

SISTEMA DE DETECÇÃO E LOCALIZAÇÃO DE VEÍCULOS SUBAQUÁTICOS

Pedro Miguel Ferreira Gonçalves



Mestrado em Engenharia Electrotécnica e de Computadores
Instituto Superior de Engenharia do Porto
2012

Tese/Dissertação, para satisfação parcial dos requisitos do
Mestrado em Engenharia Electrotécnica e de Computadores

Candidato: Pedro Miguel Ferreira Gonçalves,
Nº 1060417

Orientador: Eduardo Alexandre Pereira da Silva

Co-Orientador: Alfredo Manuel de Oliveira Martins



Mestrado em Engenharia Electrotécnica e de Computadores
Área de Especialização de Sistemas Autónomos
Departamento de Engenharia Electrotécnica

Dedico este trabalho principalmente à minha família e amigos e a todos os que me apoiaram.

(Página intencionalmente deixada em branco)

Agradecimentos

Agradeço à minha família e amigos (principalmente à Maria Costa, ao João Teixeira e ao Hugo Queirós), por me apoiarem nas horas difíceis, ao constante acompanhamento dos Engs: Eng. Eduardo Alexandre Pereira da Silva, Eng. Alfredo Manuel de Oliveira Martins, Eng. Luís Miguel Vieira Lima, Eng. André Miguel Pinheiro Dias e a todos os outros que me foram dando concelhos ao longo da minha dissertação.

Agradeço também ao Laboratório de sistemas autónomos (LSA) pelo acesso às instalações e equipamentos para o desenvolvimento deste projecto e ao Instituto Superior de Engenharia do Porto, por me proporcionar um local de ensino e aprendizagem de elevada qualidade.

(Página intencionalmente deixada em branco)

Resumo

O posicionamento e navegação de veículos subaquáticos é um problema complexo. Dada a hostilidade do meio aquático, os métodos de localização existentes (geralmente baseados em acústica) são limitados e pouco precisos, particularmente em operações junto a infraestruturas subaquáticas como oleodutos no fundo marinho profundo. Consequentemente para determinar a posição de veículos subaquáticos é necessário recorrer, entre outras, a técnicas alternativas como técnicas de visão artificial. Neste trabalho propõe-se uma solução que converge para a solução dos problemas referidos, baseada em visão stereo.

Foram estudadas técnicas de visão adequadas para localização de veículos subaquáticos, com auxílio das bibliotecas *Open Source* do *OpenCV* [1], desenvolvido um sistema de posição relativa baseada em visão e obtidos resultados experimentais em cenários de operação com um pequeno ROV (nomeadamente num tanque de testes e em exercícios navais REP2012). Finalmente, foi desenvolvido um sistema de localização externa com visão stereo para o tanque de testes do LSA, tendo o mesmo sido validado e caracterizado experimentalmente.

Os resultados obtidos demonstram que o sistema proposto é uma solução viável, e até bastante fiável, para a detecção e localização relativa ou absoluta de veículos subaquáticos, visto que para todos os métodos de visão implementados, o erro obtido foi consideravelmente baixo para as aplicações em questão. O sistema desenvolvido foi ainda comparado com um equipamento de localização de sinais acústicos (USBL), num cenário que evidencia as limitações dos sistemas de posicionamento acústico (tanque de testes) e comprovou-se que a solução proposta obteve melhores resultados. Deste modo, o sistema implementado constitui uma alternativa funcional aos seus equivalentes acústicos.

Palavras-Chave

Groudtruth, stereo, subaquático, posicionamento, detecção, infraestruturas subaquáticas, visão subaquática, veículos subaquáticos, *tracking*.

(Página intencionalmente deixada em branco)

Abstract

The navigation and positioning of underwater vehicles is a complex problem to solve, due to the harshness of the surrounding environment and the limitations and low precision of the existent localization methods (usually based on acoustic technology). These are particularly noticeable in repair operations of underwater structures, oil-pipe maintenance and mapping of the seabed. For this reason, the determination of the position of underwater vehicles alternatively recurs to artificial vision techniques, to solve the problem of the positioning relative to an underwater structure or even the absolute position. In the present work a solution for both of these problems, based on stereo vision, is proposed.

Several vision techniques, with resource to the Open Source OpenCV [1] libraries, suitable for underwater vehicles localization were studied. Furthermore, a vision based relative position system was developed and experimental results were obtained in operational scenarios with a small ROV (namely in a test water tank and in the REP2012 naval drills under the Portuguese navy). Finally, it was developed a stereo vision external localization system for the LSA test water tank. This system was then experimentally validated and fully characterized.

The obtained results show that the proposed system is a viable solution, and quite reliable, for detection and relative or absolute localization of underwater vehicles, once for all the implemented vision methods the obtained error was considerably low for the applications in cause. The developed system was also compared with an acoustic localization sensor (USBL), in a scenario that points out the limitations of these type of sensors (water tank). It was proven that the proposed solution has better results. Thus, the implemented system is a functional alternative to its acoustic equivalents.

Keywords

Groudtruth, stereo, underwater, positioning, detection, underwater infrastructures, underwater vision, underwater vehicles, tracking.

(Página intencionalmente deixada em branco)

Conteúdo		xi
1	Introdução	1
1.1	Âmbito da dissertação	1
1.2	Motivação - Enquadramento	2
1.3	Cenários de Operação	3
1.3.1	Inspeção de infraestruturas subaquáticas	3
1.3.2	Localização de veículos subaquáticos em relação a um ASV	3
1.3.3	<i>Groundtruth stereo</i>	3
1.4	Objectivos	3
1.5	Estrutura do documento	5
2	Métodos de localização para veículos subaquáticos	7
2.1	USBL - Ultra short baseline	7
2.2	LBL - Long Baseline Acoustic Positioning	9
2.3	Sistemas inerciais para Dead-Reckoning	10
2.4	DVL - Doppler Velocity Log	11
2.5	Sistemas de localização para veículos terrestres	12
2.5.1	GPS	12
2.5.2	VICON	13
2.5.3	Groundtruth stereo	13
3	Visão stereo	15
3.1	Modelo de câmara	15
3.2	Distorção Óptica	17
3.2.1	Distorção radial	17
3.2.2	Distorção por refração	17
3.2.3	Distorção tangencial	18

3.2.4	Correcção da distorção	18
3.3	Matriz de rotação e vector de translação	19
3.3.1	Coordenadas homogéneas	22
3.4	Calibração da câmara	23
3.4.1	<i>Chessboard</i>	24
3.4.2	Parâmetros Intrínsecos	25
3.4.3	Parâmetros Extrínsecos	25
3.4.4	Matriz Fundamental e Essencial	26
3.5	Homografia	27
3.6	Rectificação stereo	29
3.6.1	Geometria Epipolar	29
3.6.2	Algoritmo da rectificação stereo	31
3.7	Determinação das coordenadas (X, Y, Z) do objecto no referencial do mundo	36
4	Métodos de detecção	39
4.1	Espaços de cor	39
4.2	<i>Template Match</i>	42
4.3	Segmentação de imagem por cores	44
4.3.1	Histograma	44
4.3.2	Processo de segmentação por cor	45
4.4	Transformada de <i>Hough</i> - Detecção de linhas	46
5	Sistemas de localização	49
5.1	Arquitectura do sistema - Posicionamento relativo a uma infraestrutura . . .	49
5.2	Software - Posicionamento relativo a uma infraestrutura	50
5.2.1	Método I - Detecção de linhas verticais (<i>Full Frame</i>)	50
5.2.2	Método II - Detecção de linhas verticais (<i>Template Match</i>)	51
5.2.3	Método III - Segmentação por cor (<i>Template Match</i>)	52
5.3	Arquitectura do sistema - Visão stereo (<i>Groundtruh</i>)	53
5.4	Software - Visão stereo (<i>Groundtruh</i>)	53
5.4.1	Calibração das câmaras	54
5.4.2	Aquisição de imagem e rectificação das mesmas	55
5.4.3	Detecção do objecto subaquático pretendido	56
5.5	Implementação - Visão stereo (<i>Groundtruh</i>)	58
5.5.1	Câmara	58
5.5.2	Unidade de protecção da câmara	59
5.5.3	Tanque de testes subaquático	60
5.5.4	Posicionamento das câmaras	61

6	Resultados	63
6.1	Sistema de posicionamento relativo a uma infraestrutura	63
6.2	Sistema de localização de veículos subaquáticos - <i>Gronudtruh</i>	65
6.3	Template Match (Método I)	66
6.3.1	Erro obtido no Método I	67
6.4	Segmentação por cor (Método II)	68
6.4.1	Erro obtido no Método II	69
6.5	Qualidade da rectificação stereo	70
6.5.1	Média aritmética do erro obtido na rectificação na variação da <i>Baseline</i>	71
6.6	USBL - Leituras	73
6.7	<i>Tracking 3D</i>	73
7	Conclusão	75
	Bibliografia	77

(Página intencionalmente deixada em branco)

LISTA DE FIGURAS

1.1	Sistema de posicionamento relativo - Obtém o centro do alvo no plano da imagem ($c_{x,y}$) e caracteriza a sua dimensão (d).	4
1.2	Sistema de visão stereo para localização absoluta de veículos subaquáticos.	4
2.1	Ilustração dos atrasos de fase obtidos pelo <i>Transceiver</i> - Determinação da orientação θ	8
2.2	Representação de r , θ , ψ e da posição do <i>Transponder</i> (x , y , z) no referencial do <i>Transceiver</i> (O).	9
2.3	USBL - Tritech.	9
2.4	Sistema de posicionamento subaquático LBL.	10
2.5	Posicionamento de um veículo utilizando apenas sensores de <i>Dead-Reckoning</i>	11
2.6	<i>DVL - RDI's Bottom Tracking</i>	11
2.7	<i>GPS</i> - Localização do receptor (ponto branco).	12
2.8	<i>X3D quadrotor</i> - 4 marcas para detecção com o sistema VICON [22].	13
2.9	C - Referencial da câmara; W - Referencial do mundo.	13
3.1	<i>Pinhole camera</i> [25].	15
3.2	Projectão do plano de imagem em frente do plano <i>Pinhole</i>	16
3.3	Distorção radial provocada pelas lentes[25].	17
3.4	Distorção provocada pela refração dos raios de luz em diferentes meios, onde a densidade do meio $n_2 > n_1$	18
3.5	Exemplo - Distorção tangencial [27].	18
3.6	Projectão das coordenadas da câmara para as coordenadas do objecto.	20
3.7	Rotação 2D.	20
3.8	Rotação 3D de um objecto.	21
3.9	Translação do ponto (x' , y') do referencial $\{O'\}$ para o referencial $\{O\}$	22
3.10	Particularidade física do algoritmo de Tsai[28].	24
3.11	<i>Chessboard</i>	25

3.12	Visualização do plano do objecto descrito pela matriz H .	28
3.13	Sistema de coordenadas alinhado pelos planos de imagem.	29
3.14	Geometria Epipolar - Exemplificação.	30
3.15	Geometria Epipolar - Vantagens.	30
3.16	Situação ideal para rectificação stereo.	32
3.17	Disparidade versos Distância do objecto às câmaras.	33
3.18	Passos necessários para a rectificação stereo [25].	36
3.19	Rotação e translação do referencial da câmara para o referencial do mundo.	37
4.1	<i>Array de Bayer</i> .	40
4.2	Tons de cinza (“b”); Imagem real (“a”) - ROAZ II.	40
4.3	Luz branca - cor.	41
4.4	Tons de cor <i>RGB</i>	41
4.5	Tons de cor <i>HSL e HSV</i> .	42
4.6	Deslocação do <i>Template</i> (<i>FALCOS - Autonomous Aerial Vehicles</i>) na imagem original.	43
4.7	<i>Template Match</i> - LINCE.	44
4.8	Segmentação de cor - detecção da cor vermelha no ROAZ II.	45
4.9	Processo de segmentação por cor.	46
4.10	Representação do sistema de coordenadas polar.	46
4.11	Representação de uma recta contendo os pontos A, B, C, D e E.	47
4.12	Representação das curvas senoidais dos pontos A, B, C, D e E.	47
5.1	Arquitectura do sistema desenvolvido - Posicionamento relativo do ROV a uma infraestrutura.	49
5.2	Posicionamento relativo do ROV a uma infraestrutura - Método I.	50
5.3	Posicionamento relativo do ROV a uma infraestrutura - Método II.	51
5.4	Posicionamento relativo do ROV a uma infraestrutura - Método III.	52
5.5	Arquitectura do sistema desenvolvido - <i>Groundtruth</i> .	53
5.6	Calibração das câmaras - Diagrama de blocos.	54
5.7	Obtenção dos parâmetros extrínsecos.	54
5.8	Captura das imagens e rectificação das mesmas - Diagrama de blocos.	55
5.9	Detecção do objecto subaquático - Diagrama de blocos.	56
5.10	Câmaras utilizadas.	58
5.11	Resolução versos fps - drive (Basler pylon).	59
5.12	Unidade de protecção utilizado para proteger as câmaras.	59
5.13	Unidade de protecção utilizado para proteger as câmaras - perspectiva pelo vidro protector.	60
5.14	Suporte interno para as câmaras	60
5.15	Tanque de testes, utilizado para os ensaios experimentais (dimensões).	61

5.16	Posicionamento das câmaras - vista subaquática.	61
5.17	Posicionamento das câmaras - vista externa.	61
5.18	Variação do tamanho da <i>Baseline</i> versus área visível útil.	62
6.1	Método I (Detecção de linhas verticais – <i>Full Frame</i>).	63
6.2	Método II (Detecção de linhas verticais – <i>Template Match</i>).	64
6.3	Método III (Segmentação por cor – <i>Template Match</i>).	64
6.4	Aplicação dos métodos de visão desenvolvidos no artigo publicado[3], A e D corresponde ao método I, B ao método III e C ao método II.	65
6.5	Posição real versus determinada - <i>Template Match</i>	66
6.6	Média do erro obtido - <i>Template Match</i>	68
6.7	Posição real versus determinada - Segmentação por cor.	68
6.8	Média do erro obtido - Segmentação por cor.	69
6.9	Rectificação stereo - linhas epipolares	71
6.10	Variação do erro com o aumento da <i>Baseline</i> (b), [azul b=0.75, vermelho b=3 e verde b=6] e um Z=11.	72
6.11	Posição do veículo subaquático utilizando o sistema USBL.	73
6.12	Posição do veículo subaquático - (X, Y, Z).	74
6.13	Posição do veículo subaquático - sistema desenvolvido.	74

(Página intencionalmente deixada em branco)

LISTA DE TABELAS

6.1	Localizações reais do veículo subaquático (em metros).	66
6.2	Localizações reais versus Localizações determinadas - <i>Template Match</i> (em metros).	67
6.3	Localizações reais versus Localizações determinadas - Segmentação de cor (em metros).	69

(Página intencionalmente deixada em branco)

Acrónimos

ASV	Autonomous Surface Vehicle
DVL	Doppler Velocity Log
GPS	Global Positioning System
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
ISEP	Instituto Superior de Engenharia do Porto
LBL	Long Baseline
LSA	Laboratório de Sistemas Autónomos
MSDF	Multi-Sensor Data Fusion
PC	Personal Computer
REP2012	Rapid Environmental Picture 2012
RMS	Root-Mean-Square
ROV	Remotely Operated Vehicle
USBL	Ultra-Short Baseline
UUV	Unmanned Underwater Vehicle

(Página intencionalmente deixada em branco)

1.1 Âmbito da dissertação

É muito comum a utilização de veículos operados remotamente (ROV) de pequena dimensão em grande variedade de tarefas de inspecção subaquática. Este facto, deve-se a características tais como a alta manobrabilidade, o espaço mínimo de operação e o baixo orçamento em relação a outras soluções de inspecção. Além disso, a sua grande portabilidade permite a sua rápida implementação, bem como a possibilidade de operação a partir da costa, navio ou de qualquer outra plataforma.

Devido à imprevisibilidade do ambiente que envolve o ROV e a natureza complexa das manobras necessárias para este tipo de tarefas, a perícia da intervenção humana é a única maneira de assegurar o êxito da mesma. A teleoperação de um ROV acarreta uma série de dificuldades, principalmente por causa das correntes e forças de arrasto do cabo, sendo que além disso, muitos ROVs são subatuados. Consequentemente ganha grande importância a teleoperação assistida.

O desenvolvimento de um sistema de ajuda à teleoperação, implica o desenvolvimento de um sistema de navegação com elevada exactidão. As tecnologias que são usadas mais correntemente na determinação da posição e atitude, não nos fornecem a informação com a qualidade pretendida. Tendo em consideração este facto, a determinação da posição de veículos subaquáticos também passou a incorporar o uso de técnicas de visão artificial, de modo a obter o posicionamento em relação a uma infraestrutura, ou mesmo o seu posicionamento absoluto. Uma das tecnologias existentes da visão artificial que permitem obter a posição em três dimensões é a tecnologia stereo.

O trabalho descrito nesta dissertação e desenvolvido no âmbito do mestrado de enge-

nharia electrotécnica e de computadores na área de especialização de sistemas autónomos do ISEP, são abordadas técnicas de visão computacional, com auxílio das bibliotecas *Open Source* do *OpenCV* [1], para a aplicar à localização de veículos subaquáticos.

1.2 Motivação - Enquadramento

O LSA/ISEP têm vindo a desenvolver ao longo dos tempos, inúmeros projectos relativos à robótica marinha, tanto em veículos de superfície (ASV - ROAZ II) como em veículos subaquáticos (ROV - VideoRay PRO 3E). No caso dos veículos subaquáticos o problema de posicionamento e navegação é mais complexo do que em outros cenários de operação, já que o ambiente aquático hostil e os métodos de localização existentes (geralmente baseados em acústica) são limitados.

Como tal, existe a necessidade de desenvolver ferramentas e técnicas para a localização e posicionamento de sistemas subaquáticos. O problema de posicionamento e localização pode ser descrito em duas etapas distintas: uma em que se requer a localização de um veículo subaquático a partir de um sistema externo e outra em que o próprio veículo determina o seu posicionamento em relação a uma infraestrutura.

O LSA possui, nas suas instalações, um tanque de testes para o qual é necessária a existência de um sistema de localização e determinação de posicionamento de sistemas subaquáticos. Deste modo, o sistema proposto será implementado e validado no referido tanque de testes.

As características do ambiente subaquático representam uma infinidade de desafios difíceis para navegação de veículos subaquáticos, e estes obstáculos são muito diferentes dos problemas encontrados na terra ou no ar [2]. A atenuação da radiação acústica e electromagnética ocorre rapidamente na água, restringindo a gama de sensores acústicos e também os limites na largura de banda de comunicação. Como consequência desta absorção acústica e electromagnética, exige com que o veículo subaquático esteja submerso perto do local da missão ou da pesquisa, para reunir dados precisos e de alta resolução.

Outro desafio que se depara com a navegação subaquática é a pressão intrínseca do ambiente. A pressão debaixo de água aumenta a uma taxa de aproximadamente 1 PA (14,7 psi) a cada 10 metros de profundidade, podendo influenciar e restringir negativamente os sensores e actuadores do projecto. Outras questões, como a presença inerente de ondas e correntes subaquáticas podem tornar a tarefa do movimento do veículo mais complexa, e como resultado afectar a precisão na localização do veículo.

1.3 Cenários de Operação

O sistema proposto para implementação possuirá determinadas características, que fazem com que o sistema possa ser aplicado em diversas situações e ambientes. Contudo, dependendo da situação onde irá ser aplicado poderão surgir determinadas restrições.

1.3.1 Inspeção de infraestruturas subaquáticas

Neste tipo de ambientes o sistema funcionará correctamente, caso as condições de visibilidade subaquáticas assim o permitam. Para além desta restrição se o sistema for aplicado a altas profundidades, a calibração do mesmo (determinação dos parâmetros intrínsecos e extrínsecos) torna-se complexa. Não obstante, a localização relativa de um veículo subaquático a uma infraestrutura, a curtas distâncias, é viável.

1.3.2 Localização de veículos subaquáticos em relação a um ASV

Para situações de curto alcance os problemas acima mencionados não ocorrem, contudo o uso do método de segmentação por cor não é o mais indicado, isto porque devido à intensa luz, proveniente da luz solar, pode interferir drasticamente no método de detecção e localização do posicionamento do veículo.

1.3.3 *Groundtruth stereo*

Para ambientes de teste, onde é utilizado um *Groundtruth stereo*, é possível utilizar as máximas capacidades do sistema, já que em tanque de testes ou piscinas, a água é límpida, não existe interferência de luz externa e a calibração do sistema é bastante acessível, por outras palavras o ambiente envolvente é controlável.

1.4 Objectivos

Os objectivos desta dissertação podem ser resumidos nos seguintes pontos:

- Identificação de técnicas de visão adequadas para a localização de veículos subaquáticos;
- Desenvolvimento de um sistema de posicionamento relativo baseado em visão para robôs submarinos;
- Desenvolvimento e implementação de um sistema de localização externa para o seguimento de alvos em ambientes subaquáticos estruturados (*Groundtruth*);
- Validação e caracterização dos sistemas desenvolvidos.

As técnicas são aplicadas em duas dimensões distintas.

Uma primeira dimensão envolve o desenvolvimento de um sistema de posicionamento relativo para um veículo subaquático (Figura 1.1). Este sistema foi já objecto de uma publicação [3], apresentada numa conferência internacional (OCEANS MTS 2012) e participação nos exercícios navais REP2012, da marinha portuguesa, com testes e demonstração tecnológica.

Uma segunda dimensão compreende a implementação de um sistema de visão stereo (Figura 1.2) com o propósito de localizar veículos subaquáticos.

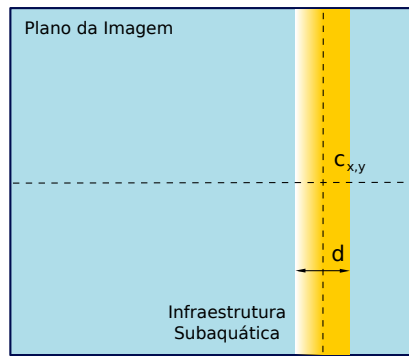


Figura 1.1: Sistema de posicionamento relativo - Obtém o centro do alvo no plano da imagem ($c_{x,y}$) e caracteriza a sua dimensão (d).

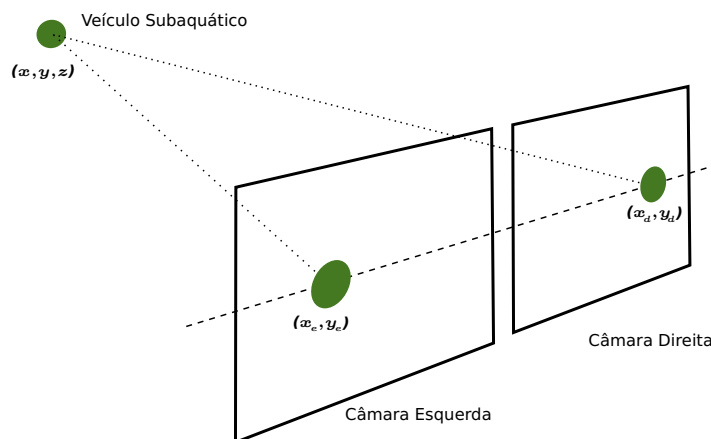


Figura 1.2: Sistema de visão stereo para localização absoluta de veículos subaquáticos.

A visão computacional possui uma série de vantagens sobre outros tipos de sensores, para aplicações subaquáticas. As câmaras são leves e não possuem uma faixa operacional mínima, ao contrário dos seus equivalentes acústicos [4]. Devido a estes factos, foi projectado um sistema de visão stereo, capaz de localizar e rastrear um veículo subaquático em

tempo real, seja em tanques de testes ou em situações reais (posicionamento de um veículo subaquático dentro de uma gruta), cujo o sistema em si seja de fácil implementação e calibração, tornando-se assim num sistema mais acessível para ambientes subaquáticos, ao contrário dos seus equivalentes acústicos (ex: *USBL*).

1.5 Estrutura do documento

No segundo capítulo, serão descritos sistemas existentes para localização subaquáticos bem como sistemas de localização externa para ambientes terrestres.

No capítulo “Visão stereo”, são apresentados alguns conceitos base de visão artificial e de visão stereo em particular. Estes constituem fundamentos das técnicas utilizadas no sistema desenvolvido.

No quarto capítulo, são descritos os métodos de detecção utilizados para identificar e seguir os veículos no plano de imagem.

O sistema de localização desenvolvido é descrito no quinto capítulo, onde é descrito inicialmente o sistema desenvolvido pela sua arquitectura, seguido do software desenvolvido seguido da sua implementação.

No capítulo seis, são apresentados os resultados bem como os testes de validação e caracterização do sistema e respectiva comparação dos resultados com outros sistemas de localização subaquáticos (ex: *USBL*).

Por fim são apresentadas algumas conclusões e comentários finais bem como respectivas linhas de trabalho futuro, tendo em vista a continuidade do presente trabalho.

(Página intencionalmente deixada em branco)

CAPÍTULO 2

MÉTODOS DE LOCALIZAÇÃO PARA VEÍCULOS SUBAQUÁTICOS

Actualmente os Unmanned Undersea Vehicles (UUV) são utilizados para diversas aplicações científicas, militares e comerciais, tais como levantamento de alta resolução do fundo do mar [5], medidas anti minas [6], inspecção e reparo de estruturas subaquáticas feitas pelo homem [7] e descoberta/localização de destroços [8].

Existem comercialmente inúmeros sensores de posicionamento subaquático [9], mas infelizmente ainda não existe um sensor que ofereça a solução perfeita para todas as necessidades de navegação subaquática, em geral, para tal tarefa utilizam-se combinações de sensores. O estado actual dos sistemas de navegação é baseado nos valores da velocidade, determinados pelo sistema Doppler (DVL), fundido com os dados obtidos, de um sistema inercial (IMU) ou acústico (LBL e SBL) [10].

2.1 USBL - Ultra short baseline

Actualmente existem diferentes sistemas de posicionamento por ondas acústicas, sendo um deles o USBL (sistema de posicionamento acústico subaquático). Um sistema de posicionamento acústico subaquático é um sistema desenvolvido para a localização, rastreamento e navegação de veículos subaquáticos (ou até mesmo mergulhadores e animais), que recorre à utilização de sinais acústicos para a determinação da distância e/ou direcção de um ou vários dispositivos emissores em relação a um ou vários dispositivos receptores.

A configuração básica de um sistema USBL, consiste num *Transceiver* e um *Transponder* por cada alvo a localizar, e ainda numa unidade de processamento, que calcula a posição do *Transponder* a partir das medições efectuadas pelo *Transceiver*.

A sequência de acontecimentos, utilizada pelo USBL, para obter a localização do alvo consiste em primeiro lugar o envio de um impulso acústico pelo *Transceiver* de modo a “interrogar” os *Transponders* que encontram-se perto da localização do *Transceiver*. O sinal acústico, enviado pelo *Transceiver*, propaga-se na água até ao *Transponder*, este recebe o sinal acústico e retorna outro impulso acústico para o *Transceiver*. Após o sinal acústico chegar ao *Transceiver*, a distância r é obtida a partir do tempo de voo (*Time of Flight -TOF*) do sinal:

$$r = v \times \frac{\Delta t}{2} \quad (2.1)$$

Na equação (2.1) “ v ” corresponde à velocidade do som na água, Δt correspondente ao tempo total que o sinal acústico, enviado pelo *Transceiver*, demorou a chegar ao *Transponder* mais o tempo que o segundo sinal acústico, do *Transponder* para o *Transceiver*, demorou na resposta do *Transponder* ao receber o sinal do *Transceiver*.

De modo a obter a posição 3D [11] do *Transponder*, o *Transceiver* contém na sua base, posição de recepção do sinal acústico, um “*array*” de receptores, este “*array*” possibilita obter o ângulo do sinal acústico recebido, através da medição da diferença de fase entre o eco recebido, entre dois receptores consecutivos pertencentes à “*array*” do *Transceiver*. A diferença de fase obtém-se, por sua vez, através da diferença dos tempos de chegada do sinal acústico retornado por cada receptor da “*array*” (Figura 2.1), obtendo-se assim o ângulo da orientação do veículo ao *Transceiver*.

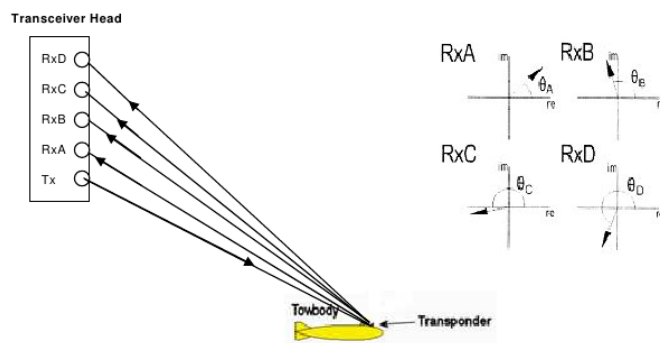


Figura 2.1: Ilustração dos atrasos de fase obtidos pelo *Transceiver* - Determinação da orientação θ .

Conhecendo o ângulo (θ) de orientação entre o *Transceiver* e o *Transponder*, é possível agora determinar a localização 3D, Figura: 2.1, do *Transponder*. Eq: (2.2, 2.3 e 2.4)

$$x = r \times \sin(\psi) \times \cos(\theta) \quad (2.2)$$

$$y = r \times \sin(\psi) \times \sin(\theta) \quad (2.3)$$

$$z = r \times \cos(\psi) \quad (2.4)$$

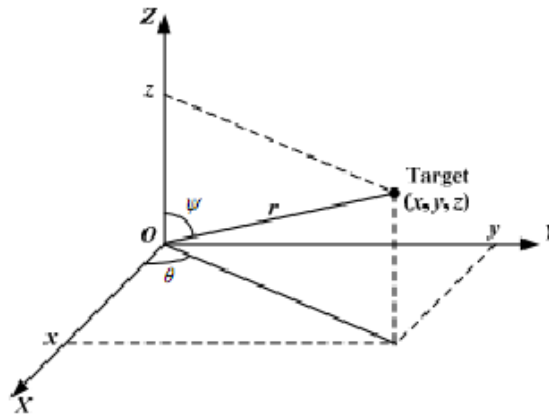


Figura 2.2: Representação de r , θ , ψ e da posição do *Transponder* (x , y , z) no referencial do *Transceiver* (O).

Um exemplo de um USBL tipicamente utilizado devido ao pequeno tamanho do *Transceiver*, fácil implementação e manuseamento do mesmo, é o USBL da *Tritech* (disponível no LSA).



Figura 2.3: USBL - Tritech.

2.2 LBL - Long Baseline Acoustic Positioning

Os sensores de posicionamento LBL (exemplo de aplicação no projecto *MARES* [12]), utilizam uma rede de transmissores colocados no fundo do mar, cujas posições são conhecidas, como pontos de referência para a navegação. Estes normalmente estão fixados na área onde se vai operar. Esta técnica permite uma precisão com um erro médio de 0.5 metros, e uma estabilidade independente da profundidade do mar. Este sistema de localização subaquático, é usado principalmente para trabalhos que requerem precisão, onde a precisão do sistema USBL não é suficiente. [13][14]

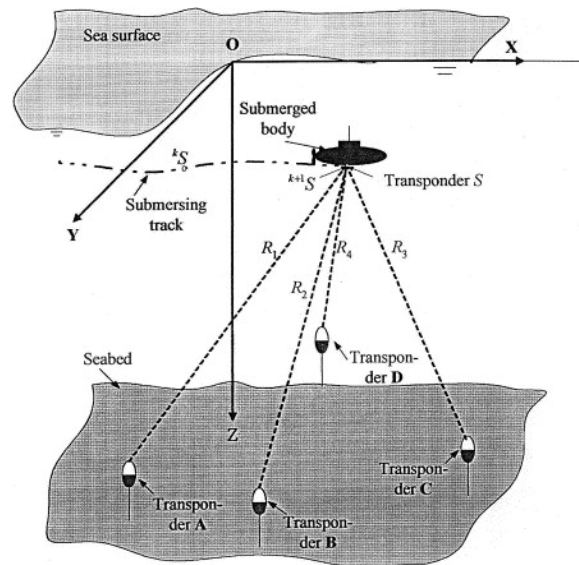


Figura 2.4: Sistema de posicionamento subaquático LBL.

A principal desvantagem é a necessidade da instalação da infraestrutura (do sistema LBL - Figura 2.4) no meio onde se vai operar. Sempre que seja necessário operar numa nova área é preciso tornar a reposicionar os faróis acústicos (*Transponder A, B, C e D*). Contudo, em geral, a área de operação do LBL é superior à do sistema USBL. Os valores R_1, R_2, R_3 e R_4 correspondem ao tempo que demora a chegar o som acústico dos *transponders* até ao veículo subaquático, com estes tempos e com as posições conhecidas dos *transponders*, é possível determina, através de trilateração, a localização do veículo subaquático [14].

2.3 Sistemas inerciais para Dead-Reckoning

Designam-se por “*Dead-Reckoning*” sistemas de localização, que continuamente processam informações captadas por sensores inerciais (acelerómetro e giroscópios) ou outros sensores internos relativos, que acumulam erros durante o processo de captura da informação [15].

A grande vantagem dos *Sensors Dead-Reckoning* provém do facto de não necessitarem de informação proveniente de referências externas, ao contrário do do sistema *USBL e LBL*, que necessitam de informação externa (*transponders*), de modo a obter a posição do veículo subaquático. Num entanto, estes tipos de sistemas de localização, não conseguem determinar, com precisão (Figura 2.5), a localização do veículo subaquático[16].

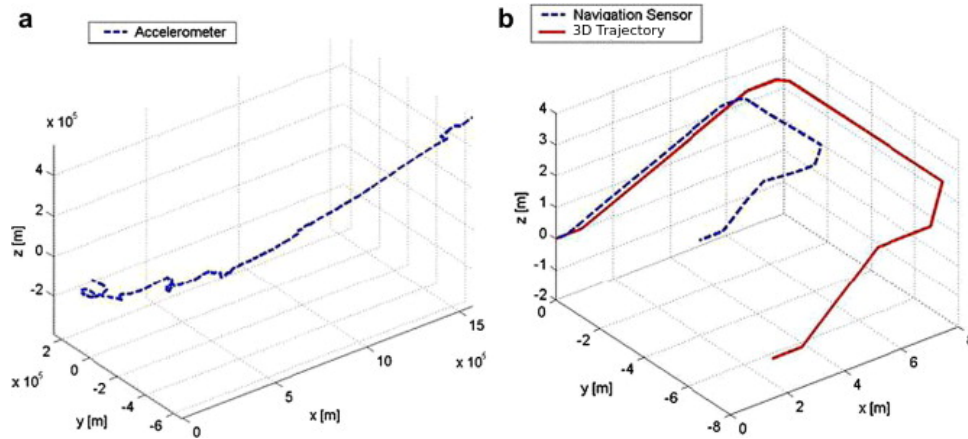


Figura 2.5: Posicionamento de um veículo utilizando apenas sensores de *Dead-Reckoning*.

Tipicamente os sensores inerciais inserem-se em uma de duas categorias, Sistemas de plataforma giro-estabilizada [17] ou sistemas fixos (*strapdown*)[18].

2.4 DVL - Doppler Velocity Log

O sistema *Doppler Velocity Log*, permite medir a velocidade, usando ondas acústicas, de um veículo subaquático (eq:2.5), através do princípio de efeito de *Doppler* [19].

O processo de determinação da velocidade, consiste na variação, da frequência de uma onda, que atinge um receptor, sendo esta diferença provocada pelo movimento da fonte emissora da onda ou do receptor (*DVL*). Através deste sistema é possível determinar se o veículo se encontra a deslocar-se mais rápido (quando há movimento de aproximação entre a fonte e o receptor, a frequência recebida é maior), ou quando a velocidade do mesmo é mais lenta (quando há movimento de afastamento entre a fonte e o receptor, a frequência recebida é menor) [20].

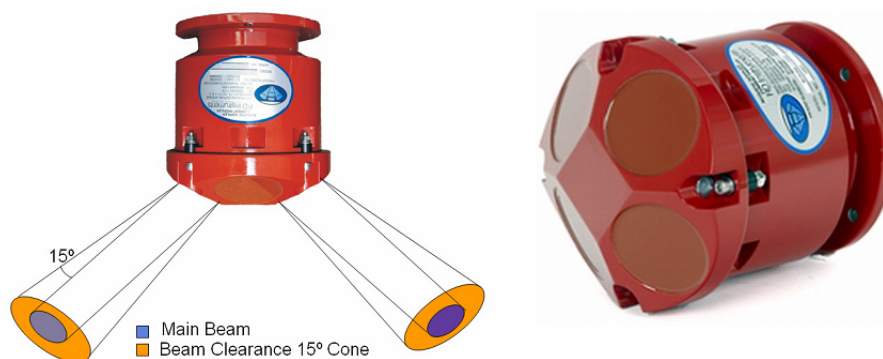


Figura 2.6: *DVL - RDI's Bottom Tracking*.

Através da equação do efeito de *Doppler* (eq:2.5), é possível determinar a velocidade do veículo. Onde F_0 corresponde à frequência recebida pelo receptor, V é a velocidade da onda

acústica emitida pela fonte (emissor), F_f representa a frequência real emitida pela fonte e V_f corresponde à velocidade da fonte. Em relação ao sinal \mp é negativo caso a fonte se esteja a aproximar e positivo quando a mesma se encontra a afastar-se.

$$F_o = F_f \left(\frac{V \pm V_0}{V \mp V_f} \right) \quad (2.5)$$

2.5 Sistemas de localização para veículos terrestres

Hoje em dia existem diversos sistemas de localização terrestre, com uma precisão relativamente alta, comparado com os sistemas de localização subaquáticos existentes actualmente. Um dos motivos que originou este facto, corresponde ao meio ambiente, sendo este muito mais acessível que o ambiente subaquático. Devido a esta vantagens é possível utilizar diversos sistemas de localização para veículos terrestres, onde por exemplo temos o GPS, VICON e Groundtruth stereo.

2.5.1 GPS

Os fundamentos básicos do GPS baseiam-se na determinação da distância entre um ponto, o receptor, a outros pontos de referência, os satélites. Sabendo a distância que separa o receptor dos três pontos (satélites) é possível determinar a posição relativa através da intersecção das três circunferências cujos raios são as distancias medidas entre o receptor e os satélites [21].

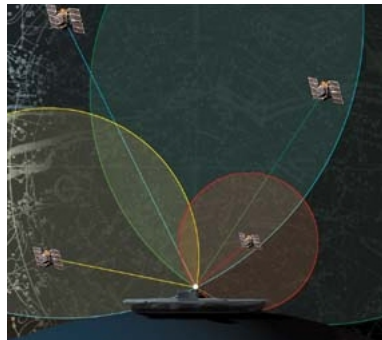


Figura 2.7: *GPS* - Localização do receptor (ponto branco).

Cada satélite transmite um sinal que é recebido pelo receptor, este por sua vez mede o tempo que os sinais demoram a chegar até ele. Multiplicando o tempo medido pela velocidade do sinal (velocidade da luz), obtemos a distância receptor-satélite, (*Distancia = Velocidade x Tempo*).

2.5.2 VICON

A tecnologia de localização VICON, consiste na localização de marcas específicas, através de múltiplas câmaras (triangulação), de modo a determinar a deslocação e orientação de um objecto. Actualmente esta tecnologia é muito aplicada para fins cinematográficos, contudo a comunidade científica de robótica tem começado aplicar a tecnologia na detecção de veículos [22].

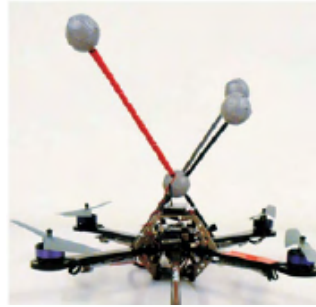


Figura 2.8: X3D quadrotor - 4 marcas para detecção com o sistema VICON [22].

2.5.3 Groundtruth stereo

O sistema *Groundtruth stereo*, tem certas semelhanças com o sistema VICON. As diferenças consistem em que num sistema *Groundtruth* usualmente são apenas utilizadas duas câmaras (*stereo vision*), outra diferença corresponde à dispensa de marcas pré-fixadas ao objecto a ser detectado. Por outras palavras neste sistema basta indicar a coordenada (no plano da imagem) do objecto a ser detectado, onde posteriormente através da correspondência do mesmo ponto nas duas imagens (*stereo correspondence*), este devolver a localização do objecto no mundo (coordenadas 3D) [23][24].

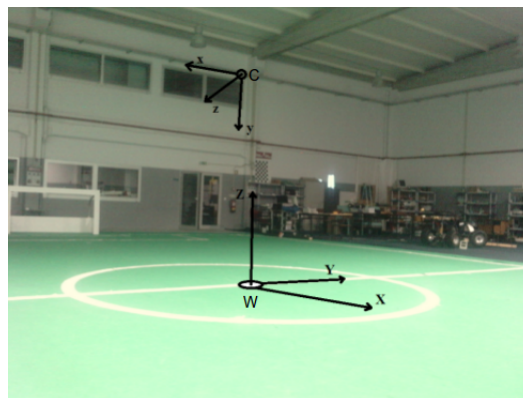


Figura 2.9: C - Referencial da câmara; W - Referencial do mundo.

(Página intencionalmente deixada em branco)

A visão stereo, corresponde ao processamento de duas imagens bidimensionais, obtidas num mesmo instante de uma determinada localização, de modo a extrair informação tridimensional geométrica correspondente ao cenário capturado pelas duas câmaras.

3.1 Modelo de câmara

O modelo de câmara mais fácil de interpretar e analisar é o modelo *Pinhole*. Neste modelo simples, a luz de um objecto é visualizada pela câmara, mas contudo apenas um único raio entra a partir de qualquer ponto em particular. Numa câmara *Pinhole* física, este ponto, é então projectado sobre um plano de imagem. Como resultado, a imagem neste plano de imagem está sempre focada, é inversa a nível de projecção e o tamanho da imagem em relação ao objecto distante é relacionado por um único parâmetro da câmara, sendo esse parâmetro a sua distância focal.

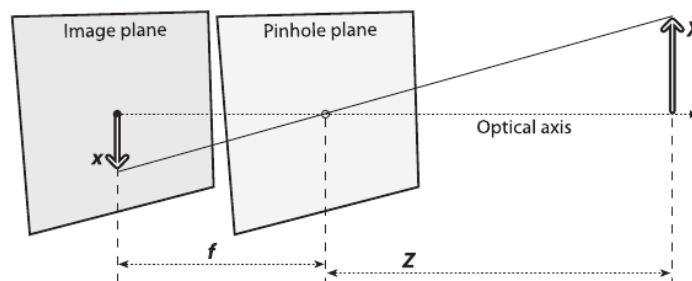


Figura 3.1: *Pinhole camera* [25].

Numa câmara *Pinhole*, a distância entre a abertura do furo e o plano de imagem é precisamente o comprimento focal. Este facto é representado na Figura: 3.1, onde f é a distância focal da câmara, Z é a distância a partir da câmara para o objecto, X é o comprimento do

objecto e x é a imagem do objecto no plano de imagem. Na figura, podemos deduzir a coordenada de x através da geometria dos triângulos semelhantes:

$$\frac{-x}{f} = \frac{X}{Z} \Leftrightarrow -x = f \frac{X}{Z} \quad (3.1)$$

Existem alguns problemas associados às câmaras *Pinhole*, sendo que o principal corresponde ao tamanho do furo, devendo este ter diâmetro finito mas diferente de zero, de modo a que uma quantidade finita de luz seja projectada sobre o plano de imagem. Contudo o tamanho do furo também não pode ser muito pequeno, porque caso contrário poderia ocorrer difracção da luz e esta será dispersa por todo o plano de imagem. Caso o diâmetro do furo seja grande demais a projecção em perspectiva não é aplicável (simplesmente no plano da imagem visualizava-se apenas um pequeno círculo).

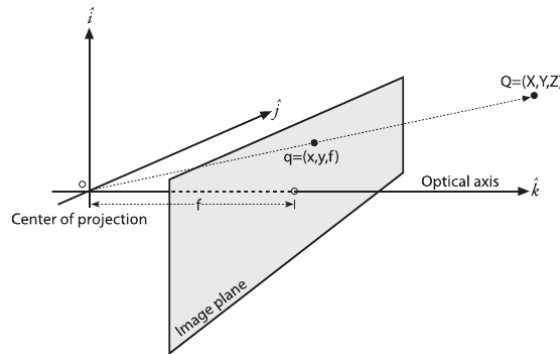


Figura 3.2: Projecção do plano de imagem em frente do plano *Pinhole*.

Um modelo equivalente, encontra-se ilustrado na Figura 3.2, onde o plano de imagem é deslocado para a frente do plano *Pinhole*. Com esta deslocação a imagem do ponto Q já não encontra-se invertida. A obtenção dos valores das coordenadas do objecto $Q(X, Y, Z)$ é dado por:

$$x = f_x \left(\frac{X}{Z} \right) + c_x \quad (3.2)$$

$$y = f_y \left(\frac{Y}{Z} \right) + c_y \quad (3.3)$$

Onde c_x e c_y correspondem ao centro óptico e f corresponde à distância focal da câmara (distância entre o centro focal e o plano de imagem).

3.2 Distorção Óptica

O processo de formação de uma imagem com uma câmara (equipada com lentes) está sujeita a fenómenos de distorção óptica. Destes, três tipos de distorção são de referir: distorção radial, tangencial e de refração. No projecto desenvolvido as principais distorções que terão de ser modelizadas e compensadas são as distorções radiais e as provocadas pela passagem da luz por diferentes tipos de materiais. As tangenciais também ocorrem, contudo são menos perceptíveis na imagem capturada.

3.2.1 Distorção radial

A distorção radial (eq: 3.5, 3.6) é facilmente distinguível na imagem (Figura 3.3), já que ocorre uma distorção geométrica, dos pixels, nas margens da imagem. Este fenómeno é normalmente chamado por efeito “barril” ou “olho de peixe”. Para este tipo de distorção o seu valor é igual a zero no centro óptico da imagem, crescendo à medida que se afasta do centro [26].



Figura 3.3: Distorção radial provocada pelas lentes[25].

Geralmente, a influência da distorção radial na imagem capturada pela câmara, é bastante acentuada quando são usadas lentes com grande abertura angular. Comparando a magnitude dos diferentes tipos de distorção, a distorção radial tem um maior impacto na imagem do que as distorções tangenciais e as distorções por refração dos raios de luz.

3.2.2 Distorção por refração

Distorção por refração, Figura: 3.4, é formada pela passagem dos raios de luz por diferentes meios, este fenómeno ocorre devido às diferentes características de refração do material onde a luz incide (Lei de Snell) [26].

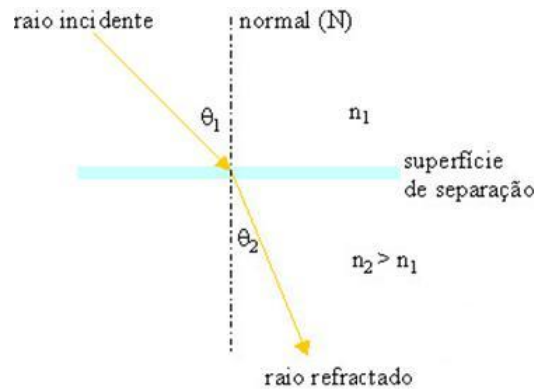


Figura 3.4: Distorção provocada pela refração dos raios de luz em diferentes meios, onde a densidade do meio $n_2 > n_1$.

Como se pode verificar na Figura: 3.4, a relação entre ângulo do raio de luz incidente (θ_1) e o ângulo do raio refractado (θ_2), pode ser obtida aplicando a lei de Snell, onde n_1 e n_2 correspondem ao índice de refração de cada um dos materiais:

$$\frac{\sin \theta_1}{\sin \theta_2} = \frac{n_2}{n_1} \tag{3.4}$$

3.2.3 Distorção tangencial

No âmbito da distorção tangencial, esta é originada por fenómenos físicos normalmente designados por efeito “prisma” ou “descentralização” (Figura 3.5). Esta distorção é resultante da impossibilidade dos fabricantes, em alinhar os eixos ópticos das lentes que compõem a objectiva [26].



Figura 3.5: Exemplo - Distorção tangencial [27].

3.2.4 Correção da distorção

Em face de um dado conjunto óptico (câmara mais lentes) é possível caracterizar-se a distorção obtendo-se um modelo que permite corrigir a imagem original de forma a obter-se uma imagem corrigida (sem distorção). Esta correção consiste, numa imagem digital, na determinação da nova localização dos pixels corrigidos, utilizando a posição inicial dos pixels da

imagem original.

No modelo de correcção, dado pelas equações 3.5 a 3.8, (x, y) correspondem à localização original do ponto com distorção a corrigir e $(x_{\text{corrigido}}, y_{\text{corrigido}})$ correspondem à nova posição do ponto sem distorção. Os parâmetros k_1 , k_2 e k_3 correspondem aos parâmetros da distorção radial e os parâmetros p_1 e p_2 correspondem aos parâmetros da distorção tangencial. O valor de r , distância ao centro óptico, pode ser obtido através da equação 3.9. [26].

Distorção Radial:

$$x_{\text{corrigido}} = x(1 + k_1r^2 + k_2r^4 + k_3r^6) \quad (3.5)$$

$$y_{\text{corrigido}} = y(1 + k_1r^2 + k_2r^4 + k_3r^6) \quad (3.6)$$

Distorção Tangencial:

$$x_{\text{corrigido}} = x + [2p_1y + p_2(r^2 + 2x^2)] \quad (3.7)$$

$$y_{\text{corrigido}} = y + [p_1(r^2 + 2y^2) + 2p_2x] \quad (3.8)$$

$$r = \sqrt{(x_d - c_x)^2 + (y_d - c_y)^2} \quad (3.9)$$

Onde (x_d, y_d) correspondem às coordenadas (x, y) de um ponto (da imagem original com distorção) e (c_x, c_y) corresponde às coordenadas do centro óptico.

3.3 Matriz de rotação e vector de translação

O modelo de projecção (eq: 3.2 ; 3.3) permite relacionar um ponto 3D (X, Y, Z) , no referencial da câmara, com a correspondente projecção bidimensional no plano de imagem (x, y) . Para obter as coordenadas de um ponto 3D de um referencial para outro diferente, é necessário obter as relações de rotação e translação entre os referenciais. Estas podem ser definidas por uma matriz de rotação R e um vector de translação \vec{t} , como se ilustra na figura seguinte, com a rotação de um ponto Q definido no referencial do objecto $\{O\}$ para o referencial da câmara.

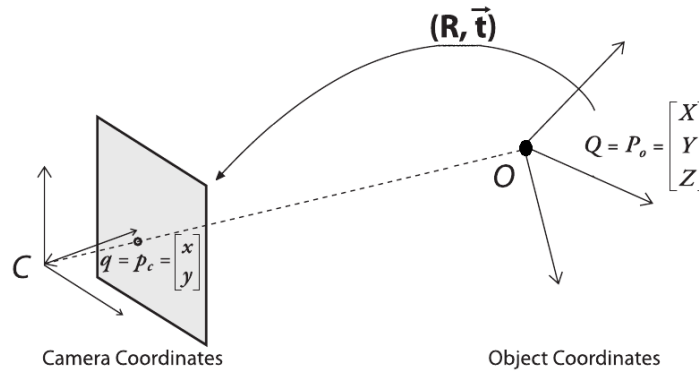


Figura 3.6: Projecção das coordenadas da câmara para as coordenadas do objecto.

Rotação 2D

Considerando um ponto $[x, y]$ num referencial $\{XY\}$, a transformação de coordenadas que nos permite obter o valor das novas coordenadas $[x', y']$ num referencial $\{X'Y'\}$, rodado com um determinado ângulo θ , é dado por uma matriz quadrada R :

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = R \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (3.10)$$

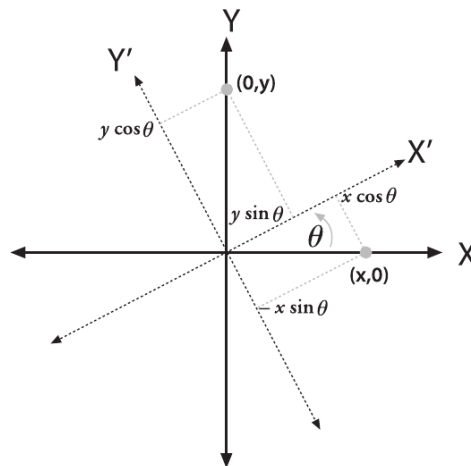


Figura 3.7: Rotação 2D.

A matriz de rotação faz com que sejam criados um novo conjunto de pontos (x', y') , utilizando como pontos de referência os pontos (x, y) .

Pode-se também visualizar que com a aplicação da matriz de rotação, ao sistema de coordenadas da Figura 3.7, com um determinado ângulo θ , é equivalente ao mesmo ponto de destino, mas em sentidos opostos (rotação em torno da origem do sistema de coordenadas, pelo mesmo ângulo θ).

Rotação 3D

Uma rotação em 3D pode ser obtida pela composição de três rotações no plano, matriz de rotação $R_x(\psi)$, $R_y(\varphi)$ e $R_z(\theta)$ [25].

$$R_x(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \psi & \sin \psi \\ 0 & -\sin \psi & \cos \psi \end{bmatrix} \quad (3.11)$$

$$R_y(\varphi) = \begin{bmatrix} \cos \varphi & 0 & -\sin \varphi \\ 0 & 1 & 0 \\ \sin \varphi & 0 & \cos \varphi \end{bmatrix} \quad (3.12)$$

$$R_z(\theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.13)$$

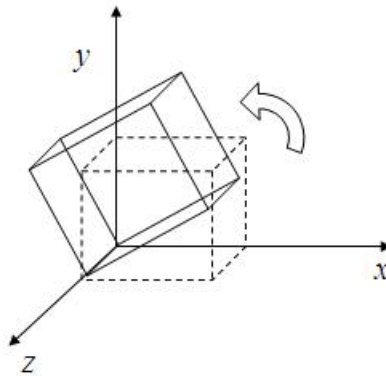


Figura 3.8: Rotação 3D de um objecto.

Se a rotação de cada um dos eixos for feita de forma sequencial, em primeiro lugar o eixo de z seguido do eixo y e posteriormente o eixo x , com os respectivos ângulos ψ (psi), φ (fi) e θ (theta), é possível obter-se uma matriz de rotação R , onde esta consiste no produto das matriz $R_x(\psi)$, $R_y(\varphi)$ e $R_z(\theta)$.

$$R(x, y, z) = R_x(\psi) \times R_y(\varphi) \times R_z(\theta) \quad (3.14)$$

Vector de translação

A translação entre um sistema de coordenadas inicial $[x, y, z]$ e um novo sistema de coordenadas final $[x', y', z']$, é dado por um vector de translação t . Sendo o ponto $[x', y', z']$ obtido por:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + t = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \quad (3.15)$$

Em que t é dado pelas coordenadas de origem do referencial $\{O\}$ no referencial $\{O'\}$ (Figura 3.9).

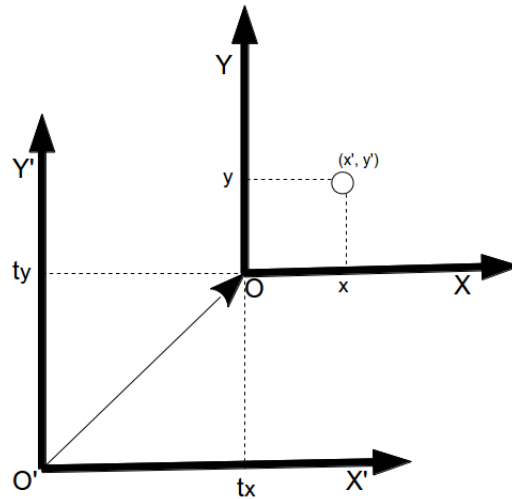


Figura 3.9: Translação do ponto (x', y') do referencial $\{O'\}$ para o referencial $\{O\}$.

3.3.1 Coordenadas homogêneas

Como foi referido anteriormente, as transformações geométricas de rotação e translação admitem uma notação matricial.

No entanto, devido à natureza do sistema de coordenadas utilizado (sistema cartesiano), as suas representações são algo inconstantes, em virtude da translação não poder ser expressa sob a forma de uma multiplicação de matrizes.

Existe no entanto um outro tipo de coordenadas muito utilizado em computação gráfica, que permite um uso mais simplificado das matrizes, designado por sistema de coordenadas homogêneas.

As coordenadas homogêneas são usadas para permitir a incorporação de matrizes separadas para a translação e rotação numa matriz única que pode ser usada para transformar coordenadas apenas por multiplicação. As coordenadas homogêneas têm mais uma dimensão do que o espaço a que respeitam.

Assim, as coordenadas homogêneas do espaço bidimensional têm a forma (x, y, w) , onde w representa um plano paralelo ao plano XY .

Desta forma as matrizes de translação e rotação em coordenadas homogêneas fica igual a:

Rotação:

$$P' = R(q) \times P \Leftrightarrow \begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (3.16)$$

$$P' = R(q) \times P \Leftrightarrow \begin{bmatrix} x' \\ y' \\ z' \\ w \end{bmatrix} = R_x(\psi) \times R_y(\varphi) \times R_z(\theta) \times \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (3.17)$$

Onde $R_x(\psi)$, $R_y(\varphi)$ e $R_z(\theta)$ correspondem a:

$$R_x(\psi) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \psi & -\sin \psi & 0 \\ 0 & \sin \psi & \cos \psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; R_y(\varphi) = \begin{bmatrix} \cos \varphi & 0 & \sin \varphi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \varphi & 0 & \cos \varphi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; R_z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.18)$$

Translação:

$$P' = T(dx, dy) \times P \Leftrightarrow \begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (3.19)$$

$$P' = T(dx, dy, dz) \times P \Leftrightarrow \begin{bmatrix} x' \\ y' \\ z' \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (3.20)$$

3.4 Calibração da câmara

Em Visão Computacional geralmente é necessário determinar os parâmetros ou coeficientes do(s) modelo(s) da(s) câmara(s) utilizado(s), que transforma(m) pontos no espaço tridimensional em pontos no plano imagem e vice-versa. A determinação dos modelos da câmara corresponde à determinação da geometria interna e das características ópticas (parâmetros intrínsecos) e/ou a orientação e posição 3D da câmara relativamente a um certo sistema de coordenadas mundo (parâmetros extrínsecos), o processo de determinação destes parâmetros é por vezes designado por estimativa da pose.

De modo a determinar os parâmetros referidos anteriormente, recorreu-se ao algoritmo

de calibração desenvolvido por Tsai [28].

O algoritmo de Tsai usa uma particularidade física do sistema de visão, sendo este o alinhamento radial. Esta particularidade é definida em função da rotação e translação relativa (excepto na componente Z), entre a câmara e os pontos de calibração [29].

O algoritmo de Tsai assume que independentemente da grandeza da distorção, a direcção do vector $O_i P_d$, vector este definido pela origem O_i e pelo ponto P_d com as respectivas coordenadas (x_d, y_d) na imagem do plano, mantém-se constante e alinhado com o vector $P_{O_z} P$, vector este formado pela origem P_{O_z} (ponto dado pela coordenada z do ponto P mais o respectivo ponto P). Assume também que distância focal f não influencia a direcção do vector $O_i P_d$ e que o ponto P no mundo é transladado e rodado em X e Y , fazendo assim com que o vector $O_i P_d$ seja sempre paralelo ao vector $P_{O_z} P$ para todos os pontos [28].

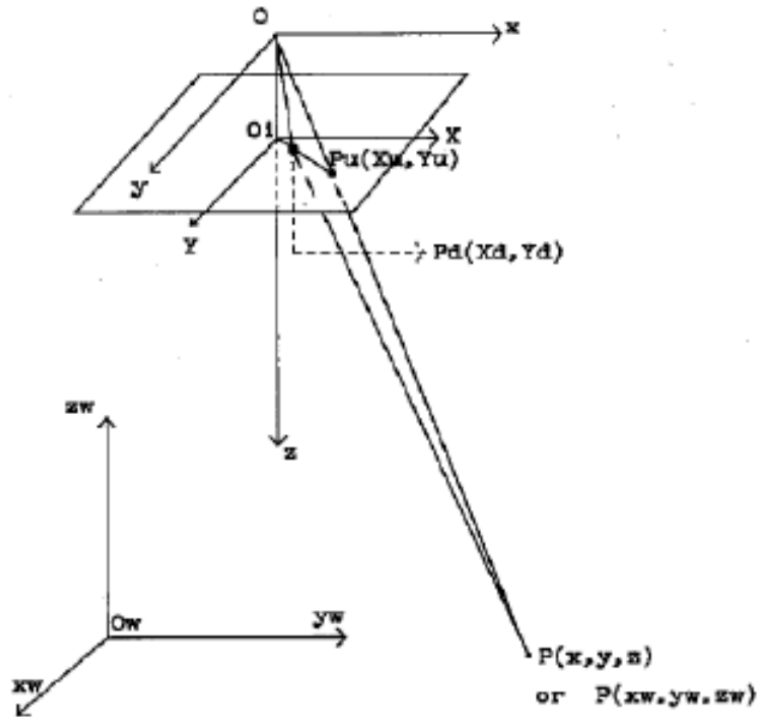
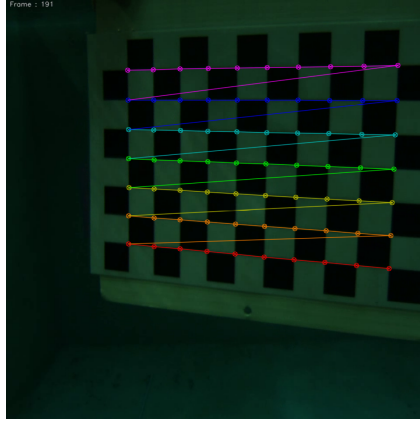


Figura 3.10: Particularidade física do algoritmo de Tsai[28].

3.4.1 Chessboard

De modo a poder implementar o algoritmo de Tsai[28], é necessário conhecer determinados pontos do mundo, e suas respectivas coordenadas no plano da imagem. Para tal recorre-se ao método de calibração, onde é utilizado um alvo tipo *Chessboard* de modo a conhecer determinados pontos físicos do mundo (x_w, y_w, z_w) , e suas respectivas coordenadas no plano de imagem da câmara (x_m, y_m) (Figura 3.11).

Figura 3.11: *Chessboard*.

Na Figura 3.11 é possível visualizar os pontos físicos do mundo (pontos ilustrados por circunferências), sendo depois calculadas as coordenadas no plano de imagem da câmara. O algoritmo de Tsai, baseia-se usando as particularidades anteriormente descritas, onde através destas particularidades é agora possível calcular os parâmetros intrínsecos (M) e extrínsecos (Ex).

3.4.2 Parâmetros Intrínsecos

Os parâmetros intrínsecos dizem respeito aos parâmetros do conjunto câmara mais lente. Estes (parâmetros intrínsecos) relacionam as coordenadas do ponto de imagem com as coordenadas do referencial da câmara, onde também correspondem a uma transformação em 2D e pode ter incluído um componente de distorção não-linear na transformação.

$$x = -(x_{im} - c_x)s_x \quad (3.21)$$

$$y = -(y_{im} - c_y)s_y \quad (3.22)$$

As características que constituem os parâmetros intrínsecos são:

- Distância focal;
- Tamanho do pixel (s_x, s_y);
- Coeficientes de distorção;
- Centro da imagem (c_x, c_y).

3.4.3 Parâmetros Extrínsecos

Os parâmetros extrínsecos correspondem à transformação do sistema de coordenadas do mundo para o sistema de coordenadas da câmara, sendo estes centrados no centro focal.

No total existem seis parâmetros extrínsecos, três correspondem à rotação do plano do mundo para o plano da câmara (R) (ângulos de *Euler* - yaw (θ), pitch (ψ) e roll (φ)) e os restantes três parâmetros correspondem à respectiva translação do vector (T) (Tsai [28]).

$$R = \begin{bmatrix} \cos \psi \cos \theta & \sin \psi \cos \theta & -\sin \theta \\ -\sin \psi \cos \varphi + \cos \psi \sin \theta \cos \varphi & \cos \psi \cos \varphi + \sin \psi \sin \theta \sin \varphi & \cos \theta \sin \varphi \\ \sin \psi \sin \varphi + \cos \psi \sin \theta \cos \varphi & -\cos \psi \sin \varphi + \sin \psi \sin \theta \cos \varphi & \cos \theta \cos \varphi \end{bmatrix} \quad (3.23)$$

$$T = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \quad (3.24)$$

Conhecidas as rotações de cada câmara (R_l , R_r , R_L e R_R) ao plano de calibração, para obter a rotação e translação do sistema, por outras palavras a relação existente entre a câmara da direita com a câmara da esquerda, é possível agora determinar a matriz de rotação R_{final} , onde a câmara de referência corresponde à câmara da esquerda:

$$P_{cam} = R \times P_w + T \quad (3.25)$$

$$P_l = R_L \times P_w + T_l \quad (3.26)$$

$$P_r = R_R \times P_w + T_r \quad (3.27)$$

$$R_{final} = R_r \times R_l^T \quad (3.28)$$

$$P_r = R_{final} \times P_l + T_{final} \quad (3.29)$$

A translação final (T_{final}) corresponde a:

$$P_r = R_R \times R_L^T (P_l - T_l) + T_r \quad (3.30)$$

$$P_r = R \times P_l + \frac{T_r - R \times T_l}{T_{final}} \quad (3.31)$$

$$T_{final} = T_l - R^T \times T_r \quad (3.32)$$

Onde T_l e T_r correspondem nomeadamente à translação pertencente à câmara esquerda e direita.

3.4.4 Matriz Fundamental e Essencial

A matriz Essencial (E), matriz 3 x 3, contém informação relativa à translação e rotação (eq:3.33), no mundo físico, de uma câmara para a outra, neste caso da cam_r para a cam_l , essencialmente é constituída por parâmetros métricos correspondentes à relação entre as duas

câmaras, não contendo nenhuma informação relativa as imagens.

$$E = S \times R \quad (3.33)$$

$$S = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix} \quad (3.34)$$

A matriz F , matriz 3 x 3, possui os mesmos parâmetros que a matriz E , mas com um acréscimo dos respectivos valores dos parâmetros extrínsecos de cada uma das câmaras. De modo a determinar a matriz fundamental F , podemos relacionar a mesma com a matriz essencial E . (eq:3.35)[30];[31]

$$F = K_r^{-T} \times E \times K_l^{-1} \quad (3.35)$$

Onde K_r e K_l correspondem aos parâmetros de calibração obtidos pelo algoritmo de Tsai e S corresponde à matriz anti-simétrica definida pelo vector de translação. Uma matriz anti-simétrica é uma matriz cuja a sua matriz transposta coincide com sua matriz oposta ($T^t = -T$), matriz produto externo (*Cross Product*).

3.5 Homografia

Em visão computacional, homografia significa a correspondência de pontos idênticos, em duas imagens distintas, tiradas num mesmo local e instante. Esta correspondência pode ser representada por uma matriz de homografia H .

A matriz de homografia ($H - 3 \times 3$), é constituída por duas partes, onde a primeira consiste na transformação física que permite localizar o plano do objecto que queremos detectar na imagem e a segunda consiste nos parâmetros intrínsecos da câmara (M) [32].

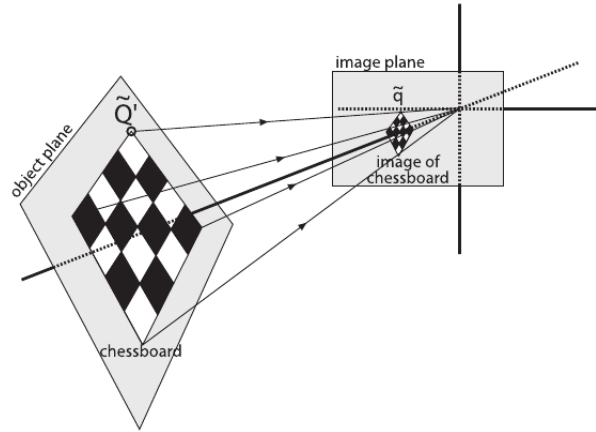


Figura 3.12: Visualização do plano do objecto descrito pela matriz H .

Utilizando a Figura: 3.12 como exemplo, é possível determinar a coordenada \tilde{q} através da aplicação da matriz H . Se utilizamos coordenadas homogêneas para descrever o ponto \tilde{Q} e \tilde{q} então temos:

$$\tilde{Q} = [X \ Y \ Z \ 1]^T \quad (3.36)$$

$$\tilde{q} = [x \ y \ 1]^T \quad (3.37)$$

Então o cálculo do ponto \tilde{q} recorrendo à matriz H corresponde a:

$$\tilde{q} = s \times H \times \tilde{Q} \quad (3.38)$$

Onde o parâmetro s , aplicado à equação eq:3.38 corresponde a um factor de escala arbitrário, de modo a que neste exemplo fosse explicito a aplicação da matriz H para o cálculo de \tilde{q} .

A transformação física da matriz H corresponde à soma de uma rotação R e de uma translação t , que relaciona as coordenadas do objecto com o plano de imagem. Continuando aplicar as coordenadas homogêneas, podemos fundir toda esta informação numa matriz W (matriz da transformação física da matriz H).

$$W = [R \ t] \quad (3.39)$$

Aplicando a matriz M e W à equação eq:3.38 obtemos:

$$\tilde{q} = s \times M \times W \times \tilde{Q} \quad (3.40)$$

Onde a matriz M é igual a:

$$M = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (3.41)$$

Como demonstrado acima, a matriz de homografia H , descrita em coordenadas homogêneas, corresponde a:

$$H = [W M] \quad (3.42)$$

3.6 Rectificação stereo

O objectivo principal da rectificação stereo é alinhar as duas imagens, uma de cada câmara, num mesmo plano de imagem, de maneira a que cada linha de pixeis de uma imagem esteja alinhada com a linha de pixeis da segunda imagem.

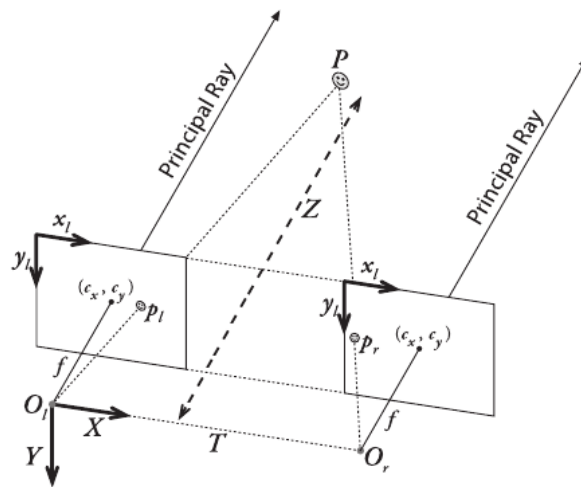


Figura 3.13: Sistema de coordenadas alinhado pelos planos de imagem.

3.6.1 Geometria Epipolar

A geometria correspondente à rectificação stereo denomina-se Geometria Epipolar, onde esta consiste na combinação de duas imagens, já rectificadas, com novos pontos denominados de “Epipolos (e_l e e_r)” (Figura 3.14).

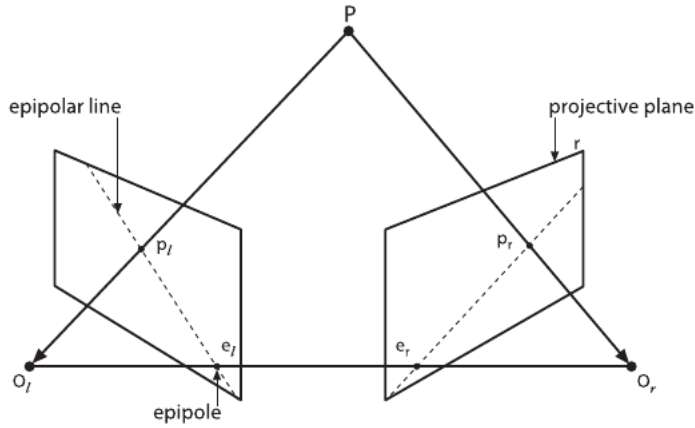


Figura 3.14: Geometria Epipolar - Exemplificação.

Para cada uma das câmaras existe agora dois centros de projecção diferentes (O_l e O_r) e um par de planos Π_l e Π_r (“*Projective Plane*”). O ponto P , no mundo físico, terá uma projecção em cada um dos planos (p_l e p_r).

Os pontos e_l e e_r são denominados por epipolos (“*Epipole*”) dos planos Π_l e Π_r respectivamente. A localização destes novos pontos é definida por uma recta que liga os dois centros focais das câmaras (O_l e O_r), sendo esta recta também reconhecida como “*Baseline*”.

Os planos de imagem Π_l e Π_r são formados através dos pontos p_l e p_r respectivamente com os pontos epipolares e_l e e_r . As linhas projectadas pelos pontos $p_l e_l$ e $p_r e_r$ são denominadas de linhas epipolares (“*Epipole Lines*”).

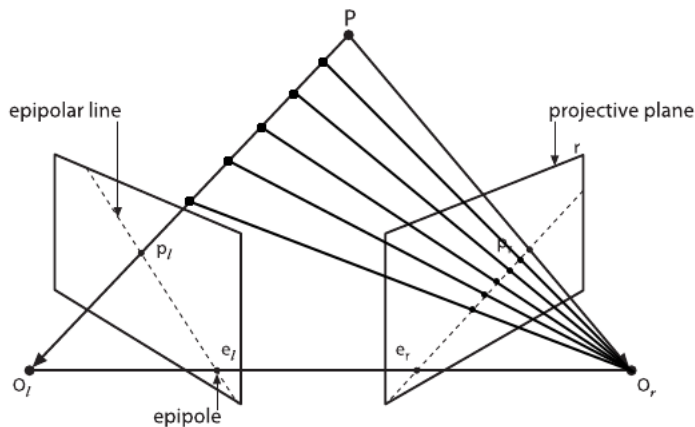


Figura 3.15: Geometria Epipolar - Vantagens.

Para compreender a utilidade dos epipolos primeiro temos recordar que, quando vemos um ponto, no mundo físico, projectado no plano de imagem (Π_l ou Π_r), o ponto pode estar localizado em qualquer posição da recta definida pelo centro de projecção de imagem (O_l ou O_r) até ao ponto $p_l e_l$ ou $p_r e_r$, contudo não é possível determinar a que distância se encontra

o ponto da câmara, sem recorrer a uma segunda câmara.

Seguindo como exemplo a Figura: 3.15, se o ponto P for apenas visualizado pela câmara da esquerda (O_l), a informação retirada, do ponto, é apenas a sua localização p_l , contudo se aplicarmos a informação da câmara da direita (O_r), já é possível verificar a que distância se encontra o ponto P da câmara esquerda (O_l).

Com a utilização da geometria epipolar é possível:

- Em cada ponto 3D, visto pela perspectiva das câmaras, está contido num plano epipolar que intersecta cada imagem na sua respectiva linha epipolar.
- Dado um ponto de uma imagem, o seu ponto correspondente na outra imagem deve estar ao longo da linha epipolar correspondente da segunda imagem. Este facto é conhecido como restrição epipolar.
- A restrição garante que todos os pontos de uma linha epipolar surgirão na mesma ordem nas imagens rectificadas e que cada ponto possui um único correspondente na imagem oposta.

Conhecendo a matriz Fundamental (F) e as coordenadas do ponto p_l , é possível determinar a linha epipolar da imagem da direita (O_r), dada por:

$$le_r = F \times p_l \quad (3.43)$$

Linha Epipolar da câmara da esquerda = le_r

3.6.2 Algoritmo da rectificação stereo

De modo a realizar a rectificação stereo, são necessárias as matrizes dos parâmetros intrínsecos (M) e de distorção (D) de cada uma das câmaras e as matrizes de rotação (R) e de translação (T) correspondentes à relação geométrica entre a câmara da direita com a câmara da esquerda. Sabendo os valores destes parâmetros é possível realizar a rectificação stereo.

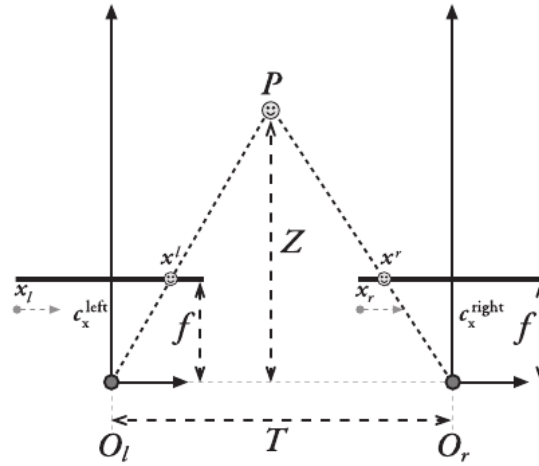


Figura 3.16: Situação ideal para rectificação stereo.

Numa situação ideal, onde as linhas epipolares de uma imagem estão perfeitamente alinhadas com as linhas epipolares da segunda imagem, que cada pixel correspondente as linhas epipolares estão devidamente alinhados entre si e que o objecto a ser detectado aparece em ambas as imagens, é possível determinar a que profundidade (*Depth*) encontra-se o objecto P (ponto). Figura: 3.16

$$\frac{T - (x^l - x^r)}{Z - f} = \frac{T}{Z} \quad (3.44)$$

$$Z = \frac{f \times T}{x^l - x^r} \quad (3.45)$$

$$T = |c_x^{left} - c_x^{right}| \quad (3.46)$$

$$d = x^{left} - x^{right} \quad (3.47)$$

Nas equações 3.44 e 3.45, Z corresponde ao valor do *Depth*, T corresponde ao módulo da diferença entre o centro focal da câmara da esquerda e do centro focal da câmara da direita (3.46), f corresponde à distância focal da câmara (onde nesta situação ideal a distância é idêntica em ambas as câmaras), d é igual ao valor da coordenada x_l , detectada na câmara esquerda, menos o valor da coordenada x_r , detectada na câmara direita (3.47).

Disparidade entre imagens

Na visão stereo, disparidade consiste na diferença que existe entre as distâncias dos pontos P_l e P_r . Estes consistem nas projecções do mesmo ponto $P(x, y)$, do objecto, nas imagens da esquerda e da direita respectivamente. Desta diferença resulta a percepção da distância em relação às câmaras.

A distância é inversamente proporcional à disparidade, isto é, quando a disparidade entre as imagens é praticamente nula a distância a que encontra-se o objecto das câmaras é acentuada, quando a disparidade é elevada a distância do objecto as câmaras é pequena. Figura 3.17

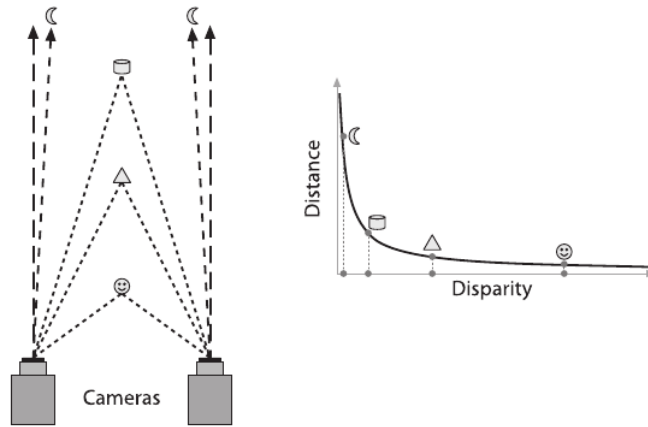


Figura 3.17: Disparidade versus Distância do objecto às câmaras.

O cálculo da disparidade (*disp*) de um ponto *P* entre duas imagens distintas, mas capturadas no mesmo instante, é efectuado através da diferença de coordenadas do ponto, no eixo de *x*, de cada uma das câmaras. Eq:3.48

$$disp = |P_{x1} - P_{x2}| \quad (3.48)$$

P_{x1} e P_{x2} correspondem às coordenadas, no eixo do *x*, do ponto *P* respectivamente em cada uma das câmaras.

Processo de rectificação stereo

Actualmente o processo mais utilizado para a rectificação stereo, correspondem ao método *Bouguet*. Este por sua vez este é uma simplificação do primeiro método apresentado por Tsai [33] e Zhang ([34]; [35]). Contudo existe também o método de rectificação desenvolvido por *Hartley* [36];[37].

***Bouguet* método**

Dada a matriz de rotação e o vector de translação (*R*, *T*) entre as imagens stereo, o algoritmo de rectificação stereo (*Bouguet*) tenta minimizar a quantidade de mudanças de projecções que são efectuadas, para cada uma das duas imagens e assim minimizar as distorções resultantes das mesmas, enquanto maximiza a área de visualização comum entre as duas imagens [25].

De modo a minimizar a distorção, provocadas pelas consecutivas mudanças de projecção, a matriz de rotação R que corresponde à matriz de rotação entre o plano de imagem da câmara da direita para a câmara da esquerda, é dividida em duas novas matrizes de rotação, sendo estas matrizes correspondentes a cada uma das câmaras (r_l matriz de rotação correspondente à câmara da esquerda e r_r a matriz de rotação correspondente à câmara da direita).

Cada sistema de coordenadas, de cada uma das câmaras, faz uma rotação igual a $1/2$ de R (matriz de rotação), de modo a que os seus raios principais fiquem paralelos à soma vectorial dos seus respectivos raios originais, fazendo assim com que exista um alinhamento coplanar mas não a nível horizontal. Um alinhamento coplanar consiste em ter pelo menos três pontos, ou mais, distintos mas ambos pertencentes ao mesmo plano de imagem.

Para determinar a matriz de rotação (R_{rect}) que desloca o epipolo da câmara da esquerda até ao infinito e que alinha as linhas epipolares a nível horizontal, começa-se a matriz pela orientação do epipolo e_1 (vector). Conhecendo o centro focal (c_x e c_y) da câmara da esquerda, a direcção do epipolo pertence ao vector de translação, entre os centros focais, das duas câmaras:

$$e_1 = \frac{T}{\|T\|} \quad (3.49)$$

O próximo vector, e_2 , tem de ser ortogonal ao e_1 .

Para o vector e_2 , utiliza-se a orientação ortogonal do raio principal (que tende a ser ao longo do plano da imagem), multiplicando de seguida pelo produto vectorial de e_1 com a orientação do raio principal, sendo depois normalizado de modo a obter o vector e_2 . Eq:3.50

$$e_2 = \frac{[-T_y \ T_x \ 0]^T}{\sqrt{T_x^2 + T_y^2}} \quad (3.50)$$

O terceiro vector (e_3) é ortogonal a e_1 e e_2 , onde a sua obtenção é efectuada através do produto de e_1 e e_2 :

$$e_3 = e_1 \times e_2 \quad (3.51)$$

Conhecendo os três vectores (e_1 , e_2 e e_3), é possível construir agora a matriz R_{rect}

$$R_{rect} = \begin{bmatrix} (e_1)^T \\ (e_2)^T \\ (e_3)^T \end{bmatrix} \quad (3.52)$$

A matriz R_{rect} “roda” o centro de projecção da câmara da esquerda, de modo a que as linhas epipolares sejam horizontais e com epipolos infinitos. O alinhamento horizontal das

duas câmaras é obtido através de (r_l matriz de rotação correspondente à câmara da esquerda e r_r a matriz de rotação correspondente à câmara da direita):

$$R_l = R_{rect} \times r_l \quad (3.53)$$

$$R_r = R_{rect} \times r_r \quad (3.54)$$

Neste método as matrizes dos parâmetros intrínsecos ($M1$ e $M2$) de ambas as câmaras também são rectificadas, de modo a obter duas novas matrizes de parâmetros intrínsecos (M_{rect_l} e M_{rect_r}), contudo estes novos parâmetros vêm retornados nas matrizes de projecção (P) de cada uma das câmaras (P_l e P_r).

$$P_l = M_{rect_l} \times P'_l = \begin{bmatrix} f_{x_l} & \alpha_l & c_{x_l} \\ 0 & f_{y_l} & c_{y_l} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (3.55)$$

$$P_r = M_{rect_r} \times P'_r = \begin{bmatrix} f_{x_r} & \alpha_r & c_{x_r} \\ 0 & f_{y_r} & c_{y_r} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (3.56)$$

Onde α_l e α_r correspondem a um parâmetro de aceitação de distorção dos pixels, onde actual nas câmaras modernas o seu valor é igual a zero

Através da utilização das matrizes de projecção (P_l e P_r) é possível pegar num ponto 3D e torna-lo num ponto 2D, em coordenadas homogêneas. Eq:3.57

$$P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ w \end{bmatrix} \quad (3.57)$$

Onde as coordenadas da imagem podem ser calculadas como ($x/w, y/w$). Pontos 2D também podem ser transformados em pontos 3D, sendo para isso necessário as suas coordenadas de imagem e respectivas matrizes de parâmetros intrínsecos (M). A matriz de transformação (Q) é constituída por:

$$Q = \begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 0 & f \\ 0 & 0 & \frac{-1}{T_x} & \frac{(c_x - c'_x)}{T_x} \end{bmatrix} \quad (3.58)$$

Todos os parâmetros que constituem a matriz Q correspondem à câmara da esquerda,

excepto o parâmetro c'_x , este corresponde ao centro focal (em x) da imagem da direita.

Para determinar um ponto 3D através de um ponto 2D (onde ambas as coordenadas têm de ser homogêneas), é necessário incluir a disparidade d , existente entre cada uma das imagens.

$$Q \begin{bmatrix} x \\ y \\ d \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} \quad (3.59)$$

Onde as coordenadas 3D podem ser calculadas através de X/W , Y/W e Z/W .

Na Figura 3.18 podemos visualizar os passos necessários para a rectificação stereo.

- Passo *a* - Captura de uma imagem no mesmo instante, mas de cada uma das câmaras.
- Passo *b* - Remoção da distorção de cada uma das imagens.
- Passo *c* - Rectificação stereo - posicionar as duas imagens num mesmo plano de imagem.

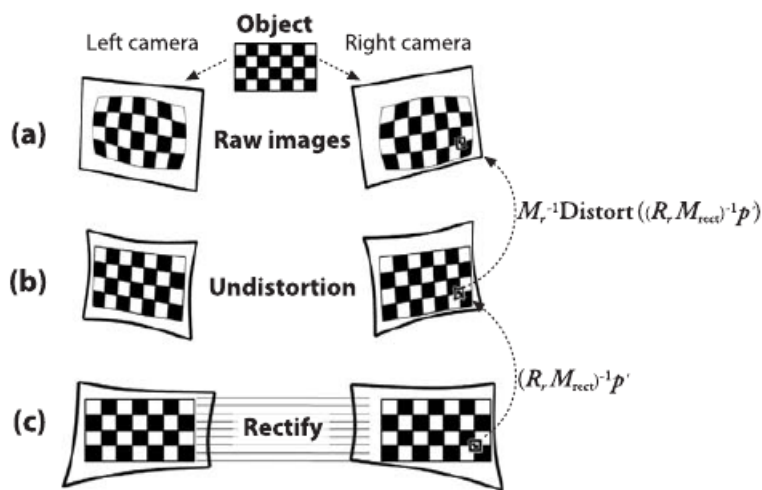


Figura 3.18: Passos necessários para a rectificação stereo [25].

3.7 Determinação das coordenadas (X, Y, Z) do objecto no referencial do mundo

Após a realização da rectificação stereo, é agora possível determinar a localização do objecto num mundo 3D. Para tal é necessário passar das coordenadas do objecto (x, y), no plano da

imagem rectificada, para as coordenadas do mundo (X, Y, Z).

Este processo tem duas etapas, sendo a primeira a determinação do ponto pretendido (P_c) na imagem sem rectificação, utilizando para tal o ponto ($P_{c_{rect}}$) da imagem rectificada e a matriz de rotação inversa (R_l^{-1}) da câmara esquerda.

$$P_c = R_l^{-1} \times P_{c_{rect}} \quad (3.60)$$

Na segunda etapa da determinação das coordenadas do objecto, no referencial do mundo, é necessário fazer uma rotação e translação do referencial da câmara para o referencial do mundo.

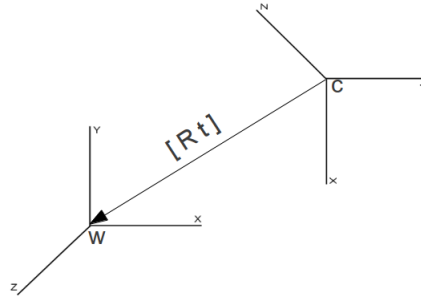


Figura 3.19: Rotação e translação do referencial da câmara para o referencial do mundo.

Obtendo as coordenadas do objecto (P_c), do plano de imagem sem rectificação, a obtenção coordenada do objecto (P_w), no referencial do mundo é obtido por:

$$P_w = R_c^{-1} \times P_c - \vec{t}_w \quad (3.61)$$

$$P_c = R_l \times P_w + R \times \vec{t} \quad (3.62)$$

Onde (R_c^{-1}) corresponde à matriz de rotação inversa, entre o referencial da câmara (esquerda) e o referencial do mundo e (\vec{t}_w) corresponde ao vector de translação, entre o referencial da câmara (esquerda) e o referencial do mundo.

(Página intencionalmente deixada em branco)

Um dos problemas mais frequentes em visão computacional, consiste na detecção de um determinado objecto ou elemento do ambiente. Actualmente existem inúmeras abordagens ao problema de tratamento de imagem de forma a obter características da mesma e em particular a identificação de um objecto na imagem. O processo de classificação da uma imagem, segundo as suas características (ex: morfologia e cor), designa-se usualmente por segmentação. Note-se que a detecção de um objecto na imagem, requer sempre um conhecimento prévio das características do mesmo, sejam elas morfologia do objecto, cor ou até mesmo uma imagem prévia do mesmo (ou similar).

4.1 Espaços de cor

Quase todas as cores do espectro da luz visível podem ser reproduzidas adicionando partes distintas de luz vermelha, verde e azul, onde de seguida o sensor *CCD* capta essa mesma luz. Um sensor *CCD*, consiste numa grelha de condensadores que são carregados por um elemento fotossensível, que por sua vez encontra-se coberto por um filtro vermelho, verde e azul. Estes filtros apenas deixam passar uma das componentes da cor, absorvendo todas as outras. [38] A esta disposição de filtros é designada por *Array de Bayer* (Figura: 4.1). Contudo uma imagem pode ser monocromáticas (tons de cinzento) ou cromática (imagem a cores). De maneira a obter uma imagem monocromática, não são utilizados filtros para filtragem, fazendo assim com que o *CCD* associe a intensidade da luz capturada, a um determinado valor de cinzento.

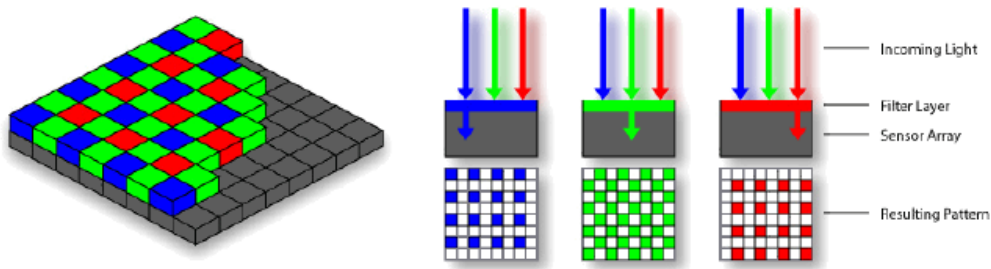


Figura 4.1: Array de Bayer.

Imagem monocromática - Tons de cinzento

Computacionalmente, as imagens monocromáticas são armazenadas utilizando-se oito bits (um byte) por pixel, o que permite 256 (0 a 255) intensidades possíveis. Contudo, a precisão deste formato não é grande, gerando muitas vezes confusão nas cores. Figura:4.2 transferência, impressão ou reprodução, e seu processamento por meios electrónicos.

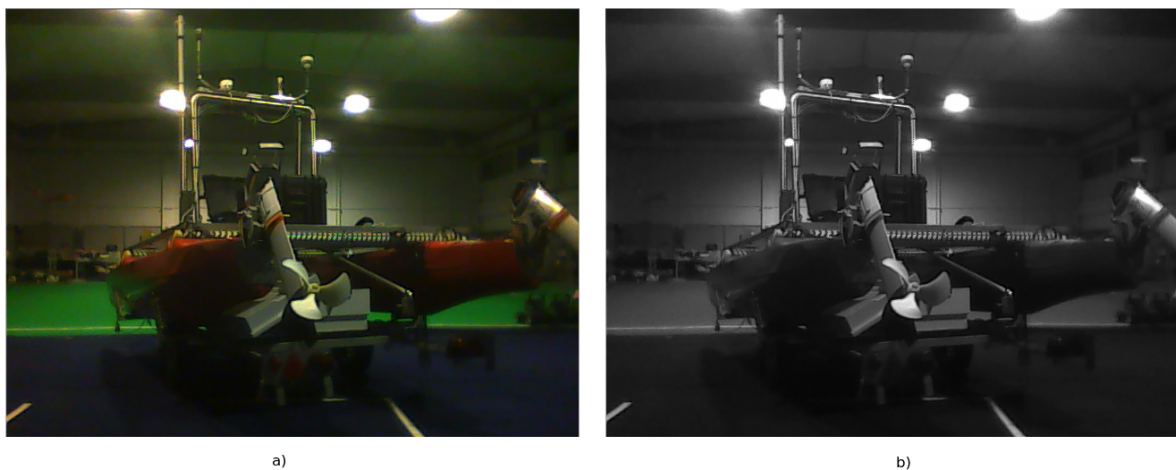


Figura 4.2: Tons de cinza (“b”); Imagem real (“a”) - ROAZ II.

Para converter qualquer cor em seu nível aproximado de cinza, deve-se primeiro obter suas primitivas vermelho, verde e azul (da escala RGB). Adiciona-se então 30% do vermelho mais 59% do verde mais 11% do azul, independente da escala utilizada (0.0 a 1.0, 0 a 255, 0% a 100%.) O nível resultante é o valor de cinza desejado.

Imagem cromática

As cores só existem se três componentes estiverem presentes, um observador, um objecto e luz. Apesar da luz branca ser normalmente encarada como "sem cor", na realidade contém todas as cores do espectro visível. Quando a luz branca atinge um objecto, este absorve

algumas das cores e reflecte as restantes. Somente as cores reflectidas contribuem para a interpretação da cor feita pelo observador.

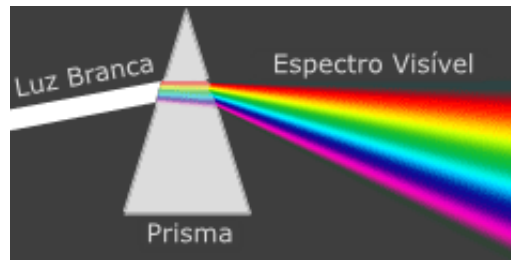


Figura 4.3: Luz branca - cor.

O espaço de cores **RGB** resulta da abreviatura do sistema de cores formado por Vermelho (*Red*), Verde (*Green*) e Azul (*Blue*). O propósito principal do sistema RGB é a reprodução de cores em dispositivos electrónicos como monitores de TV e computador, "datashows", scanners e câmaras digitais.

Uma cor no modelo de cores **RGB** pode ser descrita pela indicação da quantidade de vermelho, verde e azul que contém cada pixel da imagem. Cada um dos valores, pode variar entre um mínimo (completamente escuro - 0) e um máximo (completamente intenso - 255). Quando todas as cores estão no mínimo, o resultado é preto. Se todas estão no máximo, o resultado é branco. Figura:4.4

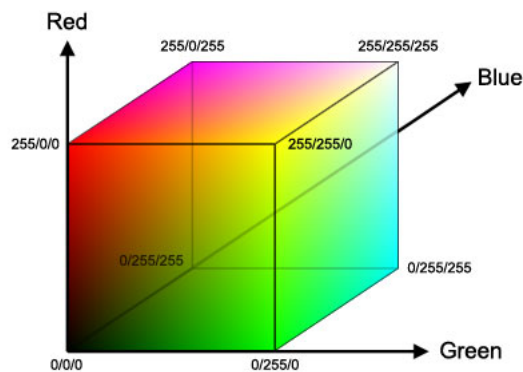


Figura 4.4: Tons de cor **RGB**

HSV e HSL

HSV é a abreviatura para o sistema de cores formadas pelas componentes *hue* (tonalidade), *saturation* (saturação) e *value* (valor). Onde cada parâmetro corresponde a:

- *hue* - Determina o tipo de cor, envolvendo todas as cores do espectro, desde o vermelho até o violeta, mais o magenta. Atinge valores de 0 a 360, mas para algumas aplicações, esse valor é normalizado de 0 a 100%.
- *saturation* - Quanto menor o valor, mais com tom de cinzento aparecerá na imagem. Atinge valores de 0 a 100%.
- *value* - Define o brilho da cor. Atinge valores de 0 a 100%.

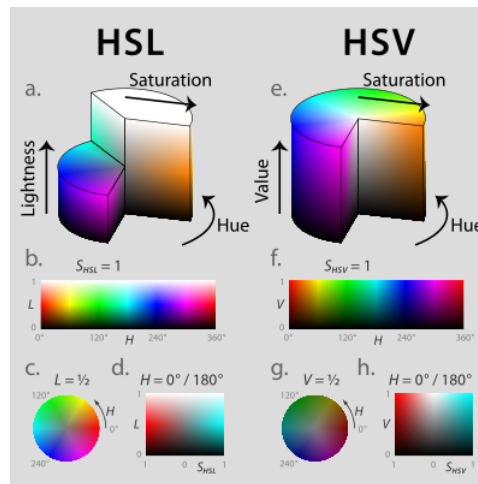


Figura 4.5: Tons de cor *HSL* e *HSV*.

O espaço de cor *HSL*, divide a informação da cor em três parâmetros, tonalidade (*hue*), saturação (*saturation*) e luminância (*lightness*). O *HSL* é idêntico ao *HSV*, mas contudo separa melhor a noção de saturação e da luminância. No *HSL*, a saturação vai desde a cor totalmente saturada até ao cinza, contudo no *HSV*, vai desde a cor totalmente saturada até ao branco [38].

4.2 Template Match

O método de *Template Match*, consiste numa comparação de uma imagem com uma outra imagem a analisar (*Template*). Esta comparação é feita deslocando o *Template* pela imagem a analisar, e a cada posição atribuir um valor, dado pela eq:4.1. Tipicamente tem como resultado um mapa 2D, com dimensões igual à dimensão dos deslocamentos que o *Template* faz sobre a imagem (Figura: 4.6), com os valores calculados na procura. A posição do mapa

resultante com valor mais próximo de zero, corresponde à posição onde o *Template* melhor se ajusta na imagem original [39].

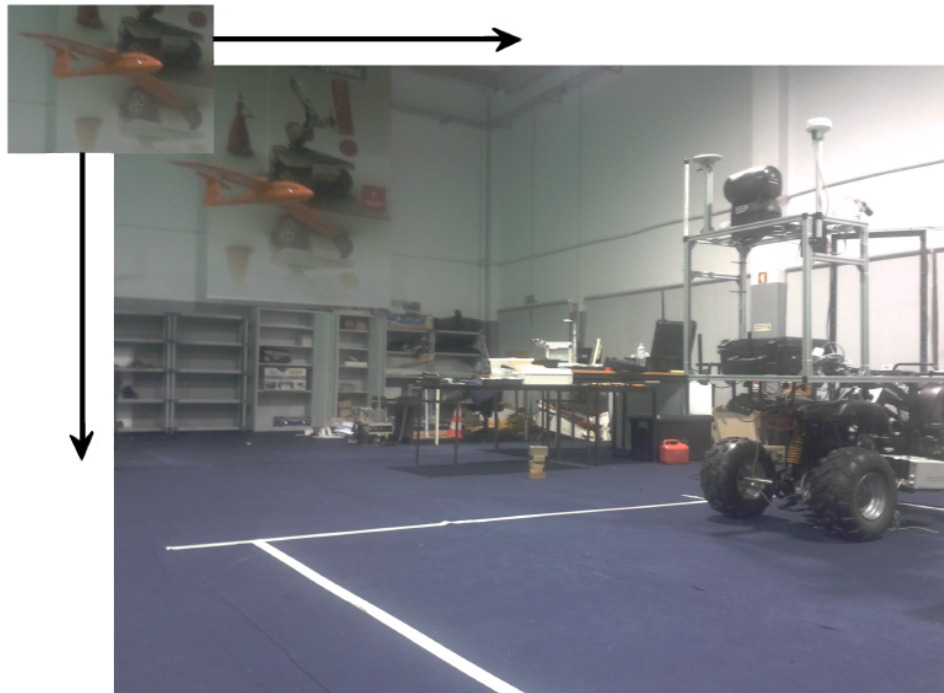


Figura 4.6: Deslocação do *Template* (*FALCOS - Autonomous Aerial Vehicles*) na imagem original.

$$R(X) = \frac{\sum_{X'} T(X') \times I(X - X')}{\sqrt{\sum_{X'} T(X')^2 \times I(X - X')^2}} \quad (4.1)$$

O valor retornado, pela eq:4.1, corresponde à correlação cruzada normalizada, ao longo do deslocamento do *Template* por todos os pixels da imagem original, onde $R(X)$ corresponde à matriz dos valores obtidos para cada pixel da imagem original, $T(X')$ é o *Template* a ser detectado, $I(X)$ corresponde à imagem original e em que X é a coordenada 2D (x, y) . Para além da correlação cruzada normalizada, existe ainda a correlação normalizada, diferença de quadrados, diferença de quadrados normalizada, coeficiente de correlação e coeficiente de correlação normalizada [39][40].

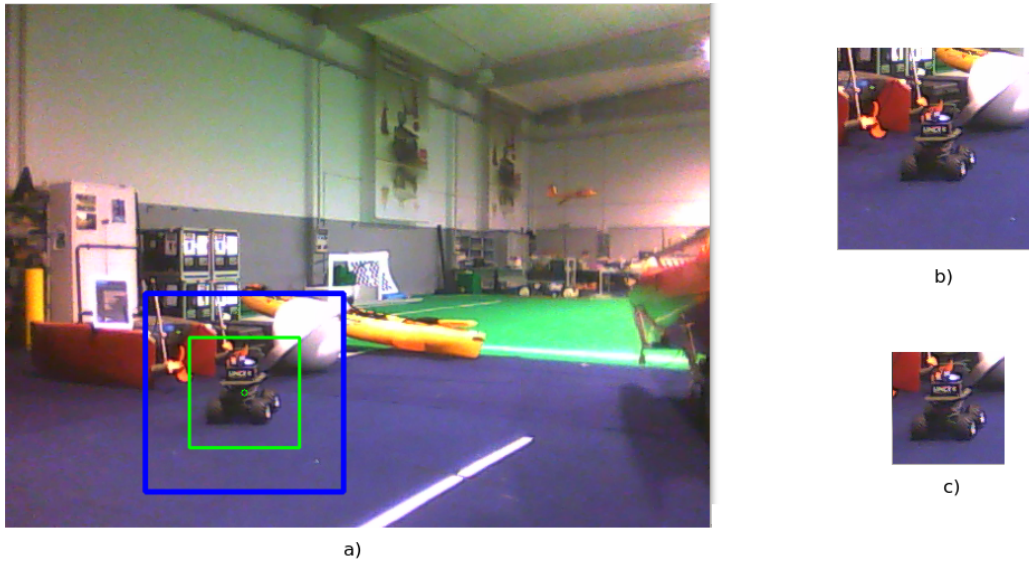


Figura 4.7: *Template Match* - LINCE.

Com a selecção do *Template* da imagem, como já referido acima, o *Template* começa a ser “procurado” na imagem (“a”) (Figura 4.7). Contudo neste exemplo, de modo a otimizar a detecção do mesmo (“c”) é definida uma menor área de procura (“b”), centrada no centro do *Template*.

4.3 Segmentação de imagem por cores

Dado uma imagem, constituída por um determinado formato de cor (ex: RGB, HSV ou HSL), o histograma de cor é obtido através da segmentação de cores da imagem. O histograma é invariante à translação e rotação segundo o eixo óptico. A conversão do conjunto de cores de uma imagem em *bins* é fundamental para a construção do histograma. Representando o espectro da cor como um conjunto pertencente ao intervalo $[0..1]$, considerando a existência de n *bins*, cada *bin* representa $1/n$ elementos de cor.

4.3.1 Histograma

O histograma representa uma equivalência existente entre um conjunto de cores de uma imagem com o conjunto de cores pretendido (*bin*). Idealmente, as cores consideradas idênticas partilham uma região centrada na cor.

A criação do histograma (tendo este o mesmo tamanho que a imagem a cores, mas contendo apenas um componente - *RGB* possui 3 componentes [*R*, *G*, *B*]) é concretizada definindo a segmentação do espaço da cor em *bins*, isto é, no espaço RGB (como exemplo), sendo cada componente de cor representada por um byte, existem 256 valores diferentes para cada componente definindo um total de $256^3 = 16777216$ cores diferentes. Se considerar-se que cada componente é representada, por exemplo, por 16 *bins* ($256/16 = 16$ *bins*), então o histograma é constituído por $16^3 = 4096$ *bins*. Isto quer dizer que cada *bin* é constituído por 4096 cores que se consideram idênticas. Dito isto, um *bin*, corresponde ao intervalo de cores que se pretende detectar na imagem.

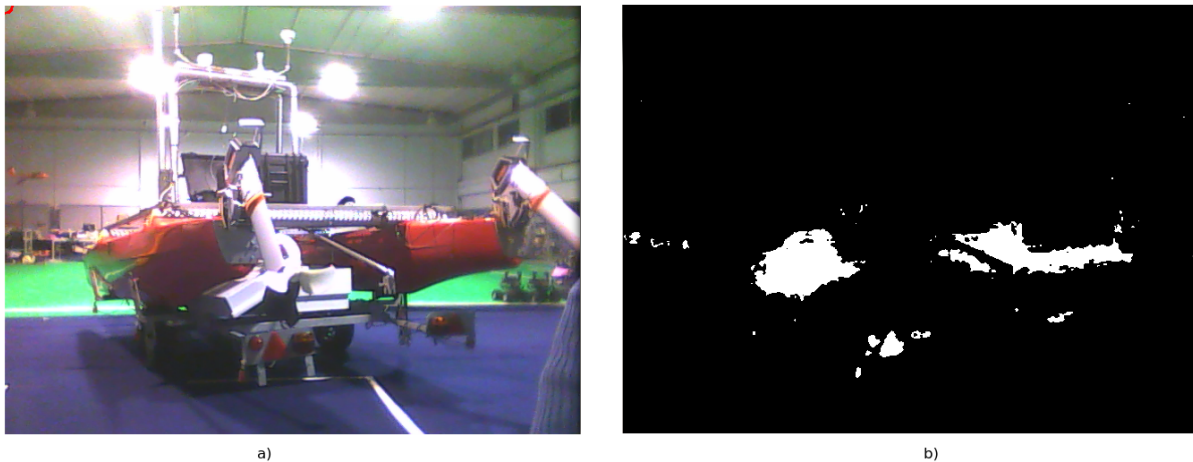


Figura 4.8: Segmentação de cor - detecção da cor vermelha no ROAZ II.

Definindo o *bin* (intervalo de cores) para a cor vermelha, é possível visualizar o resultado obtido pela segmentação (“b”), onde este indica a posição onde, na imagem original (“a”), encontra-se o conjunto de cores pretendido (*bin*). Neste caso a cor pretendida corresponde ao casco do *ROAZ II* (Figura 4.8).

4.3.2 Processo de segmentação por cor

De modo a obter uma imagem segmentada por cor, inicialmente é necessário definir um *bin* (intervalo de cores), que se pretende encontrar na imagem original. Tendo o parâmetro de cores pretendido, é necessário percorrer a imagem original, pixel a pixel, de modo a verificar se as componentes da cor presentes, no respectivo pixel a ser processado, encontram-se dentro do intervalo de cores pretendido (*bin*). Caso pertença ao intervalo de cores, então no histograma (na mesma posição/coordenada (*x*, *y*) do que a imagem original), o pixel do histograma terá valor máximo (branco), caso contrário terá valor mínimo (preto). Este processo tem de ser repetido sempre que a cor (*bin*) pretendida seja outra (Figura 4.9) [41].

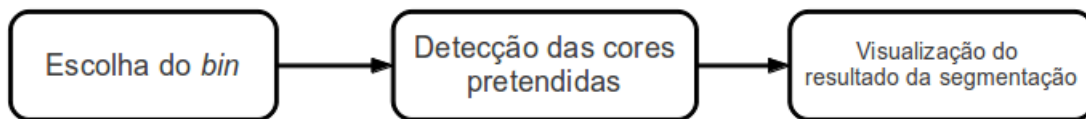


Figura 4.9: Processo de segmentação por cor.

4.4 Transformada de *Hough* - Detecção de linhas

A Transformada de Hough generalizada é uma técnica robusta para detecção de formas geométricas em imagens digitais, sendo esta mais utilizada para o reconhecimento de rectas. Em geral, a transformada é aplicada após a imagem sofrer um pré-processamento, correspondente a detecção dos contornos existentes na imagem.

De modo a descobrir uma linha numa imagem, primeiro temos de passar o sistema de coordenadas cartesiano, para um sistema de coordenadas polar.

$$y = mx + b \Rightarrow y = \left(-\frac{\cos \theta}{\sin \theta}\right)x + \left(\frac{r}{\sin \theta}\right) \quad (4.2)$$

Simplificando a equação 4.2 obtemos:

$$r = x \cos \theta + y \sin \theta \quad (4.3)$$

Onde $r > 0$ e $0 < \theta < 2\pi$.

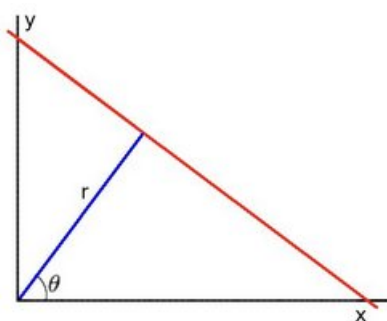


Figura 4.10: Representação do sistema de coordenadas polar.

A transformada de Hough percorre a imagem pixel a pixel associando quais os pixels pertencente à mesma recta. Isto significa se tivermos 5 pontos (x, y) :

- A - (0 , 0.5)
- B - (0.5 , 0.25)

- C - (1 , 1)
- D - (0.25 , 0.4)
- E - (0.75 , 0.1)

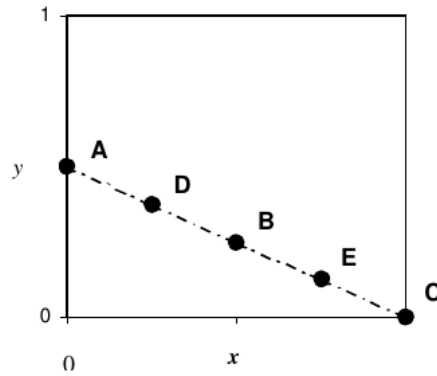


Figura 4.11: Representação de uma recta contendo os pontos A, B, C, D e E.

E recorrendo à equação 4.3, de modo a obter as curvas senoidais de cada um dos pontos, acima referidos, obtém-se:

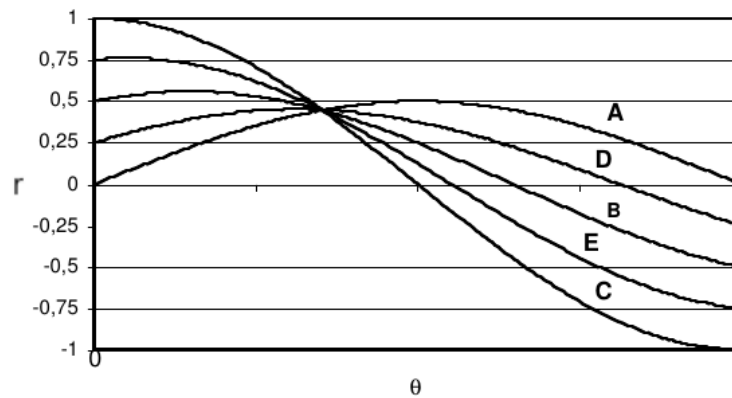


Figura 4.12: Representação das curvas senoidais dos pontos A, B, C, D e E.

Onde podemos visualizar que todas as curvas se cruzam num determinado ponto, sendo esse ponto constituído pelos parâmetros (θ, r) . Como todas as curvas cruzam-se no mesmo ponto, podemos afirmar que todos os pontos (A, B, C, D e E) pertencem à mesma linha, como se pode constatar na Figura 4.11.

(Página intencionalmente deixada em branco)

CAPÍTULO 5

SISTEMAS DE LOCALIZAÇÃO

Neste capítulo, irá ser descrito os sistemas de localização desenvolvidos (Posicionamento relativo a uma infraestrutura[3] e sistema *Groundtruth*), através da sua arquitectura, *software* e implementação do mesmo (*Hardware*). Na implementação do *software*, recorreu-se às bibliotecas de visão computacional *Open Source* do *OpenCV*[1] e *cvBlob*[42].

5.1 Arquitectura do sistema - Posicionamento relativo a uma infraestrutura

A arquitectura do sistema de posicionamento relativo a uma infraestrutura subaquática, é constituída por dois módulos:

- PC - Plataforma responsável pelo processamento dos dados obtidos na imagem capturada.
- Rov (VideoRay PRO 3E) - Módulo de aquisição de imagem.

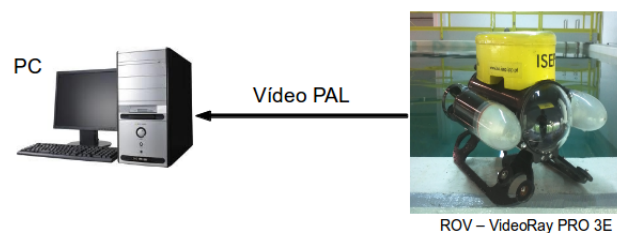


Figura 5.1: Arquitectura do sistema desenvolvido - Posicionamento relativo do ROV a uma infraestrutura.

De modo a converter o sinal de vídeo analógico (PAL) para um sinal digital, foi necessário recorrer a um conversor de sinal (*Frame Grabber*).

5.2 Software - Posicionamento relativo a uma infraestrutura

Como referido no artigo publicado [3], foram desenvolvidos três métodos de visão, de modo a fornecer informação sobre a posição relativa a uma infraestrutura. Sendo esta informação constituída por: x_{12} correspondente ao centro (no eixo do x) da infraestrutura e $(|x_1 - x_2|)$ correspondente à largura da infraestrutura. Com este último parâmetro é possível determinar se o veículo subaquático se encontra a aproximar-se ou a afastar-se da infraestrutura subaquática.

5.2.1 Método I - Detecção de linhas verticais (*Full Frame*)

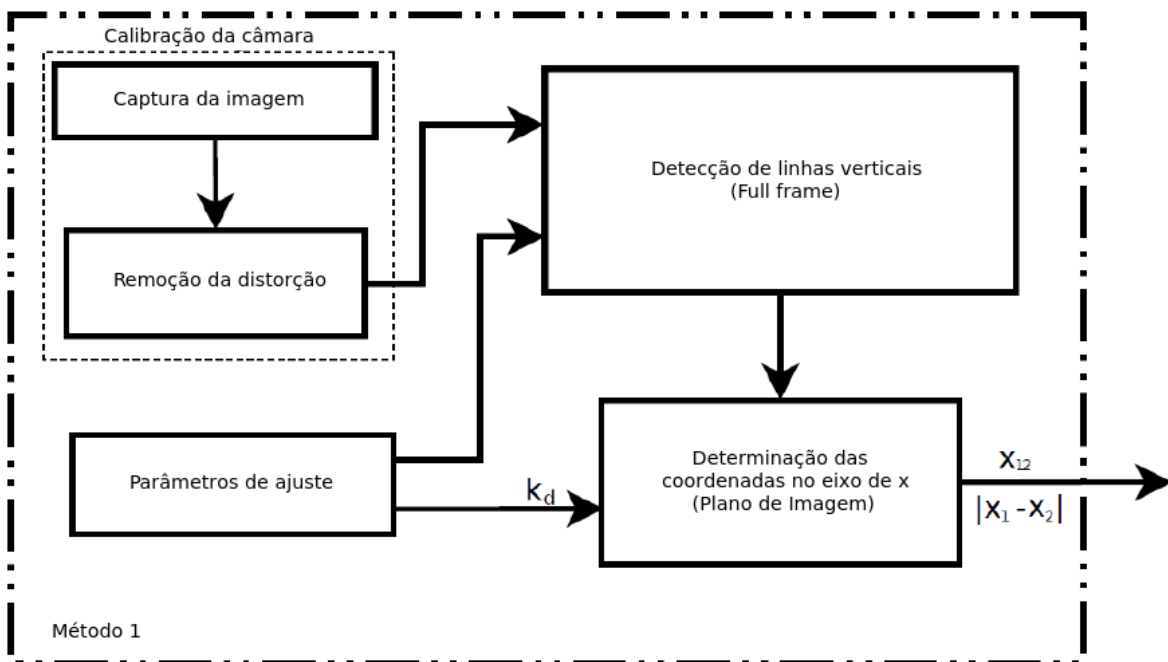


Figura 5.2: Posicionamento relativo do ROV a uma infraestrutura - Método I.

No desenvolvimento, deste primeiro método de detecção, inicialmente foi necessário remover a distorção existente na imagem. Para tal, recorreu-se ao algoritmo de *Tsai*, de modo a determinar os parâmetros de distorção a aplicar à imagem. Após a respectiva “calibração” da imagem, recorreu-se à transformada de *Hough*, de modo a determinar as linhas existentes na imagem, provocadas pelos contornos da infraestrutura subaquática, retornando esta, a posição relativa da infraestrutura no plano da imagem (em pixels).

Conhecendo o valor da distância, dos contornos exteriores da infraestrutura ($|x_1 - x_2|$), é possível agora determinar o centro da infraestrutura a localizar ($x_{12} = x_1 + \frac{|x_1 - x_2|}{2}$).

O parâmetro de ajuste, aplicável a este método, corresponde ao k_d . Este parâmetro tem a função de definir o intervalo máximo aceitável, para que duas linhas, que se encontrem próximas uma da outra, correspondam ao mesmo contorno detectado. Por outras palavras, com a utilização deste parâmetro é feita uma filtragem das linhas que correspondem ao mesmo contorno externo da infraestrutura.

5.2.2 Método II - Detecção de linhas verticais (*Template Match*)

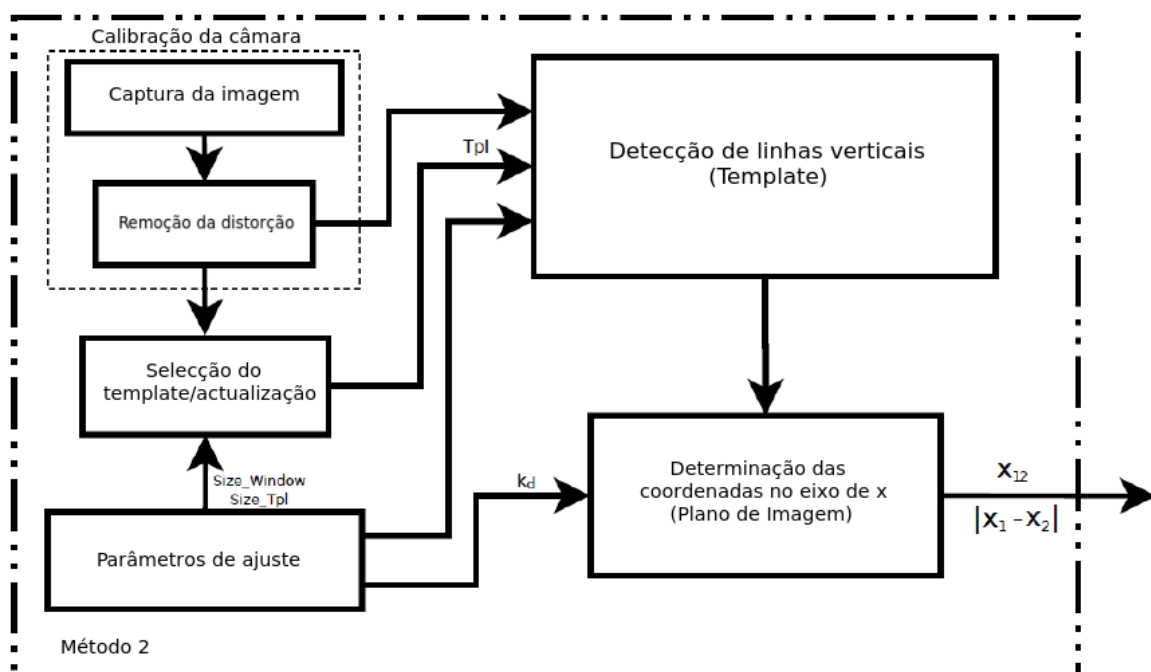


Figura 5.3: Posicionamento relativo do ROV a uma infraestrutura - Método II.

Em relação ao desenvolvimento do segundo método, as características de detecção, deste algoritmo são bastantes idênticas ao do primeiro método. As grandes diferenças consistem na área de procura pelas linhas e os respectivos parâmetros de ajuste adicionados a este método.

A área de pesquisa de linhas, neste método, é só efectuada dentro do *Template*, ao contrário do primeiro método onde a pesquisa é efectuada na totalidade da imagem. Devido a este facto, é inicialmente efectuada uma pesquisa do *Template*, pretendido da infraestrutura, fazendo de seguida uma procura do mesmo (*Template Match*). Após a determinação do *Template* pretendido, o mesmo é utilizado para a detecção dos contornos da infraestrutura. Com a aplicação desta técnica, o custo e tempo computacional diminuem. O *Template* pretendido é definido através da selecção, na imagem, da infraestrutura pretendida. É de salientar

que o *Template* é actualizado automaticamente, de modo a que o algoritmo se adapta-se as características da infraestrutura ao longo da detecção.

Os parâmetros utilizados neste método, para além do já referido no primeiro método (k_d), correspondem aos parâmetros de ajuste da *Search Window* (para o *Template match*) e os de ajuste para o *Template*, este será sempre menor que a *Search Window*. Os valores iniciais correspondem a um quadrado de cento e dez pixels para o *Template* e um quadrado de cento e setenta pixels para a *Search Window*.

5.2.3 Método III - Segmentação por cor (*Template Match*)

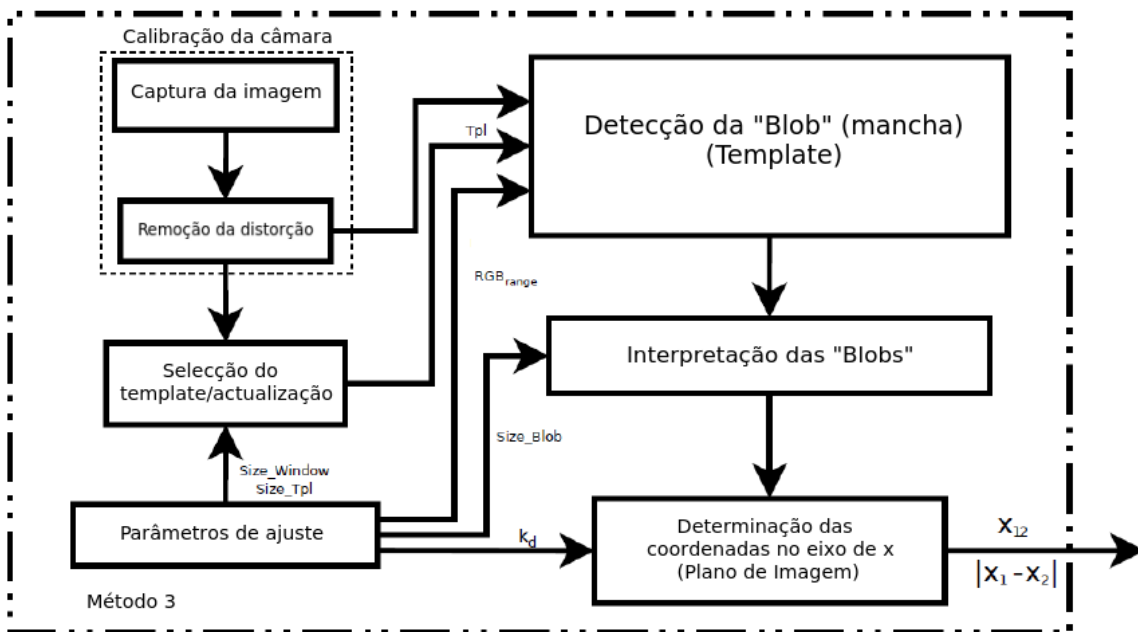


Figura 5.4: Posicionamento relativo do ROV a uma infraestrutura - Método III.

Como no segundo método, inicialmente é necessário definir o *Template* da infraestrutura. Num entanto, embora o segundo método seja semelhante a este método, a detecção da infraestrutura é efectuada através de outra técnica. Neste método, a detecção da infraestrutura não é efectuada através dos contornos da mesma, mas sim através da segmentação de cor do *Template* detectado (cvBlob). Através da segmentação da cor, obtemos uma “mancha”, onde seguidamente determinados o mínimo e máximo valor (no eixo dos x) correspondentes aos pixels das extremidades da “mancha”. Obtendo esses valores é possível determinar os valores de $|x_1 - x_2|$ e x_{12} .

O parâmetro adicionado a este método de localização, corresponde às áreas mínimas e máximas que as “manchas” podem ter, evitando assim falsos positivos.

5.3 Arquitectura do sistema - Visão stereo (*Groundtruh*)

A arquitectura do sistema de localização, é constituída por três módulos:

- PC - Plataforma responsável pelo ordem de captura para as câmaras (*Trigger*) e tratamento da informação recebida de cada uma das câmaras. Optou-se por um PC com determinadas características, de modo a processar vários processos ao mesmo tempo (Intel Core 2 Duo - 2.00GHz).
- *Switch Ethernet* - Ponto de ligação entre o PC e as duas câmaras *Ethernet Gigabit* utilizadas.
- Câmaras *Ethernet* - Módulos de aquisição de imagem.

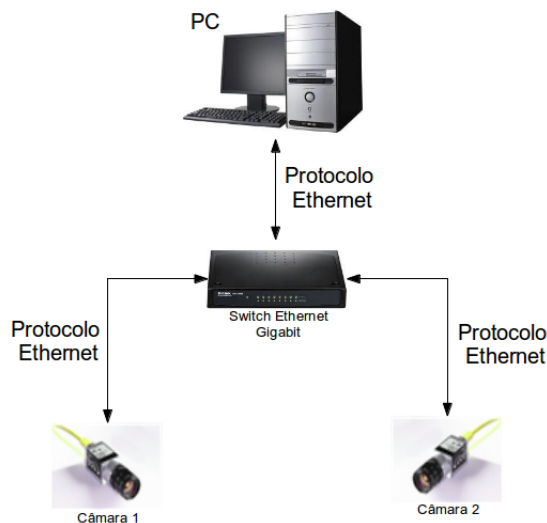


Figura 5.5: Arquitectura do sistema desenvolvido - *Groundtruh*.

O protocolo utilizado para a comunicação entre as câmaras e o PC é o “*Ethernet Gigabit*” (1000 Mbits por segundo). A utilização desta largura de banda deve-se à alta resolução de imagens e ao número de *frames* por segundo que são transmitidos na rede. A organização das conexões deste sistema, encontra-se representada na Figura: 5.5.

5.4 Software - Visão stereo (*Groundtruh*)

Nesta secção será descrito o desenvolvimento do *software* para a calibração das câmaras, captação e processamento de imagens e respectiva rectificação stereo, sendo depois descrito o *software* desenvolvido para *tracking* do veículo.

5.4.1 Calibração das câmaras

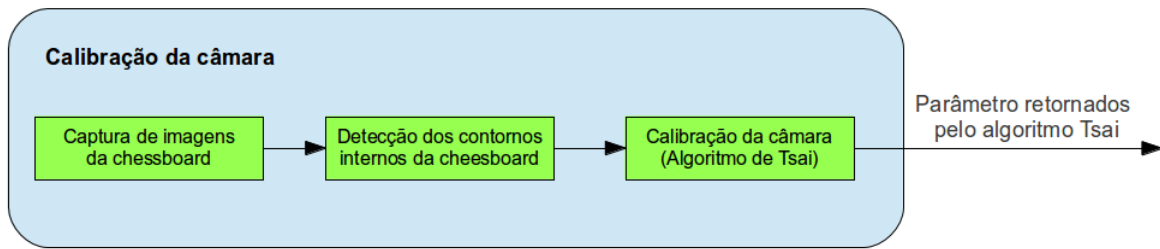


Figura 5.6: Calibração das câmaras - Diagrama de blocos.

No algoritmo, como acima delineado, começou-se por obter 30 imagens de cada uma das câmaras, onde nessas imagens encontra-se visível, em diversas inclinações e posições, ao longo do plano da imagem, uma *cheesboard* de dimensão interna 10 por 7. Esta dimensão corresponde ao número de contornos internos que a *cheesboard* possui. Após a captura das imagens, foi necessário determinar onde se encontram os contornos internos, da *cheesboard*, de modo a poder aplicar o algoritmo de *Tsai*. Resultante da aplicação do algoritmo de *Tsai* obtemos os parâmetros de distorção e os respectivos parâmetros intrínsecos, correspondentes a cada uma das câmaras. Este algoritmo é executado apenas uma vez para cada uma das câmaras.

Obtenção dos parâmetros extrínsecos

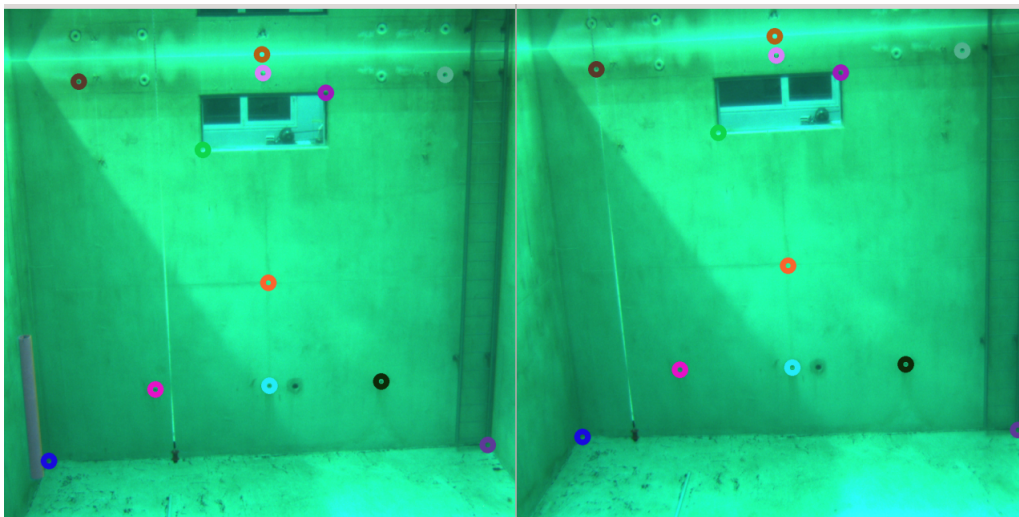


Figura 5.7: Obtenção dos parâmetros extrínsecos.

Para a obtenção dos parâmetros extrínsecos, é necessário obter o valor real (no referencial do mundo) de determinados pontos (círculos coloridos - Figura 5.7). Para tal recorreu-se a pontos conhecidos, de modo a facilitar a obtenção de X , Y e Z (em metros).

5.4.2 Aquisição de imagem e rectificação das mesmas

Na aquisição de imagem e respectiva rectificação das mesmas, foi necessário criar três *threads* (processos que correm em paralelo). Duas para obtenção das imagens de cada uma das câmaras (*thread1* e *thread2*) e uma terceira *thread* para rectificação das imagens e respectiva determinação da posição do veículo (*thread3*).

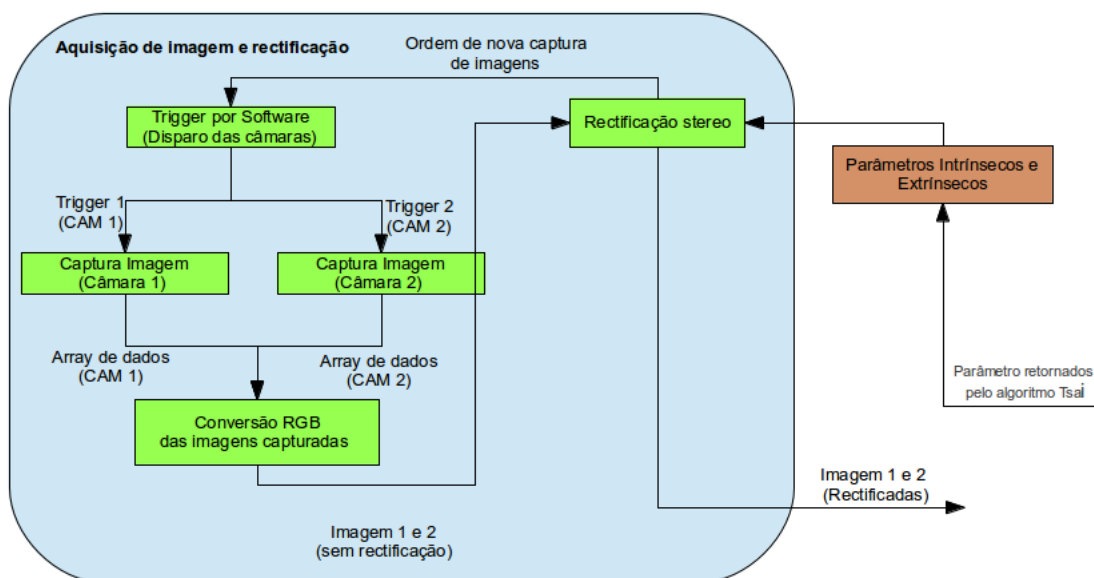


Figura 5.8: Captura das imagens e rectificação das mesmas - Diagrama de blocos.

Correspondentemente às *thread1* e *thread2*, o processo de execução corresponde ao mesmo, ambas as *threads* esperam pelo sinal de disparo (captura - *trigger*) para começarem a adquirir as imagens ao mesmo tempo, já que para determinação da posição de um objecto num sistema stereo, é vital as imagens serem capturadas no mesmo instante de tempo (daqui resulta a necessidade do PC utilizado neste sistema - *multithread*). Após a captura das imagens é necessário verificar, em cada uma das *threads*, se as *arrays* de dados (vectores de imagens), das imagens capturadas possuem informação. Caso um dos vectores de imagem não possua dados, ambos os vectores de imagem são rejeitadas e o processo de captura (*trigger*) recomeça de novo.

Após a correcta captura de ambos os vectores de imagem, das imagens capturadas, é necessário a conversão das mesmas num formato de imagem mais acessível (RGB), de modo a facilitar realização da calibração. É de salientar que devido a esta conversão (*driver* disponibilizado pela *Basler pylon*), o valor médio de *frame rate* alcançado foi de 4 *frames* por segundo para uma resolução de imagem de 960x960 *pixels*.

Com a conversão das *arrays* de imagem para RGB, as imagens resultantes da conversão, correspondentes à câmara da direita e da esquerda, são enviadas para a *thread3*. Nesta *th-*

read é aplicada a rectificação stereo, recorrendo aos parâmetros previamente determinados na calibração das câmaras (algoritmo de *Tsai*) e os parâmetros extrínsecos (caixa castanha).

Completando a rectificação stereo, o algoritmo da *thread3*, torna a dar o sinal de captura de novas imagens (*trigger*) e continua a executar o algoritmo para a respectiva localização do veículo subaquático. Este facto só é possível, pois o tempo computacional para detecção do veículo é muito inferior ao tempo computacional gasto para a conversão das *arrays* para RGB.

5.4.3 Detecção do objecto subaquático pretendido

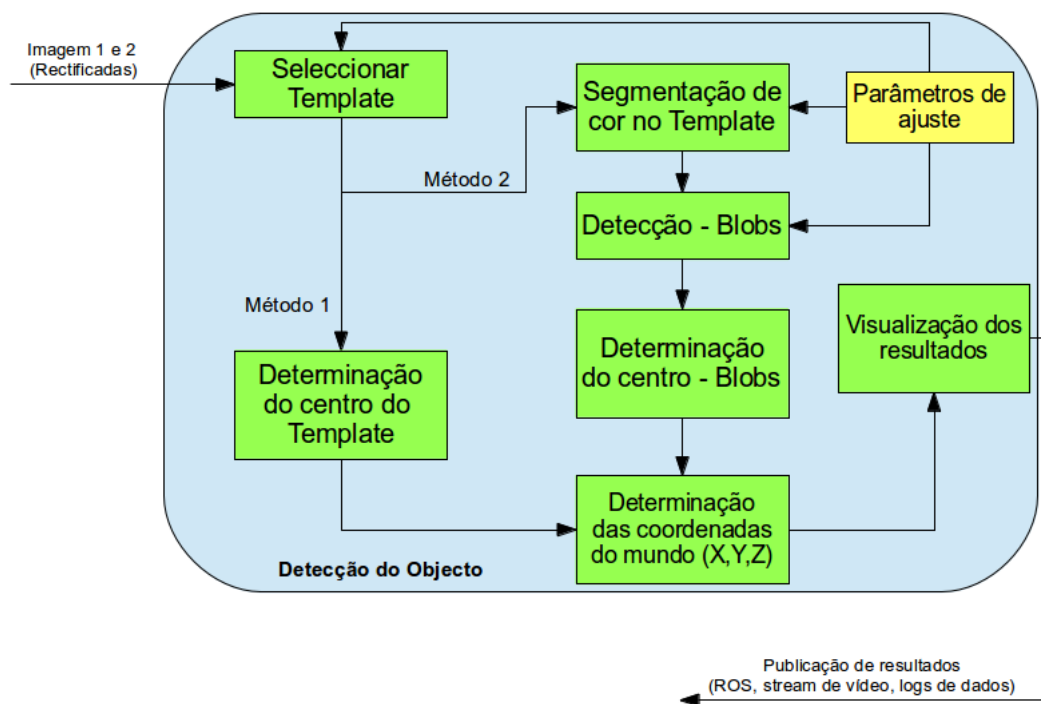


Figura 5.9: Detecção do objecto subaquático - Diagrama de blocos.

Para realizar a detecção do veículo subaquático foram implementados dois métodos. O primeiro método corresponde ao *Template match* e o segundo método corresponde ao *Template match* mais a segmentação de cor, dentro do *Template* detectado. Embora a implementação dos dois métodos seja diferente, de maneira a que os métodos funcionem correctamente, é necessário uma pré-selecção do *Template*, do veículo subaquático a localizar. Caso o *Template* seja perdido, é necessário retornar a redefinir o *Template*. O mesmo é definido seleccionando na imagem, onde está o veículo subaquático a localizar, onde o seu tamanho pode ser alterado durante a execução do algoritmo (Parâmetros de ajuste).

Com a escolha do primeiro método, é determinada a localização do *Template* na imagem, retornando as coordenadas do centro do mesmo. Com as coordenadas (x, y) , do centro do

Template, é determinada a posição real do veículo subaquático, no mundo (X, Y, Z). É de referir, que quer para o método um ou dois, a procura do *Template* é sempre efectuada dentro duma *search window* (Figura 4.7), onde o tamanho desta é sempre superior ao tamanho do *Template*. O tamanho da *search window* pode ser alterado para optimização da detecção do *Template* (Parâmetros de ajuste). Outro factor que pode também ser alterado durante a execução do algoritmo, é o tempo de actualização do *Template*, por outras palavras, quanto tempo é que o algoritmo tem de ficar à espera até poder actualizar o *Template*. Com esta opção o algoritmo torna-se capaz de se auto-reajustar em relação à detecção do *Template*, optimizando assim a detecção do mesmo, já que em situações reais um veículo subaquático irá alterar de forma e tamanho, em relação à perspectiva da câmara.

Optando pelo segundo método, a imagem do *Template* detectado é segmentado, no *bin* (intervalo de cor) definido nos parâmetros de ajuste, podendo estes também serem alterados durante a execução do algoritmo. Após processada a segmentação e retornado as posições, onde se encontra a cor pretendida, é efectuada uma filtragem para distinguir o veículo pretendido e também para remover qualquer ruído (pontos da imagem que não pertencem ao veículo subaquático, que possam surgir, Figura 4.8). Esta filtragem (Detecção - *Blobs*) é efectuada definindo a área mínima e máxima, que o veículo pode ter, (parâmetros estes ajustáveis durante a execução do algoritmo). Após a devida identificação do veículo, no resultado obtido da segmentação de cor, é determinado o centro da área detectado (*cvBlob*), sendo estas coordenadas (x, y) convertidas em coordenadas do mundo (X, Y, Z).

A visualização de resultados pode ser efectuada de diferentes maneiras. Pode ser visualizada em tempo real no PC, criação de um *log* com as coordenadas obtidas da localização do veículo subaquático, gravação de um vídeo do respectivo *tracking* do veículo subaquático com as coordenadas do mundo (X, Y, Z) fundido na imagem, ou por publicação por tópicos (vídeo mais posição real do veículo) para a plataforma ROS. Sendo este último uma mais valia, já que com a plataforma ROS é possível interligar diversos tipos de informação de diversos tipos de sensores, tornando assim este sistema de detecção de veículos subaquáticos uma ferramenta versátil na fusão de dados para detecção de veículos subaquáticos.

5.5 Implementação - Visão stereo (*Groundtruth*)

Para o desenvolvimento do sistema de localização através de visão stereo, foi necessária a aquisição e instalação de hardware. Todo o *hardware* e o respectivo tanque de testes foi disponibilizado pelo LSA.

5.5.1 Câmara

A câmara utilizada, neste projecto, foi uma câmara Basler acA 1300-30gc Figura: 5.10. As características mais importantes, de salientar para o desenvolvimento do sistema pretendido, são respectivamente à resolução máxima da câmara (que neste caso é de 1294 x 964 pixels), protocolo de comunicação (utiliza o protocolo de comunicação *Gigabit Ethernet*, até velocidades de 100Mbps), modo de *trigger* (aceita *trigger* externos, ou gerados por software), número máximo de *frames* por segundo (consegue atingir os 30 *fps*, contudo este valor varia dependendo do driver que esta-se a utilizar para a captura das imagens) e retorna imagens no formato YUYV (devido a esta característica a imagem quando capturada tem de ser convertida de modo a poder ser processada pelo OpenCV). O formato YUYV (YUV 4:2:2) corresponde a que os canais U e V sejam amostrados metade das vezes que o canal Y, fazendo a resolução horizontal ser metade da do YUV 4:4:4 (Y - luminosidade, U e V - componentes cromáticas) [38].



Figura 5.10: Câmaras utilizadas.

Embora nas especificações da câmara o número de *frames* por segundo (*Frame Rate*) seja igual a 30, neste projecto esse valor não foi alcançado devido ao drive disponibilizado pela própria empresa (Basler pylon). Com o drive disponibilizado, o valor de *Frame Rate* é inversamente proporcional com a resolução da imagem (Figura 5.11), isto significa, quanto maior for o *Frame Rate* menor será a resolução da imagem, quando menor o valor de *Frame Rate* maior será a resolução da imagem.

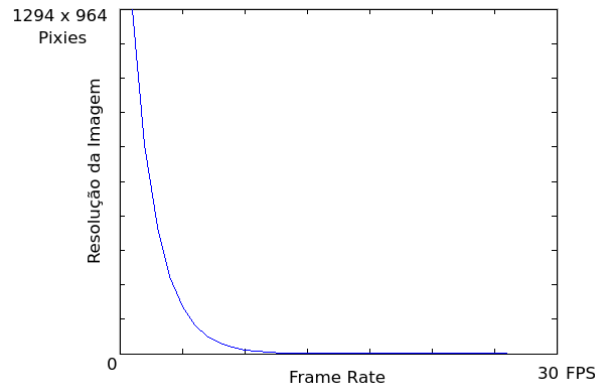


Figura 5.11: Resolução versos fps - drive (Basler pylon).

Para uma obtenção de melhores resultados a nível de localização, optou-se por usar uma resolução de imagem de 960 x 960 pixels (de modo a cobrir a maior área possível do tanque Figura 5.15), onde a resolução máxima da câmara é de 1924 x 964 pixels, o que faz com que o valor de *Frame Rate* varie entre os 4 e os 7 *frames* por segundo. Contudo recentemente surgiu um novo driver, “*Aravis*” [43] que possibilita ter um alto valor de *Frame Rate* sem prejudicar a resolução da imagem. Contudo não foi utilizado já que as informações sobre a API “*Application Programming Interface*” deste novo driver foram disponibilizadas tarde, inviabilizando a sua aplicação neste projecto em tempo útil.

5.5.2 Unidade de protecção da câmara

As unidades de protecção, desenvolvimentos para as câmaras foram fabricados em “*nylon*” resistente à água, com uma espessura de 8 milímetros, comprimento de 18 centímetros e de diâmetro 8 centímetros (Figuras 5.12 e 5.13). Optou-se pela forma cilíndrica de modo a que as unidades resistiam a altas profundidades.



Figura 5.12: Unidade de protecção utilizado para proteger as câmaras.

Para o vidro da unidade de protecção, optou-se por utilizar um vidro plano em vez de um com formato côncavo, de modo a diminuir a distorção radial do mesmo. Este possui uma

espessura de 5 milímetros e de diâmetro igual a 8 centímetros.



Figura 5.13: Unidade de protecção utilizado para proteger as câmaras - perspectiva pelo vidro protector.

Sistema de suporte para as câmaras

Para uma fixação firme das câmaras foi desenvolvido um suporte interno (em aço inox), onde fosse possível fixar as câmaras, de modo a remover o risco de mau posicionamento das mesmas, dentro das unidades protectoras. A escolha do material teve em conta o facto da alta temperatura que as câmaras atingem, transformando assim como um dissipador de calor.



Figura 5.14: Suporte interno para as câmaras

5.5.3 Tanque de testes subaquático

O tanque de testes subaquáticos existente nas instalações do Laboratório de Sistemas Autónomos (LSA), foi utilizado para o desenvolvimento do sistema de localização.

A origem do sistema de coordenadas, utilizado pelo sistema desenvolvido, encontra-se localizada à superfície do tanque de testes do lado oposto à posição das câmaras, onde o eixo de coordenadas Z dá-nos a profundidade do tanque de teste. De modo a que as medidas obtidas, dentro do mesmo, fossem positivas, foi atribuída a orientação, dos eixos 3D, que se encontra representado na Figura 5.15.

As dimensões do respectivo tanque de testes, correspondem aproximadamente a uma largura de 6 metros, comprimento de 11 metros e uma profundidade próxima dos 5 metros.

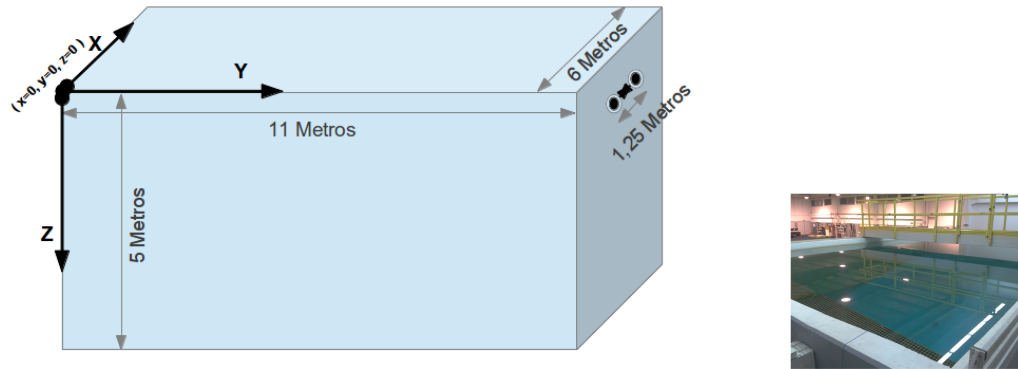


Figura 5.15: Tanque de testes, utilizado para os ensaios experimentais (dimensões).

O sistema de coordenadas foi definido desta forma, de modo a facilitar a interpretação dos resultados obtidos, pelo utilizador.

5.5.4 Posicionamento das câmaras

O posicionamento das câmaras, com uma *Baseline* de 1.25 metros, foi efectuado por forma a que o conjunto de visão, as duas câmaras, ficassem no centro da parede (a nível horizontal) e a uma profundidade de 0.20 metros, Para assim obter a maior área possível do tanque de teste, visível por ambas as câmaras (Figura 5.18).



Figura 5.16: Posicionamento das câmaras - vista subaquática.



Figura 5.17: Posicionamento das câmaras - vista externa.

As unidades de protecção das câmaras, estão fixas a uma barra de perfil de alumínio, que por sua vez foram afixas a uma plataforma, já existente no tanque de testes, a uma altura de 1.30 metros. A distância da plataforma submersa à superfície é de 1.5 metros (Figura 5.17).

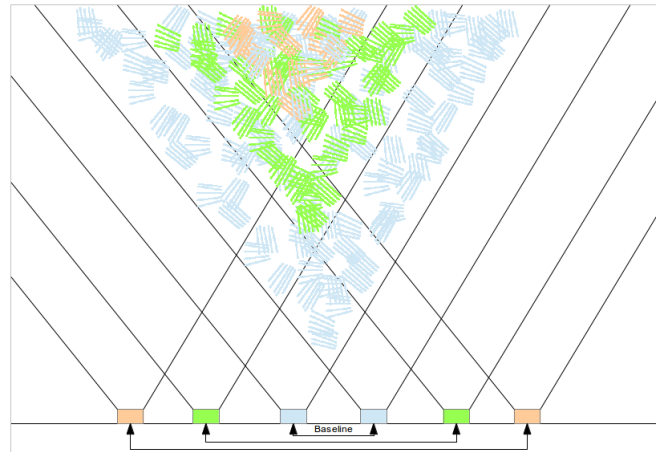
Variação da *Baseline*

Figura 5.18: Variação do tamanho da *Baseline* versus área visível útil.

Durante a fixação das câmaras no tanque de testes, verificou-se que quanto maior fosse a *Baseline*, menor seria a área de cobertura das duas câmaras, por este motivo e pela qualidade das lentes utilizadas nas câmaras (lentes com pouco ângulo de abertura), é que se optou por uma *Baseline* mais pequena, embora o erro comparado com uma *Baseline* maior, seja maior, ou seja, o erro obtido nos resultados, é inversamente proporcional à *Baseline* (Figura 5.18).

Tal como foi mencionado anteriormente, a primeira etapa deste projecto - desenvolvimento de um sistema de posicionamento relativo, baseada em visão monocular, foi já objecto de uma publicação e utilizada em missões de campo ao abrigo dos exercícios navais REP2012 da marinha portuguesa. A segunda etapa consiste num sistema de localização stereo (*Ground-truth*), podendo o mesmo ser utilizado como um sistema externo, para detecção de veículos subaquáticos (em diversos cenários) e também uma mais valia na assistência de atracagem de veículos subaquáticos (ROV) num ASV (ROAZ II).

6.1 Sistema de posicionamento relativo a uma infra-estrutura

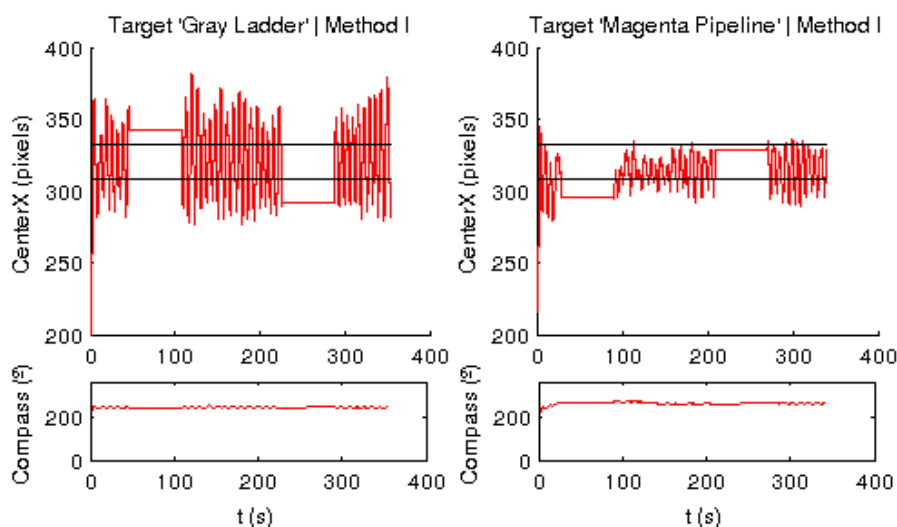


Figura 6.1: Método I (Detecção de linhas verticais – *Full Frame*).

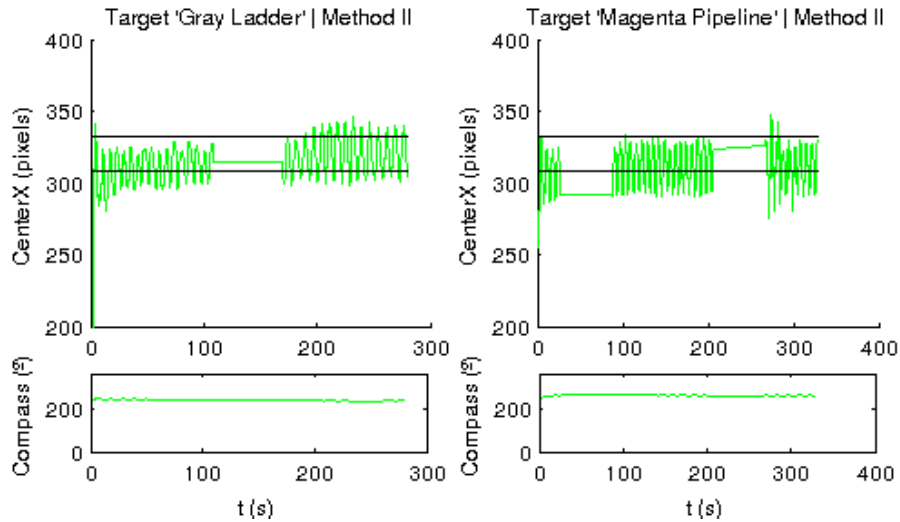


Figura 6.2: Método II (Detecção de linhas verticais – *Template Match*).

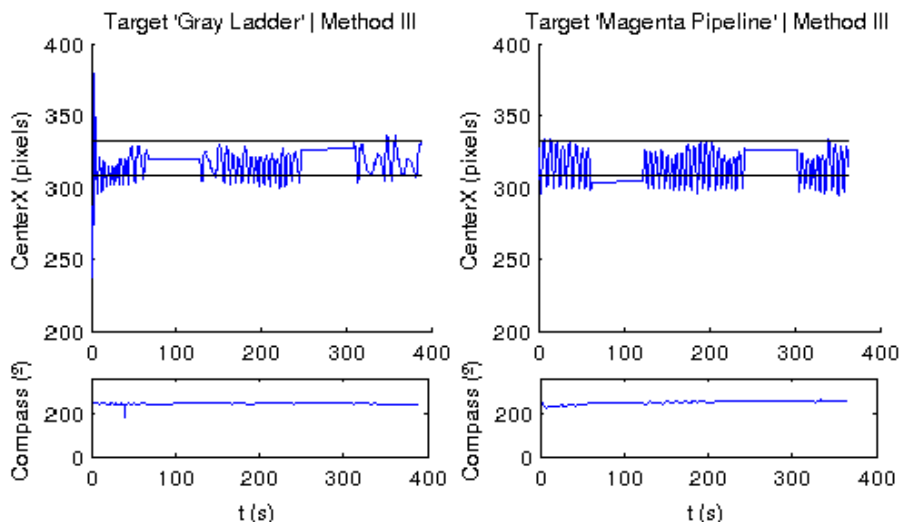


Figura 6.3: Método III (Segmentação por cor – *Template Match*).

Nas figuras acima, estão apresentados alguns dos resultados obtidos com a aplicação dos referidos métodos (I, II e III). Podemos ver que ocorre uma oscilação acentuada (no Método I) na determinação do centro da infraestrutura (x_{12} - *CenterX*), contudo esta oscilação é insignificante para o controlo do veículo subaquático, como podemos ver nos valores da bússola (*Compass*), cujos seus valores são praticamente constantes (em cada um dos métodos implementados).

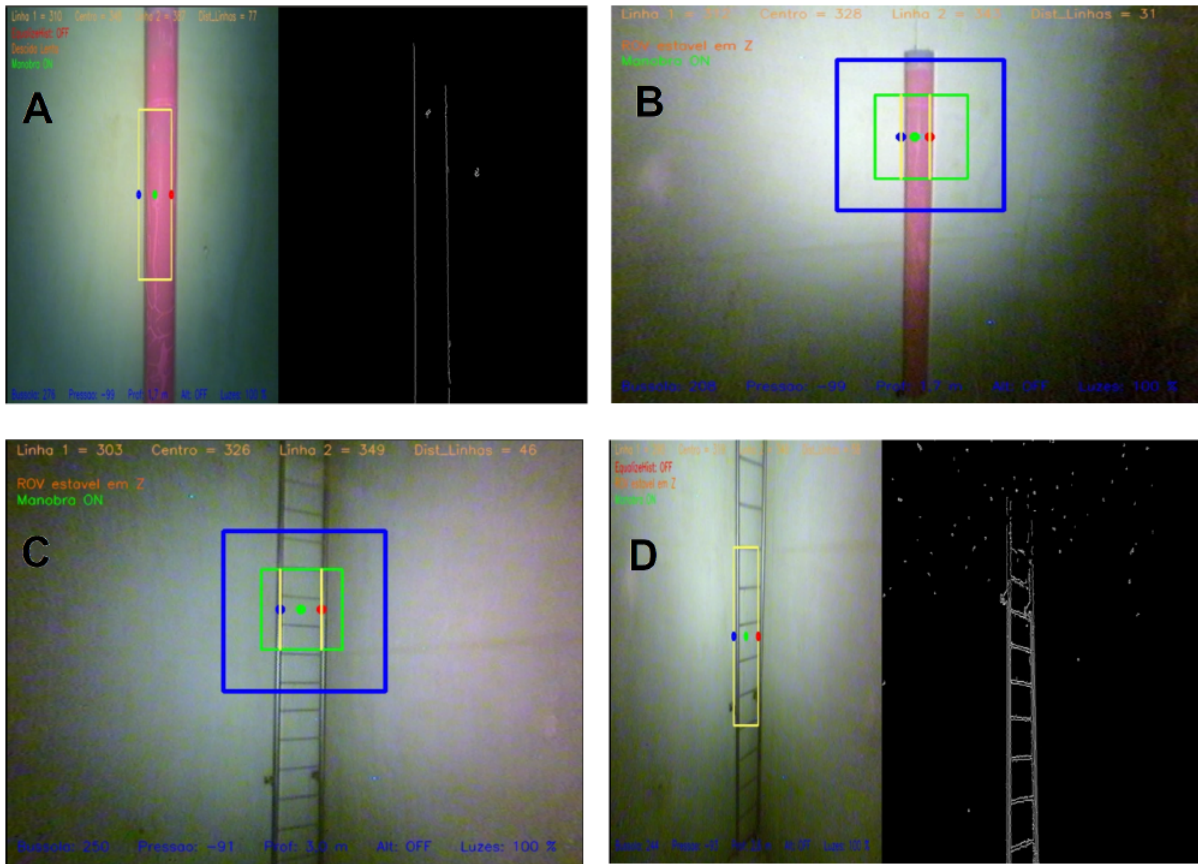


Figura 6.4: Aplicação dos métodos de visão desenvolvidos no artigo publicado[3], A e D corresponde ao método I, B ao método III e C ao método II.

6.2 Sistema de localização de veículos subaquáticos - *Gronudtruh*

Para ambos os métodos de detecção, foram usados os mesmos 22 pontos (medidas reais da localização do veículo), de modo a comparar os diferentes métodos (a nível de erro) e retirar as respectivas conclusões sobre os mesmos.

Tabela 6.1: Localizações reais do veículo subaquático (em metros).

X (metros)	Y (metros)	Z (metros)
5.48	0.00	2.33
3.91	0.00	0.64
0.78	0.00	0.28
1.00	0.00	1.00
2.00	0.00	2.00
2.00	0.00	1.00
5.00	1.00	1.00
3.50	1.00	1.00
4.06	4.22	2.00
2.00	4.22	2.00
3.00	4.22	2.00
3.00	4.22	3.00
3.00	4.22	2.50
1.00	4.50	2.00
2.00	4.50	2.00
4.00	4.50	2.00
3.00	5.00	1.00
3.00	5.00	2.00
2.00	5.23	1.00
4.06	5.23	1.00
3.00	5.23	2.00
3.00	5.23	2.50

6.3 Template Match (Método I)

Na figura abaixo, é possível visualizar as posições, determinadas pelo sistema desenvolvido, do veículo subaquático (em azul) em cada um dos eixos do referencial 3D (X , Y e Z), e as restantes cores correspondem aos valores reais da posição veículo subaquático (vermelho (X), verde (Y) e a preto (Z)).

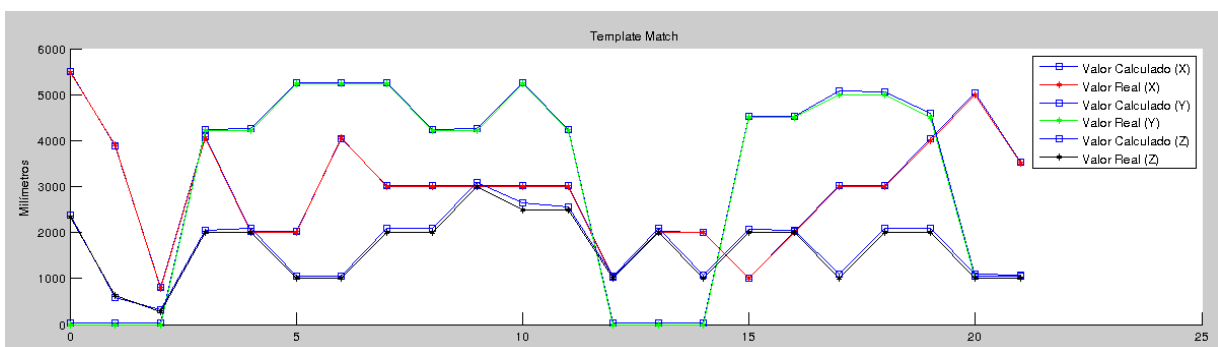


Figura 6.5: Posição real versus determinada - *Template Match*.

Tabela 6.2: Localizações reais versus Localizações determinadas - *Template Match* (em metros).

X (metros)				Y (metros)				Z (metros)			
Real	Determinado	erro	erro(%)	Real	Determinado	erro	erro(%)	Real	Determinado	erro	erro(%)
5.48	5.50	0.02	0.36	0.00	0.04	0.04	-	2.33	2.39	0.06	2.58
3.91	3.88	0.03	0.77	0.00	0.03	0.03	-	0.64	0.58	0.06	9.38
0.78	0.80	0.02	2.56	0.00	0.03	0.03	-	0.28	0.31	0.03	10.7
1.00	1.03	0.03	3.00	0.00	0.04	0.04	-	1.00	1.05	0.05	5.00
2.00	2.02	0.02	1.00	0.00	0.03	0.03	-	2.00	2.08	0.08	4.00
2.00	2.01	0.01	0.50	0.00	0.04	0.04	-	1.00	1.07	0.07	7.00
5.00	5.05	0.05	1.00	1.00	1.06	0.06	6.00	1.00	1.09	0.09	9.00
3.50	3.54	0.04	1.14	1.00	1.05	0.05	5.00	1.00	1.08	0.08	8.00
4.06	4.08	0.02	0.49	4.22	4.24	0.02	0.47	2.00	2.05	0.05	2.50
2.00	2.02	0.02	1.00	4.22	4.27	0.05	1.18	2.00	2.08	0.08	4.00
3.00	3.03	0.03	1.00	4.22	4.24	0.02	0.47	2.00	2.08	0.08	4.00
3.00	3.03	0.03	1.00	4.22	4.25	0.03	0.71	3.00	3.09	0.09	3.00
3.00	3.02	0.02	0.67	4.22	4.23	0.01	0.24	2.50	2.55	0.05	2.00
1.00	1.02	0.02	2.00	4.50	4.52	0.02	0.44	2.00	2.07	0.07	3.50
2.00	2.02	0.02	1.00	4.50	4.52	0.02	0.44	2.00	2.04	0.04	2.00
4.00	4.05	0.05	1.25	4.50	4.59	0.09	2.00	2.00	2.10	0.10	5.00
3.00	3.03	0.03	1.00	5.00	5.08	0.08	1.60	1.00	1.09	0.09	9.00
3.00	3.02	0.02	0.67	5.00	5.07	0.07	1.40	2.00	2.08	0.08	4.00
2.00	2.03	0.03	1.50	5.23	5.26	0.03	0.57	1.00	1.06	0.06	6.00
4.06	4.03	0.03	0.74	5.23	5.25	0.02	0.38	1.00	1.05	0.05	5.00
3.00	3.02	0.02	0.67	5.23	5.26	0.03	0.57	2.00	2.09	0.09	4.50
3.00	3.02	0.02	0.67	5.23	5.26	0.03	0.57	2.50	2.64	0.12	4.80

É possível verificar que o erro obtido entre a posição real, do veículo subaquático, e a posição determinada pelo sistema (aplicando o primeiro método de detecção - *Template Match*), é reduzido, rondando uma média de erro de aproximadamente quatro centímetros.

6.3.1 Erro obtido no Método I

O erro médio, obtido correspondentemente a cada um dos eixos foi de: $X = 2.57$ centímetros com um desvio padrão= 0.93 centímetros, $Y = 3.82$ centímetros com um desvio padrão= 1.97 centímetros e $Z = 7.29$ centímetros com um desvio padrão= 2.32 centímetros.

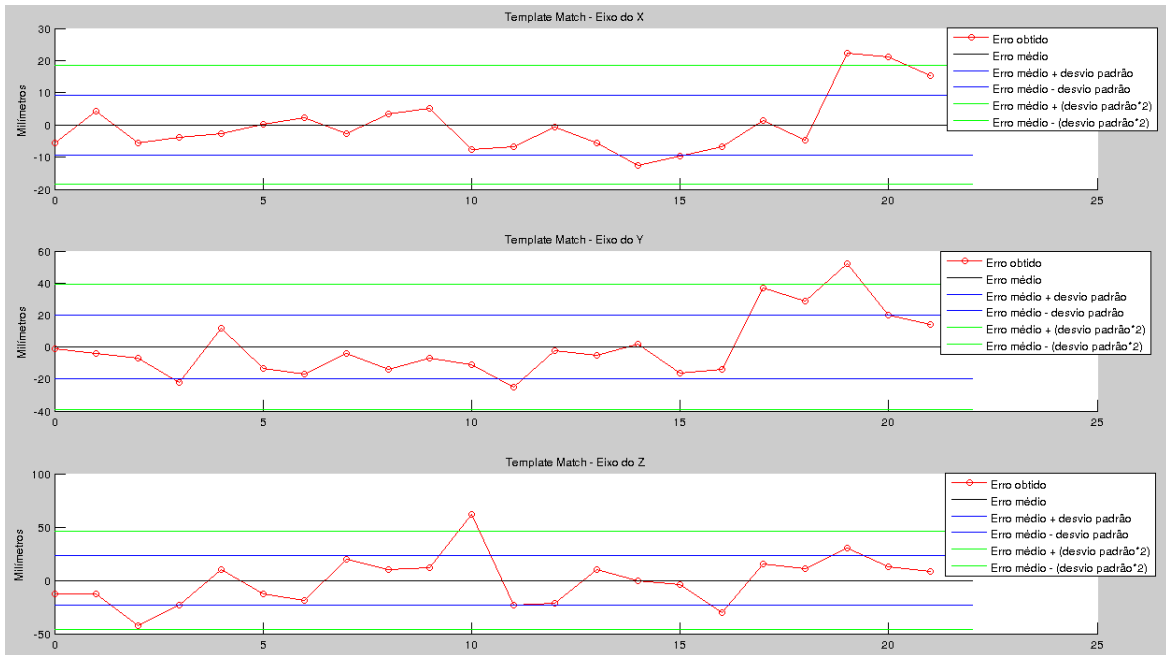


Figura 6.6: Média do erro obtido - *Template Match*.

6.4 Segmentação por cor (Método II)

Tal como mencionado acima, a figura abaixo demonstra o erro obtido, para o segundo método, entre as posições reais e determinadas, do veículo subaquático. Onde a azul temos (X, Y e Z) determinados pelo segundo método, e as restantes cores correspondem aos valores reais da posição veículo subaquático (vermelho (X), verde (Y) e a preto (Z)).

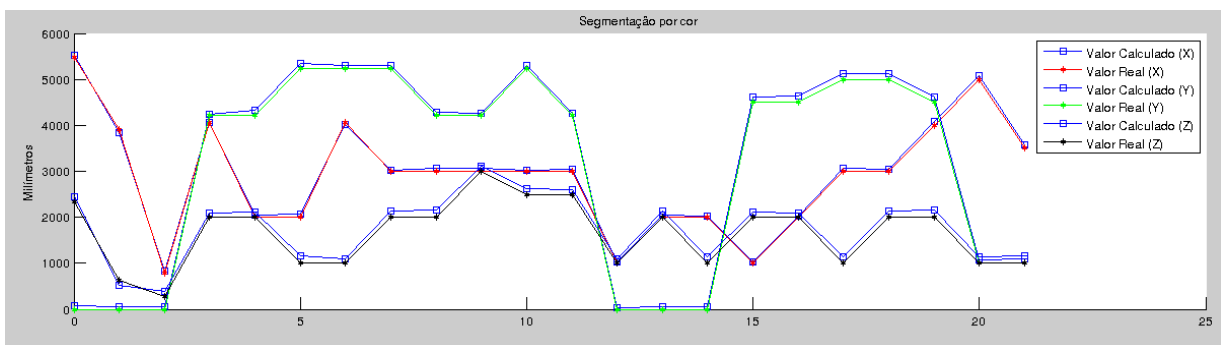


Figura 6.7: Posição real versus determinada - Segmentação por cor.

Tabela 6.3: Localizações reais versus Localizações determinadas - Segmentação de cor (em metros).

X (metros)				Y (metros)				Z (metros)			
Real	Determinado	erro	erro(%)	Real	Determinado	erro	erro(%)	Real	Determinado	erro	erro(%)
5.48	5.53	0.05	0.91	0.00	0.07	0.07	-	2.33	2.45	0.12	5.15
3.91	3.84	0.07	1.79	0.00	0.06	0.06	-	0.64	0.53	0.11	17.2
0.78	0.84	0.06	7.69	0.00	0.05	0.05	-	0.28	0.39	0.11	39.3
1.00	1.02	0.02	2.00	0.00	0.04	0.04	-	1.00	1.10	0.10	10.0
2.00	2.04	0.04	2.00	0.00	0.05	0.05	-	2.00	2.14	0.14	7.00
2.00	2.03	0.03	1.50	0.00	0.05	0.05	-	1.00	1.14	0.14	14.0
5.00	5.08	0.08	1.60	1.00	1.07	0.07	7.00	1.00	1.13	0.13	13.0
3.50	3.58	0.08	2.29	1.00	1.10	0.10	10.0	1.00	1.16	0.16	16.0
4.06	4.06	0.00	0.00	4.22	4.25	0.03	0.71	2.00	2.10	0.10	5.00
2.00	2.05	0.05	2.50	4.22	4.33	0.11	2.61	2.00	2.12	0.12	6.00
3.00	3.06	0.06	2.00	4.22	4.30	0.08	1.90	2.00	2.16	0.16	8.00
3.00	3.07	0.07	2.33	4.22	4.26	0.04	0.95	3.00	3.12	0.12	4.00
3.00	3.04	0.04	1.33	4.22	4.26	0.04	0.95	2.50	2.61	0.11	4.40
1.00	1.03	0.03	3.00	4.50	4.61	0.11	2.44	2.00	2.12	0.12	6.00
2.00	2.02	0.02	1.00	4.50	4.63	0.13	2.89	2.00	2.10	0.10	5.00
4.00	4.08	0.06	1.50	4.50	4.61	0.11	2.44	2.00	2.16	0.16	8.00
3.00	3.06	0.06	2.00	5.00	5.12	0.12	2.40	1.00	1.15	0.15	15.0
3.00	3.05	0.05	1.67	5.00	5.12	0.12	2.40	2.00	2.14	0.14	7.00
2.00	2.06	0.06	3.00	5.23	5.35	0.12	2.29	1.00	1.17	0.17	17.0
4.06	4.02	0.04	0.99	5.23	5.31	0.08	1.53	1.00	1.10	0.10	10.0
3.00	3.03	0.03	1.00	5.23	5.31	0.08	1.53	2.00	2.14	0.14	7.00
3.00	3.03	0.03	1.00	5.23	5.30	0.07	1.34	2.50	2.62	0.12	4.80

No segundo método, o erro obtido entre a posição real (do veículo subaquático) e a posição determinada pelo sistema, em comparação com o primeiro método, os resultados obtidos tiveram uma qualidade inferior. Contudo a diferença entre os erros médios, de ambos os métodos, é de aproximadamente quatro centímetros, já que o erro médio obtido para o segundo método (segmentação por cor) foi de aproximadamente oito centímetros.

6.4.1 Erro obtido no Método II

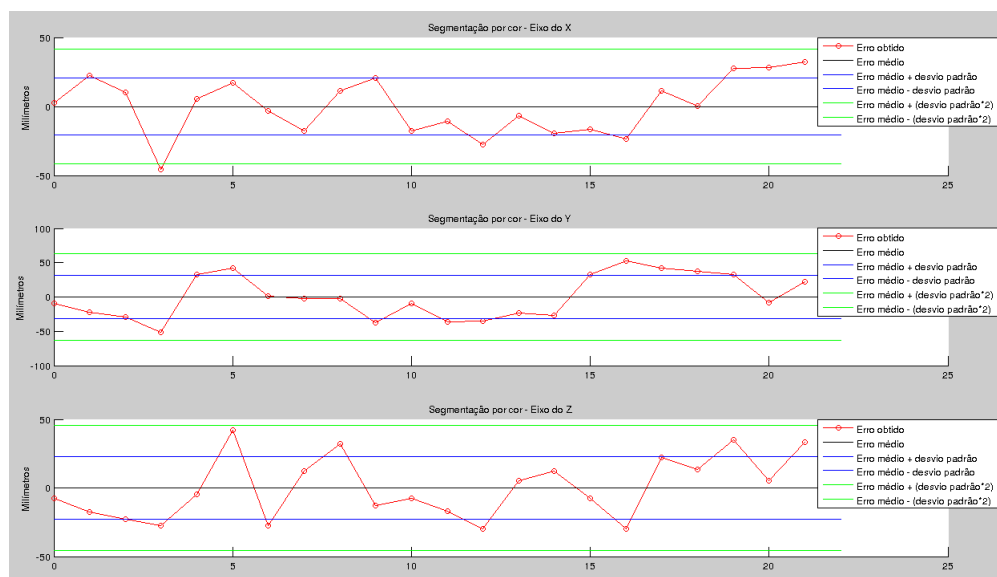


Figura 6.8: Média do erro obtido - Segmentação por cor.

Na aplicação do segundo método, o erro médio obtido para cada um dos eixos foi de: $X = 4.77$ centímetros com um desvio padrão= 2.08 centímetros, $Y = 7.78$ centímetros com um desvio padrão= 3.15 centímetros e $Z = 12.78$ centímetros com um desvio padrão= 2.29 centímetros.

6.5 Qualidade da rectificação stereo

A qualidade da rectificação stereo pode ser avaliada pelo erro médio (*RMS - Root means squars*), das linhas epipolares, por outras palavras, quanto menor for o erro retornado pelo cálculo do *RMS*, melhor será a qualidade da rectificação stereo.

$$\begin{aligned}
 err_{\text{Linhas_epipolares}} = & \\
 & |P_{cam_1}[i].x \times linha1_{epipolar}[i].x + P_{cam_1}[i].y \times linha1_{epipolar}[i].y + linha2_{epipolar}[i].z| + \\
 & + |P_{cam_2}[i].x \times linha2_{epipolar}[i].x + P_{cam_2}[i].y \times linha2_{epipolar}[i].y + linha1_{epipolar}[i].z|
 \end{aligned} \tag{6.1}$$

O cálculo da eq:6.1, é efectuado para “*npoints*” (número de pontos utilizados para obtenção dos parâmetros extrínsecos) de modo a se poder obter o valor de *RMS*. Obtendo o valor de *err* e dividindo este pelo valor de “*npoints*” obtém-se o valor do *RMS* (erro médio das linhas epipolares).

$$RMS = \frac{err}{npoints} \tag{6.2}$$

De modo a determinar as linhas $1_{epipolar}$ e linhas $2_{epipolar}$, é necessário recorrer à eq:6.4 (para as linhas $1_{epipolar}$) ou eq:6.3 (para as linhas $2_{epipolar}$) [1].

$$l_i^{(2)} = F \times p_i^{(1)} \tag{6.3}$$

$$l_i^{(1)} = F^{-t} \times p_i^{(2)} \tag{6.4}$$

Onde F corresponde à matriz fundamental, $l_i^{(1 \text{ ou } 2)}$ corresponde ao valor da linha epipolar que queremos achar (câmara da direita ou esquerda), $p_i^{(1 \text{ ou } 2)}$ corresponde aos pontos da imagem, sendo esta oposta à imagem de onde queremos as linhas epipolar. Por outras palavras, se quisermos determinar as linhas epipolares da câmara da direita, então temos de fornecer os pontos da imagem da esquerda (eq:6.3).

Na figura seguinte temos o resultado obtido na rectificação, para o sistema desenvolvido.

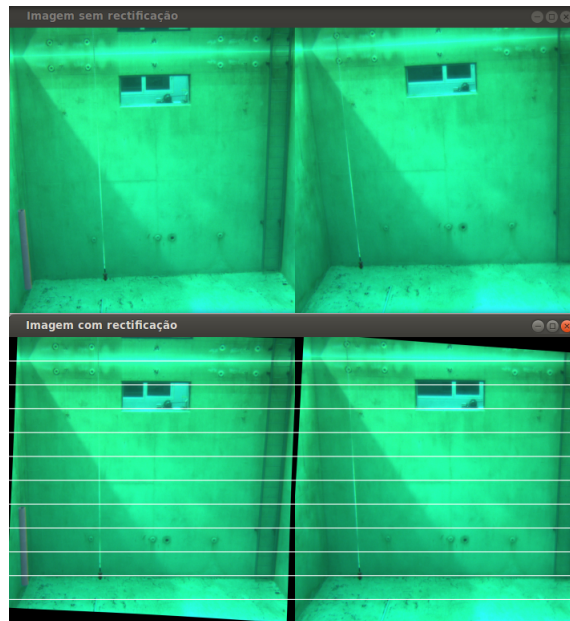


Figura 6.9: Rectificação stereo - linhas epipolares

Aplicando a equação 6.2, ao sistema desenvolvido, o valor resultante da mesma foi igual a:

$$\text{Erro médio do cálculo das linhas epipolares (RMS)} \approx 1.9 \text{ (pixeis)}$$

6.5.1 Média aritmética do erro obtido na rectificação na variação da *Baseline*

Para ser possível caracterizar o erro da rectificação stereo, do sistema desenvolvido, é necessário inicial determinar a *baseline*, correspondente ao sistema implementado. De maneira a obter o valor da *baseline* recorreu-se à equação:

$$\text{Baseline} = \sqrt{T[0]^2 + T[1]^2 + T[2]^2} \quad (6.5)$$

Onde T é o vector de translação, correspondente à translação do referencial da câmara da direita para a câmara da esquerda. O valor da *baseline*, obtido para o sistema desenvolvido foi de:

$$\text{Baseline} = 1.27 \text{ metros}$$

Obtido o valor da *baseline*, é agora possível determinar o erro obtido pela rectificação, através da caracterização das linhas epipolares, onde para tal recorreu-se à equação:

$$\varepsilon = \frac{Z^2}{b \times f} \times \varepsilon_d \quad (6.6)$$

Onde b corresponde à *baseline*, determinada através da equação 6.5, f corresponde à distancia focal das câmaras, Z é a distância do objecto às câmaras e ε_d é o ao erro obtido *RMS* (eq:6.2).

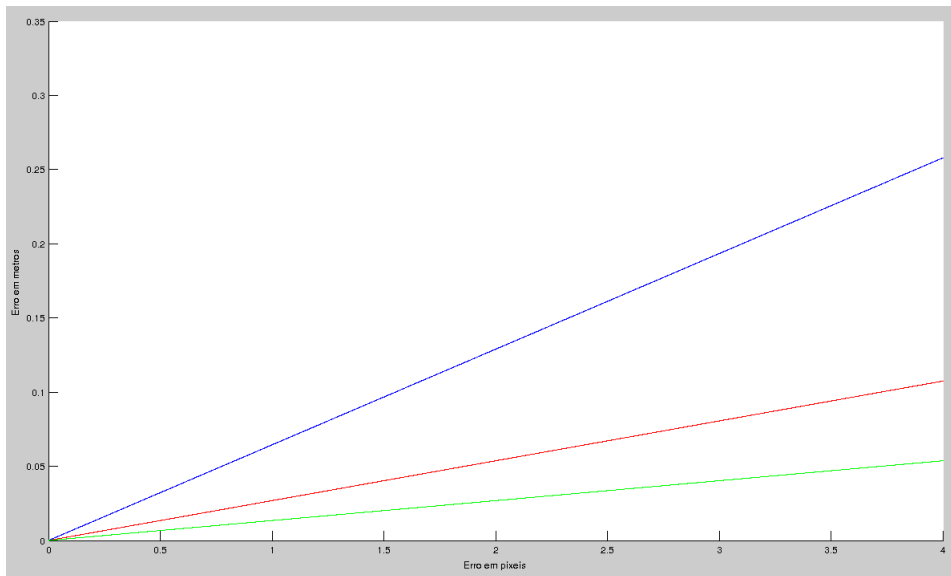


Figura 6.10: Variação do erro com o aumento da *Baseline* (b), [azul $b=0.75$, vermelho $b=3$ e verde $b=6$] e um $Z=11$.

Na figura 6.10, é possível verificar que o erro de localização do objecto, em metros, cresce à medida que a *baseline*, distância entre os centros focais de cada uma das câmaras, aumenta.

Para o sistema desenvolvido o erro médio, para uma *Baseline* = 1.27 metros foi de:

$$\varepsilon = \frac{11^2}{1.269191 \times 1501.5979} \times 1.9$$

$$\varepsilon = 12.69 \text{ centímetros}$$

6.6 USBL - Leituras

Foram efectuados testes comparativos com o sistema de localização USBL (Tritech Micro-Nav), no tanque de testes, onde os resultados obtidos não foram coerentes.

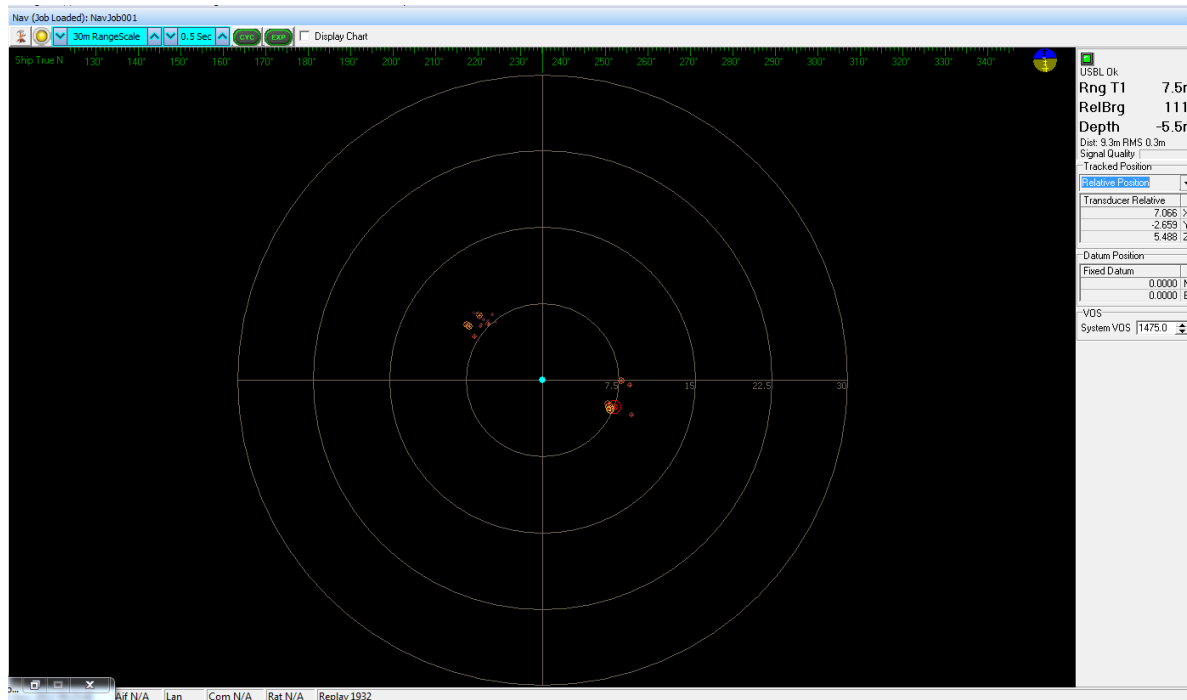


Figura 6.11: Posição do veículo subaquático utilizando o sistema USBL.

Os valores obtidos pelo USBL, não foram os mais correctos, já que o mesmo atingia valores de profundidade abaixo dos 7 metros e distâncias superiores a 15 metros, e também consecutivos saltos entre quadrantes. Fazendo assim com que o sistema USBL não seja indicado para detecção de veículos subaquáticos em cenários com pequena área de execução (inspecção de infraestruturas ou grutas subaquáticas).

6.7 Tracking 3D

É possível visualizar, nos resultados do *tracking 3D* ao veículo subaquático (Figura 6.12), que em certas zonas o posicionamento do veículo é perdido.

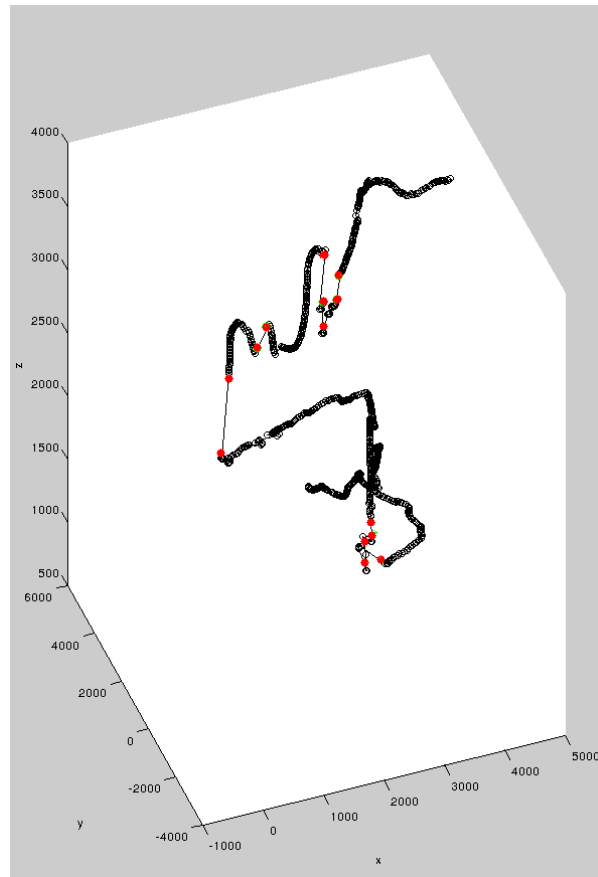


Figura 6.12: Posição do veículo subaquático - (X, Y, Z).

Estas ocorrências (pontos vermelhos) foram provocadas por momentâneas perdas de posição do veículo subaquático ou derivado ao desaparecimento do veículo da imagem de uma das câmaras, impossibilitando assim a sua detecção stereo.

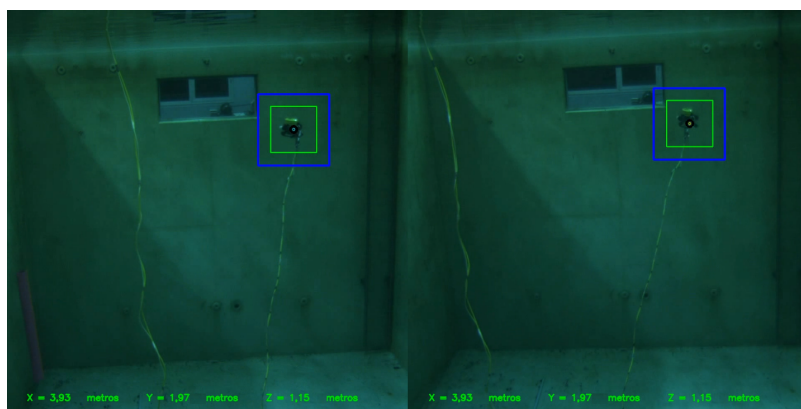


Figura 6.13: Posição do veículo subaquático - sistema desenvolvido.

Nesta dissertação estudamos técnicas de visão para o desenvolvimento de sistemas de navegação para veículos subaquáticos, com dois objectivos bem distintos. O primeiro, a localização relativa para utilizar na Inspeção de infraestruturas subaquáticas e outros veículos. O segundo, a localização de veículos (externos ao sistema), através de um sistema de visão stereo para ser usado como “*groundtruth*” no tanque de testes do LSA.

Tendo identificado e seleccionado as técnicas de visão adequadas para utilizar em ambientes subaquáticos pudemos passar à fase seguinte dos objectivos propostos.

Foi implementado um sistema de posicionamento relativo. Sendo este validado no ROV VideoRay PRO 3E [3], cujos resultados demonstraram qualidade suficiente para realizar o posicionamento semi-automático de um veículo subaquático.

Contudo, embora nos testes efectuados o primeiro método tenha originado um erro ligeiramente superior aos outros métodos implementadas, esse erro não interferiu negativamente no posicionamento semi-automático de um veículo subaquático em relação a uma infra-estrutura, como foi verificado na missão de campo REP2012.

Para o sistema de localização (“*groundtruth*”) de um veículo subaquático, foram implementadas duas técnicas de visão stereo, *Template Match* e segmentação de imagem por cor. As quais foram testadas e validadas experimentalmente no tanque de testes existente no LSA.

A validação do sistema de visão stereo implementado, foi efectuada através de testes realizados no tanque. Estes permitiram caracterizar o sistema implementado. Com os resultados obtidos nos testes efectuados, pode-se afirmar que o sistema de detecção stereo implementado, teve um sucesso significativo, na detecção de veículos subaquáticos, onde o erro médio obtido no método de detecção (*Template Match*) foi de aproximadamente 4 centímetros e no método de segmentação de cor, o erro médio obtido foi aproximadamente de 8 centímetros.

Comparando os erros obtidos, pelos dois métodos de localização, com o erro máximo previsto (média aritmética do erro = 12.69 centímetros), o erro do sistema implementado ficou abaixo do estimado, legitimando assim o sistema como uma solução viável para detecção de veículos subaquáticos, comparativamente ao sistema de detecção USBL, em cenários onde a exactidão é significativamente importante (tanques de teste, grutas subaquáticas e inspecção e manutenção de infraestruturas).

Para trabalho futuro, pretende-se aplicar o sistema de detecção de veículos subaquáticos a um ASV (ROAZ II), de modo a auxiliar a realização de acostagem de veículos subaquáticos ao ASV.

Pretende-se também aplicar o algoritmo desenvolvido à plataforma ROS (“*Robot Operating System*”), de modo a poder ser acessível a outros sistemas (como por exemplo a um DVL ou INS, de modo a fornecer uma solução de navegação ou localização mais robusta do veículo subaquático).

Com a plataforma ROS torna-se possível ainda, a utilização do driver *Aravis*, de modo assim a alcançar *Frame Rates* mais elevados, melhorando assim o tempo de reacção do sistema, na detecção do veículo subaquático.

Para além dos trabalhos futuros mencionados acima, pretende-se ainda a implementação do sistema de posicionamento de visão stereo num veículo subaquático (melhorando assim a precisão da posição relativa deste a uma infraestrutura), implementação do sistema visão stereo (*groundtruth stereo*) em cenários externos (realização de testes de campo), com o objectivo de detectar infraestruturas subaquáticas a curta distância e implementação de um *Kalman Filter* de modo a otimizar a detecção do *Template* (tentar prever onde irá estar o *Template* no próximo frame).

BIBLIOGRAFIA

- [1] OpenCV. (*Open Source Computer Vision Library*), 2012. <http://opencv.org/> ; Último acesso ao link: Outubro de 2012.
- [2] Louis Whitcomb. *Underwater Robotics: Out of the Research Laboratory and Into the Field*, 2000.
- [3] Maria J. Costa, Pedro Gonçalves, Alfredo Martins, and Eduardo Silva. In *Vision-Based Assisted Teleoperation for Inspection Tasks with a Small ROV*. OCEANS 2012 MTS/IEEE HAMPTON ROADS, 2012.
- [4] Nolan S. A High Frequency Wide Field of View Ultrasonic Sensor for Short Range Collision Avoidance Applications on Intervention Class Underwater Vehicles, 2006.
- [5] Dana R. Yoerger, Albert M. Bradley, Hanumant Singh, Barrie B. Walden, M.-H. Cormier, and W. B. F. Ryan. *Multisensor mapping of the deep seafloor with the Autonomous Benthic Explorer*, 2000.
- [6] L Freitag, M Grund, C von Alt, R Stokey, and T Austin. *A Shallow Water Acoustic Network for Mine Countermeasures with Autonomous Underwater Vehicles*, 2005.
- [7] Hayato Kondo and Tamaki Ura. *Navigation of an AUV for investigation of underwater structures*, 2004.
- [8] R. Eustice. *Large-Area Visually Augmented Navigation for Autonomous Underwater Vehicles*. PhD thesis, MIT - Woods Hole Oceanographic Institute, June 2005.
- [9] Jonathan Horgan and Daniel Toal. In *Computer Vision Applications in the Navigation of Unmanned Underwater Vehicles*, December 2008.
- [10] James C. Kinsey, Ryan M. Eustice, and Louis L. Whitcomb. *Underwater vehicle navigation: recent advances and new challenges*, 2006.

- [11] Mikhail Arkhipov. A coordinate determination algorithm for USBL systems. In *Proceedings of the 2nd WSEAS International Conference on Circuits, Systems, Signal and Telecommunications*, CISST'08, pages 50–55, Stevens Point, Wisconsin, USA, 2008. World Scientific and Engineering Academy and Society (WSEAS).
- [12] Miguel Pinto. *Sonar de Varrimento Lateral para Navegação Relativa*. PhD thesis, 2009.
- [13] Laszlo Techy, Kristi A. Morgansen, and Craig A. Woolsey. In *Long-baseline acoustic localization of the Seaglider underwater glider*. American Control Conference, 2011.
- [14] Wen-Hui Cheng. A study of increasing the precision of navigation position for submerged body. *Ocean Engineering*, 31(5–6):693–707, 2004.
- [15] Shojiro Ishibashi, Satoshi Tsukioka, Takao Sawa, Hiroshi Yoshida, Tadahiro Hyakudome, Junichiro Tahara, Taro Aoki, and Akihisa Ishikawa. In *The Rotation Control System to Improve the Accuracy of an Inertial Navigation System Installed in an Autonomous Underwater Vehicle*, pages 495–498, April 2007.
- [16] Dongjun Hyun, Hyun Seok Yang, Hyuk-Sung Park, and Hyo-Jun Kim. Dead-reckoning sensor system and tracking algorithm for 3-D pipeline mapping. *Mechatronics*, 20(2):213–223, 2010.
- [17] Roya Rahbari, Barrie W. Leach, Jeremy Dillon, and Clarence W. de Silva. Expert system for an INS/DGPS integrated navigator installed in a Bell 206 helicopter. *Engineering Applications of Artificial Intelligence*, 18(3):353–361, 2005.
- [18] Shaolin Lü, Ling Xie, and Jiabin Chen. New techniques for initial alignment of strap-down inertial navigation system. *Journal of the Franklin Institute*, 346(10):1021–1037, 2009.
- [19] Yukio Iwaya and Yôiti Suzuki. Rendering moving sound with the doppler effect in sound space. *Applied Acoustics*, 68(8):916–922, 2007.
- [20] Jeff Snyder. Doppler Velocity Log (DVL) Navigation for Doppler Velocity Log (DVL) Navigation for, 2010.
- [21] M. Sahmoudi and Moeness G. Amin. Robust tracking of weak GPS signals in multipath and jamming environments. *Signal Processing*, 89(7):1320–1333, 2009.
- [22] Guillaume Ducard, Raffaello D' Andrea. Autonomous Quadrotor Flight Using a Vision System And Accommodating Frames Misalignment, 2009.

- [23] Zhifeng Liu and Reinhard Klette. Approximated Ground Truth for Stereo and Motion Analysis on Real-World Sequences. In *Proceedings of the 3rd Pacific Rim Symposium on Advances in Image and Video Technology*, PSIVT '09, pages 874–885, Berlin, Heidelberg, 2008. Springer-Verlag.
- [24] Gabriel Fielding and Moshe Kam. Weighted matchings for dense stereo correspondence. *Pattern Recognition*, 33(9):1511–1524, 2000.
- [25] Dr. Gary Bradski and Dr. Adrian Kaehler. *Learning OpenCV*. O'Reilly, 2008.
- [26] D. C. Brown. In *Decentering Distortion of Lenses*, pages 444–462, 1966.
- [27] Fábio Santos Lobão. *Calibração com mapeamento das distorções geométricas aplicada a um sistema de visão stereo*. PhD thesis, August 2004.
- [28] Roger Y. Tsai. Radiometry. chapter A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses, pages 221–244. Jones and Bartlett Publishers, Inc., USA, 1992.
- [29] Carlos Teixeira. *Ensino Rápido de Manipuladores Industriais*. PhD thesis, July 2009.
- [30] Rimon Elias and Robert Laganière. *Projective Geometry for Three-Dimensional Computer Vision*. 2009.
- [31] Gian Luca Mariottini, Stefano Scheggi, Fabio Morbidi, and Domenico Prattichizzo. Planar mirrors for image-based robot localization and 3-D reconstruction. *Mechatronics*, 22(4):398–409, 2012.
- [32] Jianbo Su, Ronald Chung, and Liang Jin. Homography-based partitioning of curved surface for stereo correspondence establishment. *Pattern Recognition Letters*, 28(12):1459–1471, 2007.
- [33] Tsai R. Y. In *A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses*, pages 323–344. IEEE Journal of Robotics and Automation 3 (1987), 2008.
- [34] Z. Zhang. In *Flexible camera calibration by viewing a plane from unknown orientations*, pages 666–673, September 1999.
- [35] Z. Zhang. In *A flexible new technique for camera calibration*, pages 1330 – 1334, 2000.
- [36] R. I. Hartley. In *Theory and practice of projective rectification*, pages 115 – 127, 1998.
- [37] R. I. Hartley and A. Zisserman. In *Multiple View Geometry in Computer Vision*, 2006.

- [38] Pedro Caleiro. *Configuração de sistemas de visão robótica*. PhD thesis, 2008.
- [39] OpenCV. *Métodos de Template Matching*, 2012. http://docs.opencv.org/doc/tutorials/imgproc/histograms/template_matching/template_matching.html; Último acesso ao link: Outubro de 2012.
- [40] J. N. Sarvaiya, Suprava Patnaik, and Salman Bombaywala. Image Registration by Template Matching Using Normalized Cross-Correlation. In *Proceedings of the 2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies*, ACT '09, pages 819–822, Washington, DC, USA, 2009. IEEE Computer Society.
- [41] Brett Browning and Manuela M. Veloso. Real-time, adaptive color-based robot vision. In *IROS*, pages 3871–3876. IEEE, 2005.
- [42] cvBlob. (*Blob library for OpenCV*), 2012. <http://code.google.com/p/cvblob/> ; Último acesso ao link: Outubro de 2012.
- [43] The GNOME Project. *Vision library for ethernet cameras*, 2012. <http://blogs.gnome.org/emmanuel/2010/04/01/aravis-a-vision-library/> ; Último acesso ao link: Outubro de 2012.