



Universidade do Minho
Escola de Engenharia

Aníbal José Macedo Pinheiro

Sistema Robótico para Detecção e Movimento
de Objetos com Visão Estéreo Modificada



Universidade do Minho
Escola de Engenharia

Aníbal José Macedo Pinheiro

Sistema Robótico para Detecção e Movimento
de Objetos com Visão Estéreo Modificada

Dissertação de Mestrado
Ciclo de Estudos Integrados Conducentes ao Grau de
Mestre em Engenharia Eletrónica Industrial e Computadores

Trabalho efetuado sob a orientação do
Professor Doutor Agostinho Gil Teixeira Lopes

DECLARAÇÃO

Nome: Aníbal José Macedo Pinheiro

Endereço eletrónico: a52704@alunos.uminho.pt

Telefone: 917747814 / 252852894

Número do Bilhete de Identidade: 13575545

Título Dissertação: Sistema Robótico para Detecção e Movimento de Objetos com Visão Estéreo Modificada

Orientador: Professor Doutor Agostinho Gil Teixeira Lopes

Ano de conclusão: 2014

Designação do Mestrado: Mestrado Integrado em Engenharia Eletrónica Industrial e Computadores – Automação, Controlo e Robótica

Nos exemplares das teses de doutoramento ou de mestrado ou de outros trabalhos entregues para prestação de provas públicas nas universidades ou outros estabelecimentos de ensino, e dos quais é obrigatoriamente enviado um exemplar para depósito legal na Biblioteca Nacional e, pelo menos outro para a biblioteca da universidade respetiva, deve constar uma das seguintes declarações:

DE ACORDO COM A LEGISLAÇÃO EM VIGOR, NÃO É PERMITIDA A REPRODUÇÃO DE QUALQUER PARTE DESTA DISSERTAÇÃO;

Universidade do Minho, ___/___/_____

Assinatura: _____

*“Todo o grande progresso da ciência resultou
duma nova audácia da imaginação”*

John Dewey

Agradecimentos

Durante a realização desta dissertação houve vários fatores que contribuíram para que conseguisse atingir os objetivos a que me propus, entre as quais, e a mais importante, a ajuda do Professor Agostinho Gil Teixeira Lopes, que propôs este projeto para tema da minha dissertação e ter aceite orientar-me ao longo deste projeto. Para ele, o meu sincero agradecimento por sempre ter estado disponível para todo e qualquer tipo de esclarecimento, assim como a sua simpatia e pela sua compreensão por alguns acontecimentos ocorridos fora do âmbito deste projeto.

Também gostava de deixar uma palavra de apreço para com os meus amigos de trabalho, Ricardo, José e Daniel, pelo apoio prestado e pela boa disposição que sempre demonstraram, tornando, assim, o ambiente de trabalho fantástico. Não podia também deixar de referenciar o apoio dos amigos do dia-a-dia, que estiveram sempre presentes e me deram a confiança de que sempre necessitei.

Aos técnicos e funcionários das oficinas do Departamento de Engenharia Eletrónica Industrial pela disponibilidade prestada e pelo apoio que sempre demonstraram quando lhes solicitei ajuda.

Por fim, mas não menos importante, quero agradecer aos meus pais e irmãos pelo esforço realizado para que este sonho fosse possível de ser concretizado e pelo apoio dado nos momentos mais difíceis. E uma palavra também para a minha namorada, Sofia, pois sempre esteve ao meu lado a acompanhar-me nesta longa caminhada. A todos, o meu sincero agradecimento.

Resumo

A utilização de robôs nas mais diversas áreas tem sido cada vez mais usual. Dessa forma, surge a necessidade de aperfeiçoar e inovar as técnicas já existentes para tornar os sistemas robóticos mais eficazes.

Um importante fator na manipulação de objetos por sistemas robotizados prende-se com a deteção e perceção da localização espacial desses objetos. A utilização de técnicas de múltiplas câmaras (sistemas estéreo) tem sido muito explorada com as inerentes exigências computacionais que elas necessitam.

O objetivo desta dissertação é de desenvolver um sistema de visão por computador, através da utilização de duas câmaras em estéreo modificado, onde ao contrário de duas câmaras fixas, serão utilizadas duas câmaras que rodam segundo um eixo vertical, no sentido de melhorar a perceção da distância e profundidade ao objeto, numa tentativa de diminuir a exigência computacional. A ideia base é a simular a visão humana, onde os olhos convergem ou divergem, por forma a centrarem a imagem obtida por cada olho num determinado objeto, e utilizando o ângulo de rotação de cada olho para efeitos do cálculo da distância ao objeto.

Um protótipo de hardware foi desenvolvido por forma a poder acomodar um conjunto de algoritmos, desenvolvidos na biblioteca livre OpenCV, e que permitiu o desenvolvimento de um novo método de pesquisa inovador que acelera o processamento.

Palavras-chave: *Reconhecimento de objetos, sistemas estéreo, visão humana, radiais de pesquisa, localização de objetos.*

Abstract

The use of robots in several areas has been increasingly common. Thus arises the need to improve and innovate the existing techniques to make them more effective robotic systems.

An important factor in the manipulation of objects by robotic systems relates to the detection and perception of spatial location of these objects. The use of techniques from multiple cameras (stereo systems) has been widely exploited with the inherent computational requirements they need.

The objective of this dissertation is to develop a computer vision system, by using two cameras on modified stereo, where instead of using two fixed cameras, this system uses two cameras that rotate along a vertical axis to improve the perception of depth and distance to the object in an attempt to decrease the computational requirement. The basic idea is to simulate human vision where the eyes converge or diverge in order to focus the image obtained from each eye into a particular object, and using the rotation angle of each eye for the calculation of the distance to the object.

A prototype hardware was designed so that it can accommodate a set of algorithms developed using the free OpenCV library, which allowed the development of a new innovative search method that accelerates the processing.

Keywords: *Object recognition, stereo systems, human vision, radial search, object localization.*

Índice

Agradecimentos	v
Resumo	vii
Abstract	ix
Índice de Figuras	xiv
Lista de Acrónimos	xvii
Capítulo I	1
1. Introdução	1
1.1 Motivação	2
1.2 Objetivos	3
1.3 Estrutura do documento	4
Capítulo II	7
2. Revisão Bibliográfica	7
2.1 Visual Tracking Control to Fast Moving Target for Stereo Vision Robot	7
2.2 Making a step to stereo vision	9
2.3 Visão Estéreo	10
2.3.1 Calibração de uma câmara	12
2.3.1.1 Modelo da câmara	12
2.3.3 Retificação estéreo	14
2.3.2 Mapa de disparidades	14
2.4 Algoritmos de deteção de objetos	15
2.4.1 <i>SIFT (Scale-Invariant Feature Transform)</i> [11]	15
2.4.1.1 Deteção de pontos-chave	16
2.4.1.2 Descrição de pontos-chave	16
2.4.2 SURF(Speeded-Up Robust Features) [13]	17

2.4.2.1 Detecção de pontos de interesse (<i>Keypoints</i>)	17
2.4.2.2 Descrição dos pontos de interesse	19
2.5 Equipamentos comerciais	20
Capítulo III.....	21
3. Plataforma desenvolvida.....	21
3.1 Componentes utilizados.....	21
3.1.1 Servomotor <i>DF15 Metal Gear</i>	21
3.1.2 Alimentação servomotores	22
3.1.2.1 Bateria <i>KingMax</i>	22
3.1.2.2 <i>Traco Power</i> Conversor DC-DC	23
3.1.3 <i>Arduino Mega 2560</i>	23
3.1.3.1 Alimentação microcontrolador	24
3.1.3.2 Pinos de entrada e de saída	24
3.1.3.3 Comunicação	24
3.1.4 Placa <i>SD21 Servo Controller</i>	24
3.2 Circuito acionamento dos servomotores	25
3.3 Plataforma final	26
Capítulo IV.....	27
4. Controlo dos servomotores	27
4.1.1 Processamento do controlador <i>SD21</i>	27
4.1.2 Posição dos servomotores.....	28
4.1.3 Velocidade dos servomotores	28
4.2 Comunicação <i>I2C</i> entre <i>Arduino</i> e <i>Devantech SD21</i>	29
4.2.1 Biblioteca <i>Wire</i>	29
4.3 Controlo de posição dos servomotores	30

4.4 Algoritmo de controlo de posição dos servomotores	31
Capítulo V	33
5. Método desenvolvido e resultados.....	33
5.1 <i>Software</i> adotado	33
5.1.1 Sistema Operativo	33
5.1.2 Plataforma de desenvolvimento	33
5.1.3 Biblioteca.....	34
5.2 <i>Webcams</i>	35
5.2.1 Posicionamento das <i>webcams</i>	36
5.3 Processamento de imagem	37
5.3.1 Reconhecimento de um objeto	37
5.3.2 Pesquisa por radiais	39
5.3.2.1 Histogramas.....	40
5.3.2.2 Tempo de processamento	41
5.4 Localização do objeto na imagem	42
5.5 Algoritmo de processamento de imagem	46
Capítulo VI.....	49
6. Conclusão e trabalho futuro	49
Referências	51

Índice de Figuras

Figura 1- Duas imagens obtidas em pontos diferentes [2]	1
Figura 2 - Robô <i>AROS</i> [3]	3
Figura 3 - Robô <i>MARy</i>	3
Figura 4 - Esquemático Estrutura (Vista de Frente) [4]	7
Figura 5 - Plano de imagem e sistema de coordenadas [4]	8
Figura 6 – Caso1: comportamento esperado (lado esquerdo) e comportamento verificado (lado direito) [4]	8
Figura 7 – Caso2: comportamento esperado (lado esquerdo) e comportamento verificado (lado direito) [4]	9
Figura 8 - Vista de frente da plataforma [5]	9
Figura 9 - Filtragem da cor azul [5].....	10
Figura 10 - Percepção do ser humano do espaço [6].....	11
Figura 11 - Esquema do posicionamento de duas câmaras <i>stereo</i> [7].....	11
Figura 12 - Geometria epipolar [8].....	12
Figura 13 - Imagem distorcida (esquerda) e imagem calibrada (direita) [2].....	12
Figura 14 - Modelo usual da câmara pontual [9]	13
Figura 15 - Modelo aproximado da câmara pontual [9].....	13
Figura 16 - Retificação dos planos de imagem [10].....	14
Figura 17 - Gráfico com relação entre distância e a disparidade [7].....	15
Figura 18 - Imagem de uma impressão digital original e após sucessivas desfocagens [12]	16
Figura 19 - Histograma com orientações dos pontos-chave [11]	17
Figura 20 - Divisão em retângulos para tornar o processo mais rápido [13].....	18
Figura 21 - Espaço de escalas em forma de pirâmide invertida [13]	18
Figura 22 - Filtros Haar Wavelets em x (esquerda) e em y (direita) [13]	19
Figura 23 - Orientação dos pontos de interesse [13]	19
Figura 24 - Quadrado de respostas de <i>Wavelets</i> [13]	19
Figura 25 - Câmara stereo <i>MobileRanger</i> [15].....	20
Figura 26 - Câmara stereo <i>Bumblebee2</i> [16].....	20
Figura 27 - Servomotor <i>DF15 Metal Gear</i> [17].....	22
Figura 28 - Bateria <i>KingMax</i> 1000mAh [17]	23
Figura 29 - Conversor DC-DC <i>Traco Power</i> [17].....	23

Figura 30 - Microcontrolador <i>Arduino Mega 2560</i> [18]	24
Figura 31 - Placa de controlo <i>SD21</i> [19].....	25
Figura 32 - Circuito acionamento dos servomotores.....	25
Figura 33 - Sistema de visão estéreo desenvolvido	26
Figura 34- Relação entre posição em μ s e posição em Graus	28
Figura 35- Esquema dos servomotores <i>Pan</i> e <i>Tilt</i> [20]	30
Figura 36 - Exemplo de localização do alvo	31
Figura 37 - Algoritmo de controlo de posição dos servomotores.....	32
Figura 38 - Logotipo <i>Ubuntu</i> [21]	33
Figura 39 - Logotipo <i>Qt Creator</i> [22]	33
Figura 40 - <i>IDE</i> do <i>Arduino</i>	34
Figura 41 - Logotipo <i>OpenCV</i> [23]	35
Figura 42 - Câmara <i>Phillips SPC900NC</i> [24]	35
Figura 43 - Suporte metálico das webcams	36
Figura 44 - Imagens desalinhasadas	36
Figura 45 - Imagens alinhadas.....	37
Figura 46 - Tamanho das frames capturadas	37
Figura 47 - Sistema de cores <i>HSV</i> [27].....	38
Figura 48 - Objeto redondo utilizado	38
Figura 49 - Radiais de pesquisa.....	39
Figura 50 - Exemplo do desenho de pontos brancos no objeto através dos radiais	40
Figura 51 - Uso de histogramas com informação dos radiais.....	41
Figura 52 - Output com número de píxeis analisados	42
Figura 53 - Relações trigonométricas vista na imagem processada	42
Figura 54 - Relações trigonométricas	43
Figura 55 - Etapas da focalização do objeto.....	44
Figura 56 - Histograma obtido com alvo no centro.....	44
Figura 57 - Distância do objeto às câmaras influencia o ângulo de rotação	45
Figura 58 - Algoritmo de processamento de imagem.....	47

Índice de Tabelas

Tabela 1 - Registos utilizados da placa SD21	28
Tabela 2 - Caraterísticas <i>Webcams</i>	35

Lista de Acrónimos

3D – 3 dimensões

DC – Corrente Contínua

DoG – Diferença de Gaussianas

SIFT - Scale-Invariant Feature Transform

SURF - Speeded-Up Robust Features

FPS – Frames per second

FPGA – Field Programmable Gate Array

SDK- Software Development Kit

DEI – Departamento de Eletrónica Industrial

MG – Metal Gear

DC-DC – Corrente Contínua-Corrente Contínua

UART – Universal Asynchronous Receiver/Transmitter

USB - Universal Serial Bus

I2C - Inter-Integrated Circuit

LAR - Laboratório de Automação e Robótica

MP – Mepa Píxel

CCD – Charge-Coupled Device

RGB – Red, Green, Blue

HSV – Hue, Saturation, Value

Capítulo I

1. Introdução

A visão humana e animal é conhecida pela ciência, em geral, como sendo binocular, isto é, constituída por dois recetores visuais em posições diferentes, que fornecem ao cérebro imagens diferentes [1]. O controlo dos sistemas binoculares existentes não traduz com a exatidão exigida aquilo que os seres humanos e animais conseguem realizar na vida real, isto porque estes sistemas já existentes baseiam-se em câmaras paralelas e fixas, não contemplando, por isso, a possibilidade de convergência/divergência dos olhos. Para implementar a solução que se pretende desenvolver, é necessário que as câmaras estejam assentes num conjunto de eixos que permitam que ambas se movimentem livremente num dos eixos para convergir/divergir o centro do eixo central da lente da câmara, assim como eixos que possibilitem simular os outros movimentos típicos da cabeça e do pescoço de modo a reproduzirem fidedignamente a visão humana e animal.

A estereoscopia ou visão estereoscópica define-se como a forma de obtenção de informações sobre o espaço tridimensional, através da análise de duas imagens obtidas em pontos diferentes (Figura 1). Assim, a noção de profundidade e a visão 3D deve-se ao facto de cada um dos olhos visualizar uma imagem de um ângulo ligeiramente diferente.

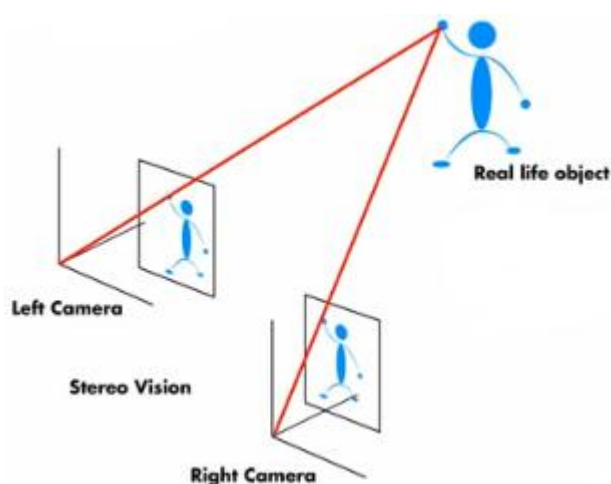


Figura 1- Duas imagens obtidas em pontos diferentes [2]

Esta técnica é amplamente usada em sistemas de vídeo e de processamento de imagem para o sistema computacional poder calcular a posição 3D, o tamanho ou a velocidade dos objetos focados pelas câmaras.

O sistema a desenvolver deverá ter a capacidade de simular a visão humana, onde os olhos convergem ou divergem, por forma a centrarem a imagem obtida por cada olho num determinado objeto, e utilizando o ângulo de rotação de cada olho para efeitos do cálculo da distância ao objeto. Para tal, serão utilizadas duas câmaras para simularem a obtenção das imagens pelos olhos, às quais estarão acoplados dois servomotor para fazerem convergir ou divergir as câmaras. Na estrutura estarão presentes mais dois servomotores, que farão rodar a estrutura de modo a simular os movimentos da cabeça e do pescoço dos humanos para focalizar a imagem obtida pelas câmaras.

1.1 Motivação

Na indústria, a utilização de sistemas robóticos tem crescido de maneira exponencial, pois são várias as áreas de aplicação em que podem ser inseridos. Nesse sentido, é necessário unir esforços para os inovar e torná-los cada vez mais eficientes.

O sistema de visão que é proposto desenvolver pode ser aplicado a várias situações, nomeadamente na determinação de profundidades e cálculo de distâncias através da localização no espaço dos objetos e, também, no reconhecimento de gestos. Outra possível aplicação prende-se com a adaptação deste sistema a robôs que já possuam sistemas de visão estéreo, nomeadamente, ao robô *AROS (Anthropomorphic Robot System)* (Figura 2) [3], desenvolvido na Universidade do Minho, em que a ideia seria substituir o atual sistema de visão por este que será desenvolvido, ou ao robô *MARy (Minho Autonomous Robot)* (Figura 3), desenvolvido na mesma universidade, uma vez que se trata de uma plataforma móvel de fácil adaptação a este tipo de robôs, podendo ser usada para resolver os problemas de deteção e localização de objetos.



Figura 2 - Robô AROS [3]



Figura 3 - Robô MARY

O facto de poder contribuir para melhorar a vida do ser humano, uma vez que, este tipo de robôs são muitas vezes usados para auxiliar pessoas que possuem dificuldades em realizar simples tarefas domésticas, torna-se num aliciente para a realização deste tema de dissertação.

1.2 Objetivos

Seguindo o que foi mencionado anteriormente, os objetivos gerais deste projeto de dissertação passa por construir uma plataforma que incorpore a eletrónica/mecânica

necessária para um sistema de visão estereoscópica. Depois de construída a estrutura, parte-se então para o desenvolvimento dos algoritmos para realizar o controlo do posicionamento dos servomotores que a irão dotar de movimentos autónomos, com o intuito de focalizar as imagens.

O passo seguinte passa por desenvolver os algoritmos para resolver o problema de processamento de imagem obtidas pelas câmaras, em que será testado um novo método de pesquisa que visa melhorar o rendimento destes sistemas. Este novo método permitirá determinar a localização do objeto nas câmaras, e posterior focalização do alvo em ambas.

Por fim, resta integrar os algoritmos desenvolvidos, de modo a que seja possível solucionar o problema que foi proposto para este tema de dissertação, e serão efetuados testes experimentais para averiguar a fiabilidade da solução desenvolvida em sistemas de visão estéreo.

1.3 Estrutura do documento

De seguida será explicada a disposição dos 6 capítulos presentes neste documento.

Neste capítulo, o 1º, é feita uma breve introdução aos sistemas de visão humana, onde estão presentes alguns conceitos fundamentais nesta área e onde é descrita a motivação para a realização deste projeto, assim como delineados os principais objetivos a cumprir.

No capítulo 2, é feita uma revisão bibliográfica onde se apresenta o que de relevante existe relacionado com este tema de dissertação, de modo a ficar a por dentro dos sistemas já existentes e que se torna indispensável conhecer para a realização da mesma.

No 3º capítulo são abordados os materiais que foram necessários para construir a plataforma que serviria de base para a implementação do sistema de visão estéreo. Nele são apresentados as principais características de cada um e, no final, apresenta-se o resultado final da estrutura.

No capítulo 4 é apresentado como foi feito o controlo dos servomotores, nomeadamente do que diz respeito á velocidade de movimentação, assim como da sua posição. Também é estudado o funcionamento da placa *Devantech* SD21 que faz esse

controlo sobre os servomotores e, por fim, como é feita a comunicação entre esta placa e o *Arduino*.

No capítulo 5 é descrito o método que foi adotado para solucionar o sistema de visão estéreo. Aqui é descrita que a forma como se determinou a localização do objeto nas imagens e apresentada a solução escolhida para focalizar o mesmo.

Por último, no capítulo 6 é feita uma conclusão sobre o trabalho efetuado e do que ficou por realizar, assim como aspetos a melhorar no futuro.

Capítulo II

2. Revisão Bibliográfica

2.1 Visual Tracking Control to Fast Moving Target for Stereo Vision Robot

Em 2010 foi publicado este artigo [4] numa conferência que decorreu em Nagaoka, no Japão, na *The 11th IEEE International Workshop on Advanced Motion Control*, que descreve como feito o controlo de acompanhamento visual do robô com visão estéreo para casos em que o alvo se encontre em movimento. Para realizar este acompanhamento visual por parte do robô utilizaram-se duas câmaras iguais ligadas a um rodapé fixo e, com o intuito de as fazer mover, foram colocados dois motores DC em cada eixo de rotação dessas câmaras. Por fim, foi ligado outro motor do mesmo tipo ao centro do rodapé para fazer rodar toda a estrutura desenvolvida, totalizando dessa forma, três motores DC que são controlados de forma independente (Figura 4).



Figura 4 - Esquemático Estrutura (Vista de Frente) [4]

Para solucionarem o problema, os autores começaram por definir o plano de imagem e sistema de coordenadas (Figura 5). De destacar o facto de cada plano de imagem ser definido para ser vertical para o eixo Z da respetiva câmara e os eixos u e v de cada plano de imagem estarem definidos para serem paralelos aos eixos x e y, respetivamente, do sistema de coordenadas da câmara.

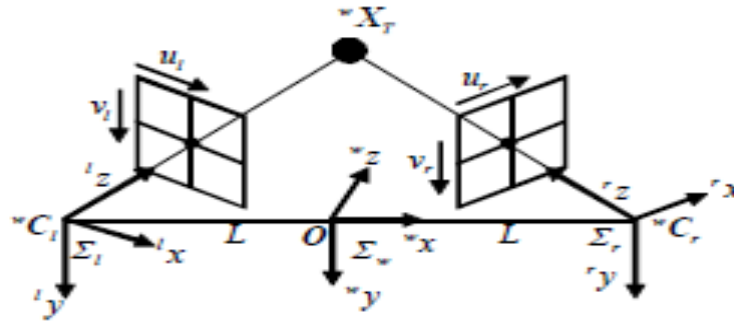


Figura 5 - Plano de imagem e sistema de coordenadas [4]

Para realizar o controlo de acompanhamento visual, os motores colocados em cada câmara fazem girar os seus próprios eixos para capturar o alvo no centro da imagem. Assim, as duas câmaras conseguem focar a imagem, e através da triangulação é possível calcular a distância a que o robô se encontra do alvo.

Para averiguar o sistema desenvolvido foram testadas duas formas diferentes de movimento do alvo: uma em que o alvo se move em alta velocidade mas com aceleração constante e outra em que se move com uma velocidade de rotação constante mas com aceleração diferente. Os principais problemas que podem gerar erros no acompanhamento prendem-se com o tempo e precisão do processamento da imagem e também com o atraso no movimento efetuado pelo robô.

No primeiro caso, em que o alvo se move em alta velocidade e com aceleração constante, o seguimento visual foi conseguido com sucesso visto que o comportamento verificado estava muito próximo com o que se esperava (Figura 6).

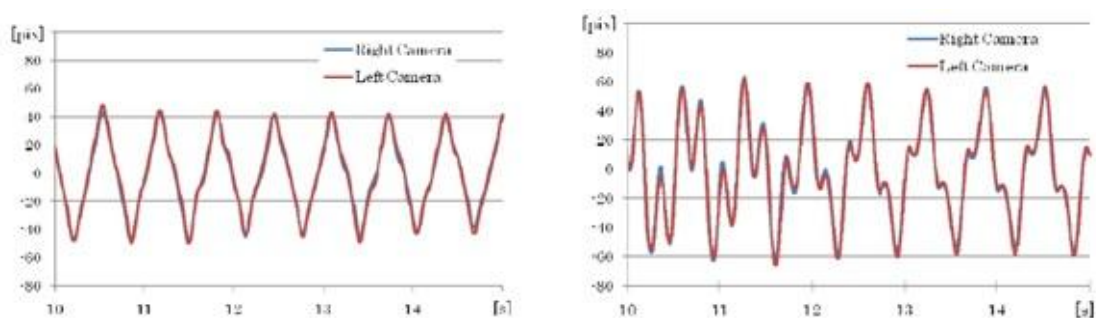


Figura 6 – Caso1: comportamento esperado (lado esquerdo) e comportamento verificado (lado direito) [4]

Através da Figura 7 pode deprender-se que no segundo caso, em que o alvo se move com velocidade de rotação constante mas com aceleração diferente, o rastreamento visual tornou-se mais complicado, sendo nestes casos necessário estimar com bastante precisão essas possíveis variações da aceleração para realizar o rastreamento visual com sucesso.

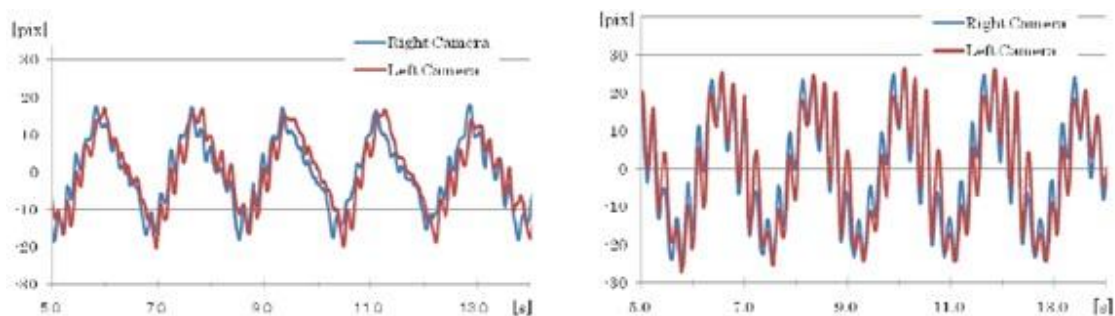


Figura 7 – Caso2: comportamento esperado (lado esquerdo) e comportamento verificado (lado direito) [4]

2.2 Making a step to stereo vision

Um outro artigo [5], publicado em 2009 e da autoria de *Andrew Kirillov*, visa analisar a construção de uma plataforma do tipo *pan-tilt*, para que possa ser usada em sistemas de visão estéreo. A plataforma resume-se a dois servomotores ligados entre si, formando o módulo *pan-tilt*, e a 2 webcams iguais, que assentam numa base em madeira, mantendo as mesmas sempre fixas sobre essa base (Figura 8).



Figura 8 - Vista de frente da plataforma [5]

A detecção dos objetos pelas câmaras é realizada por intermédio da técnica de segmentação por cores, sendo que, neste caso, foram aplicados filtros de cor para a componente azul da imagem (Figura 9).

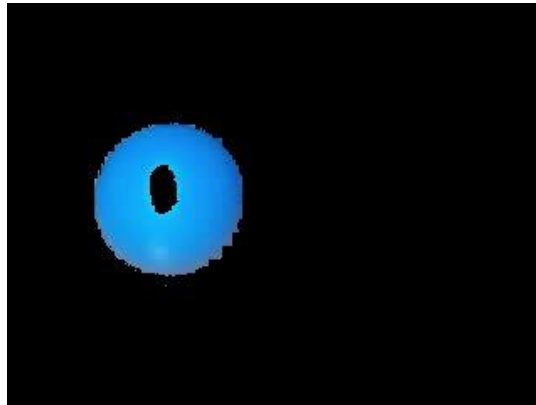


Figura 9 - Segmentação da cor azul [5]

O modo como foi realizado o rastreamento dos objetos passou pela caracterização da posição do objeto, conseguida através do cálculo do seu centro de massa, para cada câmara, uma vez que cada uma tem uma percepção diferente do objeto, pois é obtida em ângulos diferentes. A solução passou por calcular a média das coordenadas do centro de massa de cada uma, para que o objeto não fosse centrado apenas em uma das câmaras, mas sim numa posição intermédia entre as duas.

Por fim, as coordenadas do ponto médio são fornecidas aos servomotores, transformando-as em ângulos de rotação para o módulo *pan-tilt*. De realçar que neste tipo de sistemas, através da análise da diferença das coordenadas do centro de massa, é possível estimar da posição do objeto em relação às câmaras, visto que, caso essa diferença seja muito grande, significa que o objeto está próximo das câmaras e vice-versa.

2.3 Visão Estéreo

Nos sistemas tradicionais de visão estéreo são, normalmente, utilizadas coordenadas espaciais para a obter a localização dos objetos, sendo possível obtê-las através de algumas técnicas baseadas neste tipo de visão. Uma das formas mais fiável na obtenção dessas coordenadas prende-se com a captura de pares de imagens, em simultâneo, tal como acontece com o sistema de visão humano (Figura 10) [6].

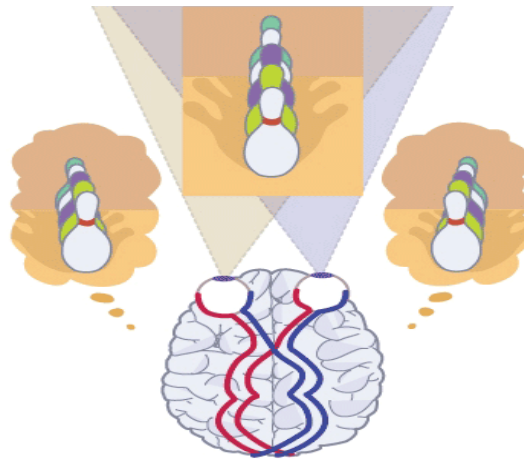


Figura 10 - Percepção do ser humano do espaço [6]

Este sistema baseia-se em combinar as semelhanças e somar as diferenças das imagens obtidas por cada olho, sendo o cérebro o responsável por essa tarefa, transformando essas duas imagens, ligeiramente diferentes, numa só. Assim, a imagem final, a combinada pelo cérebro, não se trata apenas da soma das duas imagens, mas sim de uma imagem estereoscópica tridimensional.

O princípio desta técnica pode ser aplicada a nível computacional, isto é, após a captura das duas imagens por parte das câmaras fixas, estas são analisadas ao pormenor, procurando obter correspondências entre elas e, através da análise das suas diferenças, é possível obter o valor da profundidade a que se encontra o objeto.

Na Figura 11 encontra-se o esquema representativo do posicionamento de duas câmaras estereo [7] e na Figura 12 encontra-se o plano epipolar [8] observado por duas câmaras de um ponto P , as imagens p e p' , com centros óticos em O e O' . O objetivo com este plano é estabelecer as tais correspondências referidas anteriormente entre as duas imagens p e p' , isto é, a da câmara esquerda e da direita, respetivamente.

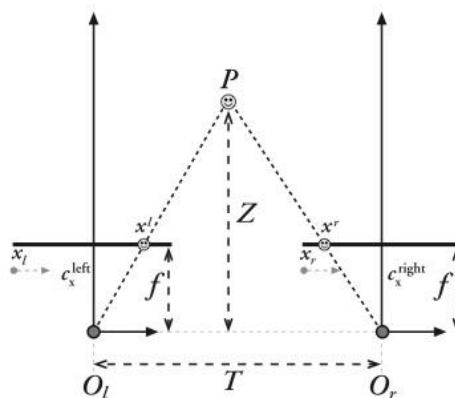


Figura 11 - Esquema do posicionamento de duas câmaras *stereo* [7]

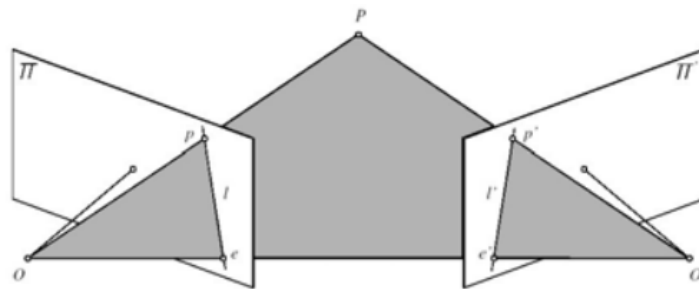


Figura 12 - Geometria epipolar [8]

2.3.1 Calibração de uma câmara

Quando se trata de visão estereo é preciso ter muita atenção à qualidade das imagens que irão ser tratadas. Isto é, as câmaras devem estar perfeitamente calibradas para que seja possível obter correspondência entre pixéis das duas imagens obtidas por elas. Esta calibração permite, por exemplo, corrigir a distorção das suas lentes, como podemos ver na Figura 13, um exemplo de uma imagem distorcida antes da calibração e uma outra imagem correta depois de calibrada.



Figura 13 - Imagem distorcida (esquerda) e imagem calibrada (direita) [2]

2.3.1.1 Modelo da câmara

Para fazer a reconstrução 3D do espaço, a forma mais usual é o modelo de câmara pontual (Figura 14). Nos sistemas de visão existentes é feita uma aproximação a este modelo, dada a impossibilidade de realização física deste. Isto é, o plano de imagem passa a ser colocado à frente do centro das lentes e o eixo de coordenadas é reposicionado, trocando, assim, as suas posições (Figura 15).

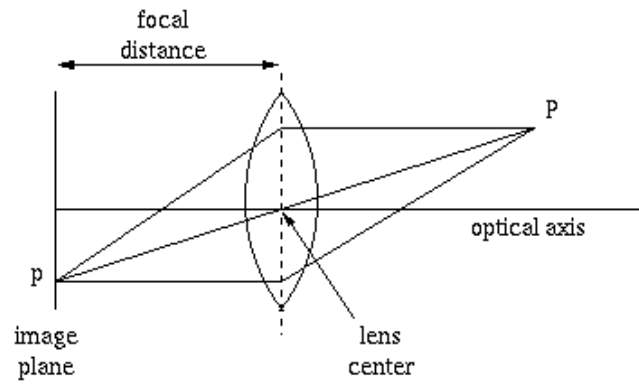


Figura 14 - Modelo usual da câmera pontual [9]

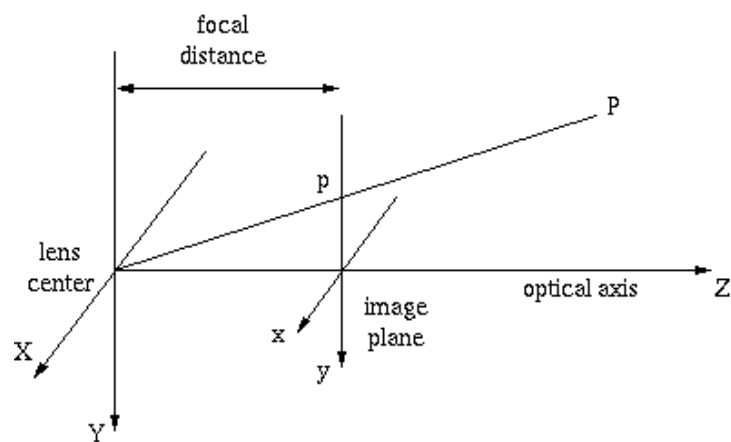


Figura 15 - Modelo aproximado da câmera pontual [9]

No modelo de câmera pontual, a relação entre as coordenadas do ponto $P(X,Y,Z)$ e as coordenadas do plano da imagem (x,y) pode ser feita pelas equações 1 e 2:

$$x = f * \frac{X}{Z} \quad (1)$$

$$y = f * \frac{Y}{Z} \quad (2)$$

onde f representa a distância entre o plano de imagem e o centro das lentes.

2.3.2 Retificação estéreo

Para obter uma correspondência entre as duas imagens, é necessário que os seus dois planos estejam bem alinhados. Manualmente, este processo torna-se bastante difícil de ser conseguido com total exatidão, pelo que, para corrigir essa situação, recorre-se à retificação desses mesmos planos. Esta técnica é bastante usual em visão computacional, para otimizar o processo de correspondência entre os pontos nas imagens (Figura 16).

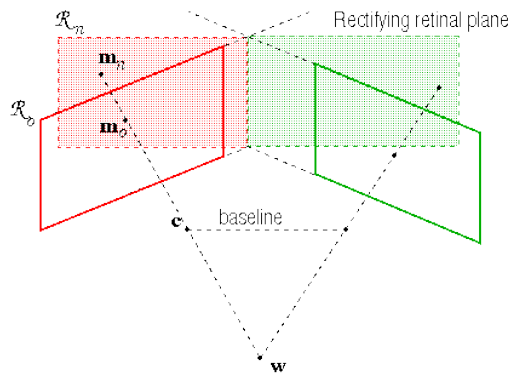


Figura 16 - Retificação dos planos de imagem [10]

2.3.3 Mapa de disparidades

O conceito de disparidade e de pontos homólogos, em visão estéreo, prende-se com o facto de com a aquisição de duas imagens poderem existir características que contêm e que se assemelham. Quando existe correspondência entre pontos de dois planos de imagem diferentes (x_l, y_l) e (x_r, y_r) , consideram-se que são pontos homólogos. No sentido inverso, quando não existe essa correspondência, é porque existe uma disparidade entre as coordenadas dos dois planos de imagem, que pode ser relacionada da forma da equação 3:

$$disp(x, y) = \begin{bmatrix} x_r \\ y_r \end{bmatrix} - \begin{bmatrix} x_l \\ y_l \end{bmatrix} = \begin{bmatrix} x_r - x_l \\ y_r - y_l \end{bmatrix} \quad (3)$$

Através da equação, é possível produzir um mapa de disparidades, deduzindo que cada ponto da imagem caracteriza-se por haver disparidades entre os planos das imagens obtidas. Na Figura 17, está representado o gráfico que representa a relação que

existe entre a distância desse ponto em relação às câmaras, com a diferença de coordenadas.

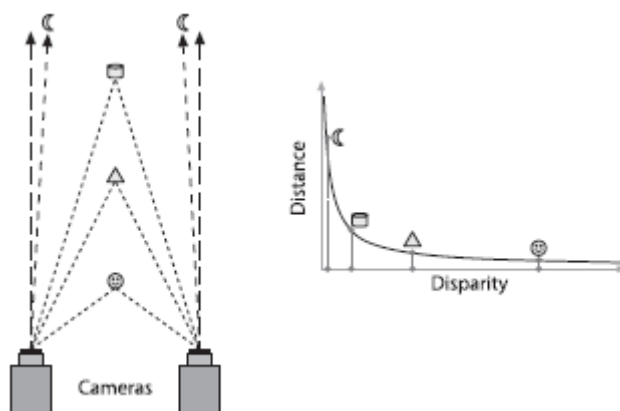


Figura 17 - Gráfico com relação entre distância e a disparidade [7]

Como podemos verificar, a profundidade é inversamente proporcional à disparidade, uma vez que à medida que a distância do objeto em relação às câmaras aumenta, a sua disparidade também aumenta, não sendo esta relação linear, como se verifica pelo gráfico. Analisando os dados obtidos desta informação, é possível usar os sistemas de visão estereoscópica para efeitos de cálculos de distâncias, tendo em conta que, estes sistemas possuem alta resolução de profundidade apenas para objetos que se encontrem relativamente perto das câmaras.

2.4 Algoritmos de deteção de objetos

Nesta área de visão por computador, existem alguns algoritmos que servem de base para as suas aplicações, e que oferecem uma grande confiabilidade quando se trata de deteção de objetos.

2.4.1 SIFT (*Scale-Invariant Feature Transform*) [11]

Este algoritmo, que foi publicado em 1999 por *David Lowe*, é usado em sistemas de visão computacional e muito útil para reconhecimento de objetos e de gestos, criação de sistemas 3D, mapeamento e navegação robótica. Esta técnica divide-se, essencialmente, em duas partes: deteção dos pontos-chave, em que se recorre à *Diferença de Gaussianas (DoG)* e a descrição desses mesmos pontos recorrendo a histogramas que indicam a sua orientação.

2.4.1.1 Detecção de pontos-chave

Com o objetivo de obter os pontos-chave da imagem, independentemente das escalas usadas, recorre-se á função Gaussiana, que provoca sucessivos desfoques na imagem, suavizando-a. Na Figura 18 encontra-se um exemplo dessa transformação, neste caso aplicada sobre uma impressão digital.



Figura 18 - Imagem de uma impressão digital original e após sucessivas desfocagens [12]

A localização dos pontos-chave na imagem é obtida, como já referido, através do cálculo da *Diferença de Gaussianas (DoG)* entre duas escalas consecutivas, demonstrado na equação 4:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (4)$$

onde k representa o fator multiplicativo para obter a escala seguinte.

2.4.1.2 Descrição de pontos-chave

Uma vez obtida a localização dos pontos de interesse, o passo seguinte passa por saber qual a magnitude (m) e orientação (θ) de cada um. Para calcular esses parâmetros recorre-se às equações 5 e 6:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (5)$$

$$\theta(x, y) = \tan^{-1}\left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)}\right) \quad (6)$$

Para organizar estes dados é criado um histograma (Figura 19) com as orientações dos gradientes de cada pixel em redor do ponto-chave. Através da análise que é feita ao histograma, são tomados em conta os seus picos, o que pode levar a que cada ponto-chave tenha mais que uma orientação.

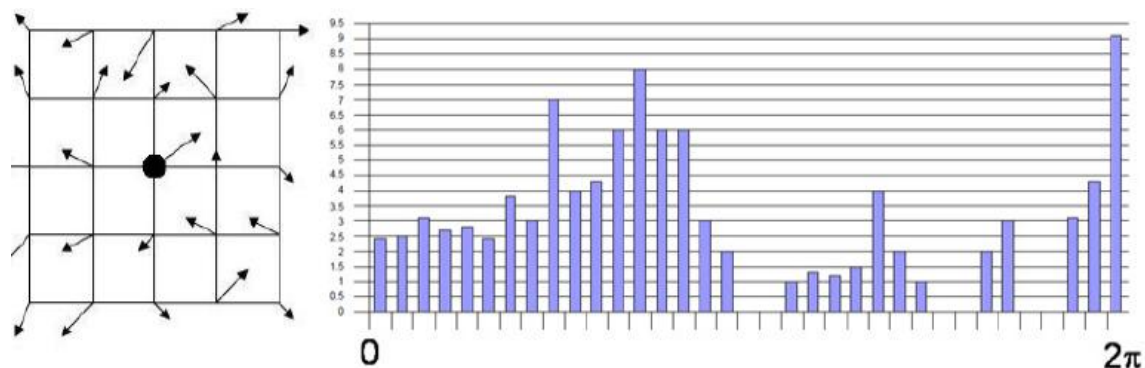


Figura 19 - Histograma com orientações dos pontos-chave [11]

2.4.2 SURF(Speeded-Up Robust Features) [13]

Este outro algoritmo foi apresentado pela primeira vez por *Herbert Bay* [14] em 2006, sendo muito eficaz em sistemas de visão computacional para deteção e reconhecimento de objetos, calibração de câmaras, entres outros. Esta técnica, que teve como base de estudo o *SIFT*, apresenta, entre outras, a principal vantagem de o seu processamento ser cerca de 3 vezes mais rápido do que *SIFT* e mais robusto no processo de deteção de objetos, isto é, consegue facilmente localizar objetos independentemente da forma ou do local em que se encontre.

2.4.2.1 Deteção de pontos de interesse (*Keypoints*)

Para detetar os pontos de interesse, esta técnica recorre à matriz Hessiana, graças à sua elevada precisão. Estes pontos são encontrados quando o determinante da matriz atingir o valor máximo. Considerando um ponto $x = (x, y)$ de uma imagem, a matriz Hessiana $H(x, \sigma)$ é dada pela equação 7:

$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \quad (7)$$

Em que $L_{xx}(x, \sigma)$, $L_{xy}(x, \sigma)$, $L_{yy}(x, \sigma)$ representam a convulsão Gaussiana da derivada de segunda ordem $\frac{\partial^2}{\partial x^2} g(\sigma)$ do ponto x da imagem. Em contrapartida, este processo tem o inconveniente de se tornar demasiado pesado, aumentando o tempo de processamento, algo que não é desejado. A solução passa, então, pela discretização das derivadas gaussianas de segunda ordem, mas, e para não cair no mesmo erro e tornar o processo novamente lento pelo aumento do número de filtros, é feita uma aproximação a esses filtros que passa por dividir em retângulos, denominados filtros de caixa, a derivada gaussiana de segunda ordem (Figura 20).

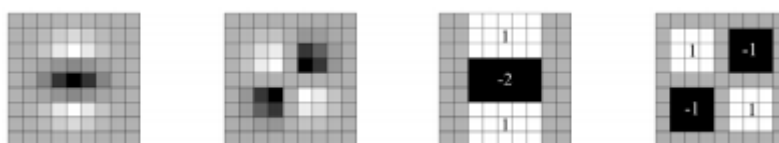


Figura 20 - Divisão em retângulos para tornar o processo mais rápido [13]

À alteração de escalas da imagem original dá-se o nome de espaço de escalas e é implementada sob a forma de pirâmide invertida (Figura 21), e independentemente dessas escalas, este algoritmo permite a obtenção dos pontos de interesse através do aumento do filtro da imagem, que consiste no uso de imagens integrais.

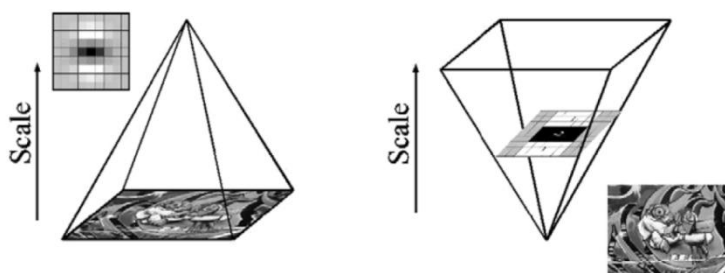


Figura 21 - Espaço de escalas em forma de pirâmide invertida [13]

Para proceder à localização destes pontos de interesse na imagem, conseguida por intermédio do cálculo do determinante da matriz Hessiana para cada píxel, onde se determina os mínimos e os máximos do mapa de resposta.

Para finalizar a parte de deteção destes pontos, resta saber qual a orientação de cada um, calculando a resposta da convulsão para os dois filtros (Figura 22), designados

por *Haar Wavelets*, e analisando-os, observa-se as variações que ocorrem na direção x e y(Figura 23).

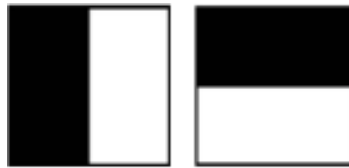


Figura 22 - Filtros Haar Wavelets em x (esquerda) e em y (direita) [13]

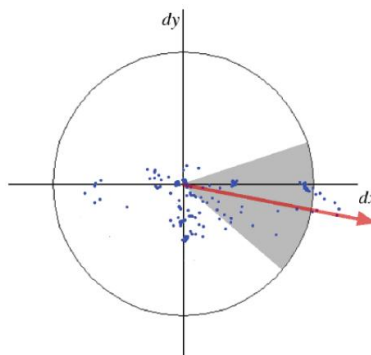


Figura 23 - Orientação dos pontos de interesse [13]

2.4.2.2 Descrição dos pontos de interesse

Por último, falta descrever os pontos de interesse que consiste em pegar num deles e partindo do seu centro implementar um quadrado, como o da Figura 24. O processo baseia-se em criar uma janela à volta do ponto de interesse e calcular, de novo, as respostas ao filtro *Haar Wavelets* (Figura 22).

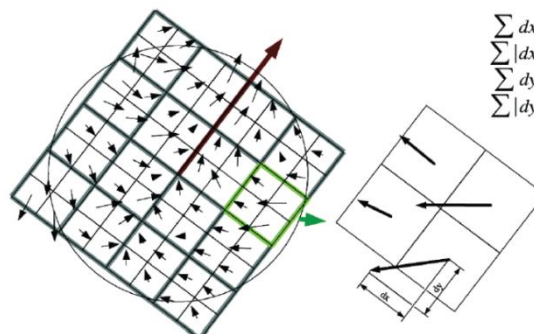


Figura 24 - Quadrado de respostas de *Wavelets* [13]

2.5 Equipamentos comerciais

Atualmente no mercado existem alguns equipamentos que serviriam de base para aplicar estas técnicas de visão stereo, nomeadamente, o *MobileRanger Stereo Camera* (Figura 25) [15]. Este equipamento pode ser aplicável a várias situações, tais como em navegação de robots móveis, rastreamento de pessoas, reconhecimento de objetos e visualização 3D da superfície. Este sistema possui uma resolução de 752x480 píxeis, a 30 FPS com FPGA e tem como enorme vantagem o seu tamanho bastante reduzido, podendo ser facilmente aplicável a sistemas do tipo *pan-tilt*.



Figura 25 - Câmara stereo *MobileRanger* [15]

Outro exemplo de equipamento é o *Bumblebee2* da Point Grey (Figura 26) [16]. Este sistema inclui uma cópia do *FlyCapture SDK*, que é utilizado para a aquisição de imagens e para o controlo das câmaras e o *Triclops SDK* que realiza a retificação e processamento das imagens stereo. É composto por duas câmaras com uma resolução de 640x480 píxeis, a 48 FPS.



Figura 26 - Câmara stereo *Bumblebee2* [16]

Capítulo III

3. Plataforma desenvolvida

Tal como foi dito anteriormente, seria necessário construir a estrutura que serviu de suporte para a realização deste projeto de dissertação. Tendo em conta os componentes que seriam precisos e com o intuito de reduzir os custos financeiros associados a este projeto, recorreu-se aos materiais que havia disponíveis no Laboratório de Automação e Robótica e nas oficinas do DEI, sendo possível adaptar duas câmaras *Philips* e dois servomotores. Os restantes componentes tiveram que ser adquiridos, nomeadamente mais dois servomotores idênticos aos disponíveis, uma placa de controlo e uma bateria para alimentação dos servomotores e, finalmente, o microcontrolador *Arduino Mega 2560*. Todos estes componentes assentam numa base circular em madeira formando, assim, a plataforma final.

3.1 Componentes utilizados

3.1.1 Servomotor *DF15 Metal Gear*

De forma a satisfazer um dos objetivos deste projeto, no qual o sistema deverá movimentar-se em todos os eixos, foram utilizados quatro servomotores *DF15 MG* (Figura 27). Cada um deles tem uma função específica, mais concretamente:

- ***Pan***: Diretamente na base e com a função de movimentar o sistema para a esquerda e para a direita;
- ***Tilt***: Acoplado no *pan* e com o intuito de mover o sistema para cima ou para baixo;
- ***Olhos***: Acoplados em cima do módulo *pan-tilt* e destinado a convergir ou divergir as câmaras.



Figura 27 - Servomotor *DF15 Metal Gear* [17]

Estes servomotores têm como características principais:

- Tensão de Operação: +48V ~ 7.2V;
- Corrente: 180mA (4.8V); 220mA (6V);
- Binário: 10Kg • cm (4.8V) 12KG • cm (6V) 15KG • cm (7.2V);
- Rolamento duplo;
- Capacidade de rotação de 120°. (60° numa direção, 60° na outra);
- Peso: 180g.

3.1.2 Alimentação servomotores

3.1.2.1 Bateria *KingMax*

A bateria representada na Figura 28 foi utilizada para fornecer tensão e corrente suficiente para alimentar o circuito de acionamento dos servomotores. Esta bateria *KingMax* tem como características principais:

- Bateria Li-Po de elevada descarga;
- Duas células de 3,7 V;
- Capacidade de fornecimento de uma corrente de 1000 mAh;
- Tensão máxima: 7,4 V.



Figura 28 - Bateria *KingMax* 1000mAh [17]

3.1.2.2 *Traco Power* Conversor DC-DC

Como a tensão que a bateria fornece é superior à suportada pelos servomotores, houve a necessidade de incorporar o conversor DC-DC (Figura 29), com o intuito de baixar essa tensão. O *Traco Power* tem como principais características:

- Suportar tensões de entrada entre 6,5 e 36V;
- Converter para uma tensão de saída de 5V;
- Corrente de saída: 1A;
- Proteção interna contra curto circuitos.



Figura 29 - Conversor DC-DC *Traco Power* [17]

3.1.3 *Arduino Mega 2560*

O *Arduino Mega 2560* (Figura 30) é um microcontrolador com 54 pinos de entradas/saídas digitais, 16 entradas analógicas, 4 *UARTs* (portas série de hardware), um oscilador de cristal de 16 MHz, uma conexão USB, uma entrada de alimentação e um botão de *Reset*.



Figura 30 - Microcontrolador *Arduino Mega 2560* [18]

3.1.3.1 Alimentação microcontrolador

Este microcontrolador pode ser alimentado via USB, através de uma fonte de alimentação externa, ou ainda através de baterias. Dado que a plataforma é didática, necessitando de constante ligação ao computador para comunicação série e processamento de dados em tempo real, foi alimentado diretamente pela porta USB do computador.

3.1.3.2 Pinos de entrada e de saída

Cada um dos 54 pinos digitais do *Arduino Mega 2560* pode ser usado como entrada ou saída. Entre os 54 pinos digitais, dois em particular serão utilizados na comunicação entre o microcontrolador e a placa de controlo dos servomotores, nomeadamente os pinos 20 e 21, SDA e SCL, respetivamente, que representam os pinos de dados e *clock*.

3.1.3.3 Comunicação

O microcontrolador possui a capacidade de comunicar com o computador, na medida em que fornece quatro portas de comunicação série *UART*. Para estabelecer a conexão entre o *Arduino* e a placa de controlo dos servomotores *Devantech SD21*, foi necessário recorrer ao protocolo *I2C*.

3.1.4 Placa *SD21 Servo Controller*

A *Devantech SD21* é uma placa de controlo de servomotores. Esta placa tem a capacidade de controlar até 21 servomotores, mais concretamente a posição e velocidade de cada um.

Os servomotores a controlar são ligados diretamente à placa, nos *sockets* mais à direita (Figura 31) e a interligação via I2C é feita através de um dos três conectores I2C colocados na parte superior esquerda da placa.

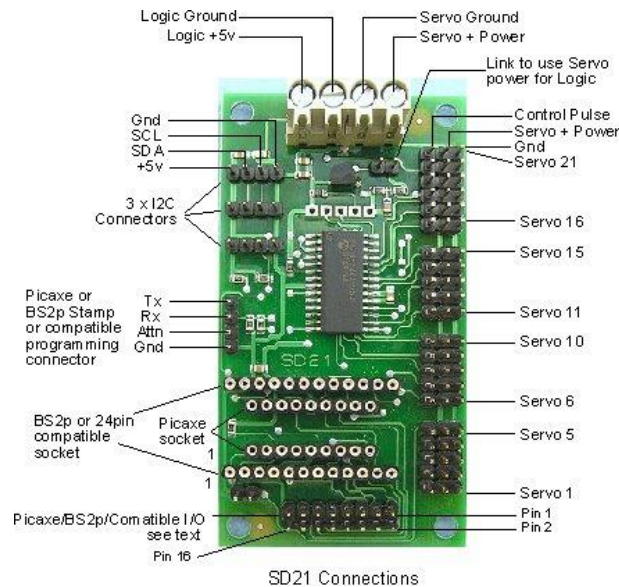


Figura 31 - Placa de controlo SD21 [19]

3.2 Circuito acionamento dos servomotores

Para realizar a integração do microcontrolador utilizado, o *Arduíno Mega 2560*, com a placa *SD21 Servo Controller* foi necessário implementar o circuito representado na Figura 32.

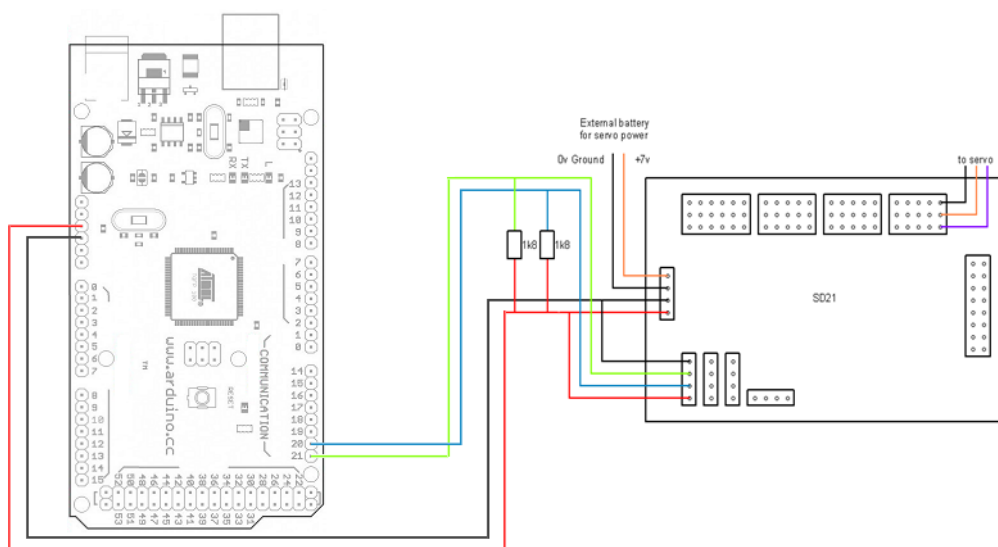


Figura 32 - Circuito acionamento dos servomotores

O controlo da posição e da velocidade dos servomotores é efetuado através da comunicação série entre o computador e o microcontrolador. Como tal, é necessário fazer a integração do *Arduíno* com a placa SD21 para que os comandos enviados acionem os respetivos pinos de controlo dos servos e, conseqüentemente, estes executem os movimentos pretendidos. Nesse sentido, foram ligadas duas resistências de 1,8 k Ω aos 5V do *Arduíno* e aos pinos de dados e de *clock* do mesmo. As saídas de *clock* e de dados do microcontrolador foram posteriormente ligadas à entrada da placa SD21 *Servo Controller*, juntamente com os 5V e a massa.

Finalmente, é necessário proceder à alimentação da placa que aciona os servomotores. Para tal, foi usada a bateria *KingMax* em conjunto com o conversor DC-DC *Traco Power*, como explicado anteriormente.

3.3 Plataforma final

Na Figura 33 encontra-se a plataforma construída com todos os elementos referenciados anteriormente, perfazendo, assim, o sistema final de visão estereoscópica que foi desenvolvido para a realização desta dissertação. Esta estrutura tem a vantagem de ser facilmente transportável e adaptável a outra situação em que possa ser aplicado este tipo de sistemas de visão.

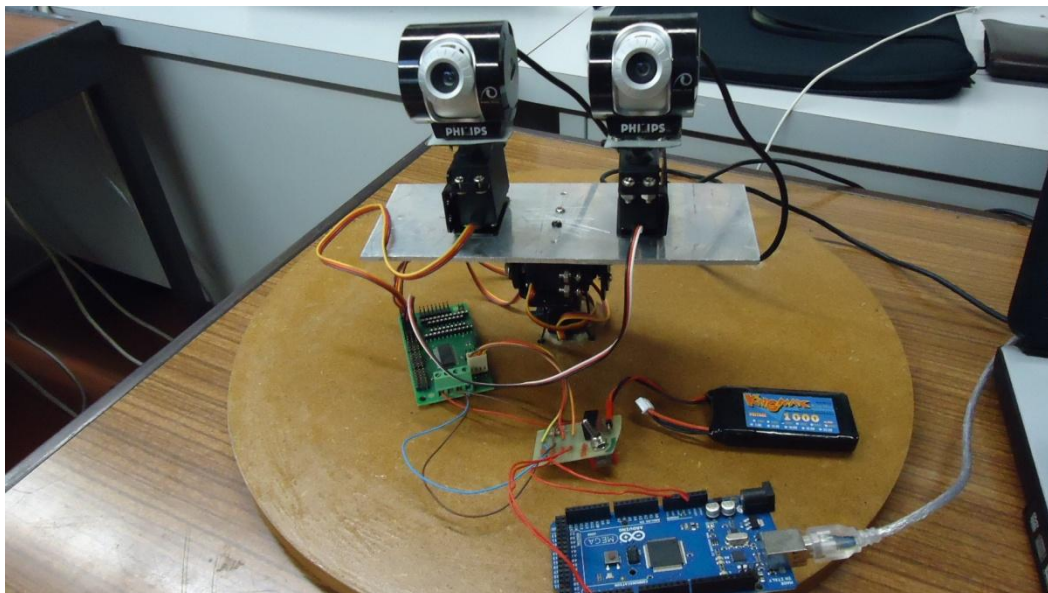


Figura 33 - Sistema de visão estéreo desenvolvido

Capítulo IV

4. Controlo dos servomotores

O processo de controlo do posicionamento, da velocidade e da direção dos servomotores requer um conhecimento prévio do funcionamento dos vários elementos constituintes da plataforma, nomeadamente, o funcionamento da placa *Devantech* SD21 e, conseqüentemente, o modo de operação dos servomotores DF15MG e a comunicação via I2C entre o microcontrolador e a placa SD21.

4.1 Funcionamento da placa de controlo *Devantech* SD21

A *Devantech* SD21 tem a capacidade de controlar a velocidade e a posição de até 21 servomotores. A definição de velocidades ou posições é feita através do envio de comandos para a PIC18F2220, presente na placa, através do barramento I2C. Existem três conectores I2C no controlador SD21, cada um com 4 ligações, a saber:

- GND
- SCL – Linha de Clock
- SDA – Linha de Dados
- +5V

4.1.1 Processamento do controlador SD21

O acesso à PIC18F2220 e, conseqüentemente, aos registos internos dedicados a cada servomotor, é feito através do barramento I2C no endereço 0xC2. Existem, para cada servomotor, três registos, nomeadamente:

- Velocidade;
- Byte Alto para a Posição;
- Byte Baixo para a Posição;

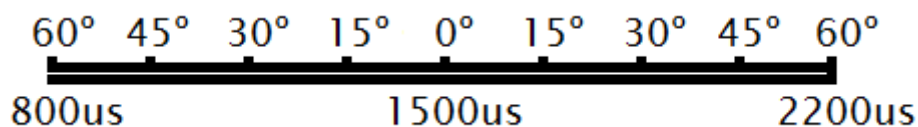
Sabendo que são utilizados quatro servomotores, os registos estão apresentados na Tabela 1:

Tabela 1 - Registos utilizados da placa SD21

Registo	Servo	Função
0	1	Velocidade
1	1	Byte Alto
2	1	Byte Baixo
3	2	Velocidade
4	2	Byte Alto
5	2	Byte Baixo
6	3	Velocidade
7	3	Byte Alto
8	3	Byte Baixo
9	4	Velocidade
10	4	Byte Alto
11	4	Byte Baixo

4.1.2 Posição dos servomotores

Trabalhando com a SD21, sabe-se que a posição é representada por um número de 16-bit definido em μs . Genericamente, a posição central do servomotor é dada por $1500\mu\text{s}$ e cada servomotor poderá ver a sua posição num limite entre $800\mu\text{s}$ e $2200\mu\text{s}$. Nesse sentido, cruzando as informações de cada servomotor *DF15 Metal Gear*, que tem um movimento de rotação máximo de 120° (60° para cada lado), poderá deduzir-se a relação entre a posição em μs e a posição em graus é a expressa na Figura 34.

Figura 34- Relação entre posição em μs e posição em Graus

4.1.3 Velocidade dos servomotores

O registo de velocidade define a velocidade à qual os servomotores se movem para a posição definida pela aplicação. Se o registo de velocidade for diferente de zero, esse valor será adicionado à posição atual do servomotor a cada 20ms , até que a posição final seja atingida.

Por exemplo, se o servomotor *Pan* se mover de 0° (1500µs) para 60° (2200µs) à direita com o registo de velocidade definido como 5 (o registo de velocidade pode ser definido entre 1 e 10, em que 1 corresponde à velocidade menor e 10 ao maior valor da velocidade de movimentação), a fórmula que determina o tempo que demorará a efetuar esse movimento é dado pela equação 8:

$$\frac{\text{Posição Final} - \text{Posição Inicial}}{\text{Registode Velocidade}} * 20ms \Leftrightarrow \frac{2200 - 1500}{5} * 20ms = 2,8 s \quad (8)$$

Nesta situação, o tempo que demorava a executar o movimento seria de 2,8 segundos.

4.2 Comunicação I2C entre *Arduino* e *Devantech SD21*

Para estabelecer a conexão entre o *Arduino* e a placa de controlo dos servomotores *Devantech SD21*, é necessário recorrer ao protocolo I2C, através o endereço 0xC2. Essa comunicação estabelece-se através da ligação entre o conector I2C presente na SD21 e os pinos de dados e de *clock* do microcontrolador, nomeadamente os pinos 20 (SDA) e 21 (SCL).

O *Arduino Mega2560* fornece suporte para este tipo de comunicação através de uma biblioteca denominada *Wire* que simplifica o uso do I2C.

4.2.1 Biblioteca *Wire*

Para o desenvolvimento do *firmware* de controlo do posicionamento dos servomotores, recorreu-se a várias funções presentes na biblioteca *Wire*, nomeadamente:

- ***Wire.beginTransmission(address)***: A função *beginTransmission()* inicia a comunicação via I2C para o dispositivo alocado no endereço *address*. No que concerne à placa SD21, o endereço é igual a 0xC2.
- ***write()***: Esta função escreve os dados enviados pela aplicação e é sempre chamada entre as funções *beginTransmission()* e *endTransmission()*;

- ***Wire.endTransmission()*** (): Esta função finaliza a transmissão via I2C que foi inicializada por uma *beginTransmission()* e transmite os dados enviados pela função *write()*;

4.3 Controlo de posição dos servomotores

Feitas as análises ao funcionamento dos elementos constituintes do sistema, é a altura de descrever como foi realizado o controlo da posição dos servomotores. Inicialmente, foram desenvolvidos os algoritmos de controlo dos dois servomotores que controlam a posição do pescoço e da cabeça, *pan* e *tilt*, respetivamente. De realçar que para estabelecer a comunicação entre o programa e o sistema de controlo do posicionamento dos servomotores, deverá ser constante a leitura da porta-série. Na Figura 35 está representado o esquema desses dois servomotores a controlar.

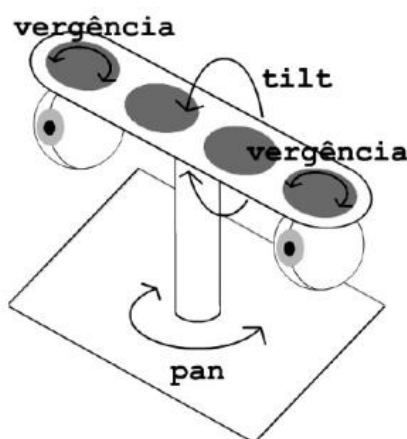


Figura 35- Esquema dos servomotores *pan* e *tilt* [20]

Quando o programa se inicia, o objeto pode encontrar-se em qualquer um dos quadrantes da imagem. Isto é, no caso do servomotor *pan*, é necessário verificar se o alvo se encontra do lado esquerdo ou do lado direito dessa imagem para indicar o sentido de movimento que deve tomar. Assim, após a deteção do alvo e tendo a informação do quadrante em que se encontra, o sistema fica informado do sentido em que se deve mover para que o mesmo fique centrado na imagem capturada pelas câmaras. Análogo é o princípio de funcionamento do servomotor *tilt*, diferindo apenas no sentido de movimentação, isto é, ajusta a sua posição para cima (caso o alvo se encontre no 1º ou 2º quadrante) e para baixo (alvo no 3º ou 4º quadrante).

Para exemplificar, na Figura 36, está apresentado uma possível localização do objeto na imagem capturada pelas câmaras. Num caso similar a este, em que o alvo se encontra no 3º quadrante, o *pan* teria de se movimentar para a esquerda e o *tilt* para baixo para se centrar o objeto na câmara.

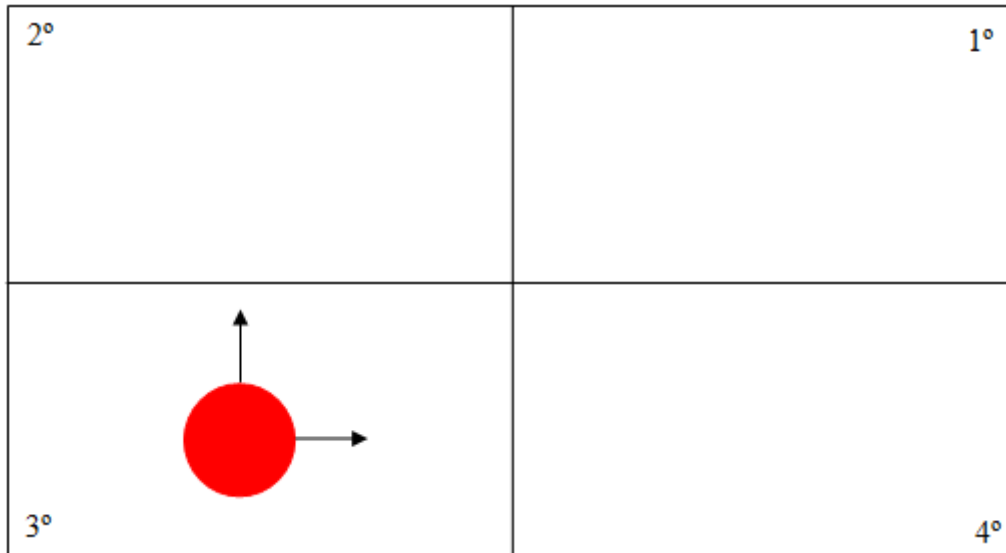


Figura 36 - Exemplo de localização do alvo

4.4 Algoritmo de controlo de posição dos servomotores

No fluxograma presente na figura 37 encontram-se descritas as fases anteriormente referidas dos servomotores, de maneira sucinta e de fácil interpretação, para centrar o objeto na imagem. De realçar apenas que o *Arduino*, que envia os comandos para executar os seus movimentos à placa que controla os servomotores, está em permanente contato com o software de desenvolvimento, através de comunicação série, recebendo os valores dos ângulos a serem aplicados.

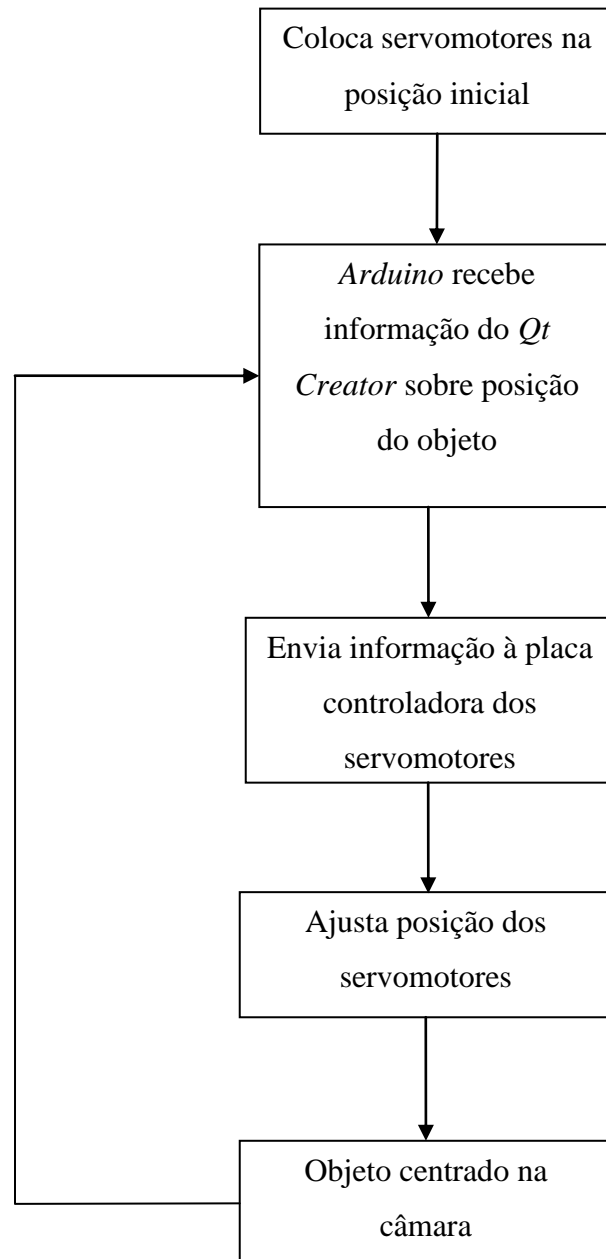


Figura 37 - Algoritmo de controlo de posição dos servomotores

Capítulo V

5. Método desenvolvido e resultados

Ao longo deste capítulo está apresentado o método adotado para a implementação do sistema de visão estéreo, recorrendo, tal como foi dito anteriormente, a duas *webcams* disponíveis no LAR. De realçar o principal foco de interesse ser o uso de radiais de pesquisa para a deteção dos objetos.

5.1 Software adotado

5.1.1 Sistema Operativo

Para a realização deste projeto foi necessário recorrer a várias ferramentas de desenvolvimento. Assim, e começando por mencionar o sistema operativo, foi usado o *Linux Ubuntu 12.04 LTS*, Figura 38.



Figura 38 - Logotipo *Ubuntu* [21]

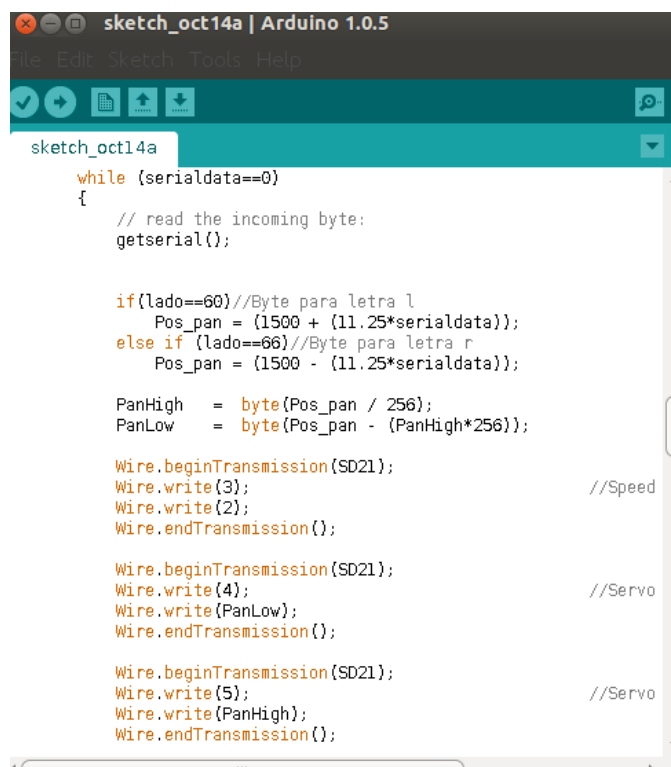
5.1.2 Plataforma de desenvolvimento

No que diz respeito aos algoritmos desenvolvidos, foram utilizadas dois softwares distintos, o *Qt Creator 4.6* (Figura 39) e o *Arduino IDE* (Figura 40). O primeiro tem como grande vantagem o facto de ser um *software* multiplataforma e foi usado para a aquisição das imagens das *webcams* e para o reconhecimento dos objetos, sempre sob a linguagem de programação C++.



Figura 39 - Logotipo *Qt Creator* [22]

Para comunicar com o *Arduino*, foi utilizado o seu *IDE*, que é uma aplicação multiplataforma escrita em *Java*. Aqui foi desenvolvido todo o código para o controlo dos servomotores da plataforma e que recebe a informação oriunda do *Qt Creator* para o correto movimento dos mesmos.



```
sketch_oct14a
while (serialdata==0)
{
  // read the incoming byte:
  getserial();

  if(lado==60)//Byte para letra l
    Pos_pan = (1500 + (11.25*serialdata));
  else if (lado==66)//Byte para letra r
    Pos_pan = (1500 - (11.25*serialdata));

  PanHigh = byte(Pos_pan / 256);
  PanLow = byte(Pos_pan - (PanHigh*256));

  Wire.beginTransmission(SD21);
  Wire.write(3); //Speed
  Wire.write(2);
  Wire.endTransmission();

  Wire.beginTransmission(SD21);
  Wire.write(4); //Servo
  Wire.write(PanLow);
  Wire.endTransmission();

  Wire.beginTransmission(SD21);
  Wire.write(5); //Servo
  Wire.write(PanHigh);
  Wire.endTransmission();
}
```

Figura 40 - Processing

O algoritmo representado na Figura 40, representa o controlo da posição e da velocidade do servomotor *pan*. O servo move-se para esquerda ou para a direita, dependendo da posição em que se encontra o alvo, neste caso sempre a velocidade definida com o valor 2.

5.1.3 Biblioteca

Para implementar a técnica de radiais de pesquisa foi necessário recorrer à biblioteca multiplataforma *OpenCV* [23] (*Open Source Computer Vision Library*) (Figura 41). Esta biblioteca, usada para o desenvolvimento de aplicações na área de Visão por Computador, possui módulos de processamento de imagem e permite, por exemplo, realizar o reconhecimento de objetos, calibração de câmaras e filtros de imagem.



Figura 41 - Logotipo *OpenCV* [23]

5.2 Webcams

As câmaras utilizadas para formar o sistema de visão estéreo são duas webcams *Philips SPC900NC*, iguais à da Figura 42.



Figura 42 - Câmara *Philips SPC900NC* [24]

Na Tabela 2 encontram-se as suas principais características.

Tabela 2 - Características *Webcams*

Câmara <i>Philips SPC900NC</i>	
Resolução	1.3 MP
<i>Frame rate</i> máximo	90 fps
Sensor	CCD
Peso	115g
Tamanho (W x H x D)	74x142x71 (mm)

Na construção da estrutura eletrónica/mecânica que serve de base para este projeto de dissertação, houve a necessidade de ter em especial atenção a estabilidade física do suporte das câmaras, devido à elevada importância deste parâmetro quando se trabalha com sistemas de visão estéreo. Ora com este fator a ter em conta, o suporte construído para as duas *webcams* é feito em metal, o que desde logo assegura uma grande estabilidade, uma vez que suporta, não só as duas câmaras, mas também dois

servomotores ligados diretamente às mesmas, e que permitem realizar o seu movimento mantendo sempre a sua posição relativa (Figura 43).

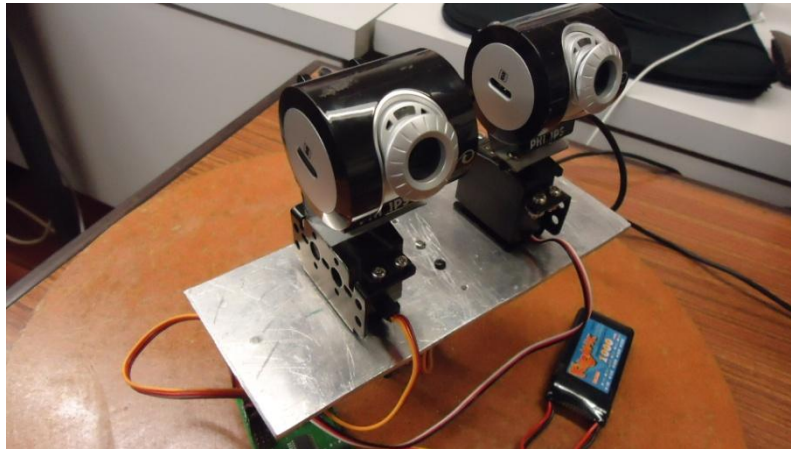


Figura 43 - Suporte metálico das webcams

5.2.1 Posicionamento das *webcams*

Após a captura das imagens por parte das webcams é necessário verificar se estas estão corretamente posicionadas, isto é, se se encontram perfeitamente alinhadas uma com a outra. Esta ação foi conseguida ajustando manualmente, uma única vez aquando da construção da estrutura, ambas as câmaras, após a captura das imagens, para que fiquem o mais alinhadas possível. De seguida apresentam-se duas situações, uma em que estão desalinhadas, Figura 44, e uma outra em que se atingiu o alinhamento possível, Figura 45.

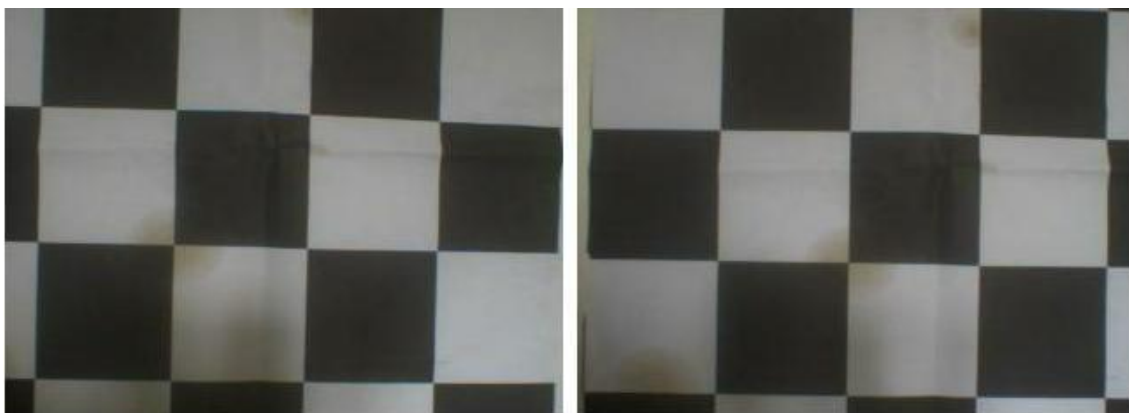


Figura 44 - Imagens desalinhadas



Figura 45 - Imagens alinhadas

5.3 Processamento de imagem

5.3.1 Reconhecimento de um objeto

Para a captura das imagens por partes das *webcams* recorreu-se a funções da biblioteca livre OpenCV, trabalhando com imagens de dimensões de 320x240 píxeis (Figura 46).

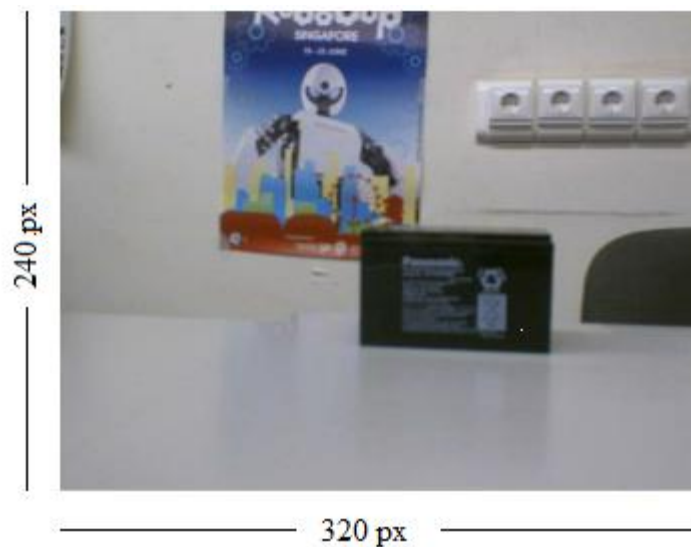


Figura 46 - Tamanho das frames capturadas

Com o intuito de reconhecer um objeto de uma determinada cor pelas *webcams*, houve a transformação das imagens RGB (*Red, Green, Blue*) [25] em HSV (*Hue, Saturation, Value*) [26]. Este sistema de cores, que foi inventado em 1974 por *Alvy Ray Smith* [26], trata-se apenas de uma transformação não linear do sistema de cores RGB e

foi usado para suportar a diferença de luminosidades que pode existir quando as imagens são capturadas. Na Figura 47 encontra-se o sistema de cores HSV sendo visto como um círculo, em que a cada tonalidade de cor corresponde um determinado ângulo. No caso desta aplicação, em que se optou pela segmentação da imagem por cores, tendo interesse pela cor vermelha, o ângulo correspondente encontra-se, após experimentação, entre o 0 e os 20°, e, com desta análise da componente vermelha, é possível isolar o objeto desejado de tudo o resto da imagem capturada.

De seguida é apresentada a equação matemática (9) para a condição de segmentação de cor.

$$P(x, y) = 255 \quad \text{Se } (H > 0 \wedge H < 10) \quad (9)$$

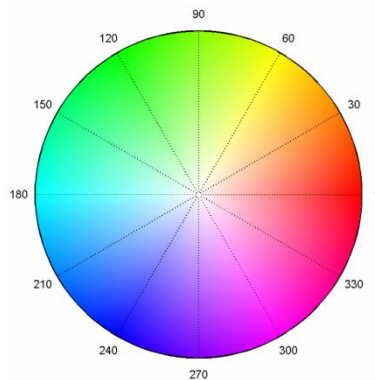


Figura 47 - Sistema de cores HSV [27]

Na Figura 48 encontra-se o objeto utilizado ao longo desta dissertação para ser reconhecido pelo sistema de visão estéreo, de cor vermelha, com a forma redonda e com 5cm de diâmetro.



Figura 48 - Objeto redondo utilizado

5.3.2 Pesquisa por radiais

Estando definida a condição para o objeto ser localizado e isolado da restante imagem, foi implementado um novo método de pesquisa baseado em radiais que percorrem a imagem capturada pelas câmaras. Estes radiais procuram píxeis que obedecam à condição anteriormente imposta (cor vermelha) e, caso seja verificada, desenha na imagem processada pontos brancos correspondentes a esses píxeis. Esta pesquisa tem a vantagem de não analisar todos os seus píxeis da imagem, principalmente os que se encontram nas extremidades da imagem, uma vez que, partindo do centro e do ângulo 0° , estes radiais são incrementados de 2 em 2 píxeis, assim como de 2 em 2 graus, o que permite reduzir o seu tempo de processamento (Figura 49). O facto de esta técnica concentrar o maior número de pontos analisados perto do centro é também uma vantagem, pois é nessa zona que se encontra o principal foco de interesse das imagens, tal como acontece com o sistema de visão humano.

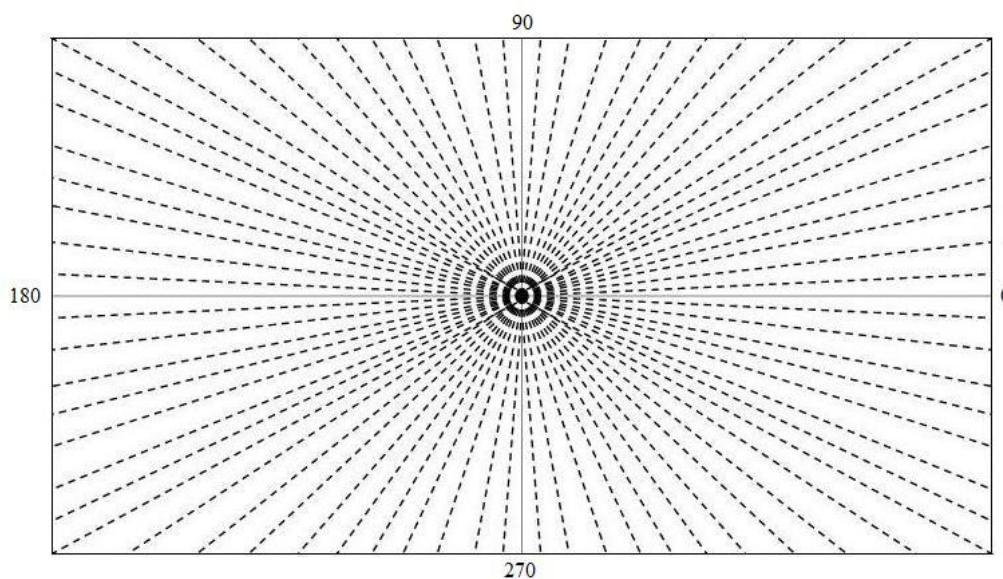


Figura 49 - Radiais de pesquisa

Na Figura 50 está exemplificado uma situação em que o objeto de cor vermelha é isolado do tudo o resto e, através da pesquisa pelos radiais, foram colocados pontos em toda a sua área. A colocação destes pontos tem como objetivo identificar com exatidão a posição em que o objeto se encontra, isto porque, a cada ângulo dos radiais estão associados um determinado número desses pontos.

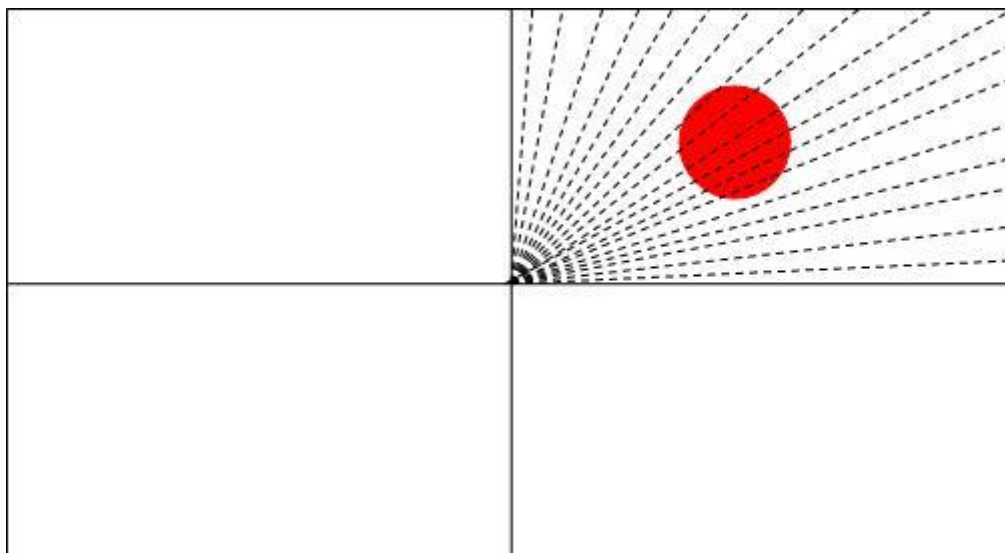


Figura 50 - Exemplo do desenho de pontos no objeto através dos radiais

5.3.2.1 Histogramas

No sentido de agrupar e posteriormente analisar os dados referentes ao número de pontos brancos sobre o objeto, foram criados histogramas para esse efeito, sendo que se trata de uma representação gráfica, em colunas (barras), de uma série de dados que foram previamente divididos em classes. No caso desta aplicação, a base de cada barra do histograma representa o valor dos ângulos dos radiais (0 a 360°) que percorrem a imagem e as barras representam o número de pontos brancos correspondentes a cada radial.

Através da análise do histograma é possível aferir o ângulo do radial que contém o maior número de pontos brancos, que corresponde à barra maior do histograma e, por conseguinte, indica-nos o valor do radial que percorre o centro do objeto, pois é nessa situação em que se atinge o maior número desses pontos.

Na Figura 51 encontra-se o resultado de uma simulação da aplicação, para exemplificar a análise que é feita aos dados. Neste exemplo, o objeto encontra-se no 3° quadrante da imagem e detetado pelos radiais que a percorrem, construindo, dessa forma o histograma e, ao analisa-lo, indica-nos o ângulo do radial do centro do alvo, neste caso com o valor de 216°, como se pode observar pelo *output* da aplicação. O valor deste ângulo será posteriormente tratado para fornecer aos servomotores a informação necessária para executar o seu movimento no sentido de centrar o objeto na câmara.

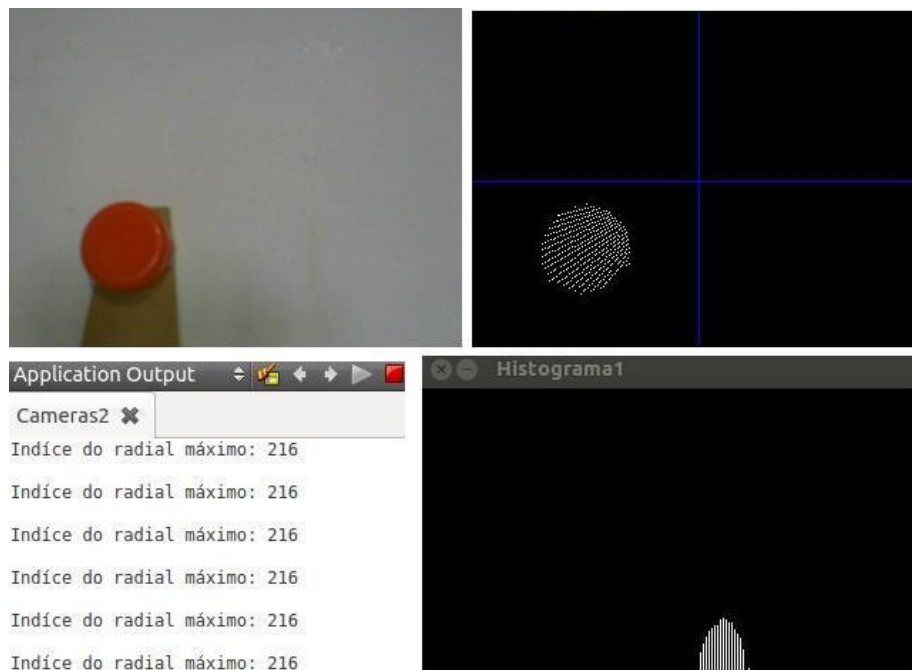


Figura 51 - Uso de histogramas com informação dos radiais

5.3.2.2 Tempo de processamento

A implementação desta nova técnica para detecção e localização de objetos, para que possa ser usada em sistemas de visão computacional, visa sobretudo otimizar o seu tempo de processamento. Quando comparada com outras técnicas existentes a este nível, a conclusão a que se chega é que o tempo necessário para analisar uma imagem das dimensões das que foram utilizadas nesta aplicação (320x240 píxeis) é bastante menor, isto porque o número de píxeis observados pelas técnicas usuais é bastante maior do que no caso de uso de radiais de pesquisa.

Para o efeito, foi incrementado um contador de píxeis analisados, aferindo, através do *output* da aplicação (Figura 52), que a cada volta de 360° que os radiais percorrem, são analisados 14028 píxeis da imagem. Se tomarmos em conta que, analisando cada imagem píxel a píxel pelos métodos convencionais, a equação 10 indica o número de píxeis que são analisados. Por fim, na equação 11 encontra-se a relação entre os valores de píxeis que são analisados em ambas as situações.

$$320 * 240 = 76800 \text{ píxeis} \quad (10)$$

$$\frac{76800}{14028} \approx 5,5 \quad (11)$$

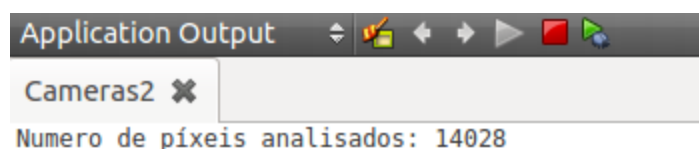


Figura 52 - Output com número de pixels analisados

Nos casos de imagens com estas dimensões, o número total de pixels analisados é de 76800, e comparando com o valor deste novo método, pode afirmar-se que o seu tempo de processamento chega a ser reduzido em cerca 5,5 vezes, um fator determinante no que diz respeito a sistemas de visão por computador.

5.4 Localização do objeto na imagem

Com o intuito de focar o objeto no centro das câmaras surge a necessidade de identificar qual a posição que o mesmo ocupa no espaço, para que a posição dos servomotores seja ajustada adequadamente. Este processo é conseguido por intermédio de relações trigonométricas que podem ser retiradas pela análise dos histogramas e, como já foi referido anteriormente, para além de ser possível identificar não só o índice do radial central do objeto (representado pela letra α), assim como a distância do raio desse radial onde se inicia a identificação do objeto (r) (Figura 53).

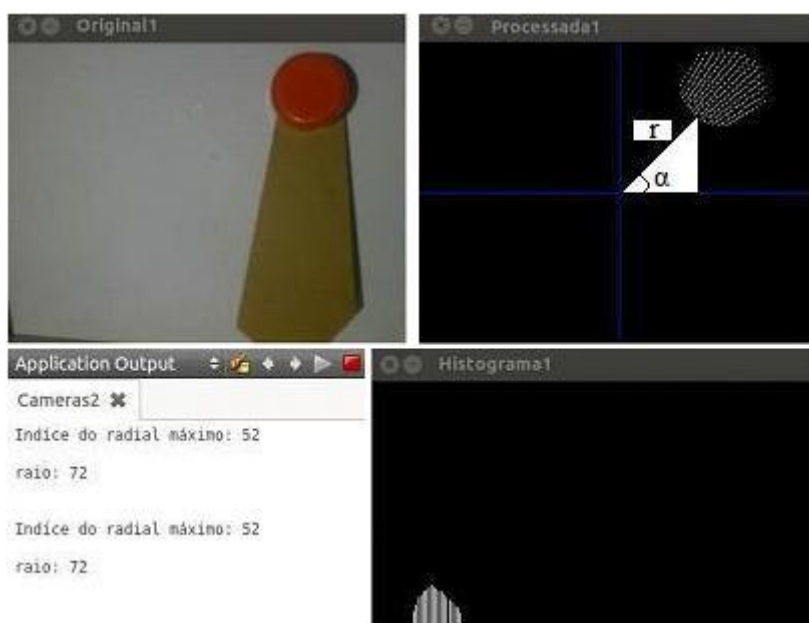


Figura 53 - Relações trigonométricas vista na imagem processada

Com esta informação, pode-se determinar o quão afastado está o objeto do centro, através das distâncias em relação à origem dos eixos x e y . Na Figura 54 encontra-se as relações trigonométricas, em que α representa o valor do radial central, r representa o valor do raio, x a distância do objeto em relação ao eixo do Ox e y indica a distância em relação ao eixo Oy , e nas equações 12 e 13 estão expressas as fórmulas de cálculo dessas distâncias.

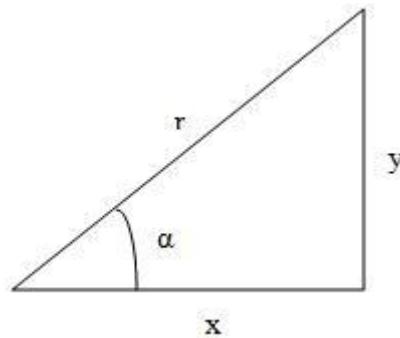


Figura 54 - Relações trigonométricas

$$\text{sen}(\alpha) = \frac{y}{r} \Leftrightarrow y = \text{sen}(\alpha) * r \quad (12a)$$

$$\text{cos}(\alpha) = \frac{x}{r} \Leftrightarrow x = \text{cos}(\alpha) * r \quad (13a)$$

Seguindo os valores que a Figura 53 nos fornece para exemplificar, deduz-se que:

$$y = \text{sen}(52) * 72 \Leftrightarrow y = 56,7 \quad (12b)$$

$$x = \text{cos}(52) * 72 \Leftrightarrow x = 44,3 \quad (13b)$$

Uma vez finalizados estes cálculos, o passo seguinte baseia-se em fornecer esta informação aos servomotores, através de comunicação série entre o *QtCreator* e o *Arduino*, que converte estes dados em ângulos de rotação, iniciando, de seguida, os seus movimentos para centrar o objeto. Na Figura 55, estão ilustradas algumas fases desse processo até o objetivo ter sido atingido com sucesso.

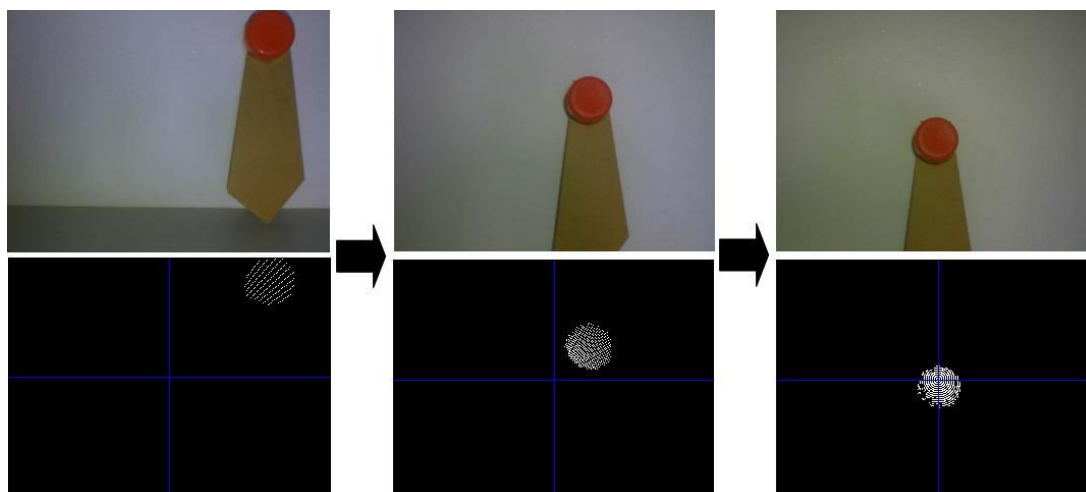


Figura 55 - Etapas da focalização do objeto

De realçar apenas o histograma que é criado quando o objeto se encontra focado na câmara, uma vez que, todos os ângulos dos radiais de pesquisa detetam pontos vermelhos, seria de esperar que todos eles criassem barras no histograma, o que veio a confirmar-se (Figura 56). Esta é a condição que é enviada para os servomotores para cessarem o seu movimento, isto porque o objetivo foi atingido, e só voltam a movimentar-se caso esta condição se deixe de verificar, ou seja, caso o objeto se mova de posição.

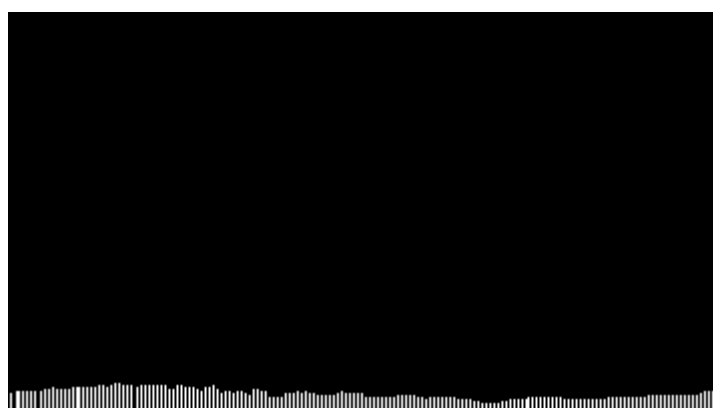


Figura 56 - Histograma obtido com alvo no centro

Todo este processo de localização do objeto na imagem de uma câmara e posterior fornecimento de informação aos servomotores, é analogamente realizado também para outra câmara. Isto prende-se com o objetivo do alvo ser focalizado não apenas numa das delas, mas sim nas duas em simultâneo. A posição em que o objeto se

encontra no espaço terá uma grande influência no ângulo de rotação dos servomotores que estão acoplados às câmaras. Na Figura 57 encontra-se o esquema que representa essa influência, uma vez que se verifica que quando o objeto se encontra mais afastado das webcams, o ângulo de rotação da câmara da direita será menor (α_2), e, pelo contrário, quanto mais próximo estiver, maior será esse ângulo (α_1).

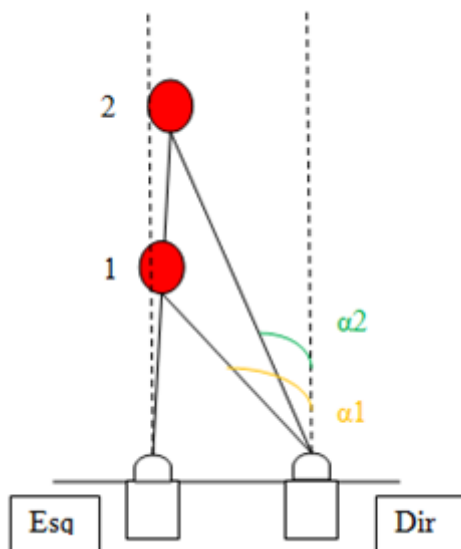


Figura 57 - Distância do objeto às câmaras influencia o ângulo de rotação

Este tipo de situações provocaria que uma câmara (a da direita neste caso) viesse a estar mais convergida do que a outra, não sendo essa a melhor forma de se resolver estes casos. A solução passaria, então, pelo ajuste da posição de toda a estrutura, nomeadamente do módulo *pan-tilt*, num caso semelhante a este, rodando-a para a esquerda para que o objeto se mova para o centro da estrutura. Com isso, as duas câmaras executariam movimentos contrários, isto é, a da esquerda teria de convergir e a da direita de divergir em relação às situações que se encontram expostas na Figura 57.

Através da obtenção dos ângulos de rotação de cada câmara, é possível calcular a distância real do objeto à estrutura, através da técnica de triangulação. Como se pode depreender, a cada posição do objeto corresponde um determinado ângulo, e linearizando esses dados seria-se capaz de se construir um gráfico/tabela contendo esses dados, que seriam obtidos após sucessivas análises experimentais.

Para finalizar, poder-se-ia aplicar os passos típicos que se encontram inerentes a sistemas de visão estéreo, nomeadamente, os processos de calibração estéreo, que determina a relação geométrica entre as duas câmaras no espaço, de retificação estéreo, permitiria que um par de imagens seja retificado, isto é, alinhasse as imagens obtidas por linhas. Por fim aplicar-se-ia também a correspondência estéreo, que permitiria encontrar as correspondências e as disparidades entre pontos nas imagens. Todos estes processos seriam naturalmente aplicados neste sistema, uma vez que, a biblioteca *OpenCV* que é usada neste projeto, incorpora funções que facilitam as suas execuções.

5.5 Algoritmo de processamento de imagem

Para demonstrar de uma forma mais simplificada o método implementado para o processamento de imagem, encontra-se na Figura 58 um fluxograma com contém todas as fases do processo desenvolvido, desde a fase inicial até ao objetivo final.

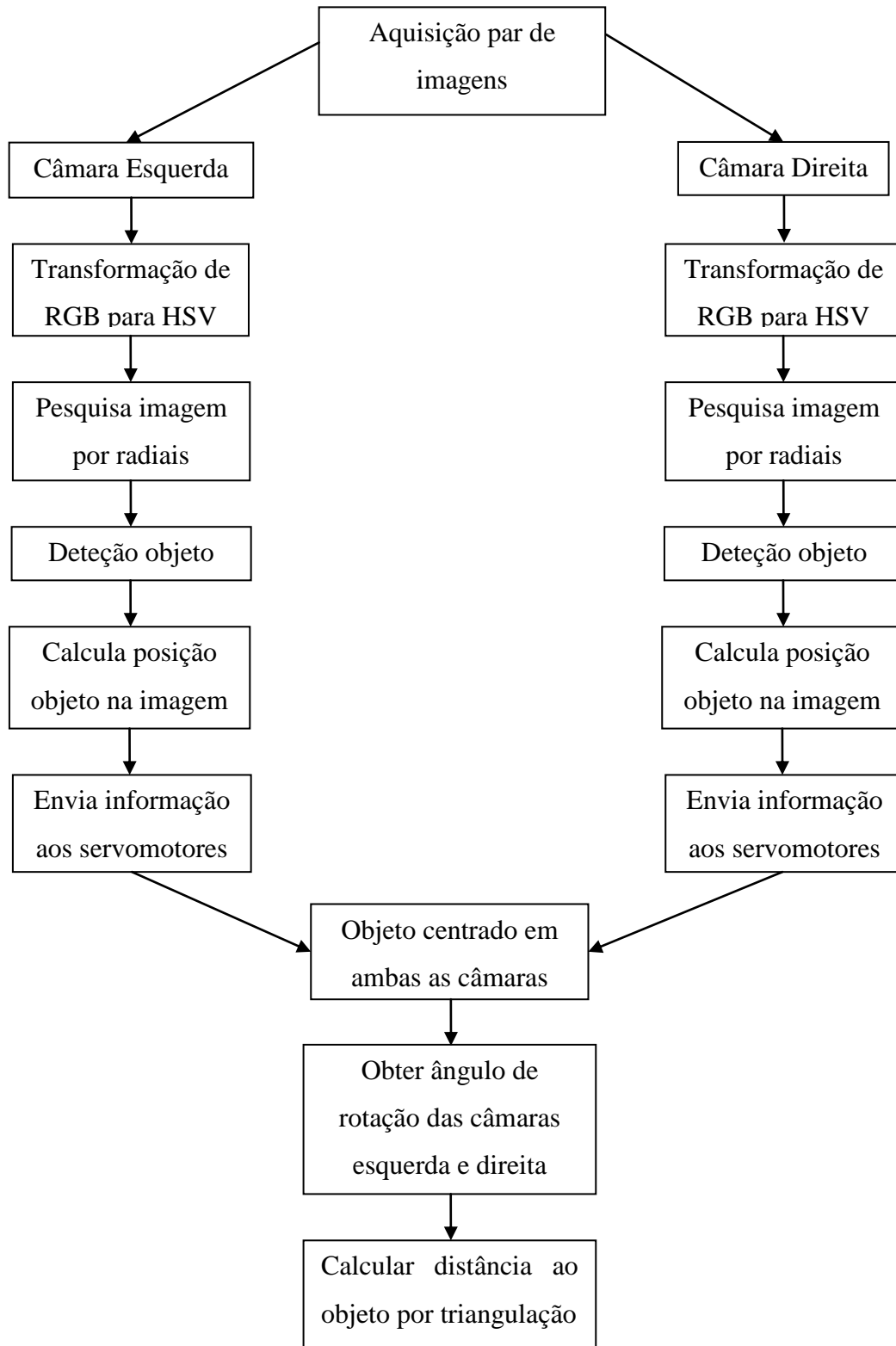


Figura 58 - Algoritmo de processamento de imagem

Capítulo VI

6. Conclusão e trabalho futuro

Finalizado o trabalho efetuado ao longo deste projeto, é altura de avaliar todos os resultados que foram obtidos. Primeiramente, e após conhecimento dos objetivos, foram analisados os componentes que estavam disponíveis no LAR para tentar obter o melhor rendimento possível, evitando custos de aquisição com material e isso foi conseguido, uma vez que grande parte deste se encontrava em bom estado de utilização, sendo possível adaptá-lo para a construção da plataforma.

Em relação ao controlo dos servomotores, a principal novidade deste sistema prende-se com a introdução de dois novos motores acoplados em cada câmara, que as dotam de livre movimento. Esta novidade neste tipo de sistemas introduz um novo conceito em sistemas de visão estereo, visto que, ao invés do habitual, em que as câmaras se encontram fixas a uma estrutura, estas possuem movimento próprio.

No que diz respeito ao método desenvolvido para deteção de objetos nas imagens, pode afirmar-se que, pelos resultados que foram obtidos em testes experimentais, este tipo de pesquisa por radiais é viável, uma vez que, a localização dos objetos na imagem foi conseguida com precisão através da análise feita por este método. Para além disso, também torna o processo de reconhecimento de objetos mais rápido, um importante passo neste tipo de sistemas, pois caso os resultados não indicassem esse caminho, não faria sentido continuar a apostar neste método.

Infelizmente não foi possível cumprir todos os objetivos que foram delineados no início deste projeto, notando que, ainda assim, cerca 80% deles foram atingidos com sucesso, entre os quais os passos de focalização de objeto numas das câmaras através dos dados obtidos pelos radiais que percorrem a imagem. Mesmo não tendo sido colocado em prática os últimos passos, foi feita uma análise teórica ao que deveria ser feito, identificando todos as etapas que a ser seguidas para conclusão deste projeto.

Assim, e perspetivando algum do trabalho que pode dar seguimento a este tema, são sugeridas algumas tarefas, nomeadamente:

- Focalização do objeto na outra câmara, que implica que haja um ajuste da posição do módulo *pan-til* para que uma das câmaras não fique mais convergida do que a outra;
- Obtenção dos ângulos de rotação dos servomotores que controlam a posição das câmaras, para efeitos de cálculo de distância ao objeto, através de triangulação;
- Aplicação de calibração e retificação estéreo, obtenção de mapa de disparidades para encontrar correspondências entre pontos de imagens obtidas em ângulos diferentes.
- Desenvolver uma aplicação para que o sistema possa detetar outros objetos com cores diferentes, sem ser necessário alterar a condição de segmentação de cor imposta para este projeto.

Referências

- [1] “Wikipedia,” [Online]. Available: <http://pt.wikipedia.org/wiki/Estereoscopia>. [Acedido em 10 2 2014].
- [2] P. G. TV, “youtube,” 09 08 2010. [Online]. <https://www.youtube.com/watch?v=Ckd1KJZreVE#t=10>. [Acedido em 7 03 2013].
- [3] R. M. G. d. Silva, “Design e construção de um robo antropomórfico. Tese de Mestrado,” Universidade do Minho, 2008.
- [4] T. S. H. E. M. I. Massaki Shibata, “Visual Tracking Control to Fast Moving Target for Stereo Vision Robot,” em *The 11th IEEE International Workshop on Advanced Motion Control*, Nagaoka, Japan, 2010.
- [5] A. Kirillov, “AForge.net,” 17 01 2009. [Online]. http://www.aforgenet.com/articles/step_to_stereo_vision/. [Acedido em 12 04 2013].
- [6] “Vision 3D,” [Online]. Available: <http://www.vision3d.com/stereo.html>. [Acedido em 15 02 2013].
- [7] G. B. a. A. Kaehler, *Learning OpenCV - Computer Vision with the OpenCV Library*, O'REILLY, 2008.
- [8] J. A. d. C. P. Gaspar, “Visão para Robótica Móvel: Detecção de Obstáculos sobre Pavimento Plano,” 1994.
- [9] L. Iocchi, “Stereo Vision: Triangulation,” 6 4 1998. [Online]. <http://www.dis.uniroma1.it/~iocchi/stereo/triang.html>. [Acedido em 5 5 2013].
- [10] A. Fusiello, “Rectification,” 2 1998. [Online]. http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/FUSIELLO/node5.html. [Acedido em 20 6 2013].
- [11] D. G. Lowe, “Object recognition from local scale-invariant features.,” em *7th International Conference on Computer Vision*, Corfu, Greece, 1999.
- [12] L. L. d. O. Junior, “Filtros Compostos e Adaptativos”.
- [13] T. T. L. V. G. Herbert Bay, “SURF: Speeded Up Robust Features,” Zurich.
- [14] “SURF,” [Online]. <http://en.wikipedia.org/wiki/SURF>.

- [15] “Mobilerobots,” 2010. [Online]. <http://www.mobilerobots.com/Accessories/MobileRanger.aspx>.
- [16] “Point Grey,” [Online]. <http://www.ptgrey.com//bumblebee2-firewire-stereo-vision-camera-systems>.
- [17] “BotnRoll,” [Online]. <http://www.botnroll.com>.
- [18] “Arduino,” [Online]. <http://www.arduino.cc/>.
- [19] “Robot Eletronics,” [Online]. <http://www.robot-electronics.co.uk/htm/sd21tech.htm>.
- [20] J. S. A. d. S. Segundo, “Um Método para Determinação da Profundidade Combinando Visão Estéreo e Autocalibração para Aplicação em Robótica Móvel,” Natal, Brasil, 2007.
- [21] “Ubuntu,” [Online]. Available: <http://www.ubuntu.com/>.
- [22] “QT Creator| Qt Project,” [Online]. <https://qt-project.org/search/tag/qt~creator>.
- [23] “OpenCV,” [Online]. <http://opencv.org/>.
- [24] “PHILIPS,” [Online]. <http://www.p4c.philips.com/cgi-bin/cpindex.SPC900NCSsoftware>
- [25] “RGB,” [Online]. Available: <http://pt.wikipedia.org/wiki/RGB>.
- [26] “HSV,” [Online]. Available: <http://pt.wikipedia.org/wiki/HSV>.
- [27] “Hobbies, Hobbits, and Hobos,” [Online]. <http://www.hobbitsandhobos.com/2011/06/applying-simple-hsv-object-tracking/>.