



Faculdade de Ciências e Tecnologia da Universidade de Coimbra
Departamento de Engenharia Electrotécnica e de Computadores
Mestrado em Tecnologias de Informação Visual

Biblioteca para Desenvolvimento de Aplicações de Realidade Aumentada Baseada em Marcadores Binários

Bruno André Santos Patrão

Júri:

Doutor Paulo José Monteiro Peixoto
Doutor Paulo Jorge Carvalho Menezes
Doutor João Pedro de Almeida Barreto

Setembro 2011

“The vital factor of virtual reality is immersion, the degree to which the user's senses are limited to the simulation and screened from a real world.”

Fjodor Ruzic

Agradecimentos

Aos meus pais e irmãos, pela compreensão, pelo suporte e apoio que me deram ao longo do meu percurso acadêmico...

À minha família, pela compreensão e pelos incentivos para eu seguir em frente...

À Ana, pelo apoio e incentivo, pela amizade e amor...

Aos meus colegas de curso pela partilha de momentos académicos e de vida, pela amizade e companheirismo...

Ao Professor Doutor Paulo Menezes, pelo apoio e encorajamento, pelos momentos de sabedoria e ensinamentos...

Resumo

O presente trabalho explora a área da Realidade Aumentada e foca-se na integração de objectos virtuais em ambientes reais em tempo real. O objectivo geral é desenvolver uma biblioteca compreensiva e de fácil utilização para integrar no OpenAR e possibilitar a criação de aplicações de Realidade Aumentada. O elemento chave para a interacção com esta biblioteca são os marcadores simples com um código binário criados para fornecer informação sobre o mundo real em tempo real. São descritos e caracterizados os processos que servem de base para o desenvolvimento deste trabalho, nomeadamente a binarização de imagem, a detecção de marcadores, a extracção e aplicação da pose a objectos virtuais. Os resultados deste trabalho comprovam a importância e utilidade de implementar um sistema de Realidade Aumentada desta natureza em diferentes áreas, tais como a interacção homem-máquina, o entretenimento, a educação, a medicina/psicologia e a indústria. Como trabalho futuro, são propostas melhorias ao nível da visualização e realismo dos ambientes virtuais.

Palavras-chave: Realidade Aumentada; Marcador Binário; Segmentação da Imagem; Pose dos Marcador; Tempo Real.

Abstract

The present work explores the area of Augmented Reality and focus on the integration of virtual objects within real ambient in real time. The main objective is to develop a comprehensive library of easy utilization in order to integrate on OpenAR and improve the possibility of create Augmented Reality applications. The key element for the interaction with that library is simple markers with a binary code created to deliver information about the real world in real time. The processes that compose the basis to the development of this work are here exposed and characterized, namely the image binarization, the markers detection, the extraction and application of pose to virtual objects. This work results prove the importance and utility of an Augmented Reality system implantation of this nature in different areas, such as human-computer interaction, entertainment, education, medicine/psychology and industry. As a proposition for future work, improvements in the visualization and realism of virtual ambient are also presented.

Keywords: Augmented Reality; Binary Marker; Image Segmentation; Marker Pose; Real Time.

Índice

Lista de Figuras.....	vii
Abreviaturas.....	ix
1. Introdução	1
1.1. Objectivos.....	4
1.2. Motivação.....	5
1.3. Estado de Arte	5
1.4. Estrutura da Tese.....	7
2. Fundamentação Teórica.....	8
2.1. Marcadores Binários.....	9
2.2. Segmentação da Imagem	10
2.3. Características da imagem	13
2.4. Posição e Orientação da Câmara.....	15
3. Desenvolvimento e Resultados.....	23
3.1. Binarização da Imagem	24
3.2. Detecção de Contornos	25
3.3. Filtragem de Dados	27
3.4. Extração da Posição e Orientação	30
3.5. Predictores de Movimento.....	32
3.6. Demonstradores.....	34
4. Conclusão.....	45
4.1. Outras Aplicações.....	46
4.2. Trabalho Futuro.....	48
Referências	51
Apêndice A.....	54

Lista de Figuras

Figura 1 - Continuum de virtualidade proposta por Milgram em 1994.....	2
Figura 2 - Etapas do processo de Realidade Aumentada desenvolvido.	4
Figura 3 – Principais passos do processo de Realidade Aumentada abordados.	8
Figura 4 – Descrição dos marcadores criados.	9
Figura 5 – Gráfico do histograma de uma imagem em níveis de cinzento.....	11
Figura 6 – Representação dos contornos de uma imagem.....	14
Figura 7 – Modelo físico (esquerda) e respectivo modelo imagem (direita).....	16
Figura 8 – Ilustração do modelo de câmara <i>pinhole</i>	17
Figura 9 – Ilustração da distância focal (f) e do ponto principal (c).....	17
Figura 10 – Resultado da binarização de Otsu.	24
Figura 11 – Processo de aproximação de contornos a polígonos mais simples.	26
Figura 12 – Resultado dos possíveis marcadores na imagem.....	27
Figura 13 – Marcadores na imagem (esquerda) e os respectivos cantos (direita).	28
Figura 14 – Modelo físico do marcador (esquerda) e sua representação no plano imagem (direita).....	29
Figura 15 – Localização do ponto de orientação (esquerda) e identificação do marcador (direita).....	29
Figura 16 – Pose do marcador (esquerda) aplicada ao objecto virtual e à câmara virtual (direita).	30
Figura 17 – Processo de funcionamento do preditor de movimento.	33
Figura 18 – Múltiplos marcadores identificados (esquerda) e objectos associados (direita).....	34
Figura 19 – Interação com base na oclusão dos marcadores.	35
Figura 20 – Mapeamento de texturas com base na informação do marcador.....	36
Figura 21 – Criação de um cubo com base no quadrado desenhado no marcador.	37
Figura 22 – Interação com objectos virtuais tirando partido da física.	38
Figura 23 – Realidade Aumentada aplicada em livros escolares.....	39
Figura 24 – Realidade Aumentada para visualização médica.	40
Figura 25 – Exposição do sujeito a situações de fobia.	40
Figura 26 – Jogo das catapultas desenvolvido em Realidade Aumentada.	41
Figura 27 – Demonstração de um marcador cúbico.	42
Figura 28 – Utilização dos marcadores para recriar o efeito de lupa.	42

Figura 29 – Realidade Aumentada num quadro eléctrico.....	43
Figura 30 – Tutorial para a reparação de uma câmara digital.	43
Figura 31 – Hilário, o robô guia.	46
Figura 32 – Aplicação do sistema desenvolvido no projecto Hilário.	47
Figura 33 – Aplicação do sistema desenvolvido no modelador FxC.	48
Figura 34 – Problema da oclusão.....	49
Figura 35 – Problema das sombra.....	49

Abreviaturas

ISR – Instituto de Sistemas e Robótica

DEEC – Departamento de Engenharia Electrotécnica e de Computadores

UC – Universidade de Coimbra

HDM – Head Mount Display

RA – Realidade Aumentada

RV – Realidade Virtual

VC – Visão por computador

CG – Computação Gráfica

OpenAR – Open Augmented Reality

OpenGL – Open Graphics Library

OpenCV – Open Computer Library

SDL – Simple Directmedia Layer

TCL/TK – Tool Command Language/Tool Kit

1. Introdução

A Computação Gráfica é actualmente uma grande indústria. As diversas aplicações da Computação Gráfica vão desde o mais óbvio como o cinema e os jogos até aos menos conhecidos como a educação e a medicina [1].

A Computação Gráfica tem dominado a indústria cinematográfica nos últimos anos proporcionando a possibilidade de integrar efeitos gerados por computador em cenas reais. O contra aqui é que estes elementos gerados por computador têm de ser adicionados depois das cenas serem filmadas, um processo que pode levar meses a concluir.

A capacidade dos computadores para lidar com tarefas mais complexas tem crescido a par com o poder de processamento dos mesmos, em particular a capacidade de executar operações graficamente pesadas em tempo real. Devido a esta grande difusão os vídeos digitais estão cada vez mais presentes no nosso quotidiano: vídeos capturados de acontecimentos reais ou produzidos por computador. Em muitos deles temos a parte virtual misturada com imagens reais.

Um domínio em grande crescimento é a Visão por Computador. Esta é a habilidade que dá ao computador a capacidade de reconhecer pontos de interesse ou extrair características específicas tanto de uma só imagem como de uma sequência (vídeo) [2]. Esta área tem vindo a convergir com a Computação gráfica.

A Realidade Aumentada, por sua vez, é uma fusão da Visão por Computador com a Computação Gráfica, onde o uso da combinação entre modelos gráficos e as sequências de imagens tira o melhor proveito dos avanços tecnológicos e do poder de processamento.

A evolução tecnológica proporcionou, na década de 90, o aparecimento da Realidade Aumentada, permitindo a sobreposição de objectos e ambientes virtuais no ambiente físico, através de algum dispositivo tecnológico (capacete de Realidade Virtual ou monitor e câmara). Essas aplicações ficaram mais acessíveis somente a partir do ano 2000, com a convergência de técnicas de Visão Computacional, *software* e dispositivos com melhor índice de custo-benefício. Além disso, o facto dos objectos

virtuais serem trazidos para o espaço físico do utilizador (por sobreposição) permitiu interacções tangíveis mais fáceis e naturais, sem o uso de equipamentos especiais. Por isso, a Realidade Aumentada tem vindo a ser considerada uma possibilidade concreta de uma próxima geração popular de interface, a ser usada nas mais variadas aplicações em espaços internos e externos [3].

Trata-se assim de uma área relativamente recente e em constante desenvolvimento e expansão, existindo diversas definições.

Segundo Insley, a Realidade Aumentada é uma melhoria do mundo real com textos, imagens e objectos virtuais, gerados por computador [4].

Paul Milgram [5] propôs um continuum de virtualidade (Figura 1), que varia entre um ambiente totalmente real e um ambiente totalmente virtual no extremo oposto. Na verdade é uma escala contínua que envolve todas as possibilidades de inserção de elementos virtuais no mundo real ou de elementos reais no mundo virtual.

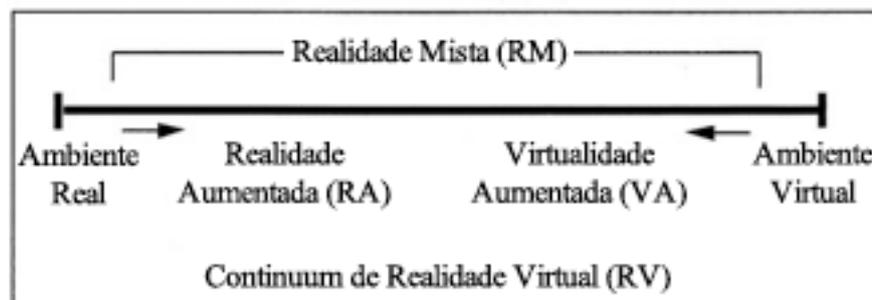


Figura 1 - Continuum de virtualidade proposta por Milgram em 1994.

A zona entre os dois extremos é chamada de realidade mista ou misturada, que consiste na sobreposição de elementos virtuais gerados por computador no ambiente físico, apresentado ao utilizador através de algum interface de visualização. Essa realidade misturada pode ser dividida em Realidade Aumentada e Virtualidade Aumentada.

A Realidade Aumentada consiste na inserção de objectos virtuais no ambiente físico, mostrando ao utilizador, em tempo real e com o apoio de algum dispositivo tecnológico, o interface no ambiente real [3]. O termo Realidade Aumentada foi cunhado no começo da década de 90 por Thomas Preston Caudell [6] enquanto

trabalhava na Boeing. Ele foi responsável por criar um sistema em que o operário usava um capacete de Realidade Virtual (HMD) e mostrava os esquemas de fios de cada avião, substituindo assim as caras placas de madeira com os esquemas.

A Virtualidade Aumentada é a inserção de representações de elementos reais no mundo virtual, usando o interface que permite ao utilizador interagir com o ambiente virtual [3].

Por sua vez, Azuma [7] refere que a Realidade Aumentada é um sistema que complementa o mundo real com objectos virtuais gerados por computador, parecendo coexistir no mesmo espaço e apresenta as seguintes propriedades: (i) combinação de objectos reais e virtuais no ambiente real; (ii) execução interactiva em tempo real; (iii) alinhamento de objectos reais e virtuais entre si; e (iv) utilização de todos os sentidos, nomeadamente da visão, da audição, do tacto, do olfacto e do paladar.

Esta tese descreve o processo de desenvolvimento de Realidade Aumentada e contribui para o melhoramento do OpenAR (um sistema em desenvolvimento no instituto de sistemas e robótica – ISR/DEEC), permitindo a criação de aplicação de Realidade Aumentada de forma simples e rápida. O sistema funciona procurando constantemente por um marcador preto e branco na sequência de imagens (vídeo) e posicionando um modelo gráfico sobre o mesmo. A vantagem do sistema desenvolvido neste trabalho em relação aos já existentes é o facto de utilizar marcadores binários de rápida detecção, possibilitando uma experiência em tempo real. Todo este processo está dividido e descrito, ao longo deste documento, em várias etapas onde cada uma tem uma tarefa específica, embora ligadas entre si (Figura 2).

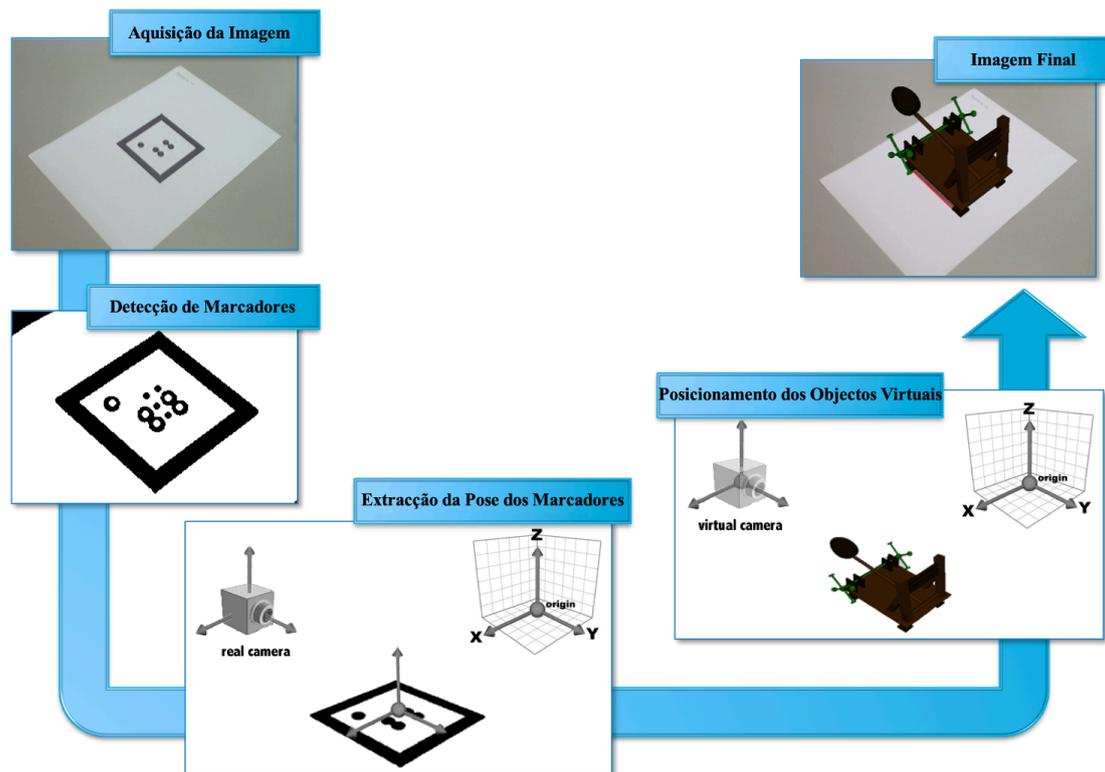


Figura 2 - Etapas do processo de Realidade Aumentada desenvolvido.

Assim, podemos concluir que a Realidade Aumentada permite um enriquecimento do ambiente real com elementos virtuais, esses elementos inseridos na cena podem ser de qualquer natureza, por exemplo, objectos tridimensionais que representam pessoas ou uma imagem plana num placar informativo.

1.1. Objectivos

O presente trabalho tem como objectivo geral criar uma biblioteca de Realidade Aumentada para desenvolver e expandir os conteúdos do OpenAR. Especificamente, este trabalho visa criar um algoritmo rápido que funcione em tempo real com base no cálculo da posição de marcadores binários no mundo real; permitir a criação e manipulação, de um modo simples e rápido, de aplicações de Realidade Aumentada; assim como, desenvolver demonstrações das funcionalidades do trabalho desenvolvido em Realidade Aumentada.

1.2. Motivação

A Realidade Aumentada é uma área de grande interesse porque possibilita o aumento da percepção, interacção e imersão do utilizador com o mundo real. Os objectos virtuais oferecem informação que o utilizador não conseguiria detectar apenas com os seus sentidos, sendo que a sobreposição desta informação no mundo real ajuda-o a executar certas tarefas. Desta forma, a Realidade Aumentada pode ser aplicada a diversas áreas do quotidiano do ser humano, nomeadamente na interacção entre o utilizador e a máquina, na educação, na saúde, no entretenimento e na indústria. O facto de, com a Realidade Aumentada, os objectos virtuais e o ambiente real parecerem coexistir pode facilitar e melhorar o estilo de vida das pessoas.

Assim, o investimento e aperfeiçoamento de técnicas de Realidade Aumentada assumem um papel importante, na medida em que possibilitam novos cenários de interacção e percepção cognitiva (por exemplo, visão, audição) entre objectos reais e objectos virtuais. Tendo em conta as suas funcionalidades, é possível imaginar um futuro em que a Realidade Aumentada é o principal método de interacção com o mundo.

1.3. Estado de Arte

A Realidade Aumentada tem várias definições. Uma delas descreve o campo da Visão por Computador que lida com a combinação de objectos gerados por computador e o mundo real, onde estes objectos são sobrepostos às cenas reais e representados em tempo real [3, 8].

Actualmente o aumento da realidade abrange muitas áreas, tendo atingindo níveis para além da pesquisa ou protótipos e entrando no campo das soluções práticas e comerciais [6]. Estas áreas incluem a educação, a medicina, a indústria, a publicidade, o entretenimento e o turismo.

Os projectos educacionais podem trazer um novo conceito de aprendizagem através de eventos interactivos ou designs específicos para livros escolares [9]. Dando este exemplo para perceber como funciona a Realidade Aumentada é elucidativo porque introduzimos dois elementos essenciais: a câmara e os marcadores (padrões). Estes

dois elementos encontram-se normalmente em todas as categorias referidas. A câmara é necessária como entrada de vídeo para aquisição de imagens e os marcadores não são mais que padrões distintos que vão ser processados por *software*. Estes para além de fornecer a posição 3D e a direcção baseada num ângulo relativo à posição da câmara, também permitem o posicionamento apropriado de cenas virtuais no ambiente real. Outros propósitos educacionais passam por enciclopédias interactivas onde se pode visualizar estruturas/cenas em 3D acompanhadas da descrição, tutoriais em que a aprendizagem é feita através da visualização de animações passo-a-passo ou mesmo viagens à volta do mundo apenas com o virar de uma página.

A ligação entre as realidades mistas e a medicina também se fortaleceu muito, uma vez que torna a aprendizagem da anatomia ou a visualização de dados médicos mais intuitivos, bem como a terapia de fobias através da exposição do sujeito. Nesta área podemos referir Albert Skip Rizzo [10] que, perante um grande número de soldados com traumas pós-guerra, sentiu necessidade de criar um programa terapêutico onde os pacientes são expostos e podem navegar livremente em cenários de guerra virtuais. Esta exposição vai-se tornando cada vez mais dinâmica com o decorrer das sessões de tratamento, o que ajuda a processar melhor as memórias emocionais. Em consequência, também sentiu que era necessário um maior conhecimento dos terapeutas face a estes traumas, então revolucionou os métodos de ensino para terapeutas de saúde mental desenvolvendo um programa onde o terapeuta pode treinar as suas capacidades e interagir com pacientes virtuais. Estes pacientes têm diversas desordens mentais relacionadas, ou não, com traumas pós-guerra.

As aplicações industriais também estão a ser usadas em larga escala [6]. Por exemplo, o aumento da realidade é usado para criar linhas guia, tanto na montagem como na reparação de produtos.

No entretenimento a Realidade Aumentada vai desde jogos simples para dispositivos móveis até métodos de interacção que permite criar e orientar cenas 3D baseadas no ponto de vista físico do utilizador [11]. Os produtores de jogos já começam a lançar jogos com estas capacidades e é muito provável que vejamos uma explosão de produtos nesta área.

Em sumário, foram referidos conceitos e produtos de Realidade Aumentada, detalhando a distinção entre *software* e *hardware* desde dispositivos de larga escala

até dispositivos móveis. Contudo, como a portabilidade neste campo é crucial, começamos a ver um avanço muito grande no sentido de facilitar o desenvolvimento de produtos de Realidade Aumentada para dispositivos móveis.

1.4. Estrutura da Tese

O presente trabalho explora o conceito de Realidade Aumentada e as diferentes fases deste processo. A estrutura da tese é composta por seis capítulos. No segundo capítulo apresenta-se a fundamentação teórica de cada fase do processo de Realidade Aumentada. Segue-se o terceiro capítulo onde é abordado o desenvolvimento prático e o contributo com algoritmos para o enriquecimento dos conteúdos do OpenAR (ISR/DEEC). Este capítulo apresenta também os resultados do trabalho desenvolvido, demonstrando todas as aplicações possíveis de desenvolver com o sistema implementado. Finalmente, no quarto é realizada a conclusão deste trabalho e são sugeridas implicações futuras para melhoria do mesmo.

2. Fundamentação Teórica

No presente capítulo são apresentados os conceitos subjacentes ao trabalho desenvolvido, descrevendo princípios e técnicas que serviram de base ao mesmo.

Antes de abordar os principais passos do processo de Realidade Aumentada, importa caracterizar o elemento chave na realização deste processo, os marcadores binários. A informação que estes transmitem ao computador é um ponto de referência no mundo real. As suas características são únicas, simples e consistentes para que possam ser rapidamente detectados e correctamente identificados. Para tal é necessário desenvolver um conjunto de passos ligados entre si, nomeadamente a binarização da imagem capturada pela câmara, a detecção de características do marcador, a extração e aplicação da pose a objectos virtuais.

De um modo geral, a binarização da imagem consiste em converter a imagem de entrada para preto e branco de forma a salientar detalhes ou características relevantes, para serem posteriormente detectados e identificados os marcadores, correspondendo ao passo de detecção de características. Uma vez detectados e identificados os marcadores, é extraída a sua pose com base na transformação entre o modelo físico e os pontos correspondentes no plano da imagem. Esta pose, quando aplicada ao objecto, vai posicioná-lo de acordo com a transformação do marcador na imagem final. Desta forma, o objecto virtual e o marcador parecem coexistir no mundo real.

Todo este processo vai ser descrito detalhadamente neste capítulo, estando representado esquematicamente na Figura 3.



Figura 3 – Principais passos do processo de Realidade Aumentada abordados.

2.1. Marcadores Binários

Em sistemas de Realidade Aumentada é crucial registar e identificar correctamente informação do mundo real através da análise de imagens capturadas do mundo real. Esta detecção e identificação devem ser executadas em tempo real e convém que sejam robustas. Para tal, é necessário colocar em cena marcadores/padrões cuja distribuição e localização são conhecidas à partida. O marcador é bidimensional, composto por um quadrado/*frame* preto em fundo branco, logo, composto por oito pontos (os quatro cantos externos e os quatro cantos internos) (Figura 4).

Um dos objectivos é que o sistema seja capaz de detectar um ou mais marcadores em cena e caso estes não sejam encontrados ou sejam mal identificados todo o processo falha. Por esse motivo, os marcadores criados possuem um código de identificação único que os distingue entre si (Figura 4). A técnica escolhida consiste numa distribuição binária de oito entradas no interior de cada marcador e o motivo desta escolha é a simplicidade dos dados a trabalhar.

Na constituição do marcador salienta-se também o ponto de orientação (Figura 4), que é essencial para a não ocorrência de erros na localização do marcador. Sem este ponto não seria possível identificar com precisão a rotação do marcador, uma vez que a forma quadrada impõe uma indeterminação de noventa graus.

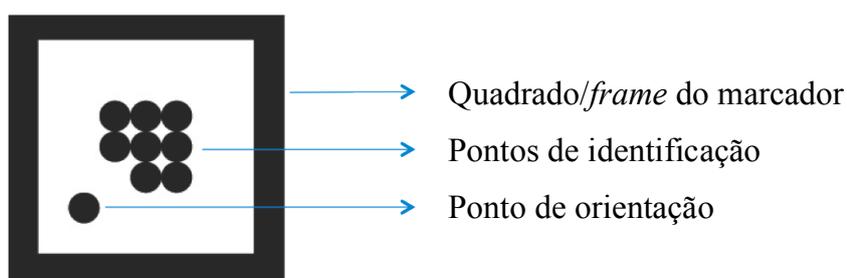


Figura 4 – Descrição dos marcadores criados.

Toda esta informação sobre os marcadores é guardada numa estrutura de dados e cada marcador encontrado na imagem tem a sua. Esta estrutura é composta pelos quatro pontos do contorno exterior, quatro pontos do contorno interior, posição do ponto de orientação, centro do marcador e código de identificação fornecido pelos pontos no interior do marcador. Nos processos descritos a seguir serão explicados como é que estes valores são obtidos.

2.2. Segmentação da Imagem

A segmentação da imagem é um dos passos mais importantes em aplicações de visão por computador. Este processo consiste em isolar diferentes elementos de uma imagem e assim determinar que características podem ser extraídas da imagem [12]. No presente trabalho, como se pretende detectar marcadores bidimensionais a preto e branco, a imagem de entrada precisa inicialmente de ser binarizada (técnica que consiste em converter uma imagem em níveis de cinzento para preto e branco, binária) [13].

Existem vários tipos limiaries para a binarização de uma imagem como limiar estático (*static thresholding*) ou limiar dinâmico (*dynamic thresholding*) [14].

O limiar estático é a forma mais simples de binarização, onde o valor de cada pixel da imagem em níveis de cinzento é convertido para preto ou branco dependendo desse limiar estático. Este método apresenta bons resultados em situações onde a luz ambiente é constante e sabendo à partida esse limiar.

O limiar dinâmico, por sua vez, calcula o limiar através da análise dos níveis de cinzento no histograma da imagem. Cada frame é analisado e o valor do limiar é calculado. A vantagem deste método é que o problema da variação de iluminação ao longo da sequência fica resolvido.

O método escolhido para ser implementado neste trabalho foi o método de Otsu [13]. O método de Otsu baseia-se na análise do histograma em níveis de cinzento de cada imagem da sequência, onde se salientam dois picos que representam o fundo e o objecto. Este método tem como pressuposto que o valor mínimo entre os dois picos é o valor ideal para o limiar.



Figura 5 – Gráfico do histograma de uma imagem em níveis de cinzento.

Como sabemos que uma imagem pode ser representada como uma função $f(x, y)$ (Figura 5), a função de binarização pode ser descrita como [15]:

$$g(x, y) = \begin{cases} R_1 & \text{se } f(x, y) \leq T \\ R_2 & \text{se } f(x, y) > T \end{cases}$$

onde R_1 e R_2 são os valores atribuídos aos pixels da imagem binarizada ($R_1 = 0$ e $R_2 = 255$) com base na relação entre o limiar (T) e os seus valores em níveis de cinzento.

Como referido, através da análise do histograma com l níveis de cinzento, destacam-se duas classes (o fundo e o objecto) que vamos denotar como C_0 e C_1 e o valor do limiar óptimo t (nível de cinzento que separa o fundo do objecto).

Então temos,

$$C_0 = \{0, 1, 2, 3, 4, \dots, t\}$$

e

$$C_1 = \{t + 1, t + 2, t + 3, t + 4, \dots, l\}$$

O limiar óptimo pode ser obtido através da maximização da seguinte função:

$$\eta = \frac{\sigma_b^2}{\sigma_t^2}$$

onde σ_b^2 é a variância entre classes, σ_t^2 é a variância total e o limiar óptimo é obtido quando encontrado o valor de t que maximiza η .

$$\sigma_t^2 = \sum_{i=0}^{l-1} (i - \mu t)^2 P_i$$

$$\mu t = \sum_{i=0}^{l-1} i P_i$$

$$\sigma_b^2 = \omega_0 \omega_1 (\mu_0 - \mu_1)^2$$

$$\omega_0 = \sum_{i=0}^t P_i$$

$$\omega_1 = 1 - \omega_0$$

$$\mu_0 = \frac{\mu_1}{\omega_0}$$

$$\mu_1 = \frac{\mu - \mu_1}{1 - \omega_0}$$

$$\mu_t = \sum_{i=0}^t i P_i$$

$$P_i = \frac{n_i}{n}$$

onde P_i é a probabilidade de ocorrência do nível de cinzento i ; n_i é o número de pixéis com o nível de cinzento i ; e n é o número total de pixéis da imagem, definido como:

$$n = \sum_{i=0}^{l-1} n_i$$

Para a selecção do limiar óptimo t de uma imagem, a classe de probabilidades ω_0 e ω_1 indicam as áreas ocupadas pelas classes C_0 e C_1 no histograma. As médias das classes μ_0 e μ_1 servem como estimativa dos níveis médios das classes na imagem em

níveis de cinzento. Em síntese, o valor máximo de η pode ser utilizado como medida de separação das duas classes C_0 e C_1 e é unicamente compreendido no intervalo:

$$0 \leq \eta \leq 1$$

A utilização do método de Otsu no presente trabalho revelou-se eficaz na binarização da imagem apresentando uma boa segmentação do objecto e do fundo, o que permitiu obter as condições necessárias para passar à próxima fase do processo.

2.3. Características da imagem

Em visão por computador, o termo *características de uma imagem* refere-se a dois possíveis significados: (i) uma propriedade global à imagem ou a parte dela, por exemplo o nível médio de cinzentos ou a área em pixéis na imagem (característica global); ou (ii) uma parte da imagem com propriedades especiais, por exemplo uma circunferência, uma linha ou uma região com textura (característica local) [16].

A sequência de operações, na maioria dos sistemas de visão por computador, começa por detectar e localizar algumas características importantes na imagem de entrada. Neste trabalho vamos-nos centrar na segunda definição acima referida, demonstrando como são detectadas e filtradas características locais, como os pontos e contornos de um marcador/padrão.

Detecção de Contornos

Um contorno é uma lista de ponto que representa, de uma maneira ou de outra, uma curva na imagem. A representação pode ser diferente dependendo da situação. No nosso caso os contornos são representados como uma sequência de pontos em que cada entrada contém informação da localização do próximo ponto da curva.

A função usada para detectar os contornos [17] requer uma imagem binária em que os contornos estejam implícitos como transições entre branco e preto (Figura 6). Com base nestas transições a função computa os contornos existentes na imagem de forma sequencial, e guarda-os numa estrutura em lista [18].

$$primeiro = c01000 \leftrightarrow c01001 \leftrightarrow h0100 \leftrightarrow c010 \leftrightarrow c000 \leftrightarrow h00 \leftrightarrow c0$$

Estes contornos são designados por cX ou hX , onde c é representado para transições de preto para branco, h para transições de branco para preto e o X representa um número hierárquico.

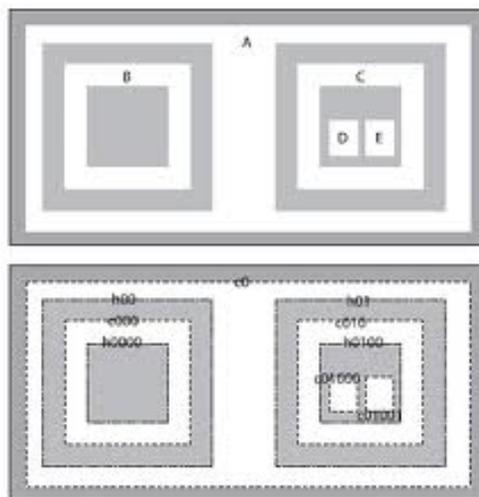


Figura 6 – Representação dos contornos de uma imagem.

Alguns destes contornos estão representados por linha contínuas, estas representam os limites exteriores das zonas pretas. Os restantes contornos representam os limites exteriores das zonas brancas (ou contornos interiores das zonas pretas).

Esta distinção entre contornos exteriores e interiores (isto é, branco para preto e preto para branco) é importante, uma vez que neste trabalho estamos interessados na pesquisa de marcadores que, na sua natureza, têm contornos dentro de outros contornos. Um exemplo de possíveis marcadores, na figura acima apresentada, poderia ser descrito pelos pares de contornos (interior e exterior):

$$\text{marcador 1} = c000 \leftrightarrow h00$$

$$\text{marcador 2} = c010 \leftrightarrow h01$$

Com o concluir desta fase obtemos uma sequência de todos os contornos detectados, onde estão incluídos os possíveis marcadores na imagem que resulta do processo de binarização. Com esta lista de contornos é possível passar à fase seguinte do processo, designadamente a selecção e filtragem dos contornos que pertencem aos marcadores.

Detecção de Marcadores

Esta fase consiste em percorrer a lista de contornos já detectados e seleccioná-los de acordo com determinadas condições, que estão relacionadas com a natureza física dos marcadores. Como cada contorno, por sua vez, é descrito por uma sequência de pontos, inicia-se o processo de triagem ao excluir todos aqueles que não são constituídos por quatro pontos, logo, quatro lados.

De seguida, a partir destes quatro pontos pertencentes ao contorno é calculado o seu centro:

$$c_x = \frac{p1_x + p2_x + p3_x + p4_x}{4}$$

$$c_y = \frac{p1_y + p2_y + p3_y + p4_y}{4}$$

onde o ponto central é denotado por (c_x, c_y) e os quatro pontos testados por (pN_x, pN_y) , com $N = 1, 2, 3, 4$.

Deste modo, a cada contorno fica associado o seu centro, sendo comparados entre si de modo a seleccionar pares com o mesmo centro, ou seja, um conjunto de oito pontos.

2.4. Posição e Orientação da Câmara

Uma vez que os marcadores já foram detectados segue-se a fase final do processo, o cálculo da posição e orientação dos marcadores em relação à câmara de vídeo, com base nos oito pontos dos contornos dos marcadores.

Para abordar este processo é importante que a calibração da câmara tenha sido previamente realizada. O método para a calibração da câmara encontra-se já implementado no OpenAR (ISR/DEEC) e é essencial, uma vez que nos fornece os parâmetros intrínsecos da câmara. Como estes são parâmetros físicos da câmara apenas é necessário calcular uma vez porque são fixos e não se vão alterar. Pelo contrário, é necessário calcular os parâmetros extrínsecos da câmara em cada imagem da sequência porque a posição desta em relação aos marcadores varia [19]. Com

efeito, a projecção resulta da combinação das matrizes dos parâmetros intrínsecos (K) com a dos parâmetros extrínsecos ($[R|t]$).

$$P = K [R|t]$$

A calibração da câmara tem como objectivo calcular o valor de cada um dos parâmetros intrínsecos. Para realizar este processo usa-se uma grelha (formato xadrez) com um conjunto de pontos cuja posição no espaço e número são conhecidos. São capturadas várias imagens desta grelha de diferentes perspectivas para melhorar o resultado e obter assim os valores dos parâmetros intrínsecos e dos parâmetros de distorção da câmara [19]. Esta informação fica armazenado e sempre disponível quando necessário.

Com a combinação entre a informação dos pontos de interesse conhecidos do modelo físico do marcador e os pontos adquiridos através de técnicas de Visão por Computador referidas anteriormente é possível determinar a transformação projectiva relativa entre o marcador e a câmara de vídeo [19].

Como se pode observar na Figura 7, existe uma transformação dos pontos do modelo físico do quadrado (à esquerda) para os pontos adquiridos do mesmo quadrado no plano imagem (à direita).

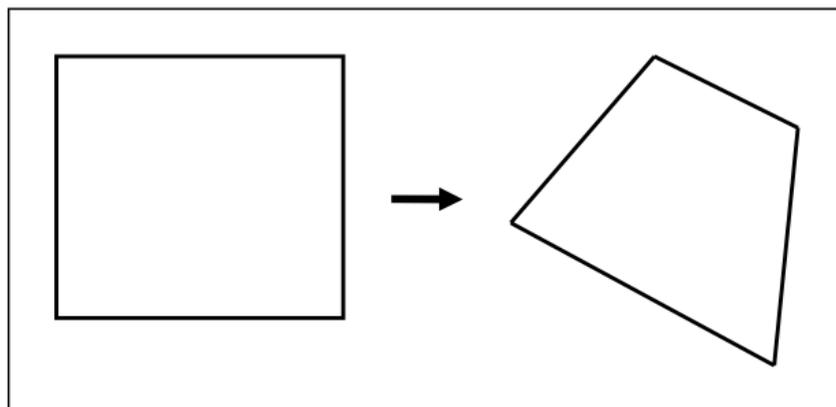


Figura 7 – Modelo físico (esquerda) e respectivo modelo imagem (direita).

Nesta fase é importante falar do modelo de câmara *pinhole* [19]. Este modelo é uma boa aproximação do modelo de câmara convencional e consiste numa caixa cúbica, fechada, com um pequeno orifício numa das faces por onde os raios de projecção vão passar e registar o seu valor no interior da caixa, na face oposta ao orifício (Figura 8).

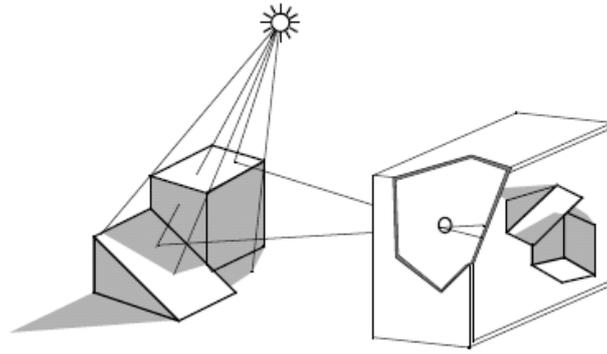


Figura 8 – Ilustração do modelo de câmera *pinhole*.

Ou seja, a cena é registada através da projecção de pontos 3D do mundo real no plano imagem usando uma transformação perspectiva.

O modelo geométrico deste tipo de câmaras incide-se essencialmente em dois parâmetros, a distância focal (f_x e f_y) e o ponto principal (c_x e c_y) que representam os parâmetros intrínsecos da câmara (K).

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

O f_x e f_y correspondem à distância focal (f), distância que separa o plano imagem do plano focal. O c_x e c_y representam o ponto principal (c), ponto de intersecção do eixo óptico com o plano imagem, o eixo óptico é um raio perpendicular ao plano imagem (Figura 9).

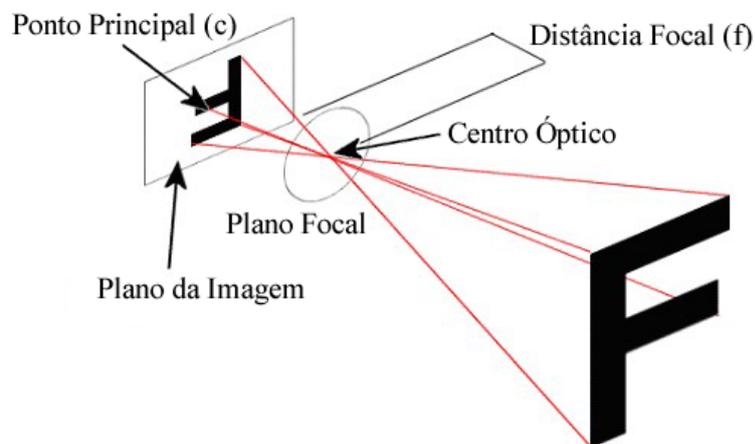


Figura 9 – Ilustração da distância focal (f) e do ponto principal (c).

Este processo pode ser alcançado através das seguintes equações [2, 19]:

$$s m = K [R|t] M$$

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Onde X , Y e Z são as coordenadas do ponto 3D no referencial do mundo (M), u e v são as coordenadas do ponto projectado no plano imagem (m), em pixéis. Como já vimos, K , é a matriz dos parâmetros intrínsecos, com f_x e f_y para a distância focal e com c_x e c_y para ponto principal da câmara. Se à imagem for aplicado algum factor de escala à imagem, todos estes parâmetros devem ser escalados pelo mesmo factor. A matriz dos parâmetros intrínsecos não depende da cena que está a ser observada, uma vez estimada (através do método de calibração da câmara), esta matriz pode ser reutilizada. À combinação das matrizes de rotação e o vector de translação ($[R|t]$) dá-se o nome de matriz dos parâmetros extrínsecos da câmara. Esta é usada, em conjunto com a matriz K , para descrever o movimento da câmara em relação a uma cena estática ou de um objecto rígido (marcador) em relação à câmara estática. Isto é, $K[R|t]$ traduz as coordenadas de um ponto 3D no mundo (X, Y, Z) para um outro sistema coordenado com base nos parâmetros da câmara.

Esta transformação é equivalente à seguinte forma (para $z = 0$):

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t$$

$$x' = x/z$$

$$y' = y/z$$

$$u = f_x * x' + c_x$$

$$v = f_y * y' + c_y$$

Como as lentes de uma câmara real normalmente contêm alguma distorção, principalmente distorção radial mas também alguma distorção tangencial, o modelo evolui para a seguinte forma:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t$$

$$x' = x/z$$

$$y' = y/z$$

$$x'' = x'(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1 x' y' + p_2 (r^2 + 2x'^2)$$

$$y'' = y'(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + p_1 (r^2 + 2y'^2) + 2p_2 x' y'$$

onde,

$$r^2 = x'^2 + y'^2$$

$$u = f_x * x'' + c_x$$

$$v = f_y * y'' + c_y$$

Os valores k_1 , k_2 e k_3 representam coeficientes de distorção radial, enquanto que o p_1 e p_2 são os coeficientes de distorção tangencial. Estes valores são calculados no processo de calibração da câmara e apresentados na forma de vector.

$$(k_1, k_2, p_1, p_2, [k_3])$$

Este vector contém quatro elementos o que significa que o $k_3 = 0$. Estes coeficientes também não dependem da cena que está a ser observada e pertencem aos parâmetros físicos da câmara.

Uma vez calculados os parâmetros intrínsecos (K) e os coeficientes de distorção da câmara,

$$s m = K [R|t] M$$

e assumindo que o modelo físico do marcador, que define um plano, encontra-se em $Z = 0$ no sistema de coordenados do mundo (por esse motivo a última coluna da matriz de rotação, r_3 , não contribui para o resultado), obtém-se a seguinte equação [20]:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$$

$$= K \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

Assim, continuam a ser usados os pontos do modelo físico (M), mas agora descritos por $M = [X, Y]$, uma vez que, Z é sempre igual a zero. Então, $\tilde{M} = [X, Y, 1]$.

Nesta fase, é possível afirmar que pontos do modelo físico (M) e pontos imagem (m) estão relacionados por uma homografia (H):

$$s\tilde{m} = H\tilde{M}$$

com,

$$H = K \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}$$

A homografia (M) é uma matriz 3×3 definida a menos de um factor de escala.

$$H = \begin{bmatrix} h_0 & h_1 & h_2 \\ h_3 & h_4 & h_5 \\ h_6 & h_7 & h_8 \end{bmatrix}$$

Com os pontos do modelo físico de um marcador ($\tilde{M} = [X, Y, W]$) e os seus correspondentes no plano imagem ($\tilde{m} = [u, v]$), é possível estimar a matriz de homografia resolvendo o sistema $Lx = 0$. L é uma matriz constituída por todos os pontos (\tilde{M} e \tilde{m}) da seguinte forma:

$$L = \begin{bmatrix} \tilde{M}_1^T & 0^T & -u_1 \tilde{M}_1^T \\ 0^T & \tilde{M}_1^T & -v_1 \tilde{M}_1^T \\ \tilde{M}_2^T & 0^T & -u_2 \tilde{M}_2^T \\ 0^T & \tilde{M}_2^T & -v_2 \tilde{M}_2^T \\ \vdots & \vdots & \vdots \\ \tilde{M}_n^T & 0^T & -u_n \tilde{M}_n^T \\ 0^T & \tilde{M}_n^T & -v_n \tilde{M}_n^T \end{bmatrix}$$

ou seja,

$$L = \begin{bmatrix} X_1 & Y_1 & W_1 & 0 & 0 & 0 & -u_1X_1 & -u_1Y_1 & -u_1W_1 \\ 0 & 0 & 0 & X_1 & Y_1 & W_1 & -v_1X_1 & -v_1Y_1 & -v_1W_1 \\ X_2 & Y_2 & W_2 & 0 & 0 & 0 & -u_2X_2 & -u_2Y_2 & -u_2W_2 \\ 0 & 0 & 0 & X_2 & Y_2 & W_2 & -v_2X_2 & -v_2Y_2 & -v_2W_2 \\ & & & \vdots & & & & & \\ X_n & Y_n & W_n & 0 & 0 & 0 & -u_nX_n & -u_nY_n & -u_nW_n \\ 0 & 0 & 0 & X_n & Y_n & W_n & -u_nX_n & -u_nY_n & -u_nW_n \end{bmatrix}$$

e x é a variável, ou seja, os oito valores que compõem a matriz de homografia (H), denotado da seguinte forma:

$$x = \begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \end{bmatrix}$$

estes resultados são obtidos através da solução dos mínimos quadrados, onde, o mínimo de pontos para resolução do sistema é quatro e quanto maior for o número de correspondências (pontos do modelo e pontos da imagem) menor será o erro na estimação da matriz de transformação.

Após estimar a matriz de homografia (H) entra a última fase deste processo, ou seja, o cálculo da rotação e translação que descrevem a transformação entre os pontos modelo e os pontos imagem.

Retomando a equação:

$$H = K[r_1 \quad r_2 \quad t]$$

e como já foram calculados os parâmetros intrínsecos da câmara (K), então,

$$H = [h_1 \quad h_2 \quad h_3] = K[r_1 \quad r_2 \quad t]$$

usando o facto de que r_1 e r_2 são ortogonais, obtém-se a seguinte relação:

$$h_1^t K^{-t} K^{-1} h_2 = 0$$

$$h_1^t K^{-t} K^{-1} h_1 = h_2^t K^{-t} K^{-1} h_2$$

ou seja,

$$r_1 = K^{-1}h_1$$

$$r_2 = K^{-1}h_2$$

$$r_3 = r_1 \times r_2$$

$$t = K^{-1}h_3$$

assim obtêm-se os parâmetros extrínsecos compostos pela rotação ($R = [r_1 \ r_2 \ r_3]$) e translação (t).

Com base nestes conceitos o sistema será capaz de: (i) projectar pontos 3D do mundo no plano imagem; (ii) estimar a posição dos marcadores no referencial da câmara a partir dos parâmetros intrínsecos, alguns pontos 3D conhecidos no mundo e as suas respectivas projecções; (iii) estimar os parâmetros intrínsecos e extrínsecos da câmara através de várias perspectivas de um marcador de calibração (cada perspectiva é descrita pela correspondência entre pontos 3D e 2D); e (iv) calcular a pose da câmara em relação a um objecto no mundo real e vice-versa, em tempo real.

3. Desenvolvimento e Resultados

Os cinco sentidos (visão, audição, tacto, olfacto e paladar) fundamentais do corpo humano formam um conjunto de funções que permite o relacionamento dos utilizadores com o ambiente. Através deles, o utilizador percebe tudo o que o rodeia e de acordo com as sensações, interage com o ambiente [3].

A Realidade Aumentada utiliza os dispositivos de saída para enviar estímulos aos sentidos humanos e os dispositivos de entrada para receber os estímulos dos utilizadores, o que possibilita a integração dos utilizadores com os mundos virtuais. Os sentidos são mais utilizados pela seguinte ordem: visão, audição, tacto, olfacto e paladar, apesar do tempo de resposta dos mesmos seguir outra ordem (sabe-se, por exemplo, que o tacto e a audição oferecem menor latência neuronal que a visão, sendo mais adequados dessa forma a estímulos que necessitem de resposta rápida). A Realidade Aumentada visa envolver todos os sentidos do ser humano de forma combinada [21].

Os diversos dispositivos de entrada/saída procuram tornar o ambiente virtual o mais natural possível. Os ambientes de Realidade Aumentada podem ser divididos em quatro partes básicas: (i) um Ambiente Físico (iluminação); (ii) um Sistema Computacional de Processamento (*hardware*); (iii) um Sistema de Visualização (saída: vídeo, áudio, etc.); e (iv) um Sistema de Interação (entrada: câmaras, *joysticks*, etc.) [21]. Para a integração das três últimas partes, é necessário um sistema de *software* que una os sistemas de forma apropriada. O OpenAR é o sistema que está a ser desenvolvido no ISR (DEEC, UC) com esse objectivo. O seu desenvolvimento é maioritariamente C/C++ e as principais bibliotecas que suporta são: o OpenGL, para criar e manipular ambientes virtuais 3D; o OpenCV, para aceder e manipular imagens capturadas pela câmara; o Bullet Physics, para integrar física nos ambientes virtuais; o SDL, para interagir através do rato e teclado; e o TCL/TK, linguagem de programação com suporte para criar interfaces gráficos.

Esta tese contribui com a adição de novas funcionalidades ao OpenAR, nomeadamente: binarizar imagens através do método de Otsu, detectar diferentes formas geométricas dada uma imagem, detectar marcadores binários e extrair a sua posição em relação à câmara real, bem como, a outros marcadores.

3.1. Binarização da Imagem

O método implementado (binarização de Otsu [13]) apresenta resultados muito positivos face às variações de luz ao longo da sequência de imagens. A imagem capturada pela câmara é enviada para a função criada para converter a imagem de entrada para preto e branco (Figura 10).

A função criada tem a seguinte forma:

bin_otsu(imgEntrada, imgSaida);

Esta função recebe uma imagem de entrada a cores (*imgEntrada*), que vai ser convertida para níveis de cinzento e através da análise do seu histograma é definido o valor do limiar a partir do qual cada pixel vai ser preto ou branco. Esta informação é guardada na imagem de saída (*imgSaida*).

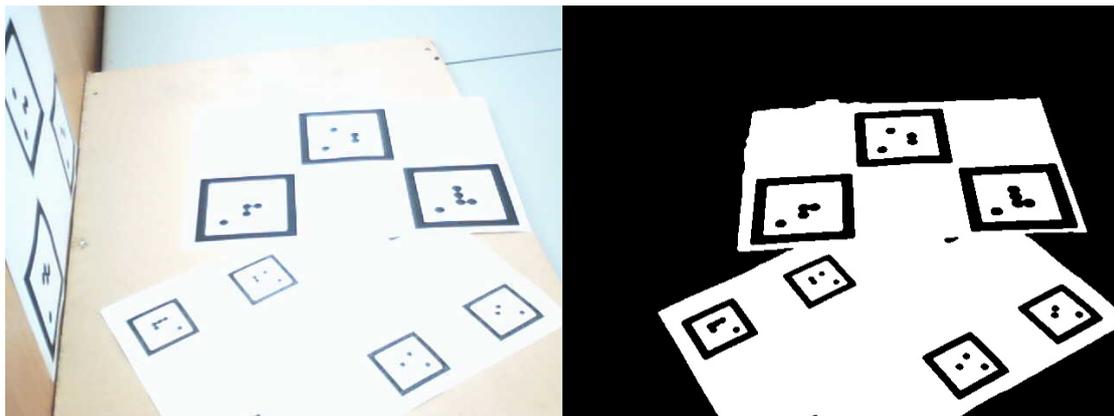


Figura 10 – Resultado da binarização de Otsu.

Esta fase é muito importante porque deixa a imagem pronta para a próxima etapa (detectar contornos), pois ao converter imagem para preto e branco vai realçar os contornos que estão implícitos como transições de preto para branco e de branco para preto.

A escolha desta técnica deve-se ao facto de apresentar melhores resultados na binarização da imagem capturada pela câmara em comparação a outros métodos e técnicas testadas, tal como a binarização da imagem utilizando um limiar estático (insensível a variações de iluminação ao longo da sequência de imagens).

3.2. Detecção de Contornos

Como muitas vezes existe ruído nas imagens, a filtragem de dados é essencial para seleccionar apenas os contornos importantes. Neste trabalho como os marcadores são quadrados o objectivo foi de isolar todos os quadriláteros na imagem.

Com todos os contornos listados entra num processo iterativo que vai percorrer e testar se são quadriláteros, respeitando algumas condições. A primeira é que o contorno tem de ter quatro vértices após a aproximação (método de aproximação de polígonos), a segunda é que a área tem de ser superior ou igual a 100 px² e a terceira é que seja convexo. Com estas condições satisfeitas obtemos o conjunto de quadriláteros detectados na imagem de entrada e termina a primeira fase de detecção de características.

Para a realização desta etapa foi criada uma função designada:

find_squares(img);

Esta função recebe como parâmetro de entrada a imagem original (*img*) e retorna uma sequência de todos os quadriláteros que correspondem às condições acima descritas.

Primeiro é necessário que a imagem esteja a preto e branco, por isso inicialmente é chamada a função criada para binarizar a imagem através do método de Otsu. Uma vez tendo a imagem a preto e branco é possível usar a função *cvFindContours* (OpenCV), cujo resultado vai ser uma lista de todos os contornos detectados na imagem. Esta função recorre às transições entre preto e branco para definir os contornos. Como nem todos os contornos pertencem a marcadores na imagem é necessário fazer uma selecção, para tal, impomos algumas condições. A primeira e mais importante é que o contorno seja um quadrilátero uma vez que os marcadores apresentam uma forma quadrangular. Para solucionar este problema passamos a sequência de contornos para outra função, *cvApproxPoly* (OpenCV), esta vai percorrer contorno a contorno e aproximá-lo a um polígono mais simples. Como explicado na Figura 11, este processo começa com um contorno (b), depois encontra os dois pontos mais afastados e conecta-os com uma linha (c). O próximo passo é procurar o ponto do contorno original que se encontra mais afastado desta linha que acabámos de desenhar e adiciona-lo à aproximação. Este é um processo iterativo (d-

f), que passa por ir adicionando à aproximação os pontos mais distantes do contorno original até que esta distância seja inferior a um valor estipulado de precisão (f).

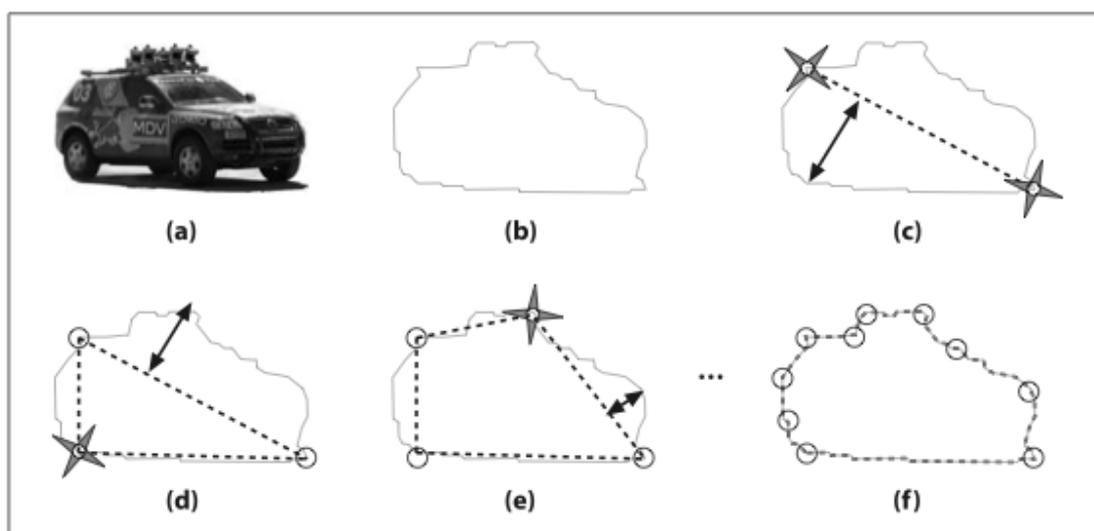


Figura 11 – Processo de aproximação de contornos a polígonos mais simples.

No final desta aproximação vamos escolher apenas os contornos importantes que representam marcadores, ou seja, que possuem quatro vértices. A segunda condição implica que a área dos contornos seja superior a 100 px^2 para que seja identificado como possível componente do marcador, sendo usada para este cálculo a função *cvContourArea* (OpenCV). Uma vez que as imagens possuem algum ruído foi necessário criar esta condição para eliminar pequenos contornos quadrangulares que são inicialmente detectados mas que não pertencem às características de um marcador e que por isso não devem ser incluídos na sequência de contornos que pertencem aos marcadores. Finalmente, a terceira e última condição deste passo exige que o quadrilátero encontrado seja convexo, ou seja, o segmento de recta que une quaisquer dois pontos desse polígono devem estar totalmente contido no seu conjunto. A função aplicada a cada contorno para obter esta informação é *cvCheckContourConvexity* (OpenCV). Esta condição revela-se igualmente essencial, pois permite detectar e remover contornos cuja detecção foi mal classificada.

A importância desta triagem de dados reside na eficácia e robustez do algoritmo de detecção. Com estas condições verificadas obtemos uma sequência de todos os quadriláteros na imagem, que são todos os possíveis marcadores na imagem (Figura 12).

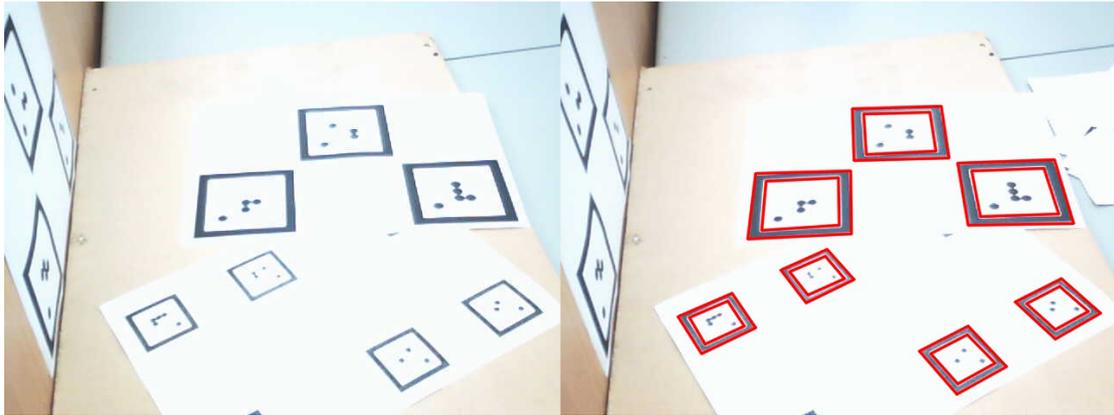


Figura 12 – Resultado dos possíveis marcadores na imagem.

Contudo, podem ainda existir contornos que não pertencem ao conjunto dos marcadores, pelo que estes dados irão ser ainda sujeitos a uma próxima fase de filtragem.

3.3. Filtragem de Dados

Neste trabalho foi implementado um método que localiza marcadores binários numa cena do mundo real para estimar a posição e orientação destes em relação à câmara de vídeo real. Os marcadores binários criados para este trabalho têm uma forma quadrada, dentro deste quadrado encontram-se várias bolas, uma num dos quatro cantos que define a orientação do marcador e as restantes distribuídas numa grelha de oito entradas que representa um código binário capaz de distinguir 2^8 (256) marcadores diferentes. Estas bolas podem ser brancas ou pretas e representam computacionalmente 0 ou 1, respectivamente. Estes marcadores podem ser impressos numa impressora normal e o sistema de Realidade Aumentada reconhece-os através das imagens capturadas pela câmara real em tempo real.

Características de um marcador

Com o concluir da fase anterior de detecção de contornos, obtemos uma lista de todos os quadriláteros encontrados na imagem de entrada. Como existem muitos quadrados nesta lista que mesmo assim não correspondem a marcadores no mundo real foi necessário encontrar uma forma de fazer a sua selecção. Para tal, o método implementado testa este volume de dados a algumas condições essenciais, uma vez

que os marcadores possuem um contorno preto num fundo branco existem duas transições branco/preto e preto/branco, logo, dois quadrados com o mesmo centro, sendo esta a primeira condição. A segunda é que dentro desses quadrados exista apenas uma bola num dos quatro cantos (bola de orientação) e por fim o código binário identificado pelas restantes bolas deve estar compreendido entre 0 e 255 (quando convertido para número decimal).

Detecção e Identificação dos Marcadores

Nesta fase, de todos os quadriláteros encontrados na imagem de entrada apenas são considerados possíveis marcadores os pares de quadriláteros com o mesmo centro. Através das coordenadas dos quatro cantos no plano imagem é calculado o seu centro, sendo comparados entre si e formados os pares, assumindo assim o cumprimento da primeira condição (Figura 13).

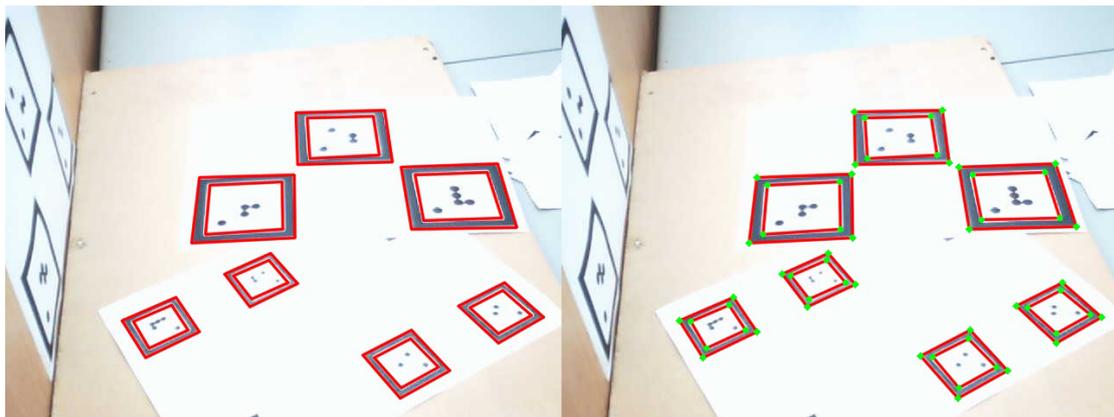


Figura 13 – Marcadores na imagem (esquerda) e os respectivos cantos (direita).

Como os oito pontos dos dois quadrados de cada marcador são conhecidos, a identificação da cor das bolas e a sua respectiva localização é feita através do mapeamento de dois quadrados (oito pontos no modelo físico do marcador) em dois quadriláteros (oito pontos dos dois quadriláteros com o mesmo centro na imagem capturada), a este processo chama-se mapeamento projectivo (que será abordado mais detalhadamente no tópico seguinte) (Figura 14).

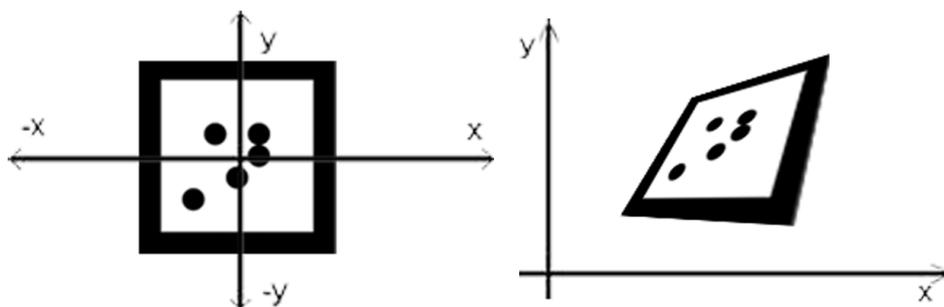


Figura 14 – Modelo físico do marcador (esquerda) e sua representação no plano imagem (direita).

O princípio base deste processo consiste em calcular a matriz de homografia entre os pontos do modelo físico do marcador e os seus correspondentes no plano imagem. Usando esta matriz, são mapeadas as posições dos pontos correspondentes a possíveis posições das bolas no plano imagem, verificando assim se existe ou não um círculo correspondente. O sistema começa por mapear as quatro possíveis posições onde uma única bola de orientação poderá estar e testada a cor nesses pixéis, sendo que apenas uma deverá ser preta para ser respeitada a segunda condição (Figura 15, à esquerda). Para concluir a presença de um marcador são mapeadas as restantes oito posições dos pontos da grelha no plano imagem e testada a cor nesses pixéis, sendo que a cor branca corresponde ao número 0 e a cor preta ao número 1 (Figura 15, à direita). O conjunto desta sequência de oito números representa um código binário compreendido entre 0 e 255 (quando convertido para número decimal), que vai ser o número de identificação do marcador.

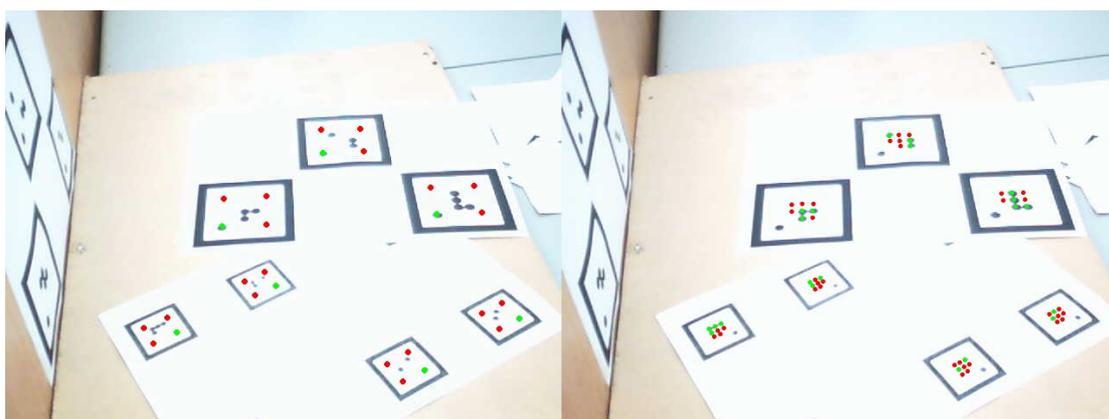


Figura 15 – Localização do ponto de orientação (esquerda) e identificação do marcador (direita).

Como um dos objectivos é que o sistema seja capaz de detectar múltiplos marcadores é necessário que cada um seja único e ao mesmo tempo que mantenham as

características principais para ser considerado como marcador. A identificação e rotulação dos marcadores são feitas através de um código binário, o que os distingue entre si de forma computacionalmente rápida.

Para realizar esta fase do processo foi implementada a seguinte função:

detectMarker(img);

Esta função começa por chamar a função *findSquare* (referida anteriormente), responsável por detectar e filtrar inicialmente os contornos, guardando-os numa sequência de dados. De seguida, testa novamente este volume de dados às condições exigidas relativamente as características do marcador. O resultado desta função é uma estrutura que guarda toda a informação sobre os marcadores, nomeadamente as coordenadas dos quatro pontos do quadrado externo, as coordenadas dos quatro pontos do quadrado interno, o centro do marcador, a translação e rotação do marcador (informação que será melhor detalhada de seguida).

3.4. Extracção da Posição e Orientação

Com os marcadores devidamente identificados na imagem passamos a esta última fase do processo que consiste em obter a translação e rotação do marcador em relação à câmara no mundo real e com a operação inversa posicionar a câmara virtual no mesmo local da câmara real, definindo assim graficamente o centro de coordenadas no centro do marcador no mundo real (Figura 16). Este passo é importante para que os modelos gráficos que vão ser aumentados (colocados sobre a imagem real) coincidam com a escala e orientação do marcador detectado.



Figura 16 – Pose do marcador (esquerda) aplicada ao objecto virtual e à câmara virtual (direita).

Nesta fase, como a calibração da câmara já foi realizada no início do processo, já temos os parâmetros intrínsecos e os coeficientes de distorção da câmara, valores essenciais para prosseguir. Os parâmetros intrínsecos estão representados numa matriz 3x3 com os valores da distância focal e o ponto principal da câmara, que são dados físicos da câmara, sendo apenas necessário fazer esta calibração uma vez. O método para realizar a calibração está implementado no OpenAR (ISR/DEEC).

Uma vez que na determinação da posição do marcador em relação à câmara temos disponível a informação das coordenadas 3D dos pontos no modelo físico do marcador e dos pontos correspondentes no plano imagem, foi usada uma função que estima a translação e rotação de um marcador, representada da seguinte forma:

cvFindExtrinsicCameraParams2(objpts, imgpts, IntrinsicMatrix, DistortionCoefficients, rotation, translation);

Esta função tem como parâmetros de entrada os pontos do modelo físico do marcador (*objpts*), os seus correspondentes no plano imagem (*imgpts*), a matriz dos parâmetros intrínsecos (*IntrinsicMatrix*) e os coeficientes de distorção (*DistortionCoefficients*). Desta função resultam dois vectores, um de rotação e outro de translação, que minimizam o erro de reprojecção (ou seja, a soma do quadrado das distâncias entre os pontos do plano imagem e os pontos do modelo físico do marcador).

Com estes vectores obtemos finalmente a localização do marcador relativamente à posição da câmara, ou seja temos a transformação que faz a ligação entre o mundo real e o mundo virtual, permitindo o perfeito alinhamento entre os objectos virtuais e objectos reais (marcadores).

A partir do momento em que chegamos a esta etapa abrem-se inúmeras possibilidades de trabalhar a Realidade Aumentada. Podem-se enumerar algumas delas: obter a posição relativa entre diferentes marcadores na cena; adquirir a posição da câmara real em relação à localização do marcador e assim posicionar a câmara virtual no mesmo local da câmara real; projectar pontos do modelo físico no plano imagem e vice-versa. Esta última possibilidade é realizada na fase de detecção de marcadores, em particular quando é testada a localização do ponto de orientação e dos pontos de identificação que constituem o código binário do marcador. Para a projecção dos pontos do marcador no plano imagem é usada a seguinte função:

cvProjectPoints2(objpts, rotation, translation, IntrinsicMatrix, DistortionCoefficients, imgpts);

Esta função projecta pontos do modelo físico do marcador (*objpts*) no plano imagem tendo em conta a sua rotação, translação, matriz dos valores intrínsecos e coeficientes de distorção da câmara, previamente calculados, resultando os pontos correspondentes no plano imagem (*imgpts*). Desta forma, torna-se trivial a obtenção da cor do pixel na imagem dado qualquer ponto do marcador que pretendemos testar.

3.5. Predictores de Movimento

Uma vez que neste trabalho estamos perante um sistema de Realidade Aumentada que funciona em tempo real e, portanto, com uma grande sequência contínua de imagens capturadas pela câmara, é fundamental e necessário garantir a fluidez do movimento dos objectos virtuais que estão a ser sobrepostos nos marcadores reais. Como por vezes a detecção da localização dos marcadores falha, devido, por exemplo, a movimentos bruscos do marcador ou da câmara, à saída do marcador do campo de visão da câmara ou a mudanças repentinas de iluminação, foi necessário prever o movimento dos marcadores com base nas suas trajectórias anteriores. Um preditor desta natureza estima o movimento baseando-se em pontos de referência (posição do marcador) das últimas imagens onde o marcador foi detectado. Assim, quando o marcador se encontra no campo de visão da câmara e é detectado, o sistema atribui a posição do marcador ao objecto virtual, mas quando a detecção falha a posição atribuída ao objecto é fornecida pelo preditor, permitindo o movimento contínuo do objecto na cena.

O filtro de Kalman [22] foi a técnica usada neste trabalho para prever o movimento dos marcadores. Este filtro é composto por um conjunto de equações matemáticas computacionalmente eficientes e permitem estimar o estado do processo. O filtro é bastante eficaz em vários aspectos, na medida em que é capaz de estimar o estado do objecto no passado, presente e futuro.

A implementação deste filtro está dividida em duas etapas [23]. A primeira etapa só é executada uma vez quando o marcador, que se pretende estimar o movimento, surge no campo de visão da câmara pela primeira vez. Nesta etapa é guardada a posição do

marcador (*pose*) e criado o modelo para a predição do movimento. Este passo é executado através da seguinte função:

initKalman(pose);

Na segunda etapa a posição do marcador é actualizada sempre que o marcador é detectado na imagem (*lastPose*), fornecendo informação útil ao preditor para estimar a trajectória do marcador, com base nas suas posições anteriores, quando este não é detectado. Para este passo criou-se a seguinte função:

updateKalman(Lastpose);

Esta função retorna a próxima posição estimada com base nas posições anteriores do marcador.

Sintetizando, quando a imagem é capturada inicia-se o processo de pesquisa da posição do marcador que quando este é detectado a posição é atribuída ao objecto virtual que lhe corresponde e mostrada imagem. No caso de o marcador não ser detectado recorre-se ao resultado fornecido pelo preditor para atribuir a posição ao objecto virtual. Através deste processo obtemos um movimento fluído e contínuo, dando a sensação de que a detecção do marcador nunca falhou. Este processo encontra-se esquematizado na Figura 17.

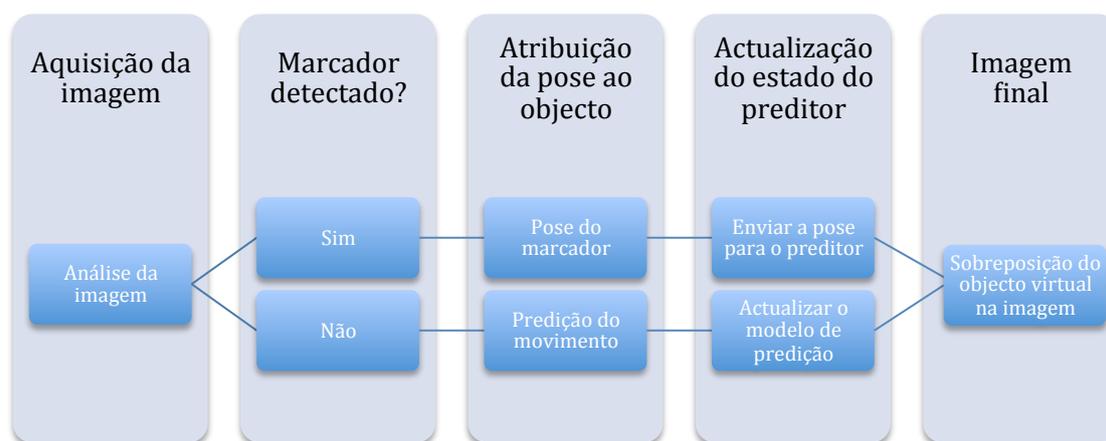


Figura 17 – Processo de funcionamento do preditor de movimento.

3.6. Demonstradores

Esta é a secção onde são apresentados os resultados práticos do desenvolvimento desta tese. Com o contributo desta, foram desenvolvidos vários demonstradores de Realidade Aumentada.

As aplicações que se seguem demonstram a aplicação e utilidade da Realidade Aumentada, em diferentes áreas, como a interacção homem-máquina, o entretenimento, a educação, a medicina/psicologia e a indústria. Estas foram as áreas abordadas para demonstrar o conceito, no entanto, o sistema deixa em aberto e disponível a possibilidade de criar aplicações em qualquer outro domínio. No Apêndice A encontra-se descrito o processo para criar uma aplicação simples de Realidade Aumentada, em que são explicadas passo-a-passo as diferentes fases do processo, bem como as funções para as executar.

Relativamente às aplicações desenvolvidas, a Figura 18 apresenta a primeira demonstração do sistema a funcionar com múltiplos marcadores em cena. Esta foi uma fase importante, na medida em que com a implementação do código binário no interior de cada marcador passou a ser possível atribuir um número de identificação que os distingue, ou seja, cada objecto virtual pode ser associado a um marcador específico e acompanhá-lo.

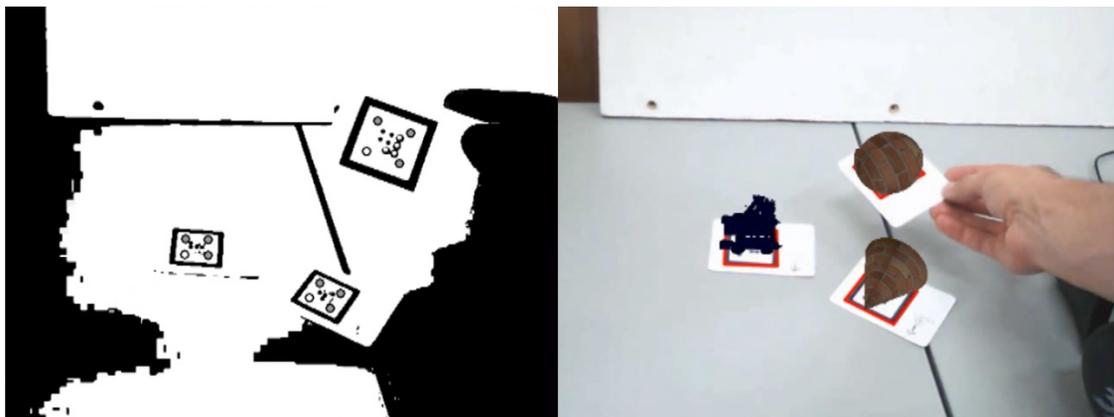


Figura 18 – Múltiplos marcadores identificados (esquerda) e objectos associados (direita).

Demonstrações de Interação

Os exemplos que se seguem são demonstrações de como a interação entre o utilizador e o computador pode ser útil, ao facilitar a manipulação de objectos virtuais através de acções do utilizador.

A sequência de imagens que se segue demonstra como o objecto pode ser rodado ou a sua escala alterada simplesmente através da oclusão dos marcadores em cena. Neste caso existem três marcadores que quando não são detectados executam diferentes acções. Como se pode observar na Figura 19, ao colocar a mão sobre o marcador à direita do utilizador o objecto vai rodar para a direita, sobre o marcador à sua esquerda vai rodar para a esquerda e sobre o que se encontra à frente vai aumentar a escala. O conceito é simples, enquanto detectados os três marcadores o sistema está estático, quando um deles deixa de ser detectado o sistema executa a ordem respectiva. Esta técnica pode ser estendida a mais marcadores, logo, a mais acções.

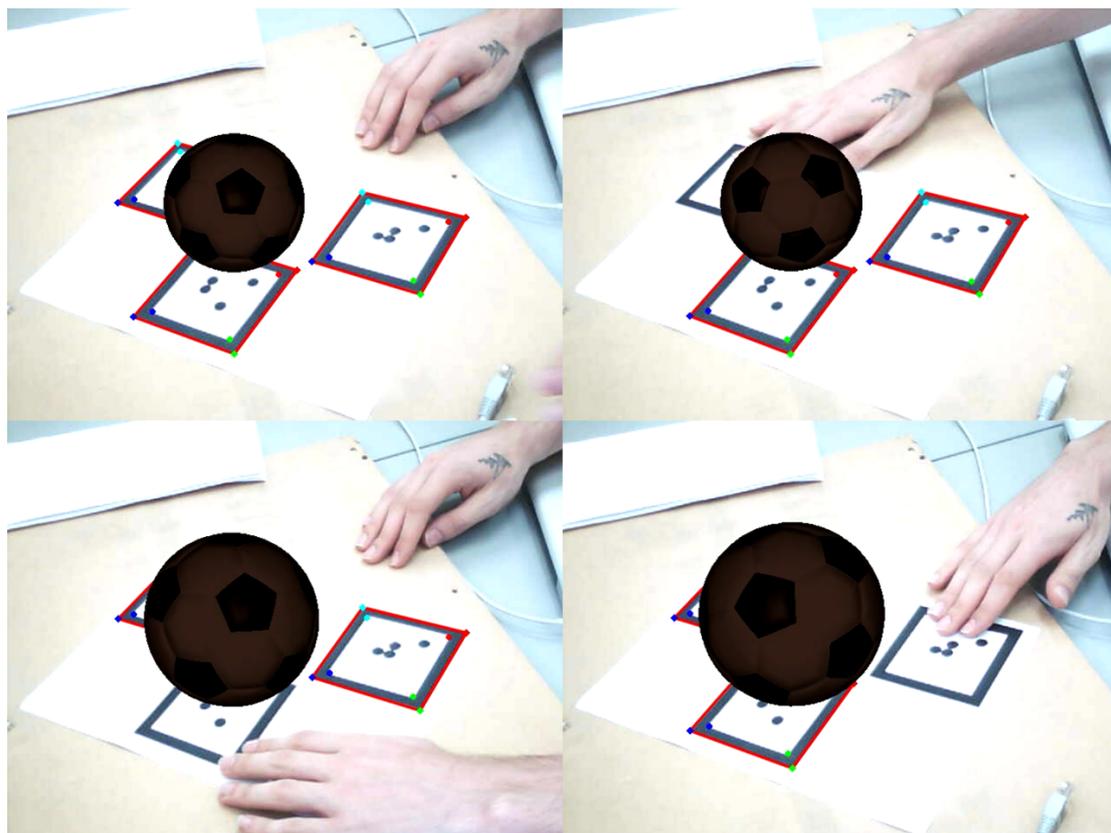


Figura 19 – Interação com base na oclusão dos marcadores.

Foi também criado um marcador especial que é na maioria das características idêntico a um marcador normal, com a excepção de que este não tem o código binário de

identificação no seu interior. É chamado de marcador zero, está impresso num quadro e pode-se escrever ou desenhar no seu interior. A informação no seu interior é rectificada e interpretada pelo sistema de Realidade Aumentada de duas maneiras distintas. A primeira para aplicar a textura nos objectos virtuais através do mapeamento directo da informação que se encontra no interior do marcador (Figura 20). Esta informação pode ser de qualquer natureza desde desenhos abstractos, a palavras ou frases.



Figura 20 – Mapeamento de texturas com base na informação do marcador.

A segunda para gerar objectos virtuais com base na forma geométrica que foi desenhada no quadro (Figura 21). Após rectificar a imagem no interior do marcador, o sistema inicia uma pesquisa nessa área por formas geométricas simples. Como demonstrado na sequência de imagens seguinte, o sistema começa por pedir ao utilizador que coloque o quadro no campo de visão da câmara (Figura 21, em cima à esquerda), uma vez localizado o quadro, o sistema detecta no seu interior um quadrado e é criado um cubo nessa posição (Figura 21, em cima à direita). Este objecto virtual é adicionado ao mundo virtual com física, ou seja, vai ser afectado pela gravidade e vai caindo em direcção ao plano da mesa (Figura 21, em baixo à

esquerda), uma vez imóvel, o sistema continua a sua rotina de pesquisa pelo quadro (Figura 21, em baixo à direita). Aqui apenas é demonstrado o exemplo do quadrado (cubo), no entanto, o sistema está preparado para detectar polígonos com qualquer número de lados.

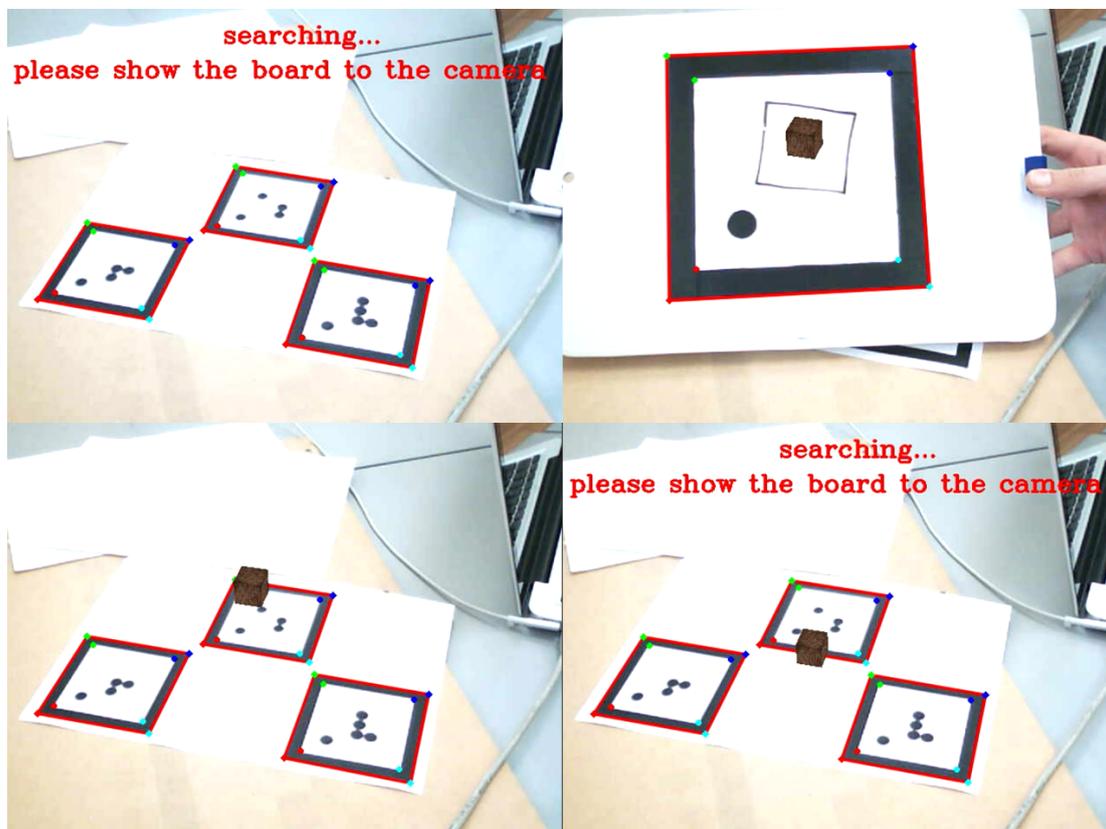


Figura 21 – Criação de um cubo com base no quadrado desenhado no marcador.

A demonstração que se segue exemplifica um método diferente das anteriores, na medida em que não usa o quadro para a interação. Na sequência de imagens seguinte encontra-se uma torre de cubos virtuais sobre a mesa e ao exercer força através da mão (murro) vamos interagindo com esses objectos (Figura 22). O conceito aqui consiste nas vibrações, que o murro na mesa produz, na posição da câmara real em relação ao marcador que está a definir o plano da mesa. Como existe simulação de física no mundo virtual o comportamento da torre de cubos virtual, face aos murros na mesa, vai ser idêntico ao de uma torre de cubos reais que ali estivesse.

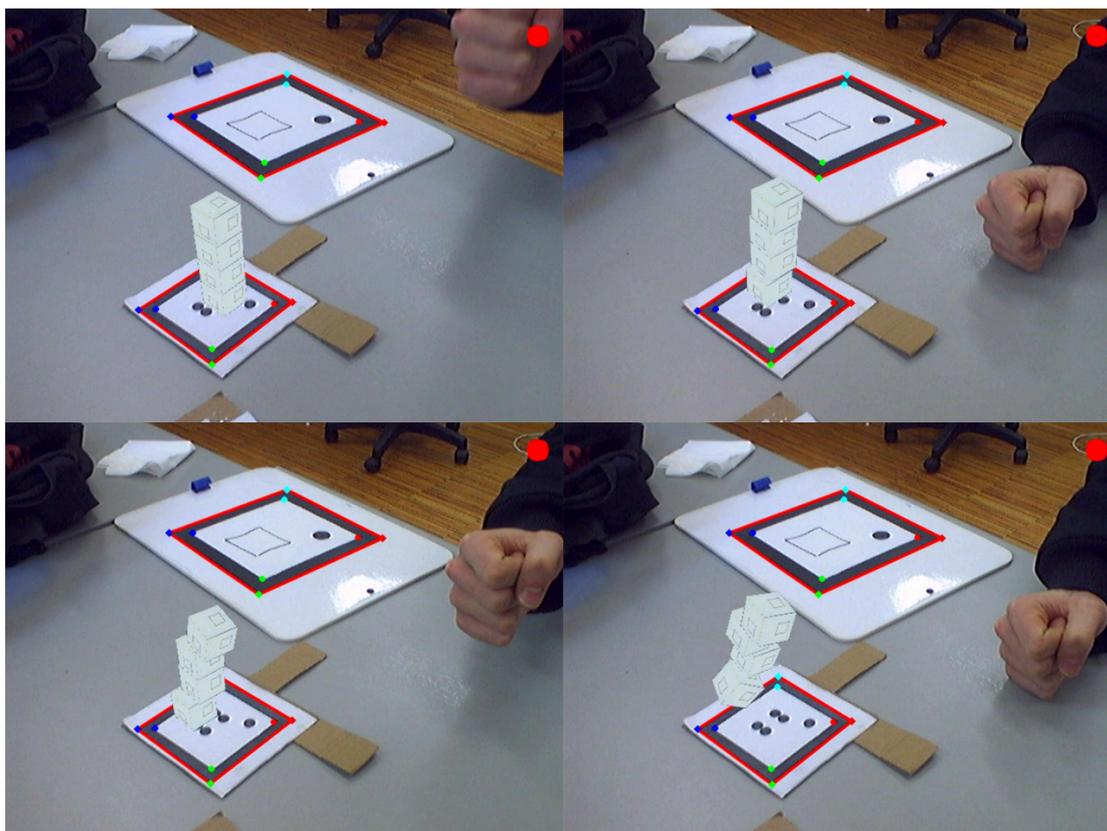


Figura 22 – Interação com objectos virtuais tirando partido da física.

Demonstrações Educacionais

O conceito que se segue demonstra como se pode auxiliar e melhorar a maneira de como o utilizador vê, compreende e aprende o mundo no âmbito escolar. Assim, ao colocar marcadores nos livros escolares para sobrepor objectos virtuais no mundo real vai ajudar na demonstração e visualização de temáticas de forma atractiva e motivadora. Nas figuras que se seguem são apresentados exemplos da aplicação de marcadores em dois livros escolares, um de física (Figura 23, à esquerda) e um de química (Figura 23, à direita). Deste modo, com o acesso a exemplos virtuais, os conteúdos teóricos tornam-se mais apelativos e interessantes, facilitando e revolucionando o processo de ensino/aprendizagem.

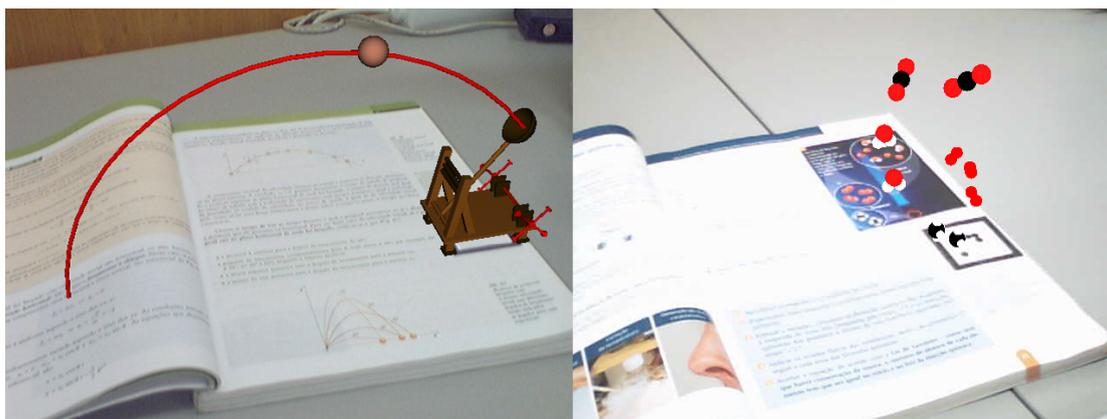


Figura 23 – Realidade Aumentada aplicada em livros escolares.

Demonstrações Médicas/Terapêuticas

Como já foi referido, também é possível aplicar a Realidade Aumentada na Medicina e na Psicologia com resultados satisfatórios quer ao nível da análise de diagnóstico (visualização do corpo humano) quer ao nível do tratamento (exposição do sujeito a situações ou animais temidos).

Uma possível aplicação na área da Medicina consiste em facilitar a localização de fracturas ósseas no corpo de um paciente com um marcador (Figura 24). Esta técnica pressupõe que seja feito, previamente, um raio-X ao paciente e que com o auxílio de um marcador, o médico vai percorrendo o corpo do paciente à procura de anomalias. A vantagem desta aplicação é que deixa de ser necessário imprimir o raio-X, pois uma vez feito, este é associado à posição do corpo do paciente. Com efeito, esta demonstração pode facilitar e tornar mais rápido o processo de análise, identificação e diagnóstico de patologias físicas.

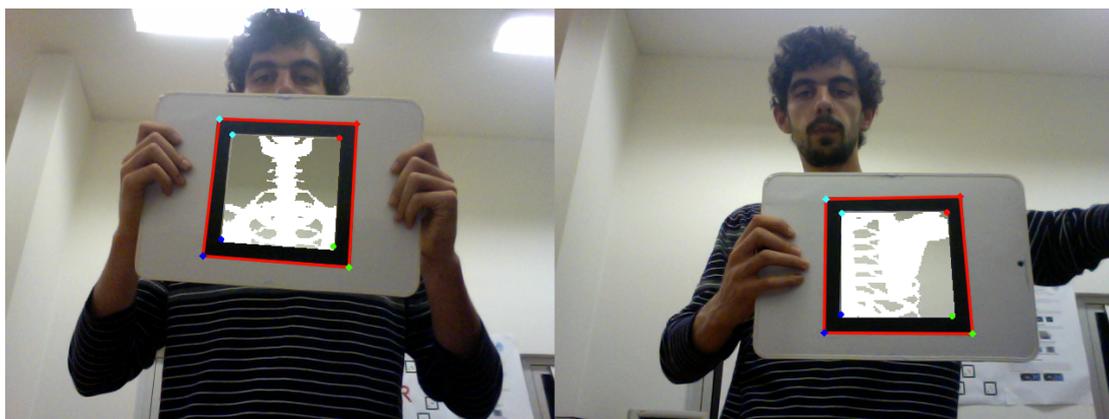


Figura 24 – Realidade Aumentada para visualização médica.

A segunda abordagem demonstrada está ligada à Psicologia através da exposição do sujeito à sua fobia. A Figura 25 ilustra o cenário de um paciente com fobia a formigas (mirmecofobia) onde a exposição ao número de formigas pode ser variável conforme o grau/intensidade da fobia. O conceito é expansível a vários tipos de fobias simplesmente alterando o modelo 3D da formiga por outro insecto ou animal. Desta forma, esta demonstração pode ser aplicada a diversas situações psicopatológicas, auxiliando o paciente a confrontar-se progressivamente com os seus medos, diminuindo a ansiedade/medo e contribuindo para ganhos terapêuticos.



Figura 25 – Exposição do sujeito a situações de fobia.

Demonstrações de Entretenimento

O entretenimento é uma área em constante expansão. Para tal, foram desenvolvidas demonstrações de possíveis aplicações conceptuais nesta área.

Na Figura 26 é apresentado o primeiro jogo de Realidade Aumentada criado com o OpenAR. O jogo foi desenvolvido para dois jogadores, em que cada um controla a sua catapulta, onde o objectivo é acertar com um projectil no adversário. Ao rodar o marcador correspondente à sua catapulta, o jogador, está a controlar a direcção do disparo. O ângulo e a força do projectil são controlados através de teclas do teclado. O jogo termina quando uma das catapultas for destruída.

Uma das vantagens desta demonstração de jogo passa pela interacção entre dois utilizadores.

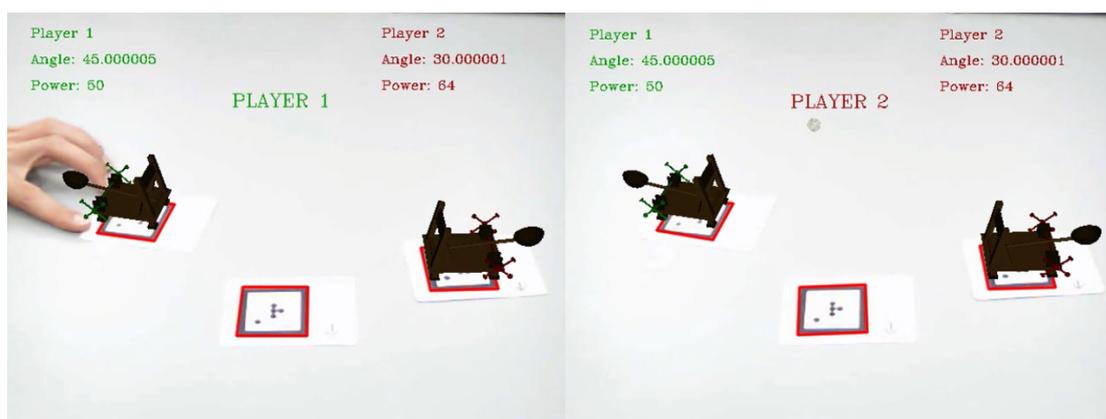


Figura 26 – Jogo das catapultas desenvolvido em Realidade Aumentada.

A próxima aplicação mostra que os marcadores não necessitam de ser planares, apresentando uma forma cúbica composta por um conjunto de seis marcadores nas diferentes faces (Figura 27). Para provar o conceito foi desenvolvido um jogo simples que consiste em substituir o cubo por um dado virtual e que após o lançamento o sistema indica ao utilizador o valor da face voltada para cima.

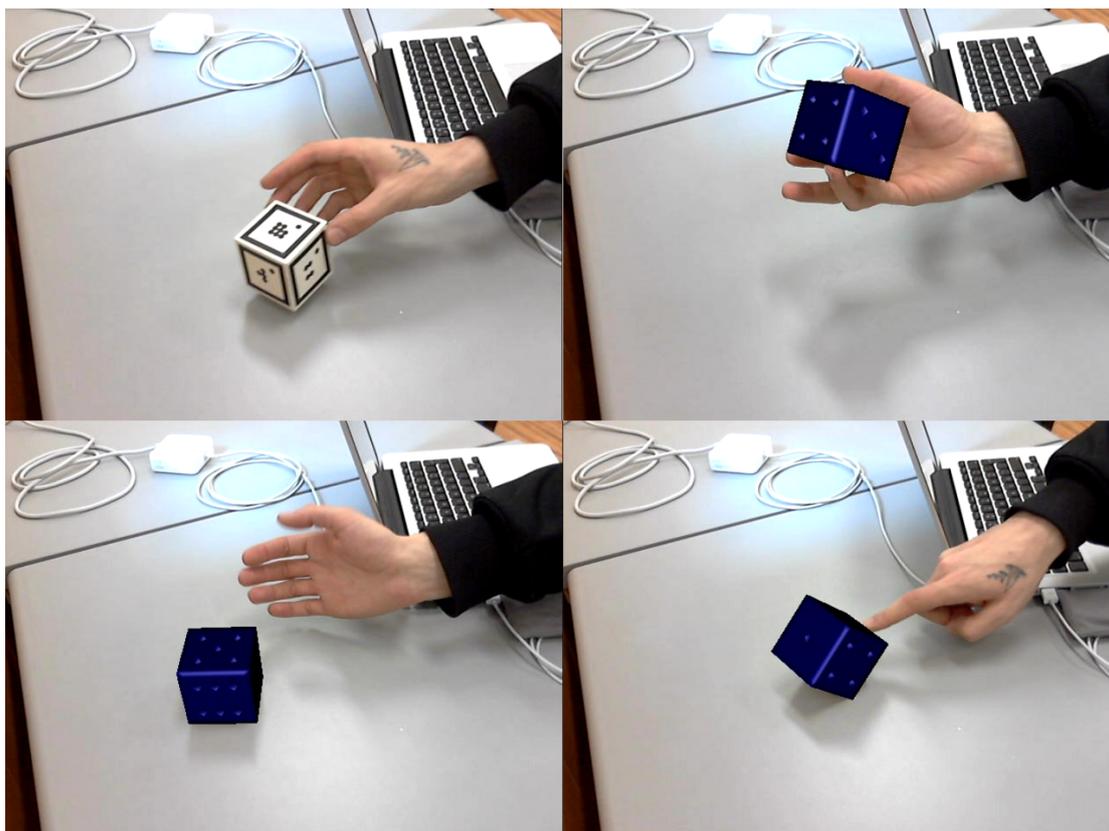


Figura 27 – Demonstração de um marcador cúbico.

Com base na informação fornecida pelo mundo real e com o auxílio de um marcador é possível, numa primeira fase, reduzir (tornar virtualmente invisível) o interior do marcador e mostrar o que se encontra por trás deste (Figura 28, à esquerda). Numa segunda fase pode-se fazer *zoom* dessa área produzindo assim o efeito de lupa em tempo real (Figura 28, à direita).

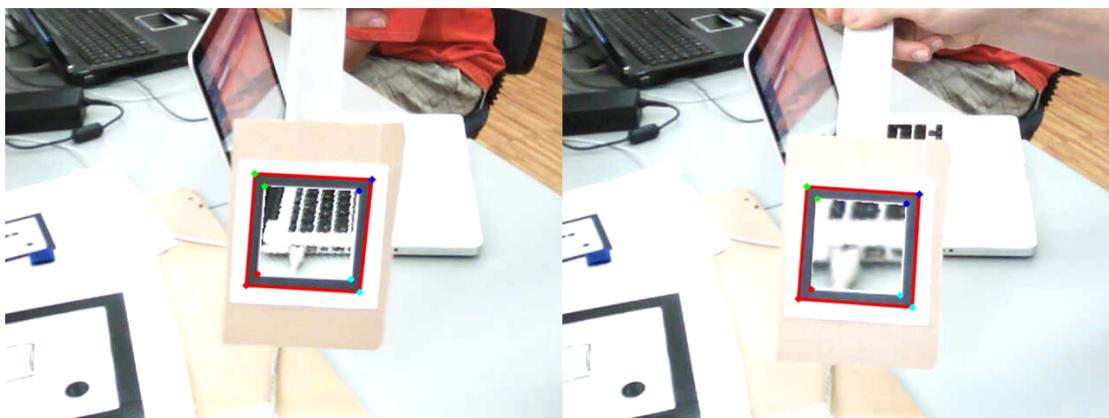


Figura 28 – Utilização dos marcadores para recriar o efeito de lupa.

Demonstradores de utilidade Industrial

Na indústria a Realidade Aumentada pode ser uma mais-valia ao auxiliar o técnico na área da construção ou reparação de equipamentos.

Um exemplo prático é a sobreposição, no mundo real, de linhas guia que representam um sistema de fios eléctricos num quadro eléctrico (Figura 29) ou instruções de montagem/reparação de equipamentos fotográficos (Figura 30).

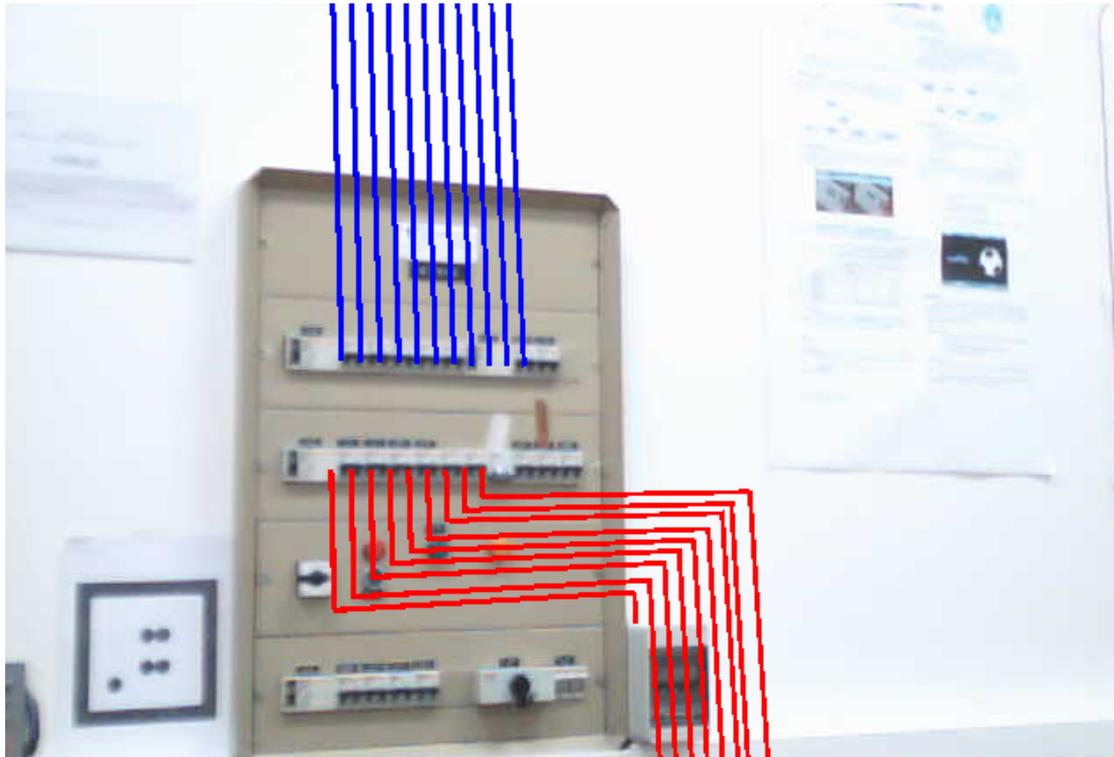


Figura 29 – Realidade Aumentada num quadro eléctrico.

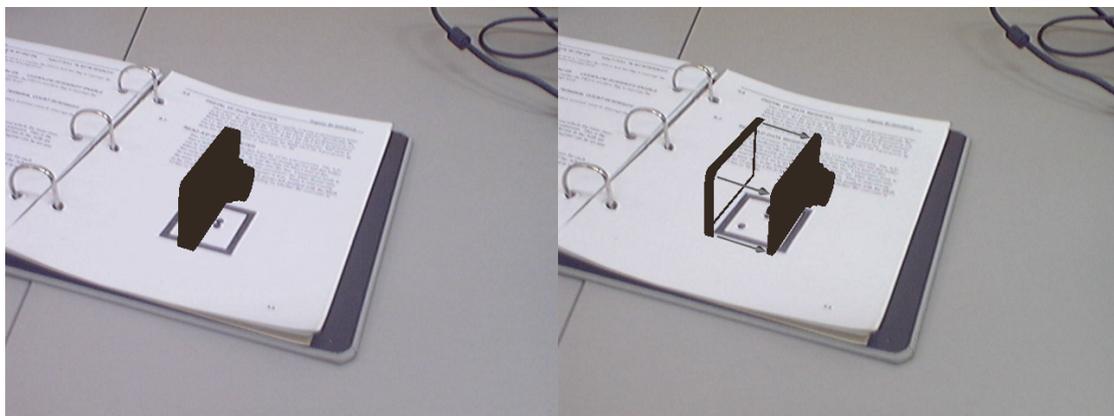


Figura 30 – Tutorial para a reparação de uma câmara digital.

Nesta área o aumento da realidade pode substituir complexos planos de instalações eléctricas ou manuais de instruções, facilitando a visualização face ao problema encontrado.

4. Conclusão

A Realidade Aumentada é um domínio científico relativamente recente e em constante expansão. Tem-se assistido ao desenvolvimento e aperfeiçoamento das suas aplicações em diversas áreas, embora seja um campo sem limites e com um futuro promissor.

O objectivo principal deste trabalho, desenvolver um sistema compreensivo e fácil de usar para produzir aplicações de Realidade Aumentada, foi atingido. Para a maioria dos cenários este sistema satisfaz as necessidades.

Especificamente, os resultados deste trabalho evidenciam a funcionalidade de aplicações e demonstrações de Realidade Aumentada em diferentes áreas, como a interacção homem-máquina, a educação, a medicina/psicologia, o entretenimento e a indústria.

Cada passo deste trabalho foi complexo e teve os seus desafios, no entanto destaca-se, pela positiva, as rotinas de detecção e identificação de marcadores que apresentam resultados muito satisfatórios ao nível do desempenho. Este foi atingido devido à implementação de algoritmos simples e ao reduzido volume de dados a tratar.

No entanto, apresenta alguns problemas, principalmente, face às variações de iluminação do ambiente real. Como é um sistema que se baseia em imagens capturadas por uma câmara para extrair informação de características, a qualidade destas é muito importante. Por vezes, quando uma grande quantidade de luz incide sobre o marcador posicionado no mundo real na imagem capturada, este não é detectado e o sistema falha. A introdução do preditor de movimento melhorou bastante os resultados, no entanto, os períodos de exposição à luz não devem ser muito longos para assegurar a fluidez do movimento. Outra limitação é a relação entre a dimensão dos marcadores e a distância a que estes se encontram da câmara, ou seja, marcadores muito pequenos não são detectados se a distância à câmara for muito grande. Para solucionar este problema o sistema foi preparado para detectar marcadores com várias escalas sem erros na extracção da pose. Existem outras limitações, que não estão directamente relacionadas com o trabalho apresentado nesta

tese mas, que de alguma forma condicionam as aplicações que podem ser criadas e são referidas como propostas para Trabalho Futuro.

Concluindo, importa salientar o contributo deste trabalho para o OpenAR, ao possibilitar o desenvolvimento de aplicações de Realidade Aumentada em tempo real e de uma forma acessível. Os resultados deste trabalho comprovam a importância e utilidade de implementar um sistema de Realidade Aumentada desta natureza.

4.1. Outras Aplicações

As duas aplicações que se seguem são exemplo da aplicação prática do contributo desta tese em dois projectos que foram desenvolvidos no ISR (DEEC/UC). O primeiro projecto chama-se Hilário. O Hilário é um robô guia cuja função é orientar os visitantes pelos corredores do ISR (DEEC/UC) e dar a conhecer os laboratórios.

O Hilário é constituído por, entre outros componentes, uma base mecânica com rodas, um computador e alguns sensores (sonares e câmara de vídeo) (Figura 31).



Figura 31 – Hilário, o robô guia.

Ao sistema de marcadores em conjunto com uma grelha dá-se o nome de poster e é um suporte muito útil na medida em que o robô por vezes precisa de se localizar e de corrigir a sua trajectória. Os marcadores informam o local onde este se encontra e que naquela zona existe uma grelha. Através do cálculo da posição do robô relativamente à grelha é recalculada a trajectória que este vai fazer ao longo do percurso da visita.

O interface gráfico do Hilário está dividido em quatro secções (Figura 32), a primeira relativa a informação de navegação e idioma, a segunda mostra a informação dos sonares, a terceira apresenta a imagem que está a ser capturada pela câmara e, por fim, a quarta mostra o mapa do ISR (DEEC/UC), bem como a localização e trajectória do robô.

A Figura 32 demonstra o início da navegação. Inicialmente, o sistema precisa de se localizar no espaço para poder calcular a trajectória desse local até ao ponto de destino, para tal, o robô é colocado em frente a um dos posters, dispostos pelos corredores do ISR (DEEC/UC), para obter informação sobre a posição onde se encontra.

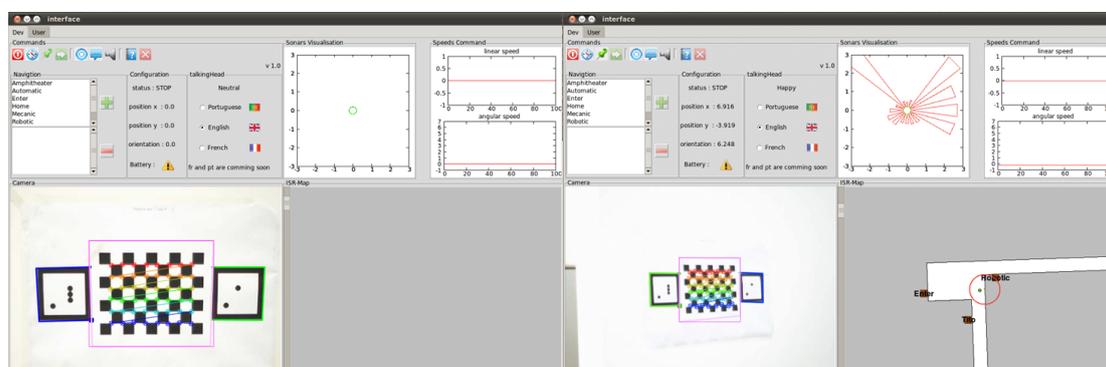


Figura 32 – Aplicação do sistema desenvolvido no projecto Hilário.

O segundo projecto onde o presente trabalho foi aplicado chama-se FxC (Figura 33). Este é um gerador e editor de cenários 3D desenvolvido no ISR (DEEC/UC) com o intuito de facilitar a criação e modelação de cenas que, posteriormente, podem ser usados em qualquer módulo/aplicação criada no OpenAR. A combinação deste trabalho com o descrito nesta tese produziu bons resultados, na medida em que tornou possível modelar um objecto ou um cenário em qualquer ponto de vista, com o simples rodar da folha de marcadores. Este processo consiste em atribuir ao objecto a posição e orientação da folha de marcadores e com essa informação definir o centro de coordenadas no mundo.

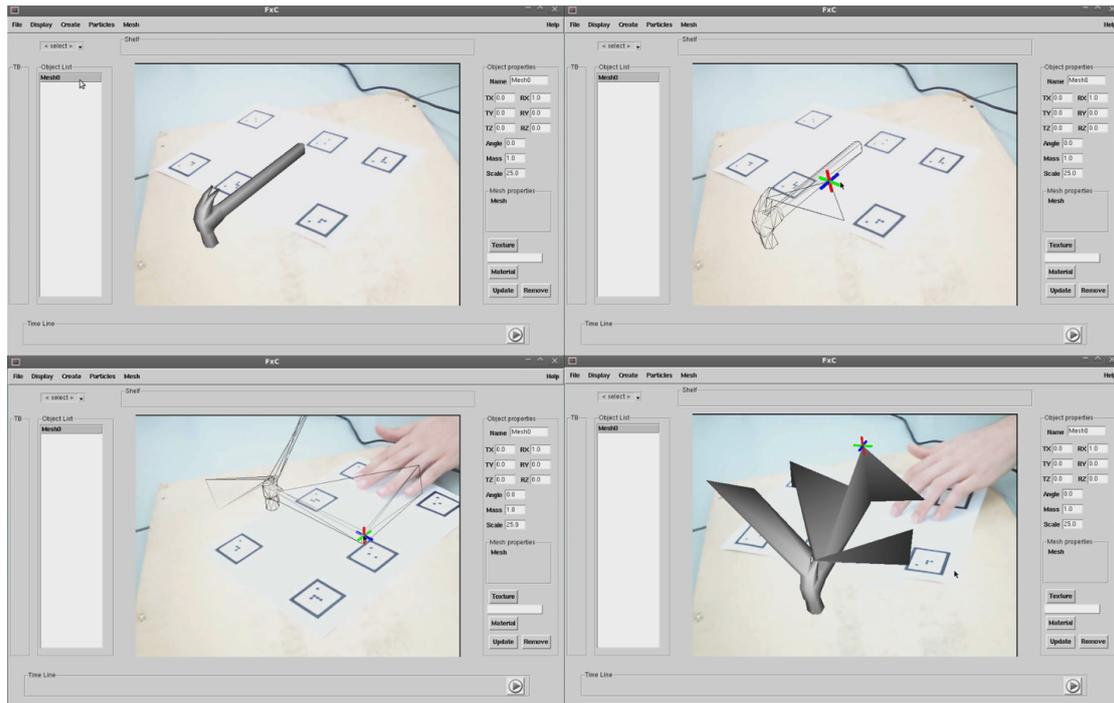


Figura 33 – Aplicação do sistema desenvolvido no modelador FxC.

4.2. Trabalho Futuro

Nesta secção são apresentadas algumas propostas para melhorar a visualização e o realismo dos ambientes virtuais sobrepostos às imagens reais, na medida em que é importante a imersão do utilizador nas aplicações de Realidade Aumentada.

A primeira proposta e mais importante, do ponto de vista do realismo, relaciona-se com o problema da oclusão. A oclusão é uma característica que influencia muito na coexistência entre objectos virtuais e reais, ou seja, a oclusão de objectos virtuais por objectos reais proporciona muito mais realismo, pois atribui características do ambiente real a objectos virtuais. Na Figura 34, a imagem da esquerda apresenta a situação actual em que se encontra o sistema e a da direita uma ilustração de como poderia ser melhorado.

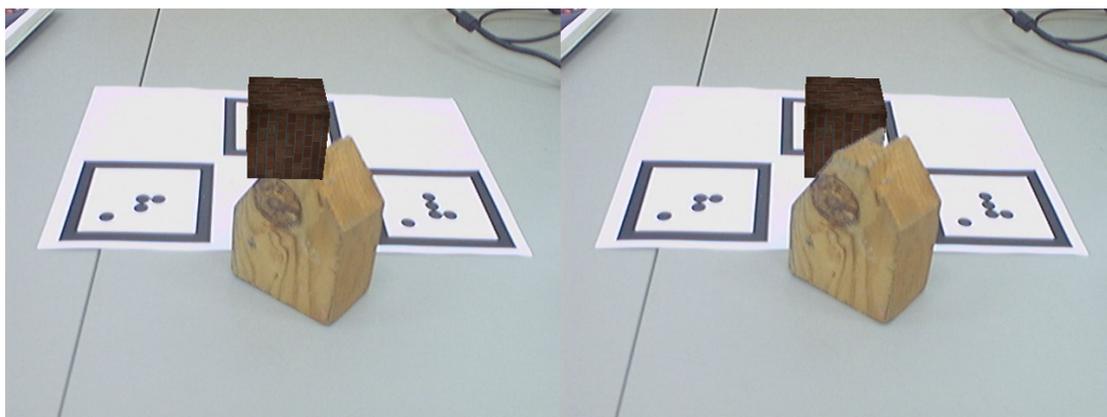


Figura 34 – Problema da oclusão.

A segunda proposta, também relacionada com o realismo, remete para o problema das sombras dos objectos virtuais nos ambientes em que estão inseridos. Na maioria das aplicações é importante que os objectos virtuais se comportem de maneira realista em relação à projecção de sombras. As sombras fornecem, indirectamente, informação sobre o posicionamento 3D dos objectos ajudando à sua coexistência com o ambiente real. Objectos sem ou com um comportamento inadequado em relação às sombras parecem irreais, logo, dificulta a imersão do utilizador no ambiente. Como podemos observar na Figura 35, na imagem da esquerda o objecto não apresenta sombra o que torna difícil localizá-lo visualmente no espaço (situação actual do sistema), em contrapartida, na imagem da direita com a informação que a sombra induz podemos afirmar que o objecto está a flutuar.

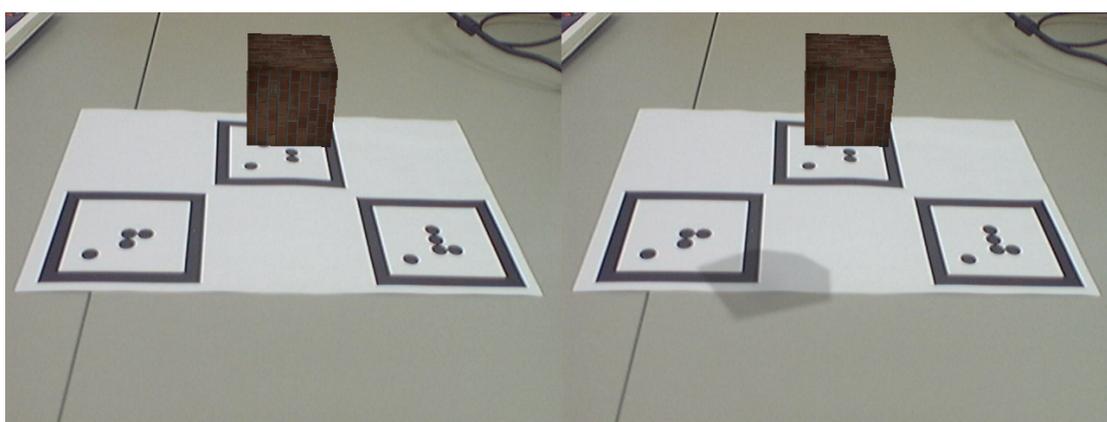


Figura 35 – Problema das sombra.

Como terceira e última proposta sugere-se integrar um sistema de animação 3D com base em *timeline* e *keyframes*, ou seja, uma linha temporal onde são apresentados os

frames que contêm informação das posições de todos os objectos em cena. A animação dos objectos é uma mais-valia porque expande o leque de aplicações que podem ser desenvolvidas com o sistema implementado por este trabalho.

Em resumo, estas são algumas sugestões a serem desenvolvidas no futuro, de modo a melhorar a qualidade e o realismo dos ambientes virtuais, permitindo um maior envolvimento do utilizador em aplicações de Realidade Aumentada.

Referências

- [1] Foley, J. D. (1996). *Computer graphics: principles and practice*. Addison-Wesley Publishing Company, Inc.
- [2] Forsyth, D. & Ponce, P. (2003). *Computer vision: A modern approach*. Prentice Hall.
- [3] Kirner, C, & Tori, R. (2004). Realidade virtual: Conceitos e tendências. *Pré-Simpósio, VII Symposium on Virtual Reality*, São Paulo.
- [4] Insley, S. (2003). *Augmented reality: Merging the virtual and the real*. ECE 399H, Oregon State University.
- [5] Milgram, P., Takemura, H., Utsumi, A., & Kishino, F. (1994). Augmented reality: A class of displays on the reality-virtuality continuum. *Telem manipulator and Telepresence Technologies*, 2351, 282-292.
- [6] Caudell, T. P. & Mizell, D. W. (1992). Augmented reality: an application of heads-up display technology to manual manufacturing processes. *System Sciences, Proceedings of the Twenty-Fifth Hawaii International Conference on*, 2(7-10), 659-669. doi: 10.1109/HICSS.1992.183317
- [7] Azuma, R. T. (1997). A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4), 355-385.
- [8] Behringer, R., Klinker, G. J., & Mizell, D. W. (1996). *Augmented reality: Placing artificial objects in real scenes: Proceedings of IWAR '98*. Natick, Massachusetts: A K Peters.
- [9] Kaufmann, H. (nd). *Collaborative augmented reality in education*. Institute of Software Technology and Interactive Systems, Austria.
- [10] Rizzo, A., Pair, J., Graap, K., Manson, B., McNerney, P. J., Wiederhold, B., Wiederhold, M, & Spira, J. (2004). A virtual reality exposure therapy: Application for Iraq war military personnel with post traumatic stress disorder: From training to toy to treatment. In M. Roy (Ed.), *NATO Advanced Research Workshop on Novel Approaches to the Diagnosis and Treatment of Posttraumatic Stress Disorder*. IOS

Press, Washington, D.C, 235-250.

[11] Azuma, R., Bailiot, Y., Behringer, R., Feiner, S., Julier, S., & MacIntyre, B. (2001). Recent advances in augmented reality. *Computer Graphics and Applications, IEEE, 21(6)*, 34-47. doi: 10.1109/38.963459

[12] Nixon, M. & Aguado, A. (2002). *Feature Extraction & Image Processing*. Cornwall, England: Newnes.

[13] Otsu, N. (1979). A threshold selection method from gray-level histogram. *IEEE Transactions on Systems Man, and Cybernetics, 9*, 62-66.

[14] Sezgin, M. & Sankur, B. (2004). Survey over Image Thresholding Techniques and Quantitative Performance Evaluation. *Journal of Electronic Imaging, 13*, 146-165.

[15] Monteiro, L. H. (nd). *Binarização por Otsu e outras técnicas usadas na detecção de placas*. Manuscrito não publicado.

[16] Lisin, D. A., Mattar, M. A., Blaschko, M. B., Learned-Miller, E. G., & Benfield, M. C. (2005). Combining Local and Global Image Features for Object Class Recognition. *Computer Vision and Pattern Recognition - Workshops, CVPR Workshops*. doi: 10.1109/CVPR.2005.433

[17] Suzuki, S. & Abe, K. (1985). Topological structural analysis of digitized binary images by border following. *Computer Vision and Image Processing, 30*, 32-46.

[18] Bradski, G. & Kaehler, A. (2008). *Learning OpenCV: Computer vision with OpenCV library*. Sebastopol: O'Reilly Media, Inc.

[19] Hartley, R. & Zisserman, A. (2003). *Multiple view geometry in computer vision* (Second Edition). New York: Cambridge University Press.

[20] Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11)*, 1330-1334.

[21] Kirner, C. & Siscoutto, R. (2007). *Realidade virtual e aumentada: Conceitos, projecto e aplicações*. Livro do Pré-Simpósio, IX Symposium on Virtual and Augmented Reality. Petrópolis, Rio de Janeiro.

[22] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transaction of the ASME—Journal of Basic Engineering*, 35-45.

[23] Evensen, G. (2002). The ensemble kalman filtre: Theoretical formulation and practical implementation. *Ocean Dynamics*, 343-367. doi: 10.1007/s10236-003-0036-

9

Apêndice A

Nesta secção é demonstrado como criar uma aplicação simples de Realidade Aumentada usando o OpenAR e a biblioteca desenvolvida neste trabalho.

Primeiro é necessário criar um novo módulo no OpenAR, para tal deve-se executar o seguinte comando na linha de comandos:

```
openarmodule -n modname
```

Este vai criar uma directoria com o nome *modname* que contém um *script* de configuração e uma aplicação teste que serve de ponto de partida para criar o programa. Esta aplicação base ao ser executada está constantemente a receber a imagem que a câmara está a capturar e a mostra-a no ecrã (em ciclo).

Para criar uma aplicação simples de Realidade Aumentada basta procurar por um marcador na imagem capturada pela câmara, calcular e aplicar a sua pose a um objecto virtual e mostrar a imagem final. A imagem final é composta pela imagem capturada e o objecto virtual.

O código que se segue é um exemplo do uso da biblioteca desenvolvida neste trabalho, cujo resultado é uma aplicação simples de Realidade Aumentada, que consiste em detectar um marcador na imagem e atribuir a sua pose a um cubo. Este vai acompanhar o movimento do marcador parecendo coexistir no mesmo espaço.

```
using namespace std;

#include <iostream>
#include <fstream>
#include <ostream>

using namespace std;

#include <stdio.h>
#include <cstdlib>
#include <math.h>
#include <getopt.h>
#include "system.h"
#include "debuginfo.h"
#include "cv.h"
#include "highgui.h"
#include "oaengine.h"
#include "oaobject.h"
#include "ar.h"
#include "bploc.h" <----- (1)
```

```

using namespace OpenAR;

int main(int argc, char **argv){

    float MkScale=1; <----- (2)
    int MkQuant=1; <----- (3)
    int MkId[MkQuant];
    MkId[0]=45; <----- (4)

    BPLocator locator(MkScale); <----- (5)

    bool capture_from_cam=false;
    unsigned long start, now;
    int viewangle=0;
    int numM=0;
    int id;
    int index;
    Pose pose;

    // This part corresponds to the typical initialization
    //of an application
    System::Init(640,480,false);
    oaEngine engine;

    engine.SetViewingAngle(30);
    SDL_WM_SetCaption("My OpenAR Module", "My OpenAR Module");
    start = System::GetTicks();
    SDL_EnableKeyRepeat(50, 50);

    ProjectiveCamera * cammodel = new ProjectiveCamera;

    //
    // a small trick for getting images from a camera or from an AVI file
    //
    CvCapture * capture;
    if( argc == 1||(argc == 2&&strlen(argv[1])==1&&isdigit(argv[1][0])))
        capture = cvCaptureFromCAM( argc == 2 ? argv[1][0] - '0' : 0 );
    else if( argc == 2 ){
        capture = cvCaptureFromAVI( argv[1] );
        capture_from_cam=false;
    }

    if (!capture) {
        printf("Cannot open camera stream\n");
        exit(1);
    }

    if (!cammodel->LoadIntrinsicFile(camparam.cal)){
        printf("Error loading camera parameters\n");
        exit(1);
    }else
        locator.setCameraModel(cammodel); <----- (6)

    engine.setIntrinsics(cammodel,640,480,1,50000);

    oaObject * cube; <----- (7)
    cube=new oaBox(10,10,10);
    cube->setname(cube);
    cube->setMass(0);

```

```

// the main cycle
// capture -> process -> display & process events
for(;;) {
    IplImage * iframe;
    IplImage * frame;
    now = System::GetTicks();

    // capture one image
    iframe = cvQueryFrame(capture);
    frame = cvCloneImage(iframe);          <----- (8)

    numM=locator.DetectMarker(frame);     <----- (9)

    for(int lk=0;lk<numM;lk++){           <----- (10)
        id=locator.getMarkerId(lk);       <----- (11)
        if(id==MkId[0]){                  <----- (12)

            index=locator.getMarkerIndex(id); <----- (13)
            pose=locator.getMarkerPose(index); <----- (14)
            if(firstTimeOnScreen==true)
                engine.InsertObject(cube); <----- (15)

            cube.setpose(pose);            <----- (16)
        }
    }

    engine.updatePhysics();
    locator.DrawMarkerFeatures(frame);     <----- (17)
    engine.setBackground(frame);          <----- (18)
    engine.drawScene();                   <----- (19)
    System::GL::SwapBuffers();

    // now process SDL events
    SDL_Event ev;
    while(SDL_PollEvent(&ev)) {
        switch(ev.type) {
            case SDL_QUIT:
                return 0;
                break;
            case SDL_MOUSEBUTTONDOWN:
                switch (ev.button.button){
                    case SDL_BUTTON_LEFT:
                        break;
                    case SDL_BUTTON_RIGHT:
                        break;
                }
                break;
            case SDL_KEYDOWN:
                switch (ev.key.keysym.sym){
                    case SDLK_ESCAPE:
                    case SDLK_q:
                        return 0;
                    case SDLK_f:
                        System::GL::ToggleFullScreen();
                        break;
                    default:
                        std::cout << "key"<< std::endl;
                        break;
                }
                break;
        }
    }
    cvReleaseImage(&frame);
}
return 0;
}

```

Legenda:

- (1) incluir a biblioteca desenvolvida;
- (2) definir a escala do marcador;
- (3) definir a quantidade de marcadores;
- (4) definir o número do marcador a utilizar;
- (5) iniciar o localizador desenvolvido;
- (6) carregar os parâmetro da câmara;
- (7) criar um cubo;
- (8) guardar a imagem capturada pela câmara;
- (9) enviar a imagem para o detector de marcadores e este retorna a quantidade de marcadores encontrados;
- (10) criar um ciclo que percorre todos os marcadores encontrados;
- (11) guardar o número de identificação de cada marcador;
- (12) procurar o número do marcador que foi definido;
- (13) caso seja encontrado o marcador definido, guardar o seu índice da lista de marcadores;
- (14) enviar o índice do marcador para o localizador desenvolvido e este retorna a sua pose;
- (15) inserir o cubo na cena;
- (16) atribuir a pose do marcador ao cubo;
- (17) desenha na imagem características dos marcadores encontrados;
- (18) coloca a imagem capturada no fundo da aplicação;
- (19) desenha o cubo sobre a imagem capturada, produzindo a imagem final.