este problema seja a causa da diferença da percepção da qualidade visual do cenário C.

Em resumo, foi possível verificar que não existem diferenças significativas na percepção das oclusões e colisões com e sem simplificação da geometria dos *proxies*. É então possível afirmar que os métodos usados para resolver os problemas de oclusões e colisões são válidos para serem aplicados no sistema de visualização híbrido.

Capítulo 5

Conclusões e Trabalho Futuro

Neste trabalho pretendeu-se elaborar uma solução que permita visualizar ambientes virtuais de alta qualidade visual em dispositivo móveis com baixa capacidade de processamento quando comparados com computadores de alto desempenho gráfico.

Para chegar a esse objetivo propôs-se um sistema híbrido que utiliza vídeos 360° pré-renderizados juntamente com *proxies* e objetos interativos em tempo real. Contudo, este sistema apresenta problemas de incoerência entre a textura de fundo e os objetos 3D de tempo real que pretendeu-se resolver, nomeadamente oclusões e colisões não processadas. Além disso pretendeu-se igualmente determinar o desempenho do sistema híbrido desenvolvido.

Para resolver os problemas de investigação relacionados com as oclusões e colisões entre os objetos 3D interativos em tempo real e a textura 360° de fundo foram criados três protótipos que implementam o sistema híbrido com as soluções descritas no capítulo 3.

Um dos protótipos foi usado em testes de desempenho que demonstraram que o sistema híbrido exibe um melhor desempenho, comparando com o sistema equivalente que utiliza o método de renderização comum completamente tridimensional. Salienta-se que esta constatação é válida tanto para imagens de fundo fixas como para vídeos de fundo em 360°. O aumento de desempenho permitiu atingir em alguns dispositivos uma visualização dos ambientes virtuais com uma taxa de refrescamento de 60 imagens por segundo quando essa taxa era menor para o método completamente tridimensional.

Relativamente à perceção das oclusões e das colisões por parte dos participantes voluntários, os testes estatísticos aplicados confirmaram que não existem diferenças significativas entre a solução híbrida e o método comum completamente tridimensional. Logo, é

Conclusões e Trabalho Futuro

possível afirmar que os métodos desenvolvidos e aplicados neste trabalho para resolver os problemas de oclusões e colisões são válidos.

Não foi possível demonstrar que o método híbrido permite fornecer maior qualidade visual que o método comum completamente tridimensional sem quebras no desempenho. Assim é necessário efetuar mais estudos nesse sentido.

Resumindo, é possível afirmar que os objetivos propostos foram maioritariamente atingidos. Conclui-se então, que a solução híbrida que usa objetos 3D interativos em tempo real sobrepostos a vídeo 360° pré-renderizado pode ser utilizada para visualizar e interagir com ambientes de realidade virtual em dispositivos móveis.

5.1 Trabalho Futuro

Apesar do sistema híbrido apresentar maior desempenho que o método tradicional, existem algumas limitações que não foram possíveis de ultrapassar neste estudo e deveriam ser sujeitas a mais investigação em trabalhos futuros.

As limitações mais importantes advêm das incoerências visuais notórias nas regiões polares das esferas de projeção da textura de fundo. Como já foi visto anteriormente a junção dos limites da textura nos polos das esferas provocam artefactos visíveis que podem ter impacto na qualidade visual. Para resolver este problema seria necessário determinar um método adequado para mapear a textura de fundo sobre as esferas ou alterar a geometria utilizada nas esferas sem provocar distorção nos polos.

O método de pré-renderização utilizado para gerar a textura de fundo dá origem a um desalinhamento entre os proxies e a textura de fundo nas regiões polares. Este efeito pode ter consequências negativas sobre a visualização das oclusões e colisões. Assim sendo, será útil explorar também métodos alternativos que permitam manter a disparidade binocular tanto na região equatorial como nas regiões polares, de forma a melhorar essa questão.

Ficam igualmente para investigações futuras os problemas de transparências, iluminação e sombra destacadas no capítulo 2, sobre os quais foram sugeridas algumas hipóteses de soluções mas que necessitam de ser devidamente estudadas, testadas e validadas.

Um próximo passo interessante na sequência do trabalho desenvolvido seria permitir movimentos de translação no mundo virtual atingindo assim seis graus de liberdade. Isso pode ser conseguido efetuando a pré-renderização em pontos separados por distâncias idênticas e aplicando técnicas de IBR para determinar em tempo real a imagem panorâmica correspondente ao ponto de visualização onde se encontra o utilizador.

Com a realização deste trabalho surge igualmente uma possível aplicação interessante no contexto da realidade aumentada. De facto, a metodologia que foi aqui descrita poderia ser adaptada para acrescentar objetos virtuais a imagens e vídeos 360° capturados do mundo real.

Referências

- [1] HTC. (2017-02-07). *Vive | Discover Virtual Reality Beyond Imagination*. Available: <https://www.vive.com/eu/>
- [2] Samsung. (2017-02-07). *Samsung Gear VR (2016) | Smartphone VR Headset Powered by Oculus*. Available: <http://www.samsung.com/uk/wearables/gear-vr-r323/>
- [3] F. Narducci, S. Ricciardi, and R. Vertucci, "Enabling consistent hand-based interaction in mixed reality by occlusions handling," *Multimedia Tools Appl.*, vol. 75, no. 16, pp. 9549-9562, 2016.
- [4] H. Regenbrecht, J. Collins, and S. Hoermann, "A leap-supported, hybrid AR interface approach," presented at the Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration, Adelaide, Australia, 2013.
- [5] K. Hayashi, H. Kato, and S. Nishida, "Occlusion detection of real objects using contour based stereo matching," presented at the Proceedings of the 2005 international conference on Augmented tele-existence, Christchurch, New Zealand, 2005.
- [6] M. M. Wloka and B. G. Anderson, "Resolving occlusion in augmented reality," presented at the Proceedings of the 1995 symposium on Interactive 3D graphics, Monterey, California, USA, 1995.
- [7] Microsoft. (2017-02-07). *Kinect for Xbox One*. Available: <http://www.xbox.com/en-GB/xbox-one/accessories/kinect>

Referências

- [8] A. Clark, T. Piumsomboon, and M. Billinghamurst, "Augmented reality micromachines," presented at the SIGGRAPH Asia 2011 Emerging Technologies, Hong Kong, China, 2011.
- [9] R. A. Newcombe *et al.*, "KinectFusion: Real-time dense surface mapping and tracking," presented at the Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality, 2011.
- [10] N. Greene, M. Kass, and G. Miller, "Hierarchical Z-buffer visibility," presented at the Proceedings of the 20th annual conference on Computer graphics and interactive techniques, Anaheim, CA, 1993.
- [11] Á. Montero, T. Zarranandia, P. Díaz, and I. Aedo, "Creating Interactive and Realistic Augmented Reality Experiences," presented at the Proceedings of the XVII International Conference on Human Computer Interaction, Salamanca, Spain, 2016.
- [12] J. Fischer, D. Bartz, and W. Straber, "Occlusion handling for medical augmented reality using a volumetric phantom model," presented at the Proceedings of the ACM symposium on Virtual reality software and technology, Hong Kong, 2004.
- [13] T. Porter and T. Duff, "Compositing digital images," *SIGGRAPH Comput. Graph.*, vol. 18, no. 3, pp. 253-259, 1984.
- [14] K. Gyllstrom, D. Miller, and D. Stotts, "Techniques for improving the visibility and "sharability" of semi-transparent video in shared workspaces," presented at the Proceedings of the 45th annual southeast regional conference, Winston-Salem, North Carolina, 2007.
- [15] D. Nowrouzezahrai, S. Geiger, K. Mitchell, R. Sumner, W. Jarosz, and M. Gross, "Light factorization for mixed-frequency shadows in augmented reality," in *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, 2011, pp. 173-179.
- [16] T. Grosch, T. Eble, and S. Mueller, "Consistent interactive augmentation of live camera images with correct near-field illumination," presented at the Proceedings of the 2007 ACM symposium on Virtual reality software and technology, Newport Beach, California, 2007.
- [17] N. Dachuri, S. M. Kim, and K. H. Lee, "Estimation of few light sources from environment maps for fast realistic rendering," presented at the Proceedings of the 2005 international conference on Augmented tele-existence, Christchurch, New Zealand, 2005.

Referências

- [18] P. Debevec, "Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography," presented at the Proceedings of the 25th annual conference on Computer graphics and interactive techniques, 1998.
- [19] C. Luksch, R. F. Tobler, R. Habel, M. Schwarzler, and M. Wimmer, "Fast light-map computation with virtual polygon lights," presented at the Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, Orlando, Florida, 2013.
- [20] D. A. Calian, K. Mitchell, D. Nowrouzezahrai, and J. Kautz, "The shading probe: fast appearance acquisition for mobile AR," presented at the SIGGRAPH Asia 2013 Technical Briefs, Hong Kong, Hong Kong, 2013.
- [21] M. Haller, S. Drab, and W. Hartmann, "A real-time shadow approach for an augmented reality application using shadow volumes," presented at the Proceedings of the ACM symposium on Virtual reality software and technology, Osaka, Japan, 2003.
- [22] F. Steinicke, K. Hinrichs, and T. Ropinski, "Virtual Reflections and Virtual Shadows in Mixed Reality Environments," in *Human-Computer Interaction - INTERACT 2005: IFIP TC13 International Conference, Rome, Italy, September 12-16, 2005. Proceedings*, M. F. Costabile and F. Paternò, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 1018-1021.
- [23] D. Lee, K. Park, and Y. Park, "Collision detection and response method for markerless camera-based games using motion boundary estimation," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 4, pp. 2178-2184, 2010.
- [24] D. Lee and Y. J. Lee, "Framework for vision-based sensory games using motion estimation and collision responses," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 3, pp. 1356-1363, 2010.
- [25] D. E. Breen, R. T. Whitaker, E. Rose, and M. Tuceryan, "Interactive Occlusion and Automatic Object Placement for Augmented Reality," *Computer Graphics Forum*, vol. 15, no. 3, pp. 11-22, 1996.
- [26] J. D. Foley, A. v. Dam, S. K. Feiner, and J. F. Hughes, *Computer graphics: principles and practice (2nd ed.)*. Addison-Wesley Longman Publishing Co., Inc., 1990, p. 1174.

Referências

- [27] W. A. McNeely, K. D. Puterbaugh, and J. J. Troy, "Advances in voxel-based 6-DOF haptic rendering," presented at the ACM SIGGRAPH 2005 Courses, Los Angeles, California, 2005.
- [28] K. Aitpayev and J. Gaber, "Collision Avatar (CA): Adding collision objects for human body in augmented reality using Kinect," in *2012 6th International Conference on Application of Information and Communication Technologies (AICT)*, 2012, pp. 1-4.
- [29] R. A. Farrugia, *Multimedia Networking and Coding*. Information Science Reference, 2012.
- [30] J. Bloomenthal, "Edge Inference with Applications to Antialiasing," *SIGGRAPH Comput. Graph.*, vol. 17, no. 3, pp. 157-162, 1983.
- [31] J. Fischer, "Real-virtual antialiasing," presented at the ACM SIGGRAPH 2008 posters, Los Angeles, California, 2008.
- [32] P. Bourke, "Spherical mirror: a new approach to hemispherical dome projection," presented at the Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia, Dunedin, New Zealand, 2005.
- [33] E. Lantz, "A survey of large-scale immersive displays," presented at the Proceedings of the 2007 workshop on Emerging displays technologies: images and beyond: the future of displays and interacton, San Diego, California, USA, 2007.
- [34] J. Ardouin, A. Lécuyer, M. Marchal, and E. Marchand, "Navigating in virtual environments with 360° omnidirectional rendering," in *2013 IEEE Symposium on 3D User Interfaces (3DUI)*, 2013, pp. 95-98.
- [35] GOG. (2017-02-07). *7th Guest, The in GOG.com*. Available: https://www.gog.com/game/the_7th_guest
- [36] fandom. (2017-02-07). *Resident Evil | Resident Evil Wiki | Fandom powered by Wikia*. Available: http://residentevil.wikia.com/wiki/Resident_Evil
- [37] J. Brosz and F. Samavati, "Shape Defined Panoramas," in *2010 Shape Modeling International Conference*, 2010, pp. 57-67.
- [38] Renderstuff. (2017-02-07). *Creating virtual 360 Panorama*. Available: <http://renderstuff.com/creating-virtual-360-panorama-cg-tutorial/>

Referências

- [39] C. Cruz-Neira, D. J. Sandin, T. A. DeFanti, R. V. Kenyon, and J. C. Hart, "The CAVE: audio visual experience automatic virtual environment," *Commun. ACM*, vol. 35, no. 6, pp. 64-72, 1992.
- [40] M. Abrash. (2017-05-12). *What VR Could, Should, and almost certainly Will be within two years.* Available: <http://3A%2F%2Fmedia.steampowered.com%2Fapps%2Fabrashblog%2Fabrash%2520Dev%2520Days%25202014.pdf>
- [41] M. Bajura, H. Fuchs, and R. Ohbuchi, "Merging virtual objects with the real world: seeing ultrasound imagery within the patient," *SIGGRAPH Comput. Graph.*, vol. 26, no. 2, pp. 203-210, 1992.
- [42] J. Liang, C. Shaw, and M. Green, "On temporal-spatial realism in the virtual reality environment," presented at the Proceedings of the 4th annual ACM symposium on User interface software and technology, Hilton Head, South Carolina, USA, 1991.
- [43] R. S. Allison, L. R. Harris, M. Jenkin, U. Jasiobedzka, and J. E. Zacher, "Tolerance of Temporal Delay in Virtual Environments," presented at the Proceedings of the Virtual Reality 2001 Conference (VR'01), 2001.
- [44] D. Chu, E. Cuervo, and K. Boos, "FlashBack: Immersive Virtual Reality on Mobile Devices via Rendering Memoization," presented at the MobiSys 2016, Singapore, Singapore, 2016.
- [45] SquareEnix. (2017-02-07). *FINAL FANTASY VII.* Available: <http://www.finalfantasyviipc.com/en>
- [46] Google. (2017-02-07). *Street View - Google Developers.* Available: <https://developers.google.com/streetview/>
- [47] F. Okura, M. Kanbara, and N. Yokoya, "Mixed-Reality World Exploration Using Image-Based Rendering," *J. Comput. Cult. Herit.*, vol. 8, no. 2, pp. 1-26, 2015.
- [48] J. Nafarrete. (2017-05-20). *First Impressions With Google's WorldSense; ILMxLAB VR Experience.* Available: <https://vrscout.com/news/google-worldsense-seurat-ilmxlab-vr/>
- [49] U. Technologies. (2017-03-11). *Unity Game Engine.* Available: <https://unity3d.com>
- [50] Autodesk. (2017-05-01). *FBX Overview.* Available: <https://www.autodesk.com/products/fbx/overview>

Referências

- [51] P. S. U. Shankar, N. Raveendranathan, N. R. Gans, and R. Jafari, "Towards power optimized kalman filter for gait assessment using wearable sensors," presented at the Wireless Health 2010, San Diego, California, 2010.
- [52] K. Nagarajan, N. Gans, and R. Jafari, "Modeling human gait using a Kalman filter to measure walking distance," presented at the Proceedings of the 2nd Conference on Wireless Health, San Diego, California, 2011.
- [53] Google. (2017-05-01). *Google VR SDK for Unity*. Available: <https://developers.google.com/vr/unity/>
- [54] eVRydayVR. (2017-02-15). *360 Panorama Capture*. Available: <https://www.assetstore.unity3d.com/en/#!/content/38755>
- [55] J. Miano, *Compressed image file formats: JPEG, PNG, GIF, XBM, BMP*. ACM Press/Addison-Wesley Publishing Co., 1999, p. 264.
- [56] U. S. Army. (2017-05-01). *2012 ANTHROPOMETRIC SURVEY OF U.S. ARMY PERSONNEL: METHODS AND SUMMARY STATISTICS*. Available: <http://www.dtic.mil/get-tr-doc/pdf?AD=ADA611869>
- [57] IETF. (2017-05-01). *MIME Type Registration for MPEG-4*. Available: <https://tools.ietf.org/html/rfc4337>
- [58] N. S. Narkhede and N. Kant, "The emerging H.264/advanced video coding standard and its applications," presented at the Proceedings of the International Conference on Advances in Computing, Communication and Control, Mumbai, India, 2009.
- [59] ITU. (2017-05-01). *H.264 : Advanced video coding for generic audiovisual services*. Available: <https://www.itu.int/rec/T-REC-H.264>
- [60] FFmpeg. (2017-05-01). *FFmpeg*. Available: <https://ffmpeg.org/>
- [61] JaeYunLee. (2017-05-01). *Easy Movie Texture*. Available: <https://www.assetstore.unity3d.com/en/#!/content/10032>
- [62] ACTINIA_Software. (2017-03-06). *FBX Importer & Exporter for Unity β*. Available: <https://www.assetstore.unity3d.com/en/#!/content/42568>

Referências

- [63] C. DeCoro and N. Tatarchuk, "Real-time mesh simplification using the GPU," presented at the Proceedings of the 2007 symposium on Interactive 3D graphics and games, Seattle, Washington, 2007.
- [64] J. Du, X. Guo, S. Lu, X. Xiao, and W. Wen, "Automatic mesh split-and-merge technique for multiple surface models," presented at the Proceedings of the 11th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry, Singapore, Singapore, 2012.
- [65] Y. Li, E. Zhang, Y. Kobayashi, and P. Wonka, "Editing operations for irregular vertices in triangle meshes," *ACM Trans. Graph.*, vol. 29, no. 6, pp. 1-12, 2010.
- [66] Blender.org. (2017-05-01). *Blender*. Available: <https://www.blender.org/>
- [67] U. Technologies. (2017-05-01). *Occlusion Culling*. Available: <https://docs.unity3d.com/356/Documentation/Manual/OcclusionCulling.html>
- [68] D. Luebke and C. Georges, "Portals and mirrors: simple, fast evaluation of potentially visible sets," presented at the Proceedings of the 1995 symposium on Interactive 3D graphics, Monterey, California, USA, 1995.
- [69] T. Roden and I. Parberry, "Portholes and planes: faster dynamic evaluation of potentially visible sets," *Comput. Entertain.*, vol. 3, no. 2, pp. 3-3, 2005.

Anexo A

Resultados de desempenho

7.1 Resultados de desempenho: Imagens/s com Vídeo: 1750×874 @ 30 imagens/s - Imagem fixa: 2048×2048

Pontos de teleporte	Samsung S4 Mini									Cube i7 Remix								
	3D			Híbrido			Híbrido			3D			Híbrido			Híbrido		
	Min	Avr	Max	Vídeo			Imagem fixa			Min	Avr	Max	Vídeo			Imagem fixa		
p01	4	11	16	8	40	56	28	52	60	8	17	28	28	45	56	36	53	60
p02	4	9	16	24	38	52	40	49	60	8	11	16	32	42	56	32	51	60
p03	4	8	16	28	37	44	40	50	60	8	10	16	32	44	52	36	49	56
p04	4	8	16	28	36	48	40	50	60	8	10	16	32	44	52	40	49	60
p05	4	10	16	28	38	52	44	52	60	8	11	20	28	41	52	36	44	56
p06	4	9	16	28	38	48	40	51	60	8	11	16	32	44	52	40	50	60
p07	4	9	16	32	39	48	40	50	60	8	11	16	32	42	52	40	51	60
p08	4	9	16	12	37	52	36	52	60	8	13	20	32	44	52	32	52	60
Média	4	9,13	16	23,5	37,88	50	38,5	50,75	60	8	11,75	18,5	31	43,25	53	36,5	49,88	59
Desvio Padrão	0	0,93	0	8,11	1,17	3,46	4,44	1,09	0	0	2,17	3,97	1,73	1,3	1,73	3,12	2,57	1,73
Aumento de imagens/s vs 3D				19,5	28,75	34	34,5	41,63	44				23	31,5	34,5	28,5	38,13	40,5
Aumento de desempenho vs 3D				5,88	4,15	3,13	9,63	5,56	3,75				3,88	3,68	2,86	4,56	4,24	3,19

Resultados de desempenho

Pontos de teleporte	Gamepad GPD XD									Lenovo Zuk Z2								
	3D			Híbrido			Híbrido			3D			Híbrido			Híbrido		
	Min	Avr	Max	Video			Imagem fixa			Min	Avr	Max	Video			Imagem fixa		
p01	8	15	20	24	42	60	52	60	60	32	56	60	56	60	60	56	60	60
p02	8	14	20	28	42	60	56	60	60	32	44	60	52	60	60	56	60	60
p03	8	13	20	28	42	60	56	60	60	28	41	60	56	60	60	56	60	60
p04	8	12	20	24	44	54	52	60	60	32	41	60	56	60	60	56	60	60
p05	8	15	20	28	43	60	56	60	60	36	53	54	52	60	60	56	60	60
p06	8	14	20	32	42	56	56	60	60	28	43	60	52	60	60	56	60	60
p07	12	13	20	28	40	52	56	60	60	36	46	60	52	60	60	56	60	60
p08	8	14	20	24	44	60	56	60	60	28	50	60	56	60	60	56	60	60
Média	8,5	13,75	20	27	42,38	57,75	55	60	60	31,5	46,75	59,25	54	60	60	56	60	60
Desvio Padrão	1,32	0,97	0	2,65	1,22	3,07	1,73	0	0	3,12	5,29	1,98	2	0	0	0	0	0
Aumento de imagens/s vs 3D				18,5	28,63	37,75	46,5	46,25	40				22,5	13,25	0,75	24,5	13,25	0,75
Aumento de desempenho vs 3D				3,18	3,08	2,89	6,47	4,36	3				1,71	1,28	1,01	1,78	1,28	1,01

Resultados de desempenho

7.2 Resultados de desempenho: Imagens/s com Vídeo: 4096×2048 @ 60 imagens/s - Imagem fixa: 4096×4096

Pontos de teleporte	Samsung S4 Mini									Cube i7 Remix								
	3D			Híbrido			Híbrido			3D			Híbrido			Híbrido		
	Min	Med	Max	Vídeo			Imagem fixa			Min	Med	Max	Vídeo			Imagem fixa		
p01	4	11	16	-	-	-	30	50	60	8	17	28	8	17	24	34	54	60
p02	4	9	16	-	-	-	44	48	60	8	11	16	8	15	24	32	52	60
p03	4	8	16	-	-	-	44	52	60	8	10	16	8	14	24	37	50	60
p04	4	8	16	-	-	-	40	51	60	8	10	16	8	15	28	38	48	60
p05	4	10	16	-	-	-	39	47	60	8	11	20	8	15	24	36	43	60
p06	4	9	16	-	-	-	38	51	60	8	11	16	8	15	24	40	52	60
p07	4	9	16	-	-	-	40	50	60	8	11	16	8	15	20	40	51	60
p08	4	9	16	-	-	-	39	54	60	8	13	20	8	17	24	31	51	60
Média	4	9,13	16	-	-	-	39,3	50,38	60	8	11,75	18,5	8	15,38	24	36	50,13	60
Desvio Padrão	0	0,93	0	-	-	-	4,09	2,06	0	0	2,17	3,97	0	0,99	2	3,20	3,14	0
Aumento de imagens/s vs 3D				-	-	-	35,3	41,25	44				0	3,63	5,5	28	38,38	41,5
Aumento de desempenho vs 3D				-	-	-	9,81	5,52	3,75				1	1,31	1,3	4,5	4,27	3,24

Pontos de teleporte	Gamepad GPD XD									Lenovo Zuk Z2								
	3D			Híbrido			Híbrido			3D			Híbrido			Híbrido		
	Min	Med	Max	Vídeo			Imagem fixa			Min	Med	Max	Vídeo			Imagem fixa		
p01	8	15	20	26	44	60	54	60	60	32	56	60	54	60	60	56	60	60
p02	8	14	20	28	42	60	56	60	60	32	44	60	51	60	60	56	60	60
p03	8	13	20	26	43	60	52	60	60	28	41	60	56	60	60	56	60	60
p04	8	12	20	24	44	58	52	60	60	32	41	60	55	60	60	56	60	60
p05	8	15	20	28	42	58	56	60	60	36	53	54	52	60	60	56	60	60
p06	8	14	20	28	44	60	52	60	60	28	43	60	54	60	60	56	60	60
p07	12	13	20	28	42	58	56	60	60	36	46	60	52	60	60	56	60	60
p08	8	14	20	28	40	60	56	60	60	28	50	60	54	60	60	56	60	60
Média	8,5	13,75	20	27	42,63	59,25	54,25	60	60	31,5	46,75	59,25	53,5	60	60	56	60	60
Desvio Padrão	1,32	0,97	0	1,41	1,32	0,97	1,85	0	0	3,12	5,29	1,98	1,58	0	0	0	0	0
Aumento de imagens/s vs 3D				18,5	28,88	39,25	45,75	46,25	40				22	13	0,75	24,5	13,25	0,75
Aumento de desempenho vs 3D				3,18	3,1	2,96	6,38	4,36	3				1,7	1,28	1,01	1,78	1,28	1,01

Anexo B

Declaração de consentimento informado apresentado aos participantes voluntários

8.1 Declaração de consentimento informado apresentado aos participantes voluntários

Declaração de consentimento informado apresentado aos participantes voluntários

CONSENTIMENTO INFORMADO, LIVRE E ESCLARECIDO PARA PARTICIPAÇÃO EM INVESTIGAÇÃO

de acordo com a Declaração de Helsínquia¹ e a Convenção de Oviedo²

Por favor, leia com atenção a seguinte informação. Se achar que algo está incorrecto ou que não está claro, não hesite em solicitar mais informações. Se concorda com a proposta que lhe foi feita, queira assinar este documento.

Título do estudo: Sistema Híbrido Baseado em Geometria e Vídeo 360 3D para Utilização em Realidade Virtual (provisório)

Enquadramento: Tese de Mestrado Integrado em Engenharia Informática e Computação na Faculdade de Engenharia da Universidade do Porto, sob orientação do Doutor Rui Pedro Amaral Rodrigues e Hugo Miguel Gonçalves Crespo Machado da Silva.

Explicação do estudo: O estudo consiste na criação de uma nova tecnologia que envolve o uso de vídeos 360 juntamente com geometria 3D com o objectivo de promover ambientes de realidade virtual com alta qualidade visual em dispositivos móveis de baixo desempenho.

Condições: A participação na investigação a decorrer é de carácter voluntário. O(s) participante(s) deste estudo podem decidir não responder a qualquer questão ou dar por terminada a participação a qualquer momento, sem que isto resulte em qualquer prejuízo ou consequência.

Confidencialidade e anonimato: A utilização dos dados recolhidos resulta exclusivamente em informação compilada no estudo indicado. Não são registados quaisquer dados de identificação relativos aos envolvidos, sendo que a colaboração destes permanecerá anónima e confidencial.

Apresento os meus sinceros agradecimentos pela sua participação.

Leonel Jorge Nogueira Peixoto
Aluno do Mestrado Integrado em Engenharia Informática e Computação
Faculdade de Engenharia da Universidade do Porto
Rua Dr. Roberto Frias, S/N
4200-465 Porto
ei12178@fe.up.pt

Assinatura/s:

.....

-o-o-o-o-o-o-o-o-o-o-o-o-o-o-o-o-

Declaro ter lido e compreendido este documento, bem como as informações verbais que me foram fornecidas pela/s pessoa/s que acima assina/m. Foi-me garantida a possibilidade de, em qualquer altura, recusar participar neste estudo sem qualquer tipo de consequências. Desta forma, aceito participar neste estudo e permito a utilização dos dados que de forma voluntária forneço, confiando em que apenas serão utilizados para esta investigação e nas garantias de confidencialidade e anonimato que me são dadas pelo/a investigador/a.

Nome:

Assinatura:

Data: /..... /.....

<p>SE NÃO FOR O PRÓPRIO A ASSINAR POR IDADE OU INCAPACIDADE (se o menor tiver discernimento deve também assinar em cima, se consentir)</p> <p>NOME: BI/CD Nº: DATA ou VALIDADE /..... /..... GRAU DE PARENTESCO OU TIPO DE REPRESENTAÇÃO: ASSINATURA</p>

**ESTE DOCUMENTO É COMPOSTO DE 1 PÁGINA/S E FEITO EM DUPLICADO:
UMA VIA PARA O/A INVESTIGADOR/A, OUTRA PARA A PESSOA QUE CONSENTE**

¹ http://portal.arsnorte.min-saude.pt/portal/page/portal/ARSNorte/Comiss%C3%A3o%20de%20%C3%89tica/Ficheiros/Declaracao_Helsinquia_2008.pdf

² <http://dre.pt/pdf1sdip/2001/01/002A00/00140036.pdf>

Anexo C

Questionário apresentado aos participantes voluntários

9.1 Questionário apresentado aos participantes voluntários

Dados do participante

*Obrigatório

Gênero *

Masculino

Feminino

Idade *

Sua resposta

Grau de escolaridade *

Ensino Básico

Ensino Secundário

Primeiro ciclo do Ensino Superior (Licenciatura)

Segundo ciclo do Ensino Superior (Mestrado)

Terceiro ciclo do Ensino Superior (Doutoramento)

Profissão *

Sua resposta

Já participou em alguma experiência de realidade virtual? *

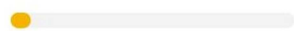
Sim

Não

Se sim, descreva brevemente

Sua resposta

PRÓXIMA

 Página 1 de 14

Questionário apresentado aos participantes voluntários

*Obrigatório

Cenário 1a

Reparou em alguma incoerência no cenário/animação que acabou de visualizar? *

	0	1	2	3	4	5	
Nenhuma incoerência	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muitas incoerências

Se sim, diga quais

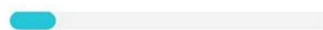
Sua resposta

Conseguiu distinguir quais os objetos mais próximos e quais os mais afastados? *

	0	1	2	3	4	5	
Não, muito mal	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sim, muito bem

VOLTAR

PRÓXIMA



Página 2 de 14

Questionário apresentado aos participantes voluntários

*Obrigatório

Cenário 1b

Reparou em alguma incoerência no cenário/animação que acabou de visualizar? *

	0	1	2	3	4	5	
Nenhuma incoerência	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muitas incoerências

Se sim, diga quais

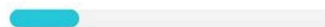
Sua resposta

Conseguiu distinguir quais os objetos mais próximos e quais os mais afastados? *

	0	1	2	3	4	5	
Não, muito mal	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sim, muito bem

VOLTAR

PRÓXIMA



Página 3 de 14

*Obrigatório

Cenário 1c

Reparou em alguma incoerência no cenário/animação que acabou de visualizar? *

	0	1	2	3	4	5	
Nenhuma incoerência	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muitas incoerências

Se sim, diga quais

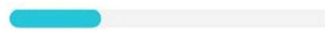
Sua resposta

Conseguiu distinguir quais os objetos mais próximos e quais os mais afastados? *

	0	1	2	3	4	5	
Não, muito mal	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sim, muito bem

VOLTAR

PRÓXIMA



Página 4 de 14

Questionário apresentado aos participantes voluntários

*Obrigatório

Cenários 1a, 1b e 1c

Em cada cenário, a bola ficou corretamente escondida ao passar por trás do objeto em primeiro plano? (0 = discordo completamente; 5 = concordo completamente) *

	0	1	2	3	4	5
Cenário 1a	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Cenário 1b	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Cenário 1c	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Em cada cenário, a bola bateu corretamente contra o obstáculo? (0 = discordo completamente; 5 = concordo completamente) *

	0	1	2	3	4	5
Cenário 1a	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Cenário 1b	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Cenário 1c	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Classifique a qualidade visual de cada cenário (0 = muito má qualidade; 5 = muito boa qualidade) *

	0	1	2	3	4	5
Cenário 1a	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Cenário 1b	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Cenário 1c	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

VOLTAR

PRÓXIMA

Página 5 de 14

Questionário apresentado aos participantes voluntários

*Obrigatório

Cenário 2a

Reparou em alguma incoerência no cenário/animação que acabou de visualizar? *

	0	1	2	3	4	5	
Nenhuma incoerência	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muitas incoerências

Se sim, diga quais

Sua resposta

Conseguiu distinguir quais os objetos mais próximos e quais os mais afastados? *

	0	1	2	3	4	5	
Não, muito mal	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sim, muito bem

VOLTAR

PRÓXIMA



Página 6 de 14

*Obrigatório

Cenário 2b

Reparou em alguma incoerência no cenário/animação que acabou de visualizar? *

	0	1	2	3	4	5	
Nenhuma incoerência	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muitas incoerências

Se sim, diga quais

Sua resposta

Conseguiu distinguir quais os objetos mais próximos e quais os mais afastados? *

	0	1	2	3	4	5	
Não, muito mal	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sim, muito bem

VOLTAR

PRÓXIMA

 Página 7 de 14

Questionário apresentado aos participantes voluntários

*Obrigatório

Cenário 2c

Reparou em alguma incoerência no cenário/animação que acabou de visualizar? *

	0	1	2	3	4	5	
Nenhuma incoerência	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muitas incoerências

Se sim, diga quais

Sua resposta

Conseguiu distinguir quais os objetos mais próximos e quais os mais afastados? *

	0	1	2	3	4	5	
Não, muito mal	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sim, muito bem

VOLTAR

PRÓXIMA



Página 8 de 14

Questionário apresentado aos participantes voluntários

*Obrigatório

Cenários 2a, 2b e 2c

Em cada cenário, a bola ficou corretamente escondida ao passar por trás do objeto em primeiro plano? (0 = discordo completamente; 5 = concordo completamente) *

	0	1	2	3	4	5
Cenário 2a	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Cenário 2b	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Cenário 2c	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Em cada cenário, a bola bateu corretamente contra o obstáculo? (0 = discordo completamente; 5 = concordo completamente) *

	0	1	2	3	4	5
Cenário 2a	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Cenário 2b	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Cenário 2c	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Classifique a qualidade visual de cada cenário (0 = muito má qualidade; 5 = muito boa qualidade) *

	0	1	2	3	4	5
Cenário 2a	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Cenário 2b	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Cenário 2c	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

VOLTAR

PRÓXIMA

Página 9 de 14

Questionário apresentado aos participantes voluntários

*Obrigatório

Cenário 3a

Reparou em alguma incoerência no cenário/animação que acabou de visualizar? *

	0	1	2	3	4	5	
Nenhuma incoerência	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muitas incoerências

Se sim, diga quais

Sua resposta

Conseguiu distinguir quais os objetos mais próximos e quais os mais afastados? *

	0	1	2	3	4	5	
Não, muito mal	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sim, muito bem

VOLTAR

PRÓXIMA

 Página 10 de 14

*Obrigatório

Cenário 3b

Reparou em alguma incoerência no cenário/animação que acabou de visualizar? *

	0	1	2	3	4	5	
Nenhuma incoerência	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muitas incoerências

Se sim, diga quais

Sua resposta

Conseguiu distinguir quais os objetos mais próximos e quais os mais afastados? *

	0	1	2	3	4	5	
Não, muito mal	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Sim, muito bem

VOLTAR

PRÓXIMA

 Página 11 de 14

*Obrigatório

Cenários 3a e 3b

Em cada cenário, a bola ficou corretamente escondida ao passar por trás dos obstáculos do tabuleiro? (0 = discordo completamente; 5 = concordo completamente) *

	0	1	2	3	4	5
Cenário 3a	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Cenário 3b	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Em cada cenário, a bola bateu corretamente contra obstáculos do tabuleiro? (0 = discordo completamente; 5 = concordo completamente) *

	0	1	2	3	4	5
Cenário 3a	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Cenário 3b	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Classifique a qualidade visual de cada cenário (0 = muito má qualidade; 5 = muito boa qualidade) *

	0	1	2	3	4	5
Cenário 3a	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Cenário 3b	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

VOLTAR

PRÓXIMA

 Página 12 de 14

Inquérito concluído. Obrigado pela sua colaboração.

Por favor, peça ao investigador para concluir o formulário.

VOLTAR

PRÓXIMA



Página 13 de 14

*Obrigatório

Duração total das experiências

Duração *

Sua resposta

ID do participante *

Sua resposta

Código da sequência de testes *

Sua resposta

VOLTAR

ENVIAR



Página 14 de 14

Nunca envie senhas pelo Formulários Google.

Anexo D

Resultados Qualitativos

10.1 Resposta dos participantes nas questões notadas de 0 a 5

Página do Questionário	Duração	ID	Cena A									Cena B									Cena C												
			A1	A2	A3	A1	A2	A3	A1	A2	A3	A1	A2	A3	B1	B2	B3	B1	B2	B3	B1	B2	B3	B1	B2	B3	C1	C2	C1	C2	C1	C2	C1
			Página 2	Página 3	Página 4	Página 5	Página 5	Página 5	Página 6	Página 7	Página 8	Página 9	Página 9	Página 9	Página 10	Página 11	Página 12	Página 12	Página 12														
20	1	0 5	1 5	0 5	5 5 5	5 5 5	5 5 5	5 5 5	5 5 5	5 5 5	1 5	0 5	0 5	5 5 5	5 5 5	5 5 5	3 4 5	0 5	1 5	5 5	5 4	5 4	5 4										
24	2	0 5	1 5	1 5	5 5 5	5 5 5	5 5 5	5 4 5	3 4 3	0 5	1 5	0 3	5 5 4	5 5 5	2 3 3	1 5	4 4	5 2	5 5	4 2	5 5	4 2											
23	3	1 5	1 5	0 5	5 5 5	5 5 5	5 4 5	3 4 3	0 5	1 5	0 3	5 5 4	5 5 5	2 3 3	1 5	4 4	5 2	5 5	4 2	5 5	4 2												
20	4	2 4	2 4	2 4	4 4 4	3 3 3	4 4 4	1 4	0 4	0 4	0 4	0 5 5	4 4 2	4 4 4	1 5	2 5	4 4	4 2	5 5	5 5	5 5												
20	5	0 5	0 5	0 5	5 5 5	5 5 5	5 5 5	4 4 4	0 4	0 4	0 4	5 5 5	5 0 5	3 3 3	1 3	0 3	3 5	2 5	3 3	2 5	3 3												
24	6	0 4	4 5	0 5	5 5 5	5 0 5	4 4 4	5 5	0 5	0 5	5 5 5	5 5 5	5 5 5	0 4 4	0 5	4 5	5 5	5 2	4 3	5 2	4 3												
20	7	0 5	0 5	1 5	5 5 5	5 5 5	5 5 5	5 5 5	0 5	0 5	0 5	5 5 5	5 5 5	5 5 5	0 5	0 5	5 5	5 5	5 5	5 5	5 5												
15	8	1 4	1 4	1 4	5 5 5	5 5 5	5 4 4	1 3	1 3	1 3	5 5 5	5 5 5	5 5 5	4 4 4	1 4	1 4	5 5	5 5	5 5	4 4	4 4												
22	9	0 5	0 5	0 4	4 4 4	4 4 4	3 3 3	1 5	0 5	1 5	4 4 4	4 4 3	4 4 3	4 4 4	0 5	1 5	4 4	5 5	3 5	5 5	3 5												
25	10	1 5	1 5	1 5	5 5 5	5 5 5	3 4 4	2 5	2 5	1 5	5 5 5	5 5 5	5 5 5	3 2 3	0 5	3 4	0 0	4 2	5 2	5 2	5 2												
20	11	0 3	0 3	0 3	5 5 5	4 4 4	4 4 4	0 4	1 4	1 2	5 5 5	4 5 4	3 4 3	1 4	1 3	5 5	3 4	4 3	4 3	4 3	4 3												
30	12	3 4	3 3	2 3	3 4 3	1 1 1	3 3 1	3 4	2 4	2 4	4 3 3	3 4 2	2 4 3	3 3	2 4	3 3	1 3	4 3	1 3	4 3	4 3												
28	13	2 3	2 3	2 3	4 4 4	2 2 2	4 3 3	1 4	0 4	1 4	3 4 4	4 4 3	3 3 3	1 3	3 2	3 2	3 1	4 3	1 4	4 3	4 3												
27	14	0 5	2 5	1 5	5 5 5	5 5 5	5 4 3	0 5	0 5	0 5	5 5 5	3 5 5	5 5 5	0 5	3 5	4 4	5 5	5 4	5 5	5 4	5 4												
23	15	2 4	2 4	3 4	4 4 4	3 4 3	4 4 4	4 4	2 4	2 5	1 1 1	2 1 4	2 4 5	1 4	4 2	3 3	1 2	4 2	4 2	4 2	4 2												
20	16	0 5	0 5	0 5	5 5 5	2 4 5	2 3 4	0 5	2 5	1 5	5 5 5	4 2 5	5 3 4	0 5	1 5	5 3	4 2	4 3	4 3	4 3	4 3												
52	17	3 4	4 5	2 4	4 4 4	3 4 3	3 5 4	3 4	3 3	3 3	5 5 4	4 4 3	3 3 3	2 3	3 4	4 3	3 3	3 4	3 3	3 4	3 4												
25	18	0 5	0 5	0 5	4 5 5	4 4 5	1 3 4	4 5	3 4	3 4	4 4 4	4 4 4	4 4 0	4 1 3	0 5	1 5	4 5	4 4	4 5	4 4	4 5												
17	19	2 5	1 5	2 5	4 5 5	5 4 2	3 5 4	1 4	1 5	0 4	3 4 4	2 4 4	2 5 3	1 4	0 5	4 4	3 4	3 4	5 3	4 5	5 3												
22	20	0 5	1 5	1 5	5 5 5	5 5 5	5 5 4	0 5	0 5	1 5	5 5 5	5 5 5	5 5 5	0 5	0 5	5 5	5 5	5 5	5 5	5 5	5 5												
22	21	0 5	0 5	0 5	5 5 5	5 5 5	4 5 3	0 5	0 5	0 5	5 5 5	5 5 5	5 5 5	3 5 4	0 5	3 5	5 5	5 5	5 5	5 5	4 3												
16	22	0 5	0 5	0 5	5 5 5	5 5 5	5 4 3	0 5	0 5	0 5	5 5 5	5 5 5	5 5 5	4 5 1	0 5	0 5	5 5	5 5	5 4	5 3	5 3												
25	23	1 3	0 5	2 2	5 5 3	4 5 3	4 5 2	0 4	1 4	1 5	4 2 3	4 1 4	4 1 3	2 2	2 4	5 2	3 3	2 3	2 3	2 3	2 3												
42	24	1 5	1 5	1 5	4 4 5	4 5 4	4 4 5	1 5	0 5	1 5	4 5 4	5 4 5	4 5 4	0 5	2 5	5 3	5 2	4 5	5 2	4 5	4 5												
22	25	1 4	2 4	2 4	5 5 5	5 5 5	3 4 4	1 4	1 4	1 4	5 5 5	5 5 5	5 5 5	4 4 4	1 4	4 5	1 5	4 4	4 4	4 4	4 4												

10.2 Resposta dos participantes nas questões abertas

Cena A – Versão A1

A sombra da bola parecia mais escura.

O impacto da bola contra o vaso não pareceu muito natural, assim como a velocidade na cadeira após queda

O movimento da bola na cadeira foi um bocado artificial. Movimento muito reto

O movimento continua a não ser muito realista, por exemplo quando cai na cadeira. No entanto, quanto bate no vaso, penso que foi mais realista.

A bola acelerou quando chegou à cadeira

Semelhante à anterior.

O movimento da queda para a cadeira continua irrealista. Noto o serrilhado nas arestas dos candeeiros, na flor e na aresta da porta. O cenário de fundo continua com baixa resolução.

1- Baixo numero de frames 2- Pneu da nave no cenário

Relativamente ao cenário anterior este as imagens neste são mais nitidas sendo possível distinguir o número da bola "15"

Incoerência com a iluminação no solo, com o candeeiro do lado esquerdo, pelo qual não existe degradação da luz após o início da mesa.

A bola quando estava quase a cair deixou uma sombra maior na parte superior da mesa

Cena A – Versão A2

A bola rolou um pouco para fora da mesa antes de cair, efetivamente.

A mesma da anterior, (sombras)

A bola parecia não estar no fundo da cadeira.

Novamente sem diferenças

A bola flutuou na beira da mesa e não fez peso na cadeira.

O movimento da bola na cadeira, após a queda da mesma da mesa, podia ser mais fluido. A flor está muito desfocada.

O movimento da bola não é muito realista. Assim como as sombras.

O cenário estava com uma iluminação mais forte. A bola teve o mesmo movimento que o cenário anterior

A bola passa além da mesa e parece flutuar no ar.

A mesma coisa.

A queda pareceu mais real, mas o amortecimento ainda não convence. Nota-se uma melhoria na definição das arestas.

melhor FPS que o anterior

Parece que a bola flutui por cima da almofada da cadeira.

Resultados Qualitativos

O impacto quando a bola bate no vaso é estranha. O impacto é estranho e a bola fica demasiado lenta.

A bola quando estava quase a cair deixou uma sombra maior na parte superior da mesa

Cena A – Versão A3

Iluminação podia estar mais apurada - maior definição de sombras

Semelhante ao cenário visualizado anteriormente

sombra continua igual em todo percurso

Movimento da bola é muito reto.

Esta imagem está mais desfocada que as anteriores.

Este cenário pareceu-me igual ao anterior, se bem que a luz do candeeiro em cima da mesa não corresponde à luz exibida no teto (para comparar, o outro candeeiro emite uma luz correspondente à sua forma). A paisagem fora da janela tem uma resolução muito baixa

O número na bola não estava tão visível.

O "salto" da bola para a cadeira.

A queda da bola para a cadeira e o amortecimento da bola na almofada não parecia muito realista.

O cenário do Céu não estava muito definido.

1- Baixo FPS mas melhor que o anterior 2- Pneu

Pareceu estar menos suavizada que a versão anterior.(menos fps)

Imagem de um dos candeeiros não parece estar na distância correta

Nota-se uma redução de iluminação da mesa e outros objectos, todavia o fundo da sala não sobre diferenças na iluminação. O vaso parece diferente e partido ao meio

A bola quando estava quase a cair deixou uma sombra maior na parte superior da mesa

Cena B – Versão B1

Parecia haver alguma incoerência nos detalhes do chão na minha vizinhança mais próxima em relação à parte mais afastada do cenário.

A imagem no geral estava muito pixelizada, a nível de cores parecia estar semelhante aos anteriores. Algum arrastamento adicional de frames, em comparação com as anteriores.

Bola não visível atrás de um vidro com display

Má qualidade.

Velocidade mais lenta da bola

O chão tinha lista arco-íris

As arestas dos objectos têm muito luminosidade.

O avião era maior em relação ao cenário anterior, e a bola deveria ter parado eventualmente.

O chão e a borda inferior da parede (da esquerda) pareciam ter riscos luminosos.

"O problema do azulejo foi resolvido, o fumo do avião e a onda de calor ficou mais convincente, o já se nota o foco. Reparei que apareceram reflexos do fogo no chão, mas o

Resultados Qualitativos

serrilhado aumentou, a definidas imagens que aparecem nos vidros aumentou, mas o serrilhado também aparece.

A roda do avião onde bate a bola continua a não convencer."

A bola termina no ar

Existe mais iluminação na cenário, de uma forma geral e, uma maior iluminação do lado direito da nave. Voltou a ver-se a estrela e, a bola parece abrandar um pouco no fim e, talvez devesse aumentar a velocidade por causa do vento criado pelo avião.

Cena B – Versão B2

A mesma questão do efeito do jato. O chão pareceu ser mais refletor.

A bola parecia rodar sobre si mesma em vez de acompanhar o piso

Imagem desfocada

A bola está mais rápida

A bola estava mais nítida que no cenário anterior.

Quando a bola passou atrás do painel semi-transparente simplesmente deixou de ser detectável.

A bola não se vê quando passa pela porta de vidro

O chão ficou igual ao primeiro. O serrilhado foi reduzido. A sombra parece muito mais irrealista. Os raios de sol parecem passar pela parede. A bola ficou com um formato estranho à medida que se afastava. Voltaram a ser colocados os focos do 1º ensaio. O reflexo do fogo foi muito mais reduzido. Na base do suporte do placar mais próximo, pude observar quadrados muito grandes cinzentos, uma textura com muito pouca resolução. Parece muito mais lisa, com perda de resolução das texturas.

Melhor grau de renderização e de frames por minuto

Notei mais fluidez no visionamento do cenário

Cena B – Versão B3

Fumo do jato quando a bola de basquete desaparece? Efeito de luz um pouco estranho

A bola desaparece antes de chegar à parede.

Imagem um bocado desfocada

A bola desaparece a meio.

As arestas estão melhores, mas a visão periférica pode ser melhorada. Continuo a ver arestas luminosas no objectos para onde não estou a olhar directamente.

Resolução baixa (mas poderá dever-se a ser um dispositivo móvel), e quanto ao movimento da bola, ao bater no canhão(?) deveria ter andado com uma velocidade ainda menor quando muda a trajetória

O mesmo.

o sol desapareceu

O painel azulejo que fica no eixo está torto. A roda onde a bola bate aparenta não ter sombra, não se encaixando bem no sítio, as escadas parecem muito pouco definidas, e não consegui

Resultados Qualitativos

perceber que o objecto que estava a apontar para o avião era um foco. O movimento da bola por de trás do vidro não me pareceu correto. O ondular das ondas de calor dos jatos não me pareceu convincente.

O sol desapareceu.

Cenário muito mais nítido

Desapareceu o sol em cima e verifiquei uma maior diferença na sombra da bola enquanto esta rodava.

Cena C – Versão C1

Algum distorção nas texturas em madeira.

A sombra da bola sobrepunha-se aos lados da caixa, parecendo que a bola tinha mais altitude, mas poderá ser normal

O ressalto é mais realista, a gravidade parece ter mais efeito (de acordo com o esperado) mas alguns movimentos ainda parecem "virtuais"

Às vezes a bola parecia que passava por cima dos mecos sem existir colisão, uma sensação geral de menos colisão do que havia no anterior.

A bola saltou muito e não parecia muito realista

Quando a bola está em cima, parece um pouco flutuante. Quando bate nos objectos de madeira o movimento é bom, porém, quando bate no fundo não é tão bom, pois é esperado algum ressalto e isso não acontece.

Estava menos iluminado e o efeito descrito anteriormente já não era tão acentuado. A velocidade da bola parecia mais realista.

O salto da bola pareceu muito artificial.

Ao longe, a bola de ténis parece uma bola de ping pong. A sombra pareceu-me muito estranha. Saltava como uma bola de borracha e aterrou de uma maneira estranha no número. Notei um serrilhado nos candeeiros e pareceu-me e fundo do céu continua com muito pouca resolução.

quer a bola quer os números estão mal posicionados

Desfoque parcial do cenário

Cena C – Versão C2

A bola, depois de cair numa das entradas, parecia estar a "pairar" sobre a parte de madeira, sem lhe tocar.

Noção 3d mais confusa e desproporcional, madeira baça.

A geometria das oclusões era menos detalhada do que o objecto, o que também se refletia nas sombras. A bola era mais saturada do que o resto da cena.

A bola parece ressaltar demasiado com impacto contra mecos

Sombra estranha quando a bola chega ao número.

Mais luz e é possível ver as riscas, azul e amarela, da bola verde.

Resultados Qualitativos

A bola quando terminou o seu trajeto, ou seja, quando chegou ao patamar de baixo, não ficou parada na superfície, mas sim no ar. Imagem desfocada. Movimento da bola artificial e saltos muito longos e rápidos.

Os números parecia que estavam a flutuar no ar

O movimento da bola é mais fluído na queda, mas o problema do ressalto no fim persiste.

A bola devia perder alguma velocidade quando ressalta, mas o mais importante é que quando ela ressalta, parece que está a saltar para fora do ecrã quando na realidade mantém-se no mesmo plano.

A sombra da bola. A forma como ela caiu ao final foi pareceu não respeitar o movimento que se mostra durante o percurso.

Os números das pontuações pareciam flutuar.

A bola parecia que ia sair da mesa. Parecia que tinha um vidro

As texturas estão muito mais definidas, o serrilhado desapareceu. A sombra da bola parece muito mais realista, já me parece uma bola de ténis ao longe mas o seu movimento parece ter piorado.

O número a vermelho parece fora de sítio, a bola parece aterrar à beira dos números em vez de ficar na calha. No teto pareceu-me ver o simbolo do infinito no eixo e no chão uma linhas curvas deformantes.

Dá a sensação que a bola parece um pouco leve de mais, o quando bate num obstáculo muda muito de direção.

A bola fica "suspensa" quando termina na pontuação. Em alguns pontos a bola fica cortada.

parece que a bola não está a ser corretamente escondida pelos separadores

A iluminação sofreu alterações, ficando a mesa de jogo mais escura e, existe uma enorme diferença no movimento da bola, sendo muito mais agradável, parecendo ter impactos menos exagerados e uma velocidade um pouco mais reduzida.