

Sistema Visual para Ground Truth de Sistemas Autónomos

NUNO FILIPE BORGES TEIXEIRA NUNES

Novembro de 2012



INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO



Sistema Visual para *Ground Truth* de Sistemas Autónomos

Nuno Filipe Borges Teixeira Nunes

Tese submetida no Âmbito do
Mestrado em Engenharia Electrotécnica e de Computadores

Orientador: José Miguel Soares de Almeida (Mestre)

Novembro de 2012

Resumo

Este trabalho visa contribuir para o desenvolvimento de um sistema de visão multi-câmara para determinação da localização, atitude e seguimento de múltiplos objectos, para ser utilizado na unidade de robótica do INESC TEC, e resulta da necessidade de ter informação externa exacta que sirva de referência no estudo, caracterização e desenvolvimento de algoritmos de localização, navegação e controlo de vários sistemas autónomos.

Com base na caracterização dos veículos autónomos existentes na unidade de robótica do INESC TEC e na análise dos seus cenários de operação, foi efectuado o levantamento de requisitos para o sistema a desenvolver.

Foram estudados os fundamentos teóricos, necessários ao desenvolvimento do sistema, em temas relacionados com visão computacional, métodos de estimação e associação de dados para problemas de seguimento de múltiplos objectos .

Foi proposta uma arquitectura para o sistema global que endereça os vários requisitos identificados, permitindo a utilização de múltiplas câmaras e suportando o seguimento de múltiplos objectos, com ou sem marcadores.

Foram implementados e validados componentes da arquitectura proposta e integrados num sistema para validação, focando na localização e seguimento de múltiplos objectos com marcadores luminosos à base de *Light-Emitting Diodes* (LEDs). Nomeadamente, os módulos para a identificação dos pontos de interesse na imagem, técnicas para agrupar os vários pontos de interesse de cada objecto e efectuar a correspondência das medidas obtidas pelas várias câmaras, método para a determinação da posição e atitude dos objectos, filtro para seguimento de múltiplos objectos.

Foram realizados testes para validação e afinação do sistema implementado que demonstram que a solução encontrada vai de encontro aos requisitos, e foram identificadas as linhas de trabalho para a continuação do desenvolvimento do sistema global.

Palavras-chave

Ground truth, visão computacional, filtro de *Kalman*, associação de dados sistemas autónomos, robótica móvel.

Abstract

This work aims to contribute to the development of an multi-camera vision based system to determine the location, attitude and tracking of multiple objects, to be used in the robotics unit of the INESC TEC and arises from the need to have accurate external information to serve as reference in the study, characterization and development of localization, navigation and control algorithms of various autonomous systems.

Based on characterization of the vehicles existing in the robotics unit of the INESC TEC and analysis of their operation scenarios, was made a requirements list for the system to develop.

Were studied the theoretical fundamentals required to the development of the system, of themes related with computational vision, estimation methods and data association for problems of multiple objects tracking.

It was proposed an architecture for the global system that addresses the various identified requirements, allowing the usage of multiple cameras and supporting the tracking of multiple objects, with or without markers.

Were implemented and validated components of the proposed architecture and integrated in a system validation, focusing on the location and tracking of multiple objects with luminous markers based on LEDs. Namely the modules for the identification of points of interest in the image, techniques to group the various points of interest of each object and perform the correspondence of the obtained measurements by the various cameras, method to determine the position and attitude of the objects, filter for tracking multiple objects.

Tests were performed to the validation and tuning of the implemented system which demonstrate that the solution found meets the requirements, and were identified the guide lines for the continuation of the development of the global system.

Keywords

Ground truth, computer vision, *Kalman* filter , data association autonomous systems, mobile robotics.

Agradecimentos

Aproveito para expressar os meus agradecimentos a todas as pessoas que me proporcionaram as condições necessárias à realização deste trabalho. Deste grupo destaco os meus pais e amigos próximos, que sempre me apoiaram. Manifesto a minha gratidão pela ajuda e paciência dispensada pelos membros do Laboratório de Sistemas Autónomos, em especial para o meu orientador, Eng. José Miguel e ao Eng. André Dias que sempre se mostraram disponíveis e me acompanharam ao longo de todo o trabalho.

Nuno Filipe Borges Teixeira Nunes

Conteúdo

Resumo	i
Abstract	iii
Agradecimentos	v
Lista de Figuras	x
Lista de Tabelas	xiii
Lista de Símbolos	xv
Acrónimos	xvii
1 Introdução	1
1.1 Aplicações	2
1.1.1 Ciências da vida	2
1.1.2 Animação	2
1.1.3 Engenharia	3
1.2 Objectivos	3
1.3 Organização da tese	4
2 Estado da Arte	5
2.1 Aplicações e sensores usados nos sistemas de <i>ground truth</i>	5
2.2 Sistemas comerciais	9
2.3 Conclusões	10
3 Caracterização do Problema	11
3.1 Estudo dos sistemas autónomos e cenários de actuação	11
3.1.1 ISePorto	11
3.1.2 Quadrotor	12
3.2 Requisitos do sistema	13

4	Fundamentos de Visão Computacional	15
4.1	Extracção de <i>features</i>	15
4.2	Projecção Perspectiva	16
4.3	Distorção provocada pela lente	19
4.4	Visão <i>Stereo</i>	20
4.4.1	Geometria epipolar	20
4.4.2	Matriz de homografia	22
4.4.3	Rectificação <i>stereo</i>	23
4.4.4	Triangulação	24
5	Métodos de estimação	27
5.1	Filtro de <i>Kalman</i>	27
5.1.1	Modelos de estado e sensor	28
5.2	Filtro de <i>Kalman</i> discreto	29
5.2.1	Modelos do sistema e observação	29
5.2.2	Previsão	30
5.2.3	<i>Update</i>	30
5.3	Fusão de dados multi-sensorial	31
5.3.1	Filtro de <i>Kalman</i> multi-sensorial	31
5.3.2	Associação de dados multi-sensorial	35
6	Implementação	39
6.1	Arquitectura do sistema	39
6.2	Calibração de parâmetros intrínsecos e extrínsecos das câmaras	40
6.2.1	Calibração dos parâmetros intrínsecos das câmaras	41
6.2.2	Calibração dos parâmetros extrínsecos	43
6.2.3	Erro associado ao <i>stereo</i>	45
6.3	Captura de imagens	46
6.3.1	Eliminação dos falsos pontos de interesse	48
6.4	Aplicação do filtro de <i>Kalman</i>	51
6.4.1	Inicializações	52
6.4.2	Previsão	53
6.4.3	<i>Update</i>	54
7	Resultados	57
7.1	Resultados	57
7.2	Objecto estático	58
7.3	Objecto em movimento	60
8	Conclusões e Trabalho Futuro	65

<i>CONTEÚDO</i>	ix
Bibliografia	69
A Calibração de extrínsecos	71
A.1 Correspondência de medidas	72

Lista de Figuras

2.1	Robô utilizado no projecto Rawseeds	6
2.2	Distribuição de marcadores num robô <i>humanoid</i> NAO	7
2.3	Passos intermédios no processamento de imagem	8
2.4	Câmara da Vicon	9
2.5	Câmara V100:R2 da OptiTrack	10
3.1	Cenário de um jogo MSL	12
3.2	Quadrotor	13
4.1	Modelo <i>pinhole</i>	16
4.2	Modelo <i>pinhole</i> com o plano da imagem à frente do centro óptico	17
4.3	Projectção de rectas do mundo na imagem dependendo do tipo de distorção causada pela lente	19
4.4	Geometria epipolar <i>stereo</i>	21
4.5	Rectificação de um par de imagens <i>stereo</i> e representação das linhas epipolares.	23
4.6	Geometria após rectificação – epipolos no infinito e planos da imagem paralelos.	24
5.1	Diagrama de blocos do ciclo do filtro de <i>Kalman</i>	29
5.2	Estrutura de uma arquitectura de fusão de dados centralizada	32
5.3	Região de validação	35
5.4	<i>Nearest-Neighbour Standard Filter</i> com apenas um <i>target track</i>	36
5.5	<i>Nearest-Neighbour Standard Filter</i> com vários <i>targets</i>	37
5.6	<i>Probabilistic Data Association Filter</i>	37
5.7	<i>Multiple Hypothesis Tracking Filter</i>	38
6.1	Arquitectura do sistema	40
6.2	Transformações dos sistemas de coordenadas durante uma reconstrução 3D a partir de coordenadas em pixels	41
6.3	Erro de projectção x, y da câmara direita, em pixels	42
6.4	Erro de projectção x, y da câmara esquerda, em pixels	43
6.5	Comparação entre as imagens com distorção e sem distorção	44
6.6	Pontos escolhidos para o cálculo dos parâmetros extrínsecos e disposição das câmaras	44

6.7	Erro das medidas 3D obtidas nas diferentes zonas do campo	46
6.8	Fluxograma do procedimento para a obtenção das coordenadas x, y em pixels dos pontos de interesse	47
6.9	Exemplo de uma imagem capturada por uma câmara, em HSV	47
6.10	Exemplo de criação de <i>clusters</i> , suas identificações e cálculo do ponto central	48
6.11	Traçado das linhas epipolares na câmara 2, baseadas nas posições dos pontos centrais dos <i>clusters</i> da câmara 1	48
6.12	Cálculo posição 3D dos pontos centrais dos <i>clusters</i> para verificar se se situam dentro do volume válido	49
6.13	Identificação dos pontos do marcador	50
6.14	Diferentes casos de associação de observações através da região de validação	52
7.1	Câmaras usadas no trabalho	57
7.2	Protótipo do marcador	58
7.3	Posição dos LEDs do marcador segundo os eixos x, y , em modo estático	58
7.4	Posição dos LEDs do marcador segundo os eixos y, z , em modo estático	59
7.5	Variação do <i>Roll Pitch Yaw</i> do objecto, em modo estático	59
7.6	Posicionamento do marcador no robô	60
7.7	Posição do centro do objecto ao longo de uma trajectória segundo os eixos x, y	61
7.8	Trajectória estimada pelo filtro	61
7.9	Variação do <i>Roll Pitch Yaw</i> do objecto	62
7.10	Inovação	62
7.11	Colisão entre dois robôs	63
7.12	Trajectória de quatro robôs - Identificação de uma oclusão	63

Lista de Tabelas

2.1	Comparação entre sistemas comerciais de <i>ground truth</i> por visão	10
6.1	Correspondência entre os <i>clusters</i> das várias câmaras	49
A.1	Medidas extrínsecos	72

Lista de Símbolos

$\mathbf{B}(k)$	Matriz de transição de entrada
$\mathbf{B}(t)$	Matriz de entrada
b	<i>Baseline</i>
cc	Centro da imagem
$c.o.$	Centro óptico
$\mathbf{D}(t)$	Matriz de ruído na observação
E	Matriz essencial
e_l	Epipolo na imagem da esquerda
e_r	Epipolo na imagem da direita
F	Matriz fundamental
f	Distância focal
f_x	Distância focal segundo a componente horizontal
f_y	Distância focal segundo a componente vertical
\mathbf{F}	Matriz de estado
$\mathbf{G}(k)$	Matriz de ruído de transição
$\mathbf{G}(t)$	Matriz de ruído
\mathbf{H}	Matriz de observação
I	Imagem
κ	Parâmetro de distorção da lente
K	Matriz dos parâmetros intrínsecos
kc	Factor de correlação
k_u	Factor de escala horizontal para conversão entre o sistema referencial métrico da imagem e o sistema referencial em pixels da imagem
k_v	Factor de escala vertical para conversão entre o sistema referencial métrico da imagem e o sistema referencial em pixels da imagem
P_c	Ponto de projecção de P_w no referencial da câmara
P_i	Ponto no sistema referencial métrico da imagem
p_l	Ponto de projecção no plano da imagem da esquerda
p_r	Ponto de projecção no plano da imagem da direita
$P_{u,v}$	Ponto no referencial em pixels da imagem

P_w	Ponto tridimensional no referencial global
$\mathbf{P}(k k - 1)$	Covariância prevista
R	Matriz de rotação
R^c	Plano do centro óptico
R_c	Referencial da câmara
R^i	Plano da imagem
R_i	Referencial da imagem
R_{ext}	Rotação entre o referencial do mundo e o da câmara
σ	Desvio padrão
t	Vector de translação
T_{ext}	Translação entre o referencial do mundo e o da câmara
u	Coordenada horizontal na imagem
u_0	Coordenada horizontal no referencial em pixels da imagem onde se projecta o centro óptico
$\mathbf{u}(k)$	Vector de entrada
$\nu(k)$	Inovação
v	Coordenada vertical na imagem
v	Coordenada vertical na imagem
v_0	Coordenada vertical no referencial em pixels da imagem onde se projecta o centro óptico
$\mathbf{v}(k)$	Adição de ruído ao movimento
$\mathbf{v}(t)$	Incerteza na evolução do estado
\mathbf{w}	Ruído existente na observação
x^c	Coordenada x no referencial da câmara
x^d	Coordenada x afectada pela distorção da lente
x^i	Coordenada x no referencial métrico da imagem
x^u	Coordenada x ignorando a distorção provocada pela lente
x^w	Coordenada x no referencial do mundo
$\mathbf{x}(k)$	Estado no instante k
$\mathbf{x}(t)$	Vector de estado
$\hat{\mathbf{x}}(k k - 1)$	Estado previsto
y^c	Coordenada y no referencial da câmara
y^d	Coordenada y afectada pela distorção da lente
y^i	Coordenada y no referencial métrico da imagem
y^u	Coordenada y ignorando a distorção provocada pela lente
y^w	Coordenada y no referencial do mundo
z^c	Coordenada z no referencial da câmara
z^w	Coordenada z no referencial do mundo
$\mathbf{z}(k)$	Observação feita no instante k
$\mathbf{z}(t)$	Vector de observação

Acrónimos

API	<i>Application Programming Interface</i>
DC	Corrente contínua
FPS	<i>Frames Per Second</i>
GigE	<i>Gigabit Ethernet</i>
GPS	<i>Global Positioning System</i>
HSV	<i>Hue, Saturation, Value</i>
ID	identificação
IMU	<i>Inertial Measurement Unit</i>
INS	<i>Inertial Navigation System</i>
JPDF	<i>Joint Probabilistic Data Association Filter</i>
KF	<i>Kalman Filter</i>
LED	<i>Light-Emitting Diode</i>
LRF	<i>Laser Range Finder</i>
LSA	Laboratório de Sistemas Autónomos
MP	Megapixel
MHTF	<i>Multiple Hypothesis Tracking Filter</i>
MSL	<i>Middle Size League</i>
MTT	<i>Multi-target tracking</i>
NNSF	<i>Nearest-Neighbour Standard Filter</i>
PDAF	<i>Probabilistic Data Association Filter</i>
UAV	<i>Unmanned Aerial Vehicle</i>
USB	<i>Universal Serial Bus</i>

Capítulo 1

Introdução

1.1	Aplicações	2
1.1.1	Ciências da vida	2
1.1.2	Animação	2
1.1.3	Engenharia	3
1.2	Objectivos	3
1.3	Organização da tese	4

Esta tese aborda o desenvolvimento de um sistema de *ground truth*¹ por visão que permite a determinação da localização, atitude e movimento de veículos e objectos.

Ao longo dos últimos anos esta matéria tem despertado interesse no seio da comunidade de robótica móvel, pois permitem obter dados que permitem analisar o funcionamento dos algoritmos de navegação e controlo.

A unidade de robótica INESC TEC possui vários veículos autónomos para os quais uma boa navegação e controlo são essenciais. Um desses casos é a equipa de futebol robótico ISePorto. Para que estes robôs consigam participar nos jogos de futebol é muito importante que se consigam localizar correctamente, para cooperarem em equipa, receberem a bola e saberem para onde a têm que rematar. Durante um jogo os robôs estão frequentemente rodeados por outros robôs que tentam chegar à bola dando origem a oclusões e colisões. Para caracterizar os algoritmos de localização e controlo dos robôs do ISePorto é preciso analisar a trajectória dos mesmos em ambiente de jogo assim como a trajectória da bola.

Outro veículo disponível na unidade de robótica INESC TEC é um quadrotor. O quadrotor é um veículo com elevada manobrabilidade e velocidade de movimentos no plano tridimensional. Neste

¹Sistema externo de localização absoluta, com elevada precisão e exactidão, para determinação de valores de referência ou assumidamente verdadeiras.

caso, é importante ter um sistema externo que indique a localização, atitude e trajetória do veículo para que se possam melhorar os métodos de controlo e até usar essa informação para realimentar a malha de controlo.

Neste contexto, o desenvolvimento de um sistema de *ground truth* por visão permitirá saber a localização de cada robô e da bola nos vários momentos do jogo e assim perceber mais facilmente que comportamento tiveram em várias situações de jogo assim como estudar os movimentos do quadrotor. A utilização de marcadores para identificar os veículos e/ou objectos a seguir é comum neste tipo de sistemas, no entanto, como há a necessidade de seguir objectos onde não podem ser aplicados marcadores, como a bola, pois alterariam a sua dinâmica. O sistema a desenvolver terá que ser capaz de seguir veículos e/ou objectos com ou sem marcadores.

Estes sistemas podem ser usados em diversas aplicações.

1.1 Aplicações

Os sistemas de medição de movimento por visão baseada em marcadores, devido à sua versatilidade, têm vindo a ser utilizadas em diferentes áreas nomeadamente no estudo das ciências da vida, na criação de animações e no desenvolvimento de vários sistemas de engenharia.

1.1.1 Ciências da vida

Os sistemas de medição de movimento por visão baseada em marcadores, no que diz respeito ao estudo das ciências da vida, podem ser utilizados no auxílio à recuperação da mobilidade [6], na análise da postura e equilíbrio, na análise do movimento efectuado pelos desportistas e no estudo dos movimentos e padrões comportamentais dos animais [7, 8, 9].

Ao aplicar estes sistemas em centros clínicos dedicados à reabilitação das funções motoras, é possível através da análise de vários pontos do corpo, ao longo de por exemplo uma caminhada, determinar se o paciente está a evoluir favoravelmente e determinar de forma mais precisa que tratamentos se enquadram melhor para se conseguir uma recuperação mais rápida e eficaz com a particularidade de ser um método não-intrusivo.

É também cada vez mais frequente ver este tipo de sistemas a ser utilizado pelos desportistas de alta competição em centros de alto rendimento, para que consigam aperfeiçoar os seus movimentos e se tornem mais competitivos ao mesmo tempo que previnem possíveis lesões resultantes de movimentos incorrectos.

Em [10], é proposto um sistema que visa auxiliar os cirurgiões ortopédicos aquando a cirurgias para correcções de deformidades da coluna.

1.1.2 Animação

Quer seja no desenvolvimento de videojogos quer seja na produção de um filme, os sistemas de medição de movimento por visão baseada em marcadores são cada vez mais utilizados por permitirem, entre outras coisas, criar personagens em computador e dotá-las de movimentos que seriam de grande

complexidade para um computador gerar sem esta tecnologia. Assim, as personagens dos filmes e videojogos, apesar de nem sempre terem o aspecto de um ser humano, são capazes de mostrar emoções através do movimento dos músculos da face e interagir com outras personagens de uma forma perfeitamente natural dando origem a uma experiência cada mais agradável para o espectador [11].

No mundo dos videojogos, actualmente duas das maiores empresas ligadas ao desenvolvimento de consolas e acessórios têm disponíveis sistemas que permitem ao utilizador interagir com diversos jogos fazendo com que uma personagem imite os movimentos corporais da pessoa. Estes sistemas vieram revolucionar a forma como os jogadores passam o seu tempo livre. O que era considerada uma actividade sedentária, passou a ser uma forma de se praticar algum exercício físico de uma forma divertida para toda a família. Estes acessórios, tal como se pode ver em [12] e [13], são sistemas de visão que através de medições tridimensionais efectuadas ao corpo do jogador, interpretam o movimento do corpo humano utilizando um sistema de *tracking* e dependendo do jogo, pode ler apenas uma parte do corpo [14] ou corpo inteiro. Há também casos em que para além da personagem ser controlada totalmente pelo corpo do jogador, o ambiente que o rodeia passa a ser o mundo virtual onde essa personagem se move ao que normalmente se designa por realidade aumentada [15].

1.1.3 Engenharia

Os sistemas de medição de movimento por visão baseada em marcadores estão a ser utilizados em alguns laboratórios relacionados com *Unmanned Aerial Vehicles* (UAVs) e robótica em geral por permitirem uma percepção da trajectória, atitude e interacção com o meio, que não seria possível a olho nu. Ao possuir um destes sistemas, o laboratório é capaz de desenvolver nós de controlo mais eficazes, realizar estudos relacionados com a cooperação entre robôs, entre outros [16].

Alguns laboratórios de hidrodinâmica e locais de testes navais preferem sistemas ópticos para testar os barcos quando sujeitos a diferentes ondas e condições de vento em vez de recorrerem a volumosos e pesados acelerómetros e giroscópios que afectariam o modelo a testar.

A aerodinâmica das asas de um avião também pode ser optimizada recorrendo a este método. Durante a fase de projecto, modelos à escala são testados em túneis de vento para mostrar como deverá ser a estrutura das asas e fuselagem de forma a reduzir o atrito e vibrações. É também testada a aeroelasticidade da estrutura dos aviões quando sujeitas a diferentes condições.

São também utilizados para analisar a interacção dos humanos com diferentes objectos, locais de trabalho e espaços de convivência.

1.2 Objectivos

Este trabalho visa contribuir para o desenvolvimento de um sistema de visão multi-câmara para ser utilizado na unidade de robótica INESC TEC na determinação da localização, atitude e seguimento de múltiplos objectos. O sistema de visão a desenvolver deverá ser capaz de seguir um conjunto de veículos e/ou objectos com marcadores de LEDs e determinar as suas localizações, atitudes e movimentações,

mesmo quando ocorram oclusões ou colisões. Os dados obtidos pelo sistema deverão ser guardados para posterior análise e deverá possibilitar a adição de vários pares de câmaras à medida que sejam necessárias. Pretende-se que o sistema seja de fácil instalação e fácil calibração.

Mais especificamente, este trabalho tem como objectivos:

- Propor uma arquitectura para o sistema global;
- Implementar e validar módulos visuais para percepção tridimensional de objectos com marcadores à base de LEDs;
- Desenvolver estimadores para o seguimento de múltiplos marcadores à base de LEDs em simultâneo;
- Implementar um protótipo do sistema de *ground truth* que permita validar os módulos desenvolvidos;
- Validar experimentalmente o protótipo do sistema desenvolvido.

1.3 Organização da tese

Este relatório está organizado em oito capítulos.

No capítulo seguinte são descritas as aplicações e os sensores usados nos sistemas de *ground truth* e apresentados alguns sistemas comerciais.

No capítulo 3, é efectuado um estudo aos sistemas autónomos que se pretende seguir e são identificados os requisitos para o sistema a desenvolver.

Nos capítulos 4 e 5 são descritos alguns fundamentos de visão computacional e métodos de estimação necessários para uma boa compreensão do trabalho desenvolvido.

Seguidamente, no capítulo 6 é proposta uma solução ao problema e descritos alguns detalhes da sua implementação.

No capítulo 7 são apresentados alguns resultados.

Por fim, no capítulo 8 estão as conclusões tiradas e o trabalho a desenvolver no futuro.

Capítulo 2

Estado da Arte

2.1	Aplicações e sensores usados nos sistemas de <i>ground truth</i>	5
2.2	Sistemas comerciais	9
2.3	Conclusões	10

Neste capítulo pretende-se apresentar alguns trabalhos desenvolvidos relacionados com o seguimento de objectos assim como alguns produtos já desenvolvido para o efeito.

2.1 Aplicações e sensores usados nos sistemas de *ground truth*

Os sistemas de *ground truth* permitem determinar a posição e orientação real de objectos. Essa possibilidade leva a que estes sistemas sejam utilizados maioritariamente na análise da qualidade de sensores e algoritmos de localização e na realimentação de malhas de controlo de veículos autónomos.

Como os sistemas a analisar e os veículos que se pretende realimentar tanto podem estar virados para aplicações em ambiente exterior ou interior e os locais onde são aplicados não têm sempre as mesmas características, os sensores que compõem os sistemas de *ground truth* não são sempre do mesmo tipo. Entre os mais utilizados estão os *Laser Range Finders* (LRFs), os *Global Positioning Systems* (GPSs) e as câmaras.

No que diz respeito a sistemas de *ground truth* ao ar livre podem ser utilizados mapas de referência de elevada exactidão para analisar a qualidade de sensores e algoritmos [17], porque os sinais GPS, em ambiente urbanos, são frequentemente afectados por obstruções e reflexos resultado em medidas imprecisas ou podem até não estar disponíveis. No entanto, nem sempre se tem esses mapas disponíveis. Nessas situações a escolha pode recair por equipar o veículo com para sistemas compostos por GPSs e outro tipo de sensores como por exemplo câmaras, *Inertial Measurement Unit* (IMU) e LRFs [18, 19].

Outras vezes pretendem-se sistemas que funcionem em ambientes interiores. As câmaras são uma

das escolhas possíveis [1, 20] podendo-se utilizar marcadores para auxiliar a detecção e o cálculo da posição ou recorrendo a sistemas de baixo custo e para áreas reduzidas [21] onde se utiliza uma *Kinect* [12]. No primeiro caso, foram colocados um total de seis marcadores fiduciais de forma e tamanho bem definidos, à volta do robô (ver figura 2.1), para que pudessem ser vistos pelas várias câmaras colocadas num plano superior com a ajuda de postes fixos. Os parâmetros intrínsecos e extrínsecos das câmaras são conhecidos. Estes marcadores têm a especificidade de poderem ser decodificados e apresentarem uma identificação (ID) única.

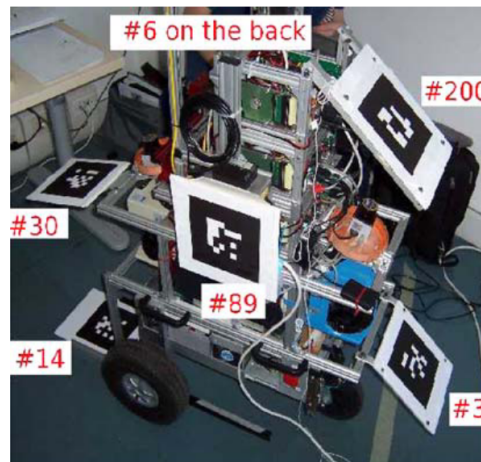


Figura 2.1: Robô utilizado no projecto Rawseeds [1]

Essa ID atribuída a cada marcador serve para que o sistema utilizado distinga cada um dos marcadores e possa calcular a posição do robô, aproveitando o facto de saberem a posição relativa de cada marcador face ao robô e terem os parâmetros extrínsecos das câmaras relativamente ao referencial do mundo. Em alternativa às câmaras usam-se LRFs [1, 20, 22] sendo preciso analisar as diversas formas e objectos existentes no ambiente onde irá estar o veículo a estudar para que não influenciem as leituras relativas veículo em questão.

Há casos em que se quer não só medir a posição e atitude geral de determinado veículo ou objecto, mas sim a posição de vários pontos dele. Para fazer isso, por vezes é recorrente utilizarem-se sistemas de visão baseados em marcadores passivos para identificar os pontos de interesse [2]. Como o robô tem um aspecto que de alguma forma se assemelha com a dos humanos, utilizaram um sistema VICON [4], composto por 15 câmaras, para abranger uma área de $6m \times 4m$, para capturar as posições e orientações das várias partes do corpo, a uma taxa de 120 Hz, assim como a das articulações recorrendo a um modelo esquelético.

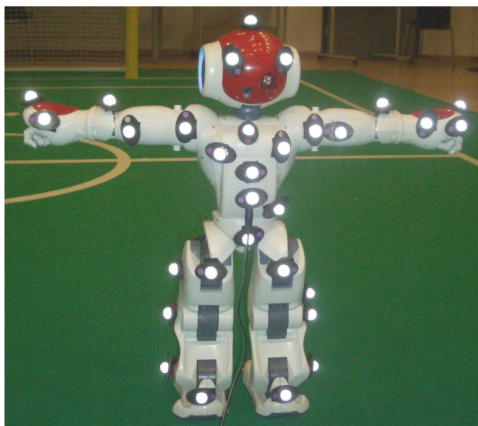


Figura 2.2: Distribuição de marcadores num robô *humanoid* NAO [2]

Estes tipos de sistemas começam a ser também utilizados em alguns laboratórios de investigação para auxiliar o controlo de UAVs [16, 23, 24] por estes não serem capazes de albergar sensores capazes de o fazer. Através da posição de cada reflector e conhecendo o modelo dinâmico e dos motores dos UAVs em questão, são processados externamente os controlos de posição e atitude e posteriormente são enviados comandos aos controladores dos UAVs para que eles se desloquem para uma determinada posição ou percorram uma trajectória previamente definida.

Em [25] são descritas três abordagens para a aquisição da posição real de um robô. Na primeira abordagem foram instaladas três câmaras nas paredes das instalações, a uma altura de aproximadamente 1,5 metros de altura relativamente ao chão. Estas câmaras foram distribuídas de forma a cobrirem a área onde os testes se iriam realizar cujas dimensões são de cerca de 6×6 metros. Para calcularem a posição do robô recorreram a marcações no chão, uma grelha cujas linhas estavam espaçadas $1m$ umas das outras, e calculavam a posição do robô relativamente a essa grelha cuja posição das várias linhas que a constituíam eram conhecidas, em pós-processamento. É dito que nas imagens capturadas pelas câmaras, as células mais próximas das câmaras aparentam ser maiores que as mais distantes o que afectou o pós-processamento das imagens e o cálculo da posição do robô. Com esta abordagem, a resolução estava limitada a meio metro e a orientação não estava disponível devido às pequenas dimensões do robô em algumas posições das imagens.

A segunda abordagem consistiu na utilização de um *Northstar Localization System* comercial. Nestes sistemas cada robô calcula a sua posição relativamente a um marcador infra-vermelho projectado no tecto. Os testes realizados a este sistema demonstraram que num raio de meio metro relativamente à origem o erro era inferior a $1cm$ em distância e inferior a 2° em ângulo. Os erros subiram para $10cm$ e 3° a um metro de distância, $35cm$ e 20° a dois metros e para distâncias superiores a dois metros relativamente à origem as posições eram praticamente ilegíveis.

Na terceira abordagem usaram um sistema de localização *Stargazer* que consiste na utilização de uma câmara activa, sensível a infra-vermelhos, para detectar um marcador passivo fixo no tecto. A vantagem principal deste método é que o processamento de imagem é feito localmente e a posição do robô fica disponível em tempo real. O erro em posição do robô quando utilizou este sistema foi de apenas $\pm 5cm$ e de $\pm 3^\circ$ na orientação, no entanto, cada marcador cobre uma área de apenas $2m^2$

portanto são precisos mais marcadores e fazer cálculos relativamente a vários marcadores de forma a cobrir a área necessária.

Para além dos custos avultados e os requisitos espaciais para instalar os três sistemas abordados neste artigo, a pouca exactidão e o facto de serem sistemas que apenas calculam a posição a duas dimensões tornam estas abordagens pouco indicadas para serem aplicadas na unidade de robótica do INESC TEC para analisar e controlar os seus sistemas autónomos.

Em [3] é descrito um sistema de *ground truth* por visão baseado em marcadores para determinar a posição 2D e orientação de vários robôs de futebol robótico da liga *small size* do *Robocup* que se movem a $2m/s$. O sistema é composto por duas câmaras com uma resolução de 384×288 pixels, colocadas a uma altura de aproximadamente $3m$, apontadas para baixo para verem o campo de cima. Juntas, o campo de visão das duas câmaras cobre na totalidade o campo que tem $6,4 \times 10,4m$. O sistema permite que se adicionem mais câmaras para poderem cobrir o campo oficial cujas dimensões são de $10 \times 14m$. Para identificarem e localizarem os robôs são colocados no seu topo marcadores coloridos. Os marcadores indicam a equipa a que cada jogador pertence e o seu número assim como permitem que seja deduzida a sua orientação. Para que o sistema funcione é preciso proceder a uma calibração de cores, de parâmetros extrínsecos e intrínsecos. Cada câmara captura imagens a uma taxa de $15Hz$ e que são segmentadas por cor para detectarem os marcadores e em seguida calculam a posição central do marcador e a orientação do robô.

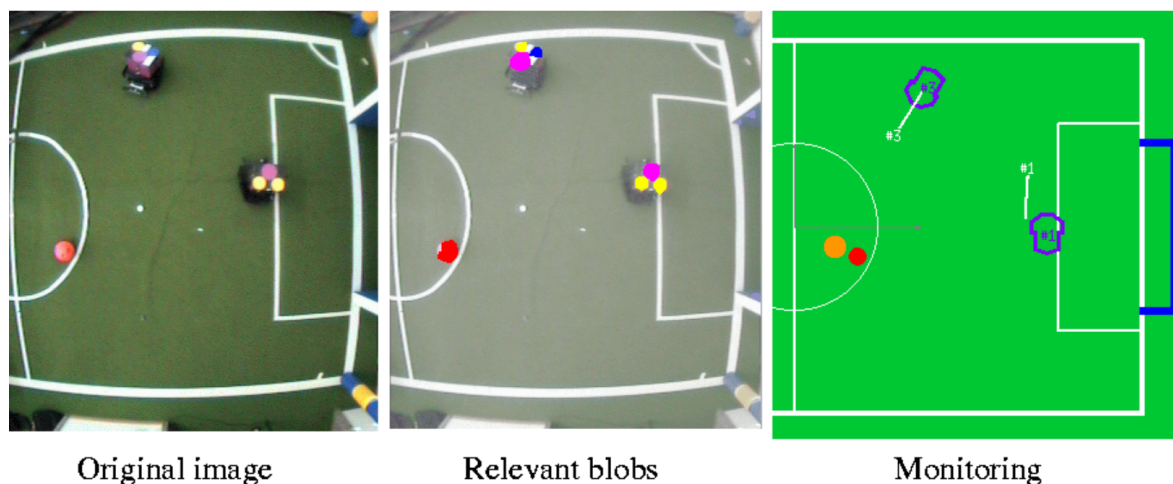


Figura 2.3: Passos intermédios no processamento de imagem [3]

Relativamente à exactidão deste sistema, através de testes com objectos estáticos determinaram que o erro em posição é inferior a $5,2cm$ e o erro em orientação é inferior a $2,3^\circ$. No que diz respeito a objectos em movimento, uma vez que têm dificuldade em determinar a suas localizações manualmente, supõem a partir de um conjunto de pressupostos que o erro em posição será inferior a $7cm$ e o erro em orientação deve ser menor que 19° .

Outro projecto bastante interessante mas que infelizmente foi descontinuado é o *GTMS - The Ground Truth Measurement System* [26]. Neste projecto estava a ser desenvolvido um programa para determinar a posição e fazer o seguimento de vários objectos no espaço tridimensional com recurso a

várias câmaras. Aos objectos eram colocados marcadores passivos que reflectiam a luz emitida por um conjunto de lâmpadas instaladas numa estrutura no tecto. Para aumentar o contraste e facilitar a detecção dos marcadores, a arena onde os testes estavam a ser feitos estava completamente revestida de preto. Este era um projecto cuja a abordagem e as características funcionais mais se assemelhava ao sistema que se pretende desenvolver durante este trabalho mas, tal como foi dito, foi descontinuado.

2.2 Sistemas comerciais

De seguida ficam algumas das mais conhecidas empresas ligadas ao desenvolvimento de sistemas de *ground truth* por visão e as características de alguns dos seus produtos.

A Vicon [4] é certamente a empresa líder mundial em sistemas de medição de movimento por visão baseada em marcadores. No mercado desde 1984, começou com o objectivo de criar uma ferramenta de análise biomecânica baseada em computador e vídeo para ajudar os atletas a melhorar a sua performance na preparação dos jogos Olímpicos. Tal como a OptiTrack [5] produzem sistemas de *tracking* recorrendo a câmaras e marcadores passivos para funcionarem em interiores.



Figura 2.4: Câmara da Vicon [4]

Caso se pretenda um sistema que funcione quer no interior quer ao ar livre, a escolha pode recair sobre a Qualisys, a MotionAnalysis ou a PhaseSpace [27, 28, 29], sendo que tanto a Qualisys oferece a possibilidade a escolha entre marcadores passivos ou activos que possibilitam a utilização dos seus sistemas ao ar livre a distancias superiores comparativamente a quando se usa marcadores passivos e a PhaseSpace só utiliza marcadores activos. Para além da opção entre marcadores activos ou passivos, a Qualisys tem câmaras dedicadas ao funcionamento debaixo de água a uma profundidade que pode ir até aos 40m com um alcance máximo de 20m.

Todas estas empresas têm disponíveis *softwares* dedicados ao *tracking* da posição actual de cada marcador, de objectos e formação de ligações esqueléticas do corpo humano a partir da posição dos marcadores e criação de um corpo virtual a partir dessa formação. A OptiTrack tem também disponível um *software* dedicado à captura de expressões faciais. Todos eles têm a possibilidade de visualização dos dados em tempo real ou gravação dos mesmos para posterior análise mais profunda.



Figura 2.5: Câmera V100:R2 da OptiTrack [5]

A conectividade do sistema pode variar entre apenas *Gigabit Ethernet* (GigE) para sistemas da Vicon, MotionAnalysis e PhaseSpace, *ethernet* e *wireless* na Qualisys ou ainda a opção entre GigE, *ethernet* ou *Universal Serial Bus* (USB) nos sistemas da OptiTrack.

Marca	Tipo de marcador possível		Ambiente de funcionamento		Opção de sistema à prova de água	Câmara com maior resolução ²	Câmara com maior FPS ³	Conectividade			
	Passivo	Activo	interior	exterior				GigE	<i>ethernet</i>	<i>wireless</i>	USB
Vicon	✓	✗	✓	✗	✗	4MP a 515FPS	1MP a 1000FPS	✓	✗	✗	✗
Optitrack	✓	✗	✓	✗	✗	4,1MP a 180FPS	0,69MP a 250FPS	✓	✓	✗	✓
Qualisys	✓	✓	✓	✓	✓	4MP a 180FPS	1,3MP a 500FPS	✗	✓	✓	✗
Motion Analysis	✓	✗	✓	✓	✗	12MP a 150FPS	2MP a 900FPS	✓	✗	✗	✗
PhaseSpace	✗	✓	✓	✓	✗	12MP a 960FPS	12MP a 960FPS	✓	✗	✗	✗

Tabela 2.1: Comparação entre sistemas comerciais de *ground truth* por visão

2.3 Conclusões

Pelo estado da arte apresentado conclui-se que cada um dos tipos de sistemas de *ground truth* desenvolvidos têm como alvo aplicações e situações muito específicas. Os sistemas apresentados para serem utilizados ao ar livre não funcionariam correctamente se aplicados em interiores enquanto que os sistemas pensados para serem aplicados em interiores teriam dificuldades em se adaptar a ambientes não tão estruturados e com iluminação variável.

No que diz respeito aos sistemas cujos sensores principais são câmaras, seriam os que melhor se adaptariam diferentes ambientes, no entanto, o uso ou não de marcadores nunca é opcional o que os torna pouco versáteis para serem aplicados na unidade de robótica do INESC TEC para analisar e controlar os seus sistemas autónomos, havendo ainda casos em que não é suportado o espaço tridimensional.

Face às contrariedades apresentadas surge a necessidade de encontrar uma solução capaz de poder ser aplicado no estudo e análise de diversos sistemas autónomos existentes na unidade de robótica do INESC TEC.

²Megapixel (MP)

³Frames Per Second (FPS)

Capítulo 3

Caracterização do Problema

3.1	Estudo dos sistemas autónomos e cenários de actuação	11
3.1.1	ISePorto	11
3.1.2	Quadrotor	12
3.2	Requisitos do sistema	13

Pretende-se ao longo desta tese, desenvolver um sistema visual para seguir um conjunto de objectos e determinar a sua localização, atitude e movimentação, para ser aplicado nas instalações da unidade de robótica INESC TEC [30]. Uma vez que a unidade de robótica INESC TEC se dedica ao desenvolvimento de sistemas autónomos com diversas funcionalidades e cenários de actuação, é importante estudar esses sistemas autónomos e os seus cenários pois afectarão a na resolução do problema.

3.1 Estudo dos sistemas autónomos e cenários de actuação

A existência de diferentes tipos de sistemas autónomos é sem dúvida um facto que tem que ser tido em conta para o sucesso do projecto. Na unidade de robótica INESC TEC existem vários veículos terrestres, aéreos e marítimos, sendo que o sistema que se pretende desenvolver recairá nos dois primeiros casos.

3.1.1 ISePorto

O ISePorto é uma equipa de futebol robótico *Middle Size League* (MSL) projectada para participar no *Robocup* e tem como objectivos o desenvolvimento e demonstração da investigação em áreas de interesse associadas aos sistemas autónomos.

Nos jogos futebol robótico MSL competem, num campo de 18×12 metros, duas equipas constituídas por cinco robôs em cada uma. Estes robôs movem-se no plano 2D com velocidades de até $5m/s$ e

com acelerações de $5m/s^2$ podendo atingir acelerações ainda maiores nas colisões. Têm que tomar decisões de forma completamente autónoma sendo que durante o jogo apenas a equipa de arbitragem lhes pode dar instruções. Isso acontece quando há, por exemplo, faltas, lançamentos laterais ou golos. Um dos árbitros anda dentro do campo e outro está num computador capaz de transmitir aos robôs as instruções do árbitro principal.

Neste cenário, o sistema a desenvolver será útil na caracterização das dinâmicas dos robôs, onde se podem aplicar marcadores, e dos outros objectos envolvidos em jogo, como a bola que se pode mover a mais de $10m/s$ quando há remates, sendo a sua aceleração muito elevada nos remates e ressaltos, onde tal não é possível por alterar a sua dinâmica. Permitirá também servir de referência para a avaliação dos algoritmos de localização e navegação dos robôs, e através de informação disponível em tempo real permitirá definir estratégias de ajuste automático em tempo real de parâmetros de estimadores e algoritmos de localização.

Para que seja possível realizar com sucesso estas tarefas, é preciso que haja sincronização de tempo entre o sistema externo e os robôs, assim como a capacidade de seguir vários objectos em simultâneo, muitas vezes próximos entre si e com a existência de oclusões.



Figura 3.1: Cenário de um jogo MSL

Como se pode perceber pela figura 3.1, as oclusões podem ocorrer por diversos motivos. A bola é o elemento em jogo mais susceptível de ser tapada pois os robôs rodeiam-na frequentemente. Os robôs também podem deixar de ser vistos pelo sistema de visão porque que querem tirar a bola à equipa adversária, cruzam-se e rodeiam-se o que acaba por tapar a visibilidade do sistema relativamente a alguns desses robôs. O próprio árbitro que anda dentro de campo e até o público que assiste aos jogos são também potenciais contribuintes para a existência de oclusões.

3.1.2 Quadrotor

O quadrotor é outro veículo que está disponível e que também merece um destaque especial na caracterização do problema. É um veículo com elevada manobrabilidade e velocidade de movimentos no plano tridimensional mas onde é possível aplicar marcadores em interferir com a sua dinâmica.

Devido à sua manobrabilidade, velocidades e acelerações lineares e angulares elevadas, estes veículos são um excelente candidato à utilização de um sistema de *ground truth* por visão não só para realizar formações devido às suas pequenas dimensões, mas sobretudo para os auxiliar no seu controlo.

Os quadrotoros são capazes de realizar bastantes manobras tais como: *hover*, que consiste em pairar no ar, *flips* laterais e frontais alterando, respectivamente, o seu *roll* e *pitch*, é também capaz de rodar sobre si mesmo em *yaw* e executar trajectórias circulares. Por vezes, as medidas fornecidas pelos sistemas de navegação tal como o GPS e o *Inertial Navigation System* (INS) não são suficientes para tirar o melhor partido deste tipo de veículo. Para que o controlo deste veículo seja melhorado, pretende-se que o sistema de *tracking* possa enviar dados para malha de realimentação de forma a melhorar o seu controlo.

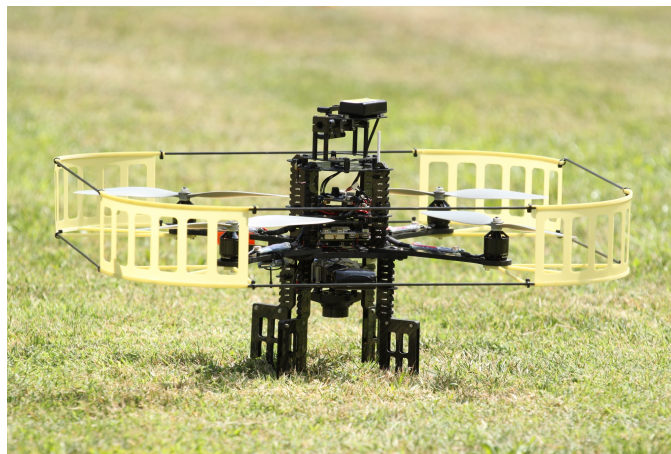


Figura 3.2: Quadrotor

3.2 Requisitos do sistema

Após o estudo dos sistemas autónomos e cenários de actuação percebe-se que será necessário desenvolver um sistema de visão multi-câmara para se obter uma medição externa exacta que determine a localização, atitude e seguimento de múltiplos objectos. Os dados obtidos servirão de referência ao estudo, caracterização e desenvolvimento de algoritmos de localização, navegação e controlo de vários sistemas autónomos.

Uma vez que o sistema deve ser utilizado em vários sistemas autónomos, pode ser útil que este consiga identificar vários tipos de marcadores, sejam eles à base de LEDs ou de *features* como cores e forma, porque diferentes estruturas e dinâmicas podem requerer diferentes tipos e formas de marcadores.

Tal como foi referido anteriormente na descrição do cenário do ISePorto, durante a trajectória desses robôs poderão ocorrer oclusões e colisões. Uma forma de diminuir o número de oclusões é aumentar o número de câmaras instaladas no local e distribuí-las de forma a aumentar as perspectivas de visão do local. Isso acontece pois se houverem mais perspectivas de visão a observar o mesmo local, a probabilidade do sistema perder momentaneamente o contacto visual com um determinado

marcador ou *feature* diminui. O facto de se ter mais câmaras disponíveis permite também aumentar o campo de visão para cobrir áreas maiores, combinar as medidas obtidas por vários sistemas *stereo* de forma a melhorar a qualidade da medida e ainda intercalar temporalmente a aquisição de imagens dos vários pares de câmaras para os casos em que sejam necessárias taxas de aquisição superiores, como por exemplo para seguir o quadrotor que é um veículo com elevada manobrabilidade e velocidade de movimentos.

As competições em que o ISePorto participa são também boas situações para estudar o comportamento geral da equipa. Como o calendário das competições é bastante apertado e os jogos são disputados em vários campos diferentes é importante que o sistema seja de fácil instalação e calibração para que possa ser utilizado nestas situações.

Então, o sistema de visão para *ground truth* deverá ser capaz de:

- Seguir um conjunto de veículos e/ou objectos com marcadores, em simultâneo com veículos e/ou objectos onde não se podem colocar marcadores;
- Detectar várias disposições de LEDs para seja possível identificar diferentes tipos de marcadores;
- Detectar diferentes *features* para seguir veículos e/ou objectos com diferentes cores e formas;
- Determinar a localização, atitude e movimentação desses veículos e objectos em tempo real;
- Suportar a existência de oclusões;
- Suportar a existência de colisões;
- Guardar os dados para posterior análise;
- Permitir a adição de vários pares de câmaras à medida que sejam necessárias;
- Ter um erro em posição máximo de 5cm a uma distância de 18m ;
- Ser de fácil instalação;
- Ser de fácil calibração;
- Ser um sistema de baixo custo;

Capítulo 4

Fundamentos de Visão Computacional

4.1	Extracção de <i>features</i>	15
4.2	Projectão Perspectiva	16
4.3	Distorção provocada pela lente	19
4.4	Visão <i>Stereo</i>	20
4.4.1	Geometria epipolar	20
4.4.2	Matriz de homografia	22
4.4.3	Rectificação <i>stereo</i>	23
4.4.4	Triangulação	24

Neste capítulo são apresentados alguns conceitos e fundamentos de visão computacional que foram utilizados durante a realização deste trabalho.

4.1 Extracção de *features*

No contexto da visão computacional, a designação de *feature* é atribuída a uma parcela da imagem que representa uma determinada particularidade de interesse com uma determinada característica. Como *feature* podem considerar-se, por exemplo, cantos (*corners*), zonas de fronteira (*edges*), linhas, curvas e pixels com intensidade díspar relativamente à sua vizinhança.

A extracção de *features* é uma tarefa de processamento de imagem de baixo nível que opera de forma densa sobre cada pixel da imagem, de modo a testar a existência de uma particularidade nesse ponto. A repetibilidade é das propriedades mais importantes nos algoritmos de detecção de *features*, porque a maior parte das aplicações de visão computacional depende da identificação das mesmas *features* ao longo de uma sequência de imagens. Por esse motivo, os algoritmos de extracção de *features* mais populares envolvem técnicas para fazer face a alterações de iluminação, perspectiva,

escala e rotação da imagem, provocadas quer pela movimentação da câmara quer pela movimentação das próprias *features*, de modo a que estas ocorrências não se reflectam na capacidade de detecção. Em contrapartida, à medida que estes algoritmos vão ficando mais robustos, os requisitos computacionais, para a sua implementação, aumentam. Factor também determinante na escolha de um algoritmo de extracção de *features*.

Para reduzir as necessidades computacionais dos algoritmos de extracção de *features*, a sua maioria opera em imagens em escala de cinza tornando-os mais simples e conseqüentemente computacionalmente menos exigentes.

4.2 Projecção Perspectiva

Designa-se de projecção perspectiva ao processo pelo qual se mapeiam pontos do mundo em pontos na imagem.

O cálculo da projecção perspectiva é efectuado com base num modelo que aproxima as características do sensor. O modelo com maior popularidade para o efeito é o modelo *pinhole*, que caracteriza a câmara através do centro óptico *c.o.*, o plano do centro óptico R^c e o plano da imagem R^i . Os planos do centro óptico e da imagem são paralelos entre si e a distância entre eles é designada por distância focal, f . O modelo *pinhole* considera que a passagem dos raios de luz é feita por um orifício com uma abertura infinitesimal antes de incidirem sobre o plano da imagem, permitindo assumir que um ponto tridimensional do cenário corresponde a um único ponto do plano da imagem.

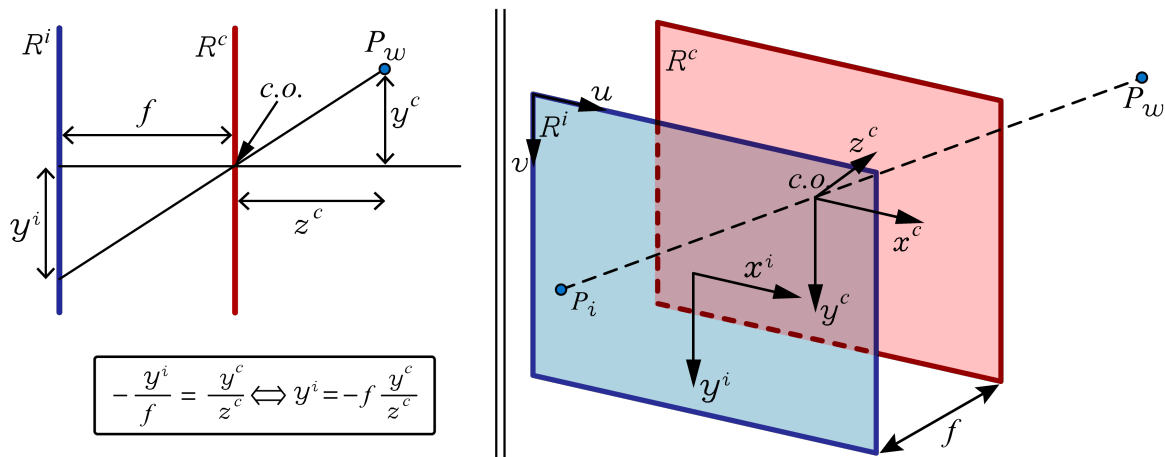


Figura 4.1: Modelo *pinhole*. À esquerda uma apresentação 2D do modelo *pinhole*. À direita a representação tridimensional acompanhada pelos sistemas referenciais da câmara e da imagem.

A figura 4.1 indica-nos a existência de uma relação geométrica simples, pelo que nos possibilita descrever a projecção por intermédio da equação linear (4.1).

$$\begin{bmatrix} x^i \\ y^i \end{bmatrix} = -\frac{f}{z^c} \begin{bmatrix} x^c \\ y^c \end{bmatrix} \quad (4.1)$$

- Onde (x^i, y^i) são as coordenadas métricas no plano da imagem, (x^c, y^c, z^c) a posição do ponto

P_c no referencial da câmara e f indica a distância focal.

A razão pelo qual o segundo membro da equação (4.1) se encontrar negado, deve-se ao facto de, de acordo com o modelo *pinhole*, a passagem de luz pelo orifício do centro óptico produzir uma inversão na imagem. Este pormenor é ultrapassado através da inversão da imagem, ou seja, deslocando o plano da imagem de modo a que este se posicione entre o ponto P_w e o centro óptico, mantendo no entanto o mesmo valor de distância focal (ver figura 4.2).

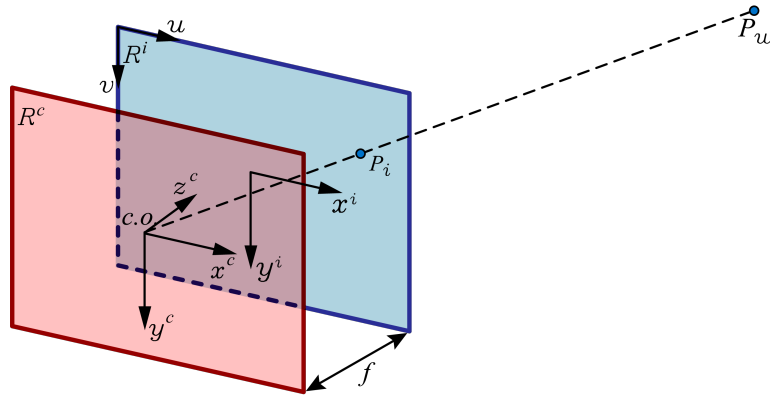


Figura 4.2: Modelo *pinhole* com o plano da imagem à frente do centro óptico

Considerando esta nova disposição do plano da imagem, pode-se reformular a equação (4.1) para que fique coerente com a deslocação realizada ao plano da imagem. O resultado desta reformulação pode ser visto na equação (4.2).

$$\begin{bmatrix} x^i \\ y^i \end{bmatrix} = \frac{f}{z^c} \begin{bmatrix} x^c \\ y^c \end{bmatrix} \quad (4.2)$$

Note-se a existência de dois sistemas referenciais no plano da imagem, o referencial com as componentes (x^i, y^i) e o referencial com as componentes (u, v) . O referencial com as componentes (x^i, y^i) é um sistema métrico e tem origem no ponto principal – ponto no plano da imagem correspondente à origem de um vector perpendicular ao plano da imagem e que intersecta o centro óptico. O referencial com as componentes (u, v) é definido em pixels e a sua origem situa-se no canto superior esquerdo da imagem.

Passemos então à formulação das transformações entre estes sistemas referenciais. Usando coordenadas homogêneas, um ponto P_w definido segundo um referencial global é expresso do seguinte modo: $P_w = [x^w, y^w, z^w, 1]$. A transformação deste ponto para o sistema de coordenadas da câmara é regida pela fórmula (4.3).

$$P_c = \left[\begin{array}{ccc|c} & & & \\ \hline & Rext(3 \times 3) & & Text(3 \times 1) \\ \hline 0 & 0 & 0 & 1 \end{array} \right] P_w \quad (4.3)$$

onde os parâmetros *Rext* e *Text* representam os parâmetros extrínsecos das câmaras.

Usando a relação da equação (4.2) pode definir-se a transformação do referencial da câmara (R_c) para o referencial da imagem (R_i) pela equação (4.4). O ponto no referencial da imagem será um vector homogéneo de dimensão 3.

$$P_i = \begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{x^c}{z^c} \\ \frac{y^c}{z^c} \\ 1 \end{bmatrix} \quad (4.4)$$

Por fim, as coordenadas de um ponto no referencial da imagem podem ser mapeadas em pixels na imagem aplicando da seguinte equação:

$$P_{u,v} = \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} P_i \quad (4.5)$$

- Onde $P_{u,v}$ representa as coordenadas homogéneas em pixels; k_v e k_u são respectivamente os factores de escala vertical e horizontal, essenciais à conversão de unidades métricas para pixels; u_0 e v_0 definem as coordenadas do ponto central da imagem em pixels.

Assim, um ponto definido no referencial da câmara ocupará uma posição em pixels na imagem dada pela equação (4.6).

$$P_{u,v} = \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{x^c}{z^c} \\ \frac{y^c}{z^c} \\ 1 \end{bmatrix} \quad (4.6)$$

A equação (4.6) é equivalente a:

$$P_{u,v} = \begin{bmatrix} f_x k_u & 0 & u_0 \\ 0 & f_y k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{x^c}{z^c} \\ \frac{y^c}{z^c} \\ 1 \end{bmatrix} \quad (4.7)$$

As coordenadas do vector $\begin{bmatrix} \frac{x^c}{z^c} & \frac{y^c}{z^c} & 1 \end{bmatrix}^T$ são designadas por coordenadas de projecção normalizadas e a matriz 3×3 da equação (4.8) é conhecida como matriz da câmara ou matriz de parâmetros intrínsecos, porque define as características físicas da câmara. A matriz de parâmetros intrínsecos é usualmente indicada através da letra K .

$$K = \begin{bmatrix} \alpha_u & s & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.8)$$

- Onde $\alpha_u = f_x k_u$, $\alpha_v = f_y k_v$ e $s = 0$. O factor s - *skew* - refere-se à não ortogonalidade entre linhas e colunas nos pixels da imagem e só em casos excepcionais toma valores diferentes de zero.

4.3 Distorção provocada pela lente

Idealmente, tal como é assumido no modelo *pinhole*, um ponto do mundo, o seu ponto de projecção na imagem e o centro óptico seriam colineares. Se assim fosse, linhas rectas no mundo dariam origem a linhas rectas na imagem, no entanto isso não acontece. O que realmente acontece é que há um desvio, geralmente provocado por uma distorção radial, entre o ponto do mundo e o seu ponto de projecção na imagem pois os raios de luz ao passarem pela lente são refractados, o que pode até ser vantajoso porque permitem aumentar o ângulo de visão e assim capturar um panorama mais alargado do mundo. Este efeito de distorção acentua-se quando a distância focal da lente e a sua qualidade óptica diminuem. [31]

A distorção pode ser descrita segundo duas componentes: tangencial e radial. A distorção tangencial⁴ tem como principal origem o desalinhamento físico dos vários elementos que constituem a lente. Nas câmaras modernas a perturbação induzida na imagem por esta componente é mínima, principalmente em lentes com distância focal fixa, pelo que nestes casos pode ser desprezada.

A distorção radial deve-se à variação do ângulo de refração à medida que aumenta a distância ao centro da lente. Quando a refração é menor nos extremos da lente, a imagem sofre distorção radial em forma de barril devido à redução da ampliação com o aumento da distância ao centro. De forma oposta, quando a refração é maior nos extremos, há um aumento da ampliação com o aumento da distância ao centro, produzindo na imagem uma perturbação em forma de almofada.

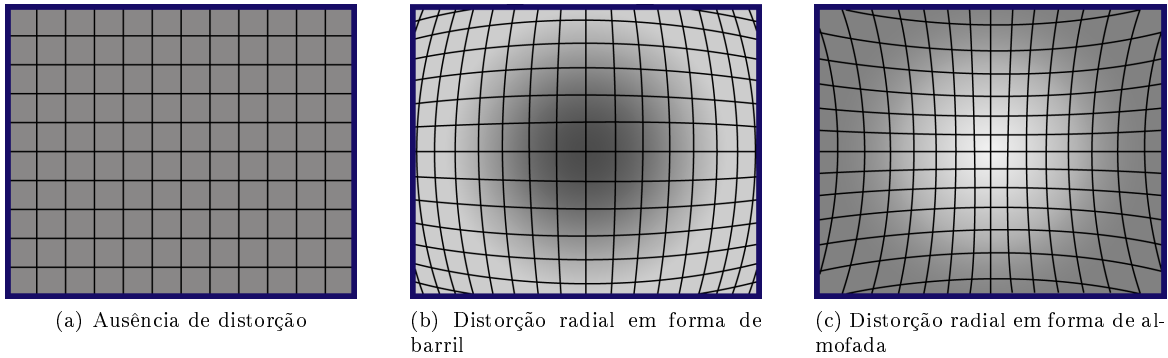


Figura 4.3: Projecção de rectas do mundo na imagem dependendo do tipo de distorção causada pela lente

A compensação do efeito de distorção da lente pode ser conseguida através da aplicação de um modelo polinomial introduzido por Brown [32]. Considerem-se as coordenadas de projecção normalizadas de um ponto não afectado por distorção: $[x^u \ y^u]^T = [x^c/z^c \ y^c/z^c]^T$. A distorção da lente origina um deslocamento deste ponto para as coordenadas $[x^d \ y^d]^T$ dadas pela seguinte equação:

$$\begin{bmatrix} x^d \\ y^d \end{bmatrix} = \begin{bmatrix} x^u \\ y^u \end{bmatrix} \left(1 + \kappa_1 r_u^2 + \kappa_2 r_u^4 + \kappa_5 r_u^6 \right) + dx \quad (4.9)$$

- Onde κ_1 , κ_2 e κ_5 são parâmetros de distorção radial; dx a deformação causada pela distorção tangencial e r_u a distância ao centro da imagem.

⁴No trabalho desenvolvido no âmbito desta tese a componente de distorção tangencial não é considerada

$$r_u = \sqrt{x^u{}^2 + y^u{}^2} \quad (4.10)$$

$$dx = \begin{bmatrix} 2\kappa_3 x^u y^u + \kappa_4 (r_u^2 + 2(x^u)^2) \\ \kappa_3 (r_u^2 + 2(y^u)^2) + 2\kappa_4 x^u y^u \end{bmatrix} \quad (4.11)$$

- Onde κ_3 e κ_4 correspondem a parâmetros de distorção tangencial.

Uma vez que o modelo de distorção opera sobre as coordenadas de projecção normalizadas, ao aplicar os parâmetros intrínsecos, geralmente obtidos por calibração, são determinadas as coordenadas dos pixels na imagem, considerando o efeito de distorção.

4.4 Visão Stereo

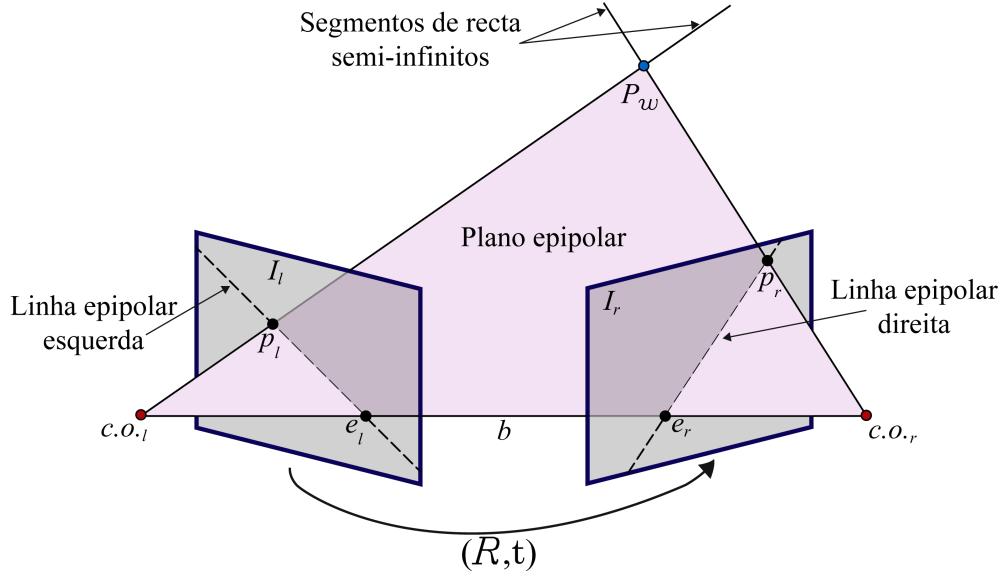
A transformação de pixels no referencial da imagem em pontos tridimensionais num referencial externo, necessários para que seja possível obter informações relativamente à profundidade desses pontos, não é tolerada pela projecção perspectiva inversa pois as coordenadas bidimensionais de um pixel em conjugação com o modelo *pinhole* não contêm informações suficientes para que a ambiguidade em profundidade seja ultrapassada. Segundo o modelo *pinhole*, vários pontos tridimensionais posicionados ao longo de um segmento de recta com origem no centro óptico, projectam-se no mesmo ponto da imagem. Por essa razão, a projecção perspectiva inversa de um pixel resulta num segmento de recta semi-infinito que intersecta as coordenadas do ponto no referencial métrico da imagem e tem como origem o centro óptico.

Para recuperar as coordenadas tridimensionais de um ponto a partir da sua projecção na imagem recorre-se a uma estratégia designada por triangulação. A triangulação consiste na visualização do ponto no mundo a partir de duas ou mais perspectivas diferentes. Para se proceder à aquisição de imagens com diferentes perspectivas pode-se integrar várias câmaras ou até pelo reposicionamento de apenas um sensor. A posição tridimensional do ponto é indicada pela intersecção dos vários segmentos de recta, resultantes da projecção inversa desenvolvida a cada perspectiva.

O procedimento de triangulação, para além da necessidade de conhecer rigorosamente a posição e orientação do sensor em cada local de amostragem, obriga à identificação do mesmo ponto nas várias imagens, para que as projecções inversas se desenvolvam relativamente ao mesmo local no mundo. Em sistemas *stereo*, a correspondência entre imagens é normalmente sustentada nas restrições impostas pela geometria epipolar.

4.4.1 Geometria epipolar

A figura 4.4 ilustra o princípio da geometria epipolar *stereo*, onde o ponto tridimensional P_w é projectado no plano da imagem de duas câmaras, cuja posição relativa (t) e orientação relativa (R) são conhecidas.

Figura 4.4: Geometria epipolar *stereo*

Os pontos identificados pelos caracteres *c.o.* representam os centros ópticos da câmara esquerda, *l*, e da câmara direita, *r*. A linha que une os dois centros ópticos é designada de *baseline*, *b*. O epipolo é o ponto no plano da imagem de uma câmara, no qual se projecta o centro óptico da outra câmara. Assim, o epipolo esquerdo, representado por e_l , corresponde à projecção de $c.o.r$ no plano I_l e, por sua vez, o epipolo direito, e_r , corresponde à projecção de $c.o.l$ em I_r . Tal como se pode verificar na figura 4.4, o plano epipolar é definido pelos pontos P_w , $c.o.l$ e $c.o.r$. O cruzamento do plano epipolar com os planos das imagens dá origem às linhas epipolares e todas as linhas epipolares possíveis intersectam os respectivos epipolos.

A restrição epipolar é uma propriedade que define que, considerando o ponto p_l como o ponto de projecção na imagem esquerda do ponto tridimensional P_w , então, o ponto p_r , resultante da projecção de P_w na imagem direita, se encontra posicionado ao longo da linha epipolar direita.

A geometria epipolar *stereo* é definida para um par de câmaras com disposição relativa constante e é independente da calibração de ambas. Pelo facto de não incorporar os parâmetros intrínsecos, a geometria epipolar, expressa pela equação (4.12), estabelece relações entre os sistemas referenciais métricos das imagens.

$$p_r^T E p_l = 0 \quad (4.12)$$

- Onde E diz respeito à matriz essencial 3×3 , que representa a geometria epipolar de forma algébrica.

$$E = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} R \quad (4.13)$$

Utilizando a matriz essencial é possível calcular a posição dos epipolos, a partir das equações (4.14)

e (4.15), bem como da linha epipolar numa imagem associada a um ponto da outra imagem (4.16) (4.17). [33]

$$Ee_l = 0 \quad (4.14)$$

$$E^T e_r = 0 \quad (4.15)$$

$$l_r = Ep_l \quad (4.16)$$

$$l_l = Ep_r \quad (4.17)$$

A matriz fundamental (4.18) corresponde à matriz essencial, representada em (4.13), acrescida dos parâmetros intrínsecos das câmaras (4.8). Assim, é possível executar as relações epipolares no referencial de pixels da imagem.

$$(K_r p_r)^T F (K_l p_l) = 0 \quad (4.18)$$

4.4.2 Matriz de homografia

A matriz de homografia estabelece relações de posição entre dois planos distintos, definindo as regras de mapeamento de pontos entre os planos. Assim, para relações entre pontos de duas imagens, a morfologia do cenário é irrelevante. Considerando-se um ponto tridimensional no mundo P_w com pontos de projecção na imagem da esquerda e da direita, respectivamente p_l e p_r em coordenadas homogéneas, a relação entre pontos das duas imagens pela matriz de homografia H (4.22) é expressa pela equação (4.19).

$$p_r = H.p_l \quad (4.19)$$

onde

$$p_l = \begin{bmatrix} x_l \\ y_l \\ 1 \end{bmatrix} \quad (4.20)$$

$$p_r = \begin{bmatrix} x_r \\ y_r \\ 1 \end{bmatrix} \quad (4.21)$$

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \quad (4.22)$$

A correspondência entre um par de pontos em duas imagens resulta em duas equações:

$$x_r = \frac{x_l h_{11} + y_l h_{12} + h_{13}}{x_l h_{31} + y_l h_{32} + h_{33}} \quad (4.23)$$

$$y_r = \frac{x_l h_{21} + y_l h_{22} + h_{23}}{x_l h_{31} + y_l h_{32} + h_{33}} \quad (4.24)$$

4.4.3 Rectificação *stereo*

A rectificação de imagens *stereo* [35] [36] [37] é uma prática comum que antecede o cálculo de distâncias. O processo de rectificação executa uma re-disposição dos pixels nas imagens de modo a simular o alinhamento dos centros focais e planos da imagem das duas câmaras. Deste modo, os epipolos são projectados para o infinito, o que torna as linhas epipolares horizontais e colineares, facilitando a correspondência entre imagens.

A operação de deformação das imagens é baseada na aplicação de uma matriz de homografia a cada imagem. O cálculo destas matrizes é governada pela imposição da colocação dos epipolos no infinito e pelo facto de as linhas epipolares se tornarem paralelas relativamente ao eixo de coordenadas u da imagem. Estas condições geram restrições a ter em conta no cálculo da matriz de homografia, contudo ainda ficam por definir quatro graus de liberdade [35]. Deste modo não existe uma solução única para o problema de rectificação e por isso existem inúmeras técnicas de cálculo das matrizes de homografia. Os métodos para cálculo das matrizes de homografia partilham a principal preocupação de minimização da distorção a aplicar às imagens. [35] [36]

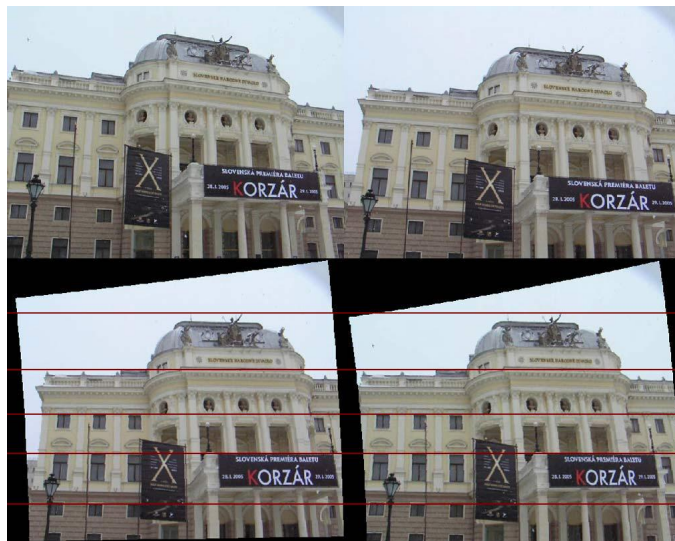


Figura 4.5: Rectificação de um par de imagens *stereo* e representação das linhas epipolares horizontais colineares fruto da localização do epipolo no infinito [38].

Após a aplicação das transformações no processo rectificação, a geometria do sistema *stereo* torna-se idêntica à apresentada na figura 4.6, o que torna bastante simples o cálculo das coordenadas tridimensionais de pontos do mundo por triangulação, tal como se pode observar nas equações fórmulas (4.25), (4.26) e (4.27).

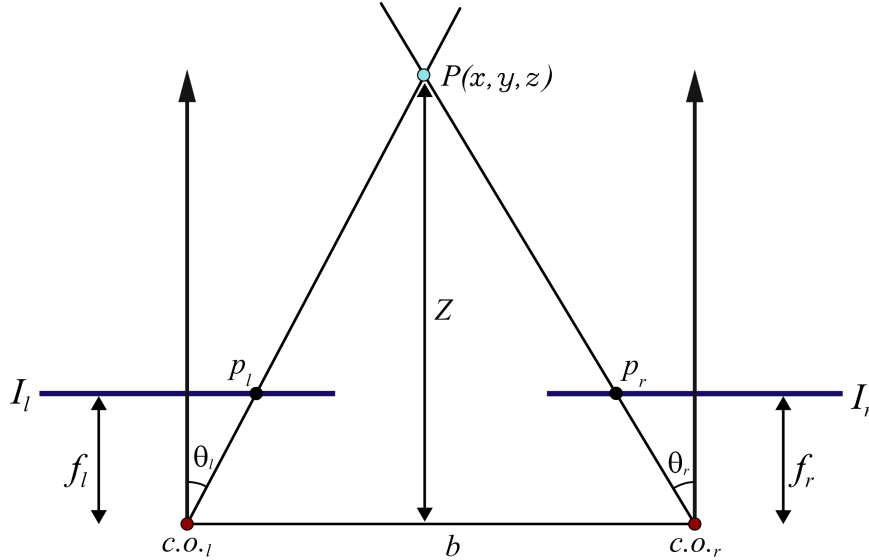


Figura 4.6: Geometria após rectificação – epípolos no infinito e planos da imagem paralelos.

$$z = \frac{fb}{x_l - x_r} \quad (4.25)$$

$$x = \frac{x_l z}{f} \quad (4.26)$$

$$y = \frac{y_l z}{f} \quad (4.27)$$

- Onde x_l e x_r são, respectivamente, as coordenadas x dos pontos p_l e p_r .

4.4.4 Triangulação

Em visão computacional, designa-se de triangulação ao processo que visa determinar as coordenadas tridimensionais de um ponto no referencial do mundo dadas as suas projecções em pelo menos duas imagens. Durante este trabalho as coordenadas tridimensionais dos pontos no referencial do mundo foram obtidas por triangulação por intermédio da rectificação. Detalhes sobre triangulação sem rectificação podem ser vistos em [39]. De seguida serão identificados os passos principais para se obterem as coordenadas tridimensionais de um ponto no referencial do mundo.

Primeiro, recorrendo a técnicas de extracção de *features* identificam-se as coordenadas em pixels de pontos de interesse em cada uma das imagens que foram capturados por câmaras e mapeados nas imagens através da projecção perspectiva.

Como as lentes das câmaras provocam uma distorção na imagem, é necessário aplicar os parâmetros intrínsecos que se obtêm por calibração e retirar a distorção aos pontos de interesse.

Quando já se tem os pontos sem distorção nas imagens segue-se a rectificação dos pontos sem distorção. O facto de se rectificarem os pontos torna o processo de fazer a correspondência dos pontos de uma imagem com os pontos da outra imagem mais simples. Isso acontece porque a geometria epipolar que permite fazer a correspondência entre as coordenadas x, y de um ponto de uma imagem com as coordenadas x, y do ponto que lhe é correspondente na outra imagem precisa de fazer essa procura correndo a imagem nas duas direcções, x e y . No entanto, como as imagens estão rectificadas, a coordenada y de um certo ponto é igual nas duas imagens, logo só é preciso identificar a coordenada x correspondente.

Feitas as correspondências dos pontos *stereo* são calculadas as coordenadas tridimensionais dos pontos no referencial do par de câmaras *stereo* rectificado, é aplicada uma rotação para o referencial não rectificado e, por fim, transformados em pontos no referencial do mundo através dos parâmetros extrínsecos obtidos também através de calibração.

Capítulo 5

Métodos de estimação

5.1	Filtro de <i>Kalman</i>	27
5.1.1	Modelos de estado e sensor	28
5.2	Filtro de <i>Kalman</i> discreto	29
5.2.1	Modelos do sistema e observação	29
5.2.2	Previsão	30
5.2.3	<i>Update</i>	30
5.3	Fusão de dados multi-sensorial	31
5.3.1	Filtro de <i>Kalman</i> multi-sensorial	31
5.3.2	Associação de dados multi-sensorial	35

Neste capítulo pretende-se abordar os métodos de estimação e associação de dados utilizados em problemas de fusão multi-sensorial. Foi escrito com base na informação e nomenclatura de [40].

5.1 Filtro de *Kalman*

O *Kalman Filter* (KF) é um estimador linear recursivo gaussiano que estima sucessivamente o estado de um sistema dinâmico $x(t)$, que evolui ao longo do tempo segundo um modelo probabilístico da dinâmica do sistema, com base em observações periódicas $z(t)$ contaminadas com ruído e de modelos probabilísticos dos sensores, representado pelos momentos de primeira e segunda ordem (média e variância).

As observações podem ser associadas a diferentes modelos de sensores o que o tornam ideal para lidar com estimativas multi-sensoriais complexas e problemas de fusão de dados. A relação entre a incerteza associada a cada observação e a incerteza no estado, associada ao modelo do sensor, vai

permitir ao filtro pesar a contribuição dessa medida na correcção do estado de forma a minimizar a variância do estado.

5.1.1 Modelos de estado e sensor

O ponto de partida do algoritmo KF é definir um modelo para os estados a serem estimados na forma espaço-temporal. O estado é definido por um conjunto de equações que definem a evolução do estado ao longo do tempo, $\mathbf{F}(t)\mathbf{x}(t)$, entradas de controlo que actuam directamente no estado alterando-o, $\mathbf{B}(t)\mathbf{u}(t)$ e uma perturbação do sistema que se considera ser ruído $\mathbf{G}(t)\mathbf{v}(t)$.

$$\dot{\mathbf{x}}(t) = \mathbf{F}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) + \mathbf{G}(t)\mathbf{v}(t) \quad (5.1)$$

- Onde:

$\mathbf{x}(t) \in \mathfrak{R}^n$ é o vector de estado de interesse,

$\mathbf{u}(t) \in \mathfrak{R}^s$ é uma entrada de controlo conhecido,

$\mathbf{v}(t) \in \mathfrak{R}^q$ é uma variável aleatória que descreve a incerteza na evolução do estado,

$\mathbf{F}(t)$ é a matriz de estado $n \times n$,

$\mathbf{B}(t)$ é a matriz de entrada $n \times s$,

e $\mathbf{G}(t)$ é a matriz de ruído $n \times q$.

Para se fazerem as estimativas ao estado são necessárias observações a esse estado. As observações estão contaminadas com ruído. Um modelo de observação então definido na forma espaço-temporal.

$$\mathbf{z}(t) = \mathbf{H}(t)\mathbf{x}(t) + \mathbf{D}(t)\mathbf{w}(t) \quad (5.2)$$

- Onde:

$\mathbf{z}(t) \in \mathfrak{R}^m$ é o vector de observação,

$\mathbf{w}(t) \in \mathfrak{R}^r$ é uma variável aleatória que descreve a incerteza da observação,

$\mathbf{H}(t)$ é a matriz de observação $m \times n$,

$\mathbf{D}(t)$ é a matriz de ruído na observação $m \times r$.

Adicionalmente, assume-se que as fontes de ruído do modelo do sistema e dos sensores, $\mathbf{v}(t)$ e $\mathbf{w}(t)$, são variáveis aleatórias gaussianas, de média nula e com covariâncias \mathbf{Q} e \mathbf{R} respectivamente, e não estão correlacionadas entre si.

Estas equações definem a evolução de um sistema contínuo no tempo com observações realizadas continuamente ao estado. Contudo, o KF é praticamente sempre implementado em modo discreto devido à necessidade de ser utilizado em sistemas computacionais.

5.2 Filtro de *Kalman* discreto

O algoritmo do KF discreto consiste em sucessivas previsões e actualizações fornecidas por observações ao estado $z(k)$ que visam a obtenção de uma estimativa do vector de estado $x(k)$ com uma variância mínima. Isto revela uma simplicidade essencial do algoritmo pois os passos efectuados consistem em simples operações de somas de observações e estimativas pesadas.

Na figura 5.1 pode-se ver um diagrama de blocos representativo do ciclo do KF.

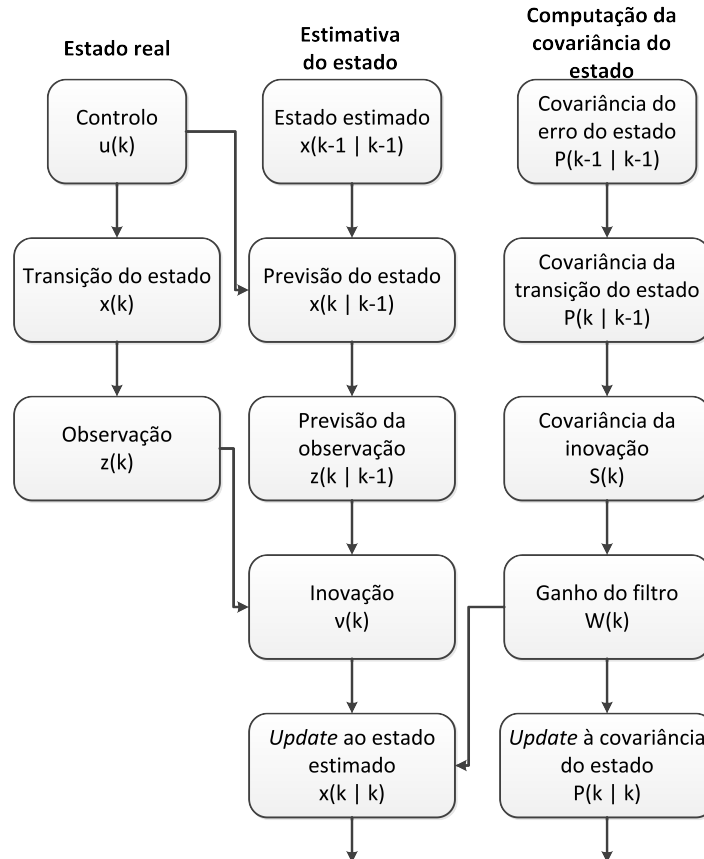


Figura 5.1: Diagrama de blocos do ciclo do filtro de *Kalman*

5.2.1 Modelos do sistema e observação

Para definir o modelo do sistema começa-se por se assumir que o sistema que se quer modelizar pode ser descrito por uma equação de transição de estado linear e discreta.

$$\mathbf{x}(k) = \mathbf{F}(k)\mathbf{x}(k-1) + \mathbf{B}(k)\mathbf{u}(k) + \mathbf{G}(k)\mathbf{v}(k) \quad (5.3)$$

- Onde $\mathbf{x}(k)$ é o estado no instante de tempo k , $\mathbf{F}(k)$ é a matriz de transição de estado, $\mathbf{x}(k-1)$ é o estado no instante tempo $k-1$, $\mathbf{u}(k)$ é um vector de entrada, $\mathbf{v}(k)$ representa a adição de ruído ao estado, $\mathbf{B}(k)$ e $\mathbf{G}(k)$ são respectivamente as matrizes de transição de entrada e ruído de transição.

As observações do estado deste sistema são realizadas de acordo com a equação linear:

$$\mathbf{z}(k) = \mathbf{H}(k)\mathbf{x}(k) + \mathbf{w}(k) \quad (5.4)$$

- Onde $\mathbf{z}(k)$ é a observação feita no instante de tempo k , $\mathbf{H}(k)$ é a matriz de observação e $\mathbf{w}(k)$ representa o ruído existente na observação.

5.2.2 Previsão

A segunda etapa do KF é a previsão. Na previsão, $\hat{\mathbf{x}}(k | k - 1)$, é feita uma estimativa do estado no instante k tendo apenas disponível informações sobre o estado apenas até ao instante anterior, $k - 1$, sem que tenham havido entretanto observações. A previsão é portanto calculada com base nas equações que descrevem a dinâmica do estado, na estimativa do estado no instante anterior com informações até esse instante e nas entradas de controlo $u(k)$ conhecidas:

$$\hat{\mathbf{x}}(k | k - 1) = \mathbf{F}(k)\hat{\mathbf{x}}(k - 1 | k - 1) + \mathbf{B}(k)\mathbf{u}(k) \quad (5.5)$$

A actualização da incerteza do estado é dada pela equação de previsão da covariância:

$$\mathbf{P}(k | k - 1) = \mathbf{F}(k)\mathbf{P}(k - 1 | k - 1)\mathbf{F}^T(k) + \mathbf{G}(k)\mathbf{Q}(k)\mathbf{G}^T(k) \quad (5.6)$$

onde $\mathbf{Q}(k)$ é a matriz de covariância do ruído do processo e $\mathbf{G}(k)$ é o Jacobiano do modelo do sistema em ordem às fontes de ruído.

Quando há observações ao estado, o filtro entra na fase de *update* e actualiza a estimativa do estado tendo em conta essas observações.

5.2.3 Update

Durante a fase de *update* é feita uma actualização ao estado estimado através de medidas que tenham sido obtidas pelos sensores do sistema. Como foi dito anteriormente, as medidas obtidas pelos sensores não são completamente correctas porque por diversos motivos, como por exemplo, pela qualidade do sensor ou interferências provocadas pelo meio ambiente. Por essa razão, no KF há uma matriz \mathbf{R} que representa o ruído existente nas medidas obtidas. Inicialmente, no *update*, é calculada a diferença entre as medidas obtidas e as medidas que se previa obter.

A diferença entre a observação $\mathbf{z}(k)$ e a observação prevista $\hat{\mathbf{z}}(k | k - 1)$ é designada de inovação e é representada por $\nu(k)$:

$$\nu(k) = \mathbf{z}(k) - \hat{\mathbf{z}}(k | k - 1) \quad (5.7)$$

onde, a previsão da observação é obtida recorrendo ao modelo do sensor:

$$\hat{\mathbf{z}}(k | k - 1) = \mathbf{H}(k)\hat{\mathbf{x}}(k | k - 1) \quad (5.8)$$

A inovação é importante pois mede o desvio entre as observações previstas com base no estado

do filtro e a sequência de observações. Na realidade, como normalmente o estado verdadeiro não está disponível para que se possa comparar com os estados estimados, a inovação é muitas vezes a única medida para se analisar o quão bem o estimador está a estimar. Para além disso, a inovação pode ser utilizada para ajudar a afinar o KF e validar medidas antes que estas sejam incorporadas nas estimativas filtradas, o que permite rejeitar “*outliers*”.

A variância da inovação pode ser obtida por:

$$\mathbf{S}(k) = \mathbf{R}(k) + \mathbf{H}(k)\mathbf{P}(k | k - 1)\mathbf{H}^\top(k) \quad (5.9)$$

A partir da inovação e do ganho de *Kalman* $\mathbf{W}(k)$ pode-se actualizar o estado estimado (5.11) assim como a sua variância (5.12).

$$\mathbf{W}(k) = \mathbf{P}(k | k - 1)\mathbf{H}^\top(k)\mathbf{S}^{-1}(k) \quad (5.10)$$

$$\hat{\mathbf{x}}(k | k) = \hat{\mathbf{x}}(k | k - 1) + \mathbf{W}(k)\nu(k) \quad (5.11)$$

$$\mathbf{P}(k | k) = \mathbf{P}(k | k - 1) - \mathbf{W}(k)\mathbf{S}(k)\mathbf{W}^\top(k) \quad (5.12)$$

5.3 Fusão de dados multi-sensorial

A fusão de dados multi-sensorial consiste em criar estimativas do estado com base em observações realizadas por vários sensores que podem ser, ou não, do mesmo tipo. Como se pretende que o sistema seja capaz de suportar vários sistemas *stereo* este é um tema importante para a elaboração do trabalho. De seguida será descrito como se pode aplicar o KF para efectuar uma fusão de dados multi-sensorial.

5.3.1 Filtro de *Kalman* multi-sensorial

Muitas das técnicas desenvolvidas para usar nos KFs com apenas um sensor podem ser aplicadas directamente em problemas relacionados com estimação multi-sensorial. Há várias formas de abordar este problema quer recorrendo a arquitecturas centralizadas quer a arquitecturas distribuídas.

Os sistemas aqui considerados são centralizados, por isso, as observações feitas pelos sensores são enviados para uma unidade de processamento central onde são processadas por um algoritmo tal como acontece com os sistemas com apenas um sensor.

Nos sistemas centralizados há vários métodos diferentes para que o KF possa lidar com fusões multi-sensoriais. Estes métodos diferem principalmente no processo de *Update*, mais concretamente na forma como consideram as várias observações dos sensores.

As duas opções mais triviais, para abordar a existência de várias medidas no mesmo instante, são a de agrupar as várias medidas numa observação do grupo de sensores, e a de considerar as várias medidas uma a uma de forma sequencial. Estes métodos designam-se de método de sensores agrupados

e método de sensores sequenciais.

De seguida será descrita uma arquitectura de fusão de dados centralizadas e os métodos de sensores agrupados e sequenciais.

Modelo de observação

Na figura 5.2 é esquematizada uma arquitectura de fusão de dados centralizada, onde cada sensor tem um modelo de observação diferente mas observa um processo comum. As observações são enviadas por processar para um centro de fusão central que combina as observações para obter uma estimativa do estado comum.

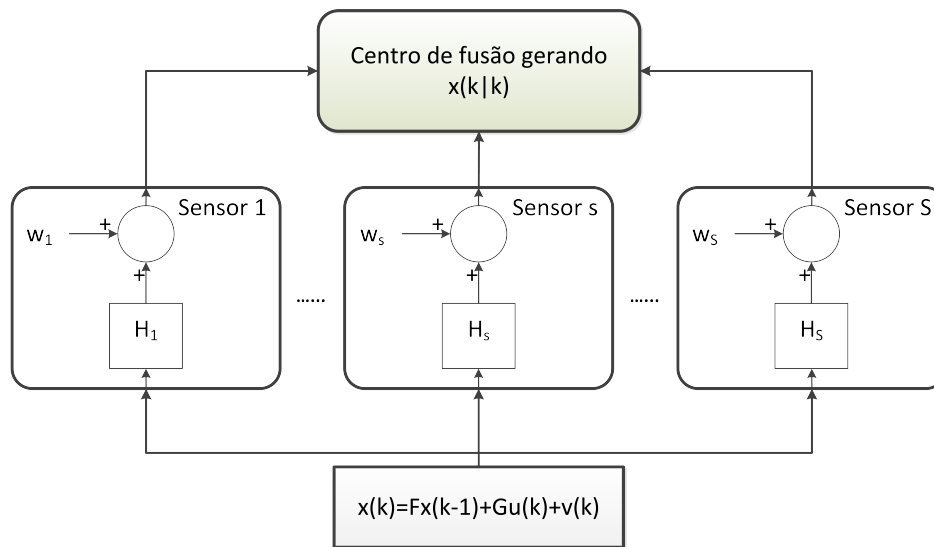


Figura 5.2: Estrutura de uma arquitectura de fusão de dados centralizada

O modelo do estado comum é obtido na forma linear e discreta.

$$\mathbf{x}(k) = \mathbf{F}(k)\mathbf{x}(k-1) + \mathbf{G}(k)\mathbf{u}(k) + \mathbf{v}(k) \quad (5.13)$$

- Onde $\mathbf{x}(\cdot)$ é o estado de interesse, $\mathbf{F}(k)$ é a matriz de transição de estado, $\mathbf{G}(k)$ é o modelo de entrada de controlo, $\mathbf{u}(k)$ é o vector de entrada de controlo, $\mathbf{v}(k)$ um vector aleatório que descreve a incerteza do modelo e incerteza do processo.

As observações do estado deste sistema são feitas em sincronia por vários sensores diferentes de acordo com um conjunto de modelos de observação linear.

$$\mathbf{z}_s(k) = \mathbf{H}_s(k)\mathbf{x}(k) + \mathbf{w}_s(k), \quad s = 1, \dots, S \quad (5.14)$$

- Onde $\mathbf{z}_s(k)$ é a observação do estado comum $\mathbf{x}(k)$, realizada pelo sensor s no instante de tempo k , de acordo com o modelo de observação $\mathbf{H}_s(k)$ e com a adição de ruído $\mathbf{w}_s(k)$.

Método de sensores agrupados

A forma mais simples e correcta de lidar com o problema de estimação multi-sensorial é combinar todas as observações e modelos de observação num “grupo de sensores”. Assim, todas as observações seriam tratadas pelo sistema como se de apenas uma observação se tratasse. A mesma técnica é aplicada aos sensores. Os modelos de observação são combinados num único mas mais complexo modelo de observação. Tendo as observações e modelos de observação agrupados pode-se usar um algoritmo idêntico ao utilizado em sistemas com um único sensor. O vector de observação composto é definido por:

$$\mathbf{z}(k) \equiv [\mathbf{z}_1^T(k), \dots, \mathbf{z}_S^T(k)]^T \quad (5.15)$$

O modelo de observação composto é definido por:

$$\mathbf{H}(k) \equiv [\mathbf{H}_1^T(k), \dots, \mathbf{H}_S^T(k)]^T \quad (5.16)$$

com

$$\mathbf{w}(k) \equiv [\mathbf{w}_1^T(k), \dots, \mathbf{w}_S^T(k)]^T \quad (5.17)$$

$$\mathbf{R}(k) = \text{diagonal}\{\mathbf{R}_1(k), \dots, \mathbf{R}_S(k)\} \quad (5.18)$$

A covariância do ruído da observação é uma matriz diagonal cujos elementos que a constituem são iguais às matrizes de covariância do ruído de observação dos sensores individuais. O conjunto de equações de observação definidas pela equação (5.14) podem ser redefinidas num grupo de modelo de observação:

$$\mathbf{z}(k) = \mathbf{H}(k)\mathbf{x}(k) + \mathbf{w}(k) \quad (5.19)$$

Com o modelo do processo descrito pela equação (5.13) e o de modelo de observação em grupo definido na equação (5.19), as estimativas do estado podem ser calculadas usando o algoritmo padrão do KF descrito no capítulo capítulo 5.2 Filtro de *Kalman* discreto, página 29.

A fase de previsão do algoritmo do KF multi-sensorial é idêntica em todos os aspectos à fase de previsão do filtro para um único sensor. Contudo, a fase de *update* será afectada pelo aumento do número de sensores. Se se tiver um vector de estado $\mathbf{x}(k)$ de dimensão n e S sensores, cada um deles com um vector de observação $\mathbf{z}_s(k)$, $s = 1, \dots, S$ de dimensão m_s juntamente com um modelo de observação $\mathbf{H}_s(k)$ de dimensão $m_s \times n$, então o vector do grupo de observação $\mathbf{z}(k)$ terá $\mathbf{m} = \sum_{s=1}^S m_s$ de tamanho e o grupo de modelo de observação $\mathbf{H}(k)$ terá dimensão $\mathbf{m} \times n$. A inovação do grupo de sensores $\nu(k)$ terá dimensão \mathbf{m} e a covariância do grupo de sensores $\mathbf{S}(k)$ será $\mathbf{m} \times \mathbf{m}$. Como o número de sensores incorporados no grupo de sensores aumenta, a dimensão do vector da inovação e a matriz da covariância da inovação também aumentam. Isto representa um problema porque a

inversa da covariância da inovação é necessária para computar a matriz de ganho $\mathbf{W}(k)$ do grupo de sensores na equação (5.10) e o custo computacional para o cálculo da inversa de uma matriz aumenta quadraticamente com a sua dimensão.

Para poucos sensores esta abordagem pode ser a mais prática de implementar. No entanto, à medida que o número de sensores aumenta, esta abordagem de fusão de dados torna-se cada vez mais pesada computacionalmente. Por essa razão, durante este trabalho optou-se pelo método dos sensores sequenciais.

Método de sensores sequenciais

No método de sensores sequenciais considera-se cada observação feita pelo sensor como independente. O *update* ao estado estimado é feito sequencialmente e para cada observação a computar, há uma matriz de previsão e ganho adequadas. Ao contrário do método dos sensores agrupados, em que era construído um único modelo do sensor através da combinação de todos os modelos dos sensores, este método considera cada modelo de sensor individualmente. Isto significa que as dimensões da inovação e da matriz da covariância da inovação não aumentam à medida do número de sensores aumenta, mas em contrapartida é necessário computar uma nova matriz de ganho a cada observação para cada sensor.

A descrição do estado a ser estimado está na forma da equação (5.13). As observações a este estado comum são feitas por S sensores de acordo com a equação (5.14).

A etapa de previsão para o estimador usando sensores sequenciais é muito semelhante à etapa de previsão do estimador que usa um único sensor.

$$\hat{\mathbf{x}}(k | k, 0) = \mathbf{F}(k)\hat{\mathbf{x}}(k-1 | k-1, S) + \mathbf{G}(k)\mathbf{u}(k) \quad (5.20)$$

com a covariância correspondente a

$$\mathbf{P}(k | k, 0) = \mathbf{F}(k)\mathbf{P}(k-1 | k-1, S)\mathbf{F}^\top(k) + \mathbf{Q}(k) \quad (5.21)$$

O *update* ao estado a cada instante de tempo pode ser computado recorrendo à equação (5.22):

$$\begin{aligned} \hat{\mathbf{x}}(k | k) = & \left[\prod_{s=1}^S (1 - \mathbf{W}_s(k)\mathbf{H}_s(k)) \right] \hat{\mathbf{x}}(k | k-1) \\ & + \sum_{i=1}^S \left[\prod_{j=i+1}^S (1 - \mathbf{W}_j(k)\mathbf{H}_j(k)) \right] \mathbf{W}_i(k)\mathbf{z}_i(k) \end{aligned} \quad (5.22)$$

Embora o custo computacional aumente à medida que o número de sensores aumenta porque é preciso calcular uma nova matriz de ganho a cada observação de cada sensor a cada instante de tempo, ao contrário do que acontece com o método dos sensores agrupados, a matriz de covariância da inovação que precisa de ser invertida não cresce, logo o custo computacional deste método é inferior.

5.3.2 Associação de dados multi-sensorial

O problema de associação de dados multi-sensorial consiste em associar correctamente as medidas efectuadas aos vários estados, isto é às várias *track*, que estão a ser observados.

O problema de associação de dados inclui questões de validação de dados, associando as medições correctas aos estados correctos e inicializar novas trajectórias (*tracks*) se necessário. Enquanto que o *tracking* convencional se preocupa com a incerteza da localização da medição, a associação de dados preocupa-se com a incerteza da origem da medição.

A associação de dados baseia-se no uso da inovação normalizada ou “região de validação”. A inovação $\nu_{ij}(k)$ é a diferença entre a observação actualmente feita $\mathbf{z}_i(k)$ e aquela prevista pelo filtro $\hat{\mathbf{z}}_j(k | k - 1) = \mathbf{H}_j(k)\hat{\mathbf{x}}_j(k | k - 1)$.

$$\nu_{ij}(k) = \mathbf{z}_i(k) - \hat{\mathbf{z}}_j(k | k - 1) = \mathbf{z}_i(k) - \mathbf{H}_j(k)\hat{\mathbf{x}}_j(k | k - 1) \quad (5.23)$$

A variância da inovação é definida por:

$$\mathbf{S}_{ij}(k) = \mathbf{H}_j(k)\mathbf{P}_j(k | k - 1)\mathbf{H}_j^T(k) + \mathbf{R}_i(k) \quad (5.24)$$

A inovação normalizada entre uma observação i e uma *track* j é dada por:

$$d_{ij}^2 = \nu_{ij}^T(k)\mathbf{S}_{ij}^{-1}(k)\nu_{ij}(k) \quad (5.25)$$

Na figura 5.3 está ilustrada a região de validação. A região está definida no espaço de observação e está centrada na previsão da observação $\mathbf{H}(k)\hat{\mathbf{x}}(k | k - 1)$.

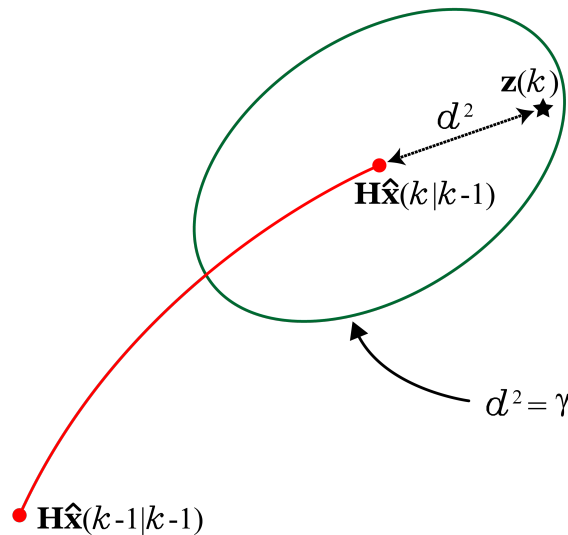


Figura 5.3: Região de validação

Ainda relativamente à figura 5.3, se uma observação estiver dentro deste volume então é considerada válida. Daí a designação de região de validação.

A inovação normalizada serve de base para todas as técnicas de associação de dados. Isso acontece

porque a inovação é praticamente a única medida de divergência do estado estimado a partir de uma sequência de observações e porque admite uma medida probabilística de associação correcta.

Existem três métodos de associação de dados multi-sensoriais que serão descritos de seguida.

Nearest-Neighbour Standard Filter

O *Nearest-Neighbour Standard Filter* (NNSF) aplica a política mais óbvia e simples de associação de dados: escolhe a medida mais próxima da medida prevista, desde que esteja dentro da região de validação, como a única observação válida e as restantes são rejeitadas, ver figura 5.4.

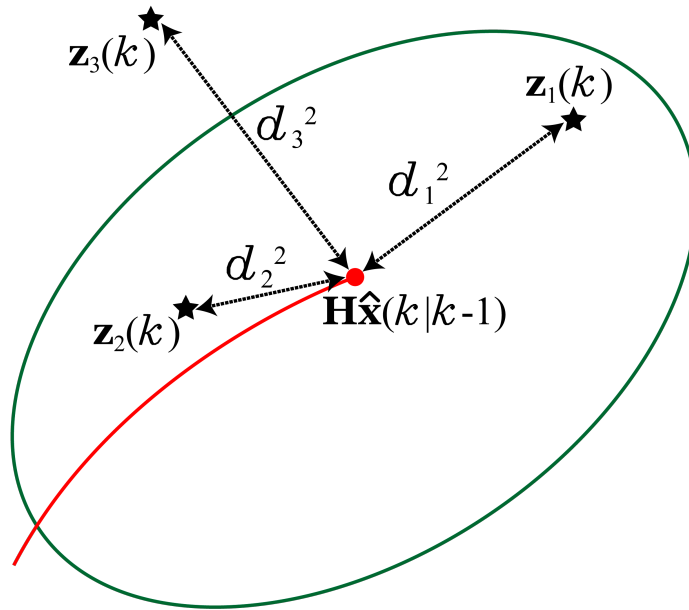


Figura 5.4: *Nearest-Neighbour Standard Filter* com apenas um *target track*

A única medição validada é usada para fazer *update* ao estado estimado do alvo. Se não houverem observações com valores de inovação normalizada inferiores ao definido por d_{ij}^2 , então nenhuma observação é associada.

Há alguns problemas com o algoritmo básico do NNSF, particularmente em situações onde há muitos falsos positivos provocados por ruído e onde há potencialmente vários alvos próximos. A figura 5.5 mostra uma situação particular em que duas regiões de validação se sobrepõem. Neste caso, associar correctamente uma observação à trajectória correspondente é complexo. Primeiro porque os pares com associação mais próxima não são necessariamente os correctos e segundo, porque a associação é dependente da ordem e portanto todas as associações devem ser consideradas no seu conjunto para se conseguir uma associação correcta.

Como neste trabalho o ruído foi previamente reduzido e a estrutura dos robôs reduz a probabilidade de haver sobreposições de regiões de validação, este foi o método escolhido para a associação de dados, sendo que caso haja alguma observação dentro de mais do que uma região de validação, essa observação é descartada. No entanto, existem dois algoritmos para ultrapassar este tipo de problemas identificados no NNSF.

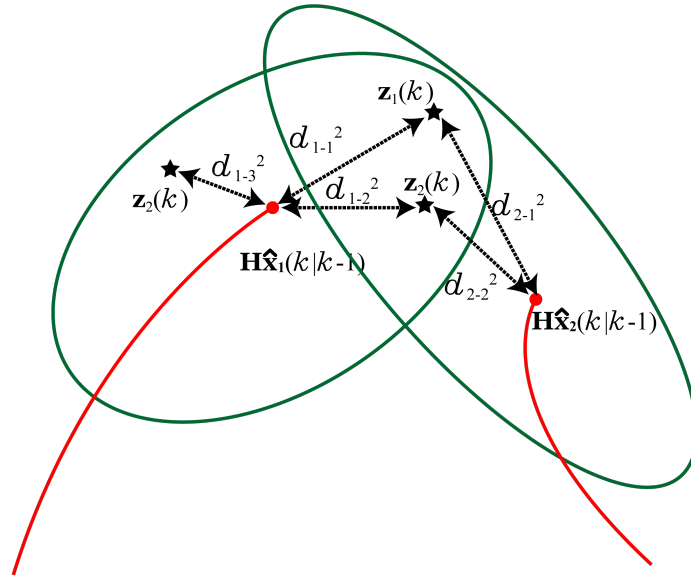


Figura 5.5: *Nearest-Neighbour Standard Filter* com vários *targets*

Probabilistic Data Association Filter

O *Probabilistic Data Association Filter* (PDAF) e o *Joint Probabilistic Data Association Filter* (JPDAF) são algoritmos Bayesianos que computam a probabilidade de associação correcta entre uma observação e a trajectória. Esta probabilidade de associação correcta é calculada para cada observação no interior da região de validação que é de seguida usada para formar uma “trajectória correcta” através de uma média pesada de todas as observações válidas.

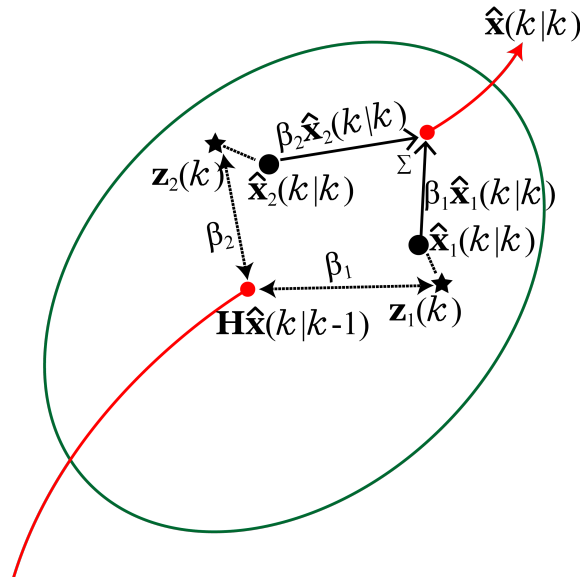


Figura 5.6: *Probabilistic Data Association Filter*

A probabilidade de associação correcta β_i é dada por:

$$\beta_i(k) \equiv P[\theta_i(k) | Z^k], \quad i = 0, 1, \dots, m_k \quad (5.26)$$

- Onde β_i é a probabilidade de associação correcta, $\theta_i(k)$ é o evento que a medição $\mathbf{z}_i(k)$ originou a partir do alvo.

A média condicional do estado no instante k pode ser escrita como:

$$\hat{\mathbf{x}}(k | k) = \sum_{i=0}^{m_k} \hat{\mathbf{x}}_i(k | k) \beta_i(k) \quad (5.27)$$

- Onde $\hat{\mathbf{x}}_i(k | k)$ é o estado actualizado condicionado ao evento $\theta_i(k)$.

$$\hat{\mathbf{x}}_i(k | k) = \hat{\mathbf{x}}(k | k-1) + \mathbf{W}_i(k) \nu_i(k), \quad i = 1, \dots, m_k \quad (5.28)$$

onde

$$\nu_i(k) = \mathbf{z}_i(k) - \mathbf{H}_i(k) \hat{\mathbf{x}}(k | k-1) \quad (5.29)$$

Multiple Hypothesis Tracking Filter

O *Multiple Hypothesis Tracking Filter* (MHTF) mantém trajetórias separadas para cada possível associação de observação. A cada instante de tempo, a observação prevista é usada para estabelecer uma região de validação, ver figura 5.7.

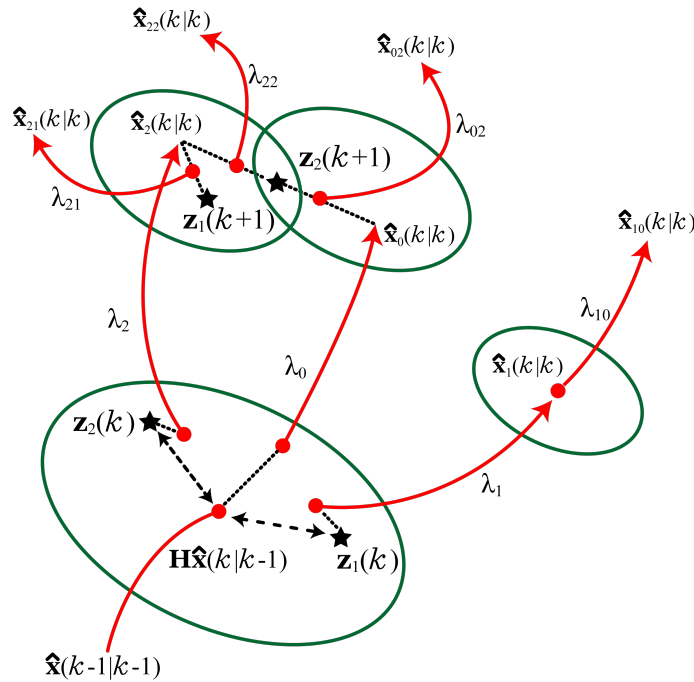


Figura 5.7: *Multiple Hypothesis Tracking Filter*

Tal como se pode observar na figura 5.7, no MHTF todas as observações validadas $\mathbf{z}_p(k)$ são usadas para estabelecer uma nova trajetória $\hat{\mathbf{x}}_p(k | k)$, criando uma ramificação com várias possíveis trajetórias entre as quais poderão estar trajetórias que realmente existem e trajetórias provocadas por falsos positivos.

Capítulo 6

Implementação

6.1	Arquitectura do sistema	39
6.2	Calibração de parâmetros intrínsecos e extrínsecos das câmaras	40
6.2.1	Calibração dos parâmetros intrínsecos das câmaras	41
6.2.2	Calibração dos parâmetros extrínsecos	43
6.2.3	Erro associado ao <i>stereo</i>	45
6.3	Captura de imagens	46
6.3.1	Eliminação dos falsos pontos de interesse	48
6.4	Aplicação do filtro de <i>Kalman</i>	51
6.4.1	Inicializações	52
6.4.2	Previsão	53
6.4.3	<i>Update</i>	54

Neste capítulo pretende-se propor uma arquitectura para o sistema, descrever o trabalho realizado durante a implementação do sistema e por fim serão demonstrados os resultados obtidos.

6.1 Arquitectura do sistema

Na figura 6.1 está representada a arquitectura do sistema. O sistema deve suportar n pares de câmaras, umas para capturar imagens de forma a que seja possível identificar pontos com intensidade luminosa superior a outras partes da imagem e outras para identificar outro tipo de *features* para os objectos em que não se podem colocar LEDs. As imagens de cada câmara passam por um bloco de detecção responsável por identificar pontos de interesse e *features*. Estes pontos de interesse a duas dimensões com referencial nas imagens das câmaras são convertidos em pontos tridimensionais com referencial no mundo através da triangulação *stereo*, que em seguida passam por um bloco de reconhecimento,

responsável por identificar todos os pontos que pertencem a cada marcador e excluir os restantes. De seguida, a posição e atitude de cada um dos marcadores identificados é enviado para o filtro de *Multi-target tracking* (MTT) que estima o estado dos vários objectos. O estado dos objectos, além de ser armazenado para uso posterior ou de ser enviado em tempo real para aplicações que utilizem essa informação, também são usados para identificar zonas das imagens em que se tem maior interesse procurar por LEDs e *features*. Para isso, as coordenadas 3D dos objectos têm que ser projectadas para o plano da imagem 2D com a ajuda dos parâmetros intrínsecos e extrínsecos que constituem também a calibração *stereo*.

Na base de dados de descritores estão informações relativas a descrição dos marcadores e *features*, como por exemplo, intensidade dos pontos luminosos, número de pontos luminosos que constituem um marcador, distância entre esses pontos, forma e cor das *features*, etc.

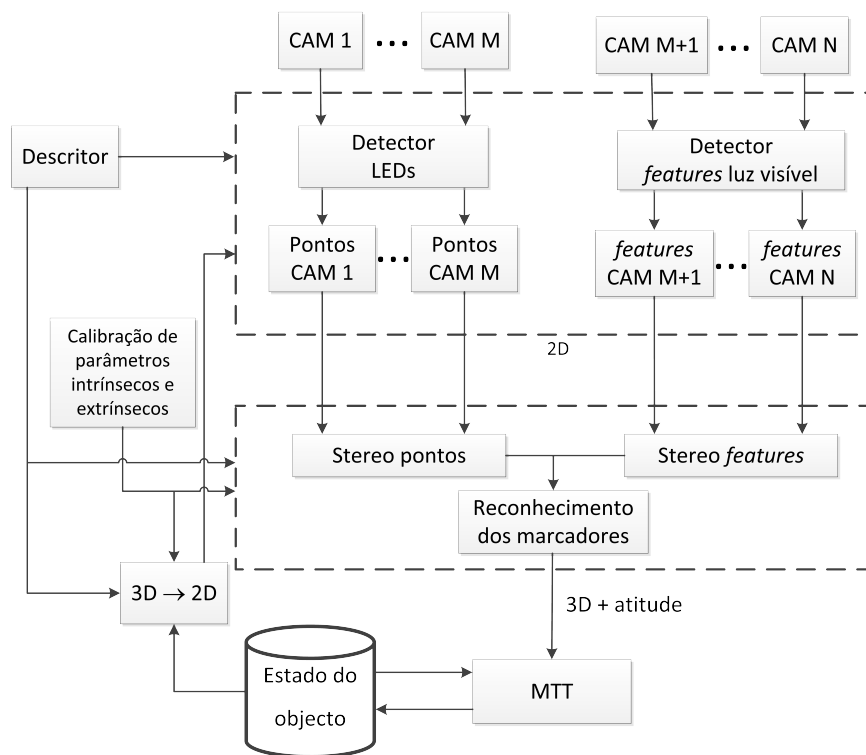


Figura 6.1: Arquitectura do sistema

Este trabalho irá focar-se na configuração do sistema *stereo*, detecção do marcador de LEDs e desenvolvimento do MTT.

6.2 Calibração de parâmetros intrínsecos e extrínsecos das câmaras

Para que haja a possibilidade da reconstrução 3D de uma imagem, é preciso levar em consideração dois tipos de parâmetros: os parâmetros intrínsecos e os parâmetros extrínsecos.

Os parâmetros intrínsecos permitem transformar as coordenadas da câmara em coordenadas da

imagem em pixels.

Os parâmetros extrínsecos permitem definir a localização e orientação do referencial da câmara em relação ao referencial do mundo, ou seja, transforma as coordenadas do objecto em coordenadas no referencial da câmara.

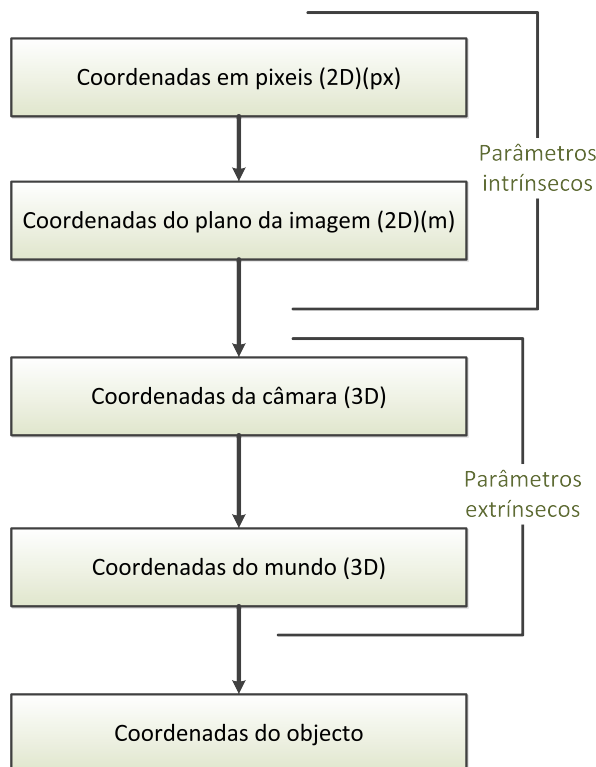


Figura 6.2: Transformações dos sistemas de coordenadas durante uma reconstrução 3D a partir de coordenadas em pixels

6.2.1 Calibração dos parâmetros intrínsecos das câmaras

O processo de calibração dos parâmetros intrínsecos tem como objectivo o cálculo dos parâmetros que permitem transformar as coordenadas da câmara em coordenadas da imagem em pixels, minimizando também os efeitos provocados pela distorção da lente, tema abordado no capítulo 4.3 Distorção provocada pela lente, página 19, devendo ser efectuada em cada uma das câmaras utilizadas.

Durante este trabalho, foi utilizada uma ferramenta de calibração de câmaras para MATLAB [41], que permite obter os diversos parâmetros da câmara, a partir da introdução de uma série de imagens com um padrão axadrezado, a dimensão de cada quadrado e o número de quadrados em cada lado. Dos vários parâmetros devolvidos pela ferramenta, destaque-se, os erros de projecção dos cantos do xadrez, o centro da imagem da câmara em pixels, cc , a distância focal, f , e o factor de correlação, kc . Com base nesta informação é possível calcular a matriz dos parâmetros intrínsecos, K . [42][43]

No que diz respeito à câmara direita, pode-se observar na figura 6.3 que, o erro de projecção dos cantos do padrão axadrezado obtido no final do processo se encontra limitado a um intervalo de ± 1.5 pixels.

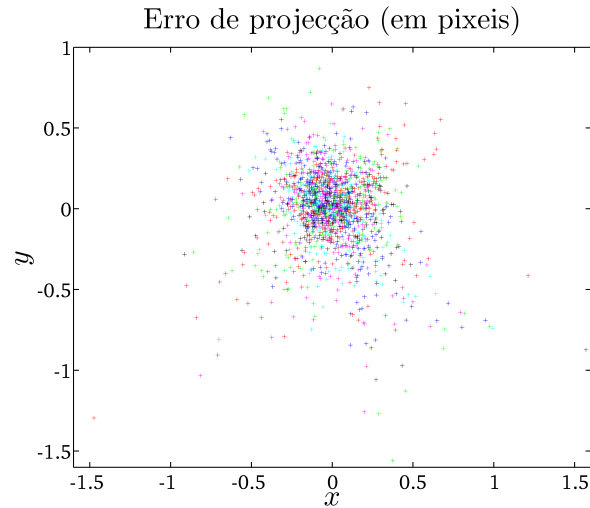


Figura 6.3: Erro de projecção x, y da câmara direita, em pixels

Este é um erro geométrico, que corresponde à distância na imagem entre o ponto projectado e o medido, e quantifica a qualidade da estimativa de um ponto tridimensional usado para recriar a verdadeira projecção do ponto.

O centro da imagem da câmara em pixels, a distância focal e o factor de correlação da câmara direita obtidos foram respectivamente:

$$cc_r = \begin{bmatrix} 652.8887 \\ 483.5179 \end{bmatrix} \quad (6.1)$$

$$f_r = \begin{bmatrix} 984.2191 \\ 976.0186 \end{bmatrix} \quad (6.2)$$

$$kc_r = \begin{bmatrix} -0.2328 \\ 0.0854 \\ -0.0043 \\ -0.0025 \\ 0 \end{bmatrix} \quad (6.3)$$

A matriz de parâmetros intrínsecos fica portanto:

$$K_r = \begin{bmatrix} 984.2191 & 0 & 652.8887 \\ 0 & 976.0186 & 483.5179 \\ 0 & 0 & 1 \end{bmatrix} \quad (6.4)$$

Ao calibrar os parâmetros intrínsecos da câmara esquerda obtém-se:

$$cc_l = \begin{bmatrix} 678.2207 \\ 467.5577 \end{bmatrix} \quad (6.5)$$

$$f_l = \begin{bmatrix} 966.0394 \\ 961.3543 \end{bmatrix} \quad (6.6)$$

$$kc_l = \begin{bmatrix} -0.2242 \\ 0.0786 \\ -0.0014 \\ -0.0001 \\ 0 \end{bmatrix} \quad (6.7)$$

$$K_l = \begin{bmatrix} 966.0394 & 0 & 678.2207 \\ 0 & 961.3543 & 467.5577 \\ 0 & 0 & 1 \end{bmatrix} \quad (6.8)$$

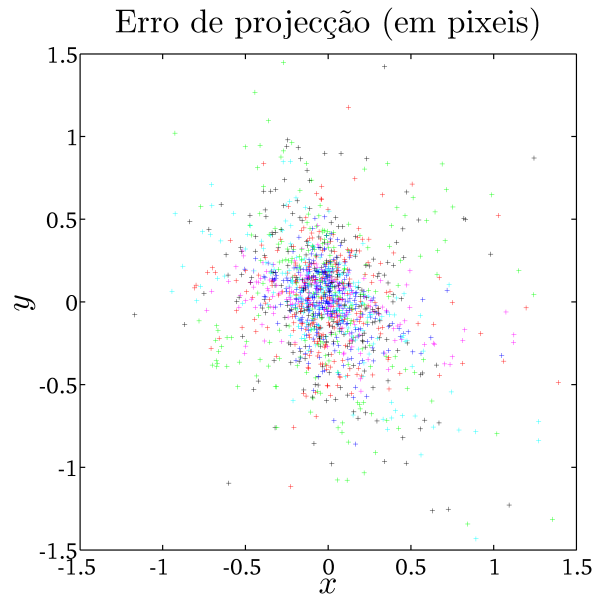


Figura 6.4: Erro de projecção x, y da câmara esquerda, em pixels

Na figura 6.5 pode-se observar uma comparação entre as imagens tiradas por cada uma das câmaras sem qualquer tipo de processamento e imagens cujos efeitos de distorção das lentes foram reduzidos com o apoio dos parâmetros intrínsecos. Ao longo deste trabalho, este processo não será efectuado a toda a imagem mas sim apenas aos pontos de interesse visando reduzir o peso computacional do programa em geral.

6.2.2 Calibração dos parâmetros extrínsecos

Os parâmetros extrínsecos permitem relacionar segundo uma rotação (R_{ext}) e uma translação (T_{ext}), o referencial do mundo e o referencial da câmara.

Para determinar matrizes R_{ext} e T_{ext} de cada uma das câmaras utilizadas no sistema, são utilizados vários pontos no referencial do mundo com as coordenadas em pixels no referencial da imagem.

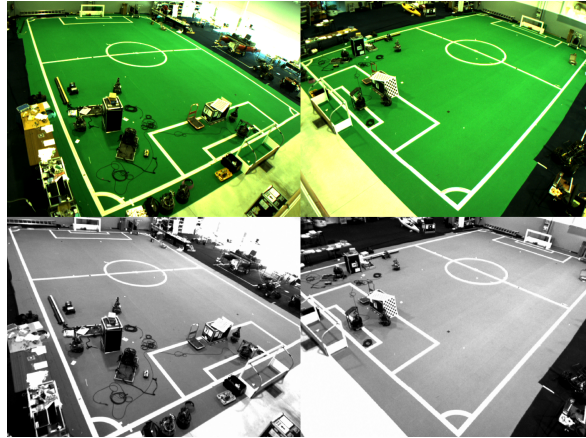


Figura 6.5: Comparação entre as imagens das câmaras esquerda e câmara direita (respectivamente na parte esquerda e direita) com distorção (na parte superior) e sem distorção (imagens na parte inferior, em monocromático)

Para que esta correspondência seja feita são escolhidos alguns pontos do campo escolhidos como referências, e são medidas as suas distâncias a 3D, em metros, em relação à origem do referencial do mundo, neste caso, o centro do campo e posteriormente, indica-se para cada um destes pontos escolhidos como referências, as coordenadas em pixels nos planos de imagens existentes a 2D.

Na figura 6.6 pode-se observar o conjunto de pontos escolhidos para se proceder ao cálculo dos parâmetros extrínsecos assim como a disposição das câmaras existentes no local.

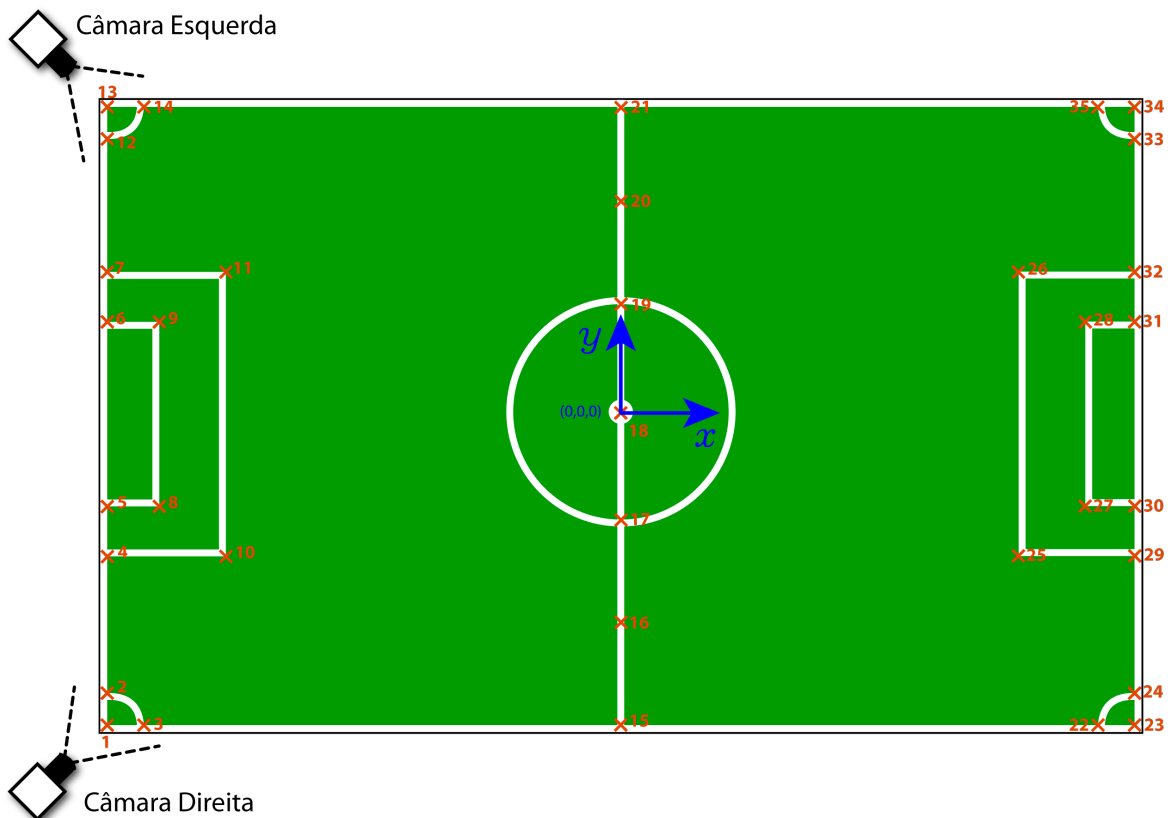


Figura 6.6: Pontos escolhidos para o cálculo dos parâmetros extrínsecos e disposição das câmaras

As correspondências entre as medidas, em metros, obtidas para cada um destes pontos e as coordenadas em pixels nas imagens tiradas em cada uma das câmaras existentes, podem ser vistas na tabela A.1 (pág. 72).

Partindo destas correspondências de pontos 2D e 3D, é possível determinar os parâmetros extrínsecos ($Rect$, $Text$) que minimizam o erro entre os pontos 2D na imagem e a projecção dos pontos 3D na imagem (usando os parâmetros intrínsecos da câmara determinados na secção anterior), através de um processo de optimização.

As matrizes de translação e rotação para a câmara direita são respectivamente a (6.9) e (6.10), enquanto que as da câmara esquerda podem ser vistas nas equações (6.11) e (6.12).

$$Text_r = \begin{bmatrix} 1.6536 & -3.6188 & 14.2908 \end{bmatrix} \quad (6.9)$$

$$Rect_r = \begin{bmatrix} 0.6606 & -0.7455 & 0.0880 \\ -0.4082 & -0.4552 & -0.7913 \\ 0.6300 & 0.4868 & -0.6050 \end{bmatrix} \quad (6.10)$$

$$Text_l = \begin{bmatrix} -1.6486 & -3.2646 & 14.1452 \end{bmatrix} \quad (6.11)$$

$$Rect_l = \begin{bmatrix} -0.5512 & -0.8135 & 0.1857 \\ -0.5588 & 0.1946 & -0.8061 \\ 0.6196 & -0.5481 & -0.5618 \end{bmatrix} \quad (6.12)$$

6.2.3 Erro associado ao *stereo*

Foi realizado um estudo para se saber qual seria o erro introduzido pelo sistema *stereo* utilizado ao longo deste trabalho, nas medidas obtidas pelo mesmo. A partir de uma grelha de pontos, no referencial do campo, de coordenadas conhecidas e espaçados $1m$ entre si, indicou-se um valor de disparidade, obtido através do desvio padrão de medidas obtidas pelo *stereo*. Projectou-se cada um dos pontos do campo pertencentes à grelha nas imagens rectificadas das câmaras com recurso às matrizes de projecção (6.13) e (6.14).

$$p_r = K_r \times \begin{bmatrix} Rect_r & Text_r^\top \end{bmatrix} \quad (6.13)$$

$$p_l = K_l \times \begin{bmatrix} Rect_l & Text_l^\top \end{bmatrix} \quad (6.14)$$

Seguidamente, utiliza-se os pontos resultantes da projecção nas duas imagens, para determinar um ponto 3D resultante da triangulação desses pontos, mas afectando-se um dos pontos de um erro, idêntico ao erro definido para a detecção de *features*, segundo a linha epipolar. Regista-se o pior caso do erro para a respectiva posição 3D.

O erro introduzido pelo sistema *stereo* nas diferentes zonas do campo pode ser visto na figura 6.7.

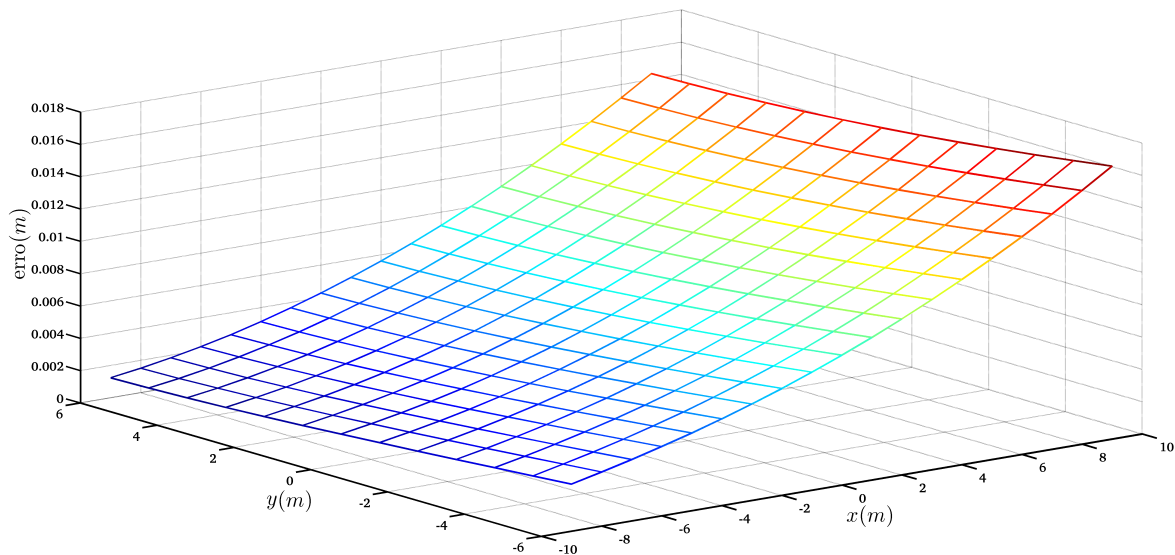


Figura 6.7: Erro das medidas 3D obtidas nas diferentes zonas do campo

Como se pode observar, o erro cresce à medida que nos afastamos das câmaras, sendo que dentro do campo o erro é sempre inferior a $1,7\text{cm}$.

Para se reduzir o erro máximo, poder-se-á, por exemplo, aplicar outro par de câmaras no outro extremo do campo e até lateralmente.

6.3 Captura de imagens

No pavilhão do Laboratório de Sistemas Autónomos (LSA) da unidade de robótica do INESC TEC há actualmente duas câmaras, instaladas num plano superior, capazes de visualizar o campo por inteiro (ver figura 6.5). Tratam-se de duas câmaras *Basler aca1300-30gc* cuja resolução é de 1294×964 pixels, têm a capacidade de adquirir imagens com sincronismo externo a uma taxa de até 30FPS e possuem comunicação GigE.[44]

O sincronismo externo permite que através de um *trigger* disparado em simultâneo nas câmaras, estas capturem as imagens no mesmo instante de tempo. Esta característica é de extrema importância para que as medidas obtidas pelo sistema *stereo* sejam o mais exactas possível.

Uma vez que o objectivo é seguir objectos com marcas luminosas, há a necessidade de conseguir controlar o tempo de exposição durante a captura de imagens e assim reduzir o número de falsos pontos de interesse originadas por reflexos, iluminação natural e artificial, etc.

Estas operações de configuração são efectuadas recorrendo à *Application Programming Interface* (API) da *Basler*.

Cada imagem capturada é convertida para *Hue, Saturation, Value* (HSV) para que se identifique as diferenças da intensidade luminosa ao longo da imagem. De seguida, são procuradas manchas de pixels cuja luminosidade seja significativamente superior ao resto da imagem e calcula-se as coordenadas x, y em pixels do centro de massa dessas manchas.

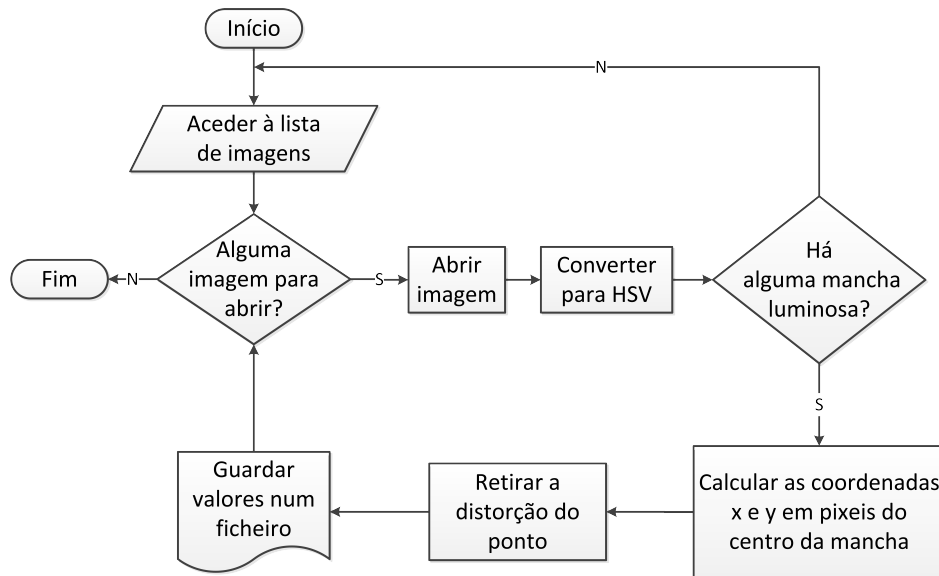


Figura 6.8: Fluxograma do procedimento para a obtenção das coordenadas x, y em pixels dos pontos de interesse

Durante a realização deste trabalho, foram colocados três LEDs numa base com a finalidade de simular um objecto que deveria ser seguido pelo sistema. Os LEDs em questão foram dispostos de maneira a que formassem um triângulo isósceles. Esta solução foi adoptada para que este objecto pudesse ser posteriormente distinguido de outros objectos capturados pelas câmaras, através da procura de um objecto, de medidas conhecidas, com três pontos distintos em que dois dos seus lados têm comprimentos iguais e o outro lado tem um comprimento inferior. Esta característica geométrica permite também identificar o ponto da frente, pois há apenas um ponto comum aos dois lados com maior comprimento e através dessa informação determinar a sua posição e atitude.

Tal como se pode ver na figura 6.9, mesmo reduzindo o tempo de exposição das câmaras, continuam a haver alguns falsos pontos de interesse (visualizados em tons de verde) e que são reduzidos posteriormente.

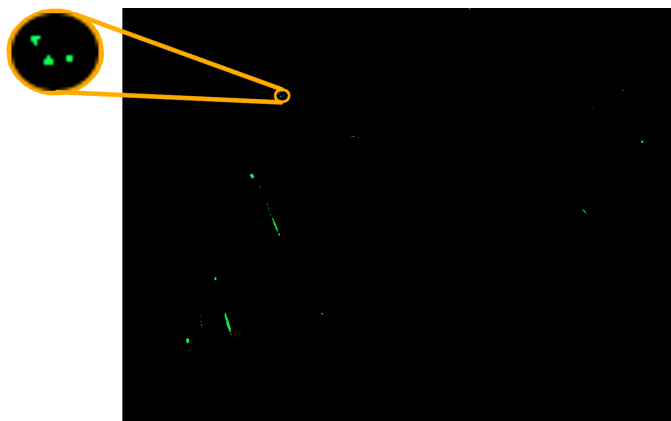


Figura 6.9: Exemplo de uma imagem capturada por uma câmara, em HSV

6.3.1 Eliminação dos falsos pontos de interesse

Após visualização de diversos pares de imagens, chega-se à conclusão de que a maior parte dos falsos pontos de interesse visíveis numa das câmaras, não teriam correspondência na outra.

Para se proceder à eliminação dos falsos pontos de interesse existentes nas imagens, definiu-se uma estratégia que consiste numa sequência de testes de validação recorrendo a restrição epipolar, restrição espacial e restrições geométricas do marcador.

Numa primeira fase, os pontos de interesse de cada imagem são agrupadas juntamente com outros pontos de interesse nas proximidades e cada um destes grupos, a que chamamos de *clusters*, recebe uma identificação numérica que os distingue dos restantes *clusters*. Para cada um dos *clusters* é calculado o seu ponto médio que é utilizado para gerar um conjunto de hipóteses *stereo*.

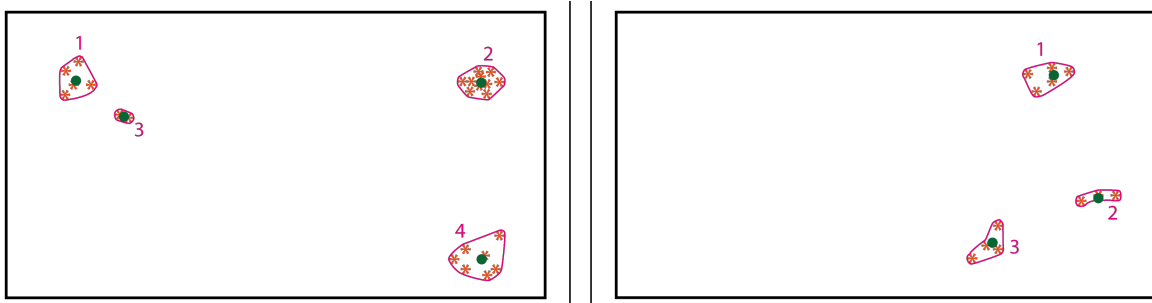


Figura 6.10: Exemplo de criação de *clusters*, suas identificações e cálculo do ponto central

Se nenhuma das linhas epipolares, calculadas com base nesses pontos médios, passar próxima de algum *cluster*, então os pontos de interesse contidos nesse *cluster* são considerados ruído e descartados.

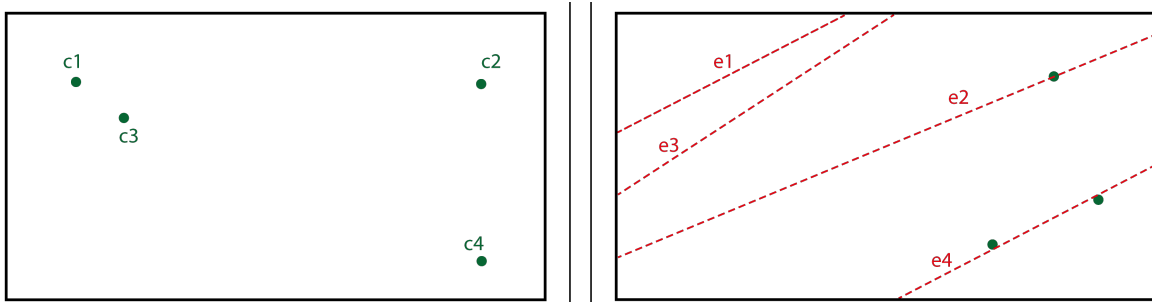


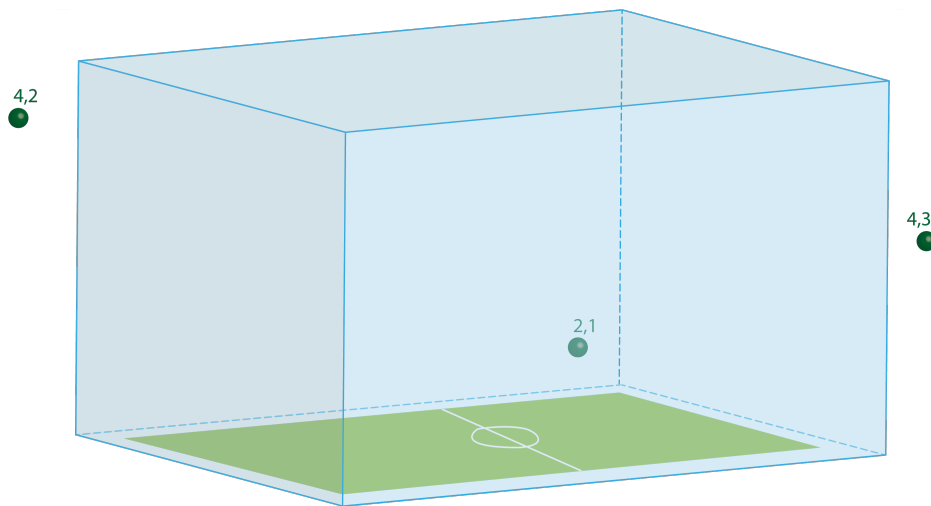
Figura 6.11: Traçado das linhas epipolares na câmara 2, baseadas nas posições dos pontos centrais dos *clusters* da câmara 1

Considere-se a título de exemplo as figuras 6.10 e 6.11. A partir dos pontos centrais dos *clusters* ilustrados como um pequeno círculo verde na figura 6.10 são traçadas as linhas epipolares na outra imagem. Como não há nenhum *cluster* próximo das linhas epipolares “e1” e “e3”, então os *clusters* 1 e 3 da imagem esquerda serão descartados. Por outro lado, relativamente às linhas “e2” e “e4” há pelo menos um *cluster* nas proximidades de cada uma dessas linhas sendo portanto criadas combinações entre esses *clusters*

Nº cluster camera 1	Nº cluster camera 2
1	—
2	1
3	—
4	2,3

Tabela 6.1: Correspondência entre os *clusters* das várias câmaras

Aos *clusters* que foram associados a outros *clusters* através da linha epipolar, verifica-se se o *stereo* dos pontos médios desses *clusters* dá origem a um ponto 3D que se situe dentro do volume do campo. Caso isso não aconteça, essa hipótese *stereo* deixa de ser considerada como uma hipótese válida e se os *clusters* envolvidos nessa validação não tiverem mais nenhuma hipótese *stereo* para ser testada, também esses pontos de interesse são descartados.

Figura 6.12: Cálculo posição 3D dos pontos centrais dos *clusters* para verificar se se situam dentro do volume válido

No exemplo adoptado, apenas a combinação *cluster 1 cluster 2* deram origem a um ponto dentro do volume considerado como válido. Por essa razão, os *clusters 4* da imagem esquerda e os 2 e 3 da imagem direita são descartados assim como todos os pontos de interesse englobados nesses *clusters*.

De seguida, todos os pontos de interesse contidos num *cluster* que não tenha sido rejeitado nos passos anteriormente descritos, são testadas. Para isso, são calculadas linhas epipolares de cada uma dos pontos de interesse de um *cluster* e verificado se no *cluster* correspondente da outra imagem, existe ou não um ponto de interesse correspondente. Nesta fase, a existência de falsos pontos de interesse já é bastante inferior. São calculadas as localizações 3D de cada um dos pontos de interesse restantes e a esses pontos 3D calculados, verifica-se se são ou não compatíveis com o modelo criado.

Para verificar se os pontos de interesse correspondem ao modelo utilizado, é calculado o seu ponto central do conjunto de *features*, são identificados os pontos da frente, esquerda e direita e calculadas as distancias entre eles para se ver se realmente correspondem com o modelo. A identificação do ponto central, frente, esquerda e direita é conseguida com o algoritmo 1.

Para auxiliar à sua compreensão, na figura 6.13 está a disposição dos pontos e respectiva identifi-

cação.

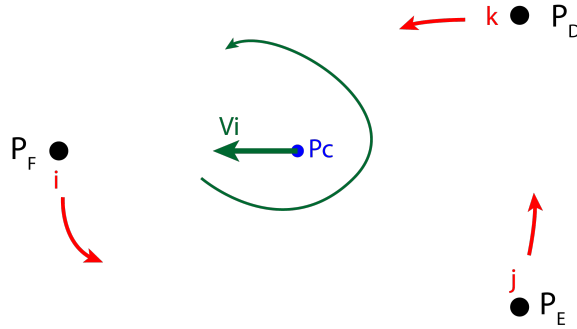


Figura 6.13: Identificação dos pontos do marcador

Algorithm 1 Verificação do modelo

```

 $Pc \leftarrow \frac{1}{n} \sum_{i=1}^n P(i)$ 
 $Minimo \leftarrow -1$ 
for  $i \leftarrow 1, i = n, i \leftarrow i + 1$  do
     $Vi(i) \leftarrow \frac{P(i) - Centro}{norm(P(i) - Pc)}$ 
end for
 $j \leftarrow 1$ 
 $k \leftarrow 2$ 
for  $i \leftarrow 1, i = n, i \leftarrow i + 1$  do
     $j \leftarrow j + 1$ 
     $k \leftarrow k + 1$ 
    if  $j = 4$  then
         $j \leftarrow 1$ 
    end if
    if  $k \geq 4$  then
         $k \leftarrow 1$ 
    end if
    if  $Vi(i).Vi(j) > Minimo$  then
         $Tras1 \leftarrow i$ 
         $Tras2 \leftarrow j$ 
         $Frente \leftarrow k$ 
         $Minimo \leftarrow Vi(i).Vi(j)$ 
    end if
end for
 $Vfrente \leftarrow Vi(Frente)$ 
 $Pfrente \leftarrow P(Frente)$ 
 $Vppf \leftarrow Vfrente \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$ 
if  $Vppf.Vi(Tras1) < 0$  then
     $Pesquerda \leftarrow P(Tras1)$ 
     $Pdireita \leftarrow P(Tras2)$ 
else
     $Pesquerda \leftarrow P(Tras2)$ 
     $Pdireita \leftarrow P(Tras1)$ 
end if

```

6.4 Aplicação do filtro de *Kalman*

Quando surge a primeira observação é inicializada uma nova trajectória. Cada trajectória possui um conjunto de informações úteis que as distinguem de outras trajectórias existentes e permitem que o filtro actue correctamente em cada uma das trajectórias existentes. A cada trajectória é portanto atribuída uma ID, um *type*, *time*, *state predict*, *state estimate*, *uncertainty predict*, *uncertainty estimate*, *innovation* e *innovation variance*. A ID é uma indicação numérica incremental para as diferenciar de outras trajectórias. Tem também o objectivo de identificar trajectórias que estejam actualmente inactivas, situação que pode ocorrer por exemplo quando é inicializada uma trajectória que mais tarde se percebe que era composta por falsos pontos de interesse. Nestes casos é atribuído uma ID negativa. O *type* tem a finalidade de permitir que seja identificado o tipo de veículo que originou aquela trajectória. Isto poderá ser útil para melhorar os parâmetros do filtro através da especificação de parâmetros para diferentes tipos de veículos. O *time* indica qual foi o último instante de tempo em que houve alguma alteração nos valores calculados relativamente aquela trajectória. Os restantes itens, tal como os nomes indicam, dizem respeito respectivamente à previsão e estimativa do estado, incertezas de previsão e do estado, inovação e variância da inovação, todos eles calculados pelo KF.

Quando aparece outra observação é necessário proceder à sua associação. A associação consiste em determinar, com a ajuda de uma região de validação, a qual trajectória pertence cada observação. Nesta fase, podem ocorrer três situações distintas: a observação está dentro de uma região de validação e é associada à trajectória correspondente, ou a observação está dentro de mais do que uma região de validação e é excluída por se considerar que é ambígua ou então, a observação não está dentro de nenhuma região de validação existente e considera-se esta observação como uma possível nova trajectória. Caso esta última hipótese se verifique, é criado um candidato que permitirá decidir se esta nova observação dará ou não origem à inicialização de uma nova trajectória. Para que isso aconteça, o candidato tem que receber três observações num determinado intervalo de tempo. Se se ultrapassar esse intervalo de tempo o candidato é excluído. Para avaliar se as observações pertencem ou não aquele candidato usa-se, mais uma vez, uma região de validação. Os candidatos são constituídos por um ID que os distinguem quer de outros candidatos quer de candidatos excluídos, um *time* que dá a indicação de qual foi o último instante de tempo em que entrou uma nova observação para aquele candidato assim como as informações relativas às posições e atitudes das observações. As informações relativas às posições e atitudes das observações incluídas no candidato são utilizadas para a inicialização da nova trajectória se isso vier a acontecer. Assim que o candidato recebe as três observações, a nova trajectória é inicializada e o candidato é considerado inactivo.

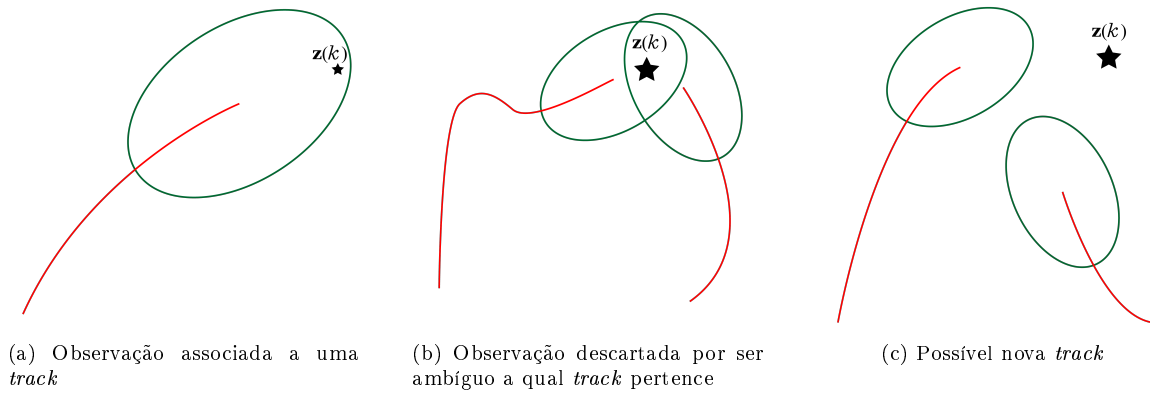


Figura 6.14: Diferentes casos de associação de observações através da região de validação

De resto, a implementação deste filtro segue a formulação descrita no capítulo 5.3.1 Filtro de *Kalman* multi-sensorial, página 31, utilizando-se o método dos sensores agrupados e a associação é do tipo NNSF.

6.4.1 Inicializações

A inicialização é a primeira fase do filtro.

$$\hat{\mathbf{x}}(k | k) = \begin{bmatrix} x \\ v_x \\ y \\ v_y \\ z \\ v_z \end{bmatrix} \quad (6.15)$$

$$\mathbf{Q} = \begin{bmatrix} \theta_x^2 & 0 & 0 \\ 0 & \theta_y^2 & 0 \\ 0 & 0 & \theta_z^2 \end{bmatrix} \quad (6.16)$$

O estado é inicializado com a posição obtida na primeira observação ao estado e velocidade zero.

$$\hat{\mathbf{x}}(k | k) = \begin{bmatrix} x_0 \\ 0 \\ y_0 \\ 0 \\ z_0 \\ 0 \end{bmatrix} \quad (6.17)$$

A covariância do processo foi inicializada com um desvio padrão de $0.35m$.

$$\mathbf{Q} = \begin{bmatrix} 0.35^2 & 0 & 0 \\ 0 & 0.35^2 & 0 \\ 0 & 0 & 0.35^2 \end{bmatrix} \quad (6.18)$$

A covariância do estado foi inicializada por:

$$\mathbf{P}(k | k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \end{bmatrix} \quad (6.19)$$

6.4.2 Previsão

A previsão do estado é dada por:

$$\hat{\mathbf{x}}(k | k-1) = \mathbf{F}(k)\hat{\mathbf{x}}(k-1 | k-1) \quad (6.20)$$

e considera-se que o objecto segue um movimento uniforme

$$\mathbf{F} = \begin{bmatrix} 1 & dt & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & dt & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.21)$$

A covariância da previsão é dada por:

$$\mathbf{P}(k | k-1) = \mathbf{F}(k)\mathbf{P}(k-1 | k-1)\mathbf{F}^T(k) + \mathbf{G}(k)\mathbf{Q}(k)\mathbf{G}^T(k) \quad (6.22)$$

onde

$$\mathbf{G} = \begin{bmatrix} \frac{dt^2}{2} & 0 & 0 \\ dt & 0 & 0 \\ 0 & \frac{dt^2}{2} & 0 \\ 0 & dt & 0 \\ 0 & 0 & \frac{dt^2}{2} \\ 0 & 0 & dt \end{bmatrix} \quad (6.23)$$

6.4.3 Update

O filtro implementado tem a particularidade de que se na fase de *update* não houverem observações durante um determinado período de tempo, neste caso $2s$, associadas a uma certa trajetória, essa trajetória é considerada inactiva. A região de validação é definida pela inversa da função distribuída acumulada da distribuição qui-quadrado com três graus de liberdade e probabilidade de 0,6.

Como se utiliza o método de sensores sequenciais, as equações na fase de *update* são repetidas por cada sistema *stereo*.

A inovação é a diferença entre a observação e a observação prevista:

$$\nu(k) = \mathbf{z}(k) - \hat{\mathbf{z}}(k | k - 1) \quad (6.24)$$

onde a observação prevista é calculada por:

$$\hat{\mathbf{z}}(k | k - 1) = \mathbf{H}(k)\hat{\mathbf{x}}(k | k - 1) \quad (6.25)$$

$$\mathbf{z}(k) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (6.26)$$

e a matriz de observação é

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (6.27)$$

A covariância da inovação é calculada por:

$$\mathbf{S}(k) = \mathbf{R}(k) + \mathbf{H}(k)\mathbf{P}(k | k - 1)\mathbf{H}^T(k) \quad (6.28)$$

com covariância nas medidas de

$$\mathbf{R} = \begin{bmatrix} 0.00001 & 0 & 0 \\ 0 & 0.00001 & 0 \\ 0 & 0 & 0.00001 \end{bmatrix} \quad (6.29)$$

O ganho de *Kalman* é dado por:

$$\mathbf{W}(k) = \mathbf{P}(k | k - 1)\mathbf{H}^T(k)\mathbf{S}^{-1}(k) \quad (6.30)$$

O estado estimado passa a ser:

$$\hat{\mathbf{x}}(k | k) = \hat{\mathbf{x}}(k | k - 1) + \mathbf{W}(k)\nu(k) \quad (6.31)$$

e a respectiva covariância,

$$\mathbf{P}(k | k) = \mathbf{P}(k | k - 1) - \mathbf{W}(k)\mathbf{S}(k)\mathbf{W}^T(k) \quad (6.32)$$

Capítulo 7

Resultados

7.1 Resultados	57
7.2 Objecto estático	58
7.3 Objecto em movimento	60

7.1 Resultados

Os resultados aqui descritos foram obtidos por duas câmaras que formam um sistema *stereo*. As câmaras utilizadas foram as *Basler acA1300-30gc*, cuja resolução é de 1294×964 pixels, colocadas em dois dos cantos do campo, num plano superior, para abranger a maior área possível. A taxa de captura utilizada foi de $10Hz$.



Figura 7.1: Câmaras usadas no trabalho

O marcador com LEDs utilizado é apenas um protótipo. Os LEDs estão colocados numa placa

de acrílico que transporta para além dos três LEDs uma bateria e um conversor Corrente contínua (DC)/DC.

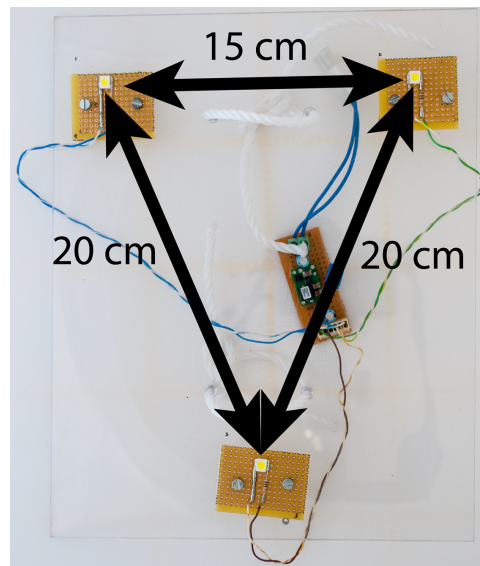


Figura 7.2: Protótipo do marcador

Esta base permite não ter o marcador preso a um único objecto ou robô sendo possível transportá-lo facilmente.

7.2 Objecto estático

Com o objectivo de analisar a repetibilidade da posição indicada pelo sistema para cada um dos três LEDs do objecto sem recorrer ao filtro MTT, coloquei o marcador de LEDs numa determinada posição do campo e foram registadas posições 3D, no referencial do mundo, de cada um dos LEDs ao longo do tempo.

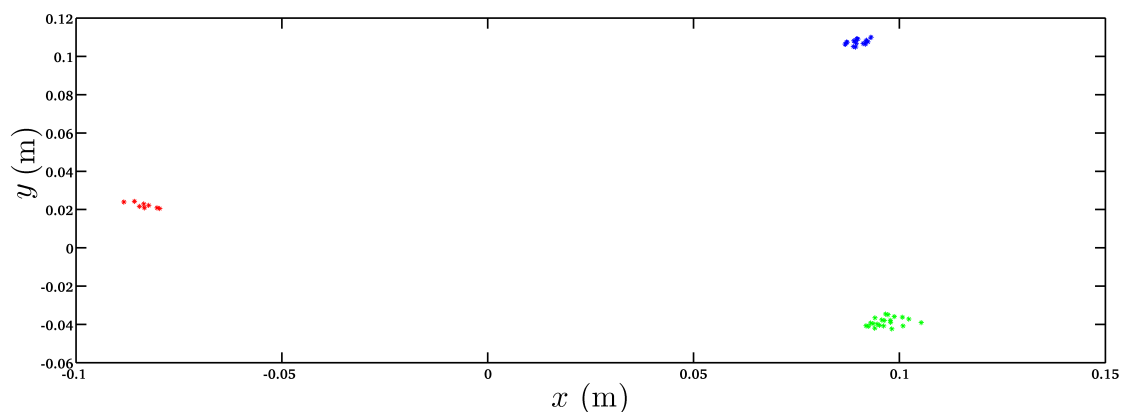


Figura 7.3: Posição dos LEDs do marcador segundo os eixos x, y , em modo estático

Na figura 7.3, pode-se observar a posição dos LEDs do marcador segundo os eixos x, y . Os LEDs da frente, esquerda e direita estão representados, respectivamente, pelas cores vermelha, verde e azul e verifica-se que a repetibilidade é bastante elevada.

O cálculo dos desvios padrão comprovam isso mesmo:

$$\sigma_{frt_x} = 0.000787m \quad (7.1a) \quad \sigma_{esq_x} = 0.0016m \quad (7.1b) \quad \sigma_{dir_x} = 0.000928m \quad (7.1c)$$

$$\sigma_{frt_y} = 0.000551m \quad (7.1d) \quad \sigma_{esq_y} = 0.0011m \quad (7.1e) \quad \sigma_{dir_y} = 0.000703m \quad (7.1f)$$

A figura 7.4 permite visualizar a repetibilidade ao longo do eixo do z sendo esta igualmente elevada.

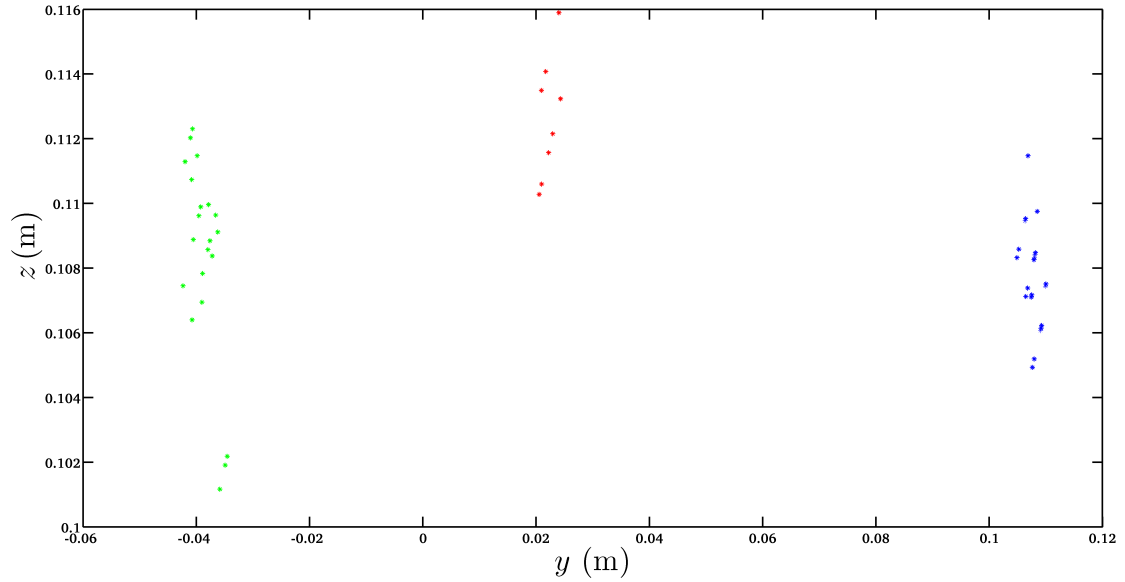


Figura 7.4: Posição dos LEDs do marcador segundo os eixos y, z , em modo estático

$$\sigma_{frt_z} = 0.000706m \quad (7.2a) \quad \sigma_{esq_z} = 0.0013m \quad (7.2b) \quad \sigma_{dir_z} = 0.000883m \quad (7.2c)$$

Esta pequena variação na posição de cada um dos LEDs do marcador provoca, naturalmente, uma variação na orientação do objecto sem que esta realmente aconteça. Na figura 7.5 pode-se ver como variam os ângulos de Euler ao longo do tempo numa situação em que o objecto está estático.

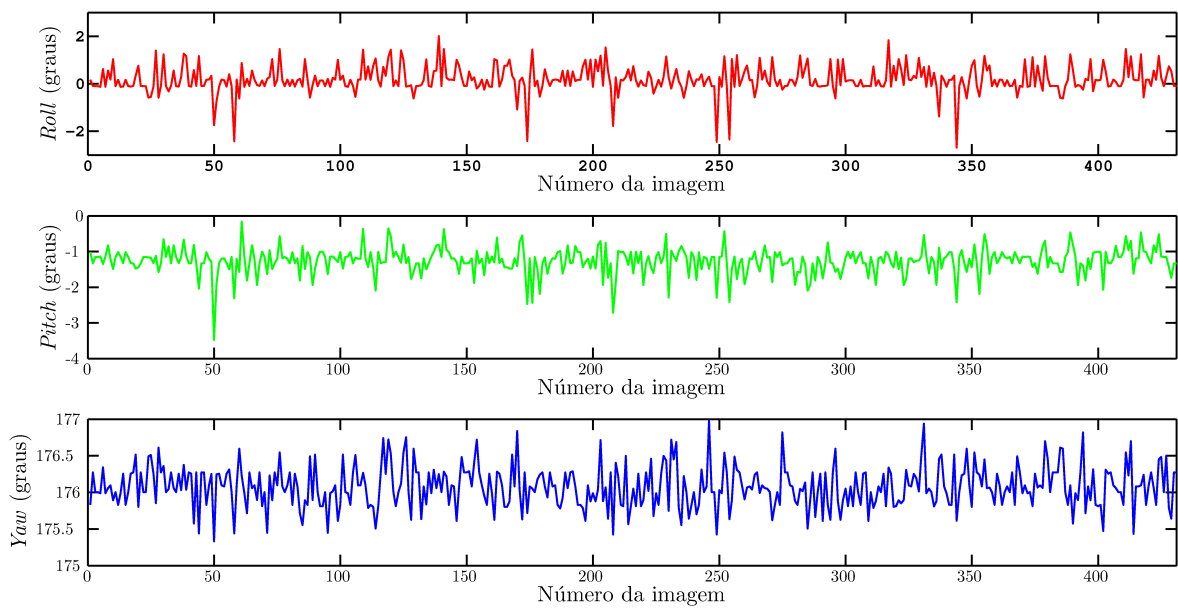


Figura 7.5: Variação do *Roll Pitch Yaw* do objecto, em modo estático

Tal como se pode observar, quer recorrendo à observação da figura 7.5 quer recorrendo ao cálculo dos desvios padrão dos ângulos de Euler (7.3), a variação não é significativa.

$$\sigma_{roll} = 0.5718^\circ \quad (7.3a)$$

$$\sigma_{pitch} = 0.3642^\circ \quad (7.3b)$$

$$\sigma_{yaw} = 0.2828^\circ \quad (7.3c)$$

7.3 Objecto em movimento

Os próximos resultados são relativos à detecção de objectos em movimento em três situações distintas que revelam alguns comportamentos do filtro que se devem destacar.

O marcador foi colocado no topo dos robôs tal como ilustra a figura 7.6.

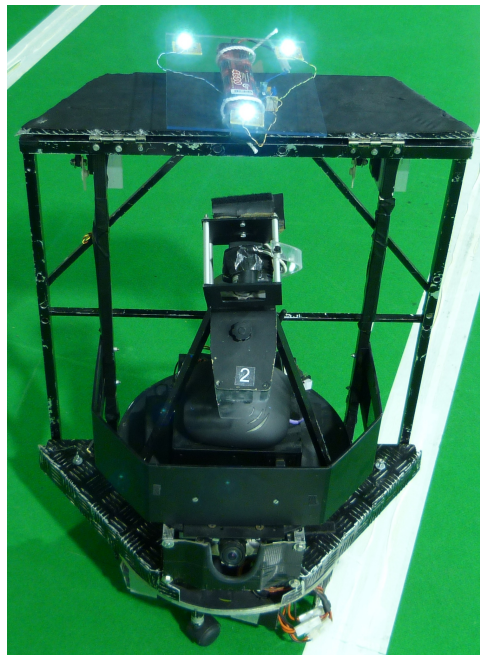


Figura 7.6: Posicionamento do marcador no robô

Na figura 7.7 está representado o centro do marcador ao longo de uma trajetória. No entanto no canto inferior esquerdo estão alguns falsos positivos. Estes falsos positivos são ruído que pode ter sido originado por reflexos, por exemplo, em superfícies metálicas ou até nas linhas do campo, que são brancas e por isso reflectem a luz que lhe incide.

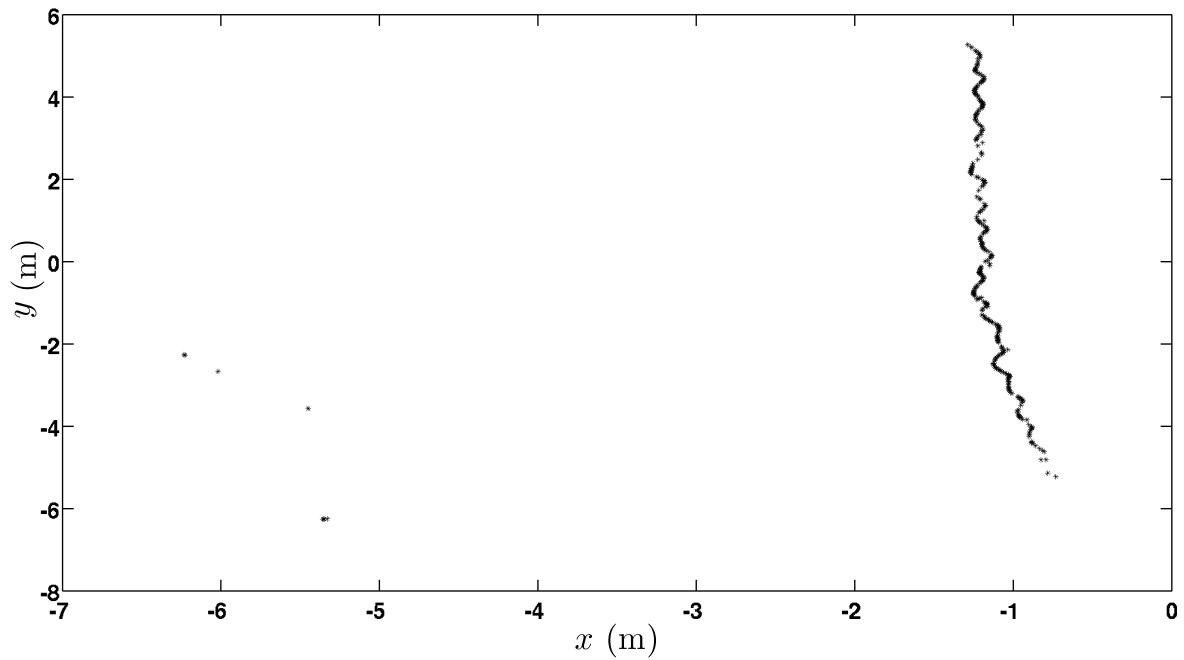


Figura 7.7: Posição do centro do objecto ao longo de uma trajectória segundo os eixos x, y

Como já foi explicado antes, o filtro tem algumas particularidades implementadas com vista à minimização dos erros provocados pelo ruído. Um deles é a criação de candidatos a novas trajectórias. Para ser criada uma nova trajectória é preciso que hajam três pontos na mesma zona que correspondam ao marcador num determinado espaço de tempo. Na figura 7.8 pode-se ver que vários desses falsos positivos não foram considerados. A vermelha está uma trajectória que foi inicializada devido à existência de falsos positivos. Como deixaram de aparecer observações que reportassem pontos naquela zona, o filtro estimou a trajectória apenas com base na previsão até que a trajectória foi considerada inactiva enquanto que a linha laranja representa a trajectória do objecto que queria seguir.

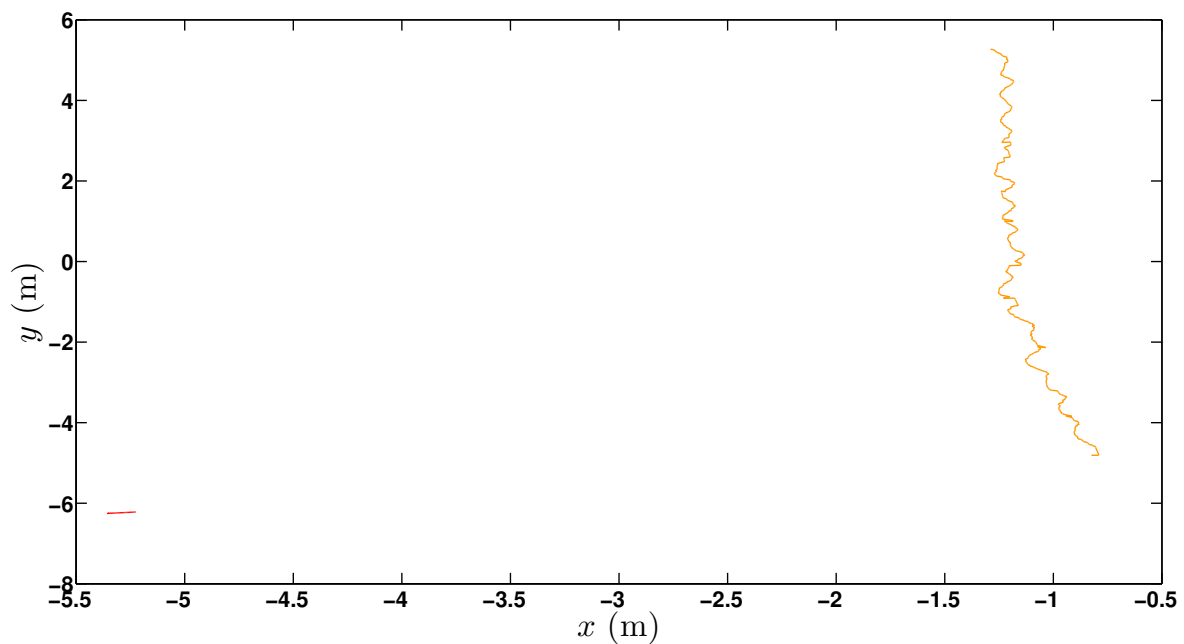


Figura 7.8: Trajectória estimada pelo filtro

A figura 7.9 representa a orientação do objecto segundo as componentes *roll*, *pitch* e *yaw*. Está assinalada uma variação brusca que se deveu a uma má identificação dos pontos do marcador. De futuro poder-se-á tentar melhorar a detecção do marcador para que aconteçam menos falsos negativos e trocas de pontos do marcador.

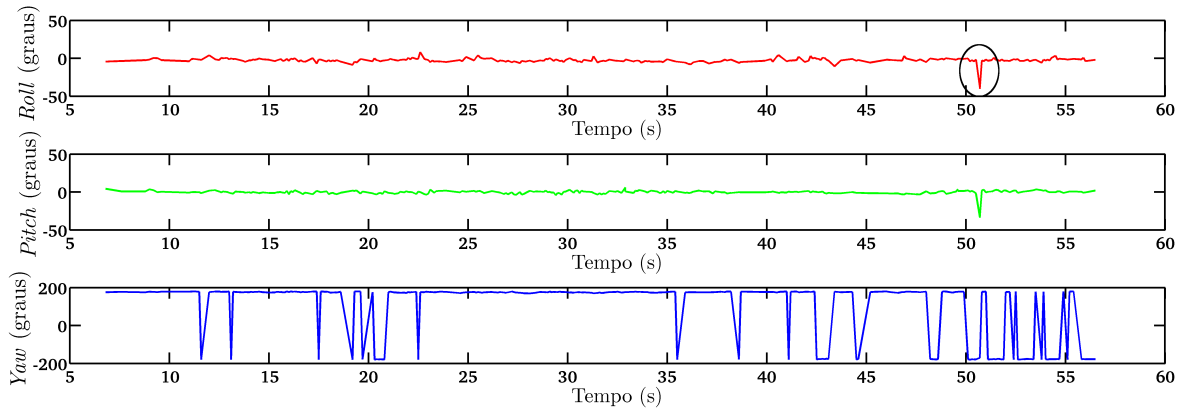


Figura 7.9: Variação do *Roll Pitch Yaw* do objecto

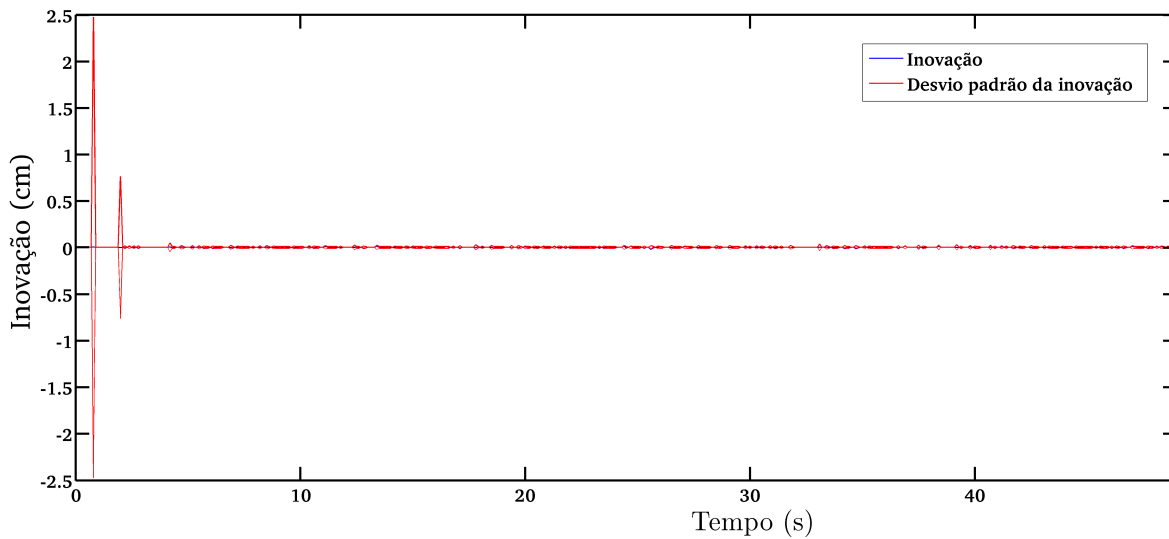


Figura 7.10: Inovação

A próxima situação representa a colisão entre dois robôs.

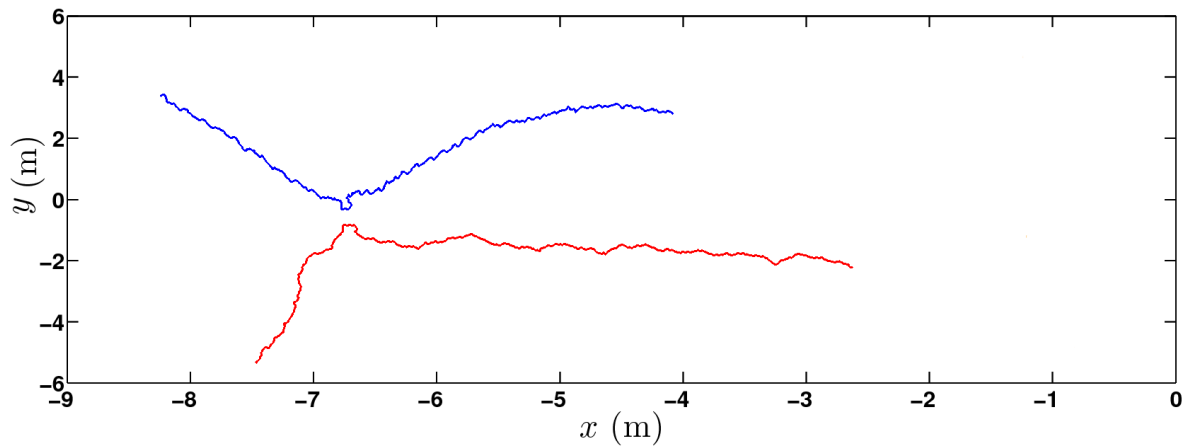


Figura 7.11: Colisão entre dois robôs

Pretende-se demonstrar na figura 7.11 que colocando o marcador no centro dos robôs, haverá sempre uma distância mínima entre os marcadores devido à estrutura dos robôs. Assim, a associação de dados pode ser feita correctamente através da região de validação impedindo que observações correspondentes a um objecto sejam associados a outro, mesmo na situação em que os marcadores estão mais próximos um do outro.

Por fim, na figura 7.12 estão representadas as trajetórias de quatro robôs, no entanto podem-se observar cinco trajetórias. Isto acontece porque houve uma oclusão que deu origem à inicialização de uma nova trajetória. A dada altura um dos robôs deixou de ser visto pelo sistema de visão e só voltou a reaparecer quando já estava fora da região de validação. Por esse motivo, uma das trajetórias foi dada como concluída por num determinado intervalo de tempo não ter recebido actualizações enquanto outra foi inicializada. Este pode ser um dos elementos a melhorar futuramente.

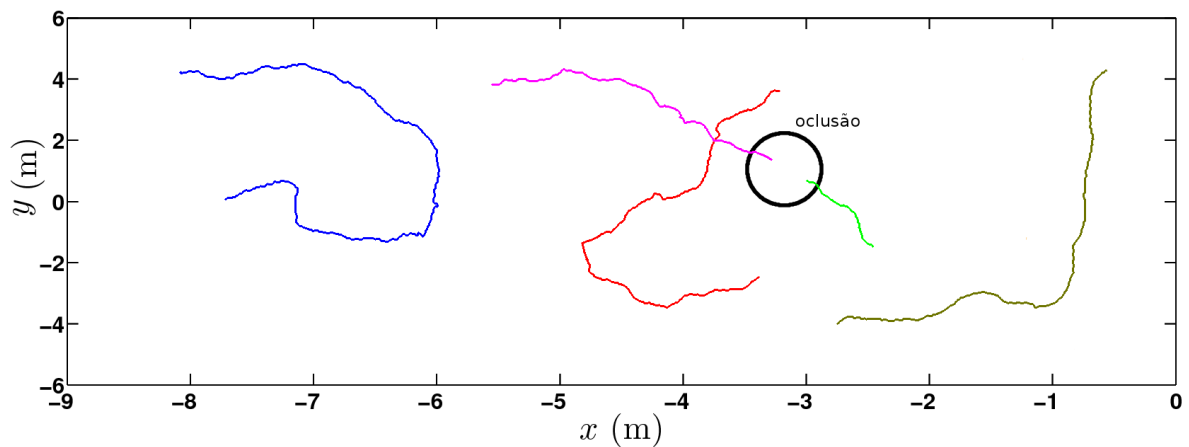


Figura 7.12: Trajectória de quatro robôs - Identificação de uma oclusão

Capítulo 8

Conclusões e Trabalho Futuro

Durante este trabalho foi projectado e implementado um sistema de visão para determinar a localização, atitude e seguimento de múltiplos objectos.

Para cumprir os objectivos foi necessário estudar os sistemas autónomos existentes e respectivos meios ambiente de actuação, efectuou-se um levantamento de requisitos para o sistema a desenvolver.

Foi necessário estudar temas relacionados com visão computacional, nomeadamente extracção de *features* por ser necessário a identificação de pontos de interesse nas imagens por forma a identificar os marcadores utilizados e temas necessários para ser possível identificar correctamente as coordenadas tridimensionais dos pontos de interesse. Foram também estudados métodos de estimação e associação de dados para conseguir estimar a localização, atitude e fazer o seguimento de múltiplos objectos.

Propôs-se uma arquitectura conceptual que permitisse cumprir todos os requisitos previamente listados, permitindo que o sistema de visão possa ter vários pares de câmaras, tenha a capacidade de detectar vários objectos com marcadores à base de LEDs ou outro tipo de *features*, proceda à determinação da suas localizações e atitudes, e faça o seguimento desses objectos mesmo quando ocorram oclusões ou colisões sendo também capaz de identificar novos objectos a seguir.

Implementou-se alguns componentes da arquitectura proposta com foco na localização e seguimento de múltiplos objectos com marcadores luminosos à base de LEDs, nomeadamente os módulos para a identificação dos pontos de interesse na imagem, técnicas para agrupar os vários pontos de interesse de cada objecto e associar com as medidas obtidas pelas várias câmaras, método para a determinação da posição e atitude dos objectos, filtro para seguimento de múltiplos objectos.

Foram realizados testes em situação estática e em movimento para a validação e afinação do sistema implementado. Os resultados demonstraram que a solução encontrada vai de encontro aos requisitos.

Após a elaboração deste trabalho, pode-se concluir que os objectivos propostos foram atingidos.

- Foi desenvolvido um sistema de visão capaz de detectar marcadores de LEDs colocados em objectos e determinar a sua localização, atitude e movimentação;
- Suporta a existência de oclusões e colisões entre robôs;

- Permite guardar os resultados para posterior análise;
- Permite adicionar mais pares de câmaras à medida que sejam necessárias;
- É de fácil instalação e calibração.

Como trabalho futuro pode-se destacar os seguintes:

- Implementação de um algoritmo capaz de determinar a localização, atitude e movimentação de objectos onde não é possível colocar marcadores;
- Melhoria do sistema de detecção de oclusões para que um objecto não origine várias trajectórias devido a oclusões;
- Aperfeiçoamento da identificação do marcador com LEDs;
- Implementar o algoritmo num sistema em tempo real para que os dados estimados possam ser utilizados na realimentação de malhas de controlo;
- Desenvolvimento de um sistema adaptativo para afinação dos parâmetros do filtro.

Bibliografia

- [1] Simone Ceriani, Giulio Fontana, Alessandro Giusti, Daniele Marzorati, Matteo Matteucci, Davide Migliore, Davide Rizzi, Domenico G. Sorrenti, and Pierluigi Taddei. Rawseeds ground truth collection systems for indoor self-localization and mapping. *Autonomous Robots*, 27(4):353–371, September 2009.
- [2] Alexander Ferrein, Gerhard Eckel, David Pirro, Tobias Kellner, Christof Rath, and Gerald Steinbauer. Providing ground-truth data for the nao robot platform. pages 133–144. Springer-Verlag, 2011.
- [3] Freek Stulp, Suat Gedikli, and Michael Beetz. Evaluating Multi-Agent Robotic Systems Using Ground Truth.
- [4] Motion Capture Systems from Vicon, . URL <http://www.vicon.com/>.
- [5] OptiTrack - Optical Motion Capture Systems and Tracking Software, . URL <http://www.naturalpoint.com/optitrack/>.
- [6] Sen Zhang, Boon Yuen Tan, Kee Hao Leo, and Ronny Quin Fai Tham. A hybrid human motion tracking system for virtual rehabilitation. *6th IEEE Conference on Industrial Electronics and Applications*, pages 1993–1998, June 2011.
- [7] Leonid Sigal, Alexandru O. Balan, and Michael J. Black. HumanEva: Synchronized Video and Motion Capture Dataset and Baseline Algorithm for Evaluation of Articulated Human Motion. *International Journal of Computer Vision*, 87(1-2):4–27, August 2009.
- [8] Qiong Wu and Pierre Boulanger. Real-Time Estimation of Missing Markers for Reconstruction of Human Motion. *XIII Symposium on Virtual Reality*, pages 161–168, May 2011.
- [9] A. Kunju, N. and Kumar, N. and Pankaj, D. and Singh, D.P. and Kumar. Algorithm for Kinematic Measurements using Active Markers. *Advance Computing Conference, 2009. IACC 2009. IEEE International*, (March):387–391, 2009.
- [10] R. Janssen, E. Lou, N.G. Durdle, J. Raso, D. Hill, A.B. Liggins, and S. Mahood. Active Markers in Operative Motion Analysis. *IEEE Transactions on Instrumentation and Measurement*, 55(3): 854–859, June 2006.
- [11] Choong Seng Chan and Flora S. Tsai. Computer Animation of Facial Emotions. *International Conference on Cyberworlds*, pages 425–429, October 2010.
- [12] Jan Smisek, Michal Jancosek, and Tomas Pajdla. 3D with Kinect. *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 1154–1160, November 2011.
- [13] Weidi Jia, Won-Jae Yi, Jafar Saniie, and Erdal Oruklu. 3D image reconstruction and human body tracking using stereo vision and Kinect technology. *IEEE International Conference on Electro/Information Technology*, pages 1–4, May 2012.
- [14] Victor Zappi, Andre Brogni, and Darwin Caldwell. Passive hand pose recognition in virtual reality. *IEEE Symposium on 3D User Interfaces (3DUI)*, 143:143–144, March 2010.
- [15] M. Po Yang and Wenyan Wu and Moniri. A Hybrid Marker-Based Camera Tracking Approach in Augmented Reality. *IEEE International Conference on Networking, Sensing and Control*, (April):622–627, 2007.

- [16] B Y Nathan Michael, Daniel Mellinger, Quentin Lindsey, and Vijay Kumar. Experimental Evaluation of Multirobot Aerial Control Algorithms. *Robotics Automation Magazine, IEEE*, 17(September):56–65, 2010.
- [17] Oliver Wulf, Andreas Nuchter, Joachim Hertzberg, and Bernardo Wagner. Ground truth evaluation of large urban 6D SLAM. *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 650–657, October 2007.
- [18] Jose-Luis Blanco, Francisco-Angel Moreno, and Javier Gonzalez. A collection of outdoor robotic datasets with centimeter-accuracy ground truth. *Autonomous Robots*, 27(4):327–351, August 2009.
- [19] R. Bostelman, T. Hong, R. Madhavan, T. Chang, and H. Scott. Performance Analysis of an Autonomous Mobile Robot Mapping System for Outdoor Environments. *2006 9th International Conference on Control, Automation, Robotics and Vision*, pages 1–6, 2006.
- [20] D Marzorati, M Matteucci, D Migliore, and D G Sorrenti. On the collection of robot-pose Ground-Truth, for indoor scenarios, in the RAWSEEDS project.
- [21] Piyush Khandelwal and Peter Stone. A low cost ground truth detection system for RoboCup using the Kinect. *Proceedings of the RoboCup International Symposium 2011*, 2011.
- [22] William Shackelford, Tsai Hong, and Tommy Chang. Inexpensive ground truth and performance evaluation for human tracking using multiple laser measurement sensors. *Proceedings of the 10th Performance Metrics for Intelligent Systems Workshop*, pages 201–206, 2010.
- [23] Daniel Mellinger, Nathan Michael, and Vijay Kumar. Trajectory Generation and Control for Precise Aggressive Maneuvers with Quadrotors. *International Symposium on Experimental Robotics*, 2010.
- [24] Yinan Cui and T. Inanc. Multiple Air Robotics Indoor Testbed. *2012 24th Chinese Control and Decision Conference (CCDC)*, pages 3487–3492, May 2012.
- [25] Quentin Alexander, Aparna Veluchamy, Harold Hockett, and Monica Anderson. Acquisition of 2-D Ground Truth Data in Multirobot Experiments.
- [26] GTMS - The Ground Truth Measurement System, . URL <http://gtms.sourceforge.net/>.
- [27] Qualisys motion capture systems, . URL <http://www.qualisys.com/>.
- [28] Motion Analysis Corporation, the Motion Capture Leader, . URL <http://www.motionanalysis.com/>.
- [29] PhaseSpace Motion Capture, . URL <http://www.phasespace.com/>.
- [30] INESC Porto à INESC TEC, . URL <http://www2.inescporto.pt/>.
- [31] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.
- [32] Duane C. Brown. Close-Range Camera Calibration. *Photogrammetric Engineering*, 37(8):855–866, 1971.
- [33] Joss Knight. Robot Navigation by Active Stereo Fixation. Technical report, Active Vision Lab, University of Oxford, 1999.
- [34] Elan Dubrofsky and Robert J Woodham. Combining Line and Point Correspondences for Homography Estimation. *Advances in Visual Computing*, pages 202–213, 2008.
- [35] Richard I Hartley. Theory and Practice of Projective Rectification. *International Journal of Computer Vision*, 35(2):115–127, 1999.
- [36] C. Loop and Zhengyou Zhang. Computing Rectifying Homographies for Stereo Vision. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 2 vol. (xxiii+637+663). Microsoft Research, IEEE Comput. Soc, 1999.

- [37] Andrea Fusiello, Emanuele Trucco, and Alessandro Verri. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12(1):16–22, 2000.
- [38] Martin Bujnak. *Reconstructing 3D mesh from video sequence*. Master thesis, Comenius University, Bratislava, Slovakia, 2005.
- [39] Emanuele Trucco and Alessandro Verri. Introductory Techniques for 3-D Computer Vision. In *Introductory Techniques for 3-D Computer Vision*, chapter 7. Prentice Hall, 1998.
- [40] Hugh Durrant-Whyte. *Introduction to Decentralised Data Fusion*. 3.1 edition.
- [41] J. Y. Bouguet. Camera Calibration Toolbox for Matlab. URL http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/parameters.html.
- [42] Zhengyou Zhang and One Microsoft Way. Flexible camera calibration by viewing a plane from unknown orientations. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 00(c):666–673 vol.1, 1999.
- [43] J Heikkila and O Silven. A four-step camera calibration procedure with implicit image correction. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1106–1112, 1997.
- [44] Basler acA1300-30gc, . URL <http://www.baslerweb.com/products/ace.html?model=167>.

Apêndice A

Calibração de extrínsecos

Neste anexo está uma tabela com a correspondência entre as medidas, em metros, de pontos conhecidos do mundo e os pixels que lhes correspondem em cada uma das imagens das câmaras. Esta tabela é necessária para a calibração dos parâmetros extrínsecos.

A.1 Correspondência de medidas

N do ponto	xdir	ydir	xesq	yesq	X	Y	Z
1	684.331331	905.996997	1265.147648	507.309810	-8.816	-5.846	0
2	612.888889	847.021021	1247.406907	516.899399	-8.816	-5.24	0
3	743.307307	843.664665	1228.227728	479.979479	-8.204	-5.846	0
4	416.182432	681.121121	1184.595095	554.778278	-8.816	-3.234	0
5	307.293794	581.149650	1121.303804	592.177678	-8.816	-1.741	0
6	80.020521	356.273774	896.740741	710.129630	-8.836	1.741	0
7	80.536537	356.513514	759.250000	780.133634	-8.836	3.232	0
8	364.997998	545.188689	1076.232733	556.216717	-8.192	-1.741	0
9	184.067568	386.480981	850.710711	660.263764	-8.214	1.741	0
10	610.012012	550.462963	1051.299800	453.128629	-6.701	-3.234	0
11	238.728228	296.338839	611.929930	594.575075	-6.715	3.232	0
12	28.236737	304.010511	516.034034	892.571572	-8.836	5.23	0
13	11.454955	287.708208	426.371371	932.368368	-8.836	5.84	0
14	54.165165	272.844344	386.095095	851.815816	-8.22	5.84	0
15	1135.651652	402.303804	844.117868	228.013013	0	-5.846	0
16	939.594219	311.802052	723.468844	234.965465	0	-2.945	0
17	869.949825	280.995495	668.568443	238.201952	0	-1.866	0
18	764.943819	236.044294	566.559184	245.274274	0	0	0
19	673.123498	198.405155	449.086712	256.302302	0	1.875	0
20	620.980105	179.345846	376.085961	264.093844	0	2.927	0
21	508.182558	136.792042	154.436311	289.506256	0	5.84	0
22	1256.102603	254.863864	618.942317	96.875375	8.208	-5.846	0
23	1261.257007	247.911411	606.116241	89.203704	8.82	-5.846	0
24	1232.128629	236.044294	584.060185	88.963964	8.82	-5.359	0
25	1110.095971	219.741992	549.897272	112.098849	6.702	-3.228	0
26	827.635761	115.934685	247.059184	122.048048	6.722	3.229	0
27	1065.983859	179.106106	460.114740	94.477978	8.195	-1.734	0
28	919.336211	126.603103	304.476852	99.033033	8.211	1.743	0
29	1141.165666	198.045546	513.217092	87.165916	8.82	-3.228	0
30	1074.015140	174.191441	453.042417	87.885135	8.82	-1.743	0
31	929.015703	123.007007	297.044920	91.001752	8.83	1.743	0
32	871.508133	103.947698	226.201827	93.998498	8.83	3.229	0
33	803.182307	83.090340	125.307933	98.913163	8.83	5.232	0
34	785.171859	76.077953	94.021897	101.070821	8.83	5.84	0
35	770.457833	80.093594	95.250563	110.061061	8.216	5.84	0

Tabela A.1: Medidas extrínsecos

