

Faculdade de Engenharia da Universidade do Porto



FEUP

Monitorização da trajetória de uma bola num jogo de Ténis de Mesa

Gustavo Guedes de Sousa

VERSÃO FINAL

Dissertação realizada no âmbito do
Mestrado Integrado em Engenharia Eletrotécnica e de Computadores
Major Automação

Orientador: Pedro Luís Cerqueira Gomes da Costa (Professor Doutor)
Co-orientador: Paulo José Cerqueira Gomes da Costa (Professor Doutor)

24/07/2012

A Dissertação intitulada

“Monitorização da Trajectória de uma Bola num Jogo de Ténis de Mesa”

foi aprovada em provas realizadas em 17-07-2012

o júri


Presidente Professor Doutor Armando Jorge Miranda de Sousa
Professor Auxiliar do Departamento de Engenharia Electrotécnica e de
Computadores da Faculdade de Engenharia da Universidade do Porto


Professor Doutor José Luís Sousa de Magalhães Lima
Professor Adjunto do Departamento de Electrotecnia da Escola Superior de
Tecnologia e Gestão de Bragança


Professor Doutor Pedro Luís Cerqueira Gomes da Costa
Professor Auxiliar do Departamento de Engenharia Electrotécnica e de
Computadores da Faculdade de Engenharia da Universidade do Porto


Professor Doutor Paulo José Cerqueira Gomes da Costa
Professor Auxiliar do DEEC - FEUP

O autor declara que a presente dissertação (ou relatório de projeto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extratos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são corretamente citados.


Autor - Gustavo Guedes de Sousa

Resumo

O Ténis de Mesa é um desporto que se destaca pelo ritmo a que é jogado, podendo a bola atingir velocidades na ordem dos 120 km/h.

Daí nasce a dificuldade de se analisar um jogo ao pormenor, pois não é possível observar-se, através de um vídeo, certas características, como a trajetória exata da bola (posição em que esta se encontra e efeito aplicado à mesma) ou os movimentos dos jogadores.

É neste sentido que a engenharia pode ser usada, pois tem ao seu dispor metodologias e tecnologias que podem providenciar, tanto aos treinadores como aos jogadores, imagens e dados claros que poderão levar à execução de relatórios de jogo mais completos e detalhados.

Neste trabalho, é apresentada uma estratégia que visa solucionar, de forma eficaz e robusta, parte dos problemas expostos.

Abstract

One of the notable characteristics of Table Tennis is the speed at which it is played, as the ball can reach speeds as high as 120 km/h.

From a simple video recording it's impossible to accurately profile a game, as the ball's position at a certain point in time is extremely difficult to determine, as well as the players' movements.

This is where engineering can become of use, as it has to it's disposal technologies and methodologies that can give to coaches and players, image and data that are clearer and could lead to game reports that are more complete and detailed.

This work aims to provide solution to the problems presented, in a well documented, efficient and robust way.

Agradecimentos

Quero deixar um agradecimento a todos os que me acompanharam ao longo do meu percurso acadêmico.

Aos meus orientadores Professor Doutor Pedro Gomes da Costa e Professor Doutor Paulo Gomes da Costa, por toda a disponibilidade demonstrada para ajudar na realização deste trabalho, por forma a cumprir os objetivos propostos.

Ao Luís Pereira, pela ajuda, tempo e partilha de conhecimentos, ao longo da execução deste trabalho final.

Ao João Vinhas, pela disponibilidade demonstrada para ajudar na realização da página web.

Ao João Pedro Gomes, pela disponibilidade e ajuda preciosa ao rever todo este documento, para que assim fosse mais fácil de ler e compreender todos os conceitos nele retratados.

A todos os meus amigos que comigo compartilharam os mesmos grupos de trabalho para execução dos vários projetos realizados ao longo do curso, ajudando a cumprir os objetivos a eles relacionados.

Pretendo também agradecer à mulher fantástica que a vida me proporcionou. À Ana, por toda a motivação, alegria e amizade manifestadas que fizeram com que fosse muito mais fácil ultrapassar certos obstáculos. Agradeço também a compreensão por ela demonstrada que fez com que acabasse a dissertação em tempo útil.

À minha família, nomeadamente, Felícia Guedes, Norberto Pinto, Zélia Guedes, Paulo Cerqueira e Bruno Pinto por todas as palavras de força que me deram confiança para atingir todos os meus objetivos.

Por fim, quero deixar um agradecimento muito especial aos meus pais. Obrigado pelo esforço realizado para me proporcionar um curso superior. Obrigado pelo vosso carinho, palavras de força e por todos os valores que me transmitiram.

Índice

Resumo	iii
Abstract.....	v
Agradecimentos	vii
Índice.....	ix
Lista de figuras	xi
Lista de tabelas	xv
Abreviaturas e Símbolos	xvii
Capítulo 1	1
Introdução.....	1
1.1 - Objetivos	2
1.2 - Planeamento.....	2
1.3 - Estrutura do Documento	3
Capítulo 2	5
Estado da Arte	5
2.1 - Futebol.....	5
2.2 - Basquetebol.....	7
2.3 - Ténis.....	11
2.3.1 - Suavização da imagem	11
2.3.2 - Filtro de Sobel.....	12
2.3.3 - Subtração do Fundo	13
2.3.4 - Detecção da bola na imagem	13
2.4 - Ténis de Mesa	16
2.4.1 - Visão Estereoscópica	16
2.4.2 - Sincronização de duas câmaras.....	17
Capítulo 3	19
Plataforma de teste utilizada	19
3.1 - Estudo e escolha das ferramentas a utilizar no projeto	19
3.1.1 - <i>Software</i>	19
3.1.2 - <i>Hardware</i>	21
3.2 - Interface Gráfica (Lazarus).....	23
Capítulo 4	25

Aquisição e Processamento de imagem.....	25
4.1 - Aquisição de Imagem	25
4.2 - Processamento de Imagem	26
4.2.1 - Primeira Abordagem	26
4.2.1.1 - Filtro de Mediana	26
4.2.1.2 - Filtro de Sobel.....	27
4.2.1.3 - Subtração do Fundo	28
4.2.2 - Segunda Abordagem.....	31
4.2.2.1 - Aplicação de uma máscara da bola na imagem	31
4.2.2.2 - Junção das 3 fases de processamento	34
4.3 - Detecção e Determinação da Posição da bola na imagem	38
4.4 - Cálculo da velocidade da bola entre 2 Frames (Imagens)	41
4.5 - Processamento de Imagem em Tempo Real	43
Capítulo 5	45
Sistema de visão estereoscópica	45
5.1 - Calibração das câmaras	46
5.2 - Reconstrução 3D.....	52
Capítulo 6	55
Testes e Resultados Obtidos	55
6.1 Resultados obtidos	57
Capítulo 7	59
Conclusões	59
7.1 - Trabalho futuro	60
Referências	63

Lista de figuras

Figura 1.1 - Diagrama de Gantt para a realização da Dissertação. Erro! Marcador não definido.	
Figura 2.1 - Procura da bola nas direções radiais a partir do jogador [18].	6
Figura 2.2 - Monitorização do movimento de bola chutada por um jogador [18].	6
Figura 2.3 - Configuração das câmaras utilizada em [19].	7
Figura 2.4 - Demonstração do método aplicado para reconhecimento da bola através da cor [19]; a) Imagem original; b) Método aplicado à imagem a).	8
Figura 2.5 - Aplicação do algoritmo de <i>Canny</i> a uma imagem [19].	8
Figura 2.6 - Aplicação do algoritmo de <i>Canny</i> a uma imagem de um jogador com a bola na mão [19].	9
Figura 2.7 - Aplicação da subtração do fundo juntamente com a transformação da imagem numa imagem binária [19]; a) Imagem original; b) Imagem binária; c) Subtração do Fundo.	9
Figura 2.8 - Algoritmo da <i>Canny</i> aplicado à imagem resultante da aplicação da subtração do fundo [19].	10
Figura 2.9 - Exemplo da aplicação do algoritmo <i>Trackin-based approach</i> [19]; a) Janela 3D de dimensão 60x60x60 cm; b) Janela 3D de dimensão 120x120x120 cm.	10
Figura 2.10 - Demonstração da aplicação de um filtro de mediana (cf. [3]).	11
Figura 2.11 - Exemplo da aplicação do filtro de Sobel numa imagem.	12
Figura 2.12 - Exemplo da aplicação da subtração do fundo [3]; a) Fundo; b) Imagem Real; c) Subtração de b) por a).	13
Figura 2.13 - <i>Frame</i> capturada pela câmara fixa situada por detrás de um jogador [20].	14
Figura 2.14 - Exemplo da deteção do fundo através de um filtro de mediana [20]; a) <i>Frame</i> original; b) Fundo.	14
Figura 2.15 - Deteção das linhas com campo [20]; a) Canal V do espaço HSV; b) Imagem resultando da aplicação do <i>Top-Hat</i> ; c) Aplicação do limiar; d) Reconstrução das linhas.	15

Figura 2.16 - Configuração de um sistema de visão estereoscópica para o Tênis de Mesa (cf. [2]).	16
Figura 2.17 - Fluxograma do algoritmo de sincronização das duas câmaras (cf. [2]).	18
Figura 3.1 - Parâmetros de configuração presentes no componente TSdpoVideo4L2.	20
Figura 3.2 - Parâmetros de configuração presentes no componente TSdpoFastForm.	20
Figura 3.3 - Câmara PointGrey FL3-U3-13S2M-CS USB3.0 [6].	21
Figura 3.4 - Imagem capturada pela câmara PointGrey FL3-U3-13S2M-CS USB3.0, com a bola demonstrada no retângulo vermelho.	22
Figura 3.5 - Câmaras Eye utilizadas na PS3.	22
Figura 3.6 - Interface gráfica realizada em Lazarus para o projeto.	23
Figura 3.7 - Exibição dos Parâmetros do vídeo capturado pela câmara.	24
Figura 4.1 - Exemplo de imagem a cores.	26
Figura 4.2 - Exemplo de imagem capturada a preto e branco com a codificação YUV420. ...	26
Figura 4.3 - Filtro de Mediana aplicado à imagem da Figura 4.3.	27
Figura 4.4 - Imagem capturada sem processamento.	27
Figura 4.5 - Filtro de Sobel aplicado à imagem da Figura 4.5.	27
Figura 4.6 - Imagem capturada com Filtro de Mediana aplicado.	27
Figura 4.7 - Exemplo do processamento da 1ª Abordagem com a bola numa superfície de tons misturados; a) Fundo; b) Imagem capturada; c) Imagem b) Processada; d) Subtração de c) por a).	29
Figura 4.8 - Exemplo do processamento da 1ª Abordagem com a bola numa superfície de cor preta; a) Fundo; b) Imagem capturada; c) Imagem b) Processada; d) Subtração de c) por a).	30
Figura 4.9 - Exemplo de operação pontual entre imagens.	32
Figura 4.10 - Demonstração do resultado da utilização de uma máscara 13x13; a) Imagem capturada; b) Máscara Aplicada à imagem a).	32
Figura 4.11 - Demonstração do resultado da utilização de uma máscara 7x7; a) Imagem capturada; b) Máscara Aplicada à imagem a).	34
Figura 4.12 - Exemplo do processamento da 2ª Abordagem; a) Fundo; b) Imagem capturada; c) Subtração de b) por a) ; Imagem final após a aplicação da máscara.	35
Figura 4.13 - Exemplo do processamento da 2ª Abordagem sem a utilização do filtro de Mediana; a) Fundo; b) Imagem capturada; c) Subtração de b) por a) ; Imagem final após a aplicação da máscara.	36
Figura 4.14 - Exemplo do processamento da 2ª Abordagem otimizado; a) Fundo; b) Imagem capturada; c) Subtração de b) por a) ; Imagem final após a aplicação da máscara.	37

Figura 4.15 - Resultado da aplicação do método para detetar e determinar o centro da bola; a) Fundo; b) Imagem Real; c) Imagem com o método aplicado e um quadrado desenhado à volta da posição calculada para centro da bola.	40
Figura 4.16 - Interface gráfica que permite a visualização das coordenadas calculadas para o centro da bola.	41
Figura 4.17 - Duas Frames sobrepostas com esquema para o cálculo do Δpos	42
Figura 4.18 - Demonstração do resultado da velocidade, bem como o tempo decorrido entre as duas frames da figura anterior e as coordenadas das duas posições da bola.	42
Figura 4.19 - Histórico da posição, velocidade e tempo decorrido entre duas <i>frames</i>	43
Figura 5.1 - Modelo <i>pinhole</i> câmara (cf. [7]).	45
Figura 5.2 - Matriz de calibração utilizada em [7].	47
Figura 5.3 - Matriz de pontos para calibração das câmaras.	48
Figura 5.4 - Suporte das 2 câmaras que compõem o sistema de visão estereoscópica.	49
Figura 5.5 - Configuração do sistema de visão estereoscópica utilizado com as respetivas medidas.	49
Figura 5.6 - Imagens capturadas com a matriz de pontos em várias posições e com as bolas nas posições conhecidas da mesa.	50
Figura 5.7 - Referencial utilizado para proceder à identificação das coordenadas dos pontos no cenário real.	51
Figura 5.8 - Modelo de um sistema de visão estereoscópica [16].	52
Figura 5.9 - Triangulação ideal dos raios das câmaras: M' é a interseção dos raios das duas câmaras (cf. [17]).	53
Figura 5.10 - Triangulação sem interceção das retas (cf. [17]).	53
Figura 6.1 - Imagens capturadas em cenários e distâncias diferentes.	56
Figura 6.2 - Detecção da bola nos vários cenários.	57
Figura 6.3 - <i>Layout</i> do vídeo realizado que ilustra o resultado final do sistema desenvolvido.	58

Lista de tabelas

Tabela 4.1 - Máscara 13x13.	31
Tabela 4.2 - Máscara 7x7.	33
Tabela 4.3 - Matriz de uma imagem exemplo com uma bola.	38

Abreviaturas e Símbolos

Lista de abreviaturas

FEUP	Faculdade de Engenharia da Universidade do Porto
FPGA	<i>Field Programmable Gate Array</i>
fps	<i>Frames per second</i>
USB	<i>Universal Data Bus</i>

Lista de símbolos

<i>cm</i>	Centímetros
<i>m</i>	Metros
<i>ms</i>	Milissegundos
<i>píxeis/s</i>	Pixéis por segundo
Π	Plano de imagem
<i>km/h</i>	Quilómetros por hora
<i>s</i>	Segundos
Δ	Varição de uma determinada grandeza
R	Matriz de Rotação
T	Matriz de Translação

Capítulo 1

Introdução

Muitos desportos têm apostado na tecnologia para facilitar aos júris dos jogos a análise das jogadas. Uma das mais robustas tecnologias é o “*Hawk-Eye*” (em português, “Olho de Falcão”), que é uma ferramenta muito utilizada no Ténis. Esta permite emitir repetições de jogadas em 3D, providenciando assim uma análise mais eficaz da jogada, para que não existam discordâncias ao decidir a validade da jogada ou até mesmo quem é o jogador que pontua.

Neste trabalho, o que se pretende não é emissão de repetições 3D em câmara lenta, mas sim a monitorização em tempo real da trajetória de uma bola num jogo de Ténis de Mesa.

Este tema advém da necessidade dos treinadores poderem facultar um treino pormenorizado, utilizando as imagens e dados de posição da bola ao longo do jogo para corrigir os movimentos dos jogadores. Em certos países (China principalmente) existe já muita pesquisa e desenvolvimento de trabalhos relacionados com ténis de mesa, em que os treinadores já aproveitam estas tecnologias.

É com este intuito que nasce a motivação para realizar uma dissertação no âmbito do Curso de Mestrado Integrado de Engenharia Eletrotécnica e Computadores.

Ao longo do documento, serão estudados e utilizados diferentes algoritmos por forma a fazer deste desporto olímpico um desporto que possa usufruir de todas as vantagens que a tecnologia pode proporcionar.

1.1 - Objetivos

Como indica o título da Dissertação, o objetivo principal deste trabalho é a monitorização da trajetória de uma bola num jogo de ténis de mesa. Esta monitorização consiste em elaborar um sistema que seja capaz de:

- Estimar a posição da bola através de várias câmaras;
- Calcular a velocidade da bola durante a trajetória;
- Detetar os pontos de contacto da bola com a mesa;
- Permitir a visualização em 3D, tanto em tempo real, como *a posteriori*, da trajetória da bola.

Por fim, o sistema desenvolvido deverá ser aplicado no centro de estágio de alto rendimento de Ténis de Mesa em Vila Nova de Gaia.

1.2 - Planeamento

Por forma a cumprir os objetivos propostos na data prevista, foram delineadas, em conjunto com o orientador, as seguintes tarefas:

- 1- Estudo do estado da arte e de outras tecnologias;
- 2- Desenvolvimento de um sistema de deteção da bola baseado em múltiplas câmaras;
- 3- Implementação do sistema projetado;
- 4- Recolha de dados e análise do desempenho do sistema;
- 5- Conclusão e escrita da Tese.

A figura seguinte demonstra o planeamento realizado num diagrama de Gantt, que tem presente as 5 tarefas expostas anteriormente, juntamente com a sua data de início e de fim.

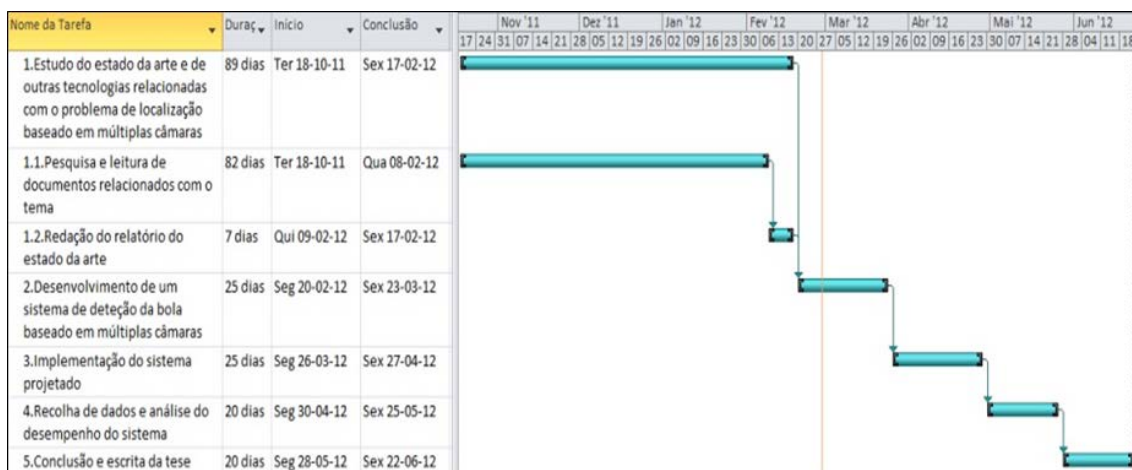


Figura 1.1 - Diagrama de Gantt para a realização da Dissertação.

1.3 - Estrutura do Documento

No Capítulo 2, é relatado o Estado da Arte, que consiste na exposição de vários algoritmos utilizados, até à data, em diversos artigos científicos relacionados com a monitorização da trajetória de uma determinada bola de um desporto. Aqui serão expostos alguns algoritmos, já desenvolvidos, que permitem a aquisição de imagens utilizando um sistema de visão estereoscópica, bem como processamento de imagem que visa a deteção de uma bola num determinado cenário.

O Capítulo 3 relata o estudo realizado para proceder à escolha das ferramentas mais adequadas ao projeto (*Software* e *Hardware*), bem como a interface gráfica desenvolvida para execução dos algoritmos elaborados.

Os algoritmos utilizados para aquisição e processamento das imagens capturadas pelas câmaras, bem como o processo para determinação das coordenadas e velocidade da bola estão expostos no capítulo 4.

O capítulo 5 descreve o sistema de visão estereoscópica desenvolvido, explicando o mesmo através de documentos relacionados com este tipo de sistemas, nomeadamente, ao nível da calibração das câmaras e reconstrução 3D da imagem.

Os testes e resultados obtidos podem ser encontrados no capítulo 6.

Por fim, as conclusões e trabalho futuro estão presentes no capítulo 7.

Capítulo 2

Estado da Arte

A monitorização da trajetória de uma bola nos diferentes desportos tem vindo a ser estudada nos últimos anos. Para tal utilizam-se algoritmos capazes de, através de imagens capturadas por câmaras, determinarem a posição exata da bola num determinado ambiente.

Nomeadamente no Ténis, Futebol, Basquetebol e no Ténis de Mesa, têm vindo a ser publicados alguns artigos que estudam a melhor configuração das câmaras, bem como os melhores algoritmos para processamento de imagem.

Ao longo deste capítulo serão ilustradas várias metodologias que já foram estudadas e implementadas.

2.1 - Futebol

Yoshinori Ohno, Jun Miura e Yoshiaki Shirai expõem em [18] um conjunto de algoritmos que permitem a monitorização do movimento dos jogadores e a determinação da posição 3D da bola num jogo de Futebol a partir de imagens capturadas por câmaras fixas.

Os algoritmos descritos em [18] consistem, numa primeira fase, na determinação da posição dos jogadores no campo, modelizando a cor do uniforme dos jogadores por um paralelepípedo, levando, assim, à extração das regiões das camisolas e dos calções dos jogadores na imagem.

Uma vez determinadas estas regiões, o sistema determina a posição dos jogadores no campo, utilizando para isso o modelo *pinhole camera*, que relaciona as coordenadas de um ponto no espaço com as coordenadas do mesmo na imagem (este modelo será explicado mais à frente neste documento). A calibração das câmaras é alcançada através da posição do jogador no campo, utilizando a posição dos pés e a altura dos jogadores. Assim, conseguem-se relacionar vários pontos entre a posição nas imagens e a posição real.

Estando determinada a posição dos jogadores, a segunda fase descrita em [18] consiste na monitorização da trajetória da bola. Para isso, é descrito um método que determina a

posição da bola, baseando-se no seu movimento, ou seja, efetua a procura da bola nas direções radiais a partir dos jogadores, como demonstra a seguinte figura.

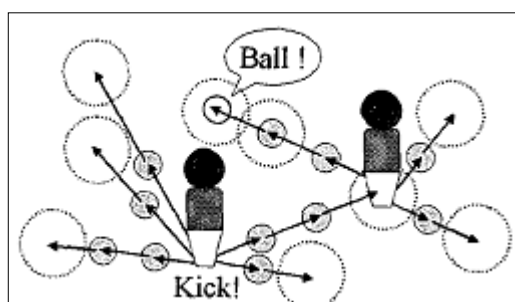


Figura 2.1 - Procura da bola nas direções radiais a partir do jogador [18].

Desta forma, o movimento da bola pode ser monitorizado em situações em que o jogador se encontra a esconder a bola na imagem. A partir da posição dos jogadores que se encontram junto da bola e da posição anterior desta, a posição atual é determinada através da procura da bola nas direções radiais do jogador. A figura seguinte demonstra o resultado do método utilizado em [18], sendo que os círculos representam os candidatos a bola numa determinada *frame* e os quadrados representam os candidatos a bola nas *frames* anteriores.

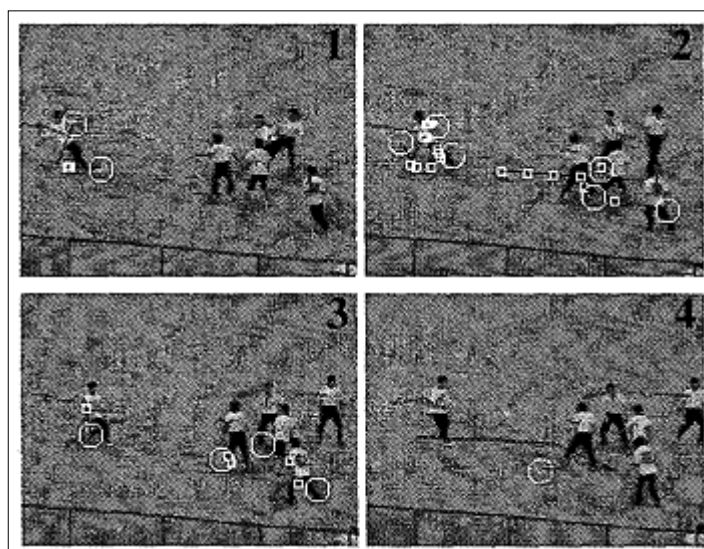


Figura 2.2 - Monitorização do movimento de bola chutada por um jogador [18].

Em [18], é ainda retratado um método que permite a determinação da posição da bola quando esta se encontra sobreposta ao jogador na imagem. Neste cenário, o sistema para de monitorizar a posição da bola, considerando temporariamente que o jogador tem a bola no seu poder. Nesse instante, o sistema continua a pesquisar a bola nos píxeis vizinhos da imagem, sendo que, assim que esta saia da posse de um determinado jogador, o sistema continuará a sua monitorização.

Uma vez detetada a posição da bola, o sistema estima a trajetória 3D da mesma, utilizando para isso equações que relacionam o movimento da bola com a lei da gravidade.

2.2 - Basquetebol

Um dos desportos em que também tem vindo a ser estudada a monitorização da trajetória da bola é o Basquetebol. Em relação ao futebol, este tem a particularidade do campo ser mais pequeno e da bola ter um tamanho maior.

Neus Agelet Ruiz expõe em [19] uma metodologia para deteção da bola utilizando imagens capturadas por câmaras fixas. A monitorização da trajetória da bola é realizada atualizando recursivamente a posição 3D desta, fazendo para isso a correspondência de vários pontos no espaço real e na imagem e as características da mesma. Para além disso, nesse mesmo documento, é realizada uma extrapolação para determinar a posição futura da bola.

Assim sendo, numa primeira fase, são escolhidas as posições das câmaras por forma a capturar apenas metade do campo. A figura seguinte demonstra a configuração utilizada em [19].

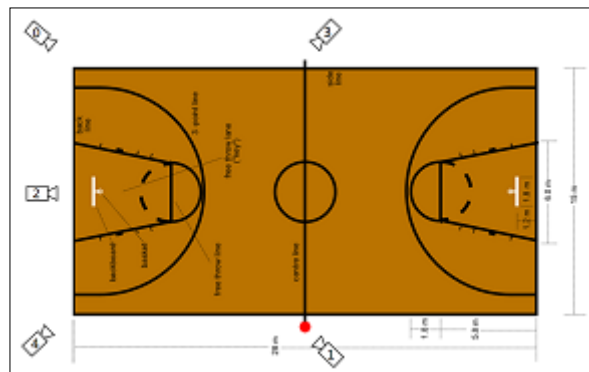


Figura 2.3 - Configuração das câmaras utilizada em [19].

Na figura anterior, estão representadas as câmaras utilizadas com o seu respetivo número, sendo que o ponto vermelho na imagem indica a origem do referencial utilizado, com $z=0$ ao nível do solo.

Neste documento, utilizam-se várias câmaras para proporcionar vários pontos de vista ao sistema, sendo, assim, mais fácil estimar a posição da bola em 3D. A calibração das várias câmaras é atingida utilizando o mesmo algoritmo já descrito acima, que consiste em fazer corresponder a posição 3D de vários pontos no cenário real à posição 2D desses mesmos pontos nas imagens capturadas.

Uma vez escolhida a posição para as câmaras, Neus Agelet Ruiz começa por proceder à deteção dos candidatos a bola em imagens isoladas. Para isso, recorre a algoritmos que utilizam a cor ou a forma da bola.

Pesquisar a bola numa imagem através da sua cor torna-se uma tarefa difícil, pois nem sempre a iluminação presente no jogo proporciona imagens em que a cor da bola é sempre a mesma. Para contornar esse fator, em [19], Neus Agelet Riuz começa por determinar a probabilidade de uma cor na imagem poder ser a bola. Para isso, em cada píxel, é calculada a

probabilidade dessa cor corresponder à bola e, após isso, é estabelecido um limiar que faz com que sejam excluídos os objetos que não correspondam à bola. Este método resulta para imagens em que a bola não se encontra desfocada (demonstrado na figura 2.4), mas, quando aplicado a uma sequência de imagens capturadas, não se revela tão preciso.



Figura 2.4 - Demonstração do método aplicado para reconhecimento da bola através da cor [19]; a) Imagem original; b) Método aplicado à imagem a).

Outra forma que permite a detecção da bola numa imagem é a utilização da forma esférica da mesma. Para isso, estão retratados dois métodos em [19] que poderão ser aplicados a uma imagem com os contornos dos objetos evidenciados:

- a) *Region Circularity* (circularidade da região): este método consiste em utilizar a variância do raio de uma dada região para determinar a sua circularidade. No entanto, este método nem sempre leva a bons resultados, uma vez que o algoritmo de detecção de contornos leva, por vezes, a resultados em que os contornos são contornos abertos, impossibilitando assim a detecção do círculo;
- b) *Circles Detection* (detecção de círculos): este método consiste na utilização da transformada de *Hough* para detetar círculos na imagem, tendo este método 50 % de sucesso, sendo que, em algumas imagens, a detecção da bola leva a resultados completamente errados.

Para detecção dos contornos na imagem, Neus Agelet Riuz utiliza o algoritmo de *Canny*, que permite uma detecção perfeita dos contornos dos objetos, como demonstra a seguinte figura:



Figura 2.5 - Aplicação do algoritmo de *Canny* a uma imagem [19].

Como a figura 2.5 demonstra, este algoritmo leva a resultados muito robustos, sendo o contorno da bola visível na sua totalidade.

No entanto, em alguns casos, a bola perde a sua forma circular, como pode ser visto na figura seguinte:



Figura 2.6 - Aplicação do algoritmo de *Canny* a uma imagem de um jogador com a bola na mão [19].

Nos casos da figura 2.6, a forma circular da bola é perdida, tornando-se mais difícil a sua deteção.

Quando a bola se move muito rapidamente, o algoritmo de *Canny* não consegue detetar os contornos da bola, levando a que esta não seja detetada. Nesses casos, em [19], é utilizada a técnica de subtração do fundo ao mesmo tempo que transforma a imagem numa imagem binária que tem o fundo preto e os objetos e jogadores brancos. Assim sendo, a subtração leva a um fundo preto com os jogadores e a bola nas suas cores originais, como pode ser observado na figura seguinte:

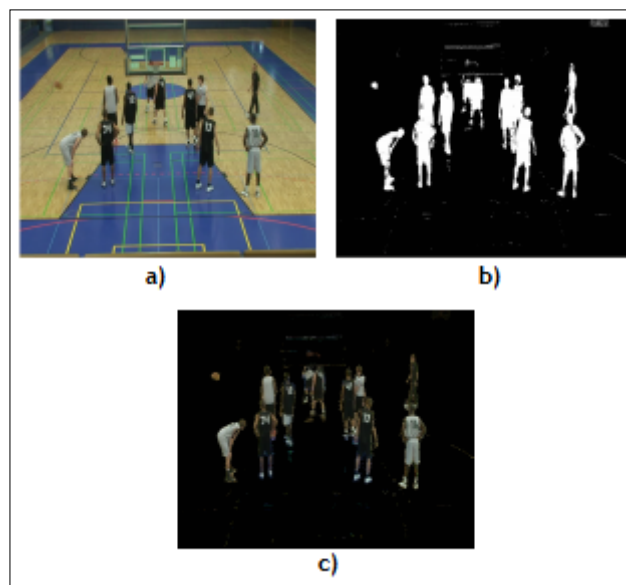


Figura 2.7 - Aplicação da subtração do fundo juntamente com a transformação da imagem numa imagem binária [19]; a) Imagem original; b) Imagem binária; c) Subtração do Fundo.

Como se pode observar na figura anterior, este método conduz-nos a um resultado bastante favorável para a deteção da bola. Aplicando o algoritmo de *Canny* para detetar os contornos da bola na imagem c) da figura 2.7, obtém-se:



Figura 2.8 - Algoritmo da *Canny* aplicado à imagem resultante da aplicação da subtração do fundo [19].

Após aplicado este algoritmo em [19], são demonstrados e aplicados os métodos mencionados acima: *Region Circularity* e *Circles Detection*.

Ainda em [19], estão relatadas duas possibilidades que permitem a determinação da posição da bola em 3D:

- a) *Detection Approach*: Esta aproximação consiste na identificação dos candidatos a bola nos vários pontos de vista (câmaras) e, após isso, tentar determinar a posição 3D através desse resultado;
- b) *Tracking-based approach*: Nesta abordagem, Neus Agelet Riuz considera que a posição da bola só pode ser encontrada dentro de um determinado volume, existindo para isso uma consistência temporal, em que o ponto de partida para detecção é a posição da bola no instante anterior. Esta abordagem consiste em quatro etapas:
 - i. Definir manualmente o ponto de partida da bola em 3D no início da sequência de imagens;
 - ii. Nessa posição, cria-se uma janela de procura 3D, que tem a forma de um cubo;
 - iii. Testam-se as diferentes posições dentro do cubo, projetando a posição de cada ponto em cada ponto de vista;
 - iv. Por fim, analisa-se a probabilidade dessa posição ser a bola, utilizando para isso a cor e a forma desta.



Figura 2.9 - Exemplo da aplicação do algoritmo *Trackin-based approach* [19]: a) Janela 3D de dimensão 60x60x60 cm; b) Janela 3D de dimensão 120x120x120 cm.

2.3 - Ténis

Tendo o Ténis muitas semelhanças com o desporto para que será estudada a monitorização da trajetória da bola (Ténis de Mesa), tornou-se relevante a procura de artigos que retratassem este tema.

Assim sendo, Jinzi Mao, em [3], estuda a monitorização da trajetória de uma bola num jogo de Ténis, começando pela deteção e determinação da posição da bola num cenário vulgar.

Para isso, são ilustradas quatro etapas: Suavização da imagem; Deteção de orlas; Subtração do fundo; Deteção da bola através da cor ou forma.

2.3.1 - Suavização da imagem

Quando se trabalha com imagens provenientes de câmaras em tempo real, a iluminação torna-se um ponto importantíssimo, pois, caso esta não seja uniforme e suficientemente clara, as imagens acabam por ser afetadas de ruído. Como nem sempre se consegue esse tipo de iluminação, torna-se necessário contrapor esse facto utilizando várias técnicas para suavização das imagens, nomeadamente, o filtro de mediana.

Este filtro consiste na suavização da imagem substituindo cada píxel pela mediana dos píxeis que o rodeiam. Assim, caso exista ruído em vários pontos da imagem, esse será suavizado.

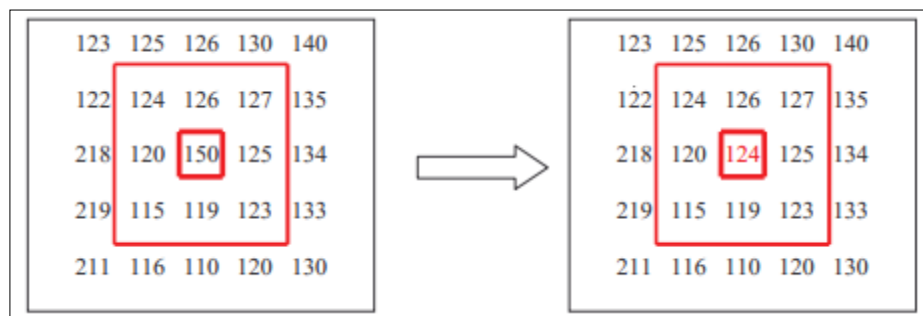


Figura 2.10 - Demonstração da aplicação de um filtro de mediana (cf. [3]).

A figura anterior demonstra um exemplo da aplicação de um filtro de mediana 3x3 numa imagem. O valor do píxel central é 150. Quando aplicado o filtro de mediana 3x3, irá ser feita a mediana entre os píxeis à sua volta num raio ditado pelo tamanho da janela. Neste caso, organizando os valores dos píxeis circundantes, obtêm-se 115, 119, 120, 123, 124, 125, 126, 127, 150. Calculando a mediana, verificamos que o resultado é 124. Por isso, o valor do píxel é substituído por 124, suavizando, assim, a imagem.

2.3.2 - Filtro de Sobel

O filtro de Sobel é uma ferramenta utilizada em [3], para detecção dos contornos de um objeto numa imagem.

Esta ferramenta consiste em duas matrizes 3x3 que são convoluídas com a imagem original. Estas duas matrizes são chamadas de máscaras, sendo que uma delas deteta as linhas horizontais, enquanto a outra deteta as linhas verticais.

$$m_{horizontal} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$m_{vertical} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Após a convolução das duas máscaras com a imagem original, obtém-se duas imagens distintas, sendo que a imagem final é dada por:

$$Imagem_{final} = \sqrt{(m_{horizontal} * Imagem_{original})^2 + (m_{vertical} * Imagem_{original})^2} \quad (2.1)$$

Com este cálculo, obtém-se uma imagem final com todos os contornos evidenciados, como demonstra a figura seguinte:

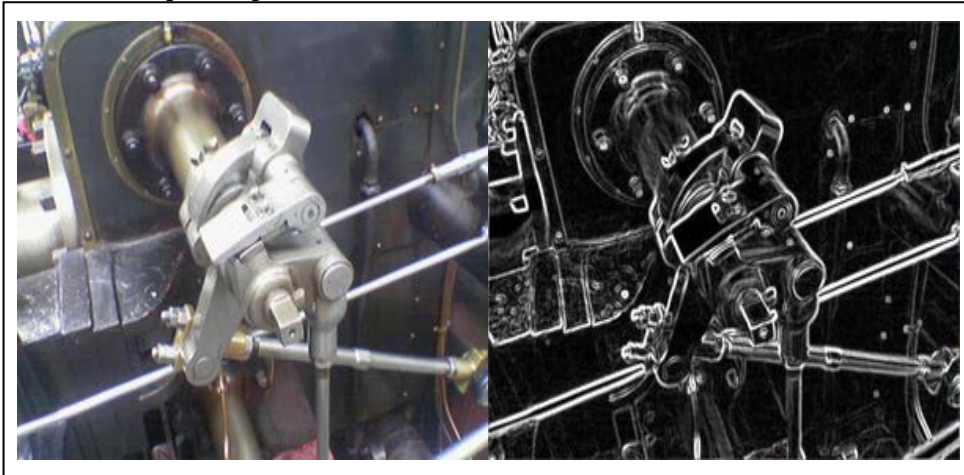


Figura 2.11 - Exemplo da aplicação do filtro de Sobel numa imagem.

2.3.3 - Subtração do Fundo

Para detetar um objeto numa imagem, é vantajoso que não existam objetos à sua volta que se possam confundir com o pretendido. Assim, em [3], está retratado um algoritmo que consiste em subtrair a imagem real (com a bola presente) pelo fundo (a mesma imagem sem a bola). Fazendo este processo, atinge-se uma imagem que só tem presente os contornos dos objetos que se pretendem detetar - o que facilita a deteção e determinação da posição da bola na imagem.

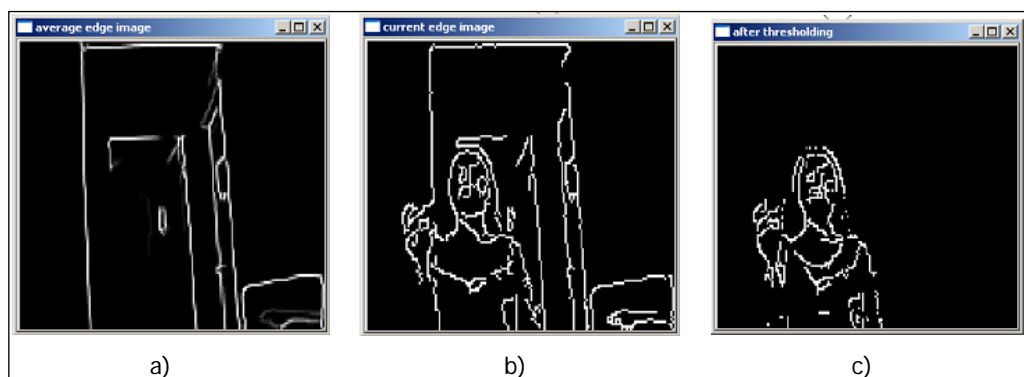


Figura 2.12 - Exemplo da aplicação da subtração do fundo [3]; a) Fundo; b) Imagem Real; c) Subtração de b) por a).

A imagem c) da figura anterior possui todos os contornos dos objetos/pessoas que não estão presentes no Fundo capturado (imagem a)).

2.3.4 - Deteção da bola na imagem

Existem várias formas de detetar objetos numa imagem, nomeadamente, através da forma ou da cor do objeto.

A procura de objetos através da forma consiste na convolução de uma máscara que tem representado o objeto (neste caso um círculo, que corresponde à bola) com a imagem original. Após isso, o píxel que ficar com maior valor na imagem final é o centro do objeto em questão.

Outra forma de detetar objetos numa imagem consiste na procura deste através da sua cor, tornando-se, desta forma, mais simples a deteção do objeto. No entanto, se na imagem existirem objetos da mesma cor, a tarefa para encontrar o objeto pretendido torna-se muito complicada.

Posto isto, ainda é possível combinar estas duas formas de detetar objetos, podendo assim efetuar uma deteção mais robusta do objeto pretendido.

Para além deste documento, existem outros documentos que retratam outras formas para a execução deste objetivo. Em [20], Baris David Ekinci e o Prof. Dr. Muhittin Gokmen, estudam uma forma de monitorizar a trajetória de uma bola em vídeos de jogos de Ténis *offline*.

Neste documento, o sistema utilizado consiste uma câmara fixa que capta o campo por detrás do jogador, como demonstra a figura seguinte:



Figura 2.13 - *Frame* capturada pela câmara fixa situada por detrás de um jogador [20].

De modo a detetar a posição da bola, neste documento, é retratado um algoritmo que permite a extração do fundo através de uma determinada imagem, bem como a deteção das linhas que delimitam a zona do jogo.

Como exposto em [20], é de notar que, num jogo de Ténis, a bola e os jogadores estão em movimento constante. Logo, para se monitorizar a trajetória da bola, ter-se-á isso presente e determinar-se-á o fundo, considerando apenas os corpos estáticos do cenário.

Assim sendo, e como os objetos que se mexem não estarão muito tempo na mesma posição, o fundo pode ser determinado usando um filtro de mediana ao longo do tempo total do vídeo.

A figura seguinte demonstra o resultado da aplicação do filtro de mediana, por forma a determinar o fundo, ou seja, o cenário em que a bola se irá movimentar.



Figura 2.14 - Exemplo da deteção do fundo através de um filtro de mediana [20]; a) *Frame* original; b) Fundo.

Após detetar o fundo, o algoritmo desenvolvido em [20] parte para a deteção das linhas do campo. Estas podem ser distinguidas através da sua forma linear e pela sua cor branca.

Assim, para evidenciar estas na imagem, utiliza-se o mapeamento da imagem do fundo no espaço HSV, em que V representa a intensidade da iluminação que pode ser usada para distinguir as linhas. Uma vez determinado V, é aplicado o operador morfológico *Top-Hat*, que permite a detecção das áreas com mais brilho que os seus píxeis vizinhos. Após isto, é determinado um limiar que permite a detecção apenas das linhas do campo, sendo, por fim, reconstruídas as linhas do campo, de forma nítida e de fácil distinção.

A figura seguinte demonstra a aplicação destes algoritmos.

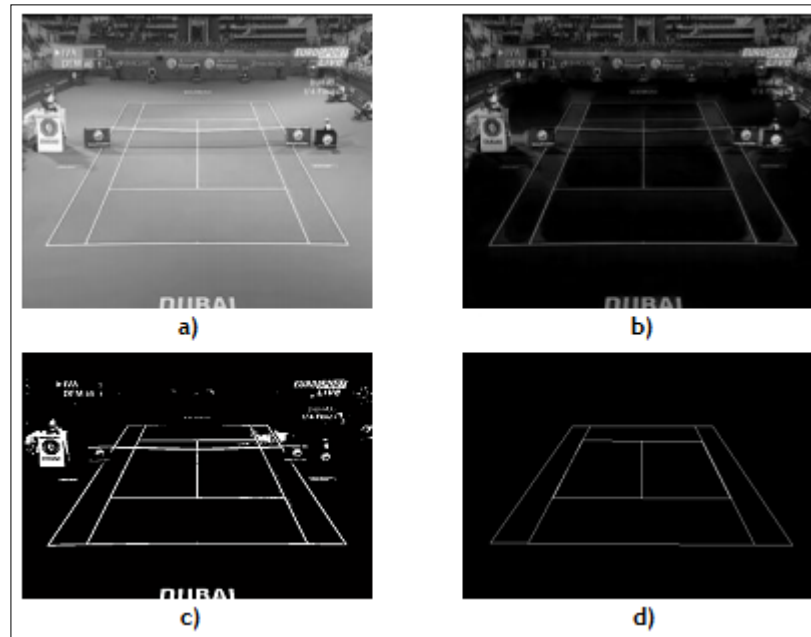


Figura 2.15 - Detecção das linhas com campo [20]; a) Canal V do espaço HSV; b) Imagem resultando da aplicação do *Top-Hat*; c) Aplicação do limiar; d) Reconstrução das linhas.

Após determinado o fundo e o modelo do campo, é retratado, em [20], uma técnica para encontrar os candidatos a bola, baseando-se na subtração de um determinado *frame* pelo fundo, procurando a posição da bola através da cor, tamanho e forma.

Uma vez encontrados os candidatos a bola, é de elevada importância obter um critério que permite discriminar candidatos que não são a bola.

Os candidatos a bola estão ligados aos seus vizinhos mais próximos na *frame* imediatamente a seguir. Assim sendo, os candidatos em que isso não se verifique são automaticamente discriminados.

Por forma a monitorizar a trajetória da bola, em [20] são utilizados dois filtros de *Kalman* 1D, ou seja, um para a posição X e outro para posição Y, que permite estimar a posição seguinte da bola, por forma a ser mais fácil a sua deteção e monitorização.

2.4 - Ténis de Mesa

Muito dos especialistas em sistemas baseados em visão têm vindo a estudar uma forma de conseguir determinar a posição de uma bola em 3D utilizando para isso duas câmaras.

No Ténis de Mesa, já existem sistemas capazes de o fazer. Estes visam a prática deste desporto por *Robots*, de forma autónoma, através de comandos enviados por um computador, baseados na aquisição e processamento de imagens.

Este sistema está ilustrado no artigo [2] e baseia-se na utilização de duas câmaras fulcrais, que são responsáveis pela aquisição da imagem, e um computador, que através dessa aquisição, faz o seu processamento, por forma a encontrar a posição da bola em 3D. Este tipo de sistemas designa-se como sistemas de visão estereoscópica.

2.4.1 - Visão Estereoscópica

A estereoscopia é um fenómeno natural que ocorre nos seres humanos quando observam um determinado cenário. Consiste na projeção de duas imagens nos olhos, que estão em pontos de observação diferentes. Aí o cérebro encarrega-se de fundir as duas imagens, criando um panorama 3D capaz de proporcionar ao ser humano a noção de profundidade, distância, posição e tamanho dos objetos.

Este fenómeno pode ser simulado por duas câmaras e um computador, que correspondem, respetivamente, aos olhos e ao cérebro.

Na figura seguinte, é demonstrada uma configuração de um sistema de visão estereoscópica aplicado ao desporto em questão.

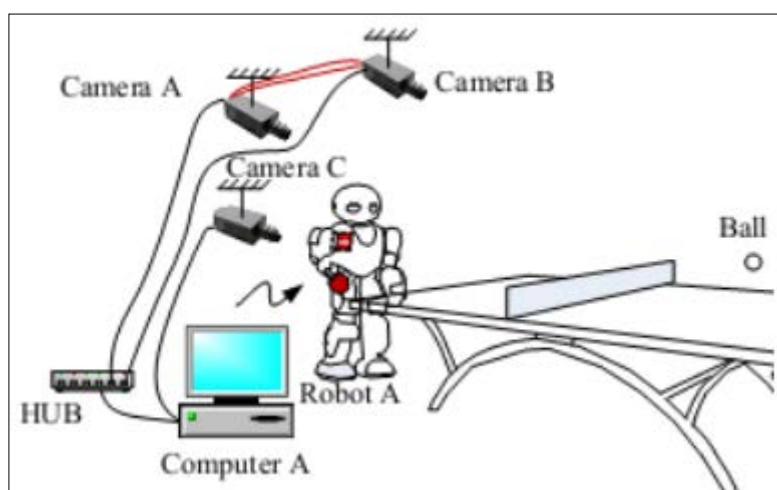


Figura 2.16 - Configuração de um sistema de visão estereoscópica para o Ténis de Mesa (cf. [2]).

Observando a figura, nota-se que as câmaras A e B e o Computador A são os componentes do sistema estereoscópico, enquanto a câmara C é responsável por capturar imagens que visam, juntamente com o computador, determinar a posição do *robot* relativamente à bola.

Outro pormenor deste tipo de sistemas é a sincronização das câmaras, ou seja, as imagens que se observem nas duas câmaras devem ser capturadas exatamente ao mesmo tempo. Esta sincronização é muito importante, pois só assim se conseguem resultados eficazes para a determinação de todos os parâmetros relacionados com a trajetória da bola.

2.4.2 - Sincronização de duas câmaras

As câmaras utilizadas em [2] são dotadas de um módulo FPGA, que permite a ligação entre elas das suas entradas/saídas digitais. Assim sendo, uma saída digital da câmara A é ligada a uma das entradas da câmara B e vice-versa, permitindo assim que as duas câmaras comuniquem entre si, de modo a identificar a altura em que podem capturar o *frame*.

A imagem seguinte demonstra o fluxograma do algoritmo utilizado para a sincronização.

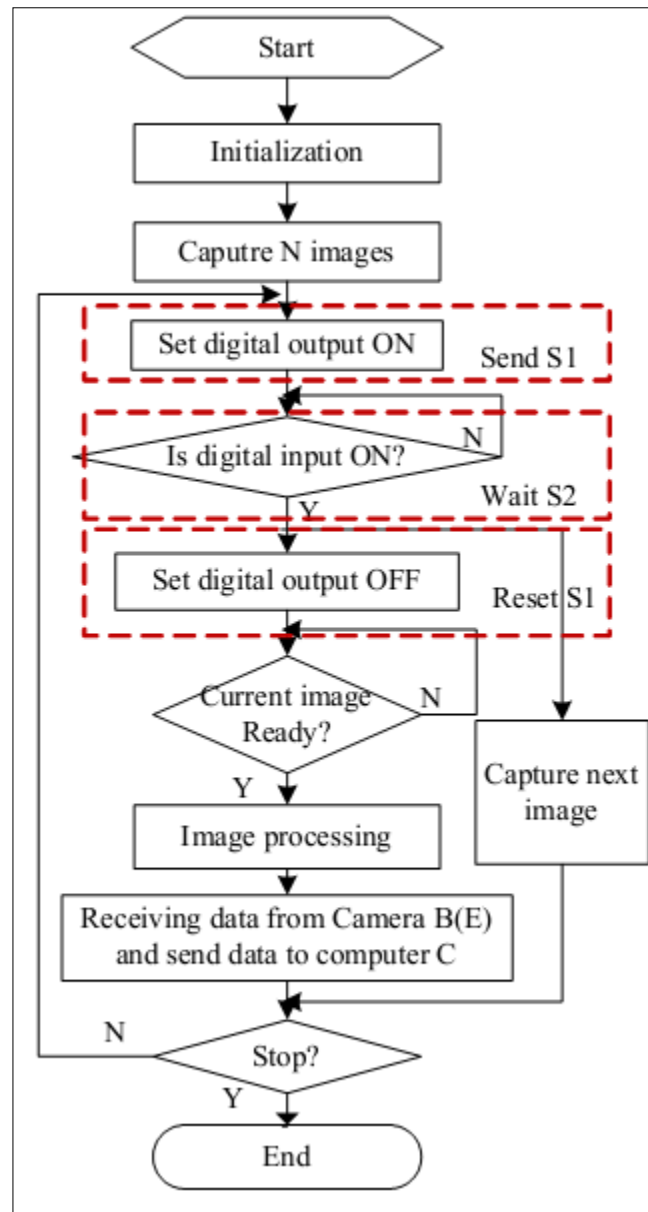


Figura 2.17 - Fluxograma do algoritmo de sincronização das duas câmaras (cf. [2]).

Observando o fluxograma nota-se que: O comando S_1 é ativado quando a câmara A comuta a sua saída a *on* e desativado quando a saída é *off*. De maneira semelhante, a câmara B possui o comando S_2 . Após a inicialização, a câmara A envia o sinal S_1 para a B. Este comando serve para informar que a câmara A está pronta e à espera do comando S_2 . Quando a câmara B recebe S_1 , envia o comando S_2 para a câmara A, começando assim o processo de aquisição e processamento de imagem. Após ser recebido o comando S_2 pela câmara A, esta faz *reset* a S_1 e começa o processo de aquisição e processamento de imagem. Quando a câmara B envia todos os dados para o computador, fica novamente à espera de S_1 .

Por outro lado, quando as duas câmaras enviam os dados extraídos das imagens para o computador, a câmara A envia o comando S_1 para B e fica novamente à espera de S_2 .

Assim é alcançada a sincronização entre as duas câmaras, estimando-se, posteriormente, através do processamento realizado pelo computador, a posição 3D da bola e a sua trajetória.

Capítulo 3

Plataforma de teste utilizada

Neste capítulo serão expostas as ferramentas escolhidas para utilizar no projeto (*Hardware* e *Software*), bem como a interface gráfica desenvolvida para execução do sistema elaborado.

3.1 - Estudo e escolha das ferramentas a utilizar no projeto

Para encontrar as ferramentas que facilitassem a realização do projeto, foi feito um levantamento do *software* e *hardware* que tem sido utilizado pelo laboratório de robótica da FEUP para a realização de Aquisição e Processamento de imagens.

3.1.1 - *Software*

O *software* utilizado ao longo deste projeto foi o programa Lazarus [4], que usa a linguagem Pascal [5]. Este foi estudado e utilizado em algumas unidades curriculares do curso.

O Lazarus oferece liberdade para a criação de interfaces capazes de fazer com que um programa elaborado seja de fácil utilização e compreensão para o utilizador.

Como a execução do projeto passa pela utilização de câmaras para aquisição de imagens de uma bola em movimento, foi utilizada uma biblioteca de componentes de Lazarus para o efeito, o 5dpo.

Esta biblioteca coloca ao dispor vários componentes relacionados com aquisição de imagem, estudados e realizados pela equipa de futebol robótico da FEUP.

Os componentes utilizados neste projeto são:

- TSdpoVideo4L2: permite o controlo de câmaras compatíveis com USB UCV, podendo ser configurados parte dos seus parâmetros, como demonstra a figura seguinte:

Properties		Events
Active	False	
BayerMode	False	
Device		
FrameRate	25	
Handle	-1	
Height	240	
Name	Video	
PixelFormat	uvcpf_YUV420	
Tag	0	
UseLibV4l	True	
Width	320	

Figura 3.1 - Parâmetros de configuração presentes no componente TSdpoVideo4L2.

- **TSdpoFastForm**: componente capaz de criar uma janela que é útil para pintura rápida de imagens, e permite o controlo total de ação da tinta, útil para realizar o processamento de imagens em tempo real. Este componente permite ainda o ajuste de alguns parâmetros da janela, como demonstra a figura seguinte:

Properties		Events
AlwaysOnTop	False	
Caption	Image	
Height	240	
Icon		
ImageType	itNormal	
LeftPos	0	
Name	Image	
Position	poCenter	
Tag	0	
TopPos	0	
Width	320	

Figura 3.2 - Parâmetros de configuração presentes no componente TSdpoFastForm.

Esta biblioteca encontra-se publicada na internet em [5].

3.1.2 - Hardware

Devido à velocidade que a bola atinge num jogo (aproximadamente 120km/h), são necessárias câmaras com uma taxa de fps na ordem dos 100fps pois só assim se consegue imagens com pouco arrastamento, de modo que a bola não fique distorcida e, dessa forma, torne o processamento da imagem mais complicado.

Para isso, foi feito um levantamento das câmaras que os laboratórios da FEUP têm ao seu dispor, chegando à conclusão que existe uma câmara PointGrey USB 3.0 [6] capaz de filmar a 120fps.



Figura 3.3 - Câmara PointGrey FL3-U3-13S2M-CS USB3.0 [6].

Esta câmara proporciona uma velocidade muito rápida, mas, devido à interface que usa para capturar imagens, é necessário um computador que tenha interface USB 3.0, acompanhado de uma elevada velocidade de processamento para gravação das imagens no disco.

Assim, tornou-se necessário efetuar testes para determinar se seria ou não vantajosa a utilização desta mesma câmara. Os testes realizados consistiram em fazer passar a bola pela janela de captura da câmara, para que se pudesse entender se seria possível ou não efetuar um processamento robusto sobre essa imagem.

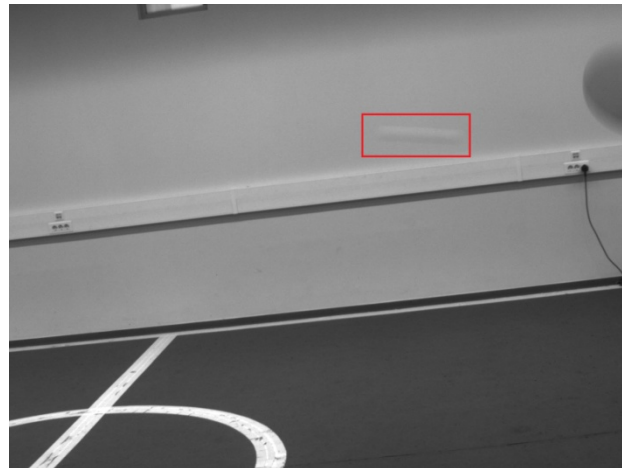


Figura 3.4 - Imagem capturada pela câmara PointGrey FL3-U3-13S2M-CS USB3.0, com a bola demonstrada no retângulo vermelho.

Através da imagem, é possível concluir-se que, apesar de a câmara ser de alta velocidade, o resultado final não é o desejável, pois a bola é quase irreconhecível através da sua forma. Para além disso, durante a realização dos testes, sentiu-se muita dificuldade no processo para aquisição das imagens, pois, o computador, apesar de ter muita velocidade de processamento, tornou-se muito lento devido à gravação das imagens no disco.

Outras dificuldades sentidas foram o não se conseguir arranjar um computador com interface USB3.0 e também a não aquisição de outra câmara igual para a realização de um processamento em 3D, devido ao seu elevado custo.

Posto isto, e sendo de elevada importância conseguir câmaras fiáveis e de baixo valor para o desenvolvimento do projeto, foram utilizadas as câmaras Eye USB 2.0 utilizadas na PS3, que permitem velocidades de captura na ordem dos 30fps, podendo estas atingir velocidades na ordem dos 120fps, ajustando para isso os parâmetros do componente TSdpoVideo4L2 (Figura 3.1). Para além disto, estas câmaras utilizam a interface USB 2.0, que está, hoje, presente em todos os computadores.



Figura 3.5 - Câmaras Eye utilizadas na PS3.

Ao longo do projeto, serão utilizadas estas câmaras para aquisição das imagens/vídeo devido à facilidade de ligação com o computador, bem como à compatibilidade com o componente TSdpoVideo4L2, já explicado anteriormente.

3.2 - Interface Gráfica (Lazarus)

Ao longo do documento, serão ilustradas várias imagens da interface gráfica realizada para dar ordem ao processamento, bem como para mostrar os resultados finais do mesmo.

Torna-se, então, necessário apresentar a interface gráfica realizada e utilizada ao longo do projeto.

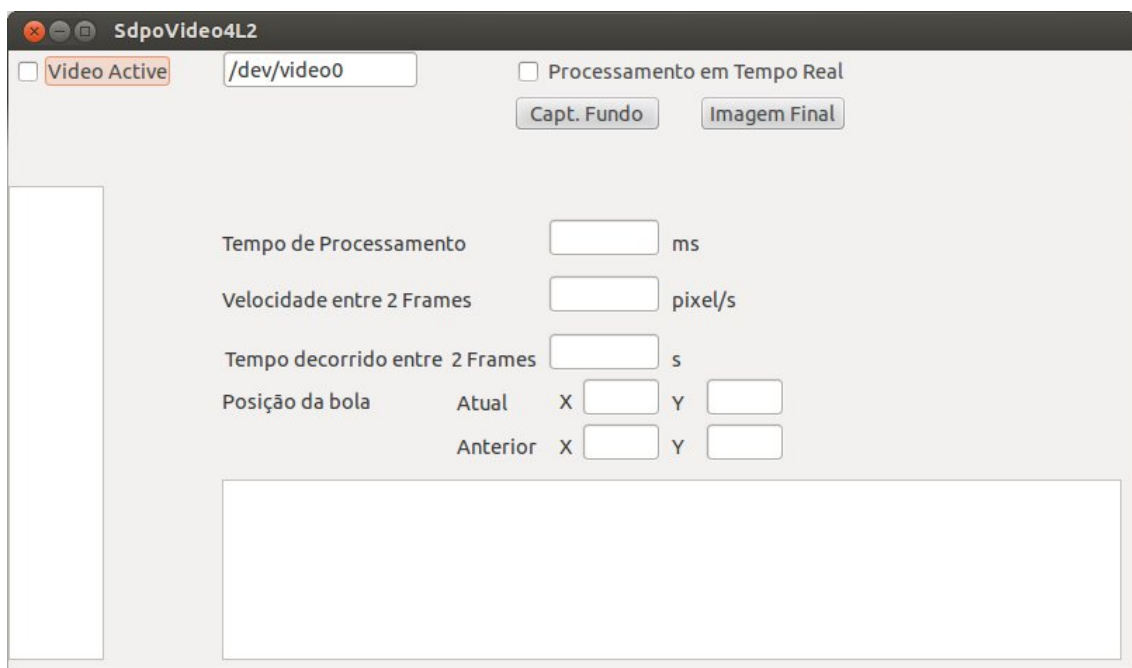


Figura 3.6 - Interface gráfica realizada em Lazarus para o projeto.

Na figura anterior, pode observar-se a existência de vários botões e campos que proporcionam a execução do sistema realizado.

- **Video Active:** Como o nome indica, permite ativar a captura de imagens pela câmara que tem o endereço ao lado "dev/video0". Este endereço pode ser mudado consoante a câmara a utilizar;

- **Capt. Fundo:** Permite capturar o fundo da "cena", isto é, o cenário em que a bola se vai movimentar.

- **Imagem Final:** Permite, depois de ser capturado o fundo, o processamento da imagem, de modo a encontrar as coordenadas da bola;

- **Tempo de Processamento:** tempo que o programa demora a processar cada imagem;

- **Velocidade entre duas Frames:** permite visualizar a velocidade da bola em píxeis/s entre duas determinadas *frames*;

- Tempo decorrido entre 2 Frames: tempo decorrido entre as 2 *frames* que permitem calcular a velocidade da bola;

- Posição da bola: permite exibir a posição x e y atuais e anteriores da bola.

Estes campos permitem a execução do programa em tempo parcial, ou seja, processar uma determinada *frame* quando é premido o botão “Imagem final”.

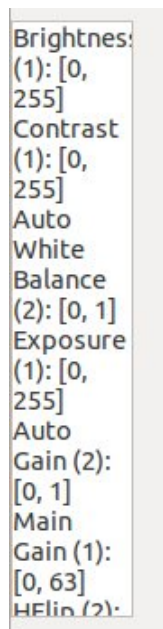
Outro campo importante e que permite o processamento da imagem em tempo real é o seguinte:

- Processamento em Tempo Real: quando selecionado permite processar em tempo real o vídeo capturado pela câmara e exibe no campo branco inferior, vários parâmetros, como por exemplo:

“Posx=120 Posy=195 Velocidade=3605.24309551766pixel/s Tempo decorrido=0.03s”

Em que Posx representa a posição da bola no eixo x, Posy a posição da bola no eixo y. A Velocidade é calculada entre duas *frames* consecutivas e o Tempo Decorrido consiste no tempo de desfasamento entre essas frames.

Para além dos campos já mencionados, falta mencionar o campo em branco do lado esquerdo que permite visualizar alguns parâmetros do vídeo capturado. Um exemplo está retratado na figura seguinte:



```
Brightnes:  
(1): [0,  
255]  
Contrast  
(1): [0,  
255]  
Auto  
White  
Balance  
(2): [0, 1]  
Exposure  
(1): [0,  
255]  
Auto  
Gain (2):  
[0, 1]  
Main  
Gain (1):  
[0, 63]  
HFlip (2):
```

Figura 3.7 - Exibição dos Parâmetros do vídeo capturado pela câmara.

Capítulo 4

Aquisição e Processamento de imagem

4.1 - Aquisição de Imagem

Como já foi dito anteriormente, foi utilizado o programa Lazarus juntamente com a biblioteca 5dpo para efetuar toda a aquisição e processamento de imagem.

Devido à simplicidade de processamento de imagens a preto e branco face às imagens a cores, optou-se por adquirir imagens a preto e branco, com uma resolução de 320*240 e uma taxa de 30fps. Para isso, usou-se o componente TSdpoVideo4L2, que nos permite configurar todos esses parâmetros e também escolher a codificação pretendida.

A codificação utilizada foi a YUV420, devido à compatibilidade com as câmaras utilizadas. Esta codificação permite adquirir imagens a preto e branco, sendo que Y corresponde ao brilho do píxel e U e V são valores de crominância que permitem obter uma imagem a cores através da imagem a tons de cinzento obtida pelos valores de Y.

Uma imagem consiste numa matriz em que cada elemento corresponde a um píxel da imagem e tem o seu próprio valor. Com a codificação usada (YUV420), é necessário que, em cada píxel da imagem, sejam guardados os valores Y, U e V. Logo, em cada píxel foi guardada uma *longWord*, que consiste numa variável com o tamanho de 4 *bytes*, em que o *byte* mais à direita corresponde ao valor V, o *byte* a seguir ao valor U, e os outros dois *bytes* ao valor Y.

Assim, tornou-se possível adquirir a imagem com o componente TSdpoVideo4L2 e imprimir a mesma recorrendo ao componente TSdpoFastForm.

Um exemplo da aquisição da imagem com a codificação mencionada encontra-se na figura 4.2, enquanto que, na figura 4.1, pode observar-se a imagem que a câmara captura usando os drivers proprietários para o efeito.



Figura 4.1 - Exemplo de imagem a cores.



Figura 4.2 - Exemplo de imagem capturada a preto e branco com a codificação YUV420.

4.2 - Processamento de Imagem

Uma vez adquiridas as imagens, foram então estudadas e escolhidas as melhores formas de processar as imagens, por forma a tornar-se mais fácil detetar e determinar a posição da bola. Foram realizadas duas abordagens semelhantes, que serão explicadas de seguida.

4.2.1 - Primeira Abordagem

A primeira abordagem, partindo do algoritmo falado no Capítulo 2, alicerça-se em quatro fases de processamento:

4.2.1.1 - Filtro de Mediana

Realização de um filtro de mediana, de forma a eliminar ruído presente nas imagens. Este ruído deve-se ao facto da iluminação presente no local em que o projeto foi elaborado não ser das melhores.

Por forma a corrigir este fator, optou-se então por utilizar uma filtro de mediana 3x3 que permitiu suavizar a imagem, pois, quanto mais nítida for a imagem, melhores resultados serão alcançados aquando do processamento que visa detetar e estimar a posição da bola.

Um exemplo do resultado deste filtro está retratado nas figuras seguintes.



Figura 4.4 - Imagem capturada sem processamento.



Figura 4.3 - Filtro de Mediana aplicado à imagem da Figura 4.3.

Nas imagens anteriores, pode comprovar-se o efeito da aplicação de um filtro de mediana 3x3, notando-se que a imagem do lado direito se encontra suavizada. Observando as duas imagens, principalmente nas partes mais escuras, pode observar-se a eliminação do ruído presente na imagem da esquerda.

Daí, conclui-se que o filtro de mediana demonstra ser uma forma eficaz de eliminar ruído nas imagens.

4.2.1.2 - Filtro de Sobel

Com vista a detetar os contornos da bola na imagem, foi utilizado o filtro de Sobel. Como explicado anteriormente, este algoritmo permite detetar, numa imagem, todos os contornos dos objetos, ficando assim com uma imagem de fundo preto com os contornos a branco.

Nas figuras seguintes, pode ser observada a aplicação deste filtro numa imagem já suavizada através da aplicação do filtro de mediana. Nesta mesmas figuras já se encontra presente a bola.



Figura 4.6 - Imagem capturada com Filtro de Mediana aplicado.



Figura 4.5 - Filtro de Sobel aplicado à imagem da Figura 4.5.

Pelas figuras anteriores observa-se que o filtro de Sobel permite a detecção do contorno da bola que se encontra dentro do quadrado vermelho. Mas para além dos contornos da bola, são detetados também outros contornos de outros objetos que nada têm a ver com a bola e que, por isso, seria conveniente eliminar.

4.2.1.3 - Subtração do Fundo

Para evitar que outros objetos sejam detetados e possam ser confundidos com a bola, tornou-se necessário encontrar uma forma de os excluir.

Uma ferramenta que o permite é a Subtração do Fundo. Esta Ferramenta consiste, como o nome indica, em subtrair a imagem original (imagem com a bola presente) pelo fundo. Este fundo é o cenário em que a bola se vai movimentar.

Assim sendo, a interface gráfica desenvolvida e demonstrada na figura 3.6 permite a utilização de dois botões, um que permite capturar o fundo (botão "Capt. Fundo"), aplicando o filtro de mediana e o filtro de Sobel ao mesmo, e outro (botão "Imagem Final") que permite capturar a imagem com a bola presente, aplicar os mesmos filtros e fazer a subtração desta pela imagem do fundo.

As imagens seguintes mostram um exemplo da junção da aplicação do filtro de mediana e de Sobel com a subtração do fundo.

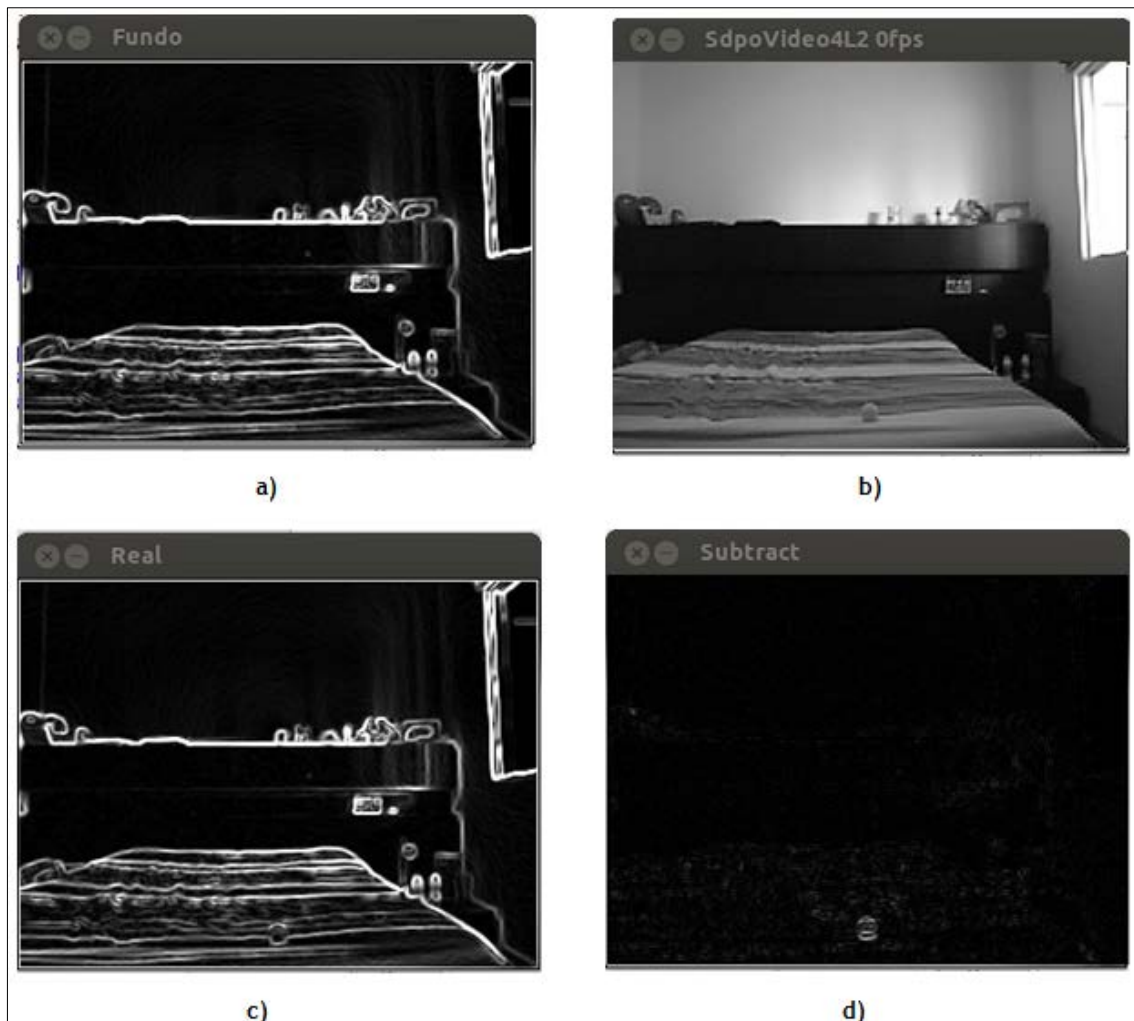


Figura 4.7 - Exemplo do processamento da 1ª Abordagem com a bola numa superfície de tons misturados; a) Fundo; b) Imagem capturada; c) Imagem b) Processada; d) Subtração de c) por a).

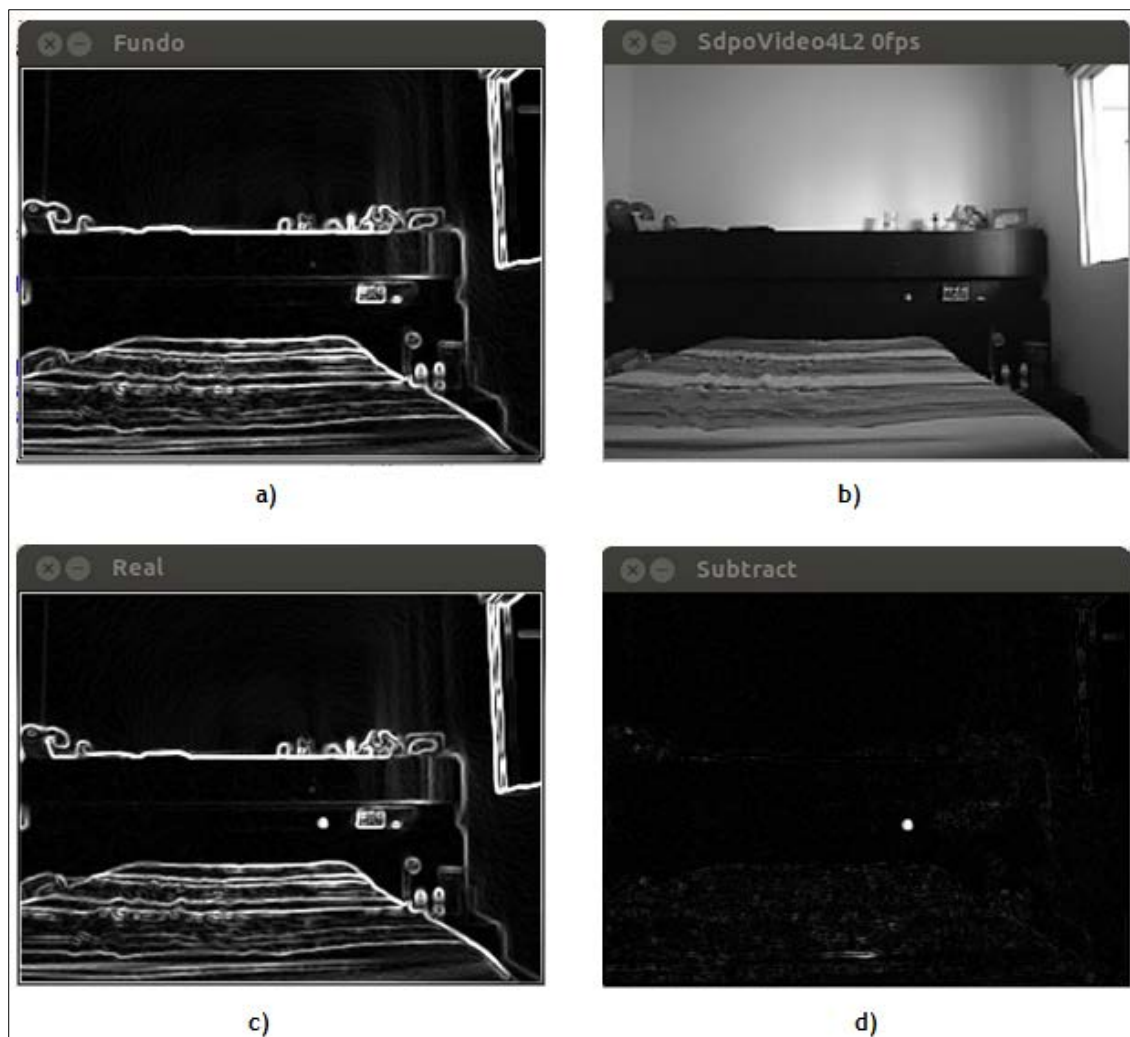


Figura 4.8 - Exemplo do processamento da 1ª Abordagem com a bola numa superfície de cor preta; a) Fundo; b) Imagem capturada; c) Imagem b) Processada; d) Subtração de c) por a).

Na figura 4.8 pode observar-se que o processamento levou a uma imagem com fundo preto e a bola a branco. No entanto, na figura 4.7, observa-se que, quando a bola se encontra numa superfície que tem vários tons, o processamento não se mostra eficaz, uma vez que não se torna possível estimar a posição da bola com facilidade pois os contornos não são muito coerentes, como mostra a figura 4.7 d).

Podemos então afirmar que, devido a este facto e também ao facto do tempo de processamento estar na ordem dos 40ms, não se torna vantajoso a utilização deste processamento, pois, para uma taxa de 30 fps, este mostra-se elevado, uma vez que, por cada imagem, só é possível processar em $1/30 = 33\text{ms}$.

Contudo, esta primeira abordagem serviu para a familiarização com o programa Lazarus e com a biblioteca de componentes 5dpo.

4.2.2 - Segunda Abordagem

Devido ao processamento utilizado na Primeira Abordagem não se ter mostrado eficaz, estudou-se e implementou-se uma nova abordagem, que utiliza o mesmo princípio da abordagem anterior, não tendo prescindido do filtro de mediana e da subtração do Fundo.

Posto isto, de forma a obter-se com maior facilidade e eficácia a posição da bola, realizou-se uma pesquisa da bola na imagem, utilizando-se uma máscara para o efeito.

4.2.2.1 - Aplicação de uma máscara da bola na imagem

Uma máscara consiste numa matriz, com um determinado tamanho, que tem esboçada a forma do objeto que se pretende encontrar, neste caso um círculo que representa a bola.

O tamanho da matriz foi escolhido, consoante a distância a que estarão as câmaras. Para isso foi impressa numa tabela uma imagem com a bola presente a uma distância de aproximadamente 3 metros, observando quantos píxeis é que a mesma ocupava na imagem.

Deste modo foi assumido o tamanho 13x13 para a máscara, com um diâmetro de 9 píxeis para a bola. Os píxeis que esboçam o círculo têm valor de 1 para durante a convolução com a imagem original, estes sejam somados. Os píxeis à volta têm valor de -1 para que assim sejam subtraídos aquando da convolução. A tabela seguinte mostra a matriz que forma a máscara da bola.

Tabela 4.1 - Máscara 13x13.

-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	1	1	1	1	1	-1	-1	-1	-1
-1	-1	-1	1	1	1	1	1	1	1	-1	-1	-1
-1	-1	1	1	1	1	1	1	1	1	1	-1	-1
-1	-1	1	1	1	1	1	1	1	1	1	-1	-1
-1	-1	1	1	1	1	1	1	1	1	1	-1	-1
-1	-1	1	1	1	1	1	1	1	1	1	-1	-1
-1	-1	1	1	1	1	1	1	1	1	1	-1	-1
-1	-1	-1	1	1	1	1	1	1	1	-1	-1	-1
-1	-1	-1	-1	1	1	1	1	1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

A utilização de uma máscara para procura de um objeto numa imagem consiste em realizar a operação de convolução entre a imagem original e a máscara em questão.

A convolução entre imagens consiste numa operação local, ou seja, para cada píxel é realizada a operação.

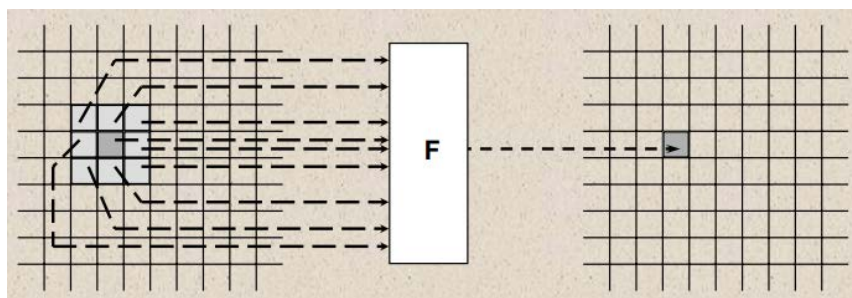


Figura 4.9 - Exemplo de operação pontual entre imagens.

A imagem anterior permite demonstrar como são realizadas as operações pontuais entre imagens, sendo que o "F" corresponde, neste caso, à função convolução.

Neste tipo de operações cada ponto da imagem resultado é obtido através da Função F, que usa como argumentos o ponto homólogo da imagem original e um conjunto de pontos vizinhos desse, que são definidos pelo tamanho da máscara utilizada (máscara 3x3 na figura 4.9).

Como nesta abordagem a Função "F" corresponde a uma convolução, sobrepõe-se a máscara em cada píxel da imagem original, sendo o resultado do píxel central a soma da multiplicação dos píxeis sobrepostos da imagem e da máscara.

Posto isto, foi elaborado o código que permite fazer esta operação e testado colocando a bola no cenário capturado pela câmara.

A figura seguinte demonstra o resultado.

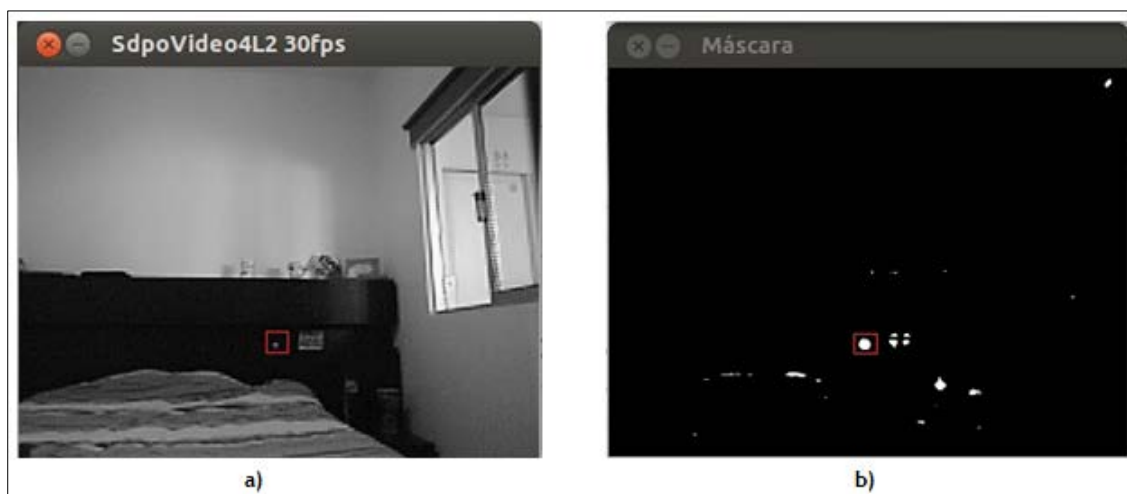


Figura 4.10 - Demonstração do resultado da utilização de uma máscara 13x13; a) Imagem capturada; b) Máscara Aplicada à imagem a).

Pela figura anterior, pode observar-se um bom funcionamento da aplicação da máscara 13x13, pois com a imagem b) da figura 4.10 torna-se mais facilitada a deteção da bola.

Os restantes objetos que aparecem nessa figura advêm do facto de, no teste efetuado, não estar incluído todo o processamento, nomeadamente, a Subtração do Fundo.

No entanto, a grande desvantagem deste processo, prende-se com o tempo de processamento estar na ordem dos 73ms, que mostra ser bastante elevado, e também com o facto de a bola ter tamanho diferente nas duas imagens, que é originado pelo tamanho um pouco exagerado da máscara.

Nasce então a necessidade de diminuir o tamanho da máscara, pois, quanto menor for o tamanho da máscara, menor será o tempo de processamento e mais eficaz será a deteção da bola.

O tamanho da máscara foi então mudado para 7x7, com um diâmetro de 5 píxeis para a bola, como demonstra a tabela seguinte:

Tabela 4.2 - Máscara 7x7.

-1	-1	-1	-1	-1	-1	-1
-1	-1	1	1	1	-1	-1
-1	1	1	1	1	1	-1
-1	1	1	1	1	1	-1
-1	1	1	1	1	1	-1
-1	-1	1	1	1	-1	-1
-1	-1	-1	-1	-1	-1	-1

O resultado da aplicação dessa mesma máscara é ilustrado na figura seguinte.

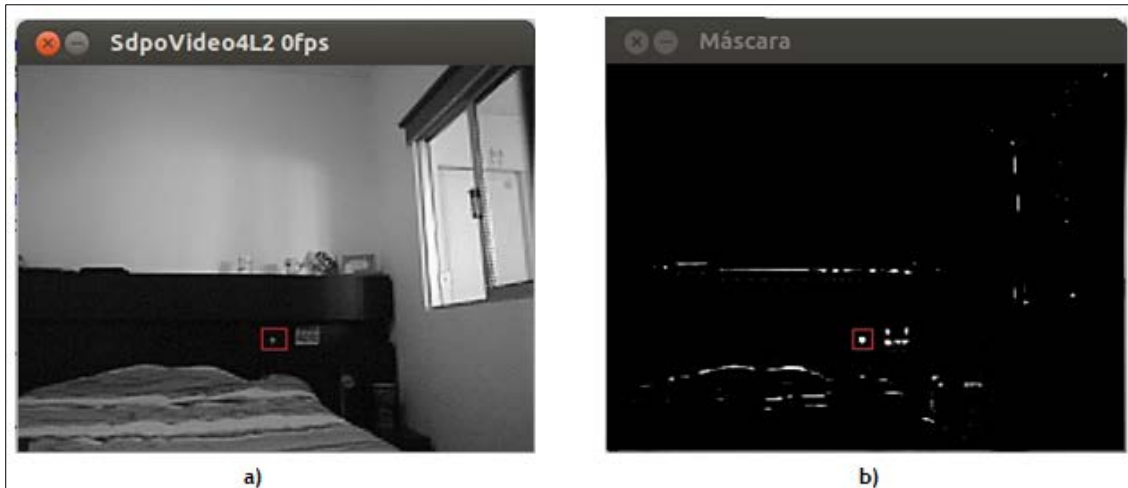


Figura 4.11 - Demonstração do resultado da utilização de uma máscara 7x7; a) Imagem capturada; b) Máscara Aplicada à imagem a).

Em semelhança com a figura 4.10, a figura 4.11 demonstra o mesmo resultado, sendo que, nesta última, a bola tem um tamanho semelhante nas duas imagens, o que se torna vantajoso para determinar a sua posição.

Para além desta vantagem, a redução do tamanho da máscara traz uma vantagem acrescida: a diminuição significativa do tempo de processamento, que está na ordem dos 25ms, que se revela ser o ideal para a taxa de fps utilizada (30fps, ou seja 1 imagem por cada 33ms).

4.2.2.2 - Junção das 3 fases de processamento

Tendo chagado ao tamanho ideal para a máscara a aplicar, torna-se agora necessário juntar as três partes constituintes do processamento (Filtro de Mediana, Subtração do Fundo, Aplicação da Máscara).

O resultado obtido foi o seguinte:

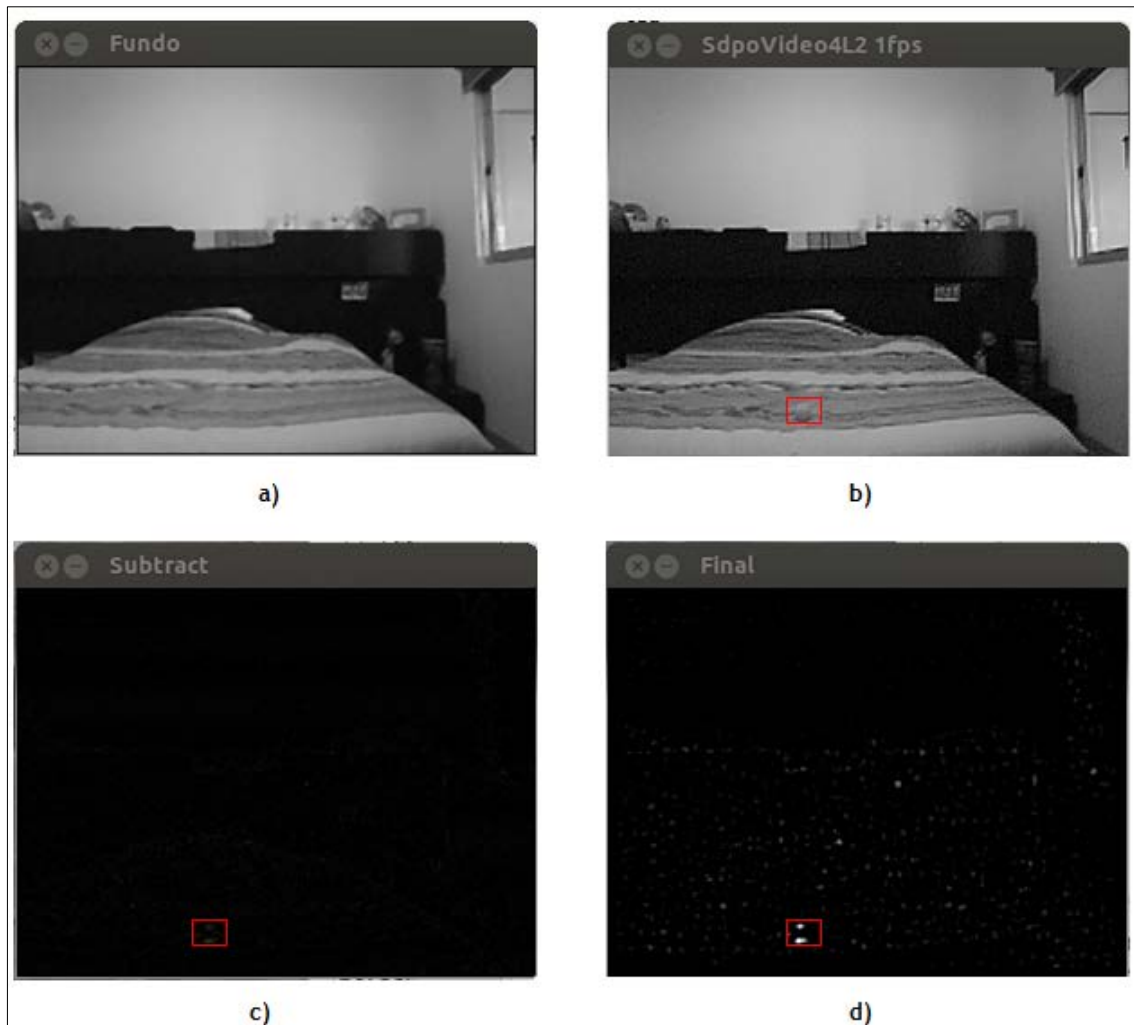


Figura 4.12 - Exemplo do processamento da 2ª Abordagem; a) Fundo; b) Imagem capturada; c) Subtração de b) por a) ; Imagem final após a aplicação da máscara.

Na figura anterior pode observar-se um exemplo da aplicação de acordo com a segunda abordagem. O tempo de processamento está na ordem dos 55ms. Torna-se, por isso, vantajoso encontrar uma alternativa para diminuir este tempo.

Apesar do filtro de mediana ser eficaz para eliminação de ruído, este leva muito tempo a processar (30ms). Assim resolveu-se excluir o filtro de mediana e observar como o algoritmo se comporta.

A figura seguinte demonstra o mesmo processo que a Figura 4.12, excluindo apenas o filtro de mediana.

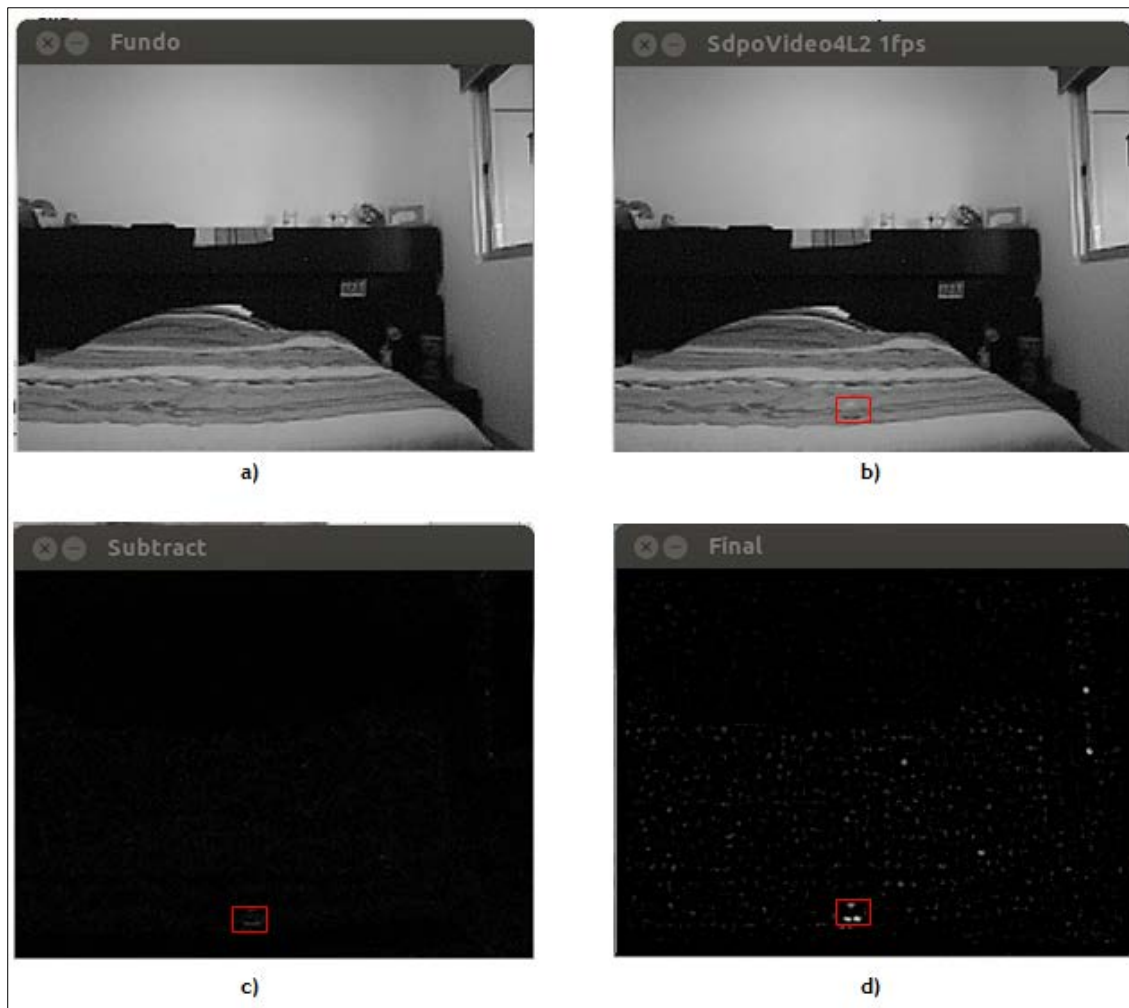


Figura 4.13 - Exemplo do processamento da 2ª Abordagem sem a utilização do filtro de Mediana; a) Fundo; b) Imagem capturada; c) Subtração de b) por a) ; Imagem final após a aplicação da máscara.

A figura anterior comprova que excluindo o filtro de mediana o algoritmo continua a comportar-se como esperado, chegando a uma imagem final em que se torna capaz de detetar a zona em que se encontra a bola (junção de vários píxeis brancos).

A grande vantagem deste processo está no tempo de processamento, que caiu para os 22ms.

Devido à exclusão do filtro de mediana, está presente algum ruído na imagem final.

Para este não interferir na deteção da bola, optou-se por “correr” a máscara apenas nas zonas prováveis onde se encontrará a bola, isto é, nos píxeis que na imagem subtraída se evidenciarem através da cor como sendo a bola.

Essa otimização será demonstrada na figura seguinte:

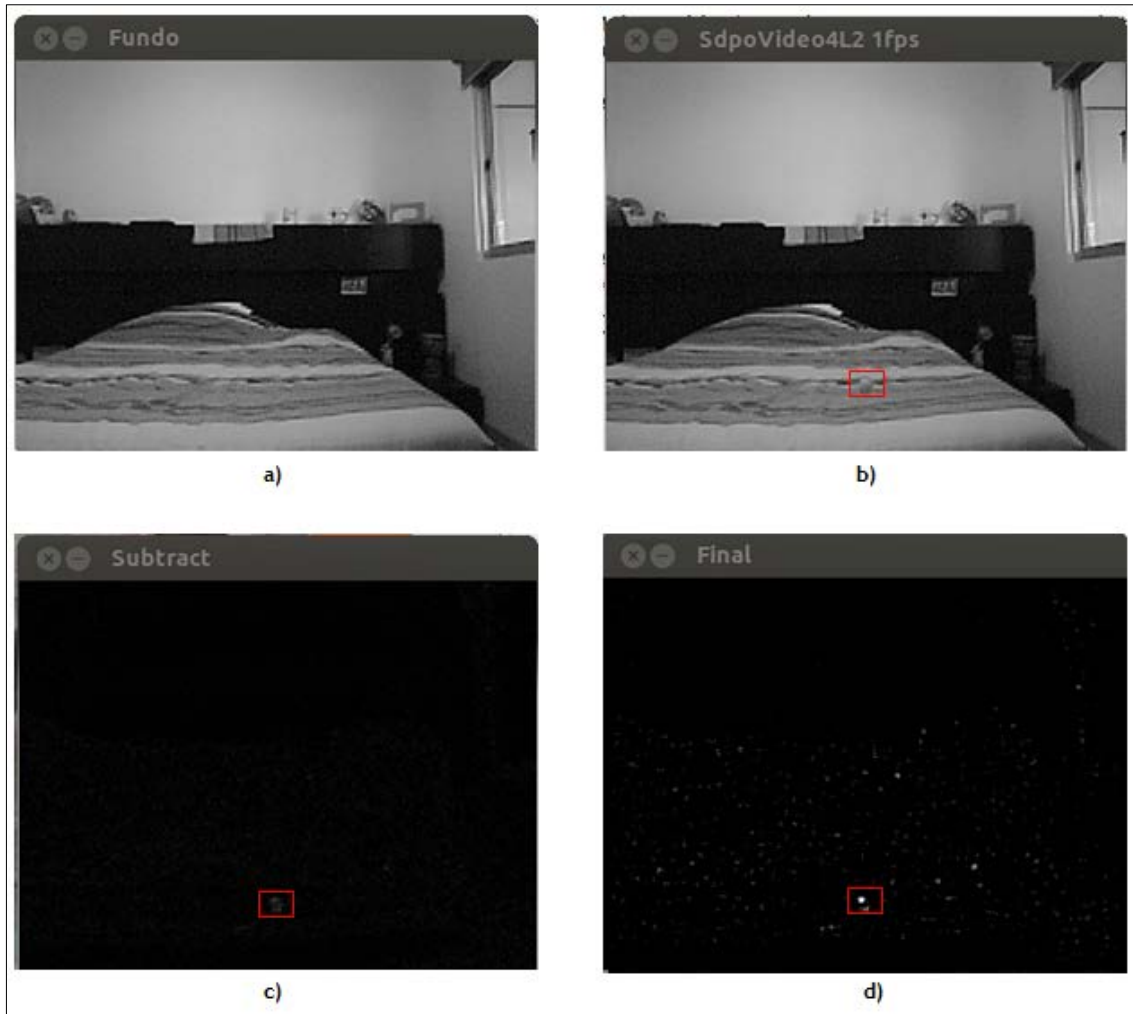


Figura 4.14 - Exemplo do processamento da 2ª Abordagem otimizado; a) Fundo; b) Imagem capturada; c) Subtração de b) por a) ; Imagem final após a aplicação da máscara.

A figura 4.14 vem comprovar que o algoritmo desenvolvido se mostra, mais uma vez, eficaz, podendo ser determinada a posição da bola com facilidade.

De notar ainda que, comparado a imagem d) das figuras 4.13 e 4.14, esta última encontra-se com menos ruído, o que se deve ao facto de ser aplicada a máscara apenas nos pixels que poderão ser a bola. Apesar de estar presente algum ruído na imagem, este não afeta a correta deteção da bola.

Esta otimização traz ainda uma vantagem extra que se deve à diminuição, ainda que substancial, do tempo de processamento, passando este para 17ms, o que se torna muito vantajoso face ao tempo disponível para o processamento (aproximadamente 33ms).

4.3 - Detecção e Determinação da Posição da bola na imagem

Uma vez processada a imagem, procedeu-se à deteção e determinação da posição da bola. Assim sendo, desenvolveu-se um algoritmo capaz de, através da forma da bola, determinar o seu centro.

Como foi exposto anteriormente, a aplicação de uma máscara consiste numa convolução desta com a imagem original.

Neste sentido, a forma encontrada para determinar o centro da bola consiste em detetar o píxel da imagem original que, quando aplicada a convolução com a máscara, fica com o valor máximo, isto é, imaginemos a seguinte matriz de uma imagem com uma bola (píxeis com valor 200) e com um fundo em tons aleatórios:

Tabela 4.3 - Matriz de uma imagem exemplo com uma bola.

15	15	15	15	15	15	15	15	15	60	60	60	15	15	15	15	15	15	15	18
15	15	18	15	15	15	15	15	18	30	60	60	18	15	18	18	15	15	15	15
15	15	15	15	15	15	15	15	15	15	20	60	15	15	15	15	15	60	60	60
60	60	60	60	60	60	60	15	15	15	15	15	60	60	18	15	18	60	30	60
60	30	60	60	30	60	60	15	15	15	15	18	60	60	60	15	15	15	60	60
60	30	20	60	30	60	60	60	60	15	15	15	30	60	60	60	60	18	30	60
15	15	15	60	30	60	60	60	30	60	60	60	15	20	60	60	60	15	15	18
15	15	18	60	30	20	20	60	30	60	30	60	15	15	15	30	60	15	15	15
15	15	15	60	60	60	60	15	15	15	60	60	15	15	18	15	20	60	60	60
60	60	60	60	30	60	60	15	15	18	30	60	15	15	15	15	15	60	30	60
60	30	15	15	15	20	60	15	15	15	30	20	60	60	60	15	15	15	15	18
60	30	15	15	15	15	15	60	60	60	60	60	60	30	60	15	15	15	15	15
15	15	15	15	15	15	18	60	30	60	30	60	60	200	200	200	60	60	60	60
15	15	60	60	15	15	15	60	30	20	15	20	200	200	200	200	200	60	30	60
15	15	15	15	15	15	15	15	15	15	15	15	200	200	200	200	200	15	15	18
15	15	18	15	15	15	18	15	15	18	15	15	200	200	200	200	200	15	15	15
15	15	15	15	15	15	15	15	15	15	60	30	60	200	200	200	60	60	60	60
60	60	60	60	60	60	60	60	60	60	15	15	15	60	30	60	60	60	30	60
60	30	60	60	60	30	60	60	30	60	15	15	18	15	60	60	15	15	60	60
60	30	20	60	60	30	20	60	30	20	15	15	15	18	30	60	18	18	30	60

Aplicando a máscara da tabela 1.3 nos píxeis que estão a azul temos o seguinte resultado:

$$\begin{aligned}
 - \quad 30: & \quad (15 \times -1) + (15 \times -1) + (15 \times -1) + (15 \times -1) + (15 \times -1) + (15 \times -1) + \\
 & (15 \times -1) + (60 \times -1) + (60 \times -1) + (60 \times 1) + (60 \times 1) + (60 \times 1) + (60 \times -1) + \\
 & (15 \times -1) + (30 \times -1) + (60 \times 1) + (60 \times 1) + (30 \times 1) + (60 \times 1) + (60 \times 1) + \\
 & (15 \times -1) + (30 \times -1) + (20 \times 1) + (60 \times 1) + (30 \times 1) + (60 \times 1) + (60 \times 1) + \\
 & (60 \times -1) + (15 \times -1) + (15 \times 1) + (60 \times 1) + (30 \times 1) + (60 \times 1) + (60 \times 1) + \\
 & (60 \times -1) + (15 \times -1) + (18 \times -1) + (60 \times 1) + (30 \times 1) + (20 \times 1) + (20 \times -1) + \\
 & (60 \times -1) + (15 \times -1) + (15 \times -1) + (60 \times -1) + (60 \times -1) + (60 \times -1) + (60 \times -1) + \\
 & (15 \times -1) = 107;
 \end{aligned}$$

- 60: Fazendo o mesmo cálculo ... = -802;
- 200: ... = 3055;
- 200: ... = 882;

Dos quatro píxeis apresentados o que, após a aplicação da máscara, fica com maior valor é o primeiro 200, que fica com o valor máximo (3055). Observando a tabela anterior, facilmente conclui-se que esse mesmo píxel é de facto o centro da bola, confirmando-se assim que o método explicado funciona corretamente.

A figura seguinte demonstra o resultado da aplicação deste método, podendo observar-se a imagem de fundo, a imagem Real e a Imagem Real com um quadrado desenhado à volta das coordenadas calculadas para o centro da bola.



Figura 4.15 - Resultado da aplicação do método para detetar e determinar o centro da bola; a) Fundo; b) Imagem Real; c) Imagem com o método aplicado e um quadrado desenhado à volta da posição calculada para centro da bola.

Observando, então, a imagem b) e c) da figura 4.15, pode concluir-se que o método funciona como esperado, sendo que o quadrado está desenhado exatamente no local onde se encontra a bola.

Para além disto, as coordenadas calculadas são ainda impressas na interface gráfica, como demonstra a figura seguinte:



Figura 4.16 - Interface gráfica que permite a visualização das coordenadas calculadas para o centro da bola.

4.4 - Cálculo da velocidade da bola entre 2 *Frames* (Imagens)

Uma vez determinada a posição da bola, um dos principais objetivos é a determinação da velocidade da bola entre duas determinadas imagens.

Assim sendo, torna-se necessário determinar a hora em que a imagem é capturada e processada, bem como as coordenadas da bola nessa mesma imagem.

Após isso, na próxima *frame* processada guarda-se os mesmos dados, podendo assim determinar a velocidade pela seguinte expressão.

$$v = \frac{\Delta pos}{\Delta t} \text{ píxeis/s} \quad (4.1)$$

Onde,

Δpos Corresponde à distância percorrida pela bola entre as duas *frames*

$$\Delta pos = \left| \sqrt{(posx_{atual} - posx_{anterior})^2 + (posy_{atual} - posy_{anterior})^2} \right| \text{ píxeis}$$

E Δt corresponde ao tempo decorrido entre as duas *frames*

$$\Delta t = (\text{Instante em que chega a 2ª Frame} - \text{Instante em que chega a 1ª Frame}) \text{ s}$$

Assim sendo, as figuras seguintes permitem demonstrar o cálculo da velocidade da bola entre duas *frames*.



Figura 4.17 - Duas Frames sobrepostas com esquema para o cálculo do Δpos .

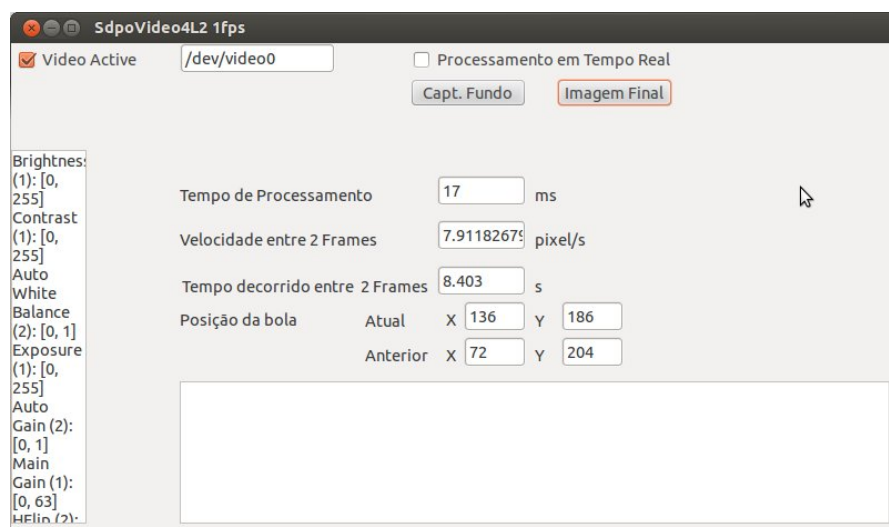


Figura 4.18 - Demonstração do resultado da velocidade, bem como o tempo decorrido entre as duas frames da figura anterior e as coordenadas das duas posições da bola.

Observando as duas figuras anteriores, pode comprovar-se a correta determinação da velocidade entre as duas *frames*, através do seguinte cálculo:

$$v = \frac{\Delta pos}{\Delta t} = \frac{\sqrt{|\Delta x|^2 + |\Delta y|^2}}{\Delta t} = \frac{\sqrt{|(posx_{atual} - posx_{anterior})|^2 + |(posy_{atual} - posy_{anterior})|^2}}{\Delta t} =$$

$$= \frac{\sqrt{|(136 - 72)|^2 + |(186 - 204)|^2}}{8.403} = 7.91183 \text{ pixels/s}$$

Assim, conclui-se que o cálculo da velocidade efetuado pelo programa elaborado está de acordo com cálculo teórico desta.

4.5 - Processamento de Imagem em Tempo Real

Chegando ao correto processamento de imagem e detecção da posição da bola em várias imagens, foi necessário desenvolver um programa capaz de fazer todo este processamento em tempo real.

Este programa utiliza os mesmos métodos já explicados neste documento, tendo apenas como diferença o processamento da imagem e detecção da bola para todos os *frames* capturados pela câmara, em vez de processar apenas as imagens pretendidas pelo utilizador.

Para isso, após ser capturado o fundo (cenário em que a bola se move), utiliza-se agora o visto "Processamento em Tempo Real" presente na interface gráfica e obtém-se o seguinte resultado:

Posx=14	Posy=179	Velocidade=424.00937562427pixel/s	Tempo decorrido=0.036s
Posx=21	Posy=190	Velocidade=360.656125403191pixel/s	Tempo decorrido=0.031s
Posx=55	Posy=183	Velocidade=307.318148576429pixel/s	Tempo decorrido=0.03s
Posx=65	Posy=182	Velocidade=164.484500438476pixel/s	Tempo decorrido=0.031s
Posx=74	Posy=181	Velocidade=105.409255338945pixel/s	Tempo decorrido=0.03s
Posx=96	Posy=180	Velocidade=95.8265957626781pixel/s	Tempo decorrido=0.033s
Posx=106	Posy=179	Velocidade=93.0081664755405pixel/s	Tempo decorrido=0.034s

Figura 4.19 - Histórico da posição, velocidade e tempo decorrido entre duas *frames*.

Este histórico é exibido no campo inferior em branco da interface gráfica descrita em 3.2, podendo assim ser visualizada a posição real da bola bem como a sua velocidade em cada instante.

Capítulo 5

Sistema de visão estereoscópica

De modo a ser desenvolvido um sistema estereoscópico capaz de determinar a posição 3D de um determinado objeto numa imagem, torna-se necessário estudar a relação matemática que relaciona as coordenadas de um ponto 3D e a sua projeção na imagem do plano.

A maior parte dos algoritmos desenvolvidos para visão computacional parte do princípio que o modelo da câmara segue o modelo *pinhole* câmara.

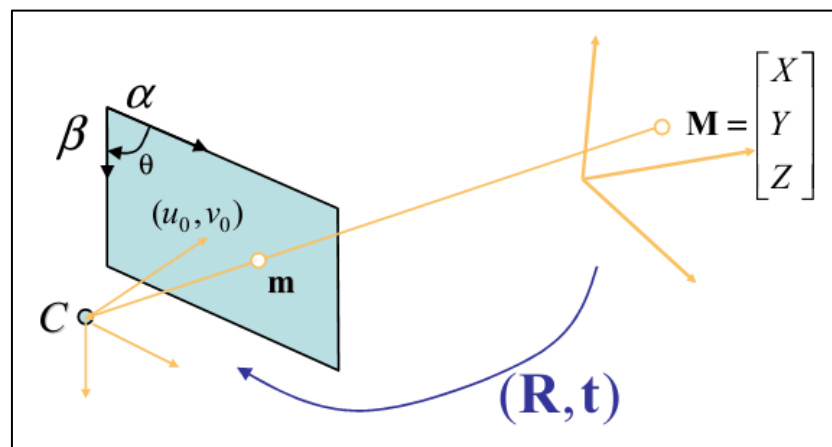


Figura 5.1 - Modelo *pinhole* câmara (cf. [7]).

Observando a figura anterior, chega-se à conclusão que o ponto m é um ponto 2D da imagem capturada pela câmara C e as suas coordenadas são $m = [u \ v]^T$. Já o ponto M refere-se a um ponto 3D no Mundo e as suas coordenadas são $M = [X \ Y \ Z]^T$.

A imagem de um ponto 3D M , denotado por m , é formada por um raio óptico vindo de M que passa no centro óptico da câmara C intersecando, assim, o plano da imagem (plano azul da figura 5.1). Os três pontos M , m e C são colineares.

Na figura 5.1, para ilustração, o plano da imagem está posicionado entre o ponto no mundo M e o centro óptico da câmara C .

A relação entre o ponto 3D M e a sua projeção na imagem m é dada por:

$$s\tilde{m} = A[R \quad t]\tilde{M} \equiv P\tilde{M} \quad (5.1)$$

$$\text{com } A = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{e } P = A[R \quad t]$$

Onde s é um fator de escala arbitrário, (R, t) são os parâmetros extrínsecos da câmara (a rotação e a translação relacionam as coordenadas do mundo com as coordenadas da câmara) e A é matriz que contém os parâmetros intrínsecos da câmara, em que (u_0, v_0) são as coordenadas do ponto principal (normalmente no centro da imagem), α e β são fatores de escala na imagem e γ é um parâmetro que descreve a inclinação dos eixos das duas imagens.

P é uma matriz 3x4 que representa a projeção da câmara contendo os parâmetros extrínsecos e intrínsecos desta.

Este modelo encontra-se explicado ao pormenor em [7], tendo esse documento ilustradas também várias formas para calibração das câmaras.

5.1 - Calibração das câmaras

Num sistema estereoscópico, um dos principais requisitos para o funcionamento do mesmo, prende-se com o facto de as duas câmaras que compõem este sistema estarem totalmente calibradas, pois só assim se torna possível obter as coordenadas de um objeto na imagem que correspondam exatamente às coordenadas desse mesmo objeto no Mundo.

Assim sendo, estudou-se várias formas de proceder à calibração das câmaras, chegando à conclusão que uma das formas mais utilizadas consiste em relacionar a posição 3D de vários pontos conhecidos do cenário real com a posição 2D destes nas imagens capturadas.

Para isso, utilizou-se um algoritmo estudado e desenvolvido por Zhang em [7]. Este algoritmo, consiste em colocar uma matriz de pontos em várias posições no cenário pretendido e, após isso, fazer corresponder as coordenadas 3D dos vários pontos da matriz nas suas coordenadas 2D das imagens capturadas.

A matriz de pontos utilizada nesse documento está demonstrada na figura seguinte.

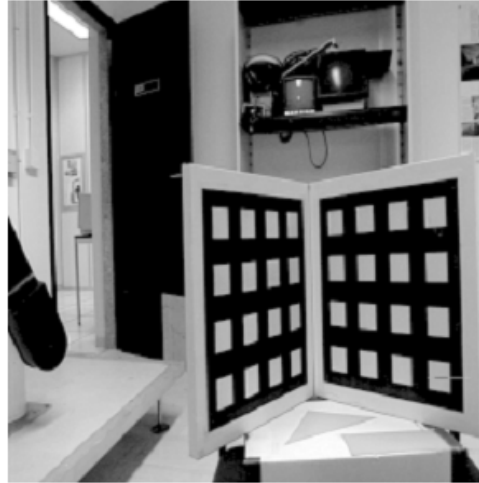


Figura 5.2 - Matriz de calibração utilizada em [7].

Sabendo todas as medidas associadas à matriz de calibração demonstrada na figura 5.2, torna-se então fácil de fazer corresponder as coordenadas do mundo real às coordenadas na imagem.

Posto isto, utilizando a equação da projeção (5.1) torna-se possível estimar os parâmetros da câmara. Assim sendo, é estimada numa primeira fase a matriz de projeção P através da equação:

$$\underbrace{\begin{bmatrix} X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & u_i X_i & u_i Y_i & u_i Z_i & u_i \\ 0 & 0 & 0 & 0 & X_i & Y_i & Z_i & 0 & v_i X_i & v_i Y_i & v_i Z_i & v_i \end{bmatrix}}_{G_i} p = 0$$

Onde $p = [p_{11}, p_{12}, \dots, p_{34}]^T$ e $0 = [0, 0]^T$.

Para n pontos no mundo, podem-se combinar todas as equações da seguinte forma:

$$Gp = 0, \text{ com } G = [G_1^T, \dots, G_n^T]^T$$

Em que a matriz G tem a dimensão $2n \times 12$.

A matriz de projeção pode então ser encontrada através de:

$$\min_p \|Gp\|^2 \text{ sujeito a } \|p\| = 1$$

Uma vez encontrada a matriz de projeção P , podem então ser recuperados os parâmetros intrínsecos e extrínsecos da câmara. Denotando a primeira submatriz 3×3 de P como sendo B e a última coluna de P como sendo b , ou seja $P \equiv [B \ b]$, utilizando a equação $P = A[R \ t]$ tem-se:

$$B = AR \quad (5.2)$$

$$b = At \quad (5.3)$$

Por (5.2) têm-se:

$$K \equiv BB^T = AA^T = \begin{bmatrix} \alpha^2 + \gamma^2 + u_0^2 & u_0 v_0 + c\beta & u_0 \\ u_0 v_0 + c\alpha & \alpha_v^2 + v_0^2 & v_0 \\ u_0 & v_0 & 1 \end{bmatrix} \equiv \begin{bmatrix} k_u & k_c & u_0 \\ k_c & k_v & v_0 \\ u_0 & v_0 & 1 \end{bmatrix} \quad (5.4)$$

Devido a P ser definido por um fator de escala, o ultimo elemento de $K = BB^T$ é usualmente diferente de 1, logo tem de ser normalizado tal que, K_{33} (o ultimo elemento de K)=1. Após isso, obtém-se

$$u_0 = K_{13} \quad (5.5)$$

$$v_0 = K_{23} \quad (5.6)$$

$$\beta = \sqrt{k_v - v_0^2} \quad (5.7)$$

$$\gamma = \frac{k_c - u_0 v_0}{\beta} \quad (5.8)$$

$$\alpha = \sqrt{k_u - u_0^2 - \gamma^2} \quad (5.9)$$

Uma vez determinados os parâmetros intrínsecos da câmara, ou a equivalente matriz A , os parâmetros extrínsecos podem ser determinados a partir das equações (5.2) e (5.3) de modo que:

$$R = A^{-1}B \quad (5.10)$$

$$t = A^{-1}b \quad (5.11)$$

Um exemplo da aplicação deste mesmo algoritmo pode ser encontrado em [7].

Analogamente ao algoritmo retratado anteriormente, foi elaborada uma matriz de pontos ilustrada na figura seguinte:

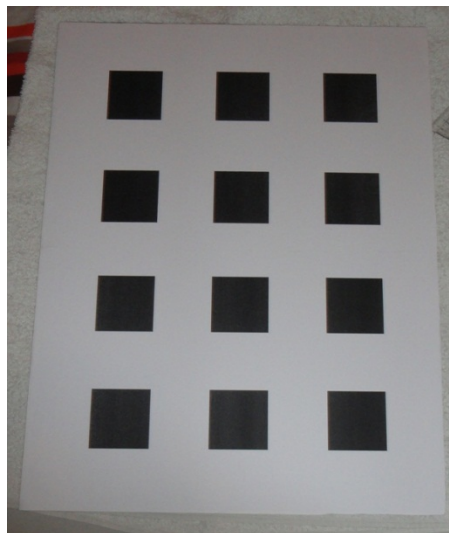


Figura 5.3 - Matriz de pontos para calibração das câmaras.

Esta matriz de pontos possui 12 quadrados de 6cm de lado com um espaço de 6cm entre cada um deles. Para calibração das câmaras, procedeu-se à montagem de um sistema estereoscópico, que consistiu num tripé com as duas câmaras no topo espaçadas de 34,5cm do centro, como demonstra a figura.

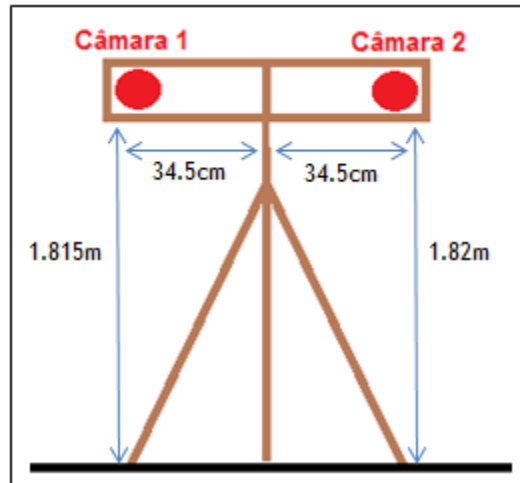


Figura 5.4 - Suporte das 2 câmaras que compõem o sistema de visão estereoscópica.

Este suporte foi colocado de forma a que as duas câmaras conseguissem capturar o cenário de jogo na sua totalidade. A posição escolhida, bem como todas as medidas associadas estão representadas na figura seguinte:

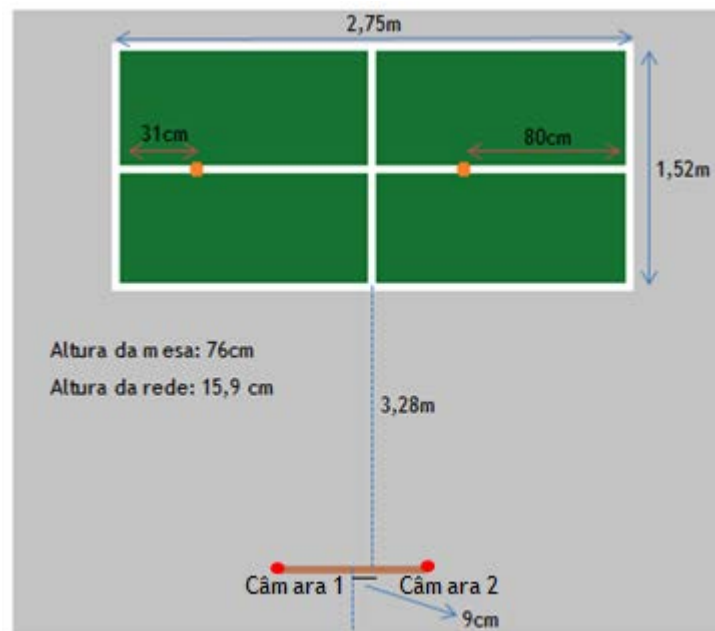


Figura 5.5 - Configuração do sistema de visão estereoscópica utilizado com as respetivas medidas.

Estando tudo preparado, procedeu-se à captura das imagens. Para tal, colocou-se a matriz de pontos em várias posições da mesa, bem como algumas bolas, nomeadamente uma em cada canto e mais duas nas posições retratadas a laranja na figura 5.5.

A figura seguinte visa retratar as posições em que foi colada a matriz de pontos assim como as bolas.



Figura 5.6 - Imagens capturadas com a matriz de pontos em várias posições e com as bolas nas posições conhecidas da mesa.

Tendo as imagens sido capturadas pelas duas câmaras, fez-se corresponder as coordenadas (u, v) dos vértices dos quadrados e do centro das bolas nas imagens na sua posição real (x, y, z) utilizando para isso as medidas efetuadas e demonstradas na figura 5.5.

Essa correspondência consistiu em clicar com o rato nos vértices de cada quadrado ou no centro da bola das imagens capturadas, extraindo assim as coordenadas do ponto (u, v) . Após isso, utilizando as medidas e escolhendo a origem do referencial no Mundo, determinou-se as coordenadas (x, y, z) de cada ponto.

O referencial utilizado está demonstrado na figura seguinte.

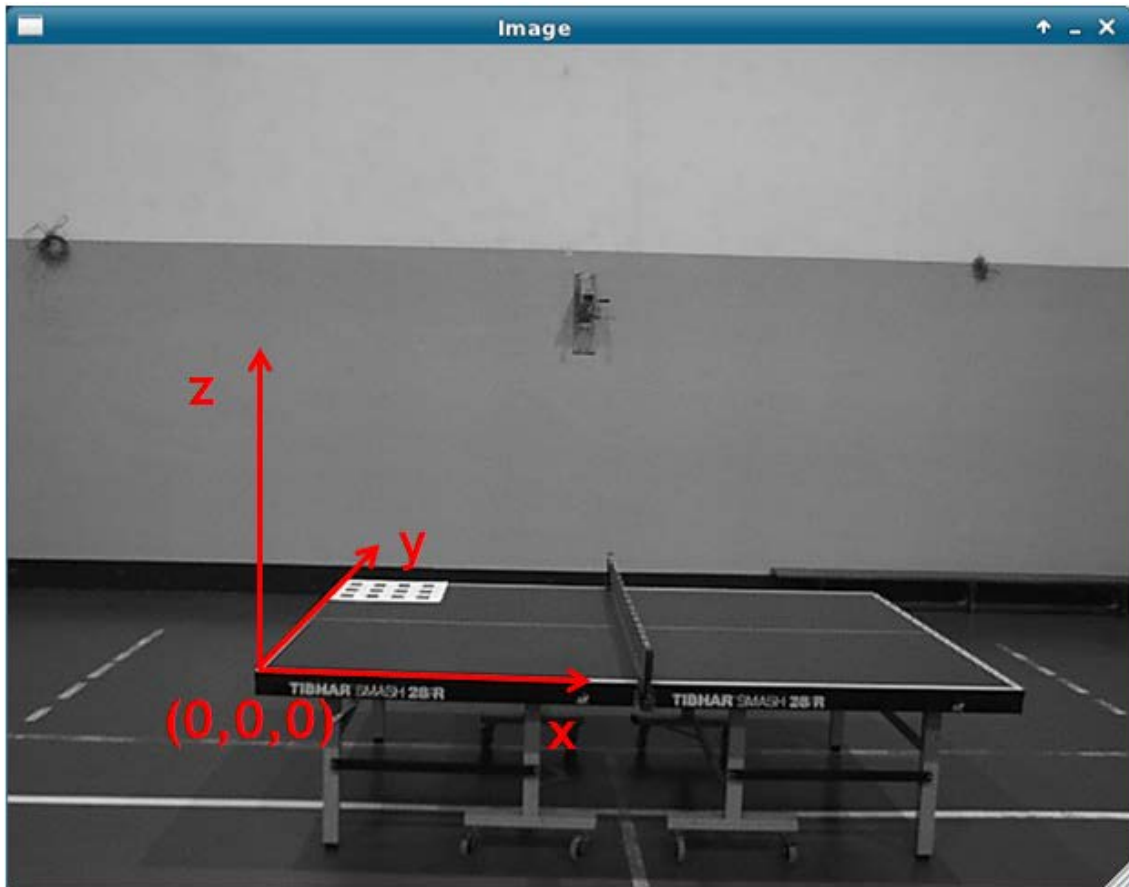


Figura 5.7 - Referencial utilizado para proceder à identificação das coordenadas dos pontos no cenário real.

Assim sendo, foram criados 2 ficheiros de texto, um para cada câmara, que possuem as coordenadas dos vários pontos no cenário real (x , y , z) em metros e as coordenadas desses mesmos pontos nas imagens (u , v) em píxéis, ficando o ficheiro de texto, para cada ponto, organizado da seguinte forma:

x	y	z	u	v
-----	-----	-----	-----	-----

Posto isto, e apesar de se ter feito corresponder as posições 3D com as posições 2D, não foi possível realizar a calibração das câmaras devido à falta de tempo. No entanto, utilizando o que foi retratado ao longo do ponto 5.1, seria possível numa próxima fase proceder à calibração destas.

5.2 - Reconstrução 3D

Depois de encontrados os pares de pontos nas duas imagens, é necessário efetuar a reconstrução 3D do ponto.

Para isso, existem vários caminhos a seguir que utilizam as matrizes de projeção de cada câmara. Uma das formas mais eficazes consiste na triangulação dos raios das câmaras estudada no documento [17]. Este método baseia-se no conceito de geometria epipolar.

Geometria epipolar é a geometria projetiva intrínseca que existe num sistema de estereoscopia. Esta geometria é independente da estrutura do cenário, dependendo apenas dos parâmetros intrínsecos das câmaras e da posição relativa destas.

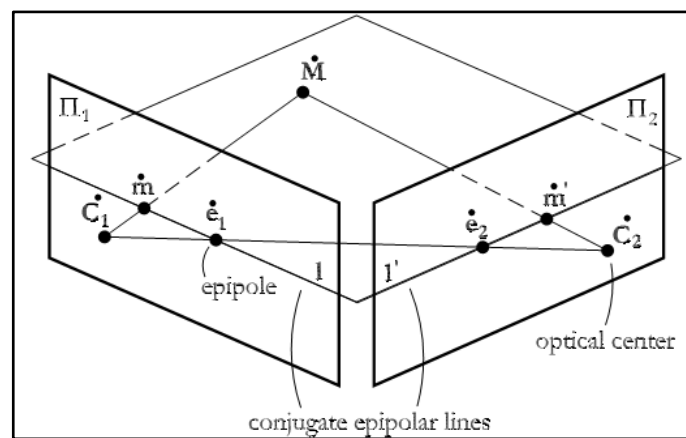


Figura 5.8 - Modelo de um sistema de visão estereoscópica [16].

Observando a figura anterior, nota-se que este modelo consiste na utilização de duas câmaras para observar um ponto \dot{M} no espaço. Assim sendo, tem-se duas imagens, uma por câmara.

Posto isto, dado um ponto \dot{M} no espaço e duas câmaras \dot{C}_1 e \dot{C}_2 , \dot{m} é a imagem do ponto \dot{M} formada pela câmara \dot{C}_1 no plano Π_1 e \dot{m}' é a imagem do mesmo ponto no plano Π_2 .

A geometria intrínseca que a figura 5.8 representa está encapsulada numa matriz, denominada matriz F . Esta matriz tem dimensão 3×3 e, se um ponto \dot{M} no espaço 3D é visto na primeira vista do sistema estereoscópico no plano da imagem \dot{m} e na segunda vista \dot{M} é visto como \dot{m}' , então estes dois pontos nos dois planos satisfazem a seguinte relação [17]:

$$\dot{m}'^T F \dot{m} = 0 \quad (5.12)$$

Assim sendo o algoritmo estudado em [17] para reconstrução 3D do ponto é exemplificado na seguinte figura.

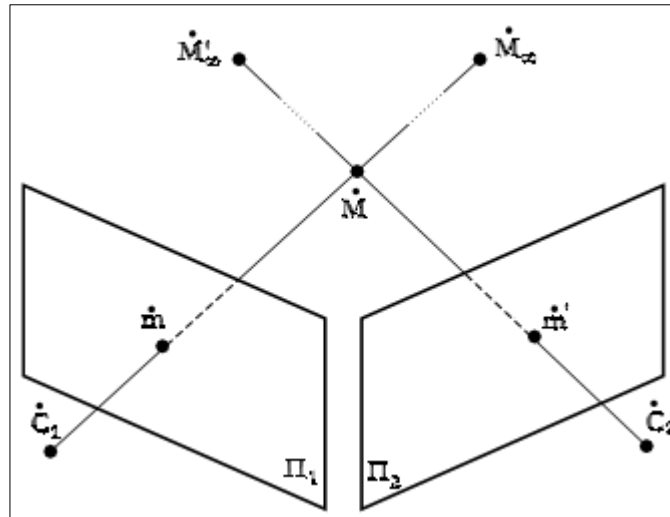


Figura 5.9 - Triangulação ideal dos raios das câmaras: M' é a interseção dos raios das duas câmaras (cf. [17]).

No entanto, quando as imagens estão afetadas de ruído as retas podem não se intercetar. Para contornar isso, em [17] está retratado um algoritmo que consiste em retirar o ponto médio da perpendicular dos dois raios. Assim sendo a figura seguinte retrata esse mesmo algoritmo.

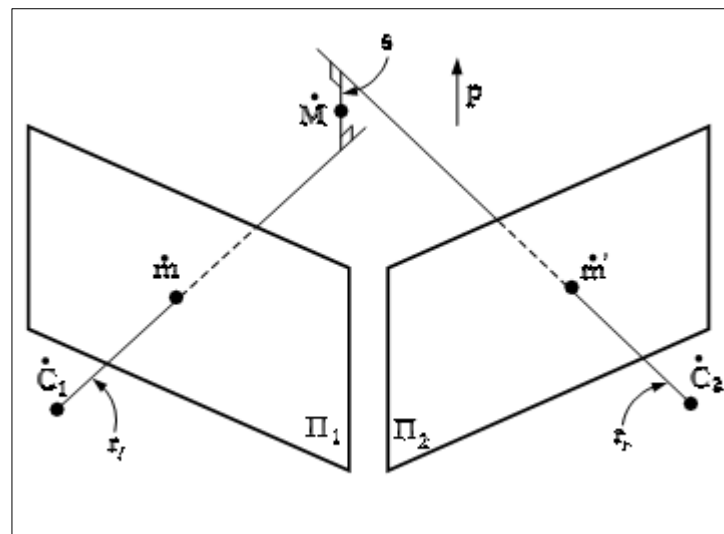


Figura 5.10 - Triangulação sem interseção das retas (cf. [17]).

Posto isto, ainda em [17], utilizando este método, Rimón Elias e Robert Laganière afirmam que a equação da linha de projeção da primeira vista pode ser representada por:

$$r_l = \alpha m \quad (5.13)$$

Onde α é um escalar. A linha de projeção da segunda vista (vista do lado direito) referido ao plano de imagem esquerdo (Π_1), pode ser expressa como:

$$r_r = \beta R^T m' + T \quad (5.14)$$

Em que B é um escalar, R a matriz de rotação e T a matriz de translação. Logo, se o vetor perpendicular a r_l e r_r é:

$$\mathbf{p} = \mathbf{m} \times \mathbf{R}^T \mathbf{m}' \quad (5.15)$$

E a linha paralela a \mathbf{p} que passa em \mathbf{am} é:

$$\mathbf{p}_1 = \mathbf{am} + c\mathbf{p} \quad (5.16)$$

Onde c é um escalar. Assim sendo, os pontos extremos do segmento de reta s podem ser obtidos, resolvendo:

$$\mathbf{am} + \mathbf{bR}^T \mathbf{m}' + c(\mathbf{m} \times \mathbf{R}^T \mathbf{m}') = \mathbf{T} \quad (5.17)$$

Consequentemente, \dot{M} é o ponto médio de s .

Uma vez estudado, este algoritmo acabou por não ser implementado. No entanto foi importante o estudo do mesmo, pois permitiu adquirir conhecimentos até agora desconhecidos.

Capítulo 6

Testes e Resultados Obtidos

Para que se pudesse testar o algoritmo desenvolvido ao longo do projeto, foram capturadas imagens com o movimento da bola em vários cenários e com vários tipos de iluminação. Só assim se conseguirá perceber o funcionamento do algoritmo em cenários distintos.

As figuras seguintes demonstram os vários cenários em que foi testado o algoritmo, bem como a deteção da bola em cada um deles.

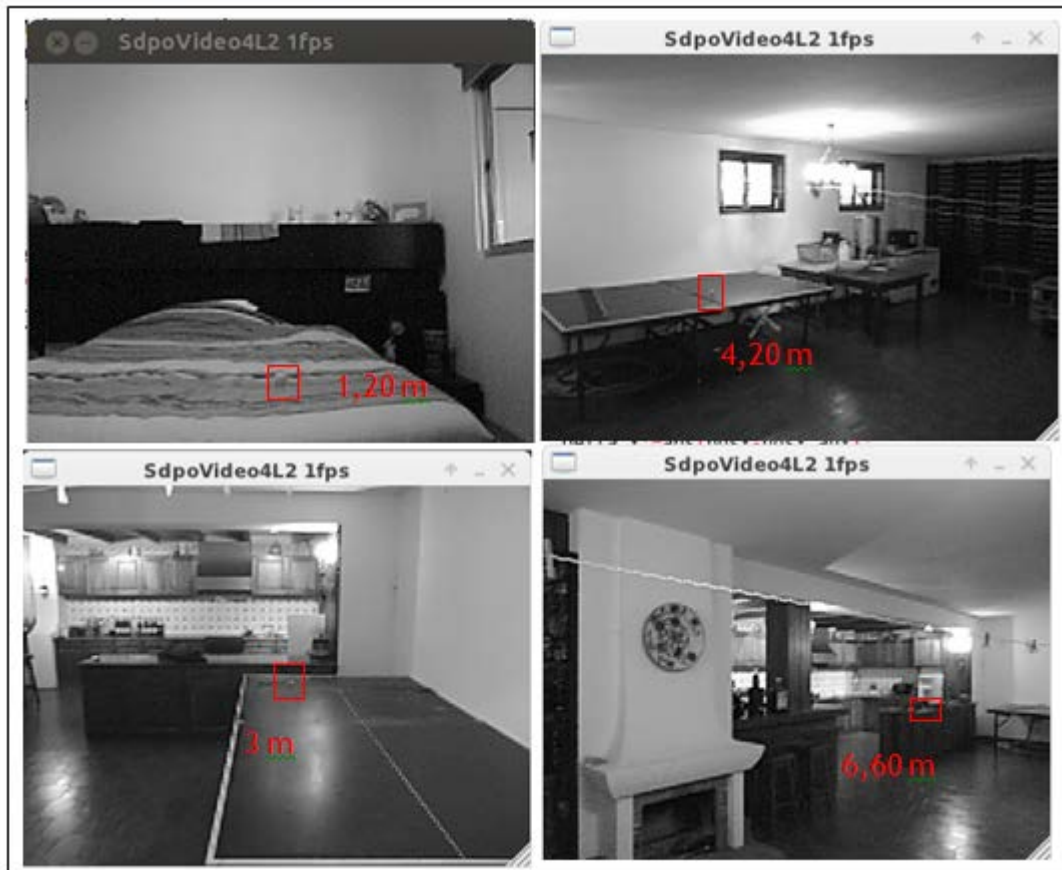


Figura 6.1 - Imagens capturadas em cenários e distâncias diferentes.

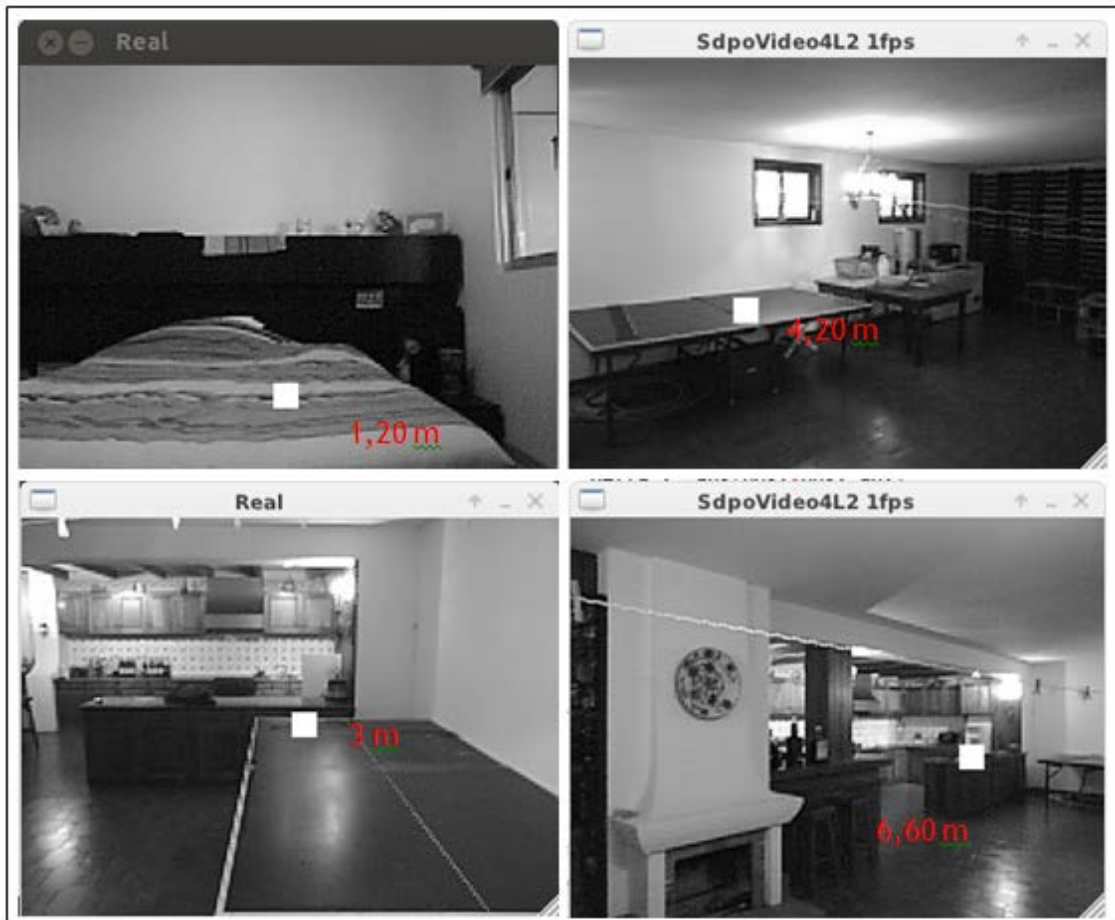


Figura 6.2 - Detecção da bola nos vários cenários.

Através das figuras anteriores, pode comprovar-se que o algoritmo desenvolvido funciona nos vários cenários com a câmara a diferentes distâncias da bola. No entanto, é de notar, que na imagem inferior direita a bola já se encontra muito distante da câmara (6,60m), o que faz com que seja quase impossível ver a bola na figura 4.1.

Posto isto comprova-se que o algoritmo se demonstra eficaz para distâncias até os 4,50m, uma vez que, quanto maior for a distância, mais erros poderão surgir.

6.1 Resultados obtidos

Os resultados obtidos durante o projeto estão expostos ao longo de todo o documento, utilizando para esse fim figuras que foram comprovando o correto funcionamento dos algoritmos desenvolvidos.

No entanto, para comprovar o correto funcionamento do resultado final obtido, realizou-se um vídeo que consiste na aquisição de imagem através de duas câmaras em que, para cada uma delas, é realizado o processamento em tempo real.

A figura seguinte, demonstra o *layout* do vídeo realizado.

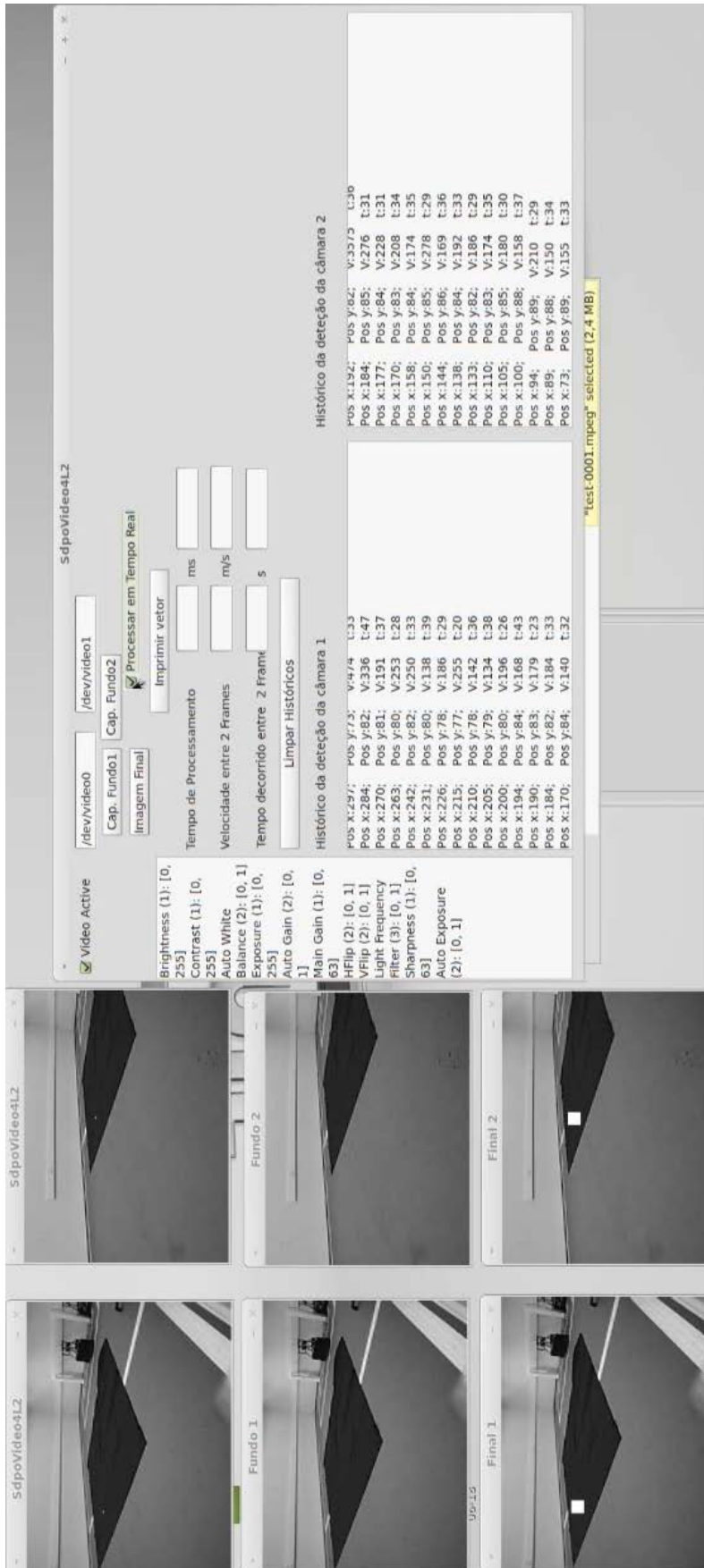


Figura 6.3 - Layout do vídeo realizado que ilustra o resultado final do sistema desenvolvido.

Capítulo 7

Conclusões

Ao longo do trabalho foram sendo cumpridos todos os objetivos delineados para o projeto.

O grande objetivo do trabalho foi alcançado: a trajetória do movimento da bola pode ser monitorizada através da determinação da posição e velocidade desta.

No entanto, esta monitorização foi apenas realizada em 2D, tendo sido estudados vários algoritmos que permitiriam a monitorização da trajetória da bola em 3D.

Verificou-se que, através de algoritmos relacionados com Sistema Baseados em Visão, se torna possível a identificação de objetos numa imagem/vídeo, qualquer que seja a sua forma.

Verificou-se também que um dos fatores que permite o correto funcionamento dos vários algoritmos é a iluminação, pois, se esta não for suficientemente boa, as imagens ficam afetadas por muito ruído, o que faz com que seja praticamente impossível realizar algum tipo de processamento sobre essas imagens.

Através do estudo e implementação de alguns algoritmos foi alcançada a determinação da posição e velocidade da bola de uma forma eficaz. Esta foi detetada em cenários que, observando as imagens, não se consegue visualizar a bola de uma forma imediata. A posição e velocidade foram determinadas em função dos píxeis de uma imagem com resolução 320 x 240.

A nível de visão estereoscópica, este projeto permitiu adquirir conceitos até agora desconhecidos, que permitem a determinação da posição 3D de um determinada objeto, através de duas câmaras.

No entanto, a este nível fica o desejo de poder aprofundar mais o assunto, devido à sua aplicação em sistemas robotizados, nomeadamente em linhas de montagem/produção.

O *software* utilizado (Lazarus) demonstrou-se uma ferramenta muito eficaz para este tipo de aplicações. Devido a ter utilizado este *software* apenas a nível de sistemas de informação (base de dados, etc.), desconhecia esta sua aplicação.

Em relação ao *hardware*, nomeadamente as câmaras testadas, conclui-se que, neste tipo de sistemas, é muito importante o estudo e escolha de câmaras que possam proporcionar resultados eficazes.

Em relação à aplicação no centro de estágio de alto rendimento de Ténis de Mesa em Vila Nova de Gaia do sistema desenvolvido, apenas foi possível comprovar a correta monitorização da trajetória da bola em 2D aquando da aquisição das imagens com a matriz de calibração colocada em vários pontos da mesa.

Por fim, pode concluir-se que este trabalho foi muito vantajoso, uma vez que fez com que tomasse conhecimento de conceitos que serão certamente importantes na minha vida profissional e me permitiu também comprovar as vantagens que a tecnologia pode oferecer ao desporto.

7.1 - Trabalho futuro

Como trabalho futuro sugiro várias áreas que poderão ser estudadas e implementadas, das quais:

a) Visão estereoscópica:

- Implementar o algoritmo para calibração das câmaras utilizando para isso os dois ficheiros e texto que contêm a correspondência entre a posição 3D no mundo e a posição 2D na imagem de diferentes pontos;
- Realizar a reconstrução 3D por forma identificar a posição da bola, não só em função dos píxeis da imagem, mas também no cenário real.

b) Determinação dos pontos de contacto da bola com a mesa:

- Uma vez implementado o sistema estereoscópico, propunha realizar um algoritmo capaz de determinar os pontos de contacto da bola com a mesa, o que é de fácil resolução se a reconstrução 3D estiver bem desenvolvida.

c) Aquisição dos movimentos do jogador com uma taxa de elevada de fps:

- Estudar a melhor forma para adquirir imagens dos movimentos do jogador, por forma a, juntamente com a trajetória monitorizada da bola, providenciar ao treinador informações que o permitam elaborar um relatório, ou mesmo efetuar um treino personalizado ao jogador.

Referências

- [1] LEI, S., L. JINGTAI, ET AL. (2009). BALL'S FLIGHT TRAJECTORY PREDICTION FOR TABLE-TENNIS GAME BY HUMANOID ROBOT. ROBOTICS AND BIOMIMETICS (ROBIO), 2009 IEEE INTERNATIONAL CONFERENCE ON.
- [2] PING, Y., X. DE, ET AL. (2011). A VISION SYSTEM WITH MULTIPLE CAMERAS DESIGNED FOR HUMANOID ROBOTS TO PLAY TABLE TENNIS. AUTOMATION SCIENCE AND ENGINEERING (CASE), 2011 IEEE CONFERENCE ON.
- [3] JINZI MAO (2006). "TRACKING A TENNIS BALL USING IMAGE PROCESSING TECHNIQUES".
- [4] <http://www.lazarus.freepascal.org/>
- [5] <http://wiki.lazarus.freepascal.org/5dpo>
- [6] <http://www.ptgreystore.com/products/162-flea3-13mp-mono-usb-30-sony-imx035.aspx>
- [7] ZHENGYOU ZHANG. "CAMERA CALIBRATION"
- [8] ZHENGTAO, Z., X. DE, ET AL. (2010). "VISUAL MEASUREMENT AND PREDICTION OF BALL TRAJECTORY FOR TABLE TENNIS ROBOT." INSTRUMENTATION AND MEASUREMENT, IEEE TRANSACTIONS ON 59(12): 3195-3205.
- [9] WONG, P. K. C. (2008). DEVELOPING AN INTELLIGENT TABLE TENNIS UMPIRING SYSTEM: IDENTIFYING THE BALL FROM THE SCENE. MODELING & SIMULATION, 2008. AICMS 08. SECOND ASIA INTERNATIONAL CONFERENCE ON.
- [10] ANDERSSON, R. L. (1990). A LOW-LATENCY 60 HZ STEREO VISION SYSTEM FOR REAL-TIME VISUAL CONTROL. INTELLIGENT CONTROL, 1990. PROCEEDINGS., 5TH IEEE INTERNATIONAL SYMPOSIUM ON.
- [11] REN, J., J. ORWELL, ET AL. (2009). "TRACKING THE SOCCER BALL USING MULTIPLE FIXED CAMERAS." COMPUTER VISION AND IMAGE UNDERSTANDING 113(5): 633-642.
- [12] CHEN, H.-T., M.-C. TIEN, ET AL. (2009). "PHYSICS-BASED BALL TRACKING AND 3D TRAJECTORY RECONSTRUCTION WITH APPLICATIONS TO SHOOTING LOCATION ESTIMATION IN BASKETBALL VIDEO." JOURNAL OF VISUAL COMMUNICATION AND IMAGE REPRESENTATION 20(3): 204-216.

- [13] KIM, S.-D., J.-H. LEE, ET AL. (1988). "A NEW CHAIN-CODING ALGORITHM FOR BINARY IMAGES USING RUN-LENGTH CODES." COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING 41(1): 114-128.
- [14] TSAI, R. (1987). "A VERSATILE CAMERA CALIBRATION TECHNIQUE FOR HIGH-ACCURACY 3D MACHINE VISION METROLOGY USING OFF-THE-SHELF TV CAMERAS AND LENSES." ROBOTICS AND AUTOMATION, IEEE JOURNAL OF 3(4): 323-344.
- [15] HARTLEY, R. I. (1997). "IN DEFENSE OF THE EIGHT-POINT ALGORITHM." PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON 19(6): 580-593.
- [16] HARTLEY, R. I. AND P. STURM (1997). "TRIANGULATION." COMPUTER VISION AND IMAGE UNDERSTANDING 68(2): 146-157.
- [17] RIMON ELIAS AND ROBERT LAGANIÈRE - PROJECTIVE GEOMETRY FOR THREE-DIMENSIONAL COMPUTER VISION.
- [18] OHNO, Y., J. MIURA, ET AL. (2000). TRACKING PLAYERS AND ESTIMATION OF THE 3D POSITION OF A BALL IN SOCCER GAMES. PATTERN RECOGNITION, 2000. PROCEEDINGS. 15TH INTERNATIONAL CONFERENCE ON.
- [19] RUIZ, N. A. (2010). TRACKING OF A BASKETBALL USING MULTIPLE CAMERAS.
- [20] EKINCI, B. D. AND M. GOKMEN (2008). A BALL TRACKING SYSTEM FOR OFFLINE TENNIS VIDEOS. PROCEEDINGS OF THE 1ST WSEAS INTERNATIONAL CONFERENCE ON VISUALIZATION, IMAGING AND SIMULATION. BUCHAREST, ROMANIA, WORLD SCIENTIFIC AND ENGINEERING ACADEMY AND SOCIETY (WSEAS): 45-48.