

Instituto Superior de Engenharia do Porto



Sistema de Modelização Tridimensional de Galerias Subterrâneas

António João Almeida Bernardo Ferreira

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

Orientador: Eduardo Alexandre Pereira da Silva (Doutor)

Co-orientador: José Miguel Soares de Almeida (Mestre)

Julho de 2011

Resumo

A presente dissertação apresenta uma solução para o problema de modelização tridimensional de galerias subterrâneas. O trabalho desenvolvido emprega técnicas provenientes da área da robótica móvel para obtenção um sistema autónomo móvel de modelização, capaz de operar em ambientes não estruturados sem acesso a sistemas de posicionamento global, designadamente GPS. Um sistema de modelização móvel e autónomo pode ser bastante vantajoso, pois constitui um método rápido e simples de monitorização das estruturas e criação de representações virtuais das galerias com um elevado nível de detalhe.

O sistema de modelização desloca-se no interior dos túneis para recolher informações sensoriais sobre a geometria da estrutura. A tarefa de organização destes dados com vista à construção de um modelo coerente, exige um conhecimento exacto do percurso praticado pelo sistema, logo o problema de localização da plataforma sensorial tem que ser resolvido.

A formulação de um sistema de localização autónoma tem que superar obstáculos que se manifestam vincadamente nos ambientes *underground*, tais como a monotonia estrutural e a já referida ausência de sistemas de posicionamento global.

Neste contexto, foi abordado o conceito de SLAM (*Simultaneous Localization and Mapping*) para determinação da localização da plataforma sensorial em seis graus de liberdade. Seguindo a abordagem tradicional, o núcleo do algoritmo de SLAM consiste no filtro de Kalman estendido (EKF – *Extended Kalman Filter*). O sistema proposto incorpora métodos avançados do estado da arte, designadamente a parametrização em profundidade inversa (*Inverse Depth Parametrization*) e o método de rejeição de *outliers* 1-Point RANSAC. A contribuição mais importante do método por nós proposto para o avanço do estado da arte foi a fusão da informação visual com a informação inercial.

O algoritmo de localização foi testado com base em dados reais, adquiridos no interior de um túnel rodoviário. Os resultados obtidos permitem concluir que, ao fundir medidas inerciais com informações visuais, conseguimos evitar o fenómeno de degeneração do factor

de escala, comum nas aplicações de localização através de sistemas puramente monoculares. Provámos simultaneamente que a correcção de um sistema de localização inercial através da consideração de informações visuais é eficaz, pois permite suprimir os desvios de trajectória que caracterizam os sistemas de *dead reckoning*.

O algoritmo de modelização, com base na localização estimada, organiza no espaço tridimensional os dados geométricos adquiridos, resultando deste processo um modelo em nuvem de pontos, que posteriormente é convertido numa malha triangular, atingindo-se assim uma representação mais realista do cenário original.

Abstract

This thesis presents a solution to the problem of underground galleries' modelling. The developed work employs techniques from the mobile robotics field to produce an autonomous mobile modelling system, able to operate in non-structured environments in which global positioning systems, such as GPS, are not available. Autonomous mobile modelling systems can become very useful to produce highly detailed virtual representations of the galleries, as well as for monitoring purposes.

The proposed modelling system moves inside tunnels collecting geometric sensor data. Organizing the data, in order to build a consistent model, requires precise knowledge about the system's trajectory inside the gallery, therefore the sensors' localization must be determined.

Formulating an autonomous localization system demands overcoming two main obstacles imposed by underground environments: structural monotony and the lack of global positioning systems.

In this context, we adopted the SLAM (Simultaneous Localization and Mapping) concept to determine the platform's localization in six degrees of freedom. Following the traditional approach, we used the EKF (Extended Kalman Filter) as the core algorithm to solve the SLAM problem. The system combines advanced state-of-the-art methods such as Inverse Depth Parametrization, and the 1-Point RANSAC algorithm for outlier rejection. Our most important contribution to the state-of-the-art consists on the fusion between visual and inertial measurements.

The localization algorithm has been tested with real data acquired on a road tunnel. The results show that the scale factor degeneration, a common phenomenon associated with monocular localization applications, can be avoided by combining visual and inertial measurements. We also proved that the introduction of visual measurements is an effective strategy to correct the trajectory drift that characterizes inertial localization applications.

The modelling algorithm organizes the geometric data in the three-dimensional space based on the estimated localization. This process results in a point cloud model, that is then converted into a triangular mesh, reaching a more realistic representation of the original scene.

Agradecimentos

Aproveito para expressar os meus sinceros agradecimentos a todas as pessoas que me proporcionaram as condições necessárias à realização deste trabalho. Deste grupo destaco os membros da minha família e amigos próximos, que sempre me apoiaram.

Manifesto a minha gratidão pela ajuda e paciência dispensada pelos membros do Laboratório de Sistemas Autónomos. Dirijo ainda um reconhecimento especial aos meus orientadores, pelo projecto extremamente aliciante que me reservaram e pelo acompanhamento atento durante todo o trabalho.

António João Almeida Bernardo Ferreira

Esta página foi intencionalmente deixada em branco.

“To copy others is necessary, but to copy oneself is pathetic.”

Pablo Picasso

Esta página foi intencionalmente deixada em branco.

Conteúdo

1	Introdução	1
1.1	Âmbito da Dissertação	1
1.2	Objectivos	3
1.3	Estrutura da Dissertação	4
2	Estado da Arte	7
2.1	Sensores e Técnicas de Mapeamento Associadas	7
2.2	Análise a Diversos Sistemas de Mapeamento	11
2.2.1	2x2D SLAM – Mapeamento de Minas	11
2.2.2	Mapeamento de Minas com LRF 3D – <i>Groundhog</i>	13
2.2.3	Mapeamento <i>Indoor</i> Através de Câmaras e LRF 3D	16
2.2.4	Mapeamento com LRF 3D Estático	18
2.3	Conclusão e Contextualização com a Problemática de Mapeamento e Localização em Túneis	20
3	Registo de Varrimentos de Sensores LRF	23
3.1	<i>Scan Matching</i> Denso (<i>point-to-point Scan Matching</i>)	24
3.1.1	<i>Iterative Closest Point</i> (ICP)	24
3.1.1.1	Redução da Complexidade Computacional	26
3.1.1.2	Convergência para o Mínimo Global	34
3.1.1.3	Eliminação de <i>Outliers</i>	35
3.1.2	<i>Iterative Dual Correspondence</i> (IDC)	35
3.1.3	<i>Metric-based ICP</i> (MbICP)	37
3.2	<i>Feature-based Scan Matching</i> (FBSM)	39
3.3	Algoritmos que não Estabelecem Correspondências	40
4	Fundamentos de Visão Computacional	41
4.1	Extracção de <i>Features</i>	42
4.1.1	Detector de Harris	43
4.1.2	<i>Scale-Invariant Feature Transform</i> (SIFT)	47
4.1.2.1	Detecção de Extremos ao longo do Espaço de Escala	48
4.1.2.2	Obtenção da Localização dos Pontos	51
4.1.2.3	Determinação da Orientação	53
4.1.2.4	Construção do Descritor	54
4.2	Correspondência entre Imagens	55
4.2.1	Comparação de Descritores do Tipo SIFT	55
4.2.2	<i>Template Matching</i>	55
4.3	Projecção Perspectiva	57

4.4	Distorção Provocada pela Lente	60
4.5	Visão <i>Stereo</i>	62
4.5.1	Geometria Epipolar	63
4.5.2	Matriz de Homografia	65
4.5.3	Rectificação <i>Stereo</i>	66
4.6	RANSAC	68
4.6.1	1- <i>Point</i> RANSAC	70
4.7	<i>Bundle Adjustment</i>	72
5	Localização	75
5.1	Visão Geral Sobre Métodos de Localização <i>Indoor</i>	76
5.2	Navegação Inercial	78
5.2.1	Mecanização segundo o referencial de nível local	80
5.2.2	Erros típicos das unidades de medição inercial	82
5.2.2.1	Erros Característicos dos Giroscópios	83
5.2.2.2	Erros Típicos dos Acelerómetros	84
5.3	SLAM - <i>Simultaneous Localization and Mapping</i>	84
5.3.1	SLAM Baseado em Filtro de Kalman Estendido	84
5.3.1.1	Formulação do Problema de SLAM Segundo o EKF	86
5.3.1.2	Consistência dos Pontos Observados	89
5.3.1.3	Requisitos Computacionais	90
5.3.2	SLAM Baseado em Filtro de Partículas	90
5.3.3	SLAM por Registo de Varrimentos Laser	92
5.3.4	Visual SLAM	93
5.3.4.1	<i>Visual SLAM</i> Monocular – Parametrização <i>Inverse Depth</i>	93
5.3.4.2	<i>Visual SLAM Stereo</i>	101
5.4	Sistemas Multirobóticos de Localização Cooperativa	102
6	Representação de Modelos 3D	105
6.1	Nuvens de Pontos	105
6.2	Malhas Poligonais	106
7	Aquisição de dados	109
7.1	Plataforma Sensorial	110
7.1.1	IMU	110
7.1.2	Inclinómetro	111
7.1.3	Sensores LRF	112
7.1.4	Câmaras CCD	113
7.1.5	Disposição dos Sensores	113
7.1.6	Transformações entre os Sistemas de Coordenadas dos Sensores	115
7.1.7	Registo do Instante das Medidas	115
7.2	Aquisição do <i>Dataset</i>	116
8	Implementação e Resultados	119
8.1	Projecto do Sistema de Localização	120
8.1.1	Pré-selecção de <i>Features</i>	120
8.1.2	Algoritmo de SLAM	123
8.1.2.1	Inicializações	125
8.1.2.2	Fase de Previsão	127

8.1.2.3	Inicialização de <i>Landmarks</i>	128
8.1.2.4	Gestão do Mapa	129
8.1.2.5	Previsão de Observações	130
8.1.2.6	1- <i>Point</i> RANSAC	131
8.1.2.7	Fase de <i>Update</i> EKF	132
8.1.2.8	Recuperação de <i>high-innovation inliers</i>	133
8.2	Resultados do Sistema de Localização	134
8.3	Construção do Modelo	145
9	Conclusões e Trabalho Futuro	153
9.1	Conclusões	153
9.2	Trabalho Futuro	154
	Referências	166
A	Cálculo dos jacobianos	167
A.1	Jacobianos do Modelo de Observação em Ordem aos Estados do Filtro	167
A.1.1	Jacobianos do Modelo de Observação em Ordem aos Estados da Localização do Corpo	167
A.1.2	Jacobianos do modelo de observação em ordem aos estados das <i>land-</i> <i>marks</i>	170
A.2	Jacobiano Utilizado na Conversão <i>Inverse-Depth</i> /Cartesianas	171
A.3	Jacobiano para Cálculo da Covariância Inicial de uma <i>Landmark</i>	172
A.4	Jacobianos da Mecanização	175
A.4.1	Cálculo da matriz F	176
A.4.2	Cálculo da matriz G	177
B	Representações de orientação	179
B.1	Ângulos de Euler	179
B.2	Quaterniões	180
B.2.1	Operações sobre quaterniões	181
B.2.2	Representação de quaterniões sob a forma de vector	182
B.2.3	Rotações através de quaterniões	183
B.2.4	Conversão de quaterniões noutras expressões de rotação	184
B.3	Matrizes de rotação	185
C	Desvio Padrão da Trajectória de Referência	187

Esta página foi intencionalmente deixada em branco.

Lista de Figuras

2.1	Núvem de pontos registada por um sensor LRF, onde é evidente a perda de resolução originada pela diminuição do ângulo de incidência.	9
2.2	Exemplo de um sistema de aquisição de varrimentos tridimensionais baseado na variação do ângulo vertical de incidência de um sensor LRF 2D.	9
2.3	Disposição dos quatro sensores LRF.	11
2.4	Representação poligonal.	13
2.5	Modelo em formato VRML.	13
2.6	Veículo de operação remota <i>groundhog</i>	14
2.7	Representação da nuvem de pontos originada por um varrimento 3D.	16
2.8	Vista elevada de uma galeria virtualmente reconstruída pelo registo de múltiplos varrimento 3D.	16
2.9	Veículo de mapeamento.	17
2.10	Estrutura de calibração.	17
2.11	Modelo geométrico em malha triangular.	18
2.12	Modelo final que incorpora dados geométricos e visuais.	18
2.13	Vista exterior da Catedral de Beauvais.	19
2.14	Modelo exterior da Catedral de Beauvais.	19
3.1	Divisão por hiperplanos de um conjunto de pontos 2D e <i>kd-tree</i> correspondente.	27
3.2	Regra da restrição esférica.	30
3.3	Orientação das transformações ao longo de três iterações.	32
3.4	Representação da aproximação linear e da interpolação parabólica.	33
3.5	Correspondências respeitando a <i>closest-point rule</i>	36
3.6	<i>Matching-range-point rule</i>	37
4.1	Deslocação da janela no método de Harris.	43
4.2	Aspecto das derivadas da imagem segundo x e y e resultado da sua multiplicação.	44
4.3	Elipse de covariância.	45
4.4	Relação entre a forma da elipse e a <i>feature</i> em causa.	45
4.5	Divisão do espaço dos valores próprios para diferenciação entre pontos de interesse e fronteiras.	46
4.6	Efeito da variação de escala da imagem na classificação da <i>feature</i>	47
4.7	Exemplo de um espaço de escala.	49
4.8	Representação de quatro oitavas e espaços de escala correspondentes.	49
4.9	Cálculo da diferença de gaussianas.	50
4.10	Imagens originadas por diferença de gaussianas ao longo de quatro oitavas.	51

4.11	Pontos considerados na pesquisa de extremos.	51
4.12	Modelo <i>pinhole</i>	57
4.13	Modelo <i>pinhole</i> com plano da imagem em frente do centro óptico	58
4.14	Projectão de rectas do mundo na imagem dependendo do tipo de distorção causada pela lente.	61
4.15	Geometria epipolar <i>stereo</i>	63
4.16	Rectificação de um par de imagens <i>stereo</i> e representação das linhas epipolares. 67	
4.17	Geometria após rectificação – epipolos no infinito e planos da imagem paralelos.	68
5.1	<i>Map matching</i>	77
5.2	Disposição dos diversos sistemas referenciais em função da Terra.	80
5.3	Esquema da mecanização segundo o referencial de nível local.	82
5.4	Determinação das coordenadas tridimensionais de um ponto pela intercepção de segmentos de recta semi-infinitos.	94
5.5	Princípio da parametrização <i>inverse depth</i>	96
6.1	Exemplo de um modelo em nuvem de pontos.	106
6.2	Exemplo de um modelo em malha poligonal.	107
7.1	Sensor inercial iNAV-FMS-E.	111
7.2	Inclinómetro 3DM-GX1.	112
7.3	Sensor <i>laser range finder</i> SICK LMS-200.	112
7.4	Câmara JAI modelo CB-080GE.	113
7.5	Plataforma sensorial transportada sobre o veículo de mapeamento.	113
7.6	Grande plano do sensor 3DM-GX1 posicionado sobre a câmara da esquerda. 114	
7.7	Veículo de mapeamento na zona do túnel, instantes antes da recolha de dados. 116	
7.8	Diferença de iluminação entre uma imagem captada à entrada do túnel e outra adquirida já no interior.	117
8.1	Filtragem de correspondências através do método RANSAC tradicional. À esquerda, coorespondências pela comparação de descritores SIFT. À direita as correpondências aceites pelo método de RANSAC.	121
8.2	Distribuição das <i>features</i> observadas em função da sequência de <i>frames</i> . . . 122	
8.3	Fluxograma do algoritmo de SLAM.	124
8.4	Representação da orientação dos sistemas de eixos dos sensores.	125
8.5	Sequência de rotações que alinham o sistema referencial do sensor iNAV-FMS-E com o sistema referencial do sensor 3DM-GX1.	126
8.6	Diferença de orientação do referencial da câmara entre a solução utilizada neste trabalho (esquerda) e a versão original da parametrização em profundidade inversa (direita).	129
8.7	Representação da janela de validação.	133
8.8	Trajectória tridimensional calculada por cada método.	135
8.9	Posição final das várias trajectórias.	135
8.10	Vista superior dos percursos (unidades em metros).	136
8.11	Erros em posição segundo a direcção Norte.	137
8.12	Erros em posição segundo a direcção Este.	138
8.13	Erros em posição segundo a direcção do centro da Terra.	138
8.14	Erros em orientação segundo o eixo <i>roll</i>	139

8.15	Erros em orientação segundo o eixo <i>pitch</i>	140
8.16	Erros em orientação segundo o eixo <i>yaw</i>	140
8.17	Erros em velocidade segundo a direcção do centro da Terra.	142
8.18	Erros em velocidade segundo a direcção Norte.	143
8.19	Erros em velocidade segundo o a direcção Este.	144
8.20	Disposição do sistema de eixos do sensor LRF lateral.	145
8.21	Sequência de transformações que, quando aplicada aos ponto no referencial do sensor LRF lateral, os define segundo o referencial do corpo.	146
8.22	Nuvem de pontos do túnel visualizada a partir de uma das extremidades.	147
8.23	Nuvem de pontos vista de outra perspectiva.	147
8.24	Nuvem de pontos de toda a estrutura.	148
8.25	Secção ampliada da nuvem de pontos.	148
8.26	Secção convertida em malha poligonal.	148
8.27	Malha poligonal com iluminação.	149
8.28	Superfícies resultantes da eliminação das fronteiras entre poligonos.	150
8.29	Modelo final após etapa de filtragem geométrica.	151
B.1	Ilustração de uma rotação operada através de um quaternião.	180
B.2	Exemplo de uma sequência de rotações para alinhamento entre dois sistemas de coordenadas utilizando os Ângulos de Euler.	186
C.1	Desvio padrão nos estados de posição.	187
C.2	Desvio padrão nos estados de orientação.	188

Esta página foi intencionalmente deixada em branco.

Lista de Símbolos

$\frac{\partial \bullet}{\partial \bullet\bullet}$	Derivada parcial de \bullet em ordem a $\bullet\bullet$
σ	Desvio padrão
σ^2	Variância
\otimes	Convolução
λ	Valor próprio
det	Determinante
I	Imagem
R^i	Plano da imagem
R^c	Ponto do centro óptico
P_w	Ponto tridimensional no referencial global
P_c	Ponto de projecção de P_w no referencial da câmara
$P_{u,v}$	Ponto no referencial de píxeis da imagem
P_i	Ponto no sistema referencial métrico da imagem
x^i	Coordenada x no referencial métrico da imagem
y^i	Coordenada y no referencial métrico da imagem
x^c	Coordenada x no referencial da câmara
y^c	Coordenada y no referencial da câmara
z^c	Coordenada z no referencial da câmara
x^w	Coordenada x no referencial do mundo
y^w	Coordenada y no referencial do mundo
z^w	Coordenada z no referencial do mundo
x^d	Coordenada x afectada pela distorção da lente
y^d	Coordenada y afectada pela distorção da lente
x^u	Coordenada x ignorando a distorção provocada pela lente
y^u	Coordenada y ignorando a distorção provocada pela lente
z^u	Coordenada z ignorando a distorção provocada pela lente
f	Distância focal
f_x	Distância focal segundo a componente horizontal
f_y	Distância focal segundo a componente vertical
$c.o.$	Centro óptico
u	Coordenada horizontal na imagem
v	Coordenada vertical na imagem
C	Matriz de rotação
C_x	Matriz de rotação em torno do eixo x
C_y	Matriz de rotação em torno do eixo y
C_z	Matriz de rotação em torno do eixo z
C_{xyz}	Matriz de rotação em torno dos eixos x, y e z
$C^{\bullet\bullet}$	Matriz de rotação do referencial \bullet para o referencial $\bullet\bullet$

t	Vector de translação
(t_x, t_y, t_z)	Componentes do vector de translação t
k_u	Factor de escala horizontal para conversão entre o sistema referencial métrico da imagem e o sistema referencial em píxeis da imagem
k_v	Factor de escala vertical para conversão entre o sistema referencial métrico da imagem e o sistema referencial em píxeis da imagem
κ	Parâmetro de distorção da lente
u_0	Coordenada horizontal no referencial em píxeis da imagem onde se projecta o centro óptico
v_0	Coordenada vertical no referencial em píxeis da imagem onde se projecta o centro óptico
K	Matriz dos parâmetros intrínsecos
(X_w, Y_w, Z_w)	Coordenadas de um ponto no referencial global
e_l	Epipolo na imagem da esquerda
e_d	Epipolo na imagem da direita
p_l	Ponto de projecção no plano da imagem da esquerda
p_r	Ponto de projecção no plano da imagem da direita
F	Matriz fundamental
E	Matriz essencial
ϕ	Ângulo <i>roll</i>
θ	Ângulo <i>pitch</i>
ψ	Ângulo <i>yaw</i>
$x(k)$	Vector de estados no instante k
$x_v(k)$	Estados do veículo no instante k
$l_N(k)$	Estado da <i>landmark</i> N no instante k
$P(k)$	Covariância no instante k
$\hat{x}(k k-1)$	Estado previsto
$\hat{x}(k k)$	Estado estimado
$P(k k-1)$	Covariância prevista
$P(k k)$	Covariância estimada
\hat{z}	Observação prevista
r^n	Posição da plataforma sensorial definida no referencial de nível local
q^n	Orientação da plataforma sensorial definida no referencial de nível local
v^n	Velocidade da plataforma sensorial definida no referencial de nível local
q()	Conversão para quaterniões
R	Conversão em matriz rotação
g^n	Aceleração da gravidade no referencial de navegação
\bullet^n	Grandeza \bullet definida no referencial de nível local
\bullet^b	Grandeza \bullet definida no referencial do corpo

Acrónimos

2D	Bidimensional
3D	Tridimensional
6 DoF	<i>Six Degrees of Freedom</i>
AMOS	<i>Autonomous MOBILE Systems</i>
BPA	<i>Ball-Pivoting Algorithm</i>
CCD	<i>Charge-coupled device</i>
CenSure	<i>Center Surround Extrema</i>
DoG	Diferença de Gaussianas
ECEF	<i>Earth Centered, Earth Fixed</i>
FAST	<i>Features from Accelerated Segment Test</i>
FBSM	<i>Feature-based Scan Matching</i>
FIFO	<i>First In, First Out</i>
GigE	<i>Gigabit Ethernet</i>
GPIO	<i>General Purpose Input/Output</i>
GPS	<i>Global Positioning System</i>
HAYAI	<i>High-speed and Yet Accurate Indoor/Outdoor Tracking</i>
ICP	<i>Iterative Closest Point</i>
ID3	<i>Iterative Dichotomiser 3</i>
IDC	<i>Iterative Dual Correspondence</i>
IMU	<i>Inertial Measurement Unit</i>
INS	<i>Inertial Navigation System</i>
IPP	Instituto Politécnico do Porto
ISEP	Instituto Superior de Engenharia do Porto
Kd-tree	<i>K-dimensional tree</i>
LoG	Laplaciano de Gaussianas
LRF	<i>Laser Range Finder</i>
LSA	Laboratório de Sistemas Autónomos
MbICP	<i>Metric-based Iterative Closest Point</i>
MEMS	<i>Micro Electro-Mechanical Systems</i>
NCC	<i>Normalized Cross Correlation</i>
NDT	<i>The Normal Distributions Transform</i>
OpenGL	<i>Open Graphics Library</i>
PPS	<i>Pulse per Second</i>
RANSAC	<i>Random Sample Consensus</i>

RBPF	<i>Rao-Blackwellized Particle Filter</i>
RTK	<i>Real Time Kinematic</i>
SAD	<i>Sum of Absolute Differences</i>
SBA	<i>Sparce Bundle Adjustment</i>
SfM	<i>Structure from Motion</i>
SIFT	<i>Scale-Invariant Feature Transform</i>
SLAM	<i>Simultaneous Localization and Mapping</i>
SSD	<i>Sum of Squared Differences</i>
STCNN	<i>Spherical Triangle Constraint Nearest Neighbor</i>
SVD	<i>Singular Value Decomposition</i>
VRML	<i>Virtual Reality Modelling Language</i>
ZNCC	<i>Zero-mean Normalized Cross Correlation</i>

Capítulo 1

Introdução

Conteúdo

1.1	Âmbito da Dissertação	1
1.2	Objectivos	3
1.3	Estrutura da Dissertação	4

1.1 Âmbito da Dissertação

Esta dissertação aborda o tema do mapeamento robótico dinâmico, que pode ser definido como o processo de aquisição de modelos espaciais do ambiente físico circundante, por parte de um ou mais robôs móveis. Ao longo dos últimos anos esta matéria tem despertado grande interesse no seio da comunidade de robótica móvel, na medida em que constitui uma etapa determinante no desenvolvimento de robôs com elevado grau de autonomia, pois a execução de tarefas evoluídas requer conhecimentos robustos acerca das características do espaço envolvente. Inúmeros robôs fazem uso de mapas para desenvolver processos como a localização [1] [2] [3], o planeamento de movimento [4] e a detecção de obstáculos [5], [6], [4].

Recentemente o mapeamento robótico assumiu um enquadramento mais abrangente, surgindo sistemas de aquisição de informações para usos extra-robô. Embora em algumas aplicações a locomoção autónoma da plataforma de mapeamento seja crucial, devido à perigosidade ou inacessibilidade de humanos às zonas de interesse, uma das tendências passa pela substituição do veículo robótico por uma plataforma controlada por um condutor humano. Consequentemente logo a designação de mapeamento robótico deixa de

fazer sentido, sendo mais adequado referir o processo como sistema de aquisição de modelos geométricos do ambiente. Com a introdução de um operador humano são facilmente ultrapassadas algumas das dificuldades demonstradas por veículos autónomos, com ênfase na operação em ambientes dinâmicos, que exigem estratégias robustas de planeamento de movimento e detecção de obstáculos. Deste modo, os investigadores podem dedicar-se exclusivamente ao desenvolvimento das capacidades de mapeamento. Contudo os métodos de obtenção dos modelos são equivalentes aos empregues em sistemas robóticos, razão pela qual a maior parte dos estudiosos desta matéria pertence à comunidade da robótica móvel.

Como evidenciado em [7] o processo de mapeamento torna-se bastante simples quando a localização do veículo é conhecida. No entanto, na generalidade dos casos a localização não está garantida à partida e, em ambientes sem estruturas que facilitem esta tarefa, ambos os processos são solucionados em simultâneo, o que implica um elevado acréscimo de complexidade. Actualmente, este é um dos problemas mais desafiantes no campo da robótica móvel e toma a designação de mapeamento e localização em simultâneo – SLAM (*Simultaneous Localization and Mapping*). Este paradigma surgiu como resposta à grande dificuldade de localização exacta de robôs em ambientes desconhecidos, desprovidos de estruturas dedicadas ao fornecimento de informações de posição – *beacons* – e onde se manifesta a inacessibilidade a tecnologias de posicionamento global, designadamente GPS. A estratégia de SLAM pode ser formulada do seguinte modo: um robô largado num ponto inicial arbitrário deverá ser capaz de ganhar conhecimento sobre o ambiente, por intermédio exclusivo de sensores a bordo à medida que se vai deslocando, as informações sensoriais são interpretadas e, com fundamento nestas, é construído um mapa relativamente ao qual o robô se localiza. Resumidamente o conceito de SLAM consiste numa simbiose entre os processos de mapeamento e localização, pois a construção do mapa depende dos dados de localização e vice-versa.

Os sistemas de mapeamento 3D diferenciam-se especialmente pelo ambiente em que operam, existindo diferenças consideráveis entre aqueles que exploram ambientes fechados (*indoor*) e os que actuam em cenários a céu aberto (*outdoor*). A razão desta divergência deve-se especialmente aos métodos de localização favoráveis a um e outro cenário, por exemplo em ambientes *outdoor* a localização pode usufruir das vantagens dos sistema GPS enquanto que em alguns ambientes interiores o piso pode corresponder a um plano horizontal, o que também simplifica o problema de localização. A construção de um mapa

em ambientes não estruturados, onde se salienta especialmente a irregularidade do piso ou a sua inclinação, levam a que o veículo se mova num espaço tridimensional, logo a localização do mesmo passa a ser definida em seis graus de liberdade – 6 DoF (*six degrees of freedom*) – três deles reflectem a posição global, segundo as coordenadas x, y e z, enquanto que os restantes definem a atitude, ou seja a orientação tridimensional usualmente representada pelos ângulos *roll*, *pitch* e *yaw*.

Os modelos tridimensionais construídos pelos sistemas de modelização são normalmente exibidos em suporte de realidade virtual. Este modo de apresentação contempla a possibilidade de interacção por parte do utilizador, permitindo movimentar o modelo em todas as direcções e escolher a melhor perspectiva de visualização.

1.2 Objectivos

Esta dissertação incide sobre o desenvolvimento de um sistema de modelização tridimensional de estruturas do tipo túnel. O referido sistema será montado num veículo automóvel e deverá ser capaz de, através de sensores e equipamentos computacionais, colher informações sensoriais à medida que um veículo se desloca ao longo de um túnel. Os dados adquiridos serão usados para calcular com exactidão a localização do veículo no espaço tridimensional bem como produzir um modelo geométrico do espaço.

As características estruturais de um túnel são propícias à manifestação severa do efeito *sensor aliasing* [8], advindo daí uma contrariedade que põe em causa a viabilidade de um sistema com estas pretensões. O fenómeno de *sensor aliasing* caracteriza-se pela monotonia das informações adquiridas pelos sensores, fruto da incidência sobre ambientes uniformes do ponto de vista estrutural. Nestas situações, a aquisição de informações em zonas diversas resulta numa percepção sensorial idêntica, facto que introduz ambiguidade no processo de localização. Dentro de um túnel, a carência de informações de posição é mais notória no sentido da translação, pois segundo esta componente a estrutura é bastante semelhante ao longo de toda a galeria.

Face à ausência de desenvolvimentos dirigidos para localização autónoma e mapeamento deste tipo de estruturas, os objectivos desta dissertação passam pelo cumprimento dos seguintes pontos:

1. Identificação e análise das técnicas empregues em sistemas de mapeamento tridimensional de ambientes com ausência de estruturas de posicionamento global, como são os *underground*;
2. Aprofundamento de conhecimentos na área da localização autónoma, incidindo especialmente em estratégias que possibilitem ultrapassar as dificuldades impostas pela monotonia das estruturas em causa;
3. Avaliação das técnicas mais promissoras do estado da arte e formalização de uma arquitectura robusta, que permita atingir um sistema de localização e mapeamento tridimensional de elevada exactidão, com aplicação em túneis e minas;
4. Implementação da arquitectura e teste do sistema com base em dados adquiridos em ambiente real.

1.3 Estrutura da Dissertação

No próximo capítulo é apresentado um estudo do estado da arte, onde figuram sistemas de modelização tridimensional *indoor*. Foram seleccionadas soluções com características diferentes de modo a obter uma visão sobre as diversas abordagens ao problema. Um aspecto transversal a todos os métodos diz respeito à aplicação de sensores *laser range finder* (LRF) na aquisição das informações geométricas, a partir das quais se constrói o modelo tridimensional. Por esse motivo, precedendo a referência aos sistemas de modelização tridimensional, é feita menção aos vários tipos de sensores LRF e às etapas de tratamento dos dados geralmente empregues na construção do modelo. Por fim, em jeito de conclusão, os estudos apresentados são enquadrados com a problemática de mapeamento de túneis. Deste processo despertam alguns argumentos desfavoráveis, que põem em causa a aptidão dos sistemas analisados à operação no interior de estruturas em túnel.

O terceiro capítulo aprofunda conhecimentos ligados aos processos de registo de pontos adquiridos por intermédio de sensores LRF. O algoritmo ICP (*Iterative Closest Point*) é estudado de forma mais completa, sendo referidas estratégias para melhoramento do desempenho e da robustez.

No quarto capítulo são introduzidas técnicas de visão computacional ligadas aos processos de localização.

O capítulo quinto é voltado para a análise de soluções de localização aplicadas na área da robótica móvel, com o objectivo de identificar metodologias adequadas à localização dentro de túneis.

O capítulo sexto versa sobre formas de representação de modelos tridimensionais de estruturas de grande dimensão.

O capítulo designado por “Aquisição de dados” incide sobre o sistema de aquisição de informações sensoriais, destacando-se a apresentação dos sensores e suas características. São abordadas questões essenciais como a relação entre sistemas referenciais e o registo do instante temporal de cada medida.

O oitavo capítulo desvenda a arquitectura escolhida para o sistema de localização, bem como a estratégia empregue na construção do modelo. São ainda apresentados e analisados os resultados obtidos.

Por último, no capítulo nono, são tecidas algumas considerações finais e lançadas ideias para possíveis melhoramentos do sistema, a implementar no futuro.

Esta página foi intencionalmente deixada em branco.

Capítulo 2

Estado da Arte

Conteúdo

2.1	Sensores e Técnicas de Mapeamento Associadas	7
2.2	Análise a Diversos Sistemas de Mapeamento	11
2.2.1	2x2D SLAM – Mapeamento de Minas	11
2.2.2	Mapeamento de Minas com LRF 3D – <i>Groundhog</i>	13
2.2.3	Mapeamento <i>Indoor</i> Através de Câmaras e LRF 3D	16
2.2.4	Mapeamento com LRF 3D Estático	18
2.3	Conclusão e Contextualização com a Problemática de Mapeamento e Localização em Túneis	20

2.1 Sensores e Técnicas de Mapeamento Associadas

Os métodos de aquisição de modelos digitais fazem uso de uma grande variedade de sensores, contudo os sensores *Laser Range Finder* e as câmaras encontram-se no topo das preferências. Os sensores LRF representam o ambiente de forma densa por intermédio de um conjunto de pontos com coordenadas bem definidas segundo o sistema referencial do sensor, o que os torna indicados para a percepção das características geométricas do cenário. As imagens produzidas pelas câmaras captam informações diversas como cor, textura e geometria, todavia, a projecção de pontos do plano da imagem para o cenário é afectada por inúmeras fontes de erro como a má calibração, a sensibilidade a variações de iluminação e o erro na atribuição de correspondências, sendo difícil atingir mapas densos de profundidade com as precisões exigidas por uma aplicação de mapeamento. Na maior

parte dos casos esta condicionante afasta as câmaras da tarefa de percepção da geometria da cena. Porém a representação realista do cenário leva a que o modelo não seja apenas constituído por dados geométricos, passando a incorporar as informações relativas à cor e textura das superfícies apreendidas pelas câmaras [9] [10] [11] [12].

Dentro da categoria dos sensores LRF e no contexto do mapeamento volumétrico é oportuno mencionar as diferenças entre os dispositivos que executam varrimentos bidimensionais e os que geram uma nuvem de pontos tridimensional em redor de si próprios. Apesar das aplicações de mapeamento volumétrico terem como objectivo a construção de um modelo tridimensional, verifica-se que os sensores LRF 2D são bastante populares, especialmente em ambientes *indoor*. Geralmente são instalados numa plataforma móvel com o plano de varrimento na vertical, de modo a captar sucessivas secções de corte perpendiculares ao sentido de movimento que, com base na localização do veículo, são dispostas segundo um sistema de coordenadas global, de modo a formar uma nuvem de pontos [7] [13] [14] [15] [16]. Segundo esta disposição, superfícies paralelas ao plano de varrimento não são detectadas. Zhao e Shibasaki [15], numa abordagem em cenário *outdoor*, anulam as oclusões recorrendo a dois LRFs 2D dispostos com uma orientação angular relativa de 45° .

Os sensores LRF 3D são menos propícios aos fenómenos de oclusão, no entanto são significativamente mais dispendiosos e possuem algumas características que os tornam menos interessantes para aplicações de mapeamento 3D de estruturas *indoor*. Uma característica desfavorável diz respeito à abertura limitada segundo o plano vertical [17] [18], facto que impede a percepção de uma zona em forma de cone sobre e sob o sensor. Huber e Vandapel [19] inclinam propositadamente o sensor para que incida sobre o piso e o tecto na sua proximidade. Por outro lado, uma estrutura do tipo corredor digitalizada por intermédio de um LRF 3D perde resolução à medida que diminui o ângulo de incidência do feixe laser nas superfícies.

Porém, um LRF 3D permite apreender de uma só vez um volume considerável logo, no enquadramento desta dissertação, vislumbra-se um benefício relativo à utilização deste face ao seu congénere 2D, pois a aquisição de uma secção volumétrica alargada num único varrimento poderá significar uma maior tolerância do sistema ao aumento de velocidade do veículo de mapeamento.

Ao longo dos últimos anos popularizaram-se as configurações que associam sensores

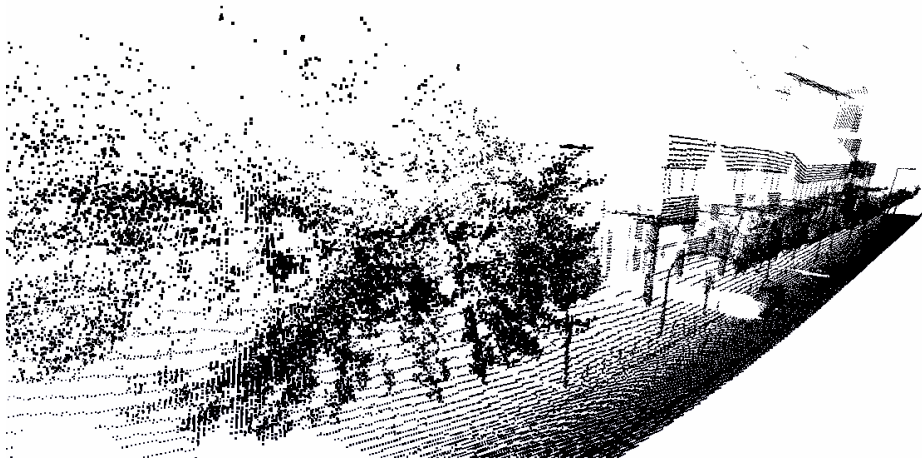


Figura 2.1: Nuvem de pontos registada por um sensor LRF, onde é evidente a perda de resolução originada pela diminuição do ângulo de incidência.

LRF 2D a um mecanismo de orientação (figura 2.2), dotando assim o sistema de um grau de liberdade adicional de modo a simular um sensor de varrimento tridimensional [20] [10] [21]. O inconveniente mais relevante destas configurações prende-se com a morosidade da etapa de aquisição de informações, que obriga à imobilização do veículo para que todos os segmentos adquiridos a cada posição angular sejam relativos a uma localização comum. Só assim é possível garantir o agrupamento consistente de todas as secções num único conjunto de pontos 3D, cuja disposição espacial reproduza fielmente o espaço envolvente.

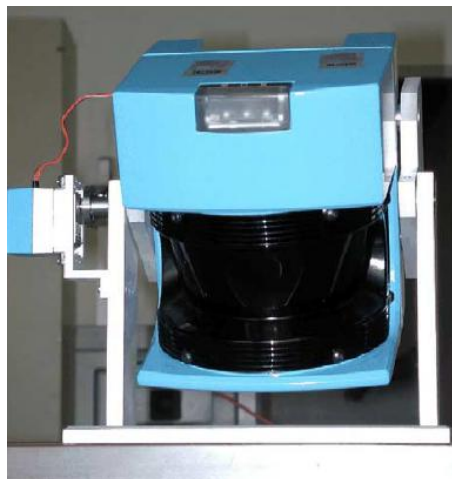


Figura 2.2: Exemplo de um sistema de aquisição de varrimentos tridimensionais baseado na variação do ângulo vertical de incidência de um sensor LRF 2D.

A construção de um modelo pela integração de múltiplas imagens captadas por um LRF 3D é normalmente decomposta em duas etapas, nomeadamente registo (*registration*) e integração (*integration*). O termo “registo” refere-se ao cálculo de uma transformação de corpo rígido¹ que traduz o desalinhamento e afastamento entre duas imagens que, embora parcialmente, devem representar uma zona comum. O processo de integração ocorre após o registo e diz respeito à manipulação dos dados por forma a atingir uma representação única da mesma área, eliminando dados sobrepostos e redundantes. A precisão conseguida na fase de registo reflecte-se directamente na qualidade da junção das superfícies, neste campo os métodos baseados em minimização de distâncias [22] [23] são os mais utilizados.

Em sistemas onde o modelo é construído a partir de medições de LRF, a nuvem de pontos final sofre um processo de conversão, do qual resulta uma representação sob a forma de superfícies poligonais. Nesta etapa são empregues estratégias [24] [12] [25] [26] para indagar a nuvem de pontos de modo a formar grupos de pontos vizinhos, cujas coordenadas irão definir os vértices da superfície poligonal. O tipo de superfície poligonal relaciona-se directamente com a dimensão dos conjuntos formados contudo, em aplicações de mapeamento tridimensional, são mais comuns as superfícies triangulares, pois a sua formação implica uma perda mínima de informação [27].

Para produzir um cenário visualmente mais realista as superfícies poligonais podem ser preenchidas com informações de textura apreendidas através câmaras calibradas [9] [10] [12]. Os vértices de cada polígono são projectados no plano da imagem, pela aplicação dos parâmetros intrínsecos, extrínsecos e coeficientes de distorção, permitindo desta forma seleccionar o fragmento correspondente, que poderá ser alvo de transformações de perspectiva para se ajustar perfeitamente à superfície. É necessário assegurar que a disposição das câmaras permite visualizar toda a área de operação do sensor LRF, caso contrário não existem informações suficientes para preencher o modelo na totalidade.

¹Uma transformação de corpo rígido é definida por uma rotação e uma translação arbitrarias aplicadas a um determinado objecto. Caracteriza-se pela preservação das características geométricas do corpo, tais como ângulos e comprimentos. Assim, a transformação de corpo rígido, produz consequências exclusivamente na posição e orientação do objecto face ao mundo.

2.2 Análise a Diversos Sistemas de Mapeamento

Nesta secção são mencionados trabalhos de destaque no âmbito do mapeamento 3D em ambientes indoor.

2.2.1 2x2D SLAM – Mapeamento de Minas

Em [14] é apresentado um método para a aquisição de mapas tridimensionais por parte de um veículo não motorizado, cujos testes decorreram numa exploração mineira subterrânea. Embora não sejam divulgados valores específicos, devido à inexistência de *ground truth* com qualidade, é assegurado pelos autores que o sistema possui um grau de precisão satisfatório, atestado pela semelhança entre as plantas da infra-estrutura e os mapas resultantes das experiências. A aquisição de dados é levada a cabo por intermédio de quatro sensores *SICK Laser Range Finder*, dois deles especificamente dedicados à estimação da localização do veículo, sendo os restantes aplicados na tarefa de percepção da estrutura da mina para efeitos de construção do mapa.



Figura 2.3: Disposição dos quatro sensores LRF.

Os sensores *Laser Range Finder* empregues na localização encontram-se instalados na parte dianteira do veículo e apontam em frente, todavia um deles executa um varrimento horizontal enquanto o plano de acção do outro é vertical; configuração que aproxima de forma incompleta a estimação da localização em três dimensões, através de uma estratégia

conhecida como *scan matching*. Os métodos de *scan matching* calculam uma estimativa da discrepância em posição e orientação entre dois varrimentos subsequentes, pelo estabelecimento de correspondências entre ambos. Os valores calculados reflectem a movimentação do veículo entre os dois instantes de amostragem, logo a sua integração permite perceber a trajectória do veículo. A localização 3D em [14] é decomposta em dois processos de *scan matching* 2D a operar separadamente nos planos vertical e horizontal, abordagem à qual se dá o nome de 2×2D SLAM. Apesar da incapacidade de estimação do grau de liberdade *roll*, esta estratégia confere a robustez necessária à construção de mapas de grande escala, mantendo o erro em posição na ordem dos centímetros para distâncias percorridas de centenas de metros.

Não existe uma independência total entre os processos de localização, sendo trocadas algumas informações com o objectivo de identificar situações excepcionais que podem comprometer os mecanismos de *scan matching*. Tomemos como exemplo uma depressão no pavimento capaz de provocar a alteração do ângulo *pitch* do veículo, de tal forma que o varrimento do sensor horizontal intercepte o solo. Neste caso a porção do varrimento correspondente à detecção do solo iria resultar na identificação de um obstáculo fantasma e prejudicar, senão mesmo levar à falha, do algoritmo de *scan matching*. Tendo em conta que a depressão no solo pode ser facilmente detectada pelo processo de localização segundo a dimensão vertical, é legítima a utilização destas informações pelo processo de localização complementar para evitar situações de ambiguidade deste tipo. Relativamente ao processo de localização vertical, podem surgir obstáculos fantasma em locais onde se verifique um elevado grau de curvatura das paredes e o varrimento vertical incida sobre essas superfícies.

O modelo tridimensional do espaço é constituído por secções de corte verticais, captadas sucessivamente pelos sensores com plano de varrimento vertical e perpendicular à direcção de movimento. Cada corte é projectado segundo um sistema referencial global, por intermédio dos valores de localização, procedimento do qual resulta uma nuvem de pontos. O conjunto de pontos é submetido a uma etapa de filtragem que agrupa num único elemento pontos representantes de uma área comum. Posteriormente os pontos são associados para formar polígonos, que proporcionam uma representação contínua das estruturas. Finalmente, o modelo poligonal é convertido para o formato de realidade virtual VRML (*Virtual Reality Modelling Language*) – um formato de ficheiro dedicado à representação de ambientes tridimensionais. A visualização do modelo num *software* dedicado

à apresentação deste formato de dados permite ao utilizador interagir com o modelo e explorá-lo de forma virtual.

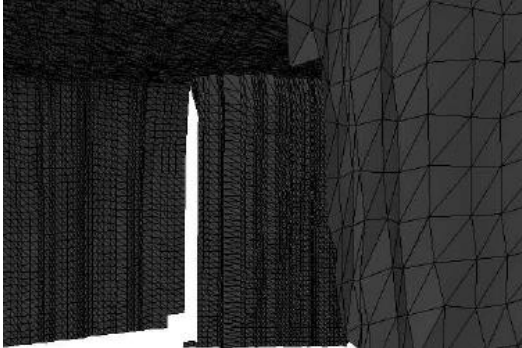


Figura 2.4: Representação poligonal.



Figura 2.5: Modelo em formato VRML.

2.2.2 Mapeamento de Minas com LRF 3D – *Groundhog*

No artigo [28] são relatadas as características de um sistema de mapeamento de minas, que utiliza no processo de exploração um veículo de operação remota [16] com o nome *Groundhog*. Ao contrário do método apresentado no ponto anterior, este sistema garante o cálculo da localização em seis graus de liberdade. A aquisição de dados sensoriais fica a cargo de dois sensores SICK LRF, cada qual colocado numa das extremidades do veículo e montado numa estrutura que modifica a sua inclinação segundo o plano vertical. O mecanismo proporciona uma abertura angular de 60° , que a par da abertura horizontal de 180° própria do sensor, permitem registar imagens 3D com resolução horizontal de 361 pontos por 341 pontos na vertical. Durante a aquisição de um varrimento o veículo encontra-se imobilizado para que todos os pontos do conjunto partilhem uma localização comum, pois o algoritmo aplicado na etapa de registo assim o exige.

A solução mais simples para registo de varrimentos 3D tem o nome de *pairwise matching*, segundo ela cada varrimento é alinhado relativamente ao conjunto que oferece a maior superfície de sobreposição. Esta estratégia é perfeitamente válida em aplicações onde a sobreposição de superfícies ocorre apenas entre pares de varrimentos, porém a possibilidade de existência de zonas de sobreposição em mais que dois varrimentos pode resultar em mapas inconsistentes, pois os erros de registo acumulam-se e diminuem a precisão das transformações. Por este motivo, os autores desenvolveram um algoritmo ao qual atribuem o nome de *simultaneous matching*, baseado no predecessor proposto por Pulli [29], que regista simultaneamente os vários varrimentos com zonas de sobreposição



Figura 2.6: Veículo de operação remota *groundhog*. No canto inferior direito pormenor do mecanismo de inclinação de um dos sensores LRF.

comuns, conseguindo assim minimizar o erro global e desse modo evitar a construção de cenários inconsistentes.

O algoritmo *simultaneous matching* considera o primeiro varrimento como referência – *master scan* – a este é atribuída uma posição fixa. Para cada varrimento subsequente está prevista uma etapa inicial onde é executado um registo pouco rigoroso, com base no algoritmo *pairwise matching*. Neste passo é considerada uma aproximação inicial dada pela hodometria do veículo, o que acelera a convergência e diminui significativamente o consumo de requisitos computacionais. Em seguida é criada uma fila FIFO constituída unicamente pelo novo varrimento. Decorre então o registo de alta precisão do novo varrimento, acompanhado da minimização global do erro, pelo cumprimento da seguinte sequência de acontecimentos, que se repete enquanto existirem elementos na fila:

1. O primeiro varrimento da fila é removido e sobre ele recai o processo de alinhamento;
2. Se o varrimento actual não corresponder ao *master scan* procede-se à pesquisa dos varrimentos vizinhos – aqueles que possuem zonas de sobreposição comuns a este com mais de 250 pontos. O conjunto de varrimentos vizinhos serve de referência ao alinhamento do varrimento actual através da aplicação do algoritmo ICP [22].
3. Caso o varrimento actual sofra uma alteração de posição ou orientação pela aplicação da transformação calculada no processo anterior, todos os varrimentos vizinhos que ainda não se encontrem na fila FIFO são adicionados a esta estrutura.

De acordo com sequência de acontecimentos apresentada, um varrimento registado à posteriori pode produzir influências na disposição dos varrimentos anteriormente registados, permitindo assim usufruir do aumento de informações para minimizar o erro de registo.

No trabalho em análise é ainda proposta uma metodologia de redução da densidade de dados, com o intuito de melhorar o desempenho computacional do sistema sem causar prejuízos ao nível da precisão do modelo. Esta abordagem contempla ainda a filtragem de ruídos gaussianos e do ruído conhecido como *salt and pepper noise*, este último ocorre especialmente em *edges*, quando o feixe laser incide sobre superfícies distintas de forma intermitente. A filtragem de ruído gaussiano é conseguida pela substituição da distância Euclidiana de um determinado pontos pela média registada numa vizinha de 7 pontos. O ruído *salt and pepper* é minimizado através do mesmo procedimento, mas só tem efeito quanto a diferença entre o valor médio e o valor do ponto for superior a um determinado limite². Por sua vez, a redução da densidade de dados recai especialmente sobre os pontos com menor distância face ao sensor, pois apresentam uma descrição mais densa da superfície. Nestes casos os pontos numa dada vizinhança são aglomerados num único elemento, processo que simultaneamente origina a diminuição do número de pontos e leva a cabo a filtragem de ruído gaussiano. Como resultado da etapa de redução de dados, a distância relativa entre pontos vizinhos passa a ser no mínimo igual a 10 cm.

As preocupações com a diminuição da complexidade do algoritmo levaram ao emprego de uma estrutura do tipo *kd-tree* na pesquisa das correspondências entre pontos necessárias ao algoritmo ICP.

O modelo é apresentado sob a forma de uma nuvem de pontos por intermédio de um programa desenvolvido em OpenGL, que permite assim visualizar a estrutura de qualquer perspectiva.

²É proposto pelos autores o valor de 200 cm

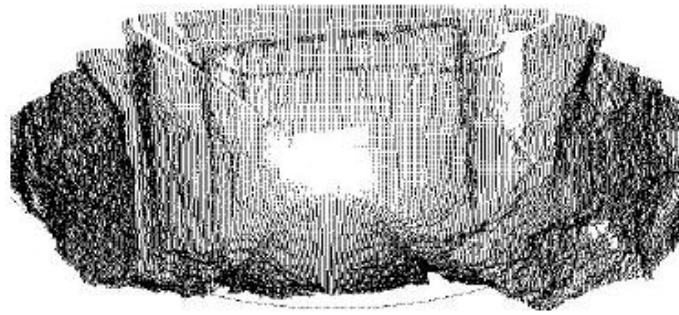


Figura 2.7: Representação da nuvem de pontos originada por um varrimento 3D.

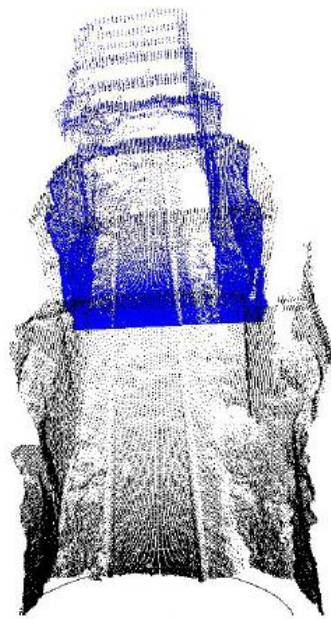


Figura 2.8: Vista elevada de uma galeria virtualmente reconstruída pelo registo de múltiplos varrimento 3D.

2.2.3 Mapeamento *Indoor* Através de Câmaras e LRF 3D

O artigo [9] apresenta um método de mapeamento *indoor* que integra de um sensor *Laser Range Finder* 3D com oito câmaras CCD e sensores de *dead reckoning*. O sistema produz um modelo 3D que incorpora características geométricas e textura do ambiente, percebida através das imagens de intensidade. As limitações computacionais da época impediram a implementação de um processo em tempo real e totalmente autónomo, contudo a disposição física dos sensores e as técnicas de tratamento de dados são dignas de referência.

Numa fase antecedente à missão de mapeamento, o veículo é posicionado sob uma estrutura de calibração constituída por alvos brancos esféricos com posição global conhecida, relativamente à qual são adquiridas informações pelos sensores para determinar os parâmetros que transformam medidas definidas no plano da imagem dos sensores para um sistema referencial global.

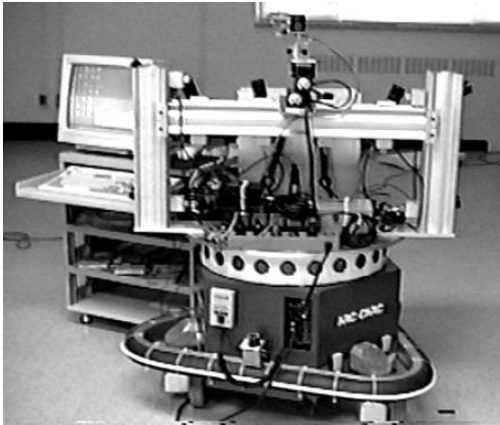


Figura 2.9: Veículo de mapeamento.



Figura 2.10: Estrutura de calibração.

A etapa de mapeamento inicia-se com a movimentação do veículo pelo cenário para aquisição de informações sensoriais. Ao longo do percurso são efectuadas paragens sucessivas, durante as quais os sensores realizam medições. A imobilização do veículo durante os períodos de amostragem garante que os dados de todos os sensores são referenciados segundo a mesma posição global, logo é assegurada a validade dos parâmetros obtidos pela calibração para transformação entre sistemas referenciais. Ao longo do percurso são colocados alguns alvos visuais, com o intuito de proporcionar robustez e facilitar as tarefas de visão computacional.

Uma vez explorado o cenário dá-se a transferência de dados sensoriais para um equipamento informático externo, onde é executada a etapa de pós-processamento que leva à construção do modelo tridimensional. Inicialmente corre um algoritmo de extracção de *features* nas imagens recolhidas pelas câmaras CCD, devido sobretudo a erros de correspondência o processo é supervisionado por um operador humano. A posição das *features* no plano da imagem em conjunto com os parâmetros de calibração são remetidos ao método de optimização *Bundle Adjustment*, que calcula a posição e orientação das câmaras segundo um sistema referencial global, o que permite inferir a localização da plataforma. Por sua vez os valores de localização são utilizados para referenciar, segundo o sistema de

coordenadas global, os pontos adquiridos pelo sensor LRF.

A nuvem de pontos resultante do processo anterior possui uma elevada densidade de dados, o que na altura comprometia qualquer tentativa de representação geométrica e interação com a mesma, devido aos custos computacionais inerentes. Por outro lado, a distribuição de pontos pelo espaço não é uniforme e as zonas de sobreposição de varrimentos contêm informações redundantes, facto que não acrescenta benefícios ao modelo e o torna mais complexo. Por estes motivos a nuvem de pontos é sujeita à remoção dos pontos redundantes, sendo em seguida convertida numa representação geométrica em malha triangular, o que facilita a tarefa de renderização [30]. Tendo em consideração a posição e orientação das câmaras calculada pelo método *Bundle Adjustment*, é possível preencher os polígonos da representação geométrica com a intensidade luminosa e textura apreendidas através das câmaras. Neste sentido os vértices de cada polígono são projectados no plano da imagem de modo a definir a secção correspondente, que em seguida sofre transformações de perspectiva de modo a encaixar perfeitamente na área do polígono. O resultado consiste numa representação realista da geometria e textura do cenário.

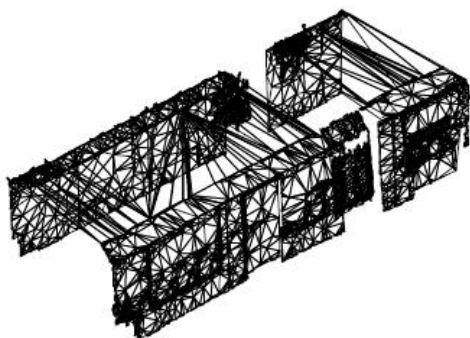


Figura 2.11: Modelo geométrico em malha triangular.

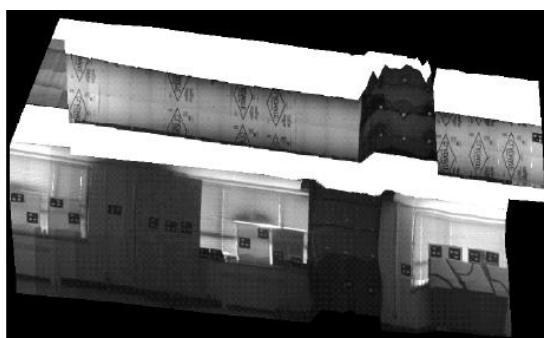


Figura 2.12: Modelo final que incorpora dados geométricos e visuais.

2.2.4 Mapeamento com LRF 3D Estático

Em [31] é usado o sensor estático *Laser Range Finder 3D Cyrax 2500* numa operação de mapeamento que envolveu o registo das estruturas interiores e exteriores da Catedral Beauvais. Este sensor dispensa a utilização de uma plataforma móvel, é sucessivamente colocado manualmente em posições afastadas, para obter nuvens de pontos diferenciadas mas com zonas de sobreposição.

O registo dos vários conjuntos de dados segundo um sistema de coordenadas global desenvolve-se em três etapas. Inicialmente são extraídas rectas a partir das várias nuvens de pontos, que servem para estabelecer correspondências entre pares de conjuntos de pontos que representam parcialmente uma mesma zona, através das correspondências é estimada e aplicada aos conjuntos uma transformação grosseira. O segundo passo consta de um processo de registo global da totalidade dos dados, que tenta alinhar todas as superfícies com base nas informações calculadas sobre os pares de conjuntos no primeiro momento. Finalmente, o algoritmo ICP [22] encarrega-se do refinamento das transformações entre todos os conjuntos de dados.



Figura 2.13: Vista exterior da Catedral de Beauvais.

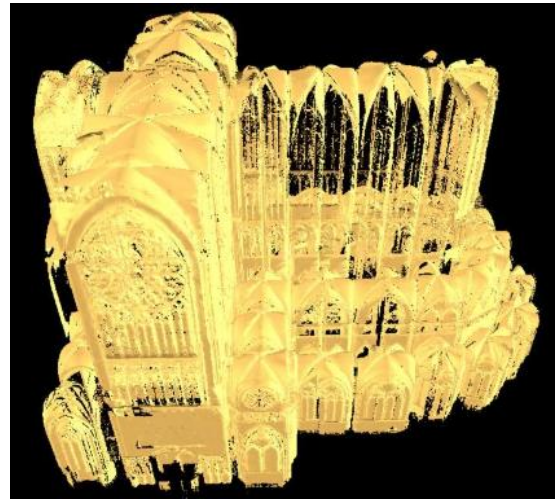


Figura 2.14: Modelo exterior da Catedral de Beauvais.

2.3 Conclusão e Contextualização com a Problemática de Mapeamento e Localização em Túneis

Pelo estado da arte apresentado conclui-se que o número de graus de liberdade dos sensores LRF afecta em grande medida as estratégias utilizadas na construção de modelos tridimensionais. A opção por um LRF 2D com plano de varrimento vertical pressupõe a existência de um processo de localização independente, visto que o conhecimento da localização do sensor é fundamental na etapa de registo dos varrimentos. Neste domínio os sensores LRF 3D ganham vantagem, devido ao facto de num varrimento tridimensional ser possível captar zonas comuns a outros varrimentos, o que permite a implementação de estratégias de registo com base na comparação entre a geometria descrita por estes conjuntos de dados. As transformações calculadas pelo registo dos diversos varrimentos podem ser integradas com o objectivo de representar a trajectória praticada pelo veículo de mapeamento.

No contexto do mapeamento de túneis é previsível a ineficácia da estratégia apresentada para o registo de varrimentos 3D, pois a monotonia estrutural de um túnel implica a ausência de zonas bem diferenciadas, o que gera ambiguidade no processo de estabelecimento de correspondências entre os diversos conjuntos. Todavia, este facto não invalida a utilização de um LRF 3D no levantamento do modelo da estrutura, pois é plausível a aplicação uma estratégia de registo baseada em localização, à semelhança das abordagens com LRF 2D. Como anteriormente referido os sensores de varrimento tridimensional permitem a deslocação do veículo de mapeamento a velocidades superiores.

Comparativamente às extracções mineiras, que possuem paredes extremamente irregulares, os túneis caracterizam-se pela sua forma semi-cilíndrica com superfícies suaves. De novo, a escassez de singularidades estruturais torna inviável a utilização exclusiva de sensores LRF 2D na etapa de localização. Estes podem proporcionar boas informações de distância e orientação relativa entre o sensor e as paredes e o tecto contudo, no sentido do movimento as informações serão praticamente nulas, impedindo a estimação dos estados que se manifestam nesta direcção.

O trabalho [9] recorre a técnicas de visão computacional para determinar a localização, no entanto a colocação de alvos ao longo do percurso e a necessidade de intervenção humana na selecção de correspondências entre imagens tornam este método pouco atractivo. Para além destes factos os percursos apresentados nos testes correspondem a distâncias

percorridas muito inferiores às espectáveis para o sistema abordado nesta tese.

Face às contrariedades mencionadas urge a necessidade de encontrar uma solução de localização capaz de produzir estimativas altamente fiáveis, o que poderá obrigar à concepção de um sistema de fusão multi-sensorial, composto por sensores com características complementares. Estes factos fundamentam o estudo sobre métodos de localização levado a cabo mais à frente no capítulo (5).

Esta página foi intencionalmente deixada em branco.

Capítulo 3

Registo de Varrimentos de Sensores LRF

Conteúdo

3.1	<i>Scan Matching</i> Denso (<i>point-to-point Scan Matching</i>)	24
3.1.1	<i>Iterative Closest Point</i> (ICP)	24
3.1.2	<i>Iterative Dual Correspondence</i> (IDC)	35
3.1.3	<i>Metric-based ICP</i> (MbICP)	37
3.2	<i>Feature-based Scan Matching</i> (FBSM)	39
3.3	Algoritmos que não Estabelecem Correspondências	40

No âmbito do mapeamento robótico os algoritmos de *scan matching* adquirem uma grande importância, na medida em que constituem a resposta ao problema de registo de varrimentos provenientes de sensores LRF, crucial em algumas abordagens de localização e igualmente determinante na agregação de múltiplas imagens de LRF 3D com vista à construção de um único modelo. O processo de *scan matching* pode ser definido como a análise de dois conjuntos de dados – o conjunto alvo e o conjunto de referência ou mapa – adquiridos em posições próximas, com o objectivo de calcular a transformação de corpo rígido que translada os pontos do conjunto alvo para o conjunto de referência.

Tendo em conta que a discrepância entre varrimentos subsequentes é causada pela alteração da posição e atitude do veículo, a transformação de corpo rígido, calculada pela correspondência de varrimentos, fornece uma informação de qualidade sobre o movimento do veículo.

A grande variedade de métodos de *scan matching* é justificada pela adequação deste princípio a uma larga gama de condições operacionais. As características mais distintivas prendem-se com a sequência de evolução do algoritmo (linear ou iterativa), com o tipo de correspondências estabelecidas entre os dois conjuntos de dados e com o método de cálculo através do qual se chega à transformação de corpo rígido [23].

3.1 *Scan Matching* Denso (*point-to-point Scan Matching*)

Os algoritmos de *scan matching* denso são apropriados para qualquer tipo de ambiente, nomeadamente cenários não estruturados, pois utilizam os dados brutos dos sensores sem recorrer à detecção de *features*. Entre as várias abordagens destacam-se dois métodos exaustivamente estudados – ICP [22] (*Iterative Closest Point*) e IDC [32] (*Iterative Dual Correspondence*). Ambos implementam soluções iterativas que se processam em duas fases: numa primeira instância são estabelecidas correspondências entre pontos 3D dos dois conjuntos de dados. Posteriormente é calculada por optimização matemática a transformação de corpo rígido que aproxima os varrimentos. As iterações sucedem-se até que um critério de paragem se verifique [33].

3.1.1 *Iterative Closest Point* (ICP)

O método ICP [22] estabelece as correspondências entre dois conjuntos de N pontos 3D, $\{A_i\}$ e $\{R_i\}$, segundo a regra do ponto mais próximo – *closest-point rule*. Conhecidas as correspondências, $\{p_{a,i}, p_{r,i}\}$, é aplicada uma estratégia de optimização matemática, que minimiza os erros em distância entre os pontos dos varrimentos 3.1, para calcular a transformação de corpo rígido (R, t) .

$$f(R, t) = \frac{1}{N} \sum_{i=1}^N \|p_{a,i} - R(p_{r,i}) - t\|^2 \quad (3.1)$$

No final de cada iteração, a transformação (R_k, t_k) é aplicada aos pontos do varrimento alvo e procede-se à actualização das variáveis de transformação cumulativas (R, t) , que registam o resultado da soma de todas as transformações calculadas ao longo da sequência de iterações. O processo repete-se até à convergência.

A convergência é antecipada através da utilização de uma estimativa inicial fornecida por um sensor de *dead reckoning* – odometreia ou sensores inerciais. Nesta perspectiva, a

estimativa é aplicada aos pontos do varrimento alvo antes da primeira iteração, resultando na aproximação dos varrimentos e conseqüente diminuição das iterações necessárias à estimação de uma transformação precisa [23]. Neste caso, as variáveis de transformação cumulativas devem ser inicializadas com o valor da estimativa inicial.

Nüchter [34] aponta quatro métodos comumente utilizados na implementação do processo de otimização: um método baseado na decomposição em valores singulares de matrizes (SVD) [35], o recurso a quaterniões para representar rotações [36], a representação da totalidade da transformação de corpo rígido por quaterniões [37] e um processo que utiliza matrizes ortogonais para representar rotações [38]. A análise a estas abordagens relatada em [39] não revela diferenças significativas em termos de tempo de processamento, estabilidade e precisão. Todavia, o método SVD salienta-se devido a pequenos ganhos em precisão e estabilidade, mas em contrapartida manifesta um ligeiro aumento de complexidade.

O término de um ciclo de iterações é definido pela satisfação de uma condição de paragem. Essa condição baseia-se normalmente na comparação de aspectos do processo ICP com valores limite definidos à priori, a partir dos quais se considera que a solução encontrada atinge a qualidade desejada. As condições de paragem mais comuns são:

1. Paragem caso a magnitude do incremento relativo em rotação e translação seja inferior a um determinado valor:

$$\frac{R_k}{R} < \varepsilon_{Rr} \quad \cap \quad \frac{t_k}{t} < \varepsilon_{tr} \quad (3.2)$$

2. Paragem caso a magnitude do incremento absoluto em rotação e translação seja inferior a um determinado valor:

$$R_k < \varepsilon_{Ra} \quad \cap \quad t_k < \varepsilon_{ta} \quad (3.3)$$

3. Paragem perante uma variação no erro residual inferior a um dado valor:

$$\sum_{i=1}^N \|p_{a,i} - R(p_{r,i}) - t\|^2 < \varepsilon_{er} \quad (3.4)$$

4. Paragem logo que o número de iterações ultrapasse um limite definido pelo utilizador: $k > k_{máx}$. Esta condição é normalmente conjugada com uma das anteriores através da operação lógica *ou*.

O algoritmo ICP manifesta algumas fragilidades tais como a lenta convergência especialmente face a ambientes curvos, pois as correspondências definidas pela *closest-point rule* originam pouca informação sobre a rotação [32]. Por outro lado a sensibilidade a mínimos locais não assegura a convergência do algoritmo para a solução óptima. Para além disso, o algoritmo assume a existência de uma correspondência para todos os pontos, o que na prática não se verifica devido à possibilidade de existência de zonas sem sobreposição entre os varrimentos. As áreas não comuns podem aparecer devido à movimentação do sensor e conseqüente incidência sobre zonas diferentes do mundo ou à existência de situações de oclusão. Ao longo dos últimos anos o algoritmo ICP tem sido alvo de alguns melhoramentos [23] [40], com ênfase nos aspectos abordados em seguida.

3.1.1.1 Redução da Complexidade Computacional

A complexidade computacional do algoritmo original ICP aumenta quadraticamente em relação ao número de pontos em análise nos dois grupos $O(N_r N_a)$ [41], onde N_r diz respeito ao número de pontos no conjunto de referência e N_a representa a quantidade de pontos do varrimento alvo. Para conjuntos de dados com elevada resolução a complexidade torna-se incomportável e impede a operação em tempo real. Como resposta surgiram várias técnicas de aceleração do algoritmo que, com base no seu fundamento, podem ser divididas em três grupos: redução do número de iterações [42], redução da resolução dos dados [41] ou aceleração do método de pesquisa dos pontos mais próximos [43] [44] [45] [46].

– *Kd-trees*

Um grande número de pesquisas procura o melhoramento do método de pesquisa de correspondências, pois esta constitui a etapa computacionalmente mais complexa do algoritmo ICP [44]. Com este objectivo Besl e McKay [22] sugeriram a utilização de árvores binárias multidimensionais (*kd-trees*) na pesquisa de correspondências. Em [47] é comprovada a diminuição da complexidade para $O(N_r \log N_a)$ no melhor caso.

A *kd-tree*, introduzida por [48], constitui uma estrutura que promove divisões recursivas de um espaço de k dimensões, de modo a organizar pontos nele dispostos. As divisões são estabelecidas por hiperplanos¹ ortogonais a um eixo de coordenadas, em que cada hiperplano intercepta um ponto do conjunto. A partição do espaço não ocorre sempre segundo a mesma dimensão, de forma sequencial ordenada são executadas divisões ortogonais relativamente a todas as dimensões. A raiz da árvore representa a totalidade do conjunto. Todos os nós não terminais, incluindo a raiz, representam uma divisão do conjunto e determinam a divisão da árvore em dois ramos. Por sua vez, os ramos representam os subconjuntos resultantes da partição imposta pelo nó. Os nós terminais não possuem ramos, pois correspondem ao menor subconjunto de dados que é possível formar (1 elemento).

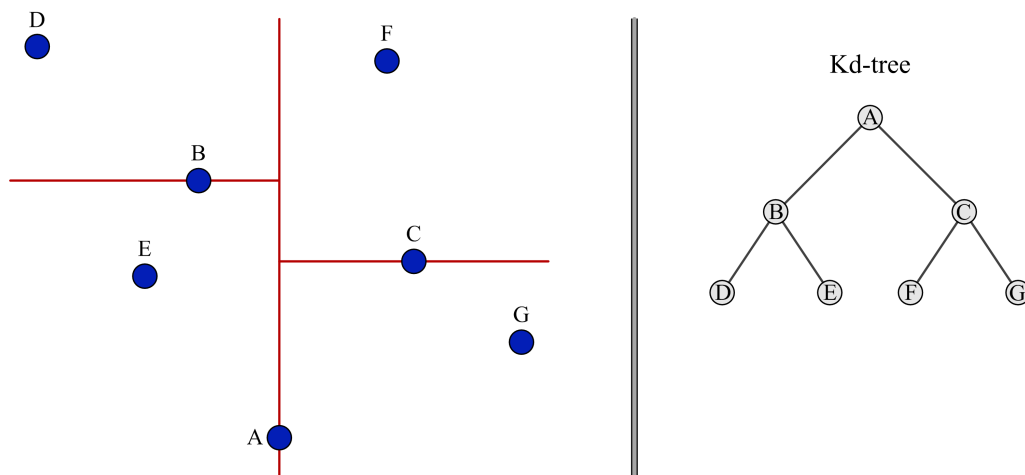


Figura 3.1: Divisão por hiperplanos de um conjunto de pontos 2D e *kd-tree* correspondente.

Em [46] são exploradas os benefícios das *kd-trees* na busca do vizinho mais próximo de um ponto fornecido como referência ao algoritmo. A pesquisa torna-se bastante eficiente, visto que as propriedades de segmentação do espaço promovidas pelas *kd-trees* permitem descartar sucessivamente uma grande percentagem de hipóteses. Com vista à determinação do vizinho mais próximo (pesquisa *Nearest Neighbor*), o algoritmo sugerido em [46] impõe os seguintes passos:

¹Em geometria, um hiperplano de um espaço com dimensão n consiste num sub-espaço de dimensão $n - 1$. Particularizando, no espaço tridimensional os hiperplanos são definidos por planos, enquanto que num espaço bidimensional consistem em rectas

1. O algoritmo foca inicialmente o nó raiz e em seguida percorre a árvore de forma recursiva no sentido descendente. O caminho a tomar em cada nó (esquerda ou direita) depende do grau de superioridade entre o valor do ponto de referência (segundo a dimensão em que ocorre a divisão do espaço pelo nó) e o valor do nó.
2. Quando for atingido um nó terminal, o ponto correspondente é guardado com o estatuto de melhor hipótese.
3. Então, o algoritmo retrocede normalmente por intermédio da arquitectura recursiva, percorrendo a árvore em sentido inverso e executando os seguintes passos em cada nó:
 - (a) O ponto representado pelo nó actual adquire o estatuto de melhor hipótese se a distância entre ele e o ponto de referência for inferior à compreendida entre o ponto de referência e o ponto de melhor hipótese até então.
 - (b) Para verificar a possibilidade de existência de pontos mais próximos localizados do outro lado do hiperplano é testada a intercepção entre este e uma hiperesfera, centrada no ponto de referência e com raio dado pela distância entre o ponto de referência e o ponto de melhor hipótese.
 - i. Se a intercepção se verificar, existe a possibilidade de existirem pontos mais próximos do pontos de referência do outro lado da divisão, logo o algoritmo deve percorrer esse ramo da árvore completando os passos já descritos (passos 1,2 e 3).
 - ii. Caso a intercepção falhe, não existem pontos mais próximos no outro ramo da árvore e o algoritmo passa à análise do nó superior.
4. Após o regresso ao nó raiz e à aplicação do passo 3, a pesquisa está completa e o ponto do conjunto mais próximo do ponto de referência corresponde ao ponto de melhor hipótese.

Sproull em [49] apresenta o conceito *eigen k-d tree*. Segundo a sua estratégia os hiperplanos deixam de ser obrigatoriamente ortogonais aos eixos de coordenadas, passando a ter uma orientação que, de acordo com a disposição dos dados, permita dividir de forma mais eficiente o conjunto facilitando a pesquisa. A orientação do hiperplano de separação é definida como sendo a perpendicular do vector próprio correspondente ao valor próprio

de maior magnitude da matriz de covariância dos dados, ou seja, perpendicular à direcção de maior variância dos dados.

– Algoritmo *STCNN / Kd-tree*

Greenspan e Godin [43] contribuem com uma estratégia para o melhoramento da pesquisa de correspondências, pela conjugação de uma *kd-tree* com a estratégia STCNN (*Spherical Triangle Constraint Nearest Neighbor*). A melhoria proposta encontra fundamento no teorema da convergência [22], que dá conta da redução monotónica da distância entre os varrimentos em cada iteração. De acordo com esta presunção, Greenspan e Godin assumem que existe uma grande probabilidade dos pontos correspondentes se manterem ao longo de várias iterações, por esse motivo as correspondências da iteração anterior são usadas com estimativas na iteração actual.

Numa fase inicial, para cada ponto \vec{q}_i do conjunto alvo $\{A_i\}$ é calculado o ponto mais próximo \vec{p}_i no conjunto de referência $\{R_i\}$ através de um método convencional (ex: *kd-tree*) e é construída uma lista, $L = \{\vec{p}_{viz,1}, \dots, \vec{p}_{viz,k}\}$, que regista os pontos vizinhos até uma distância ε – ε -neighborhood – ordenados de forma crescente em função da distância $\|\vec{p}_i - \vec{p}_{viz,i}\|$.

A partir da primeira iteração a correspondência anterior é considerada como estimativa, \hat{p}_c . O primeiro passo de validação da estimativa corresponde ao teste da restrição esférica – *spherical constraint* – que tem como objectivo verificar se o ponto mais próximo do ponto de referência \vec{q}_i está contido na vizinhança ε de \hat{p}_c . A regra da restrição esférica assume que um ponto $\vec{p}_{viz,i}$, contido na lista de pontos vizinhos L , se encontra mais próximo do ponto de referência \vec{q}_i que a estimativa \hat{p}_c caso $\vec{p}_{viz,i}$ esteja contido na esfera $S_{\hat{p}_c}$, centrada no ponto \hat{p}_c com raio de $2\|\vec{q}_i - \hat{p}_c\|$. Assim, se a inequação $\varepsilon > 2\|\vec{q}_i - \hat{p}_c\|$ se verificar é garantido que o ponto mais próximo de \vec{q}_i corresponde à estimativa ou está contido na lista L , caso contrário a pesquisa não pode ser restringida a estes pontos e é necessário aplicar um método de pesquisa a todos os pontos do conjunto de referência $\{R_i\}$, para determinar a correspondência bem como a respectiva lista de ponto vizinhos.

Uma vez satisfeita a regra da restrição esférica, o ponto mais próximo pode ser escolhido simplesmente pelo cálculo e comparação das distâncias entre os pontos da vizinhança e o ponto alvo \vec{q}_i . Porém, perante uma vizinhança de pontos alargada esta estratégia não é eficiente, sendo aconselhável a aplicação de outra regra de restrição geométrica denominada

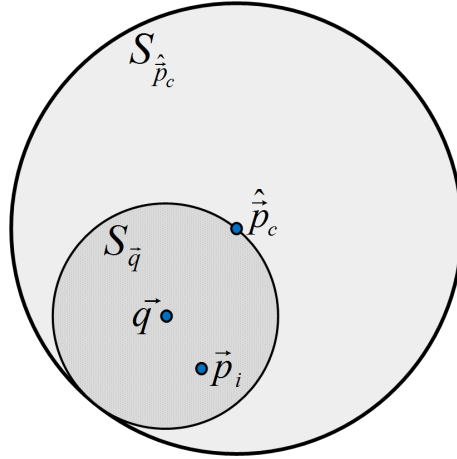


Figura 3.2: Regra da restrição esférica.

restrição triangular – *triangle constraint*. Segundo esta abordagem é criado o ponto \vec{p}_d , cuja função é registar o valor provisório do ponto mais próximo de \vec{q}_i . Inicialmente o ponto \hat{p}_c assume o estatuto de ponto mais próximo de \vec{q}_i e \vec{p}_d toma esse valor, $\vec{p}_d \leftarrow \hat{p}_c$. Em seguida os pontos da lista L são submetidos ao teste da restrição triangular para verificar a possibilidade de algum deles se encontrar mais próximo de \vec{q}_i que \vec{p}_d . O processo incide consecutivamente sobre os elementos da lista L , $\{\vec{p}_{viz,1}, \dots, \vec{p}_{viz,k}\}$, testando a validade da inequação. Sempre que a condição seja respeitada, existe a possibilidade de $\vec{p}_{viz,n}$ estar mais próximo de \vec{q}_i que \vec{p}_d . A confirmação é atingida através da satisfação da condição $\|\vec{q}_i - \vec{p}_{viz,n}\| < \|\vec{q}_i - \vec{p}_d\|$, levando a que o valor de \vec{p}_d seja substituído pelo de $\vec{p}_{viz,n}$. Se para algum elemento a condição $\|\vec{q}_i - \hat{p}_c\| - \|\vec{p}_{viz,n} - \hat{p}_c\| \leq \|\vec{q}_i - \vec{p}_d\|$ não for válida então $\|\vec{q}_i\| \cdot \|\vec{p}_{viz,n} - \hat{p}_c\| > \|\vec{q}_i - \vec{p}_d\|$, logo o ponto $\|\vec{q}_i\| \cdot \|\vec{p}_{viz,n} - \hat{p}_c\| > \|\vec{q}_i - \vec{p}_d\|$ não pode ser considerado mais próximo de \vec{q}_i que \vec{p}_d . Como consequência da ordenação dos pontos da lista de forma crescente relativamente à distância $\|\vec{p}_{viz,i} - \hat{p}_c\|$, todos os pontos com índice superior a $\|\vec{p}_{viz,i} - \hat{p}_c\|$ também violam a regra da restrição triangular e o valor de \vec{p}_d regista definitivamente o ponto mais próximo de \vec{q}_i .

O algoritmo *STCNN / Kd-tree* imita o espaço de pesquisa do ponto mais próximo, proporcionando uma redução drástica da complexidade computacional da etapa de correspondência. Greenspan e Godin [43] nos testes levados a cabo registaram ganhos superiores a 500% relativamente ao método com *kd-trees*.

– *Cached kd-tree*

A natureza recursiva do algoritmo ICP, a par do teorema da convergência referido no ponto anterior, tornam adequado o emprego da estrutura *cached kd-tree* [50] na etapa de determinação do ponto mais próximo. Esta estrutura difere das *kd-trees* originais somente pela inclusão de um apontador suplementar associado a cada nó. O apontador extra é direccionado para o respectivo nó de hierarquia superior, tomando este o valor nulo no nó raiz.

Na primeira iteração a pesquisa de pontos mais próximos segue um algoritmo idêntico ao utilizado nas *kd-trees*. A única excepção reside no facto de o retorno da função comunicar não só o ponto mais próximo, como também o valor do apontador para o nó desse ponto na árvore, sempre que o nó em causa seja terminal. Esta informação adicional é guardada conjuntamente com a correspondência entre pontos, mantendo-se armazenada até que a próxima iteração tire partido dela.

Recordando o algoritmo de pesquisa em *kd-trees*, cada procura inicia-se no nó raiz e procede até ao nó terminal, ocorrendo em seguida o processo de *backtracking*. Nas *cached kd-trees* os apontadores guardados em conjunto com as correspondências da iteração anterior permitem iniciar a pesquisa no nó terminal, não sendo necessário percorrer a árvore no sentido descendente. Deste modo, o algoritmo fica reduzido apenas ao mecanismo de *backtracking*, razão pela qual a complexidade computacional diminui.

Considerando I o número de iterações, a pesquisa em *kd-trees* regista no melhor caso uma complexidade da ordem de $O(I * N_r \log N_r)$, ao passo que a *cached kd-tree* no melhor caso é da ordem $O(N_a * (I + \log N_r))$. A comparação entre os dois métodos levada a cabo em [50] exalta a redução em 50% do tempo de pesquisa em *cached kd-trees*. Porém, há que ter em conta o aumento da memória necessária ao armazenamento dos apontadores adicionais, $O(N_r)$, e do valor dos apontadores que são guardados de iteração para iteração, $O(N_a)$.

– *Variação da Resolução dos Dados*

Em [41] é proposta uma abordagem de aceleração do algoritmo ICP que assenta na variação do nível de representação dos dados. Nas iterações iniciais é executada uma amostragem grosseira dos pontos e à medida que a transformação do corpo rígido é refinada, o volume de dados aumenta de modo a introduzir mais informações no processo.

Segundo este princípio, nas iterações onde é considerado um menor número de pontos, a etapa de pesquisa de correspondências torna-se menos complexa, contribuindo assim para uma redução significativa do tempo de processamento.

Adicionalmente, as iterações menos exigentes originam rotações e translações mais significativas, levando a que a convergência seja atingida mais precocemente quando comparado com o algoritmo ICP original. Este fenómeno traduz-se na redução do número de iterações.

Os resultados atestam a eficiência do algoritmo sem prejuízo ao nível da precisão do resultado. Os ganhos em velocidade são consideráveis, até 27 vezes mais rápido que o método ICP com *kd-tree* e 1300 mais célere que o algoritmo ICP original.

– *Accelerated ICP*

A par da formulação do algoritmo ICP, Besl e McKay [22] apresentam a versão *Accelerated ICP*, que atinge melhor desempenho do ponto de vista computacional, por meio da redução do número de iterações. O *Accelerated ICP* introduz uma etapa extra responsável pela comparação das transformações (R, t) calculadas nas três iterações mais recentes. Caso as transformações em causa manifestem uma orientação aproximada, é possível extrapolar uma transformação a partir do escalonamento da transformação corrente, em que o factor de escala é uma função do erro médio quadrado e da magnitude das transformações incrementais das três iterações anteriores.

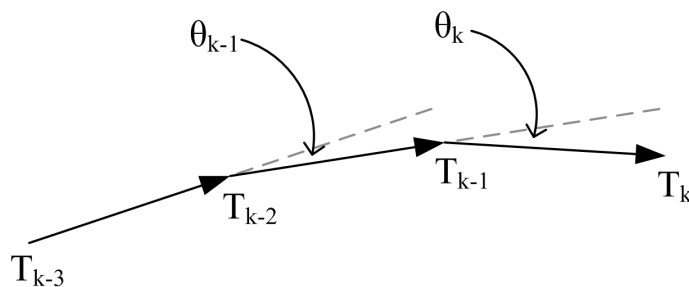


Figura 3.3: Orientação das transformações ao longo de três iterações.

Para que a extrapolação não degrade a qualidade do método ICP é essencial um bom alinhamento entre a direcção das transformações ao longo das últimas três iterações, como indicado em (3.5). O desalinhamento máximo permitido em [22] é $\delta\theta = 10^\circ$.

$$\theta_k < \delta\theta \quad \wedge \quad \theta_{k-1} < \delta\theta \quad (3.5)$$

O erro médio quadrado nas iterações é representado por d_k , d_{k-1} e d_{k-2} e são definidos três argumentos v_k , v_{k-1} e v_{k-2} , que serão utilizados como parâmetros na extrapolação, de tal forma que:

$$v_k = 0, \quad v_{k-1} = -\|\Delta \vec{q}_k\|, \quad v_{k-2} = -\|\Delta \vec{q}_{k-1}\| + v_{k-1} \quad (3.6)$$

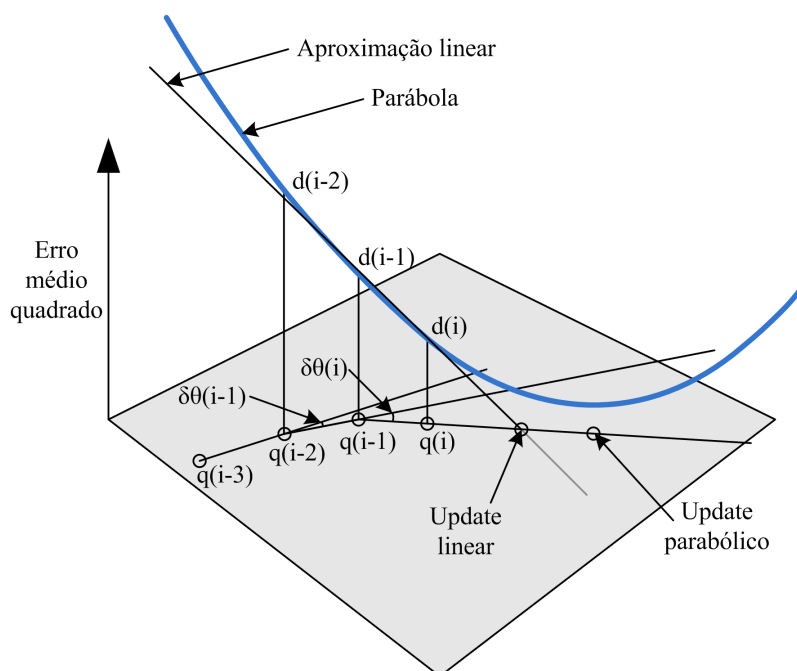


Figura 3.4: Representação da aproximação linear e da interpolação parabólica.

Com base nestes três pontos é calculada uma aproximação linear, $d_1(v)$, e uma interpolação parabólica, $d_2(v)$:

$$d_1(v) = a_1(v) - b_1, \quad d_2(v) = a_2(v)^2 + b_2(v) + c_2 \quad (3.7)$$

O ponto resultante da intersecção da aproximação linear com o plano proporciona um possível valor para a extrapolação designado *update linear* (3.8). Por sua vez, a determinação do extremo da parábola dá origem ao valor do *update parabólico* (3.9).

$$v_1 = -\frac{b_1}{a_1} \quad (3.8)$$

$$v_2 = -\frac{b_2}{2a_2} \quad (3.9)$$

É definido um valor máximo para a transformação resultante da extrapolação, $v_{máx}$, sendo sugerido em [22] $v_{máx} = 25\|\Delta \vec{q}_k\|$. O *update* empregue na extrapolação é seleccionado com base nas condições seguintes:

1. Se $0 < v_2 < v_1 < v_{máx}$ ou $0 < v_2 < v_{máx} < v_1$ é usado o *update* parabólico. A transformação incremental é dada pela fórmula:

$$\vec{q}_k = \vec{q}_k + \frac{v_2 \Delta \vec{q}_k}{\|\Delta \vec{q}_k\|} \quad (3.10)$$

2. Se $0 < v_1 < v_2 < v_{máx}$ ou $0 < v_1 < v_{máx} < v_2$ é usado o *update* linear. A transformação incremental é dada pela fórmula (3.11).

$$\vec{q}_k = \vec{q}_k + \frac{v_1 \Delta \vec{q}_k}{\|\Delta \vec{q}_k\|} \quad (3.11)$$

3. Se $v_1 > v_{máx}$ e $v_2 > v_{máx}$ é usada a transformação máxima. A transformação incremental é dada pela fórmula (3.12).

$$\vec{q}_k = \vec{q}_k + \frac{v_{máx} \Delta \vec{q}_k}{\|\Delta \vec{q}_k\|} \quad (3.12)$$

A implementação desta abordagem é bastante vantajosa, pois um ciclo de 15 a 20 iterações do algoritmo *Accelerated ICP* pode equivaler a mais de 50 iterações do algoritmo original.

3.1.1.2 Convergência para o Mínimo Global

Apesar do algoritmo ICP convergir monotonicamente para um mínimo local, qualquer que seja a rotação e translação entre conjuntos de pontos, existe a possibilidade de esse mínimo local não corresponder ao mínimo global [22].

A medida mais eficaz para forçar a convergência no mínimo global consiste na inicialização da transformação por intermédio de uma estimativa de boa qualidade [33] [51].

A aplicação da estimativa ao varrimento alvo aproxima as duas superfícies e guia implicitamente o algoritmo para a zona do mínimo global, promovendo a convergência neste ponto.

A existência de pontos não comuns em dois conjuntos de dados, *outliers*, pode também inviabilizar a convergência no mínimo global. Para eliminar esta fonte de erro é usual implementar um método de filtragem de falsas correspondências antes do cálculo das transformações.

3.1.1.3 Eliminação de *Outliers*

São designados por *outliers* os pontos de um varrimento que incidem sobre uma região do espaço não representada pelo outro conjunto de pontos. A consideração de *outliers* no cálculo da transformação de corpo rígido pode ser significativamente prejudicial, pois a correspondência errónea atribuída a estes pontos na etapa de correspondência introduz informações incorrectas no processo.

A influência de *outliers* é minorada pela implementação de uma estratégia de filtragem das correspondências com qualidade duvidosa. A metodologia mais comum passa pela rejeição de pontos correspondentes cuja distância relativa seja superior a um dado valor. Este valor deve ser dinâmico, de modo a adaptar-se à tendência de diminuição da distância entre os conjuntos de pontos ao longo das iterações [29]. Pulli em [29] propõe a combinação da estratégia anterior com a eliminação das 10% piores correspondências segundo uma determinada métrica, usualmente a distância entre pontos. Já em [52] são dados como incongruentes os pares cuja distância relativa excede um valor múltiplo do desvio padrão dos resíduos estimados.

3.1.2 *Iterative Dual Correspondence (IDC)*

Lu e Milos tornaram-se pioneiros na área de *scan matching* quando no ano de 1994 introduziram o algoritmo IDC (*Iterative Dual Correspondence*) [32]. A motivação principal por detrás deste trabalho prende-se com a lenta convergência do predecessor ICP, especialmente perante superfícies com curvaturas acentuadas, facto que é justificado pela pouca informação sobre rotação fornecida pelas correspondências determinadas segundo a *closest-point rule*.

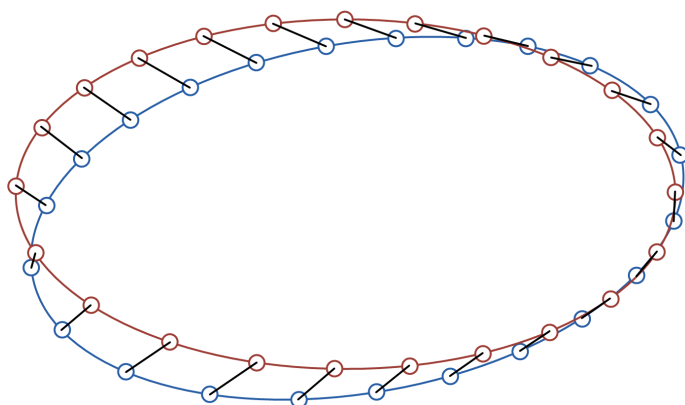


Figura 3.5: Correspondências respeitando a *closest-point rule* não facultam boas informações sobre rotação, a inconsistência na orientação dos segmentos que unem os pontos correspondentes comprova esta afirmação.

Como forma de contornar esta fragilidade os autores idealizaram uma nova métrica denominada *matching-range-point rule*, através da qual se adquirem correspondências que revelam de forma significativa a componente rotacional. A ideia fulcral da *matching-range-point rule* é a de que a translação foi previamente calculada e a principal preocupação é a determinação da orientação. Relegando para segundo plano a translação, a *matching-range-point rule* selecciona o par p_i do ponto q_i com base na semelhança das distâncias polares entre os pontos e a origem dada pela posição do sensor. É ainda definido um limite máximo angular B_ω para o estabelecimento da correspondência, facto que limita o erro rotacional máximo que o método é capaz de corrigir em cada iteração, prevenindo no entanto falsas correspondências entre pontos não relacionados.

O algoritmo IDC combina as métricas *closest-point rule* e *matching-range-point rule*, usufruindo deste modo das boas prestações da primeira no cálculo da translação e empregando a segunda na determinação da rotação. A conjugação das duas regras traduz-se na antecipação da convergência e no aumento da estabilidade relativamente a diferenças de orientação consideráveis entre os dois conjuntos de pontos. O algoritmo IDC desenrola-se de forma iterativa, em que cada ciclo contempla a seguinte ordem de tarefas:

1. Para cada ponto q_i do varrimento alvo,
 - (a) Aplicar a *closest-point rule* para determinar a correspondência p'_i .
 - (b) Recorrer à *matching-range-point rule* para seleccionar a correspondência p''_i .

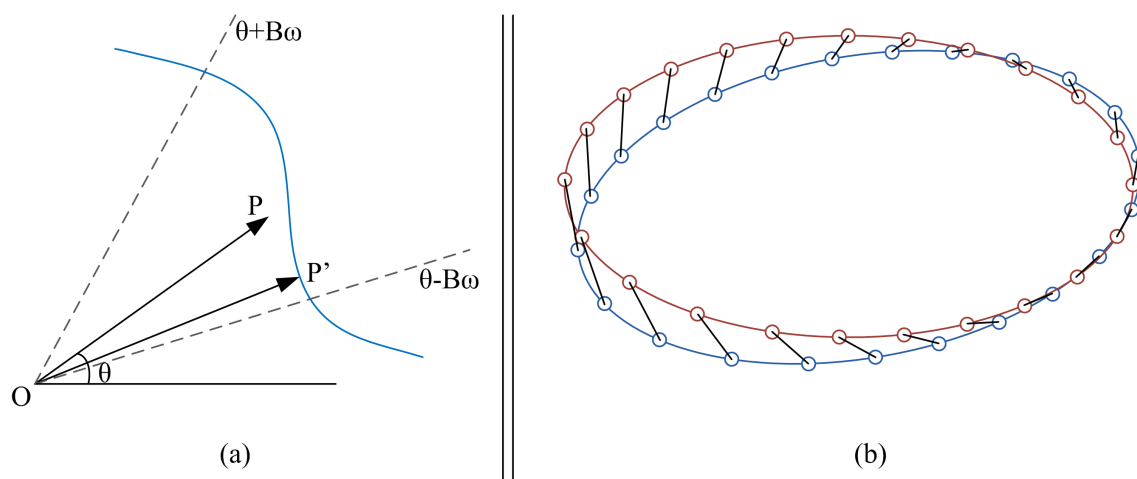


Figura 3.6: Matching-range-point rule. (a) Representação geométrica da métrica. (b) Correspondências obtidas pela aplicação da regra. A informação sobre rotação é perceptível pela orientação consistente dos segmentos que unem os pares.

2. Calcular a solução de mínimos quadrados (R_1, t_1) através das correspondências obtidas pela *closest-point rule*.
3. Calcular a solução de mínimos quadrados (R_2, t_2) através das correspondências obtidas pela *matching-range-point rule*.
4. Construir a transformação (R_2, t_1) , aplica-la aos pontos do conjunto alvo e integração das variáveis de transformação cumulativas.

O método ICP é bastante popular devido à qualidade dos resultados conseguidos [53]. Contudo, a separação entre o cálculo da rotação e translação leva a que estas duas componentes não se encontrem directamente relacionadas, podendo deteriorar os resultados. Por outro lado, apesar da redução do número de iterações resultante da antecipação da convergência, a necessidade de produzir dois tipos de correspondências em cada iteração eleva a complexidade computacional do algoritmo face ao ICP.

3.1.3 Metric-based ICP (MbICP)

O algoritmo MbICP [54] contém a mesma estrutura do algoritmo ICP, porém introduz uma regra de correspondência inovadora, que assimila simultaneamente informações sobre translação e rotação, colmatando deste modo a maior falha do algoritmo IDC. Como anteriormente referido a regra *closest-point rule* baseada apenas na distância euclidiana

apresenta um fraco desempenho na estimação de erros rotacionais. A métrica empregue no MbICP representa uma medida de distância no plano, definida como a norma mínima da transformação de corpo rígido que sobreposição dois pontos. Ao considerar a totalidade da transformação, a métrica agrega um membro relativo à translação e uma componente rotacional

Uma transformação de corpo rígido 2D pode ser definida por um vector $q = (x, y, \theta)$. Em [54] considera-se que a norma da transformação q é:

$$\|q\| = \sqrt{x^2 + y^2 + L^2\theta^2} \quad (3.13)$$

- Onde o parâmetro L constitui um número real que indica a quantidade da componente rotacional a considerar.

Dados dois pontos $p_1 = (p_{1x}, p_{1y})$ e $p_2 = (p_{2x}, p_{2y})$, a distância entre eles é definida como:

$$d_p(p_1, p_2) = \min\{\|q\| \mid q(p_1) = p_2\} \quad (3.14)$$

Sendo:

$$q(p_1) = \begin{pmatrix} x + p_{1x} \cos \theta - p_{1y} \sin \theta \\ y + p_{1x} \sin \theta - p_{1y} \cos \theta \end{pmatrix}$$

Infelizmente não existe uma expressão analítica para o cálculo da distância (3.14), todavia é apresentada uma aproximação (3.15), válida quando a norma mínima da translação é pequena.

$$d_p^{ap}(p_1, p_2) = \sqrt{\delta_x^2 + \delta_y^2 - \frac{(\delta_x p_{1y} - \delta_y p_{1x})^2}{p_{1y}^2 + p_{1x}^2 + L^2}} \quad (3.15)$$

Com $\delta_x = p_{2x} - p_{1x}$ e $\delta_y = p_{2y} - p_{1y}$.

Na generalidade dos testes executados em [55] o MbICP produz correspondências com grande qualidade, atingindo uma percentagem de associações falsas inferior a 1%. O número de iterações necessário para atingir a convergência é sensivelmente idêntico ao do IDC, com vantagem para este último. O algoritmo ICP converge nitidamente mais tarde, revelando um acréscimo médio de iterações superior a 1/3 relativamente ao MbICP e ao IDC. Por outro lado o algoritmo IDC, pela necessidade de cálculo de dois conjuntos de

correspondências, regista o pior desempenho computacional. Neste campo o MbICP é líder, todavia o método ICP produz tempos de resposta aproximados.

3.2 *Feature-based Scan Matching (FBSM)*

Perante um cenário estruturado onde a identificação de *features* (por exemplo cantos ou superfícies planas) seja viável [56], poderá ser benéfico recorrer a estas particularidades no processo de comparação dos varrimentos, devido ao ganho computacional que advém da utilização de um menor número de informações.

Em [57] os pontos obtidos pelo sensor são comparados com rectas definidas num modelo fornecido à priori, segundo o método do vizinho mais próximo (*Nearest Neighbor*). Com base nas correspondências estabelecidas e nos dados de *dead reckoning* adquiridos por odometria, são calculadas uma rotação e uma translação, que resultam da minimização da distância total entre os pontos e as linhas a eles associadas. A transformação é aplicada aos pontos e o processo repete-se até que os resultados apresentem a qualidade desejada.

O projecto AMOS [58] implementa uma melhoria do método anterior, pois as linhas são extraídas a partir de uma sequência de varrimentos, não sendo necessário possuir conhecimento prévio do ambiente.

Mais recentemente o algoritmo HAYAI (*High-Speed and Yet Accurate Indoor/Outdoor Tracking*) [59] apresenta uma performance notável quer em ambiente *indoor* ou *outdoor*. Foi desenvolvido para operar em robôs rápidos com velocidade máxima de 4 m/s, o que obrigou ao esforço bem sucedido de redução da complexidade computacional, de modo a atingir respostas em tempo real. A determinação das transformações entre dois varrimentos resulta da minimização do erro em distância entre pares de *features*. A robustez do método é assegurada pela utilização de *features* invariantes a rotações e translações e pela aplicação de métodos de filtragem de ruído das medidas.

O estudo [60] oferece uma melhoria ao nível do algoritmo de extracção de *features*. Com base na análise das várias fontes de ruído é atribuído um valor de incerteza a cada feature, que será utilizado tanto no processo de correspondência como na etapa de estimação da transformação.

3.3 Algoritmos que não Estabelecem Correspondências

Esta secção reporta alguns algoritmos que exploram um paradigma distinto, nestes casos não são estabelecidas correspondências directas entre pontos ou features. Ao evitar o problema de associação de dados estas abordagens eliminam a etapa que normalmente introduz os erros mais significativos nos processos de *scan matching* [61].

Weiss e Puttkammer [62] recorrem a histogramas para estabelecer a relação entre dois varrimentos 2D, que resulta da determinação do máximo da função de correlação cruzada. Em primeiro lugar é construído o histograma angular com o objectivo de estimar a rotação entre os dois conjuntos de dados. O cálculo da translação assenta sobre a análise dos histogramas segundo a direcção x e y, que são elaborados após a determinação e aplicação da direcção mais provável. A precisão e tempo de processamento atingidos por este algoritmo dependem da escala de discretização dos histogramas. A obtenção de precisões adequadas ao processo de mapeamento acarreta um elevado dispêndio computacional [59].

Biber e Straßer [14] apresentam o método iterativo NDT (*The Normal Distributions Transform*). Através da divisão do espaço, por células em torno do sensor, alcança-se uma função definida por ramos diferenciável, que representa o espaço 2D sob a forma de uma distribuição de probabilidade. Uma estimativa da movimentação do sensor é obtida pela minimização da função através do método de Newton, que requer o cálculo da primeira e segunda derivadas. Tarefas como a rejeição de outliers e o ajuste do tamanho das células requerem uma calibração minuciosa, o que torna este método pouco adequado à maioria das aplicações de *scan matching*. Em contrapartida, uma boa afinação dos parâmetros e a implementação de melhorias no algoritmo de Newton proporcionam uma performance acima da média [53].

A eficiência do algoritmo é comprovada por Biber e Straßer. Nos testes efectuados, um processador a 1.4GHz processa 97 varrimentos por segundo.

Capítulo 4

Fundamentos de Visão Computacional

Conteúdo

4.1	Extracção de <i>Features</i>	42
4.1.1	Detector de Harris	43
4.1.2	<i>Scale-Invariant Feature Transform</i> (SIFT)	47
4.2	Correspondência entre Imagens	55
4.2.1	Comparação de Descritores do Tipo SIFT	55
4.2.2	<i>Template Matching</i>	55
4.3	Projectção Perspectiva	57
4.4	Distorção Provocada pela Lente	60
4.5	Visão <i>Stereo</i>	62
4.5.1	Geometria Epipolar	63
4.5.2	Matriz de Homografia	65
4.5.3	Rectificação <i>Stereo</i>	66
4.6	RANSAC	68
4.6.1	<i>1-Point</i> RANSAC	70
4.7	<i>Bundle Adjustment</i>	72

A visão animal é um sentido bastante poderoso, na medida em que permite obter um vasto leque de informações sobre o ambiente circundante sem ser necessário efectuar qualquer tipo de contacto físico. Através dela é possível identificar e deduzir a posição de

objectos, formular relações entre eles e interagir com o mundo de forma inteligente, para além de constituir o sentido que serve de suporte aos mecanismos de navegação.

Com o intuito de dotar as máquinas destas capacidades, ao longo dos últimos trinta anos, a problemática de visão computacional tem sido estudada com afinco, de tal modo que actualmente existem inúmeras soluções comerciais para aplicações que vão desde o controlo de qualidade em linhas de produção, até aparelhos de apoio ao diagnóstico médico. Na área da robótica móvel é empregue na determinação da localização e movimentação dos robôs, na detecção e identificação de obstáculos estáticos ou móveis, na construção de mapas do ambiente e em mecanismos de aprendizagem.

O processo inicia-se com a aquisição de uma ou mais imagens de intensidade, e em seguida podem ser tomadas medidas para a redução de ruído, após as quais a imagem estará em condições de ser analisada. O estudo da imagem consiste na tentativa de identificação de pontos de interesse ou de conjuntos de píxeis com as mesmas características, que de alguma forma se relacionam com propriedades de objectos ou padrões observados no mundo em três dimensões.

4.1 Extracção de *Features*

No âmbito da visão computacional a designação de *feature* é atribuída a uma parcela da imagem que representa uma determinada particularidade de interesse. Como *feature* podem considerar-se píxeis com intensidade díspar relativamente à vizinhança, cantos (*corners*), zonas de fronteira (*edges*), linhas e curvas. Nos processos de localização autónoma as *features* mais utilizadas são cantos, porém também é válido o emprego de *edges*, linhas e pontos com intensidade dissonante.

A extracção de *features* é uma tarefa de processamento de imagem de baixo nível, que opera de forma densa sobre cada píxel da imagem, de modo a testar a existência de uma particularidade nesse ponto. A repetibilidade é das propriedades mais importantes nos algoritmos de detecção de *features*, pois a maior parte das aplicações de visão computacional depende da identificação das mesmas *features* ao longo de sequências de imagens. Neste contexto, os algoritmos de extracção de *features* mais populares desenvolvem técnicas para fazer face a alterações de perspectiva, iluminação, escala e rotação da imagem provocadas pela movimentação da câmara, de modo que estas ocorrências não se reflectam na capacidade de detecção. Em contrapartida, ao caminhar para a solução mais robusta

manifesta-se o aumento dos requisitos computacionais, factor também determinante na escolha de um algoritmo de extracção de *features*.

A maioria dos algoritmos de detecção de *features* operam em imagens em escala de cinza. Ao não trabalhar sobre imagens a cores os algoritmos tornam-se mais simples e computacionalmente menos exaustivos.

Em seguida são explanados dois algoritmos adequados a aplicações de localização autónoma.

4.1.1 Detector de Harris

O algoritmo de detecção de Harris [63], normalmente referido como um detector de cantos, possibilita a extracção de cantos e de *edges*, embora a estratégia de detecção de cantos seja no fundo sensível a pontos de interesse. A categoria dos pontos de interesse abrange cantos, pontos extremos de linhas, pontos isolados com intensidade local discrepante e pontos de uma curva onde a curvatura é máxima. Na prática, a maior parte dos métodos de detecção de cantos consiste em detectores de pontos de interesse, como consequência, nas situações em que seja necessário marcar apenas os cantos é necessária uma análise local mais exaustiva.

O método de Harris é um melhoramento do proposto por Moravec [64], este com carências ao nível da imunidade ao ruído, da sensibilidade excessiva a *edges* e da influência da orientação da imagem na detecção dos pontos. O método de Harris consiste na análise da energia em cada ponto da imagem, através da determinação do grau de correlação entre o píxel e uma janela que se desloca na sua vizinhança.

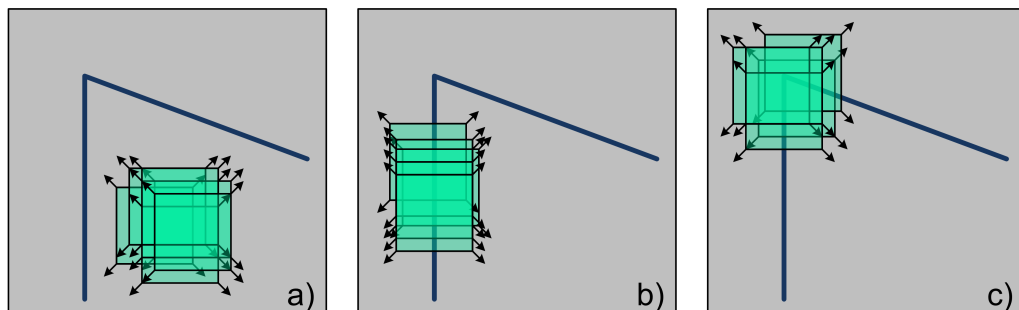


Figura 4.1: Deslocação da janela no método de Harris. a) Deslocação da janela numa área uniforme produz pouca energia. b) Variação da energia segundo uma direcção indica uma zona de fronteira. c) Variação em todas as direcções revela um ponto de interesse.

A energia é dada pela fórmula:

$$E(x, y) = \sum_{u,v} w_{u,v} (I(x+u, y+v) - I(x, y))^2 \quad (4.1)$$

- Onde (u, v) dizem respeito às coordenadas dos pontos da vizinhança, (x, y) indicam o ponto da imagem em análise, E a energia, I a intensidade dos píxeis da imagem e w a função respeitada na geração dos valores da janela

Para que o processo seja menos sensível à rotação dos cantos a função w deve corresponder a uma gaussiana circular, obedecendo à seguinte expressão:

$$w(u, v) = e^{-\frac{u^2+v^2}{2\sigma^2}} \quad (4.2)$$

- Onde: $u^2 + v^2 \leq s^2$ e s define o raio.

$$E(u, v) \cong [u, v] M \begin{bmatrix} u \\ v \end{bmatrix} \quad (4.3)$$

Para pequenos deslocamentos $[u, v]$ a expressão da energia pode ser aproximada recorrendo à expansão de Taylor de primeira ordem, resultando:

- Onde $M = \begin{bmatrix} a & c \\ c & b \end{bmatrix}$. Sendo: $a = \left(\frac{\delta I}{\delta x}\right)^2 \otimes w$, $b = \left(\frac{\delta I}{\delta y}\right)^2 \otimes w$, $c = \frac{\delta I}{\delta y} \frac{\delta I}{\delta x} \otimes w$

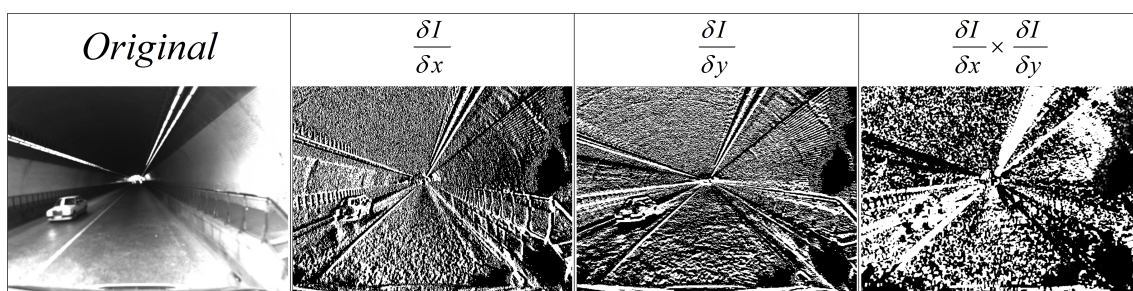


Figura 4.2: Aspecto das derivadas da imagem segundo x e y e resultado da sua multiplicação.

A matriz simétrica M constitui uma medida da covariância da intensidade do gradiente, suavizada pela convolução com a gaussiana. A covariância pode ser representada sob a forma de uma elipse, onde os valores próprios se relacionam com o alongamento dos braços da elipse.

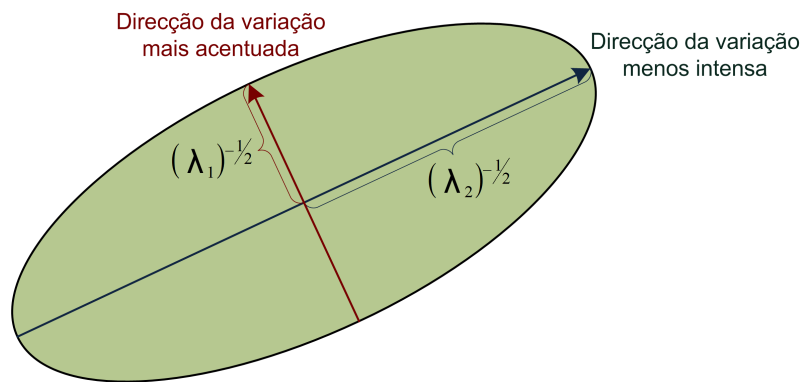


Figura 4.3: Elipse de covariância.

A classificação do ponto é ditada pelos valores próprios da matriz M , podendo ocorrer três situações distintas:

- λ_1 e λ_2 são pequenos – E é praticamente constante em todas as direcções e estamos perante uma zona uniforme;
- $\lambda_1 \ll \lambda_2$ ou $\lambda_1 \gg \lambda_2$ – indicam um *edge*;
- λ_1 e λ_2 são valores elevados – existe um ponto de interesse.

O formato da elipse relaciona-se com o tipo de *feature* do ponto, como se mostra na figura (4.4).

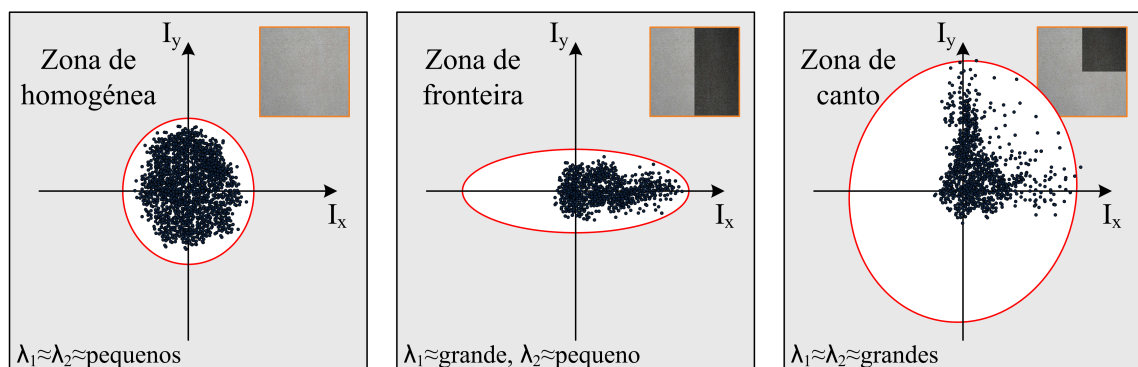


Figura 4.4: Relação entre a forma da elipse e a *feature* em causa.

Outro método de análise baseia-se no cálculo da amplitude da função de resposta R para cada ponto, obtido pela aplicação da fórmula:

$$R = \det(M) - k(\text{traço}(M))^2 \quad (4.4)$$

- Onde:

$$\det(M) = \lambda_1 \lambda_2 = ab - c^2 \quad (4.5)$$

$$\text{traço}(M) = \lambda_1 + \lambda_2 = a + b \quad (4.6)$$

- Com $k = 0.04$, segundo sugerido em [63].

Os máximos locais de R correspondem aos pontos de interesse. No entanto, é usualmente definido um valor mínimo (*threshold*), abaixo do qual os pontos de interesse são desprezados. Desta forma reduz-se o número de pontos, filtrando os de menor intensidade.

A análise pode ser feita graficamente pela divisão do espaço dos valores próprios, como espelha a figura (4.5).

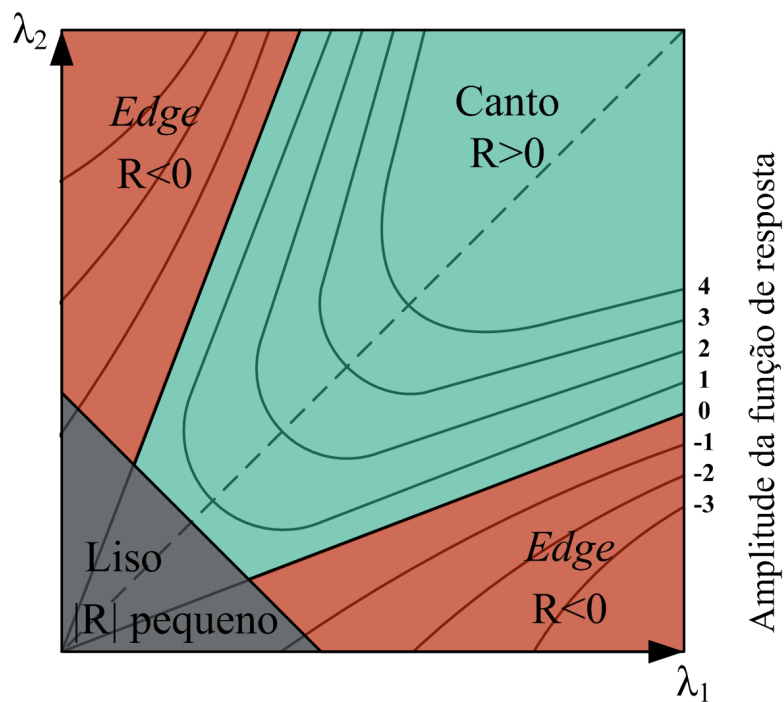


Figura 4.5: Divisão do espaço dos valores próprios para diferenciação entre pontos de interesse e fronteiras.

– **Propriedades**

- Pelo facto de serem utilizadas derivadas na composição da matriz M o método torna-se bastante insensível a alterações de intensidade globais, pois só as diferenças entre intensidades dos pontos são consideradas.
- Muitas das referências ao método de Harris referem-no erradamente como sendo isotrópico, ou seja imune a variações de rotação da imagem. O cálculo da matriz M é feito com base nos gradientes horizontais e verticais, levando a que a orientação dos pontos de interesse tenha impacto na qualidade da detecção. Este problema é mais significativo para rotações de $45^\circ + k\frac{\pi}{2}$.
- Por último, pode constatar-se que a avaliação do tipo de *feature* é influenciada pela relação entre o seu tamanho e a dimensão da janela. Significa que o processo não é invariante a alterações na escala da imagem.

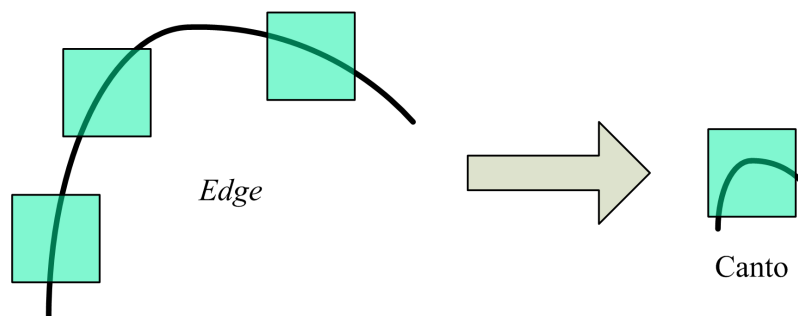


Figura 4.6: Efeito da variação de escala da imagem na classificação da feature.

4.1.2 *Scale-Invariant Feature Transform (SIFT)*

O algoritmo de visão computacional SIFT desenvolvido por David Lowe [65] permite atingir um elevado grau de repetibilidade na detecção de pontos de interesse, visto só serem seleccionados aqueles que apresentam características invariantes relativamente a alterações de escala e rotação. Adicionalmente, este método oferece alguma imunidade à presença de ruído na imagem e a variações de iluminação, associa ainda a cada ponto de interesse um descritor cuja informação caracteriza não só o ponto como os píxeis na sua vizinhança. O descritor partilha as mesmas características de invariância a alterações de

escala e rotação e contém informação altamente distintiva, com base na qual é possível estabelecer correspondências entre pontos comuns em várias imagens.

De forma a otimizar o desempenho computacional, é adoptada uma abordagem de filtragem em cascata, onde os pontos em análise são submetidos a testes de validação antes da execução das tarefas mais dispendiosas computacionalmente. O algoritmo SIFT pode ser seccionado em quatro etapas:

- Detecção de extremos ao longo do espaço de escala;
- Obtenção da localização dos pontos;
- Determinação da orientação;
- Geração dos descritores.

4.1.2.1 Detecção de Extremos ao longo do Espaço de Escala

A primeira fase de selecção foca a robustez da detecção dos pontos face a alterações na escala da imagem. A pesquisa baseia-se no estudo da estabilidade dos pontos ao longo de uma gama de escalas, utilizando uma função de escala contínua denominada espaço de escala. Esta operação tem como objectivo a eliminação progressiva do detalhe da imagem, para assim simular as variações de escala dos objectos. De acordo com os trabalhos de Koenderink [66] e Lindeberg [67] o *kernel*¹ de espaço de escala utilizado corresponde a uma função gaussiana, uma vez que esta minimiza a deturpação da imagem e consequente introdução de falsos detalhes.

Assim, o espaço de escala, $L(x, y, \sigma)$, de uma imagem é obtido através da convolução do kernel gaussiano de escala variável, $G(x, y, \sigma)$, com a imagem de entrada, $I(x, y)$, onde σ representa o grau de suavização:

$$L(x, y, \sigma) = G(x, y, \sigma) \otimes I(x, y) \quad (4.7)$$

O operador \otimes indica a operação de convolução segundo x e y , enquanto que o *kernel* gaussiano $G(x, y, \sigma)$ é dado por:

$$G(x, y, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)} \quad (4.8)$$

¹Em visão computacional a grelha que define o pesos dos píxeis num filtro linear é designada por *kernel* [68].



Figura 4.7: Exemplo de um espaço de escala.

Na figura (4.7) é representado o exemplo de um espaço de escala, resultante da sucessiva convolução da imagem original com o *kernel* gaussiano. Após esta operação a imagem de entrada é amostrada por um factor múltiplo de 2 originando imagens de menor dimensão. A elas aplica-se o mesmo método de filtragem com o propósito de produzir uma cascata estruturalmente semelhante à exibida na figura (4.8).

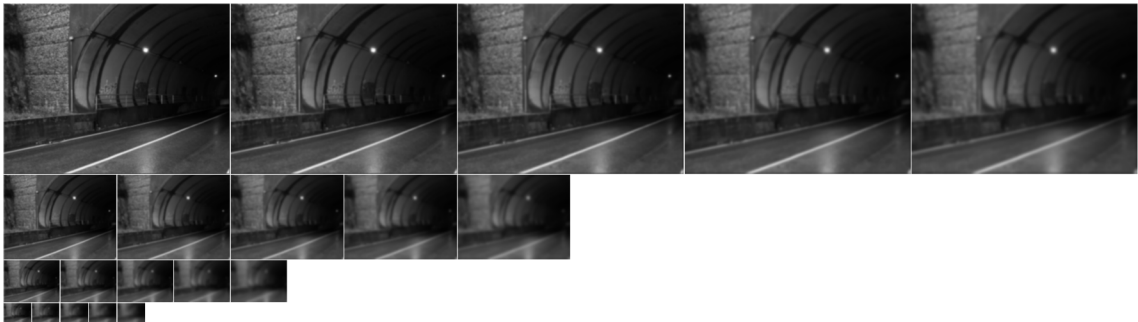


Figura 4.8: Representação de quatro oitavas e espaços de escala correspondentes.

As imagens de igual dimensão formam grupos designados por oitavas. O algoritmo SIFT prevê a construção de quatro oitavas, compostas por espaços de escala de cinco imagens, contudo estas dimensões podem sofrer ligeiros desvios de modo a adequar o algoritmo à resolução das imagens e ao número de pontos de interesse desejados.

O Laplaciano de Gaussianas (LoG) é uma medida isotrópica² resultante das derivadas de segunda ordem de uma imagem. O LoG de uma imagem realça regiões com variação abrupta de intensidade, essa propriedade justifica a sua aplicação em algoritmos de detecção de *edges* e cantos, como é o caso do detector de Harris. Porém a elevada sensibilidade das segundas derivadas ao ruído constitui um grande inconveniente, de modo que

²No contexto do processamento de imagem a propriedade de isotropia indica a aplicação equitativa de uma determinada operação em todas as direcções, sem existir qualquer sensibilidade ou tendência face à orientação da imagem. Um bom exemplo de uma operação isotrópica é a suavização de imagens pela convolução com um kernel gaussiano. Contudo é importante frisar que a implementação de uma operação teoricamente isotrópica pode não replicar de forma absoluta esta propriedade, pois a disposição dos píxeis segundo uma grelha discreta inviabiliza operações no espaço contínuo.

habitualmente a imagem é submetida a um processo de suavização gaussiano para garantir a estabilidade do método.

O cálculo de derivadas de segunda ordem obriga a grande dispêndio computacional, por isso o algoritmo SIFT, usufruindo do espaço de escala, substitui o Laplaciano de Gaussianas pela equivalente Diferença de Gaussianas (DoG), $D(x, y, \sigma)$.

$$D(x, y, \sigma) = ((G(x, y, k\sigma) - G(x, y, \sigma)) \otimes I(x, y)) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (4.9)$$

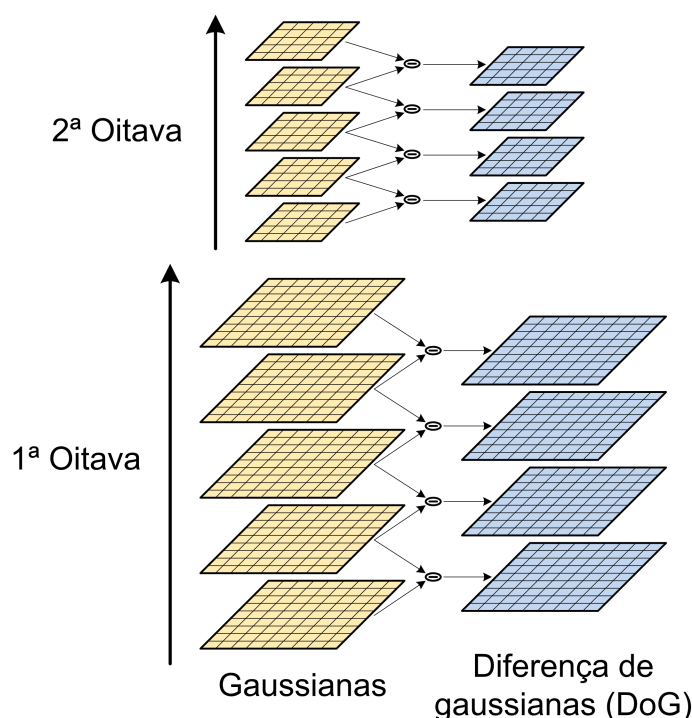


Figura 4.9: Cálculo da diferença de gaussianas.

A Diferença de Gaussianas executa simplesmente a subtração entre imagem de duas escalas consecutivas (figura 4.9), atingindo um patamar de eficiência computacional superior ao Laplaciano de Gaussianas e preservando a propriedade de invariância relativamente a alterações de escala.

A detecção dos extremos locais, nas imagens provenientes das diferenças de gaussianas, é feita comparando cada ponto com os oito vizinhos mais próximos na mesma imagem e com os pontos das imagens adjacentes no espaço de escala, abrangidos por uma janela de 3×3 com centro no ponto em análise (figura 4.11). O ponto é considerado um extremo

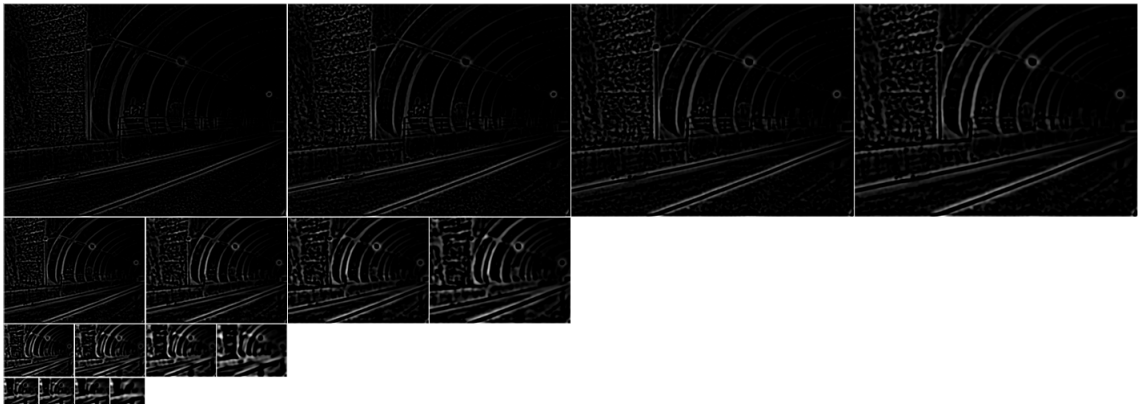


Figura 4.10: Imagens originadas por diferença de gaussianas ao longo de quatro oitavas.

local caso o valor de intensidade seja superior ou inferior ao registado por cada um dos vizinhos. Note-se que a pesquisa de extremos locais não recai sobre as escalas superior e inferior, devido à escassez de pontos vizinhos.

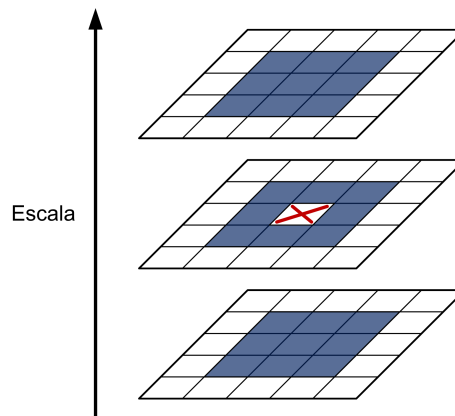


Figura 4.11: Pontos considerados na pesquisa de extremos.

4.1.2.2 Obtenção da Localização dos Pontos

O processo de detecção de extremos ao longo do espaço de escala origina um excesso de pontos extremos candidatos, muitos deles instáveis. Nesta etapa, a localização dos extremos é melhorada e atinge uma precisão sub-píxel, em seguida é executado um estudo baseado no valor dos píxeis ao redor do ponto, para identificar e posteriormente rejeitar aqueles que registam baixos níveis de contraste, sendo por isso mais sensíveis a ruído.

Para cada ponto candidato é levada a cabo uma interpolação dos dados na vizinhança, com o objectivo de determinar com exactidão a posição do extremo local. A interpolação

é executada com o recurso à expansão segundo a série de Taylor até ao termo de segunda ordem, da função de espaço de escala, $D(x, y, \sigma)$, centrada no ponto candidato.

$$D(x) = D + \frac{\delta D^T}{\delta x} x + \frac{1}{2} x^T \frac{\delta^2 D}{\delta x^2} x \quad (4.10)$$

- Onde D e as respectivas derivadas são calculadas em relação ao ponto candidato e $x = (x, y, \sigma)^T$ indica o *offset* relativamente ao ponto candidato.

O incremento de localização do extremo local, \hat{x} , face à localização do ponto candidato é dada pela derivação desta função em ordem a x , seguida a igualdade a zero, ficando:

$$\hat{x} = \frac{\delta^2 D^{-1}}{\delta x^2} \frac{\delta D}{\delta x} \quad (4.11)$$

Um incremento \hat{x} superior a 0.5 em qualquer dimensão é o indicio de que o extremo se encontra mais próximo de outro ponto candidato, logo a posição do ponto candidato é corrigida e a interpolação é executada relativamente a esta nova localização. Caso contrário, o incremento é adicionado à posição do ponto candidato, de modo a reflectir na posição do extremo o aumento de precisão adjacente à interpolação. À posição do extremo é associada a componente σ que indica a sua escala. A determinação da localização do extremo no espaço de escala baseia-se no método de pirâmides híbridas [69].

A validação do nível de contraste do candidato implica o cálculo do termo de segunda ordem da expansão de Taylor relativamente ao *offset* \hat{x} . Caso o valor resultante desta operação seja inferior a 0.03 o ponto candidato é excluído.

Finalmente, devido ao facto da função diferença de gaussianas produzir respostas significativas ao longo de zonas de fronteira, é fundamental executar uma filtragem de pontos candidatos com localização deficiente e resposta coerente com a manifestada por *edges*. Um ponto correspondente a uma zona de fronteira caracteriza-se por uma curvatura principal pouco acentuada ao longo do *edge*, a par de uma curvatura principal bastante pronunciada no sentido transversal. A determinação da magnitude das curvaturas principais resulta do cálculo da matriz Hessiana 2×2 , H , relativamente ao ponto candidato e respectiva escala.

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (4.12)$$

Os valores próprios de H são proporcionais às curvaturas principais porém, com vista à redução da complexidade do algoritmo, o cálculo dos valores próprios é substituído pela análise da proporção entre a magnitude de ambos, fórmula (4.13).

$$\frac{\text{traço}(H)^2}{\text{Det}(H)} \leq \frac{(r+1)^2}{r} \quad (4.13)$$

- Onde $\text{traço}(H) = D_{xx} + D_{yy}$, $\text{Det}(H) = D_{xx}D_{yy} - (D_{xy})^2$ e o caracter r representa a proporção entre a magnitude dos valores próprios.

O caracter r regista o valor limite admissível para discrepância entre os valores próprios³.

4.1.2.3 Determinação da Orientação

O algoritmo SIFT atribui a cada ponto de interesse um valor de orientação que depende da evolução dos gradientes de intensidade nessa zona. Posteriormente o descritor é construído relativamente a este parâmetro de rotação, estratégia vital para alcançar imunidade de detecção relativamente à rotação da imagem, pois em caso de correspondência entre pontos, a diferença de orientação indica a rotação a efectuar entre os descritores para que a sua comparação seja possível.

O processo implica o cálculo da magnitude, $m(x, y)$, e orientação, $\theta(x, y)$, do gradiente na imagem $L(x, y, \sigma)$ para cada píxel na vizinhança do ponto de interesse, com σ correspondente à escala mais próxima do ponto de interesse.

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (4.14)$$

$$\theta(x, y) = \tan^{-1} \left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \right) \quad (4.15)$$

Os valores calculados são agregados num histograma com 36 barras, que discretizam a orientação com uma resolução de 10° . A intensidade de uma barra do histograma corresponde à magnitude do gradiente segundo a direcção associada à barra. A contribuição da magnitude calculada em cada píxel é pesada por uma janela Gaussiana circular, com desvio padrão 1.5 vezes superior à escala do ponto de interesse. Os picos do histograma correspondem a orientações dominantes.

³Em [65] é utilizado o limite máximo de $r = 10$

Uma vez completo o histograma, são atribuídas ao ponto de interesse as orientações correspondentes ao maior pico, em conjunto com os picos locais que apresentem um valor superior a 80% do pico máximo. Caso sejam conferidas múltiplas orientações a um determinado ponto de interesse, são criadas réplicas desse ponto, com a mesma localização e escala, às quais será atribuída uma única orientação. Finalmente é ajustada uma parábola às três barras do histograma mais próximas do pico, com o intuito de interpolar a posição do pico para efeitos de aumento da precisão nesta grandeza.

4.1.2.4 Construção do Descritor

Nos passos anteriores foram encontradas as localizações dos pontos de interesse ao longo das várias escalas, tendo sido ainda atribuídos valores de orientação concordantes com os gradientes de intensidade na vizinhança. Estas etapas garantem invariância ao enquadramento do ponto na imagem, escala e rotação. Nesta última fase é calculado para cada ponto de interesse um descritor altamente distintivo e invariante face a alterações de iluminação e perspectiva. As operações descritas em seguida incidem sobre a imagem com escala mais próxima da atribuída ao ponto de interesse.

O desenvolvimento do descritor inicia-se com a criação de um conjunto de histogramas de rotação com oito barras, relativos a uma região de 4×4 píxeis. São construídos 16 histogramas que cobrem uma área de 16×16 píxeis centrada no ponto de interesse. Para cada histograma contribuem os valores magnitude e orientação calculados para os píxeis dentro da respectiva janela 4×4 . As magnitudes são pesadas por uma função gaussiana com desvio padrão correspondente a metade da largura da janela do descritor. O descritor corresponde ao conjunto de valores de todos os histogramas representados em forma de vector. O vector é normalizado com vista à obtenção de uma caracterização imune a variações de iluminação e perspectiva. Para reduzir os efeitos de iluminação não linear é adicionado a todos os elementos o valor 0.2, seguido de uma nova operação de normalização.

4.2 Correspondência entre Imagens

A correspondência de *features* em múltiplas imagens é executada geralmente através de comparação entre descritores ou recorrendo ao método de *Template Matching*.

4.2.1 Comparação de Descritores do Tipo SIFT

Os descritores codificam as propriedades da imagem em redor do ponto de interesse, criando para cada ponto uma marca de identidade própria. Sempre que as *features* possuam um descritor a correspondência deve basear-se na comparação destes elementos.

Segundo o algoritmo SIFT [65] o descritor de um dado ponto de interesse é representado sob a forma de um vector, que armazena os dados de 4×4 histogramas com 8 barras, o que resulta num total de 128 elementos. No método SIFT dois descritores de *features* em imagens diferentes são considerados equivalentes caso a distância Euclidiana entre o par seja inferior à manifestada por todas as outras combinações. Note-se que a distância Euclidiana em causa é calculada através dos 128 elementos contidos no descritor e não corresponde à diferença de posição das *features* na imagem.

$$E_{i,j} = \|d_i - d_j\|_2 \quad (4.16)$$

Com vista à redução do número de associações erradas, caso a menor distância entre descritores seja inferior a 60% da distância para a segunda melhor hipótese, a correspondência não é estabelecida.

4.2.2 *Template Matching*

A técnica de *Template Matching* procura identificar porções de imagem idênticas pela comparação directa da intensidade dos píxeis. É portanto uma técnica indicada para lidar com *features* desprovidas de descritor, como é o caso das produzidas pelo detector de Harris ou pelo algoritmo FAST.

A pesquisa de correspondências inicia-se com a construção de um *template*, composto pelos píxeis dentro de uma janela quadrada centrada na posição da *feature* na imagem de referência. Em seguida o *template* desloca-se sobre a imagem alvo e em cada posição é calculada uma medida que traduz a semelhança entre os dois conjuntos de píxeis que se sobrepõem.

A medida de semelhança resulta da aplicação de um método de correlação linear, como por exemplo a Soma do Quadrado das Diferenças (*Sum of Squared Differences* (SSD)):

$$SSD(x, y) = \sum_{(i,j) \in W} \left(Tp(i, j) - I(x - i, y - j) \right)^2 \quad (4.17)$$

- Onde Tp representa o *template* com dimensão W centrado no píxel (x, y) da imagem alvo I . Os índices i e j correspondem à localização de um elemento no *template*.

A quantidade SSD mede o quadrado da distância Euclidiana entre o *template* e a parcela da imagem alvo, logo um resultado próximo de zero indica uma forte semelhança. A SSD pode ser substituída pela Correlação Cruzada Normalizada (*Normalized Cross Correlation* (NCC)), que, apesar de provocar uma carga computacional superior, produz uma resposta imune a variações lineares de iluminação [70].

$$NCC(x, y) = \frac{\sum_{(i,j) \in W} \left(Tp(i, j) \cdot I(x + i - y + j) \right)}{\sqrt{\sum_{(i,j) \in W} \left(Tp(i, j) \right)^2 \cdot \sum_{(i,j) \in W} \left(I(x + i - y + j) \right)^2}} \quad (4.18)$$

O valor de saída da NCC enquadra-se entre 0 e 1. Uma resposta próxima de um 1 indica um elevado grau de semelhança.

Constata-se que as medidas anteriores têm tendência a associar o *template* com zonas na imagem alvo de intensidade superior e padrão entre píxeis idêntico. Para garantir que um *template* escuro é combinado com um zona escura na imagem à função de medida NCC subtrai-se a média de intensidade em cada uma das zonas, resultando na medida *Zero-mean Normalized Cross Correlation* (ZNCC).

$$ZNCC(x, y) = \frac{\sum_{(i,j) \in W} \left(Tp(x, y) - \mu(Tp) \right) \cdot \left(I(x + i, y + j) - \mu(I_c(x, y)) \right)}{\sqrt{\sum_{(i,j) \in W} \left(Tp(x, y) - \mu(Tp) \right)^2 \cdot \sum_{(i,j) \in W} \left(I(x + i, y + j) - \mu(I_c(x, y)) \right)^2}} \quad (4.19)$$

- Onde $\mu(Tp)$ indica a média de intensidade dos píxeis do *template* e $\mu(I_c)$ representa a média de intensidade dos píxeis da imagem alvo sobre os quais se localiza o *template*.

4.3 Projecção Perspectiva

O processo pelo qual se mapeiam pontos do mundo em pontos na imagem é conhecido pela designação de projecção perspectiva. A projecção perspectiva constitui uma operação essencial em aplicações de localização baseadas em visão computacional.

O cálculo da projecção perspectiva é efectuado com base num modelo que aproxima as características do sensor. Neste domínio constata-se a popularidade do modelo *pinhole*, que caracteriza a câmara através do centro óptico *c.o.*, o plano do centro óptico R^c e o plano da imagem R^i . O plano do centro óptico é paralelo ao plano da imagem e a distância entre eles é designada por distância focal, f . O modelo *pinhole* considera a passagem dos raios de luz por um orifício de abertura diminuta antes da incidência sobre o plano da imagem, permitindo assumir que um ponto tridimensional do cenário corresponde a um único ponto no plano da imagem.

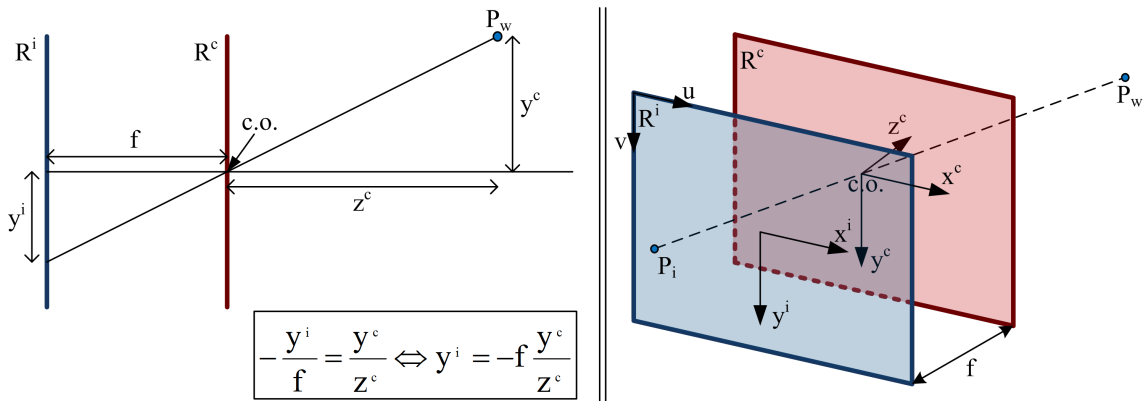


Figura 4.12: Modelo *pinhole*. À esquerda uma apresentação 2D do modelo pinhole. À direita a representação tridimensional acompanhada pelos sistemas referências da câmara e da imagem.

A imagem (4.12) indica uma relação geométrica simples, que possibilita descrever a projecção por intermédio da equação linear (4.20).

$$\begin{bmatrix} x^i \\ y^i \end{bmatrix} = -\frac{f}{z^c} \begin{bmatrix} x^c \\ y^c \end{bmatrix} \quad (4.20)$$

- Onde (x^i, y^i) são as coordenadas métricas no plano da imagem, (x^c, y^c, z^c) a posição do ponto P_w no referencial da câmara e f indica a distância focal.

De acordo com o modelo *pinhole*, a passagem do raio de luz pelo orifício do centro óptico produz uma inversão na imagem, razão pela qual o segundo membro da equação (4.20) se encontra negado. Este pormenor é ultrapassado pela inversão da imagem, o que equivale à deslocação do plano da imagem de modo que este se posicione entre o ponto P_w e o centro óptico, mantendo no entanto o mesmo valor de distância focal (figura 4.13).

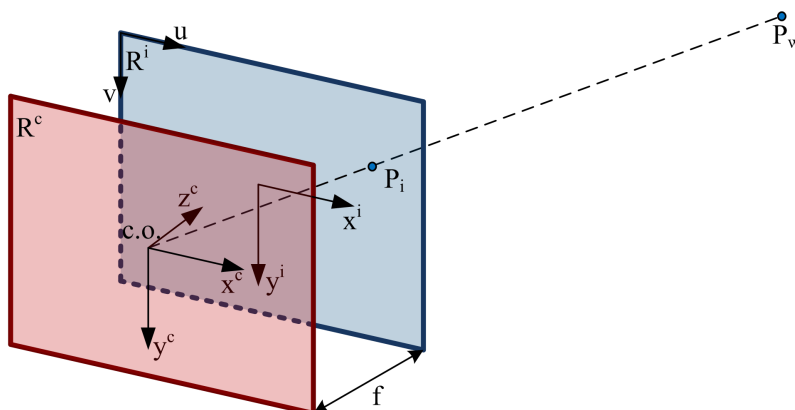


Figura 4.13: Modelo *pinhole* com plano da imagem em frente do centro óptico.

Dada a nova disposição do plano da imagem pode reformular-se a equação (4.20) ficando (4.21):

$$\begin{bmatrix} x^i \\ y^i \end{bmatrix} = \frac{f}{z^c} \begin{bmatrix} x^c \\ y^c \end{bmatrix} \quad (4.21)$$

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

Passemos então à formulação das transformações entre estes sistemas referenciais. Usando coordenadas homogêneas, um ponto P_w definido segundo um referencial global é expresso do seguinte modo: $P_w = [x^w, y^w, z^w, 1]$. A transformação deste ponto para o sistema de coordenadas da câmara é regida pela fórmula (4.22).

$$P_c = \left[\begin{array}{ccc|c} & & & \\ & R(3 \times 3) & & t(3 \times 1) \\ & & & \\ \hline 0 & 0 & 0 & 1 \end{array} \right] P_w \quad (4.22)$$

Usando a relação da fórmula (4.21) pode definir-se a transformação do referencial da câmara (R_c) para o referencial da imagem (R_i) pela fórmula (4.23). O ponto no referencial da imagem será um vector homogéneo de dimensão 3.

$$P_i = \left[\begin{array}{ccc} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 1 \end{array} \right] \left[\begin{array}{c} \frac{x^c}{z^c} \\ \frac{y^c}{z^c} \\ 1 \end{array} \right] \quad (4.23)$$

Finalmente as coordenadas de um ponto no referencial da imagem podem ser mapeadas em píxeis na imagem pela aplicação da seguinte fórmula:

$$P_{u,v} = \left[\begin{array}{ccc} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{array} \right] P_i \quad (4.24)$$

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

Assim, um ponto definido no referencial da câmara ocupará uma posição em píxeis na imagem dada pela fórmula (4.25).

$$P_{u,v} = \left[\begin{array}{ccc} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{array} \right] \left[\begin{array}{ccc} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 1 \end{array} \right] \left[\begin{array}{c} \frac{x^c}{z^c} \\ \frac{y^c}{z^c} \\ 1 \end{array} \right] \quad (4.25)$$

A fórmula (4.25) é equivalente a:

$$P_{u,v} = \begin{bmatrix} f_x k_u & 0 & u_0 \\ 0 & f_y k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{x^c}{z^c} \\ \frac{y^c}{z^c} \\ 1 \end{bmatrix} \quad (4.26)$$

As coordenadas do vector $\begin{bmatrix} \frac{x^c}{z^c} & \frac{y^c}{z^c} & 1 \end{bmatrix}^T$ são designadas por coordenadas de projecção normalizadas. A matriz 3×3 da fórmula (4.26) é conhecida como matriz da câmara ou matriz de parâmetros intrínsecos, pois define as características físicas da câmara. A matriz de parâmetros instrínsecos é usualmente indicada através da letra "K".

$$K = \begin{bmatrix} \alpha_u & s & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.27)$$

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

Os parâmetros intrínsecos e extrínsecos são habitualmente obtidos por calibração experimental, estando disponíveis aplicações em código aberto para este efeito [71] [72]. Como indicado pela fórmula (4.26), os parâmetros intrínsecos possibilitam a conversão entre o referencial da câmara e o referencial em píxeis da imagem. Quanto aos parâmetros extrínsecos definem uma rotação e uma translação do referencial da câmara para outro referencial, que em aplicações de localização coincide com o referencial do veículo.

4.4 Distorção Provocada pela Lente

O modelo *pinhole* considera que um ponto do mundo, o seu ponto de projecção na imagem e o centro óptico da câmara são colineares, logo linhas rectas no mundo dão origem a linhas rectas na imagem. Na realidade isso não acontece devido à perturbação causada pela lente das câmaras. Os raios de luz ao passarem pela lente são refractados, permitindo assim aumentar o ângulo de visão de modo a capturar um panorama mais alargado do mundo. Contudo a lente produz um efeito de distorção na imagem, que se acentua à medida que a distância focal e/ou qualidade da lente diminuem [73].

A distorção pode ser descrita segundo duas componentes: radial ou tangencial. A distorção tangencial é originada fundamentalmente pelo desalinhamento físico dos elementos da lente. Nas câmaras modernas esta componente pode ser desprezada, pois a perturbação induzida na imagem é mínima, principalmente em dispositivos com distância focal fixa⁴.

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

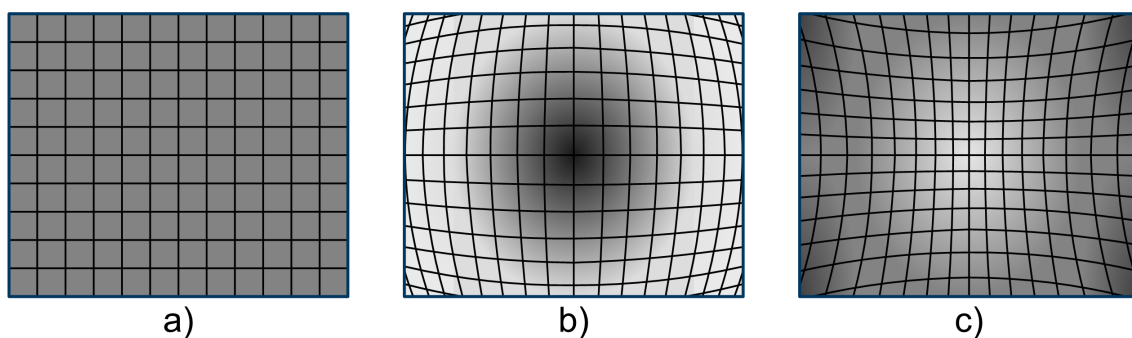


Figura 4.14: Projecção de rectas do mundo na imagem dependendo do tipo de distorção causada pela lente. a) Ausência de distorção. b) Distorção radial em forma de barril. c) Distorção radial em forma de almofada.

A compensação do efeito de distorção da lente pode ser conseguida pela aplicação de um modelo polinomial introduzido por Brown [74]. Considerem-se as coordenadas de projecção normalizadas de um ponto não afectado por distorção: $[x^u \ y^u]^T = [x^c/z^c \ y^c/z^c]^T$. A distorção da lente origina um deslocamento deste ponto para as coordenadas $[x^d \ y^d]^T$ dadas pela seguinte equação.

$$\begin{bmatrix} x^d \\ y^d \end{bmatrix} = \begin{bmatrix} x^u \\ y^u \end{bmatrix} \left(1 + \kappa_1 r_u^2 + \kappa_2 r_u^4 + \kappa_5 r_u^6 \right) + dx \quad (4.28)$$

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

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

$$r_u = \sqrt{x^{u2} + y^{u2}} \quad (4.29)$$

$$dx = \begin{bmatrix} 2\kappa_3 x^u y^u + \kappa_4 (r_u^2 + 2(x^u)^2) \\ \kappa_3 (r_u^2 + 2(y^u)^2) + 2\kappa_4 x^u y^u \end{bmatrix} \quad (4.30)$$

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

Como o modelo de distorção opera sobre as coordenadas de projecção normalizadas, a determinação das coordenadas dos píxeis na imagem, considerando o efeito de distorção, é conseguida pela aplicação dos parâmetros intrínsecos.

Geralmente os parâmetros de distorção são obtidos por calibração.

4.5 Visão Stereo

A projecção perspectiva inversa não tolera a transformação de píxeis no referencial da imagem em pontos tridimensionais num referencial externo, pois as coordenadas bidimensionais de um píxel em conjugação com o modelo *pinhole* não possuem informações suficientes para ultrapassar a ambiguidade em profundidade. Segundo o modelo *pinhole*, vários pontos tridimensionais posicionados ao longo de um segmento de recta com origem no centro óptico, projectam-se no mesmo ponto da imagem. Assim, a projecção perspectiva inversa de um píxel resulta num segmento de recta semi-infinito que intersecta as coordenadas do ponto no referencial métrico da imagem e tem com origem o centro óptico.

A estratégia empregue na recuperação das coordenadas tridimensionais de um ponto a partir da sua projecção na imagem é designada por triangulação, e consiste na visualização do ponto no mundo a partir de duas ou mais perspectivas diferentes (paralaxe). A aquisição de imagens de diferentes perspectivas pode ser conseguida pela integração de várias câmaras ou pelo reposicionamento de apenas um sensor. A posição tridimensional do ponto é indicada pela intersecção dos vários segmentos de recta, resultantes da projecção inversa desenvolvida a cada perspectiva. Para além da necessidade de conhecer rigorosamente a posição e orientação do sensor em cada local de amostragem, o procedimento de triangulação obriga à identificação do mesmo ponto nas várias imagens, para que as projecções inversas se desenvolvam relativamente ao mesmo local do mundo. Em

sistemas *stereo* a correspondência entre imagens é normalmente sustentada nas restrições impostas pela geometria epipolar.

4.5.1 Geometria Epipolar

A imagem (4.15) ilustra o princípio da geometria epipolar *stereo*, onde o ponto tridimensional P_w é projectado no plano da imagem de duas câmaras, entre os quais se conhece a posição relativa (t) e a orientação relativa (R).

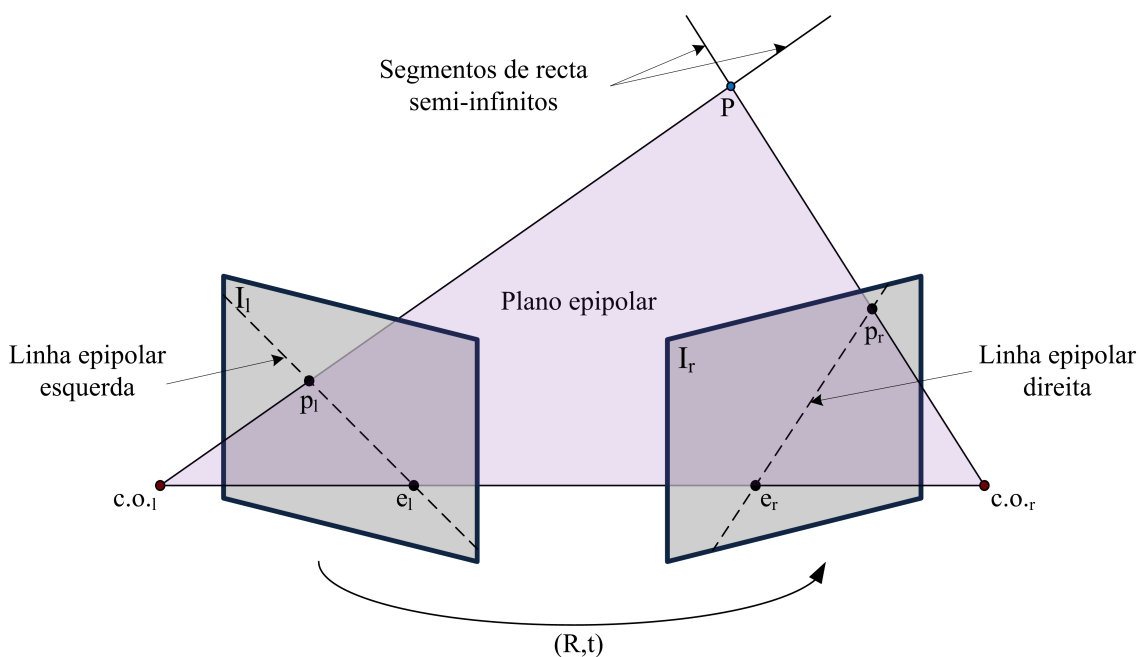


Figura 4.15: Geometria epipolar *stereo*.

Os pontos identificados pelos caracteres *o.c.* representam os centros ópticos da câmara da direita, r , e da câmara da esquerda, l , à linha que os une é dado o nome de *baseline*, b . O ponto, no plano da imagem de uma câmara, no qual se projecta o centro óptico da outra câmara é designado por epípolo (*epipole*). Assim o epípolo esquerdo, e_l , corresponde à projecção de $c.o.r$ no plano I_l e o epípolo direito, e_r , à projecção de $c.o.l$ em I_r . O plano epipolar é definido pelos pontos P_w , $c.o.r$ e $c.o.l$. Por sua vez, o cruzamento do plano epipolar com os planos das imagens origina as linhas epipolares. Todas as linhas epipolares possíveis intersectam os respectivos epípolos.

A propriedade conhecida como restrição epipolar – *epipolar constraint* – define o seguinte: considerando o ponto p_l como o ponto de projecção na imagem esquerda do ponto

tridimensional P_w , o ponto p_r , fruto da projecção de P_w na imagem da direita, encontra-se posicionado ao longo da linha epipolar direita.

A geometria epipolar *stereo* é definida para um par de câmaras com disposição relativa constante e é independente da calibração de ambas. Pelo facto de não incorporar os parâmetros intrínsecos, a geometria epipolar, expressa pela fórmula (4.31), estabelece relações entre os sistemas referenciais métricos das imagens.

$$p_r^T F p_l = 0 \quad (4.31)$$

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

$$F = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} R \quad (4.32)$$

Utilizando a matriz fundamental é possível calcular a posição dos epipolos (4.33) (4.34), bem como da linha epipolar numa imagem associada a um ponto da outra imagem (4.35) (4.36) [75].

$$F e_l = 0 \quad (4.33)$$

$$F^T e_r = 0 \quad (4.34)$$

$$l_r = F p_l \quad (4.35)$$

$$l_l = F p_r \quad (4.36)$$

A matriz Essencial (4.37) corresponde à matriz Fundamental acrescida dos parâmetros intrínsecos das câmaras (4.27), assim é possível executar as relações epipolares no referencial de píxeis da imagem.

$$(K_r^T p_r)^T E (K_l^T p_l) = 0 \quad (4.37)$$

4.5.2 Matriz de Homografia

A matriz de homografia estabelece relações de posição entre dois planos distintos, definindo as regras de mapeamento de pontos entre planos. Assim, para relações entre pontos de duas imagens a morfologia do cenário observado é irrelevante. Considere-se um ponto tridimensional no mundo P_w com pontos de projecção na imagem da direita e imagem da esquerda, respectivamente p_r e p_l em coordenadas homogêneas. A relação entre pontos das duas imagens pela matriz de homografia H (4.39) é expressa da seguinte forma (4.38).

$$p_r = H \cdot p_l \quad (4.38)$$

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \quad (4.39)$$

A matriz de homografia tem dimensão 3×3 e possui oito graus de liberdade. A correspondência entre um par de pontos em duas imagens resulta em duas equações:

$$x_r = \frac{x_l h_{11} + y_l h_{12} + h_{13}}{x_l h_{31} + y_l h_{32} + h_{33}} \quad (4.40)$$

$$y_r = \frac{x_l h_{21} + y_l h_{22} + h_{23}}{x_l h_{31} + y_l h_{32} + h_{33}} \quad (4.41)$$

Estas equações definem o seguinte sistema

$$H = \begin{cases} x_r = \frac{x_l h_{11} + y_l h_{12} + h_{13}}{x_l h_{31} + y_l h_{32} + h_{33}} \\ y_r = \frac{x_l h_{21} + y_l h_{22} + h_{23}}{x_l h_{31} + y_l h_{32} + h_{33}} \end{cases} = \begin{cases} x_l h_{11} + y_l h_{12} + h_{13} - x_r x_l h_{31} - x_r y_l h_{32} = x_r h_{33} \\ x_l h_{21} + y_l h_{22} + h_{23} - y_r x_l h_{31} - y_r y_l h_{32} = y_r h_{33} \end{cases} \quad (4.42)$$

Fazendo $h_{33} = 1$ é necessário determinar oito incógnitas para calcular os elementos da matriz de homografia. Como para cada par de pontos correspondentes se obtêm duas equações, o cálculo dos elementos da matriz de homografia exige no mínimo quatros pares de pontos correspondentes, sendo necessário que pelo menos três sejam não colineares.

$$H = \begin{bmatrix} x_{l1} & y_{l1} & 1 & 0 & 0 & 0 & -x_{r1}x_{l1} & -x_{r1}y_{l1} \\ 0 & 0 & 0 & x_{l1} & y_{l1} & 1 & -y_{r1}x_{l1} & -y_{r1}y_{l1} \\ x_{l2} & y_{l2} & 1 & 0 & 0 & 0 & -x_{r2}x_{l2} & -x_{r2}y_{l2} \\ 0 & 0 & 0 & x_{l2} & y_{l2} & 1 & -y_{r2}x_{l2} & -y_{r2}y_{l2} \\ x_{l3} & y_{l3} & 1 & 0 & 0 & 0 & -x_{r3}x_{l3} & -x_{r3}y_{l3} \\ 0 & 0 & 0 & x_{l3} & y_{l3} & 1 & -y_{r3}x_{l3} & -y_{r3}y_{l3} \\ x_{l4} & y_{l4} & 1 & 0 & 0 & 0 & -x_{r4}x_{l4} & -x_{r4}y_{l4} \\ 0 & 0 & 0 & x_{l4} & y_{l4} & 1 & -y_{r4}x_{l4} & -y_{r4}y_{l4} \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = \begin{bmatrix} x_{r1} \\ y_{r1} \\ x_{r2} \\ y_{r2} \\ x_{r3} \\ y_{r3} \\ x_{r4} \\ y_{r4} \end{bmatrix} \quad (4.43)$$

O cálculo da matriz de homografia como apresentado anteriormente produz um resultado único, pois a utilização de apenas 4 correspondências origina uma matriz com característica igual a 8. A utilização de um maior número de correspondências no cálculo da matriz de homografia, com vista ao melhoramento do resultado, pode não permitir a obtenção de uma única solução, devido à utilização de valores redundantes afectados por erro [76]. Nestes caso é necessário recorrer a métodos de optimização matemática para aproximar a solução mais precisa.

4.5.3 Rectificação *Stereo*

A rectificação de imagens *stereo* [77] [78] [79] é uma prática comum, que antecede o cálculo de distâncias. O processo de rectificação executa uma re-disposição dos píxeis nas imagens de modo a simular o alinhamento dos centros focais e planos da imagem das duas câmaras. Deste modo, os epipolos são projectados para o infinito, o que torna as linhas epipolares horizontais e colineares, facilitando a correspondência entre imagens.

A operação de deformação das imagens é baseada na aplicação de uma matriz de homografia a cada imagem. O cálculo destas matrizes é governada pela imposição da colocação dos epipolos no infinito e pelo facto de as linhas epipolares se tornarem paralelas relativamente ao eixo de coordenadas u da imagem. Estas condições geram restrições a ter em conta no cálculo da matriz de homografia, contudo ainda ficam por definir quatro graus de liberdade [77]. Deste modo não existe uma solução única para o problema de rectificação e por isso existem inúmeras técnicas de cálculo das matrizes de homografia.

Os métodos para cálculo das matrizes de homografia partilham a principal preocupação de minimização da distorção a aplicar às imagens [77] [78].

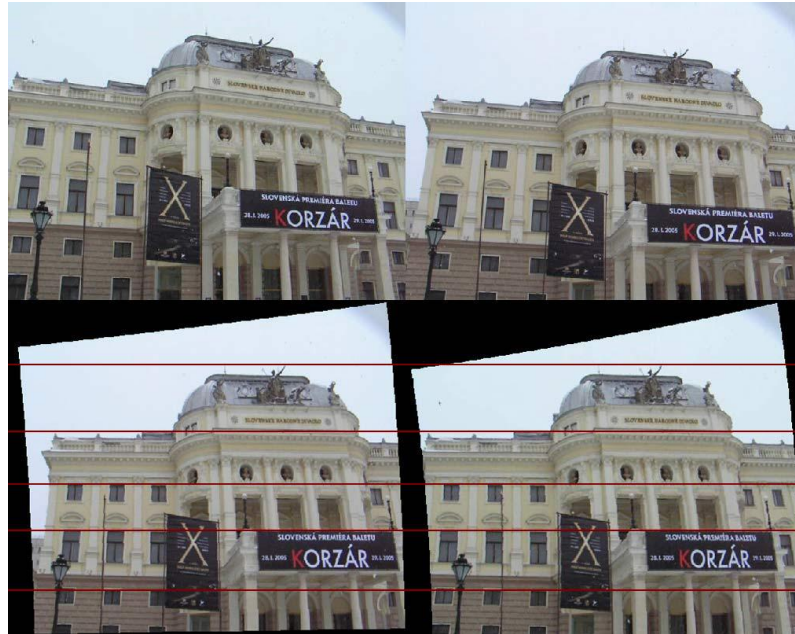


Figura 4.16: Rectificação de um par de imagens *stereo* e representação das linhas epipolares horizontais colineares fruto da localização do epípolo no infinito.

Após a aplicação das transformações no processo rectificação, a geometria do sistema *stereo* torna-se idêntica à apresentada na figura (4.17), o que torna bastante simples o cálculo das coordenadas tridimensionais de pontos do mundo por triangulação, fórmulas (4.44, 4.45, 4.46).

$$z = \frac{fb}{x_l - x_r} \quad (4.44)$$

$$x = \frac{x_l z}{f} \quad (4.45)$$

$$y = \frac{y_l z}{f} \quad (4.46)$$

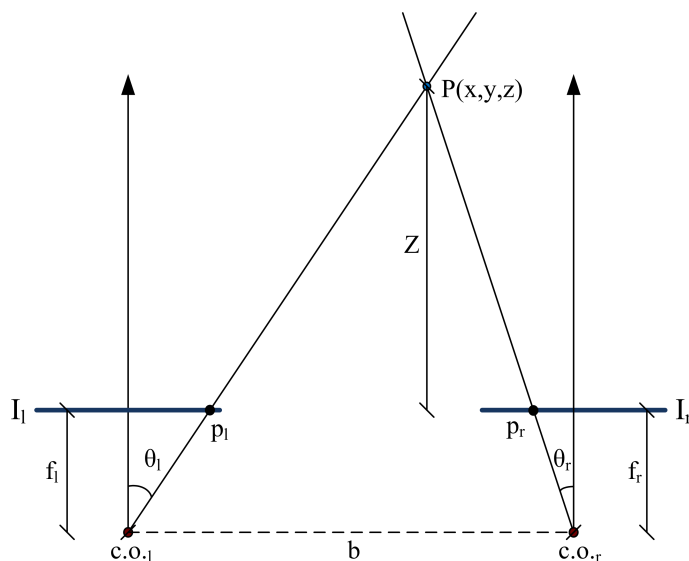


Figura 4.17: Geometria após rectificação – epipolos no infinito e planos da imagem paralelos.

4.6 RANSAC

O algoritmo *Random Sample Consensus* (RANSAC) [80] constitui uma opção robusta para estimação de parâmetros de um modelo matemático a partir de um conjunto de dados com elementos discrepantes – *outliers* – um elemento é considerado *outlier* caso não respeite as condições do modelo estimado. No campo da visão computacional, o algoritmo RANSAC, é aplicado fundamentalmente em processos de estimação da matriz fundamental, na correcção de correspondências entre pares de imagens *stereo* [81] e na distinção entre alvos estáticos e dinâmicos [82].

O algoritmo recebe como entradas o conjunto de dados, o modelo através do qual se pretende aproximar a distribuição dos dados e parâmetros que definem os níveis de confiança mínimos admissíveis para a aproximação. O algoritmo divide-se em dois momentos que se repetem de forma iterativa:

- Hipótese – É seleccionada aleatoriamente uma amostra dos dados através da qual se calcula uma estimativa dos parâmetros do modelo. A amostra é composta pelo número mínimo de elementos necessário à determinação dos parâmetros do modelo.
- Teste – No segundo instante, os restantes elementos do conjunto são comparados com o modelo definido pelos parâmetros calculados no passo anterior, de modo a conferir o nível de ajuste entre ambos. A qualidade de aproximação do modelo relaciona-se

com o número de pontos com ele consistentes, *inliers*. Caso o número de *inliers* de uma iteração supere os resultados obtidos anteriormente, o conjunto de *inliers* (*consensus set*) é armazenado e os parâmetros do modelo voltam a ser estimados com base neste conjunto.

A quebra do ciclo do algoritmo é desencadeada no instante em que uma das seguintes condições de paragem se verifique: o erro no modelo é inferior a um determinado limite, o número de *inliers* atinge uma dimensão satisfatória ou é alcançado um valor máximo de iterações. O número de iterações necessárias para assegurar, com um grau de probabilidade p , a escolha de pelo menos uma amostra aleatória composta exclusivamente por *inliers* respeita a seguinte fórmula:

$$N_{\text{iterações}} = \frac{\log(1 - p)}{\log(1 - (1 - \epsilon)^m)} \quad (4.47)$$

Onde ϵ representa a proporção de *outliers* no conjunto de dados e m reflecte o número mínimo de elementos necessários à definição do modelo. Normalmente o número máximo de iterações é ajustado segundo a fórmula anterior considerando $p = 0.99$ [73].

Dimensão da amostra	Proporção de <i>outliers</i> (ϵ)						
	5%	10%	20%	25%	30%	40%	50%
s							
1	2	2	3	4	4	6	7
2	2	3	5	6	7	11	17
3	3	4	5	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

Tabela 4.1: Número de iterações que assegura, com uma probabilidade de $p = 0.99$, a escolha de pelo menos uma amostra sem *outliers* com dimensão igual a s de um conjunto com proporção de outliers ϵ

A técnica de RANSAC aplicada à aproximação da matriz de homografia constitui uma estratégia robusta de filtragem de correspondências falsas entre pares de imagens do mesmo cenário. O algoritmo selecciona aleatoriamente quatro pares de *features* e, através das suas coordenadas, executa o cálculo da matriz de homografia que caracteriza este conjunto. Em seguida as restantes correspondências são classificadas como *inliers*

ou *outliers* dependendo do grau de concordância com a matriz calculada. A solução com maior número de *inliers* é seleccionada e as correspondências *outliers* são quebradas.

4.6.1 1-Point RANSAC

A correspondência entre *features* ao longo de sequências de imagens é uma prática comum aos processos de *Structure from Motion* e *Visual SLAM*. As ligações formuladas entre imagens resultam normalmente de algoritmos que executam a comparação de descritores ou aplicam métodos de correlação. Neste ponto, a presença de *features* muito semelhantes pode induzir correspondências falaciosas, pelo que é imperioso desenvolver à posteriori um mecanismo de filtragem capaz de identificar e descartar as ligações incoerentes.

O algoritmo RANSAC está estabelecido como o método genérico de estimação de modelos com base em conjuntos de dados contaminados por *outliers*. Esta técnica, quando aplicada nos processos de localização, tem como objectivo a selecção do maior conjunto de correspondências que implicitamente exprimem a mesma movimentação.

Recordando a tabela (4.1), o número de iterações que garante a escolha de um conjunto de amostras não contaminado por *outliers* cresce com o aumento do número de elementos necessários à formulação do modelo. Em sistemas monoculares calibrados, o cálculo da movimentação 3D implica um mínimo de cinco correspondências [83]. Porém, a estimação de movimentos planares obriga ao conhecimento de apenas duas correspondências [84]. Já a combinação de restrições não holonómicas, próprias de um automóvel, com movimentos planares permitiu em [85] reduzir à unidade o número de correspondências. A diminuição da cardinalidade do conjunto de amostras necessárias ao desenvolvimento do modelo é vantajosa dado que reduz o número de iterações espectáveis do algoritmo RANSAC, contudo as restrições ao movimento tidas em conta em [84] e [85] são incompatíveis com a dinâmica característica de sistemas com espaço de trabalho tridimensional.

Recentemente foi apresentada uma estratégia genérica, vocacionada para sistemas com seis graus de liberdade, denominada 1-Point RANSAC [86]. O modelo é definido com base numa única correspondência, graças à consideração de informações sobre o movimento da câmara, introduzidas sob a forma de distribuições probabilísticas. Estas distribuições são mantidas e actualizadas pelo algoritmo EKF, que se ocupa da estimação dos estados associados ao problema de SLAM.

A redução da dimensão da amostra implica um ganho computacional considerável. Tomando como exemplo um conjunto de correspondências onde a percentagem de *inliers* é 50%, o número de hipóteses aleatórias necessárias para escolher, com uma probabilidade de 0.99, uma amostra constituída somente por *inliers* baixa de 146 (em amostras com cinco elementos) para 7 (no algoritmo 1-Point RANSAC).

O algoritmo 1-Point RANSAC inicia-se com o ciclo de formulação de hipóteses e respectivo teste. Em cada iteração ocorre a selecção aleatória de uma correspondência e procede-se ao cálculo do estado estimado, conforme as equações de *update* do EKF. Nesta fase a matriz de covariância não é actualizada, mantendo baixo o custo computacional associado à geração de hipóteses.

A verificação dos dados de suporte envolve a previsão da posição de cada *feature* de acordo com o estado estimado correspondente à hipótese em estudo. À discrepância entre a posição prevista e a posição definida pela correspondência, na imagem mais recente, é atribuído o nome de inovação. As correspondências cuja inovação permaneça a baixo de um dado valor⁵ são compatíveis com o estado estimado, logo verificam o modelo actual e são consideradas *low-innovation inliers*.

O número de iterações é estabelecido de acordo com a fórmula 4.47 considerando a dimensão actual do conjunto de *low-innovation inliers*. No final do ciclo hipótese/teste a hipótese com maior suporte corresponde à melhor solução encontrada. As correspondências *low-innovation inliers* passam a *inliers* definitivas, quanto às correspondências que não aproximam o modelo podem corresponder a *outliers* ou a *inliers* com inovação elevada – *high-innovation inliers* – fruto de *landmarks* próximas da câmara, afectadas de forma significativa pelo movimento, ou por *landmarks* inicializadas recentemente, com níveis de incerteza elevados.

A diferenciação entre *outliers* e *high-innovation inliers* tem lugar após a actualização definitiva dos estados, pela execução da etapa de *update* parcial com *low-innovation inliers*, que consta das operações de *update* naturais do filtro EKF observando apenas o conjunto de *inliers*. A actualização do estado produz a redução do erro correlacionado entre *inliers* e *high-innovation inliers*, tornando estes últimos compatíveis com o modelo. Em seguida, para cada correspondência, é formulada a previsão da posição da *feature* pela propagação do estado estimado através do modelo de projecção da câmara. A correspondência é

⁵Em [86] o valor limite é de um píxel.

resgatada para o grupo de *inliers* caso seja compatível com a previsão, ou seja, se localize numa zona elíptica com probabilidade de 99% em torno da posição prevista. Os elementos excluídos pelo teste classificam-se como *outliers*, enquanto que a fracção reentrante no grupo de *inliers* provoca uma nova etapa de *update*, designada por *update* parcial com *high-innovation inliers*.

A partição da fase de *update* não implica o acréscimo da complexidade computacional [87].

4.7 Bundle Adjustment

O método *Bundle Adjustment* consiste num processo de optimização matemática voltado para a visão computacional normalmente associado a aplicações de *Structure from Motion* e reconstrução tridimensional [9] [81] [88]. O método refina simultaneamente a posição tridimensional das *features* observadas pelas câmaras e os parâmetros associados aos sensores, por intermédio da minimização dos erros de reprojecção (*reprojection errors*). O erro de reprojecção indica a divergência métrica entre um ponto registado na imagem e o local, no plano da imagem, onde o mesmo ponto é projectado, tendo em conta a sua posição tridimensional, os parâmetros e localização da câmara.

Considere-se um conjunto de pontos tridimensionais X_j , captados por um conjunto de câmaras com parâmetros intrínsecos K^i . As coordenadas do ponto j obtidas pela câmara i são representadas por x_j^i . A partir do conjunto de coordenadas nas imagens x_j^i o método *Bundle Adjustment* pretende determinar as matrizes dos parâmetros intrínsecos das câmaras K^i e as coordenadas dos pontos X_j , tal que $K^i X_j = x_j^i$. Como as medições contêm uma componente de erro, a equação $K^i X_j = x_j^i$ não será exactamente satisfeita, por esse motivo é procurada a solução de maior probabilidade (*maximum likelihood*) considerando como gaussianos os erros nas medidas. Neste contexto o problema passa a ser: estimar as matrizes \hat{K}^i e as coordenadas \hat{X}_j que se projectam exactamente nos pontos da imagem \hat{x}_j^i , tal que $\hat{K}^i \hat{X}_j = \hat{x}_j^i$, minimizando simultaneamente o erro de reprojecção. A expressão analítica deste procedimento é exposta na fórmula (4.48).

$$\min_{(\hat{K}^i, \hat{X}_j)} \sum_{ij} d(\hat{K}^i \hat{X}_j, \hat{x}_j^i)^2 \quad (4.48)$$

Onde $d(x, y)$ representa a distância geométrica no plano da imagem entre os pontos x e y com coordenadas homogêneas.

Este método tolera a impossibilidade de projecção de pontos fora do campo de visão da câmara, e permite a caracterização individual de cada medida em termos de covariância. Como desvantagens podem referir-se a necessidade de uma boa inicialização dos parâmetros e a elevada complexidade do problema de minimização, que advém do grande número de parâmetros envolvidos.

Esta página foi intencionalmente deixada em branco.

Capítulo 5

Localização

Conteúdo

5.1	Visão Geral Sobre Métodos de Localização <i>Indoor</i>	76
5.2	Navegação Inercial	78
5.2.1	Mecanização segundo o referencial de nível local	80
5.2.2	Erros típicos das unidades de medição inercial	82
5.3	SLAM - <i>Simultaneous Localization and Mapping</i>	84
5.3.1	SLAM Baseado em Filtro de Kalman Estendido	84
5.3.2	SLAM Baseado em Filtro de Partículas	90
5.3.3	SLAM por Registo de Varrimentos Laser	92
5.3.4	Visual SLAM	93
5.4	Sistemas Multirobóticos de Localização Cooperativa	102

Este capítulo tem como objectivo a identificação e descrição de estratégias de localização em ambiente *indoor* que possam ser moldadas à realidade de localização dentro de túneis. O conceito de localização diz respeito à estimação da posição e orientação de um ou mais veículos segundo um sistema de coordenadas global. Este processo encontra-se bastante dependente da utilização de sensores, logo lida com representações incompletas do mundo e está sujeito aos erros e ambiguidades contidos nas medições, razões que impedem a obtenção de resultados óptimos [8].

5.1 Visão Geral Sobre Métodos de Localização *Indoor*

A estratégia mais básica de localização toma o nome de *dead reckoning*, e consiste na estimação da disposição espacial do veículo pela integração numérica de medidas consecutivas. Os incrementos podem surgir de medições feitas por sensores inerciais ou através da contabilização da velocidade e direcção das rodas – odometria. Acontece que cada incremento possui uma componente de erro, que ao longo do processo de integração se acumula no valor de localização calculado, resultando num aumento monotónico da incerteza, fenómeno designado por *random walk*. O aperfeiçoamento do modelo do sistema ou a opção por um sensor com melhor performance não impedem a degradação da estimativa, apenas atenuam esta tendência.

O crescimento desmedido da incerteza torna geralmente as abordagens de *dead reckoning* por si só inviáveis em processos de localização a longo termo. Contudo, quando conjugadas com outros tipos de sensores, que periodicamente permitem corrigir os erros acumulados, as técnicas de *dead reckoning* são vantajosas devido ao bom desempenho em curtos espaços de tempo e às taxas de amostragem tipicamente elevadas [89], que permitem preencher com informação os intervalos entre instantes de amostragem dos outros sensores.

Para manter a incerteza dentro de limites aceitáveis é necessário introduzir no sistema medições de localização absolutas [90]. Em aplicações *indoor* onde a utilização de pontos de referência artificiais com posição global conhecida está fora de questão, as soluções mais eficazes passam pela localização baseada em mapas. O paradigma de localização através de mapas aplica o princípio de *map matching* [89], onde um mapa local construído através dos dados sensoriais num dado instante é comparado com um mapa global, servindo as correspondências entre ambos para estimar a localização segundo o referencial global.

A origem do mapa global é uma questão determinante no desenvolvimento dos algoritmos e produz modos de operação perfeitamente diferenciados. Por um lado, pode ser fornecido um mapa global preexistente, o que obriga a uma fase de reconhecimento antecedente à operação do veículo. Porém, a construção de um mapa à priori pode tornar-se bastante difícil e demorada como consequência de ambientes complexos e extensos, além disso a representação resultante é rígida e não permite a adaptação a novas circunstâncias como a alteração do cenário ou a exploração de zonas desconhecidas.

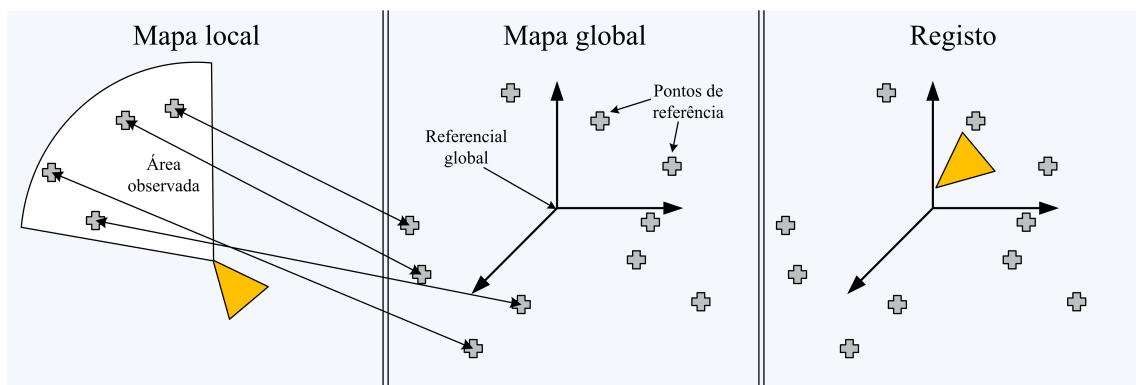


Figura 5.1: *Map matching* – Um mapa local é comparado com um mapa global de modo a determinar a posição do veículo segundo o referencial global.

A outra vertente designada por SLAM (*Simultaneous Localization and Mapping*) explora o conceito de mapeamento e localização em simultâneo e exclui qualquer tipo de conhecimento prévio acerca do cenário. No mapa, que inicialmente se encontra vazio, são sucessivamente inseridos pontos de referência seleccionados por algoritmos de análise dos dados sensoriais. Cada elemento do mapa é definido geralmente pela sua posição global, o que implica a transformação da medida do sistema referencial do veículo para o referencial global. Esta transformação implica a utilização da posição e orientação estimadas do veículo, logo a imprecisão nestes valores terá implicações na qualidade da posição global do ponto de referência no mapa. Sempre que um elemento do mapa é visualizado pelos sensores é calculada a discrepância entre a medida e o valor previsto, informação que é usada na etapa de estimação da localização. Assim, a localização e a posição dos pontos de referência no mapa encontram-se correlacionados e os erros patentes numa das partes irão manifestar-se na outra e vice-versa.

O núcleo de um algoritmo de SLAM corresponde normalmente a um método de estimação probabilístico, capaz de lidar com valores de incerteza quer nas componentes da localização quer nas referentes aos elementos do mapa. Deste modo é possível pesar a contribuição de cada elemento do mapa para a estimação da localização. A incerteza de um ponto registado no mapa depende normalmente da incerteza na localização do veículo no instante da medida e dos erros característicos do sensor. A incerteza é uma grandeza com dimensão igual ao número de estados do elemento a que se refere, assim é possível moldá-la de modo a adaptar-se às características dos sensores que observam o alvo. À medida que os elementos do alvo são repetidamente observados a estimativa da sua posição

aproxima-se da realidade e a incerteza diminui monotonicamente.

5.2 Navegação Inercial

Um sistema de navegação inercial (INS) baseia-se na medição das acelerações e velocidades angulares do corpo em movimento e usa capacidades computacionais para determinar constantemente a posição e a orientação do objecto em relação a um determinado sistema de referência. Os sistemas de navegação inercial são fundamentais em aplicações como a navegação de aeronaves, embarcações, mísseis, veículos espaciais, submarinos, entre outros.

A localização baseada em medições inerciais consiste num método de *dead reckoning*, pois cada nova posição é calculada em função da última posição conhecida acrescida de um incremento, dado pelas medições obtidas no último instante de amostragem. Os dados inerciais são adquiridos por unidades de medição inercial (IMU – *Inertial Measurement Unit*), constituídas por conjuntos de três acelerómetros ortogonais e três giroscópios posicionados também de forma ortogonal, que medem respectivamente acelerações lineares e velocidades angulares, através das quais, por intermédio de integração, se recupera a movimentação tridimensional do corpo.

A trajectória é expressa segundo um sistema referencial global pré-definido, cuja escolha toma particular importância pelo facto de influenciar a prestação do sistema de localização, bem como definir as zonas em que este pode operar. As medidas do sensor são relacionadas com o referencial global e convertidas em dados de localização, pela aplicação de um modelo adequado ao tipo de sistema referencial escolhido. Esta tarefa de conversão é comumente designada por mecanização. Entre os vários sistemas referenciais empregues em sistemas de navegação inercial destacam-se os listados em seguida.

1. Referencial inercial - Eixos (x^i, y^i, z^i) , com a origem no centro de massa da Terra, onde x^i aponta na direcção do ponto vernal e z^i coincide com o eixo de rotação médio da Terra¹.
2. Referencial ECEF (*Earth Centered Earth Fixed*) - Eixos (x^e, y^e, z^e) , onde a origem está no centro de massa da Terra. O eixo x^e está contido no plano equatorial e

¹Num referencial inercial as leis de Newton são válidas, logo, na ausência de forças externas, um corpo inicialmente em repouso mantém-se nesse estado, enquanto que um corpo em movimento rectilíneo uniforme preserva essas características ao longo do tempo

intersecta o meridiano médio de Greenwich e o eixo z^e coincide com o eixo z^i definido anteriormente. O referencial ECEF roda sobre o eixo z^e , com velocidade e sentido de rotação iguais aos manifestados pela Terra.

3. Referencial de navegação - Os eixos são indicados por (N, E, D) e a origem situa-se na posição local do sistema de navegação, sendo que o eixo N está orientado segundo a direcção Norte, o eixo E na direcção Este e o eixo D aponta na direcção do centro da Terra. A origem do sistema referencial de navegação coincide com a do referencial do veículo, logo o sistema referencial de navegação acompanha a movimentação do veículo.
4. Referencial de nível local - Os eixos do referencial de nível local (x^n, y^n, z^n) possuem uma orientação idêntica aos do referencial de navegação, porém a origem do referencial é estática, facto que o transforma num referencial inercial. Este referencial é muito utilizado em aplicações de navegação inercial, porém possui algumas restrições: só é válido em curtos espaços de tempo em comparação com o período de rotação da Terra e em pequenas áreas geográficas relativamente à curvatura da Terra.
5. Referencial do corpo - Definido pelos eixos (x^b, y^b, z^b) com origem no centro do veículo ou no centro do eixo das rodas, tratando-se de um veículo com tracção diferencial ou com direcção. Geralmente x^b aponta para a frente, y^b para a direita e z^b para baixo. Uma rotação em torno de x^b é designada “*roll*”, em torno de y^b “*pitch*” e em torno de z^b “*yaw*”.

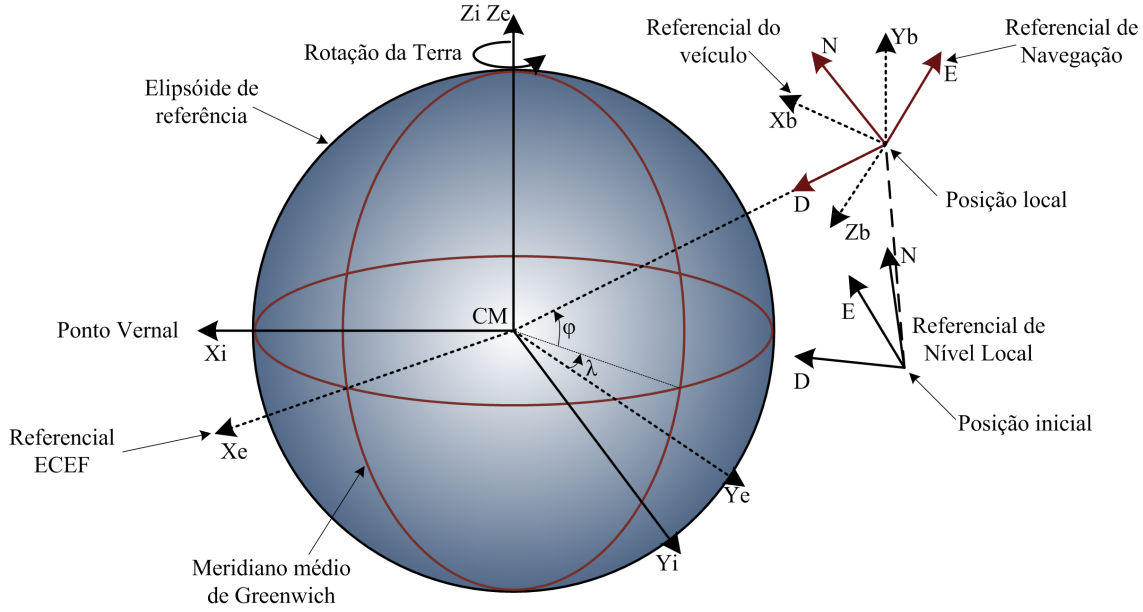


Figura 5.2: Disposição dos diversos sistemas referenciais em função da Terra.

5.2.1 Mecanização segundo o referencial de nível local

O referencial de nível local aproxima a superfície terrestre por um plano, o que origina as restrições temporais e espaciais anteriormente referidas, contudo, como consequência, as equações de mecanização tornam-se bastante simples (5.1).

$$\begin{bmatrix} r^n(k+1) \\ \Theta^n(k+1) \\ v^n(k+1) \end{bmatrix} = \begin{bmatrix} r^n(k) + v^n(k+1)\Delta t \\ \Theta^n(k) + (E_b^n w^b(k+1)\Delta t) \\ v^n(k) + (C_b^n a^b(k+1) + g^n)\Delta t \end{bmatrix} \quad (5.1)$$

- Onde $r^n = [r_x^n, r_y^n, r_z^n]^T$, $\Theta^n = [\phi^n, \theta^n, \psi^n]^T$ e $v^n = [v_x^n, v_y^n, v_z^n]^T$ indicam respectivamente posição, orientação e velocidade relativamente ao referencial de nível local. Δt estabelece o intervalo de tempo associado às medidas de aceleração linear e velocidade angular, representadas por $a^b = [a_x^b, a_y^b, a_z^b]^T$ e $w^b = [w_{roll}^b, w_{pitch}^b, w_{yaw}^b]^T$. C_b^n consiste na matriz rotação que roda do referencial do sensor para o referencial de nível local. A aceleração da gravidade no referencial de nível local é indicada por $g^n = [0, 0, g]^T$. A matriz E_b^n transforma a rotação sofrida pelo sensor em rotação sob a forma de ângulos de Euler.

Nas equações da mecanização verifica-se que a posição é calculada pela integração directa da velocidade. O mesmo não acontece com os estados dedicados à orientação e

velocidade, onde quer as velocidades angulares quer as acelerações lineares sofrem rotações com o intuito de alinhar estes dados com o sistema referencial de nível local.

Na equação de cálculo da velocidade, a aceleração da gravidade, presente nas medidas provenientes do sensor, é compensada pela adição do vector g^n , que contém o valor de aceleração da gravidade expresso segundo o referencial de nível local. A transformação das acelerações lineares para o referencial de nível local é conseguida pela aplicação da rotação definida pela matriz C_b^n , que por sua vez é calculada com base na orientação estimada. Por conseguinte, os erros em orientação manifestam-se no cálculo de velocidade, não só devido à rotação incorrecta dos dados, com também pela compensação defeituosa da aceleração da gravidade. Os erros introduzidos na velocidade vão repercutir-se na posição calculada originando um desvio face à trajectória verdadeira.

A matriz rotação C_b^n , destinada ao alinhamento das medidas de aceleração linear com o referencial de nível local, é obtida pela multiplicação das matrizes de rotação pela ordem $z - y - z$, que são compostas a partir dos ângulos contidos em Θ .

Os ângulos de Euler são definidos de acordo com uma sequência de rotações, por isso a taxa de variação temporal dos ângulos de Euler, $[\partial\phi/\partial t \ \partial\theta/\partial t \ \partial\psi/\partial t]^T$, não corresponde directamente à velocidade de rotação a que o sensor é submetido $[w_x^b \ w_y^b \ w_z^b]^T$. Para uma sequência de rotações *yaw-pitch-roll*, o primeiro ângulo (ψ) é submetido a duas rotações, o segundo (θ) sofre apenas uma, enquanto que o último (ϕ) não é afectado. Assim a velocidade angular do corpo em função da taxa de variação dos ângulos de Euler respeita a relação seguinte:

$$\begin{aligned} \begin{bmatrix} w_x^b \\ w_y^b \\ w_z^b \end{bmatrix} &= R(\phi)R(\theta) \begin{bmatrix} 0 \\ 0 \\ \partial\psi/\partial t \end{bmatrix} + R(\phi) \begin{bmatrix} 0 \\ \partial\theta/\partial t \\ 0 \end{bmatrix} + \begin{bmatrix} \partial\phi/\partial t \\ 0 \\ 0 \end{bmatrix} = \\ &= \begin{bmatrix} 1 & 0 & -\text{sen}(\theta) \\ 0 & \text{cos}(\phi) & \text{sen}(\phi)\text{cos}(\theta) \\ 0 & -\text{sen}(\phi) & \text{cos}(\phi)\text{cos}(\theta) \end{bmatrix} \begin{bmatrix} \partial\phi/\partial t \\ \partial\theta/\partial t \\ \partial\psi/\partial t \end{bmatrix} \end{aligned} \quad (5.2)$$

A taxa de variação temporal dos ângulos de Euler é obtida pela inversão de (5.2):

$$\begin{bmatrix} \partial\phi/\partial t \\ \partial\theta/\partial t \\ \partial\psi/\partial t \end{bmatrix} = \begin{bmatrix} 1 & \text{sen}(\phi)\tan(\theta) & \text{cos}(\phi)\tan(\theta) \\ 0 & \text{cos}(\phi) & -\text{sen}(\phi) \\ 0 & \text{sen}(\phi)\text{sec}(\theta) & \text{cos}(\phi)\text{sec}(\theta) \end{bmatrix} \begin{bmatrix} w_x^b \\ w_y^b \\ w_z^b \end{bmatrix} = E_b^n = \begin{bmatrix} w_x^b \\ w_y^b \\ w_z^b \end{bmatrix} \quad (5.3)$$

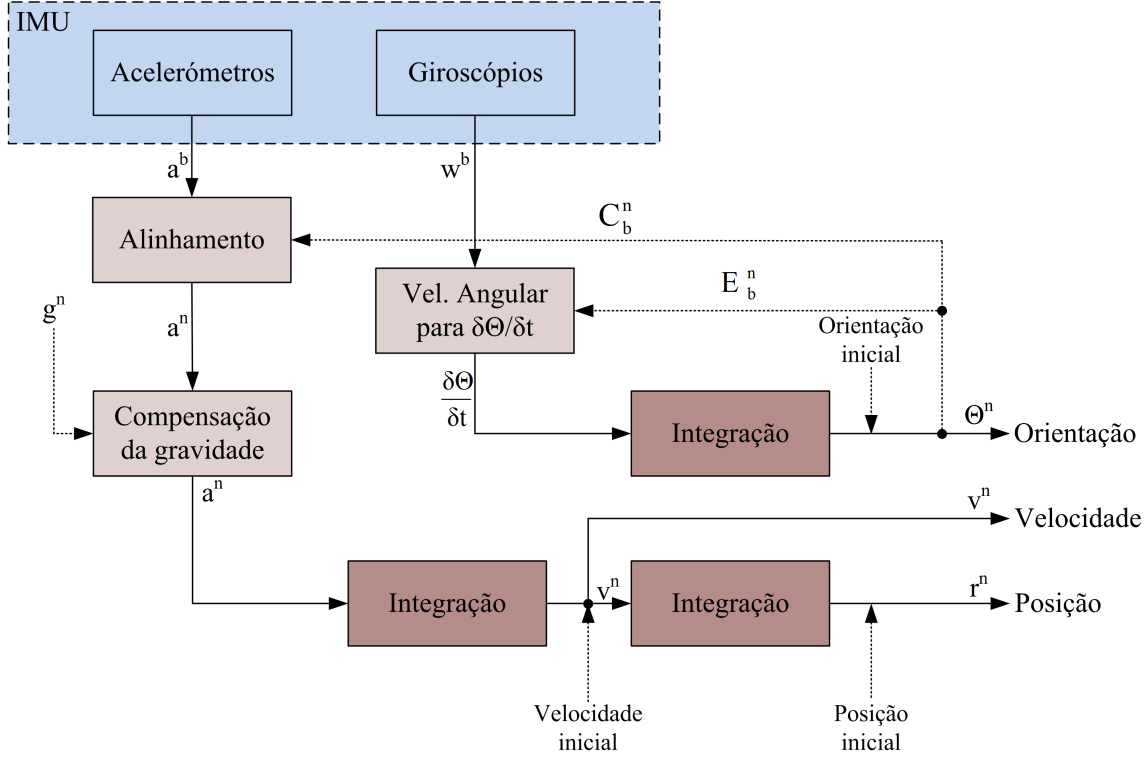


Figura 5.3: Esquema da mecanização segundo o referencial de nível local.

5.2.2 Erros típicos das unidades de medição inercial

Os erros induzidos na localização através de INS resultam em grande medida de erros intrínsecos aos sensores e deficiências na instalação e calibração. A consideração de medidas perturbadas por ruído dá origem ao fenómeno designado por *integration drift*, causado pela integração dos erros ao longo do tempo, resultando no aumento desmedido da incerteza nos valores de localização calculados.

5.2.2.1 Erros Característicos dos Giroscópios

As fontes de erro dominantes nos giroscópios são as seguintes.

1. *Bias / Drift rate* ($^{\circ}/h$ ou rad/s) – O erro *bias* consiste numa perturbação de nível constante no sinal de saída do sensor, que é independente do sinal na entrada. Por definição o erro *bias* não se altera durante o período de operação, contudo pode variar cada vez que o sensor é ligado. Este tipo de erro é normalmente expresso em $^{\circ}/h$ e por vezes toma a designação de *drift rate*. O método mais simples de quantificar o nível de *bias* de um sensor passa pela medição do valor médio de saída mantendo o giroscópios em repouso. Um *drift rate* de ε quando integrado origina um erro angular que cresce linearmente com o tempo: $\theta(t) = \varepsilon * t$.
2. Ruído aleatório (*random noise*) – O sinal de saída é perturbado por um ruído de origem termo-mecânica que provoca flutuações com uma frequência muito superior à frequência de amostragem. Como consequência as medições são perturbadas por ruído branco, que consiste numa sequência de variáveis aleatórias não correlacionadas com média igual a zero e variância finita σ^2 . Os fabricantes especificam este erro através do designado ângulo de *random walk* com unidades $^{\circ}/\sqrt{h}$.
3. Sensibilidade a alterações de temperatura – As flutuações externas de temperatura ambiente bem com o auto-aquecimento do sensor induzem uma alteração do valor de *drift*, originando um erro em orientação que cresce de forma linear com o tempo. Os giroscópios podem conter sensores de temperatura, que tornam possível a correção automática de erros causados por variações de temperatura.
4. Erros de calibração – Os erros de calibração são determinísticos e devem-se a falhas no alinhamento dos eixos, a erros de factor de escala e a erros de linearidade. Estes factores manifestam-se somente quando o sensor é submetido a rotações e causam um acréscimo de *drift*. Alguns sensores são capazes de estimar e compensar os erros de calibração.

5.2.2.2 Erros Típicos dos Acelerómetros

Os erros associados aos acelerómetros são análogos aos identificados no caso dos giroscópios.

1. *Bias* (mg) – O erro de *bias* retém as mesmas características indentificadas no caso dos giroscópios, porém devido à dupla integração da aceleração no cálculo da posição, um erro ε produz um erro em posição que cresce quadraticamente em função do tempo: $s(t) = \varepsilon * \frac{t^2}{2}$.
2. Ruído aleatório – O erro de ruído aleatório é normalmente expresso em $m/s/\sqrt{t}$.
3. Sensibilidade a alterações de temperatura – As variações de temperatura a que o sensor é sujeito têm consequências no valor de *bias*, esta influência é medida pelo *offset* em aceleração originado pelo aumento da temperatura em um grau, $g/^\circ C$.

5.3 SLAM - *Simultaneous Localization and Mapping*

Apesar de inúmeros trabalhos bem sucedidos no âmbito da localização e mapeamento em simultâneo (SLAM) ainda persistem algumas limitações para as quais as soluções encontradas são escassas e imperfeitas, nomeadamente ao nível da degradação da precisão da localização e do mapa para percursos de longas distâncias, da dificuldade de associação de dados face a elevadas incertezas no mapa e localização, do aumento dos requisitos computacionais à medida que o mapa se torna mais completo e da dificuldade em lidar com erros não gaussianos e média diferente de zero.

5.3.1 SLAM Baseado em Filtro de Kalman Estendido

A abordagem dominante de SLAM foi introduzida em [91] e consiste na aplicação do filtro de Kalman estendido (EKF – *Extended Kalman Filter*) [92] ao paradigma de localização e mapeamento em simultâneo. O mecanismo recursivo do EKF é favorável à implementação de abordagens incrementais de SLAM em tempo real.

Na abordagem EKF SLAM os mapas registam de forma discreta as particularidades do espaço envolvente recorrendo a distribuições gaussianas mono ou multivariável. À semelhança das demais estratégias de SLAM, para além das grandezas inerentes à representação de cada ponto de referência (*landmark*), os mapas agregam ainda a estimativa

da localização do veículo – posição e orientação. As distribuições gaussianas são mantidas em duas estruturas fundamentais do filtro de Kalman estendido, designadamente o vector de estados e a matriz de covariância. No vector de estados constam os valores médios da distribuição, ao passo que a matriz de covariância caracteriza a incerteza no valor dos estados.

Dado que os sistemas de SLAM não possuem conhecimento prévio do cenário, o mapa inicial diz respeito unicamente à localização do veículo acompanhado pela matriz de covariância, através da qual se declara a incerteza no valor inicial dos estados. À medida que os sensores observam pontos de referência, ocorre a sua inclusão no mapa de forma dinâmica. A primeira estimativa da localização de um ponto é dada pela projecção da medida de distância relativa segundo o referencial global, ao passo que a covariância depende das incertezas na medida e localização do veículo. A incerteza na localização de um ponto recentemente inserido no mapa encontra-se correlacionada exclusivamente com a localização do veículo. Após alguns instantes de estimação é espectável que a correlação se estenda a outros elementos do mapa. Uma forte correlação entre os pontos de referência do mapa indica que a posição relativa entre esses elementos possui uma confiança elevada.

O algoritmo EKF SLAM contempla as duas etapas típicas do EKF – previsão (*prediction*) e actualização (*update*) – às quais se juntam dois novos processos, designadamente a inserção de novos pontos de referência e a associação de dados entre as observações e o mapa. Na fase de previsão ocorre a propagação da posição da plataforma com base em técnicas de *dead reckoning*. Dado que este processo introduz erros na estimativa, a matriz de covariância é actualizada de modo a reflectir o aumento de incerteza na posição. Uma estimativa da posição do veículo menos coerente provoca a diminuição da confiança na distância relativa entre a plataforma e os pontos de referência, logo a incerteza na correlação entre estes estados também aumenta. Em suma, na fase de previsão mantêm-se as seguintes variáveis: posição dos pontos de referência, incerteza na posição dos pontos de referência e incerteza na correlação entre os pontos de referência.

A observação de um ponto de interesse desencadeia o método de associação de dados, o qual determina se o elemento corresponde a um previamente registado no mapa ou diz respeito a uma nova *landmark*. A pesquisa do vizinho mais próximo constitui o método de associação de dados mais popular. A observação de um ponto desconhecido leva à sua incorporação no mapa, o que obriga ao aumento da dimensão das matrizes. Caso uma

observação seja atribuída a um determinado ponto do mapa, entra em acção o mecanismo de *update*, que tira partido das informações da medida, de modo a actualizar a localização do veículo e a reduzir a incerteza na localização e na posição da *landmark* observada, bem como a incerteza de todas as que com ela se correlacionam.

5.3.1.1 Formulação do Problema de SLAM Segundo o EKF

O filtro de Kalman estendido é um algoritmo recursivo concebido para o cálculo de correcções de sistemas não-lineares a partir de medidas externas. As correcções são pesadas por ganhos, calculados de acordo com a estimativa actual dos erros nos estados do sistema, com o objectivo de minimizar o erro médio quadrático.

Nesta secção é apresentada a formulação da abordagem tradicional de SLAM [91]. Este método assume a existência de referências estáticas no mundo e capacidade de detecção deste pontos por parte de sensores a bordo do veículo. O vector de estados contém o conjunto de quantidades escolhidas para representar o estados do sistema. No caso de uma aplicação de SLAM o vector de estados agrupa a localização do veículo e a posição das *landmarks* no instante k do seguinte modo:

$$x(k) = [x_v(k), l_1(k), \dots, l_N(k)]^T \quad (5.4)$$

- Onde x_v representa o estado do veículo nos seis graus de liberdade $(x_v(k), y_v(k), z_v(k), \phi_v(k), \theta_v(k), \psi_v(k))$ e $l_i(k) = (x_i(k), y_i(k), z_i(k))$ diz respeito à localização global da *landmark* i .

O vector de estados pode possuir variáveis adicionais para a representação de erros dos sensores e outras grandezas de interesse.

A matriz de covariância no instante k será:

$$P(k) = \begin{bmatrix} P_{vv}(k) & P_{v1}(k) & \cdots & P_{vN}(k) \\ P_{1v}(k) & P_{11}(k) & \cdots & P_{1N}(k) \\ P_{2v}(k) & P_{21}(k) & \cdots & P_{2N}(k) \\ \vdots & \vdots & \ddots & \vdots \\ P_{Nv}(k) & P_{N1}(k) & \cdots & P_{NN}(k) \end{bmatrix} \quad (5.5)$$

- Onde v representa os estados associados ao veículo e $[1, \dots, N]$ agrupa os estados das *landmarks*.

Os elementos da diagonal da matriz de covariância contêm a variância dos erros dos estados individuais. Os elementos fora da diagonal estabelecem uma medida de correlação entre os elementos correspondentes. As correlações são importantes, pois permitem a estimação indirecta de estados a partir de medições relativas a estados correlacionados. As correlações surgem naturalmente com a execução do algoritmo a partir das relações estabelecidas pela matriz transição de estado ou por intermédio das medidas consideradas.

• Fase de Previsão

Na fase de previsão ocorre a propagação do estado anterior, (k) , para o instante de tempo seguinte, $(k + 1)$. Nesta etapa é utilizado um modelo de transição de estado não-linear, que descreve a evolução do sistema ao longo do tempo. Para pontos de referencia estacionários o modelo atribui dinâmica apenas ao veículo, com base nas entradas de controlo, $\mu(k)$, perturbadas por ruído gaussiano e média zero, $v(k)$ (fórmula 5.6):

$$F\left(x(k-1), \mu(k), v(k)\right) = \begin{bmatrix} F_v\left(x_v(k-1), \mu(k), v(k)\right) \\ 0 \\ \vdots \end{bmatrix} \quad (5.6)$$

– Onde F_v corresponde ao modelo de transição de estado.

Na fase de previsão o valor dos estados é calculado de acordo com o modelo do transição de estado:

$$\hat{x}(k|k-1) = \begin{bmatrix} \hat{x}_v(k|k-1) \\ \hat{l}_i(k-1|k-1) \\ \vdots \\ \hat{l}_N(k-1|k-1) \end{bmatrix} = \begin{bmatrix} F_v\left(\hat{x}_v(k-1|k-1), \mu(k), v(k)\right) \\ \hat{l}_i(k-1|k-1) \\ \vdots \\ \hat{l}_N(k-1|k-1) \end{bmatrix} \quad (5.7)$$

O cálculo da matriz de covariância implica a linearização do modelo de transição de estado em torno dos estados actuais, pelo cálculo das derivadas parciais relativamente aos estados, $\nabla F(k)$.

$$P(k|k-1) = \nabla F(k) * P(k-1|k-1) * \nabla^T F(k) + Q \quad (5.8)$$

– Onde Q regista a variância dos erros associados ao modelo de transição de estado.

Assumindo alvos estacionários, o ruído do processo não se manifesta nos estados das *landmarks*, por esse motivo a matriz Q é nula para estados não relacionados com a localização do veículo (5.9):

$$Q = \begin{bmatrix} Q_v & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \quad (5.9)$$

• Fase de *Update*

No instante em que um alvo do mapa é observado ocorre a actualização do sistema, através das equações de update do EKF. Cada observação, $z(k)$, proporciona ao sistema um acréscimo de informação – inovação – que consiste na diferença entre a medida produzida pelos sensores e a medida prevista, tendo em conta o estado actual do sistema. A medida prevista é obtida pela aplicação do modelo de observação h , responsável pela projecção do valor dos estados no espaço das observações.

$$\text{inovação} = z(k) - h(\hat{x}(k|k-1)) \quad (5.10)$$

A covariância associada à observação, S , depende da linearização do modelo de observação em torno do estado actual, conseguida pelo cálculo das derivadas do modelo de observação em ordem aos estados do sistema, indicadas por $\nabla H(k)$. A matriz S é calculada do seguinte modo:

$$S(k) = \nabla H(k) * P(k|k-1) * \nabla^T H(k) + R(k) \quad (5.11)$$

– Onde R representa a matriz com os valores de variância associados à medida.

O algoritmo prossegue com o cálculo da matriz $W(h)$, conhecida como ganho de Kalman.

$$W(k) = P(k|k-1) * \nabla^T H(k) * S^{-1}(k) \quad (5.12)$$

Por fim, é calculada a correção dos estados e atualizada a matriz de covariância.

$$\hat{x}(k|k) = \hat{x}(k|k-1) + W(k) * \left[z(k) - H(\hat{x}(k|k-1)) \right] \quad (5.13)$$

$$P(k|k) = P(k|k-1) - W(k) * S(k) * W(k)^T \quad (5.14)$$

Propriedades

- Assumindo que os sensores se encontram a bordo do robô todas as observações traduzem medições relativas entre o veículo e os alvos, logo a construção de um mapa por si só não fornece informações de posicionamento absolutas. A única informação absoluta conhecida pelo sistema é a especificada através das condições iniciais. Por este motivo, a incerteza na posição global das *landmarks* nunca será inferior à definida no instante inicial para a localização do veículo;
- Como os alvos não possuem dinâmica, os erros do modelo de transição de estado não influenciam os estados das *landmarks*, logo a covariância a eles associada só se modifica na fase de update. Visto que na etapa de *update* os valores de covariância se mantêm ou diminuem, a incerteza na posição dos alvos nunca aumenta;
- Teoricamente, com a evolução do tempo e a sucessão de observações, o grau de incerteza na estimação da posição dos alvos tende para o mesmo valor, resultando num mapa com elevada correlação entre os vários elementos.

5.3.1.2 Consistência dos Pontos Observados

As implementações tradicionais com EKF requerem um conjunto de *features* bem distintas e facilmente identificáveis. A diferenciação entre os alvos no mapa deve ser suficientemente clara para que os pontos de interesse não sejam confundidos ao nível do mecanismo de associação de dados, o que pode levar a uma falha no algoritmo visto que o EKF só tolera ruídos gaussianos [93]. Por este motivo as abordagens segundo filtro de Kalman estendido são forçadas a ignorar uma grande quantidade de informações sensoriais de forma a reduzir a ambiguidade de associação entre alvos. Como consequência os mapas construídos carecem de descrições geometricamente detalhadas do ambiente.

5.3.1.3 Requisitos Computacionais

Considerando k o número de pontos de referência, representados no mapa pela sua posição global tridimensional, e tendo em conta que a localização tridimensional do veículo é definida em seis graus de liberdade, o vector de estados do filtro terá uma dimensão de $(3 * k + 6)$, enquanto que a matriz de covariância cresce de forma quadrática relativamente ao número de alvos no mapa $(3 * k + 6)^2$. Por outro lado, cada vez que surge uma observação relativa a um elemento do mapa, é necessário processar a etapa de *update*. Daqui se depreende que a introdução de pontos no mapa eleva o custo computacional quadraticamente [94], com impacto mais vincado na fase de *update*. Este facto fundamenta os esforços dispendidos na busca de soluções mais eficientes como [95], onde o processo de SLAM não se desenvolve na totalidade do mapa, considerando apenas uma vizinhança em redor da localização estimada do veículo.

A partição do mundo em sub-regiões com pouca correlação também constitui um método promissor para superar o aumento das exigências computacionais [96]. O ambiente é dividido em múltiplos sub mapas que representam regiões referenciadas globalmente, em que cada porção possui a sua própria estimação de posição do veículo, um conjunto de *features* com a respectiva estimação de localização e uma matriz de covariância. As sub-regiões sobrepõem-se ligeiramente para evitar excessivas transições entre sub mapas. O tamanho e localização de cada sub mapa são especificados à priori de acordo com a densidade de alvos espectável. Caso o robô visite áreas para as quais não existe um sub mapa, este deverá ser criado para guardar as informações dessa zona. Se o veículo voltar a uma zona já inspeccionada é carregado o sub mapa previamente construído e, em conjunto com as informações do sub mapa da zona anterior, a posição do robô é melhorada. Deste modo, apenas os alvos mais próximos do robô são utilizados no processo de localização e mapeamento, reduzindo a complexidade computacional do mecanismo de estimação.

5.3.2 SLAM Baseado em Filtro de Partículas

Os filtros de partículas caracterizam-se pela utilização de um conjunto de partículas (também designadas por amostras ou pontos de suporte) posicionadas de forma aleatória no espaço dos estados, a cada uma das quais é associado um peso que reflecte a probabilidade com que a partícula aproxima o valor dos estados. O peso conferido a cada partícula é determinado pelo modelo de observação, que compara a disparidade entre a localização

da amostra e o valor que resulta da projecção de uma observação no espaço de estados. As partículas, através da posição e peso, representam uma aproximação da distribuição manifestada pelas grandezas em análise. Este método é capaz de lidar com distribuições não gaussianas e multimodais e para um número de partículas suficientemente elevado a função densidade posterior aproxima-se da estimativa óptima [97].

A complexidade computacional nos filtros de partículas acompanha de forma proporcional o aumento do número de amostras, facto que invalida a aplicação destas técnicas à problemática de SLAM, pois a necessidade de um conjunto de partículas para representação de cada *landmark* obriga a uma enormidade de amostras mesmo para mapas pouco complexos. Esta limitação é ultrapassada em [98] pela utilização de um filtro de partículas *Rao-Blackwellized* (RBPF), que se baseia no facto de a estimação das *landmarks* ser condicionalmente independente dada a trajectória do veículo. Esta circunstância permite factorizar de forma exacta a distribuição posterior de SLAM de modo a isolar a estimação da trajectória da estimação do mapa [99].

O método FastSLAM [100] é um algoritmo de referência na abordagem ao problema de SLAM através do filtro RBPF. A solução FastSLAM decompõe o problema de SLAM num problema de localização e um conjunto de problemas de estimação de *landmarks* condicionados pela localização estimada. A estimativa da localização do veículo é executada por intermédio de um filtro de partículas, onde cada amostra representa uma possível trajectória e possui o seu próprio mapa. A estimação da posição de cada *landmark* é executada através de um EKF dedicado, o que obriga a $(P * M)$ EKFs para estimar a totalidade das *landmarks* nos mapas de todas as partículas, onde P refere o número de partículas e M expressa a dimensão do mapa. Neste princípio destacam-se duas grandes vantagens, designadamente a redução da complexidade computacional resultante do emprego de filtros de menor dimensão e a possibilidade de cada partícula desenvolver uma etapa de associação de dados autónoma.

A incorporação de observações e de informações de movimento da plataforma no processo FastSLAM desenrola-se recursivamente em quatro etapas. Na primeira fase a posição das partículas é actualizada com base na posição anterior acrescida do controlo aplicado ao veículo. De seguida é estimada a localização das *landmarks* correspondentes às observações adquiridas no último instante de tempo, pela actualização dos filtros de Kalman estendidos que representam as *landmarks* observadas. A atribuição de uma observação a

uma determinada *landmark* é governada pelo método de associação de dados, que pode produzir correspondências diferentes em cada partícula, dado que a localização do ponto de suporte tem influência na correspondência. Posteriormente é conferido um peso às partículas, peso esse que é indirectamente proporcional à diferença entre o mapa previsto e o mapa resultante da actualização pelas observações. Por último é desencadeado o processo de *resampling*, através do qual se combate o fenómeno designado por degeneração das amostras.

O método FastSLAM incorpora uma estratégia baseada em árvores binárias, que permite atingir uma performance superior na etapa de actualização da posição das *landmarks*. Na generalidade do algoritmo é atingida uma complexidade computacional da ordem de $O(P \log M)$. As soluções de SLAM deste tipo caracterizam-se por uma amostragem muito esparsa do espaço de estados, visto que a proporção entre o número de partículas e a dimensão do espaço é bastante baixa. Este facto compromete a convergência do método especialmente perante uma má escolha da distribuição e número de partículas [101] e a utilização de modelos de movimentação com pouca precisão [102].

5.3.3 SLAM por Registo de Varrimentos Laser

No âmbito de SLAM baseado em registo de varrimentos laser identificam-se duas vertentes de estudo dependentes do número de dimensões espaciais representadas pelos respectivos varrimentos – assunto previamente abordado no capítulo do estado da arte (2.2). Uma das características que diferencia os métodos de SLAM dos demais algoritmos de localização incrementais diz respeito à capacidade de diminuição significativa da incerteza no mapa e na localização perante a identificação de uma situação de *loop closure*. Esta situação ocorre no instante em que o veículo visita uma área já explorada, da qual constam informações no mapa. Um sistema de localização baseado em registo de varrimentos não possui a capacidade de identificação deste tipo de situação, logo é incapaz de estabelecer correspondências entre as observações obtidas nesse instante e a porção do mapa que representa essa região. As potenciais informações, resultantes das correspondências determinadas na situação de *loop closure*, permitem a correcção dos erros acumulados na estimação ao longo do percurso e contribuem para a construção de um mapa coerente. Neste contexto a designação de SLAM por registo de varrimentos consiste na combinação entre a capacidade de localização incremental, proporcionada pelo registo de varrimentos,

com uma estratégia de identificação de situações de *loop closure*, que permita efectuar a correcção dos erros acumulados. Os estudos propostos nas referências [103], [16] apresentam algoritmos de SLAM por registo de varrimentos de sensores LRF.

Geralmente os túneis rodoviários ou ferroviários são constituídos por uma única galeria, ao longo da qual é impraticável a inversão do sentido de marcha, pelo que é fisicamente impossível retornar a uma zona anteriormente explorada, logo a inclusão de uma estratégia de *loop closure* seria inútil. Assim, na aplicação alvo desta dissertação a utilização de varrimentos de sensores LRF no processo de localização obriga apenas à implementação de um algoritmo de registo.

5.3.4 Visual SLAM

Ao longo das últimas duas décadas os sistemas de visão computacional têm sido introduzidos com sucesso nos processos de localização. A quantidade e diversidade de informações captadas pelas câmaras, a disponibilidade de algoritmos robustos de tratamento dos dados, a acessibilidade a sensores com custos reduzidos e baixos consumos de energia são factores preponderantes para o elevado interesse suscitado nesta área.

No âmbito do paradigma de SLAM os sistemas de visão computacional constituem um meio eficaz de obtenção de pontos de referências. Um aspecto determinante na arquitectura do sistema prende-se com o número de sistemas ópticos utilizados. Caso se opte por um sistema monocular, as *features* de uma dada *frame* fornecem exclusivamente informações de orientação (*bearing feature*), o que requer precauções adicionais especialmente aquando da sua inicialização. Em sistemas com duas ou mais câmaras é viável o cálculo das coordenadas tridimensionais das *features*, processo que requer uma boa calibração do sistema de visão e obriga à correspondência entre *features* nas várias imagens. Quanto aos métodos de estimação, a introdução de *features* visuais não produz qualquer tipo de condicionantes específicas, podendo ser utilizados os métodos anteriormente referidos.

5.3.4.1 Visual SLAM Monocular – Parametrização *Inverse Depth*

A imagem obtida por uma câmara monocular resulta da projecção de pontos tridimensionais num plano bidimensional – plano da imagem. Como demonstrado na secção (4.3), a determinação da posição de um ponto no plano da imagem é exequível segundo o princípio da projecção perspectiva, que relaciona a posição do ponto com a posição da

câmara e seus parâmetros intrínsecos. Porém a operação inversa de cálculo da posição tridimensional do ponto não é possível pela aplicação da operação de projecção perspectiva inversa, dado que não existem informações sobre a distância entre o ponto e a câmara.

Para inferir a posição tridimensional de um ponto é necessário analisar a sua posição na imagem ao longo de uma sequência de imagens adquiridas de diferentes posições. Cada imagem permite traçar um segmento de recta 3D semi-infinito (*3D semi-infinite ray*) que intersecta a *feature* tendo como origem o centro óptico da câmara. Pela intersecção dos segmentos relativos a cada posição da câmara obtém-se uma estimativa da posição tridimensional da *feature* (figura 5.4). A quantidade de imagens a utilizar neste processo depende do grau de paralaxe, que consiste no maior ângulo entre os raios semi-infinitos dos vários instantes de amostragem. Como é facilmente perceptível a ambiguidade em profundidade reduz-se à medida que aumenta a paralaxe e diminui a distância entre o ponto e o sensor.

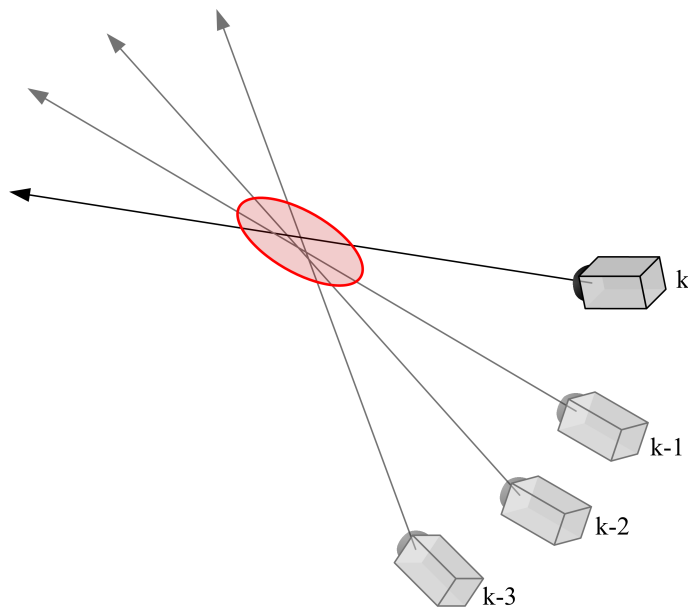


Figura 5.4: Determinação das coordenadas tridimensionais de um ponto pela intercepção de segmentos de recta semi-infinitos observados ao longo do tempo. Esta operação é potenciada pela paralaxe da câmara.

A impossibilidade de cálculo da posição de uma *landmark* pela análise de uma única frame origina um problema de inicialização da estimação, pois nos métodos de estimação empregues nos sistemas de SLAM a inicialização de uma *landmark* está dependente da introdução de uma aproximação inicial da posição tridimensional. Uma forma de contor-

nar este problema passa pela não inclusão imediata das *landmarks* no mapa, estas são canalizadas para um processo intermédio com vista à acumulação de informação ao longo de várias *frames*, até que se verifique um grau de paralaxe que possibilite o calculo da posição, procedendo-se posteriormente à sua inicialização no método de estimação.

Davison [104] opta pela representação inicial das *landmarks* por intermédio de uma linha 3D semi-infinita, com inicio na posição estimada da câmara e direcção concordante com a indicada pela posição da feature na imagem. Sobre o segmento de recta, até a uma distância inferior a cinco metros, é lançado um filtro de partículas uni-dimensional, que é actualizado até se verificar um nível de convergência tal que permita a representação da posição da *landmark* por uma gaussiana. Após este ciclo de estimação de posição a *landmark* é adicionada ao mapa, gerido de acordo com as regras da abordagem EKF SLAM. Esta estratégia não responde eficazmente em ambientes de larga escala, dado que seria necessário lançar partículas a distâncias superiores a cinco metros. Na perspectiva de SLAM com fundamento no filtro RBPF, a inicialização por intermédio de linhas causa um aumento de complexidade suplementar, que deriva da necessidade de replicar o processo em cada partícula [105]. Genericamente, nas estratégias de inicialização retardada (*delayed initialization*) como [104], as features em espera encontram-se fora do ciclo de estimação, logo não contribuem para o processo de SLAM, para além disso podem ser sujeitas a pouca paralaxe ao longo de um grande número de imagens, existindo a possibilidade de nunca serem introduzidas no mapa.

Em [106] e [107] é proposta uma solução mais eficiente que consiste na representação de *features* sob a forma de profundidade inversa (*Inverse Depth Parametrization*). A parametrização em profundidade inversa admite a aproximação por uma gaussiana, logo uma *feature* definida neste formato pode ser introduzida directamente em algoritmos de SLAM baseados em EKF. O vector de estados de uma solução deste género (5.15) possui a parcela x_v relativa aos estados da câmara bem como as componentes $(y_1, y_2, \dots, y_n)^T$, onde cada qual identifica uma *feature*.

$$x = [x_v, y_1, y_2, \dots, y_n]^T \quad (5.15)$$

O vector x_v possui 13 componentes (5.16), respectivamente posição da câmara relativamente ao referencial global r^{wc} (3 estados), orientação da câmara relativamente ao referencial global q^{wc} (definida por quatérnios – 4 estados), velocidade linear da câmara

v^w (3 estados) e velocidade angular da câmara ω^W (3 estados).

$$x_v = [r^{wc}, q^{wc}, v^w, \omega^w]^T \quad (5.16)$$

Uma *feature* na forma de profundidade inversa é representada por seis parâmetros (5.17): a posição da câmara no primeiro instante de visualização (x_{vi}, y_{vi}, z_{vi}) , a orientação do segmento de recta semi-infinito definido pelas componentes (θ_i, ϕ_i) , respectivamente ângulo de azimute e ângulo de elevação, e finalmente o inverso da profundidade ($\rho_i = 1/d_i$) da *feature* ao longo do segmento de recta.

$$y_i = [x_{vi}, y_{vi}, z_{vi}, \theta_i, \phi_i, \rho_i]^T \quad (5.17)$$

O vector y_i codifica, segundo o referencial global, o segmento de recta semi-infinito do instante em que ocorre a primeira observação da *feature* (figura 5.5).

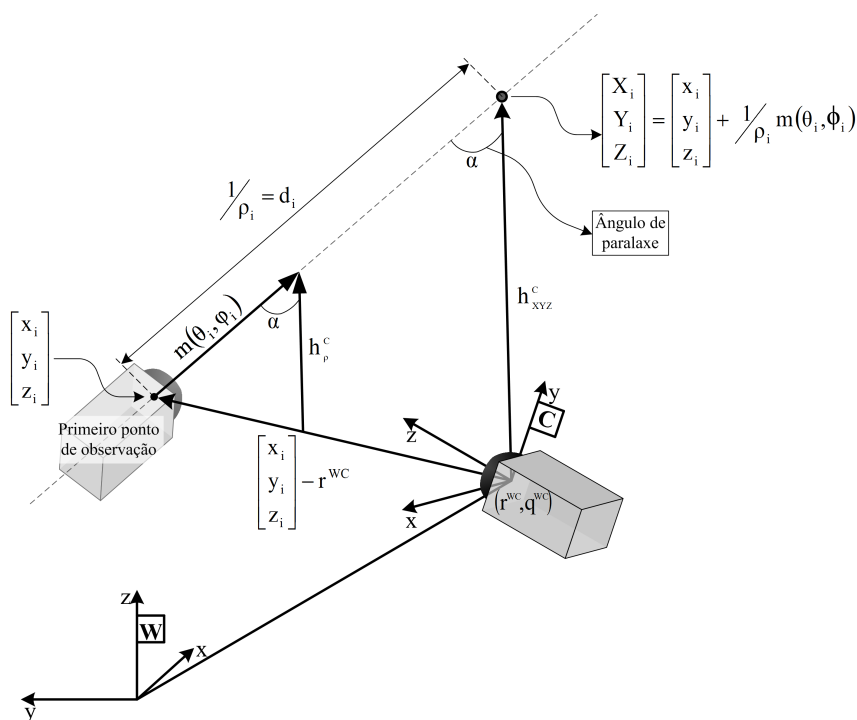


Figura 5.5: Princípio da parametrização *inverse depth*.

As coordenadas Euclidianas (X_i, Y_i, Z_i) do ponto no referencial do mundo relacionam-se com os parâmetros de y_i de acordo com a equação (5.18).

$$\mathbf{x}_i = \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} + \frac{1}{\rho_i} m(\theta_i, \phi_i) \quad (5.18)$$

- Onde:

$$m(\theta_i, \phi_i) = \begin{bmatrix} \cos(\phi_i) \sin(\theta_i) \\ -\sin(\phi_i) \\ \cos(\phi_i) \cos(\theta_i) \end{bmatrix} \quad (5.19)$$

• Modelo de Observação

O modelo de observação corresponde à projecção na imagem de uma *landmark* definida no referencial global, operação que pode ser decomposta em três passos: a projecção da *landmark* no referencial da câmara, a aplicação do modelo de distorção e finalmente a determinação das coordenadas na imagem pela aplicação dos parâmetros intrínsecos.

A projecção no referencial da câmara é expressa por intermédio do vector direcção tridimensional $h^c = [h_x, h_y, h_z]^T$. Para *landmarks* codificadas em coordenadas euclidianas o vector direcção é calculado de acordo com a fórmula (5.20).

$$h^c = h_{XYZ}^c = \mathbf{R}(q^{wc})^{-1} \left(\begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} - r^{wc} \right) \quad (5.20)$$

Para features em *inverse depth* o vector direcção h^c obtém-se pela fórmula (5.21).

$$h^c = h_\rho^c = \mathbf{R}(q^{wc})^{-1} \left(\rho_i \left(\begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} - r^{wc} \right) + m(\theta_i, \phi_i) \right) \quad (5.21)$$

- $\mathbf{R}(q^{wc})^{-1}$ refere-se à matriz rotação do referencial da câmara para o referencial do mundo.

Como a câmara não observa directamente o vector direcção h^c , mas sim a sua projecção no plano da imagem, é necessário aplicar o modelo de distorção seguido dos parâmetros intrínsecos, para determinar a posição em píxeis do ponto observado. Este procedimento

é levado a cabo por uma sequência de operações começando pela normalização do vector direcção (fórmula 5.22).

$$h_u^c = \begin{bmatrix} \frac{h_x^c}{h_z^c} \\ \frac{h_y^c}{h_z^c} \end{bmatrix} \quad (5.22)$$

Em seguida dá-se a correcção da distorção (fórmula 5.23).

$$\begin{bmatrix} h_{d_x}^c \\ h_{d_y}^c \end{bmatrix} = \begin{bmatrix} h_{u_x}^c \\ h_{u_y}^c \end{bmatrix} \left(1 + \kappa_1 r_u^2 + \kappa_2 r_u^4 + \kappa_5 r_u^6 \right) \quad (5.23)$$

Por fim as coordenadas em píxeis são obtidas (fórmula 5.24).

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} h_{d_x}^c f_x k_u + u_0 \\ h_{d_y}^c f_y k_v + v_0 \end{bmatrix} \quad (5.24)$$

• Inicialização de *Landmarks*

A parametrização de uma *landmark* aquando da sua inserção no vector de estados depende da localização estimada da câmara (\hat{x}^{wc} , \hat{q}^{wc}), da observação da *feature* ($h = (u, v)^T$), e dos parâmetros que caracterizam a distribuição de probabilidade associada à profundidade (ρ_0 , σ_ρ).

$$\hat{y}(\hat{x}^{wc}, \hat{q}^{wc}, h, \rho_0) = \left[\hat{x}_i \ \hat{y}_i \ \hat{z}_i \ \hat{\theta}_i \ \hat{\phi}_i \ \hat{\rho}_i \right]^T \quad (5.25)$$

O ponto inicial do segmento de recta semi-infinito é dado pela posição da câmara.

$$\text{origem do raio} = \begin{bmatrix} \hat{x}_i \\ \hat{y}_i \\ \hat{z}_i \end{bmatrix} = \hat{r}^{wc} \quad (5.26)$$

Os ângulos de orientação do segmento de recta semi-infinito tem que ser determinados, o que é conseguido pela aplicação do inverso do modelo de observação. Sendo assim, a primeira etapa diz respeito à aplicação do inverso dos parâmetros intrínsecos às coordenadas da *feature*, de modo a obter as coordenadas distorcidas no referencial da câmara.

$$h_d^c = \begin{bmatrix} \frac{(u_d - u_0)}{f_x k_u} \\ \frac{(v_d - v_0)}{f_y k_v} \end{bmatrix} \quad (5.27)$$

Na etapa seguinte ocorre a remoção da distorção, empregando o inverso do modelo de distorção.

$$h_u^c = \begin{bmatrix} \frac{h_{d_x}^c}{1 + \kappa_1 r_u^2 + \kappa_2 r_u^4 + \kappa_5 r_u^6} \\ \frac{h_{d_y}^c}{1 + \kappa_1 r_u^2 + \kappa_2 r_u^4 + \kappa_5 r_u^6} \end{bmatrix} \quad (5.28)$$

O parâmetro r_d é facilmente calculado por (5.29), porém a determinação de r_u não é imediata, devido ao desconhecimento das coordenadas dos pontos sem distorção. O valor de r_u é obtido com base em (5.30) pela aproximação segundo o método de Newton-Raphson.

$$r_d = \sqrt{x_d^2 + y_d^2} \quad (5.29)$$

$$r_d = r_u (1 + \kappa_1 r_u^2 + \kappa_2 r_u^4 + \kappa_5 r_u^6) \quad (5.30)$$

Por último, o segmento de recta semi-infinito é transformado para o referencial global de acordo com a fórmula (5.31), finalmente os ângulos de azimute e elevação podem ser extraídos por (5.32).

$$h^w = \mathbf{R}(\hat{q}^{wc}) \begin{bmatrix} h_{u_x}^c \\ h_{u_y}^c \\ 1 \end{bmatrix} \quad (5.31)$$

$$\begin{bmatrix} \theta_i \\ \phi_i \end{bmatrix} = \begin{bmatrix} \arctan(h_x^w, h_z^w) \\ \arctan\left(-h_y^w, \sqrt{h_x^w{}^2 + h_z^w{}^2}\right) \end{bmatrix} \quad (5.32)$$

A covariância dos parâmetros \hat{x}_i \hat{y}_i \hat{z}_i $\hat{\theta}_i$ e $\hat{\phi}_i$ é baseada na covariância do erro da medida (R_i) e na covariância estimada dos estados $P(k|k)$. Quanto à profundidade inversa inicial (σ_ρ) e ao desvio padrão associado (ρ_0) são definidos empiricamente, de modo que 95% da região de confiança cubra uma gama de profundidades desde infinito até zonas muito

próximas da câmara². A covariância dos estados após a inicialização de uma nova *landmark* é obtida do seguinte modo:

$$P(k|k)_{nova} = J \begin{bmatrix} P(k|k) & 0 & 0 \\ 0 & R_i & 0 \\ 0 & 0 & \sigma_{\rho_0}^2 \end{bmatrix} J^T \quad (5.33)$$

- Onde J representa o jacobiano do inverso do modelo de observação em relação aos estados do mapa.

- **Conversão da parametrização em *inverse depth* para codificação euclidiana**

O maior inconveniente da parametrização em profundidade inversa diz respeito à duplicação do número de estados necessários à representação de uma *landmark* no mapa, o que obriga a um esforço computacional suplementar. Por isso, logo que a incerteza na profundidade da *landmark* se torna gaussiana é conveniente a substituição da parametrização em profundidade inversa pela codificação em coordenadas Euclidianas. Nesse sentido, após a etapa de *update*, para cada *landmark* em profundidade inversa no mapa, é calculado o índice de linearidade L_d .

$$L_d = \frac{4\sigma_d}{d_i} |\cos\alpha| \quad (5.34)$$

- Na equação σ_d resulta da divisão $\frac{\sigma_\rho}{\sigma_i^2}$, sendo $\sigma_\rho = \sqrt{P_{y_i y_i}}$ e $P_{y_i y_i}$ a submatriz de covariância com dimensão 6×6 associada à *landmark*. O termo $\cos\alpha$ é dado por $m^T h_{XYZ}^w \|h_{XYZ}^w\|^{-1}$, onde $h_{XYZ}^w = \hat{x}_i - \hat{r}^{wc}$ e o termo \hat{x}_i é obtido pela equação (5.18). Por último $d_i = \|h_{XYZ}^w\|$.

As *landmarks* com índice de linearidade inferior a um *threshold*³ são convertidas para codificação em coordenadas euclidianas através da equação (5.18). A matriz de covariância é actualizada na sua totalidade pela multiplicação do jacobiano J .

²Em [108] são utilizados os valores $\hat{\rho}_0 = 1$ e $\sigma_\rho = 1$, resultando numa gama de profundidades que vai desde o mínimo de 0.33 unidades de distância até ao infinito

³Em [106] são aceites as *landmarks* cujo índice de linearidade seja inferior a 0.1

$$P = JPJ^T, \quad J = \text{diag} \left(I, \frac{\partial x_i}{\partial y_i}, I \right) \quad (5.35)$$

- **Escala da trajectória e mapa**

Quando apenas uma câmara monocular é utilizada na estimação da trajectória e do mapa, surgem dificuldades na determinação da escala do cenário, na medida em que a distância entre a câmara e as *landmarks* não é observada [109].

Segundo a abordagem de inicialização de *landmarks* em *inverse depth*, as únicas informações métricas introduzidas no sistema consistem nos valores de profundidade inversa inicialmente atribuídos a cada *landmark*. Considerando que estes valores são na maior parte das vezes aproximações grosseiras, é espectável alguma ambiguidade na escala do mapa e localização.

Numa grande parte dos trabalhos, este problema é contornado pela introdução, no instante inicial, de medidas precisas sobre o movimento da câmara, ou pela visualização de um objecto alvo com dimensões conhecidas pelo sistema [104]. Contudo, dado que esta é a única informação métrica fidedigna, os desvios em escala tornam-se inevitáveis ao longo do tempo.

5.3.4.2 *Visual SLAM Stereo*

A opção por um sistema *stereo* ultrapassa as limitações dos sistemas monoculares, na medida em que possibilita a determinação das coordenadas tridimensionais das *landmarks*, permitindo a especificação da trajectória segundo o sistema métrico e ultrapassando ao mesmo tempo as dificuldades de inicialização. A precisão no cálculo de distâncias em sistemas *stereo* degrada-se à medida que as câmaras se afastam do ponto no mundo. Quanto maior for a distância entre as ópticas (*baseline*) maior será a tolerância ao aumento do espaço que separa o ponto das câmaras. Contudo, todos os sistemas *stereo* possuem um limite de distância a partir do qual o cálculo da profundidade é afectado por uma grande incerteza. Assim para pontos distantes o sistema *stereo* comporta-se de forma semelhante a um dispositivo monocular [110].

No interior de um túnel é previsível a existência de *landmarks* distantes, logo será proveitoso, nomeadamente para efeitos de correcção de atitude, empregar um sistema de

visão *stereo* aliado a um algoritmo de SLAM capaz de conciliar no mesmo mapa *landmarks* tridimensionais e pontos de referência distantes definidos somente pela orientação. Este assunto é explorado em [110], estudo onde são definidas duas regiões utilizadas para classificar as *landmarks* quanto à sua proximidade relativamente às câmaras. Durante o processo de detecção e seguimento de features as imagens da câmara direita são consideradas como referência na inicialização de *landmarks*. Os pontos de interesse – neste caso uma variação dos cantos de Harris [111] – são extraídos na imagem da direita e classificados de acordo com a disparidade relativamente à imagem da esquerda. Os pontos considerados próximos pela análise da disparidade são inicializados como *landmarks* com posição tridimensional conhecida, caso contrário é-lhe atribuída a parametrização de *inverse depth* seguida da inicialização de acordo com a sua projecção na imagem da direita. Com vista ao melhoramento da eficiência do mecanismo de seguimento das *landmarks*, a posição das *features* no instante seguinte é prevista de acordo com a movimentação estimada da câmara, sendo atribuído a cada ponto de interesse uma região da imagem cujos limites dependem da incerteza nos parâmetros da *feature* e da incerteza na movimentação da câmara.

5.4 Sistemas Multirobóticos de Localização Cooperativa

As pesquisas no domínio dos sistemas multirobóticos incidem sobre estratégias de coordenação de conjuntos de robôs móveis tendo em vista a redução do tempo necessário à realização das tarefas, o aumento da precisão com que são executadas e a possibilidade de atribuição de uma especialidade a cada robô, o que resulta num acréscimo de robustez, pois a avaria de um dos elementos não põe em causa a totalidade da missão, ao contrário do que sucederia com a falha de um único robô dotado de todas as competências.

Tratando-se de grupos de robôs móveis o conhecimento da localização relativa entre os membros do grupo é um aspecto crucial para o sucesso das missões, dado que esta informação serve de base a processos importantes como a negociação de tarefas e o planeamento de trajectórias. Algumas investigações na área de localização multirobótica cooperativa exploram o conceito de referência móvel (*portable landmark*), nestes casos cada robô do grupo serve de ponto de referência global nos processos de localização dos restantes elementos. O uso de referências móveis aliado a estratégias de movimentação

coordenadas permite a estimação precisa da localização em cenários onde esta tarefa é dificultada pela ausência de pontos de referência naturais.

Em [112] o grupo de robôs é dividido em duas equipas, uma movimenta-se enquanto a outra mantém uma posição estática servindo de referência. No instante seguinte os papéis invertem-se e o processo repete-se até que ambas as equipas atinjam o objectivo. Já em [113] apenas um robô se move de cada vez, ao passo que o resto do grupo forma um triângulo equilátero de modo a providenciar pontos de referência com informação geométrica.

A opção por uma estratégia semelhante à proposta em [112] pode ser vantajosa num processo de localização dentro de um túnel, visto que a introdução de uma referência móvel à frente do veículo de mapeamento permite ultrapassar a ambiguidade de posição segundo a direcção de movimento. Porém esta estratégia impõe algumas condicionantes como a obrigatoriedade de imobilização periódica do veículo de mapeamento e a necessidade de conservação do contacto visual entre ambos os intervenientes.

Esta página foi intencionalmente deixada em branco.

Capítulo 6

Representação de Modelos 3D

Conteúdo

6.1	Nuvens de Pontos	105
6.2	Malhas Poligonais	106

Um sistema de modelização tridimensional deve possuir uma etapa de organização dos dados de modo a torná-los visualmente apresentáveis. Esta tarefa não é trivial, pois na maioria dos casos as informações tridimensionais são representadas sob a forma de estruturas bidimensionais, como por exemplo a impressão numa folha de papel ou a representação num monitor de computador. Consequentemente, é indispensável a aplicação de técnicas voltadas para a conversão de estruturas tridimensionais em imagens bidimensionais, processo designado por renderização. O processo de renderização não é objecto de estudo nesta dissertação, existindo uma grande variedade de programas dedicados a esta tarefa, todavia a forma de organização dos dados tridimensionais numa fase pré renderização toma especial relevância, na medida em que se relaciona directamente com o grau de realismo e detalhe empregues na representação final.

6.1 Nuvens de Pontos

À representação geométrica de um conjunto de pontos tridimensionais dá-se o nome de nuvem de pontos. O processo de construção de uma nuvem de pontos é bastante simples, pois não são formuladas relações entre os elementos do conjunto de pontos. Cada ponto é posicionado individualmente de acordo com as suas coordenadas face a um sistema referencial comum.



Figura 6.1: Exemplo de um modelo em nuvem de pontos.

A representação sob a forma de nuvem de pontos tem algumas vantagens, entre elas pode referir-se a facilidade de renderização (importante em conjuntos de dados com dimensão elevada); o facto de o processo de renderização incidir directamente nas coordenadas dos pontos, não sendo necessário qualquer tipo de pré tratamento ou organização dos dados; a compatibilidade entre os dados fornecidos por sensores LRF com o formato aceite pela aplicação de renderização é conseguida através de uma simples conversão de coordenadas polares para coordenadas cartesianas; adicionalmente, não ocorrem perdas de informação desde que a totalidade do conjunto seja representada.

Em contrapartida, devido à ausência de superfícies, perdem-se as noções de volumetria e de profundidade, bem como a capacidade de interpretação da figura à medida que aumenta a ampliação.

Outra desvantagem prende-se com o dispêndio computacional inerente à gestão de todos os pontos. Em situações específicas grande quantidade de pontos pode conter informações redundantes, como por exemplo em modelos constituídos essencialmente por superfícies planas.

6.2 Malhas Poligonais

As malhas poligonais descrevem o modelo através de polígonos convexos formados por três tipos de elementos: vértices, fronteiras e faces [114]. As malhas poligonais arbitrarias reúnem num único modelo vários tipos de polígonos, enquanto que as malhas poligonais regulares são formadas por apenas um tipo [115]. A representação de modelos tridimensionais de estruturas em larga escala é executada especialmente através de malhas poligonais triangulares, que proporcionam uma boa aproximação da maior parte das superfícies.

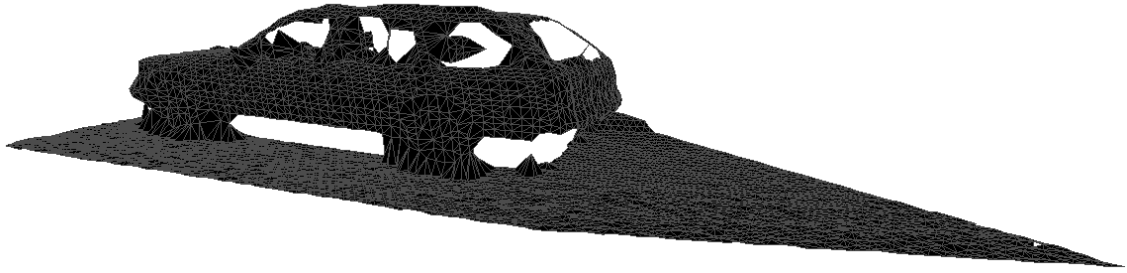


Figura 6.2: Exemplo de um modelo em malha poligonal.

No trabalho apresentado neste documento, são geradas malhas triangulares a partir de nuvens de pontos por intermédio do algoritmo BPA (*Ball-Pivoting Algorithm*) [26]. O algoritmo toma como entradas uma nuvem de pontos e um parâmetro representado pelo símbolo ρ_s , que especifica o raio da esfera utilizada na pesquisa de pontos vizinhos. A esfera deve possuir uma dimensão ajustada à densidade da nuvem de pontos, de modo que seja reduzida a probabilidade de atravessar as superfícies sem tocar em nenhum ponto.

Inicialmente são escolhidos três pontos, de forma aleatória, através dos quais se define o primeiro triângulo da malha, designado por triângulo semente. Em seguida a esfera de raio ρ_s é posicionada, mantendo contacto superficial com os três pontos do triângulo semente. A esfera é então rodada, em torno do eixo que intercepta dois dos pontos do triângulo, até estabelecer contacto com outro ponto não pertencente à malha. O ponto encontrado é conjugado com os dois pontos de base, formando um novo triângulo que integra imediatamente a malha. O processo repete-se para cada *edge* da malha com o propósito de formar mais triângulos e assim expandir a malha triangular. No caso de todos os *edges* terem sido verificados e de ainda existirem pontos da nuvem por interpolar, ocorre a selecção de um novo triângulo semente e o processo é reiniciado.

Esta página foi intencionalmente deixada em branco.

Capítulo 7

Aquisição de dados

Conteúdo

7.1	Plataforma Sensorial	110
7.1.1	IMU	110
7.1.2	Inclinómetro	111
7.1.3	Sensores LRF	112
7.1.4	Câmaras CCD	113
7.1.5	Disposição dos Sensores	113
7.1.6	Transformações entre os Sistemas de Coordenadas dos Sensores	115
7.1.7	Registo do Instante das Medidas	115
7.2	Aquisição do <i>Dataset</i>	116

Os sistemas de localização característicos dos robôs móveis são desenvolvidos com vista à operação em tempo real, pois o conhecimento da localização é imprescindível à maior parte dos processos de tomada de decisão. No âmbito desta dissertação a operação em tempo real não é crucial, e do ponto de vista funcional vislumbram-se algumas vantagens da opção por um algoritmo de localização em pós-processamento. O recurso a uma estratégia de pós-processamento alivia os requisitos computacionais da etapa de aquisição de dados, reduzindo a quantidade de hardware a bordo do veículo, bem como as necessidades energéticas do sistema. Com base nos dados recolhidos podem ser processados vários algoritmos de localização, tornando viável o estabelecimento de comparações entre as diferentes metodologias. Por fim, a construção de um dataset reduz o número de viagens do veículo de mapeamento, pois as diversas abordagens de localização podem ser testadas e afinadas com base nos dados recolhidos numa única viagem.

7.1 Plataforma Sensorial

Seguindo a opção por um sistema de localização pós-processado, foi desenvolvida uma plataforma sensorial vocacionada para o registo de informações adequadas aos mecanismos de localização e modelização tridimensional. A plataforma sensorial é composta por um sensor IMU, um inclinómetro, dois receptores GPS, duas câmaras CCD e dois sensores LRF. Os sensores foram instalados de modo a facilitar o transporte sobre um automóvel, possibilitando a recolha de informações no interior de túneis rodoviários.

A escolha do tipo de sensores e sua disposição é influenciada pelas estratégias analisadas nos capítulos estado da arte (capítulo 2) e localização (capítulo 5). Todos os dispositivos instalados na plataforma sensorial faziam já parte do recheio do LSA.

Em seguida são indicadas as características fundamentais de cada sensor, posteriormente é estabelecida a disposição rígida dos mesmos na plataforma sensorial.

7.1.1 IMU

O sistema inercial utilizado é fabricado pela empresa iMAR [82], modelo iNAV-FMS-E (figura 7.1). Consiste num sensor vocacionado para aplicações de navegação, pois fornece informações bastante precisas a uma taxa máxima de $400Hz$. Os giroscópios de fibra óptica apresentam um drift de $0.75^\circ/h$ ao passo que os acelerómetros possuem bias de $1.5mg$.

A informação pode ser fornecida em vários formatos, nomeadamente dados brutos (*raw data*), velocidade linear, velocidade angular, ângulo *roll*, ângulo *pitch*, ângulo *yaw*, posição e informações de GPS. A transmissão de dados pode ser executada através dos protocolos de comunicação Ethernet (TCP/IP ou UDP), CAN ou RS-232.

O sensor possui internamente um receptor GPS com frequência de amostragem de $5Hz$. O IMU possui algoritmos internos que em pós-processamento executam operações de filtragem baseadas em filtro de Kalman, conseguindo por este meio elevar a precisão da trajectória. A localização baseada exclusivamente em GPS proporciona um erro em orientação inferior a 0.1° e erro em posição até $2m$, para receptores DGPS os limites de erro baixam respectivamente para 0.03° e $1m$, em pós-processamento a integração entre IMU e RTK-GPS atinge erros inferiores a 0.01° em orientação e $10cm$ em posição. Dada a ausência de sinal GPS, durante o percurso no interior do túnel irá manifestar-se o erro dos

sensores inerciais, resultando numa trajectória com taxas de *random walk* em orientação de $0.1^\circ/\sqrt{h}$ e incerteza nas medições de aceleração na ordem de $50\mu g/\sqrt{Hz}$.



Figura 7.1: Sensor inercial iNAV-FMS-E.

7.1.2 Inclinómetro

Foi adicionado ao conjunto de sensores um inclinómetro de marca MicroStrain, modelo 3DM-GX1 [116]. O sensor é composto por três giroscópios que determinam as alterações dinâmicas de orientação, possui ainda um trio de acelerómetros e outro trio de magnetómetros para percepção da orientação estática.

Os dados podem tomar cinco formatos: ângulos de Heuler, quatérniões, matriz rotação, saída analógica com tensão proporcional aos ângulos de Heuler ou dados brutos de todos os sensores (*raw data*). A taxa de saída é de $100Hz$ para os quatro primeiros formatos e de $300Hz$ para a saída de dados brutos.

O 3DM-GX1 tem uma performance manifestamente inferior ao iNAV-FMS-E, os giroscópios registam um ângulo de *random walk* de $3.5^\circ/\sqrt{h}$, as medidas dos acelerómetros caracterizam-se por erros de $0.4mg/\sqrt{Hz}$. A comunicação de dados é estabelecida pelos protocolos RS-232 ou RS-485.



Figura 7.2: Inclinómetro 3DM-GX1.

7.1.3 Sensores LRF

Os sensores LRF proporcionam um varrimento bidimensional, são do fabricante SICK [117] e tomam a designação de *Laser Measurement System* LMS. Um é indicado para ambientes interiores (modelo LMS-200) enquanto que o outro (LMS-291) incorpora um sistema de correcção automático de interferências causadas por condições atmosféricas adversas, como chuva e nevoeiro, sendo por esse motivo adequado a operações em cenário exterior. Ambos emitem feixes *laser* na gama dos infra-vermelhos e facultam uma abertura angular de 180° com três gamas de resolução seleccionáveis – 0.25° , 0.5° ou 1° . A resolução métrica é de $10mm$ e os erros típicos situam-se entre os $\pm 15mm$ para o LMS-200 e entre os $\pm 35mm$ para o LMS-291. O alcance máximo para alvos com reflectividade reduzida (10%) fica limitado a 10 e 30 metros respectivamente para o LMS-200 e para o LMS-291. A frequência de amostragem é de $75Hz$ e a comunicação de dados é estabelecida pelos protocolos RS232 ou RS422.



Figura 7.3: Sensor *laser range finder* SICK LMS-200.

7.1.4 Câmaras CCD

O sistema inclui duas câmaras CCD do fabricante JAI [118], modelo CB-080GE. A resolução máxima é de $1032(h) \times 778(v)$ com *frame rate* de 30frames/s . O ganho é ajustável manualmente entre -30dB e $+12\text{dB}$. Estão à disposição pinos GPIO que, entre outras funções, permitem o disparo controlado por dispositivos externos. Na aquisição do dataset o instante de disparo das câmaras é comandado pelo IMU. A comunicação utilizada na transferência de informações das câmaras para a plataforma computacional respeita o protocolo *Gigabit Ethernet* (GigE).



Figura 7.4: Câmara JAI modelo CB-080GE.

7.1.5 Disposição dos Sensores

Os sensores foram instalados numa estrutura metálica rígida, susceptível de ser transportada no topo de um automóvel (figura 7.5).

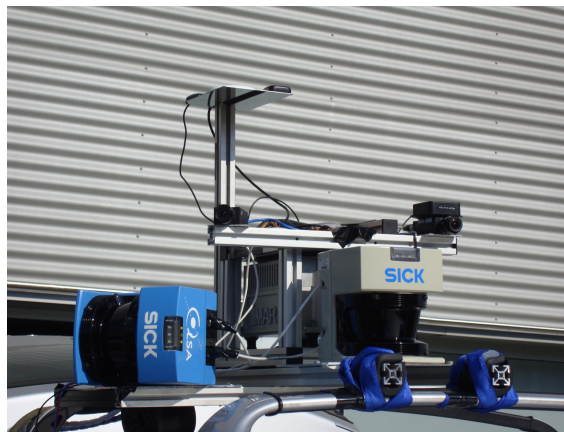


Figura 7.5: Plataforma sensorial transportada sobre o veículo de mapeamento.

O sensor iNAV-FMS-E foi posicionado com o eixo x na direcção do movimento, o eixo y aponta para a esquerda do veículo e o eixo z aponta para cima. O sensor LRF com maior alcance (LMS-291) encontra-se em frente do iNAV-FMS-E e o varrimento desenvolve-se

sobre o plano horizontal. O sensor LMS-200 foi colocado à direita do iNAV-FMS-E e está rodado -90° segundo *roll* e -40° segundo *yaw*, de forma a captar um perfil vertical da parede do túnel. As câmaras encontram-se separadas por uma *baseline* de 41.5cm e entre elas manifesta-se uma diferença de orientação de 25° . O sensor 3DM-GX1 encontra-se sobre a câmara da esquerda, o eixo *x* aponta para a retaguarda, o eixo *y* aponta para a esquerda e o eixo *z* no sentido do centro da terra.

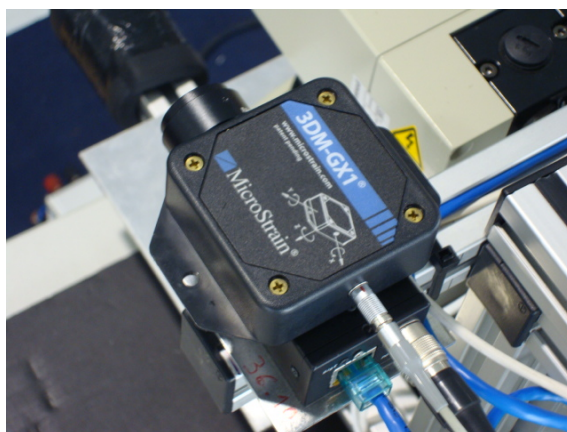


Figura 7.6: Grande plano do sensor 3DM-GX1 posicionado sobre a câmara da esquerda.

As câmaras visualizam uma porção comum do mundo, logo é possível correr algoritmos de visão *stereo* sobre esta área. A inclinação para a lateral do veículo da câmara direita tem duas justificações, por um lado a vista lateral alargada permite observar *features* com paralaxe mais acentuada, outra vantagem prende-se com o registo na imagem da zona de incidência do sensor LMS-200, tornando possível o preenchimento do modelo com informação visual.

O sensor LMS-200 através do varrimento no plano vertical apreende a geometria da parede direita bem como parte do tecto. Estes dados não são suficientes para construir um modelo tridimensional da estrutura da galeria na sua totalidade, no entanto a informação recolhida permite testar e desenvolver algoritmos voltados para a construção de nuvens de pontos e para a criação de malhas poligonais.

O sensor LMS-291 aponta em frente e regista um plano de varrimento horizontal. Os dados por ele adquiridos permitem testar o registo de varrimentos 2D, para perceber a viabilidade desta estratégia aplicada à localização.

7.1.6 Transformações entre os Sistemas de Coordenadas dos Sensores

A fusão correcta das informações dos vários sensores obriga à transposição das medidas para um sistema de coordenadas comum, neste sentido é obrigatório conhecer as transformações físicas, rotação e translação, entre os sistemas referenciais de todos os sensores. Esta informação é obtida por calibração utilizando programas *open source* ou algoritmos próprios dos sensores.

A relação entre os referenciais das câmaras é obtida pela calibração *stereo* através da *toolbox* de calibração de câmaras e sistemas *stereo* disponível para Matlab [71].

A transformação entre os referenciais das câmaras e os referenciais dos sensores LRF é estimada por calibração aplicando o método apresentado em [119], do qual existe uma implementação em Matlab [120].

A relação entre as câmaras e os sensores inerciais é determinada com recurso à *toolbox* InerVis [121], baseada no método desenvolvido por Jorge Lobo e Jorge Dias [122].

A determinação da posição da antena receptoras de GPS é estimada por algoritmos internos do sensor iNAV-FMS-E.

7.1.7 Registo do Instante das Medidas

Em sistemas de fusão sensorial o conhecimento do instante das medidas constitui um aspecto de grande relevância, sendo crucial no estabelecimento de relações temporais entre as medidas.

O sistema de aquisição de dados atribui a cada medida uma etiqueta com o respectivo instante de recepção pelo sistema computacional – *timestamp*. A etiqueta de tempo é baseada no tempo de CPU, no entanto a precisão do relógio do CPU degrada-se ligeiramente com o passar do tempo devido à imperfeição do cristal oscilador, fenómeno designado por *clock drift*. O acesso a referências de tempo fornecidas pelos relógios atómicos das estações GPS permite corrigir o relógio de CPU periodicamente.

No instante em que ocorre a medição, os sensores LRF emitem um impulso num pino de saída, este evento chega à unidade computacional através da porta série e é atendido pelo módulo PPS (*Pulse Per Second*), disponível a partir do *kernel* LINUX 2.6.34. O instante de chegada do impulso determina o tempo a atribuir ao varrimento transmitido em seguida via protocolo RS422.

Dado que o disparo das câmaras é comandado pelo iNAV-FMS-E, o tempo atribuído a cada *frame* consiste no instante em que é enviada a ordem de amostragem. Estes instantes são guardados pelo sistema operativo do iNAV-FMS-E sob a forma de um ficheiro de texto.

7.2 Aquisição do *Dataset*

O *dataset* foi adquirido num túnel rodoviário com uma extensão de aproximadamente 100 de metros, situado no arredores da cidade do Porto mais concretamente na localidade de Vilar de Luz.



Figura 7.7: Veículo de mapeamento na zona do túnel, instantes antes da recolha de dados.

À excepção das câmaras CCD, todos os sensores adquirem informações à taxa de amostragem máxima. Precavendo fenómenos de latência nas comunicações, bem como dificuldades ao nível do armazenamento das imagens, as câmaras operam à taxa de 5 Hz.

A velocidade do veículo não foi alvo de um controlo cuidadoso, apesar de ter sido respeitada a velocidade máxima de 35 km/h. O valor da velocidade praticada produz um impacto mais notório nas imagens adquiridas pelas câmaras CCD, pois velocidades elevadas geram discrepâncias acentuadas de cenário entre *frames* consecutivas. Ao limitar a velocidade a 35 km/h é possível garantir que a distância entre a posição da câmara, em instantes de amostragem contíguos, não ultrapassa os 2 metros, o que é aceitável para a aplicação de localização desenvolvida.

A fraca iluminação no interior da galeria associada à incapacidade de ajuste automático do ganho das câmaras dificultou a aquisição de imagens com a qualidade. A grande variação da intensidade luminosa ao longo de todo o percurso perturba fortemente as ima-

gens, originando nos instantes de transição entre o exterior e o interior imagens totalmente saturadas. Em zonas mais interiores a iluminação é diminuta e contribui para o escurecimento das imagens.

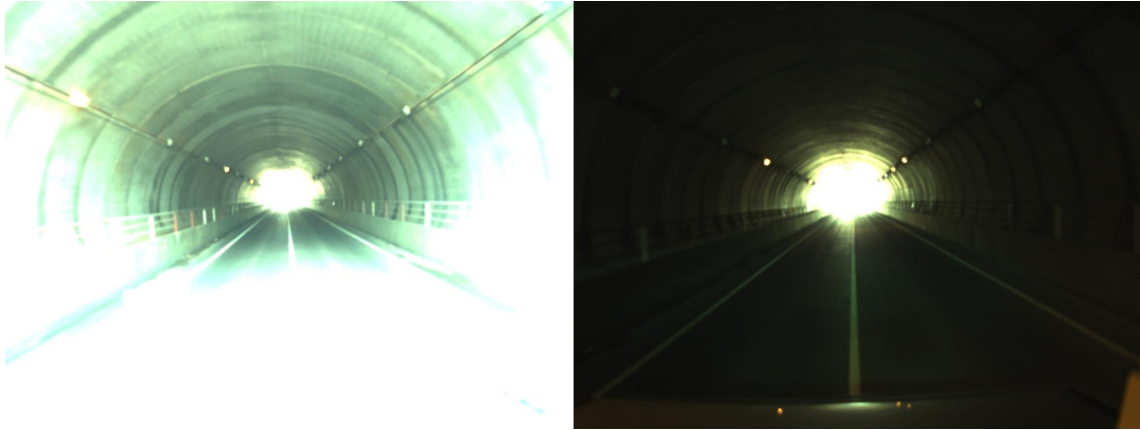


Figura 7.8: Diferença de iluminação entre uma imagem captada à entrada do túnel e outra adquirida já no interior.

Esta página foi intencionalmente deixada em branco.

Capítulo 8

Implementação e Resultados

Conteúdo

8.1	Projecto do Sistema de Localização	120
8.1.1	Pré-selecção de <i>Features</i>	120
8.1.2	Algoritmo de SLAM	123
8.2	Resultados do Sistema de Localização	134
8.3	Construção do Modelo	145

O algoritmo de modelização tridimensional foi segmentado em dois blocos, um deles trata a determinação da localização da plataforma sensorial, enquanto o outro incide sobre a construção do modelo. O processo de localização funde os dados dos sensores inerciais com informações extraídas a partir das imagens capturadas pelas câmaras CCD. As informações do sensor iNAV-FMS-E não contribuem activamente para o cálculo da localização. A trajectória calculada por este sensor é bastante precisa e será utilizada exclusivamente para efeitos de comparação – *ground truth*. Relativamente ao bloco dedicado à construção do modelo, regista-se a contribuição do sensor LMS-200 na aquisição de varrimentos verticais da parede do túnel à direita do veículo. Estas medidas conjugadas com os dados de localização, obtidos pelo primeiro bloco, permitem gerar uma nuvem de pontos, sobre a qual opera o algoritmo BPA com o propósito de construção de uma malha poligonal.

8.1 Projecto do Sistema de Localização

O processo de determinação da localização da plataforma sensorial consiste num algoritmo de SLAM baseado no estimador EKF, ao qual se associam contribuições do estado da arte como a parametrização em *inverse depth*, a utilização de *features* do tipo SIFT, a rejeição de correspondências inconsistentes pelo algoritmo 1-Point RANSAC e a previsão da movimentação da plataforma através da mecanização inercial segundo o referencial de nível local.

A inclusão do sistema inercial visa sobretudo a introdução de informações métricas no filtro, de modo a contrariar a ambiguidade em escala característica dos sistemas de SLAM monoculares. Segundo [123] esta abordagem permite dispensar a visualização de alvos com dimensões conhecidas no instante inicial, com a vantagem de fornecer informações métricas ao longo de todo o percurso.

O algoritmo de localização foi implementado em Matlab, tendo como suporte a aplicação de Visual SLAM desenvolvida por Javier Civera e José Montiel [124], dois investigadores com crédito internacional pela sua pesquisa em aplicações de localização através de sistemas visuais monoculares. A escolha desta base de trabalho deve-se ao facto de incorporar dois métodos de vanguarda, designadamente a parametrização de *features* em *inverse depth* e a integração do algoritmo 1-Point RANSAC. Contudo, a aplicação desenvolvida no âmbito desta dissertação regista alterações profundas relativamente ao código de base, motivadas pela introdução de *features* do tipo SIFT, pela incorporação de medidas inercias na fase de previsão do filtro e pela alteração dos modelos das câmaras – modelo *pinhole* e modelo de distorção – de modo compatibilizar o modelo de observação com as câmaras utilizadas.

8.1.1 Pré-selecção de *Features*

As grandes variações de iluminação registadas nas imagens do *dataset* traduzem-se na redução do contraste especialmente nas *frames* adquiridas em zonas mais interiores do túnel, condicionando a repetibilidade do algoritmo de extracção de *features*. Geralmente os algoritmos de Visual SLAM são desenvolvidos com vista à operação em tempo real, por isso cada *frame* é processada de forma isolada e as *landmarks* são introduzidas no filtro sem que aja a garantia de ocorrência de observações futuras. No nosso caso, dado que toda a sequência de imagens está disponível à partida, foi implementado, à margem do

algoritmo principal de SLAM, um processo de extracção e correspondência de *features* que considera todo o historial de imagens. Esta abordagem traz dois grandes benefícios, desde logo a possibilidade de identificar as *features* com elevado número de observações ao longo da sequência de *frames*, de modo a garantir a viabilidade das *landmarks* introduzidas no mapa. Por outro lado, a não incorporação da extracção de *features* no algoritmo de SLAM reduz significativamente a complexidade do algoritmo, tornando-o mais rápido.

O processo de extracção e correspondência de *features* foi implementado em Matlab com recurso à *toolbox* de extracção de *features* SIFT [125]. O programa começa por extrair *features* na imagem de referencia (imagem inicial), às quais associa o respectivo descritor próprio do método SIFT. Após esta etapa ocorre a extracção de *features* na *frame* posterior seguida do cálculo dos descritores. Os descritores atribuídos aos pontos de interesse são comparados, resultando daí as correspondências entre as duas *frames* contíguas. As correspondências são então analisadas pelo método RANSAC aplicado à estimação da matriz de homografia. Para este fim, foi utilizada a *toolbox* de RANSAC para Matlab [126]. Foi imposto um limite de erro elevado de modo a descartar correspondências absurdas, preservando no entanto as correspondências com grande inovação.

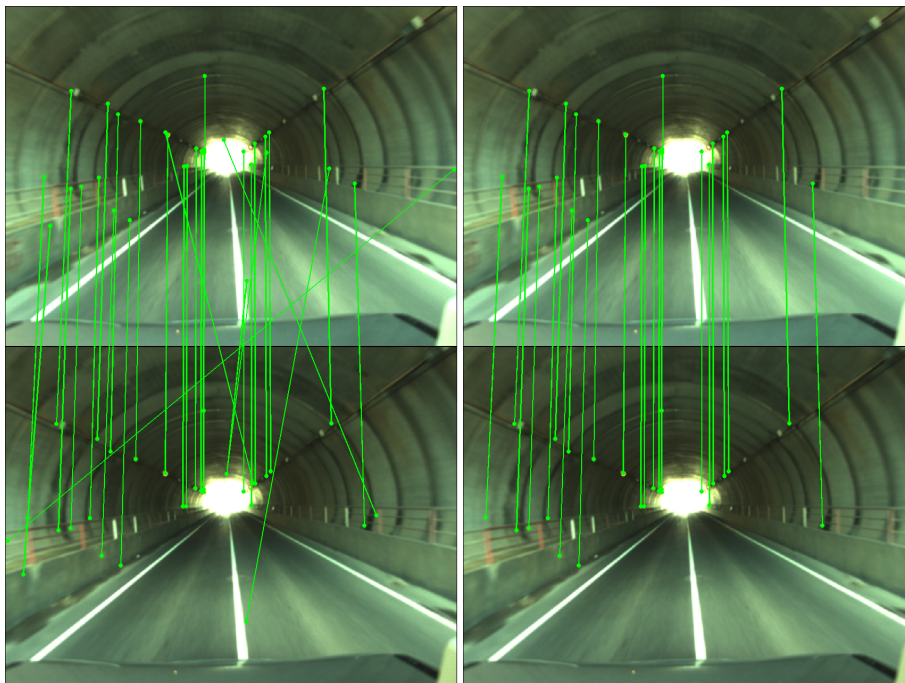


Figura 8.1: Filtragem de correspondências através do método RANSAC tradicional. À esquerda, corespondências pela comparação de descritores SIFT. À direita as correspondências aceites pelo método de RANSAC.

Após a execução dos passos anteriores o procedimento repete-se até que todas as *frames* tenham sido analisadas, a *frame* do instante actual passa a ser a *frame* de referencia e dá-se a comparação com a imagem seguinte.

Para cada *feature* válida é reservado um elemento num *array* de estruturas, que armazena informações relevantes tais como a posição da *feature* em cada *frame*, o índice de todas as *frames* em que a *feature* é observada, o descritor mais recente, bem como alguns campos suplementares destinados ao algoritmo de SLAM. É ainda definido um número mínimo de observações abaixo do qual as *features* são excluídas do *array*, pois num curto espaço de tempo é pouco provável que a *landmark* seja visualizada com um ângulo de paralaxe suficientemente elevado de modo a favorecer a estimação da sua posição no mundo.

A figura (8.2) representa a duração das *features* extraídas ao longo da sequência de *frames*. No eixo vertical do gráfico encontram-se dispostas as várias *features*, a sua duração em *frames* é representada pelo comprimento do respectivo traço horizontal.

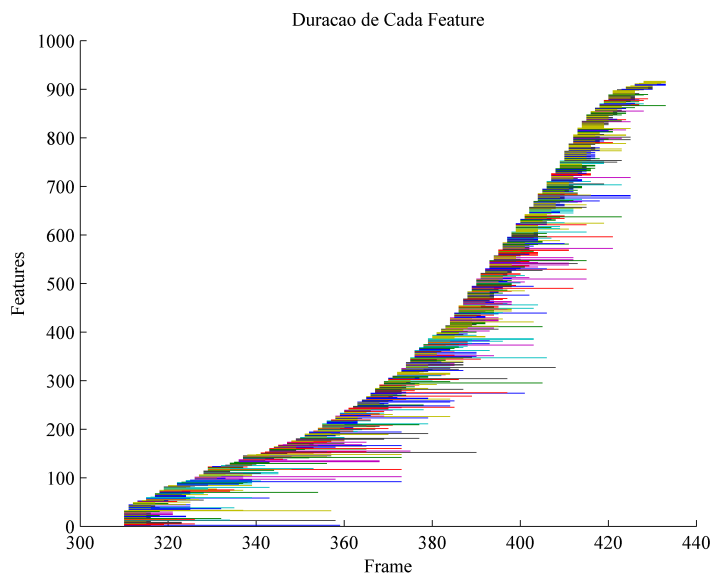


Figura 8.2: Distribuição das *features* observadas em função da sequência de *frames*.

8.1.2 Algoritmo de SLAM

A estimação da localização do veículo de mapeamento é conseguida pela fusão das medidas do sensor 3DM-GX1 com as *features* extraídas nas imagens captadas pela câmara da esquerda. No vector de estados do filtro EKF são reservados nove elementos dedicados à estimação da localização, respectivamente posição, orientação e velocidade, definidas em relação ao referencial de nível local.

Originalmente o método de parametrização em *inverse depth* define a orientação através de um quaternião. Neste trabalho, à semelhança do que acontece em [123], optou-se pela utilização de ângulos de Euler que, para além de constituir uma representação mais intuitiva, facilita os cálculos executados na mecanização inercial. Não é expectável que o sistema se aproxime do ponto de singularidade que caracteriza os ângulos de Euler¹, logo não existe qualquer inconveniente na utilização deste tipo de representação, que aliás reduz em um elemento a dimensão do vector de estados.

O algoritmo de SLAM conjuga as etapas básicas do método EKF com rotinas relacionadas com os métodos do estado da arte incorporados. A sequência de acções é indicada no fluxograma da figura (8.3).

¹A singularidade no ângulos de Euler ocorre quando a rotação intermédia instantânea alinha os eixos da primeira e terceira rotações. Para uma sequência de rotações *roll/pitch/yaw* a singularidade acontece para ângulos *pitch* = $\pm 90^\circ$

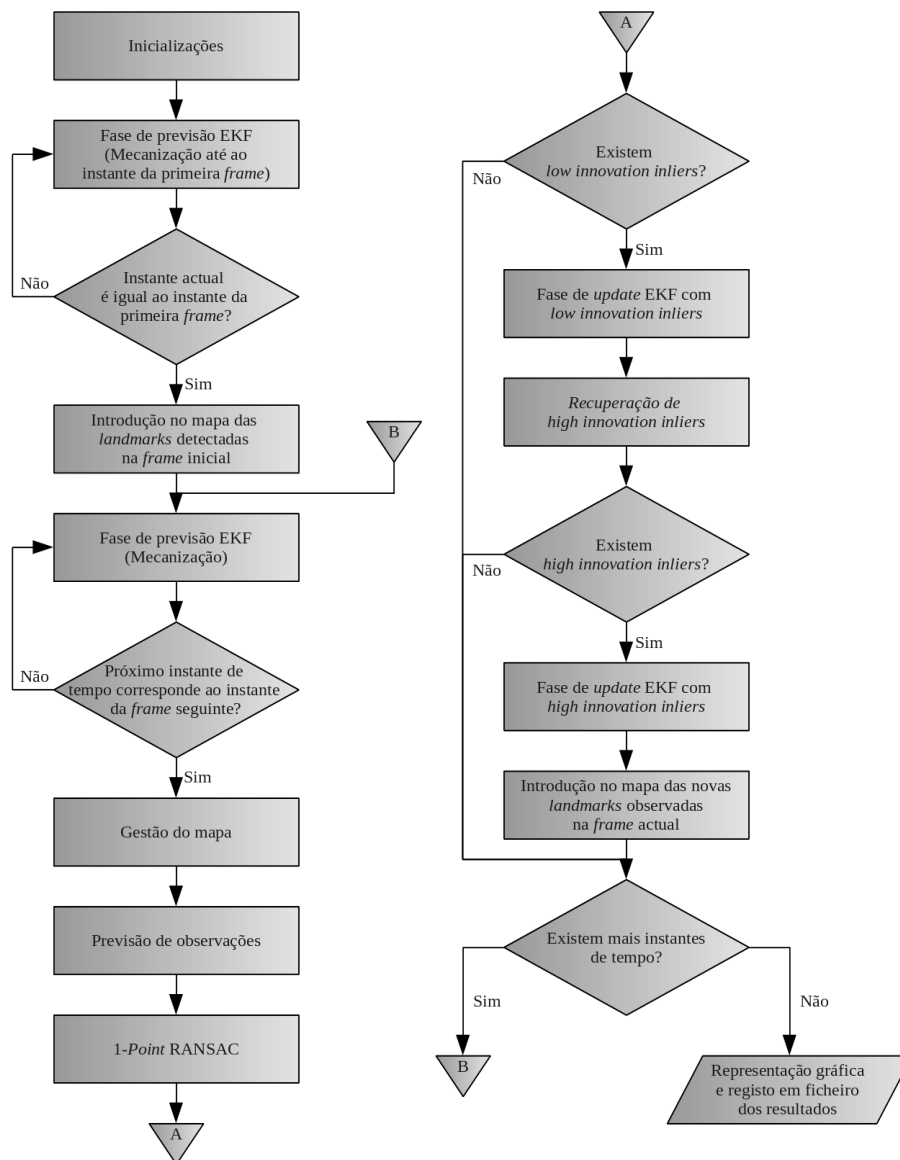


Figura 8.3: Fluxograma do algoritmo de SLAM.

8.1.2.1 Inicializações

Na fase de inicialização ocorre a definição dos diversos parâmetros utilizados no filtro e a leitura dos ficheiros que contêm os dados dos sensores. Neste instante sucede também a declaração das relações entre os sistemas referenciais dos sensores. A *toolbox* utilizada na determinação da relação entre os sensores inerciais e as câmaras não produziu resultados fiáveis. Face à indisponibilidade de outro método, as transformações consideradas no algoritmo consistem em aproximações grosseiras, estabelecidas por mera observação da disposição dos sensores.

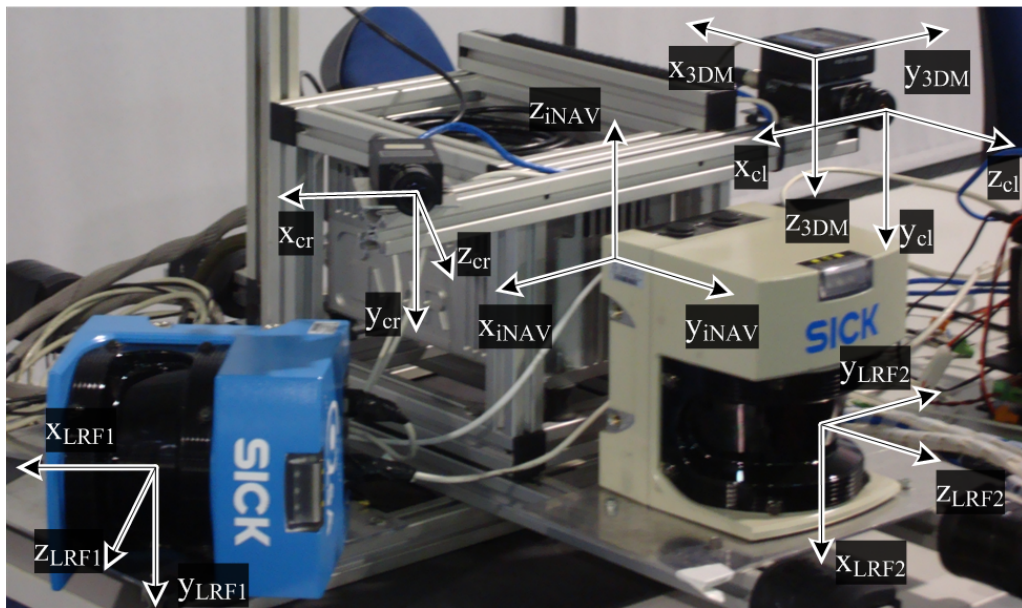


Figura 8.4: Representação da orientação dos sistemas de eixos dos sensores

Tem ainda lugar a inicialização do vector de estados e da matriz de covariância do filtro. Considera-se que no instante inicial a posição do veículo coincide com a origem do referencial de nível local, de modo que a posição inicial é $r^n = [0 \ 0 \ 0]^T$. A orientação inicial, que traduz o desalinhamento dos eixos do sensor 3DM-GX1 em relação a Norte, Este e ao centro da Terra, é obtida pela transformação para o referencial de nível local dos valores de orientação calculados pelo sensor iNAV-FMS-E no instante inicial. A localização pós-processada do sensor iNAV-FMS-E é definida segundo as direcções Este, Norte e sentido oposto ao centro da Terra – referencial ENU – assim o desalinhamento dos eixos do sensor 3DM-GX1 partindo das medidas do sensor iNAV-FMS-E é obtido através das relações ilustradas na figura (8.5).

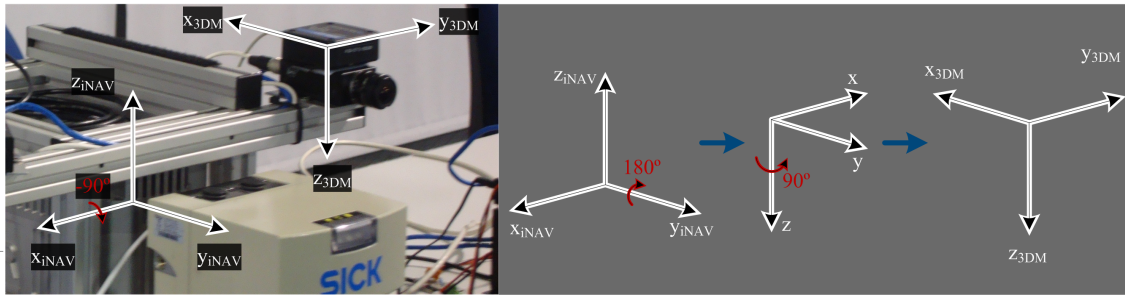


Figura 8.5: Sequência de rotações que alinham o sistema referencial do sensor iNAV-FMS-E com o sistema referencial do sensor 3DM-GX1.

A velocidade inicial é fornecida pelo sensor iNAV-FMS-E. Como as velocidades são definidas segundo o referencial de nível local é necessário efectuar a transformação de velocidades entre o referencial ENU e o referencial de nível local:

$$\begin{bmatrix} v_N^n \\ v_E^n \\ v_D^n \end{bmatrix} = \begin{bmatrix} v_N^n(iNAV-FMS-E) \\ v_E^n(iNAV-FMS-E) \\ -v_U^n(iNAV-FMS-E) \end{bmatrix} \quad (8.1)$$

A inicialização dos estados com os dados da trajectória de referência promove uma comparação justa entre a trajectória de referencia e a calculada pelo sistema de localização, contudo a imprecisão nas transformações entre referenciais introduz à partida um factor de erro, que se vai manifestar durante todo processo degradando a qualidade da localização estimada.

No instante inicial não existe qualquer correlação entre os erros dos estados, pelo que apenas a diagonal da matriz de covariância toma valores diferentes de zero. Aos estados que expressam posição é atribuída a variância de 0.01 metros, os estados relativos à orientação partem com uma variância de 0.008 radianos ao passo que a variância das velocidades é de 0.02 m/s. Estes valores dizem respeito à incerteza da transformação entre o referencial do sensor iNAV-FMS-E e o referencial do corpo. A matriz de covariância inicial fica:

$$P_0 = \begin{bmatrix} 0.01 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.008 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.008 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.008 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.02 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.02 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.02 \end{bmatrix} \quad (8.2)$$

8.1.2.2 Fase de Previsão

O modelo de movimento utilizado na fase de previsão do EKF consiste nas equações de mecanização segundo o referencial de nível local (secção 5.2.1). As possíveis *landmarks* presentes no mapa não possuem dinâmica, portanto apenas os estados relativos à localização sofrem modificações, calculadas do seguinte modo:

$$\begin{bmatrix} r^n(k|k-1) \\ \Theta^n(k|k-1) \\ v^n(k|k-1) \end{bmatrix} = \begin{bmatrix} r^n(k-1|k-1) + v^n(k|k-1)\Delta t \\ \Theta^n(k-1|k-1) + E_b^n w^b(k)\Delta t \\ v^n(k-1|k-1) + (C_b^n a^b(k) + g^n)\Delta t \end{bmatrix} \quad (8.3)$$

Numa primeira fase a etapa de previsão é desencadeada após a inicialização, de modo a propagar o movimento do veículo, desde o instante de inicialização até ao momento de aquisição da primeira *frame*, com base na sequência de medidas inerciais disponível nesse intervalo de tempo. Com a entrada do algoritmo no ciclo de previsão/update, típico do método recursivo EKF, a fase de previsão preenche com informação os intervalos entre *frames*.

A mecanização inercial é classificada como uma técnica de localização de *dead reckoning*, onde a incerteza cresce em função do tempo. Sucede então a actualização da matriz de covariância, através da fórmula de cálculo imposta pelo algoritmo EKF (fórmula 8.4), com o intuito de fazer reflectir o aumento da incerteza na covariância dos estados.

$$P(k|k-1) = \nabla F(k) * P(k-1|k-1) * \nabla^T F(k) + Q \quad (8.4)$$

A matriz $\nabla F(k)$ indica o jacobiano do modelo do processo relativamente aos estados contidos no vector de estados. A matriz Q resulta da operação $G(k)Q_v G(k)^T$, onde $G(k)$ é a matriz jacobiana do modelo do processo em ordem às fontes de erro, a matriz Q_v é nula à excepção da diagonal onde se definem as variâncias de cada fonte de erro.

O cálculo das matrizes jacobianas pode ser encontrado no Anexo (A.4).

8.1.2.3 Inicialização de *Landmarks*

No instantes de inicialização as *landmarks* são introduzidas no filtro sob a forma de profundidade inversa, provocando a extensão das dimensões do vector de estados e da matriz de covariância, de modo a incorporarem as informações deste novo elemento. Uma *landmark* em profundidade inversa é definida por seis parâmetros, posição da câmara no instante de primeira visualização, ângulos de azimute e elevação e profundidade inversa. A posição da câmara no instante inicial esta acessível no vector de estados e a profundidade inversa é definida pelo utilizador, porém os ângulos de azimute e elevação são obtidos a partir das componentes o vector direcção no referencial da câmara, resultante da aplicação do inverso do modelo de observação.

O cálculo dos ângulos de azimute e elevação processa-se como descrito na secção (5.3.4.1) à excepção das fórmulas (5.31) (5.32). A transformação do vector direcção do referencial da câmara para o referencial de nível local (5.31) impõe a conjugação de duas transformações, em primeiro lugar do referencial da câmara para o referencial do corpo, seguida da transformação do referencial do corpo para o referencial de nível local:

$$h^n = C_b^n C_c^b \begin{bmatrix} h_{u_x}^c \\ h_{u_y}^c \\ 1 \end{bmatrix} \quad (8.5)$$

Os parâmetros da câmara obtidos por calibração assumem uma orientação do referencial da câmara diferente da considerada na versão original da parametrização em profundidade inversa (figura 8.6).

A fórmula de cálculo dos ângulos de azimute e elevação a partir do vector direcção reflecte a divergência em orientação dos eixos do referencial da câmara, ficando:

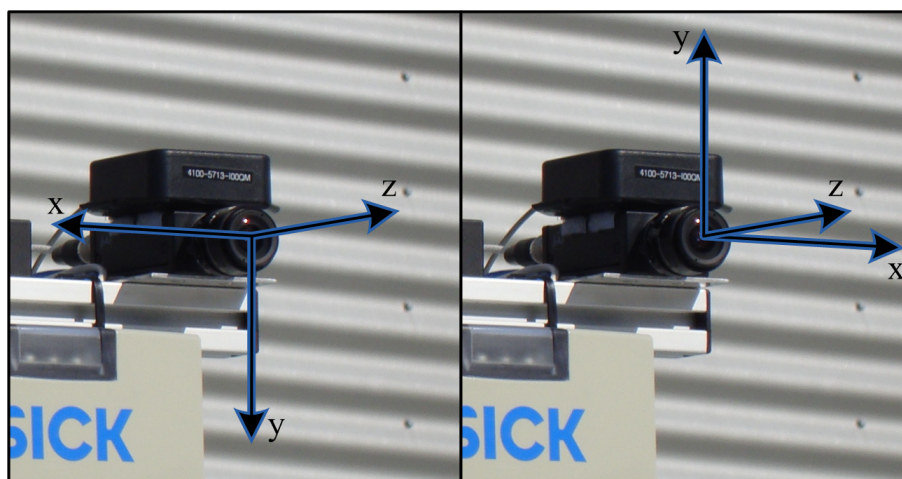


Figura 8.6: Diferença de orientação do referencial da câmara entre a solução utilizada neste trabalho (esquerda) e a versão original da parametrização em profundidade inversa (direita).

$$\begin{bmatrix} \theta_i \\ \phi_i \end{bmatrix} = \begin{bmatrix} \arctan(-h_x^n, h_z^n) \\ \arctan\left(h_y^n, \sqrt{h_x^{n2} + h_z^{n2}}\right) \end{bmatrix} \quad (8.6)$$

A nova matriz de covariância é calculada de acordo com a fórmula (5.33). O método de cálculo do jacobiano do inverso do modelo de observação em ordem aos estados do filtro encontra-se descrito em anexo (A.3).

8.1.2.4 Gestão do Mapa

O bloco de gestão do mapa implementa a eliminação de *landmarks* do mapa e a conversão de *landmarks* definidas em *inverse depth* para a representação em coordenadas cartesianas.

Outro grande benefício ainda não mencionado do processo de pré-selecção de *features* consiste no conhecimento do momento em que cada *landmark* deixa de ser observada. Nesse instante a *landmark* é removida do mapa, levando à redução da dimensão do vector de estados e da matriz de covariância, o que acarreta ganhos computacionais.

A conversão de *landmarks* parametrizadas em *inverse depth* para a representação em coordenadas cartesianas constitui outra estratégia de redução dos requisitos computacionais, mais uma vez através da diminuição do número de estados. A conversão entre as duas representações é conseguida pela relação da fórmula (5.18) e acontece caso o índice de

linearidade (fórmula 5.35) ultrapasse o valor de 0.1. A matriz de covariância é modificada tal como indicado em (5.35) empregando o jacobiano (A.19).

8.1.2.5 Previsão de Observações

Este bloco executa a primeira tarefa da fase de *update* do EKF, que corresponde à aplicação do modelo de observação para determinação das observações previstas. O modelo de observação é aplicado sucessivamente a cada uma das *landmarks* do mapa, de modo a projectá-las no espaço das observações, neste caso a imagem.

Com este propósito, foi implementada uma modificação do modelo de observação exposto na versão original da parametrização em *inverse depth*(5.3.4.1). A variante desenvolvida adapta-se à existência de um sistema referencial adicional – referencial do corpo – à definição da orientação do corpo através de ângulos de Euler e à divergência entre a orientação do referencial da câmara, dado pela *toolbox* de calibração, e a orientação assumida no artigo original. As alterações associadas a estas discordâncias manifestam-se apenas sobre a equação que projecta as *landmarks* segundo o referencial métrico da câmara, que para *landmarks* em *inverse depth* passa a ficar:

$$h^c = h_\rho^c = C_b^c C_n^b \left(\rho_i \left(\begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} - r^n - C_b^n t_{bc}^b \right) + m(\theta_i, \phi_i) \right) \quad (8.7)$$

Já para *landmarks* em coordenadas cartesianas a projecção do referencial de nível local para o referencial métrico da imagem respeita a fórmula (8.8)

$$h^c = h_{XYZ}^c = C_b^c C_n^b \left(\begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} - r^n - C_b^n t_{bc}^b \right) \quad (8.8)$$

- Onde C_b^c representa a matriz rotação do referencial do corpo para o referencial da câmara, de forma semelhante C_n^b indica a matriz rotação do referencial de nível local para o referencial do corpo. C_b^n é a inversa de C_n^b , logo define a rotação do referencial do corpo para o referencial de nível local. Por fim, t_{bc}^b consiste na translação do referencial da câmara face ao referencial do corpo definida segundo este último.

A alteração da orientação dos eixos do referencial métrico da câmara repercute-se na fórmula de cálculo da matriz $m(\theta_i, \phi_i)$, que passa a ser:

$$m(\theta_i, \phi_i) = \begin{bmatrix} -\cos(\phi_i)\text{sen}(\theta_i) \\ \text{sen}(\phi_i) \\ \cos(\phi_i)\cos(\theta_i) \end{bmatrix} \quad (8.9)$$

As fórmulas de compensação da distorção (5.23) e projecção na imagem (5.24) completam o modelo de observação.

O cálculo da observação prevista é acompanhada pela determinação da covariância associada à observação (fórmula 5.11), o que exige o cálculo das derivadas do modelo de observação em ordem aos estados do sistema, indicadas por $\nabla H(k)$. O método de cálculo deste jacobiano é abordado em pormenor no anexo (A.1).

8.1.2.6 1-Point RANSAC

Após o cálculo das observações previstas desenvolve-se a fase de filtragem das correspondências entre *features*, previamente estabelecidas pela comparação entre descritores SIFT. Neste processo é empregue o algoritmo iterativo 1-Point RANSAC (secção 4.6.1).

Em cada ciclo é escolhida uma correspondência de forma aleatória, através da qual se calcula o estado estimado pela aplicação das equações de *update* do EKF (equações 5.11, 5.12 e 5.13). Recorrendo ao modelo de observação é calculada a observação prevista para todas as *landmarks* presentes no mapa, com base no estado estimado hipotético. Sucede em seguida a análise de proximidade entre as observações previstas e a respectiva posição verdadeira da *feature* na imagem. O grau de proximidade é avaliado através da distância euclidiana entre as coordenadas da observação prevista e a posição efectiva da *feature* na imagem.

Neste trabalho foi escolhido um limite de distância de dez píxeis, abaixo do qual se considera que uma determinada correspondência suporta a hipótese estabelecida. Durante a sucessão de iterações é mantida uma solução provisória, correspondente à hipótese com maior suporte até então.

O número de ciclos hipótese/teste é ajustado de forma dinâmica de acordo com o número de correspondências existentes nesse instante de tempo e com o número de *inliers*

atribuídos à solução provisória, de modo que a probabilidade de encontrar uma amostra constituída exclusivamente por *inliers* seja de 0.99.

$$N_{\text{iterações}} = \frac{\log(1 - 0.99)}{\log\left(1 - \left(1 - \frac{\text{suporte atribuído à hipótese}}{\text{número de correspondências}}\right)\right)} \quad (8.10)$$

Quando esgotado o número de iterações, a solução provisória torna-se definitiva e as correspondências que suportam essa hipótese são denominadas por *low-innovation inliers*.

8.1.2.7 Fase de *Update* EKF

Na etapa de *update* o valor dos estados é corrigido com base na discrepância entre as observações previstas e as observações efectuadas. Como em cada instante de *update* está disponível mais que uma observação, é utilizada a estratégia de *update* em grupo, que de uma só vez processa todas as observações pela aglutinação de matrizes [127]. Segundo este conceito, as matrizes z , h , ∇H e R sofrem uma expansão, de modo a concentrarem os dados referentes às n *landmarks* observadas.

$$z(k) = \begin{bmatrix} z_{l1}(k) \\ \vdots \\ z_{ln}(k) \end{bmatrix} \quad (8.11a)$$

$$\nabla H(k) = \begin{bmatrix} \nabla H_{l1}(k) \\ \vdots \\ \nabla H_{ln}(k) \end{bmatrix} \quad (8.11b)$$

$$h(k) = \begin{bmatrix} h_{l1}(k) \\ \vdots \\ h_{ln}(k) \end{bmatrix} \quad (8.11c)$$

A covariância do ruído das observações – matriz R – é construída dispondo sobre a diagonal os blocos relativos a cada observação.

$$R = \begin{bmatrix} R_{l1}(k) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & R_{ln}(k) \end{bmatrix} \quad (8.12)$$

A formulação anterior permite a estimação dos estados através das equações de *update* do EKF (equações 5.11, 5.12 e 5.13), considerando em simultâneo todas as observações disponíveis. A actualização da matriz de covariância segue também a equação típica do EKF (equação 5.14).

8.1.2.8 Recuperação de *high-innovation inliers*

Respeitando o algoritmo *1-Point RANSAC*, no seguimento da primeira fase de *update* parcial com *low-innovation inliers* sucede a etapa de recuperação de *high-innovation inliers*, que assenta na geração de uma janela de validação no espaço das observações [127].

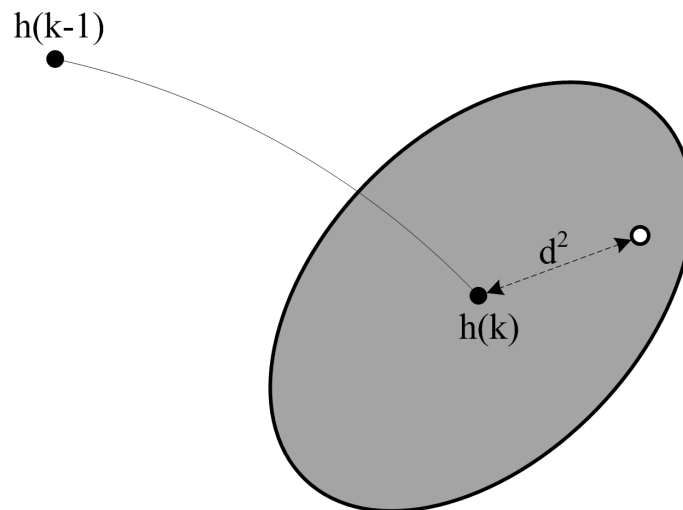


Figura 8.7: Representação da janela de validação

A janela de validação é calculada individualmente para cada *landmark* classificada como *outlier* na primeira instância do algoritmo *1-Point RANSAC*. A janela de validação possui uma forma elíptica, cujas proporções se relacionam com os valores próprios da matriz inversa de covariância da inovação – $S^{-1}(k)$. A escala da elipse é definida pelo utilizador, com base na probabilidade de associação desejada, e o centro encontra-se no ponto da observação prevista. Uma determinada observação torna-se *high-innovation inlier* caso a observação real esteja contida na área interna da janela.

A implementação deste método de validação assenta na determinação da inovação normalizada, que implica o cálculo da observação prevista, considerando o estado estimado resultante da etapa de *update* com *low-innovation inliers*. A partir da observação prevista e da observação real calcula-se a inovação relativa à *landmark* l_i :

$$\hat{i}_{li} = z_{li}(k) - h_{li}(x(k|k)) \quad (8.13)$$

Segue-se o cálculo da covariância da inovação.

$$S(k)_{li} = \nabla H_{li}(k)P(k|k)\nabla^T H_{li} + R_{li} \quad (8.14)$$

Por fim é determinado o valor da inovação normalizada.

$$d_{li}^2 = \hat{i}_{li}^T S_{li}^{-1} \hat{i}_{li} \quad (8.15)$$

Quando a sequência de inovações evidencia média zero e ruído branco – comportamento esperado de um EKF bem afinado – a inovação normalizada é uma variável aleatória qui-quadrado. Assim, é possível estabelecer, com um dado valor de probabilidade, um limite a baixo do qual se considera que a inovação normalizada corresponde a uma observação coerente. O valor limite utilizado é de 5.9915, que se traduz na aceitação das correspondências correctas com uma probabilidade de 0.95.

As correspondências válidas são classificadas como *high-innovation inliers* e vão alimentar a etapa de *update* parcial com *high-innovation inliers*.

8.2 Resultados do Sistema de Localização

A fiabilidade da trajectória determinada pelo sistema de localização será avaliada por comparação com a trajectória pós-processada obtida pelo sensor iNAV-FMS-E (*ground truth*). Para aferir as vantagens da fusão sensorial entre as medidas do sensor 3DM-GX1 e as informações adquiridas a partir das imagens, a trajectória calculada pelo algoritmo de SLAM vai ser confrontada com a trajectória dada pela mecanização inercial segundo o referencial de nível local e com a estratégia de SLAM monocular [106], que utiliza um modelo de movimento com velocidade constante. Todas as trajectórias são inicializadas com os mesmos dados de posição, velocidade e orientação.

As trajectórias calculadas por cada método são apresentadas conjuntamente nas figuras seguintes.

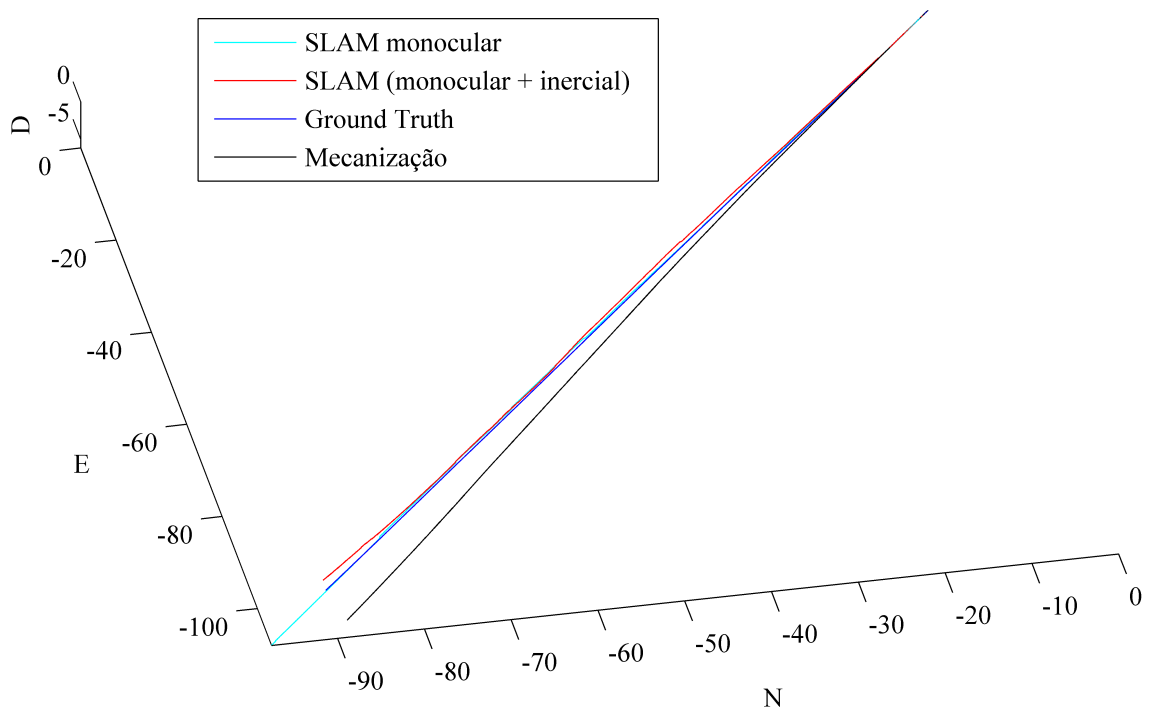


Figura 8.8: Trajectória tridimensional calculada por cada método (unidades em metros).

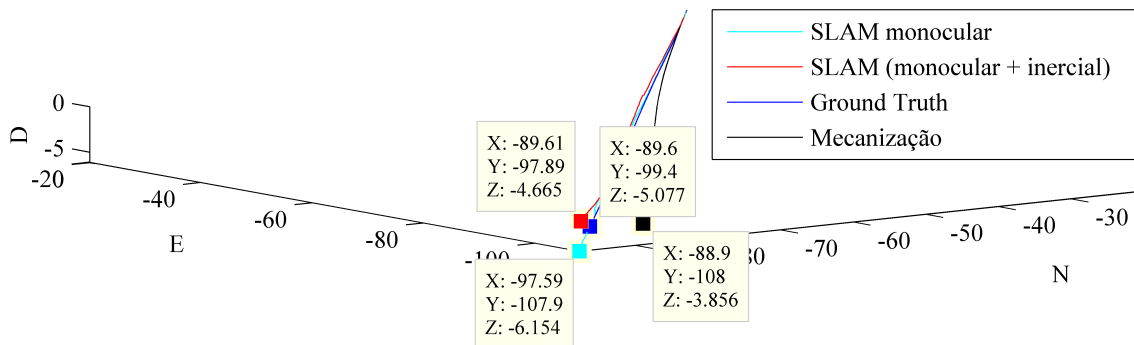


Figura 8.9: Posição final das várias trajectórias (unidades em metros).

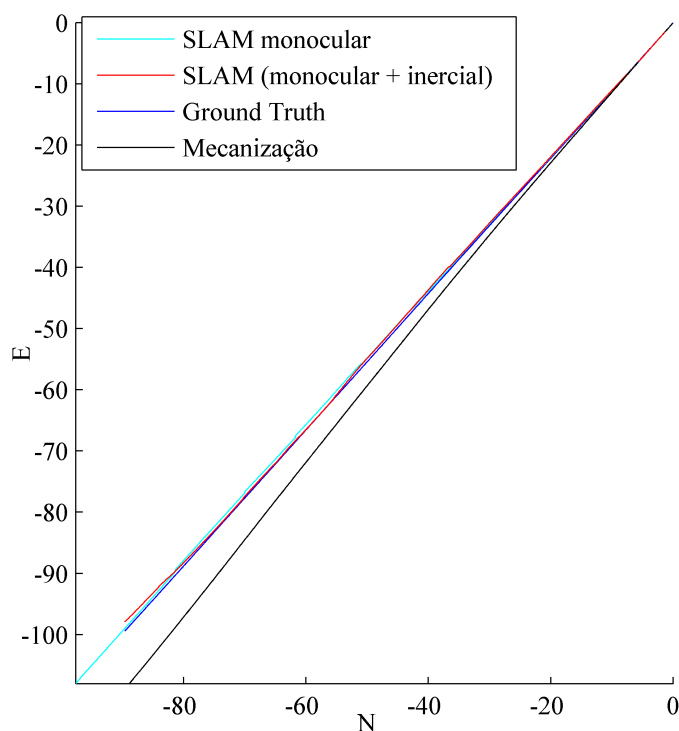


Figura 8.10: Vista superior dos percursos (unidades em metros).

Das figuras anteriores pode concluir-se que a trajectória relacionada com a mecanização inercial é a que regista o maior afastamento relativamente à trajectória de referência. Como referido anteriormente, as técnicas de *dead reckoning* caracterizam-se pela integração sucessiva de erros, o que origina grandes desvios com o decurso do tempo. A referida trajectória no último instante de tempo dista 8.7 metros da trajectória de referência.

O sistema de SLAM monocular apresenta um afastamento final de 11.7 metros face à trajectória de referência, porém, apesar do erro em posição ser superior ao registado pela mecanização, em termos do caminho percorrido, a estratégia de SLAM monocular desenvolve-se numa área mais próxima da trajectória de referência. A distância percorrida superior, apresentada por este método, pode dever-se à ambiguidade de escala propícia dos algoritmos de *Visual SLAM* monoculares.

O método de SLAM baseado na fusão de medidas inerciais com informações visuais produz resultados mais precisos. As mais-valias deste método em comparação com a navegação inercial e o método de *Visual SLAM* monocular são evidentes; a introdução de observações visuais contribui marcadamente para a correcção da orientação, compensando drasticamente o *drift* da mecanização inercial; a fusão de medidas inerciais pelo algoritmo

de *Visual SLAM* monocular resulta numa melhor estimação da posição, pois a entrada de informações métricas durante todo o percurso previne a definição dos estados segundo um factor de escala desconhecido.

Vejamos em seguida, para cada método, a evolução dos erros em posição relativamente à trajectória de referência.

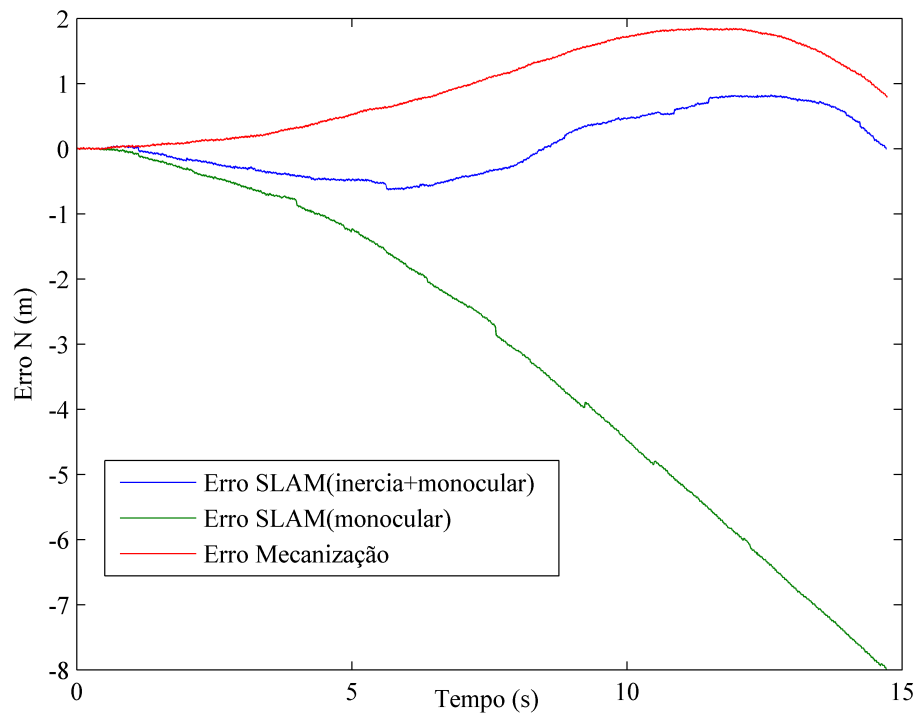


Figura 8.11: Erros em posição segundo a direcção Norte.

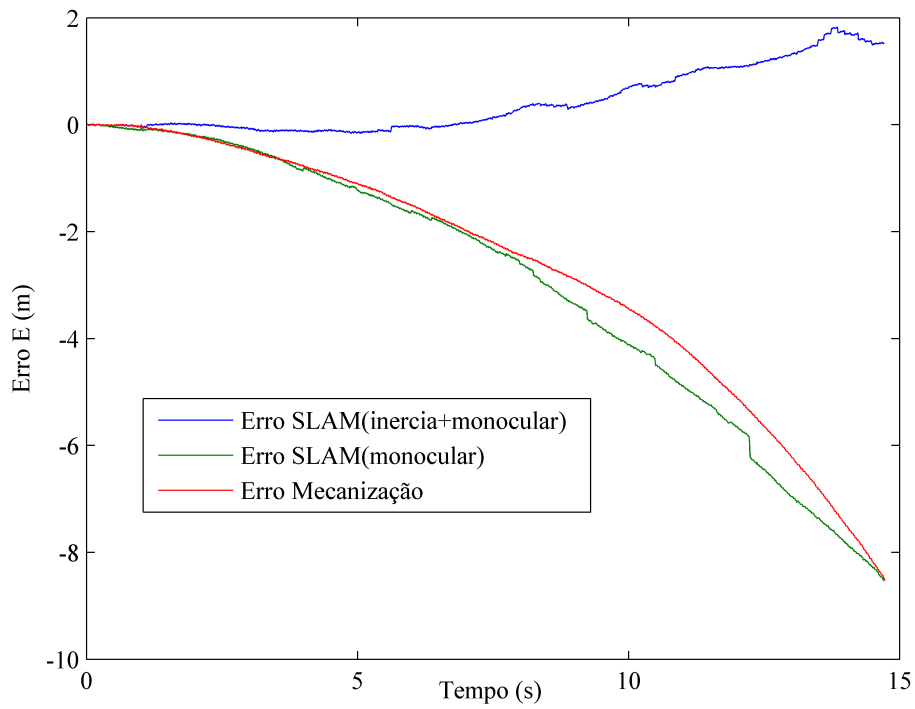


Figura 8.12: Erros em posição segundo a direcção Este.

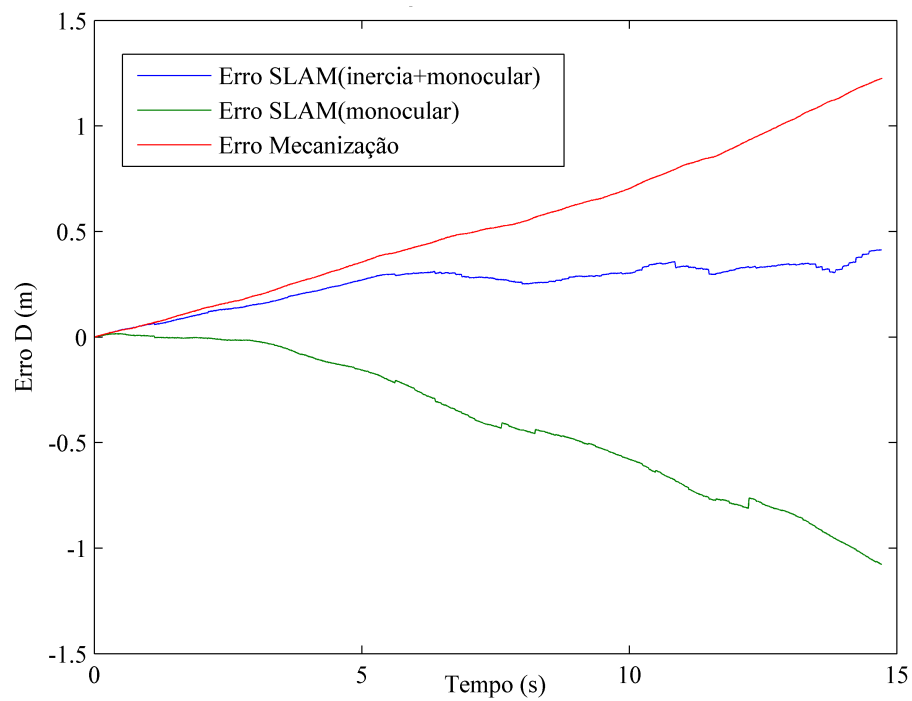


Figura 8.13: Erros em posição segundo a direcção do centro da Terra.

Mais uma vez se verifica a supremacia do método de SLAM com medidas inerciais e visuais face às outras duas modalidades mais simples. Os erros dos estados de posição mantêm-se baixos, logo estes estados não divergem significativamente.

Na mecanização inercial, os estados que registam a posição segundo a direcção Este e a direcção do centro da terra degradam-se visivelmente ao longo do percurso. Num sistema de navegação inercial, a precisão nos estados relativos à posição é muito penalizada pelos ruídos das medidas, dado que acumulam os erros causados pela dupla integração dos ruídos em aceleração, sendo ainda influenciados pelos erros contidos na orientação estimada, introduzidos aquando do cálculo da velocidade.

O algoritmo de *Visual SLAM* monocular manifesta uma evolução regular dos estados de posição. O declive praticamente constante leva a crer que a posição está a ser compensada de forma balanceada, e que a divergência face à trajectória de referência se deve ao factor de escala incerto. Uma indicação que atesta a convergência de um filtro consiste na correcção dos estados em ambos os sentidos, ora esse comportamento é evidenciado especialmente nos estados Norte e Este, onde é possível visualizar transições em sentidos contraditórios.

Passemos à análise dos erros em orientação.

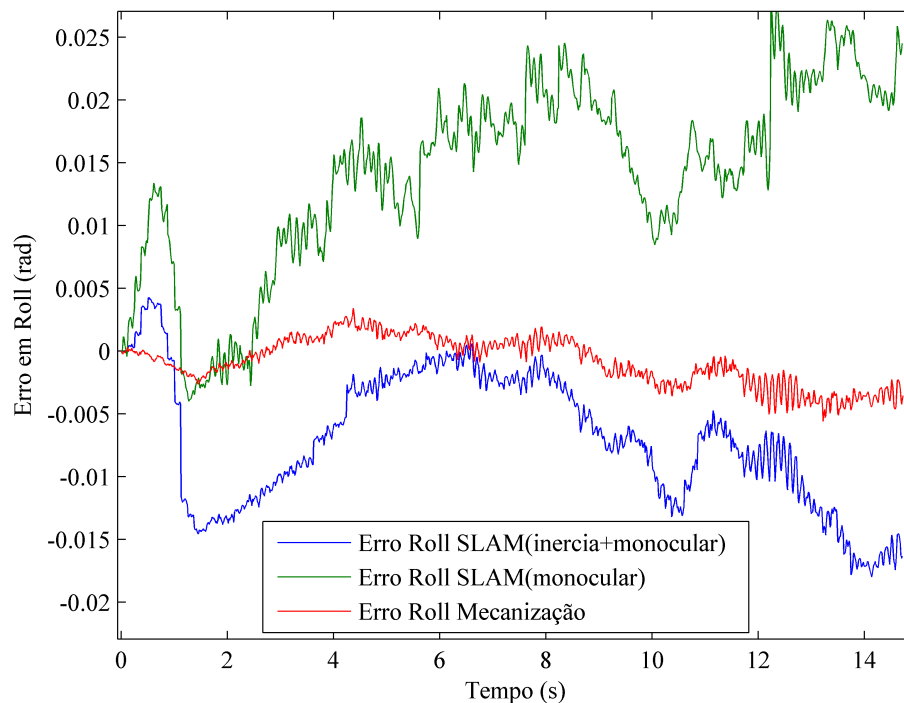
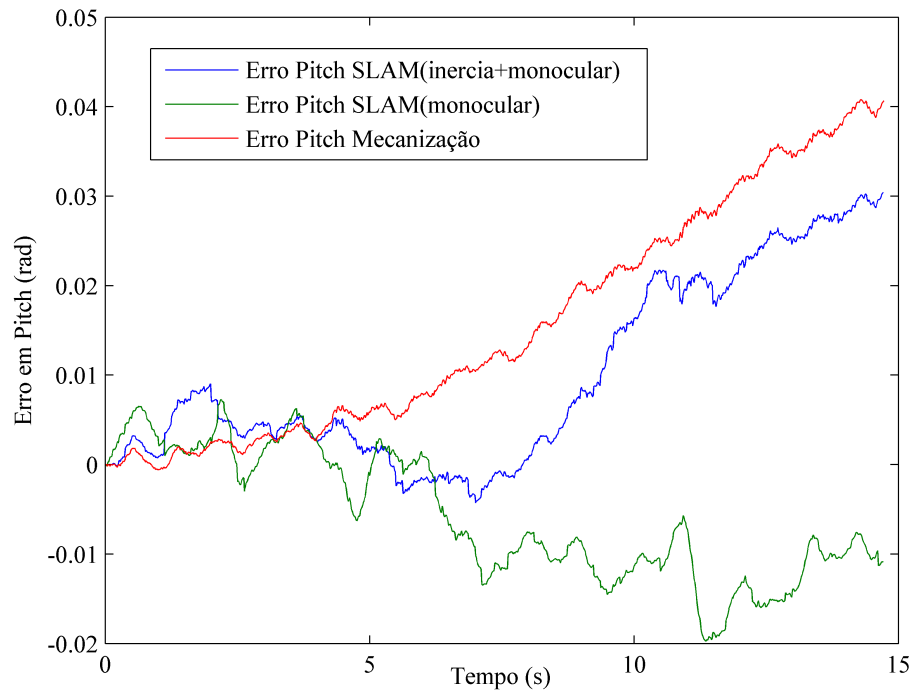
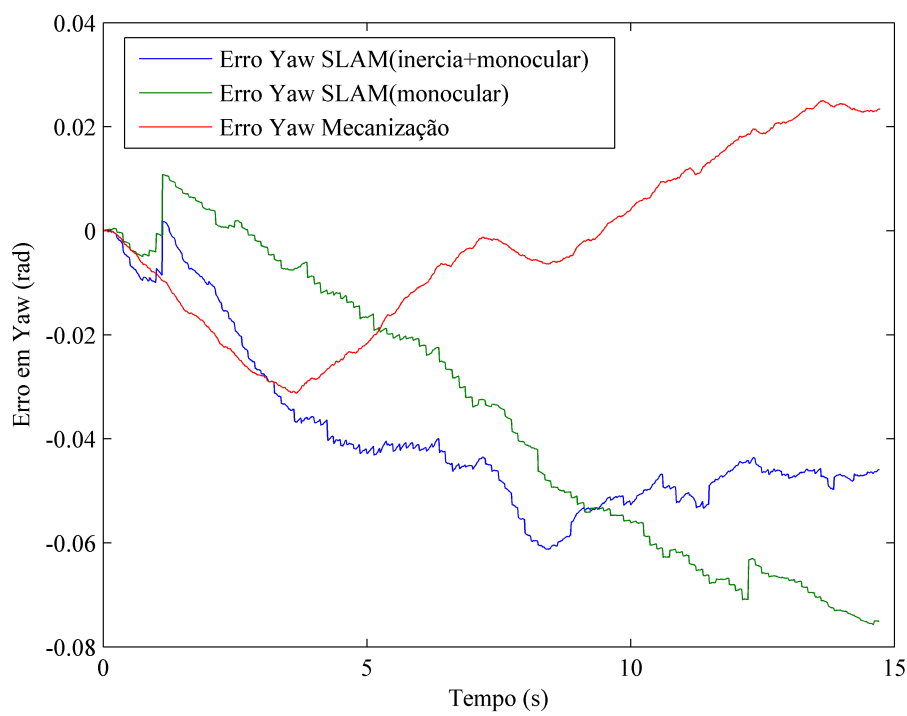


Figura 8.14: Erros em orientação segundo o eixo *roll*.

Figura 8.15: Erros em orientação segundo o eixo *pitch*.Figura 8.16: Erros em orientação segundo o eixo *yaw*.

Anteriormente, na análise dos estados de posição, foi atribuída uma classificação fraca ao percurso calculado pela mecanização inercial, devido à não coincidência com a trajectória de referência. O desvio da trajectória foi encarado como premonição de um erro em orientação considerável, contudo, curiosamente, a mecanização inercial aparenta possuir a estimação de orientação mais precisa, realçando-se os erros reduzidos dos ângulos *yaw* e *roll*. Tal poderá não cooresponder à verdade, pois os erros exibidos anteriormente são influenciados pela rotação entre os referenciais dos sensores inerciais, que foi determinada de forma grosseira. Assim, os erros patentes na transformação entre referenciais, adulteram o cálculo dos erros em orientação levando a uma interpretação errada do funcionamento dos sistemas de localização.

Permanece então a convicção de que os erros em orientação, associados à localização determinada pelo sistema de navegação inercial, podem ser superiores aos quantificados nas figuras anteriores.

Por sua vez, os erros em orientação associados aos métodos de SLAM parecem ser exagerados, considerando o alinhamento satisfatório entre as trajectórias estimadas e a trajectória de referência.

Conclui-se que a incerteza na transformação entre os sistemas referenciais, para além de introduzir erros nos mecanismos de estimação, causa dificuldades na análise dos resultados.

O comportamento dos erros nos estados métricos, designadamente posição e velocidade, contém pistas que permitem perceber os desequilíbrios em orientação. Já na análise aos estados de posição foi possível tecer algumas previsões relativamente aos erros em orientação presentes em cada trajectória. Infelizmente, a análise dos erros em orientação não suportou essas mesmas considerações, pelos motivos já enunciados. Neste contexto, a interpretação dos erros em velocidade poderá contribuir para uma clarificação do que realmente sucede nos estados de orientação.

Os erros em velocidade são afectados por uma componente de erro relacionada com os estados de orientação, pois é com base na orientação estimada que se processa o cálculo da rotação que alinha as medidas dos acelerómetros com as direcções Norte, Este e centro da Terra. Os ruídos nas medidas de aceleração angular constituem outra fonte de erro relevante, que se manifesta na velocidade de forma amplificada, devido ao cálculo de integração.

O estudo do erro em velocidade segundo a direcção do centro da terra é bastante

proveitoso. Recorde-se a expressão da mecanização inercial voltada para o cálculo da velocidade (equação 5.1). Pode constatar-se que um desalinhamento do eixo vertical tem consequências significativas no desempenho do sistema, pois a compensação da aceleração da gravidade passa a ser mal executada, levando a um erro notório na velocidade vertical.

Como se comprova na figura (8.17) o erro em velocidade no sentido do centro da Terra calculado pela mecanização é diminuto, logo o eixo vertical não se encontra desalinhado, ou seja os ângulos *roll* e *pitch* estão bem definidos. Esta observação contrasta com erro apresentado anteriormente para o ângulo *pitch*.

Erro menor no estado de velocidade vertical é atingido pelo algoritmo de fusão sensorial, indicando uma melhor estimação dos ângulos *roll* e *pitch*.

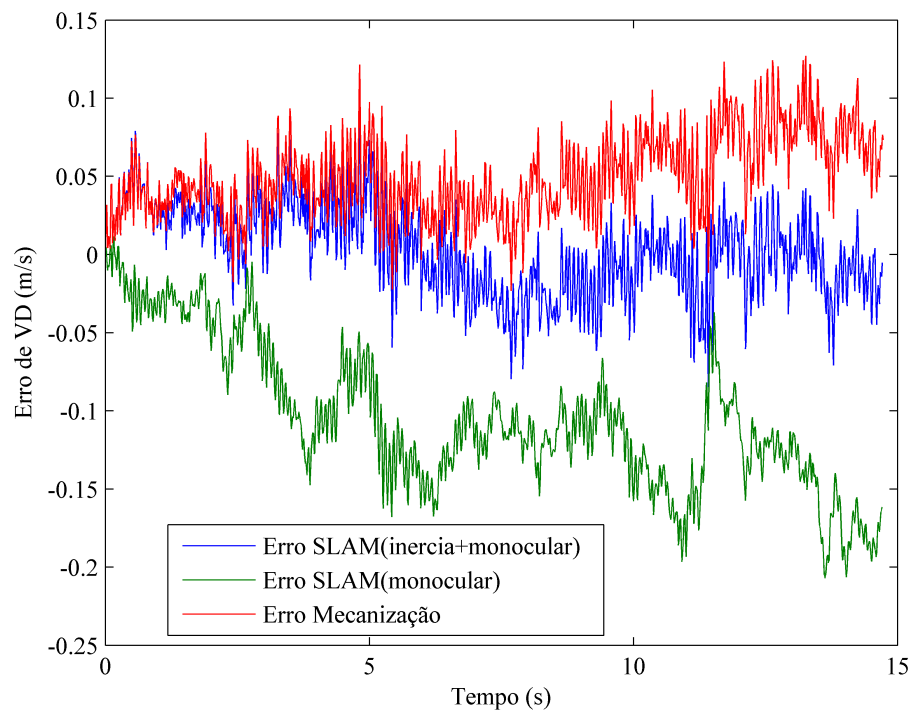


Figura 8.17: Erros em velocidade segundo a direcção do centro da Terra.

Novamente, para a localização obtida pela mecanização, o erro em velocidade segundo a direcção Este é bastante elevado e sofre uma acentuação de declive por volta dos 10 segundos, instante a partir do qual o erro *yaw* sofre um aumento considerável.

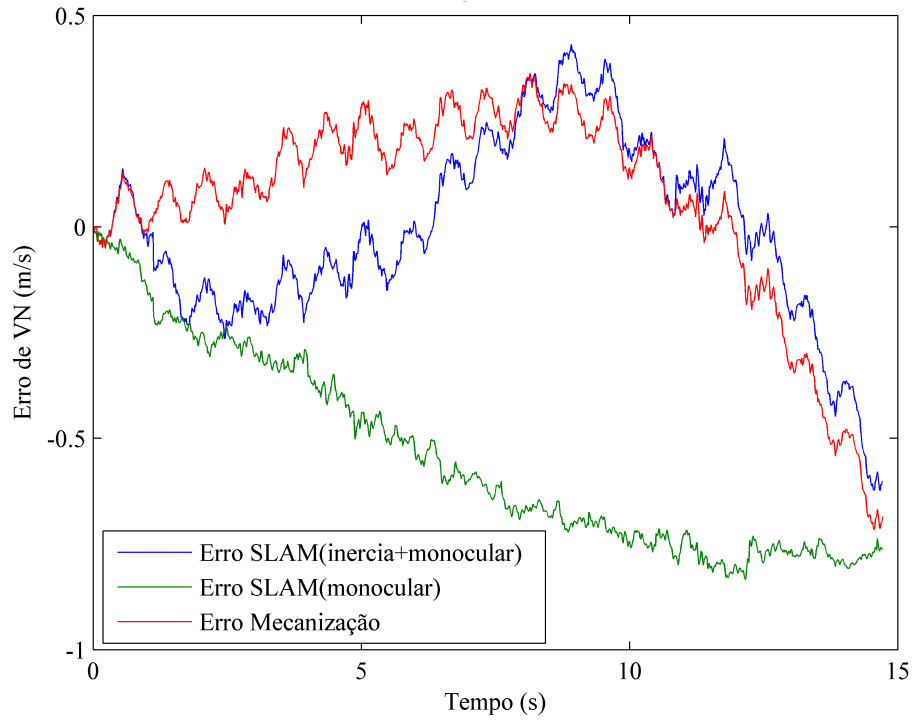


Figura 8.18: Erros em velocidade segundo a direcção Norte.

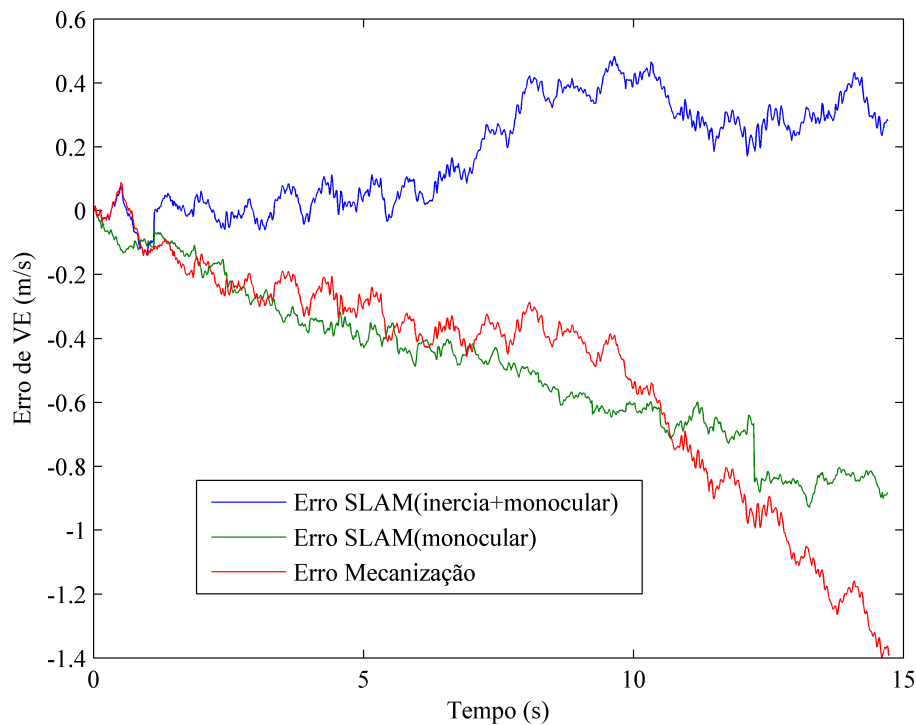


Figura 8.19: Erros em velocidade segundo o a direcção Este.

A reflexão aqui apresentada permite concluir que a fusão sensorial entre as medidas inerciais e as informações visuais é vantajosa, pois conjuga dois tipos de informação complementares, atingindo um desempenho superior ao manifestado pelas implementações isoladas de ambos os métodos.

Os erros em orientação encontram-se deturpados pelas imprecisões contidas nas transformações entre referenciais. O comportamento dos erros contidos nos estados métricos indicia, para a localização obtida por mecanização, um erro em *yaw* superior, enquanto que relativamente aos métodos de SLAM o erro calculado parece ser exagerado. Esta afirmação é aliás corroborada pelo aspecto geométrico das trajectórias calculadas por cada método.

8.3 Construção do Modelo

A primeira etapa do sistema de construção do modelo foi desenvolvida em Matlab e consiste na criação de uma nuvem de pontos a partir dos varrimentos do LRF lateral. Com esse propósito, as medições, inicialmente definidas em coordenadas polares – orientação e distância – são convertidas em coordenadas cartesianas, de acordo com a orientação do sistema de eixos atribuído ao sensor (figura 8.20):

$$\begin{bmatrix} x^{LRF} \\ y^{LRF} \\ z^{LRF} \end{bmatrix} = \begin{bmatrix} 0 \\ -\text{distância} * \cos(\text{orientação}) \\ \text{distância} * \sin(\text{orientação}) \end{bmatrix} \quad (8.16)$$

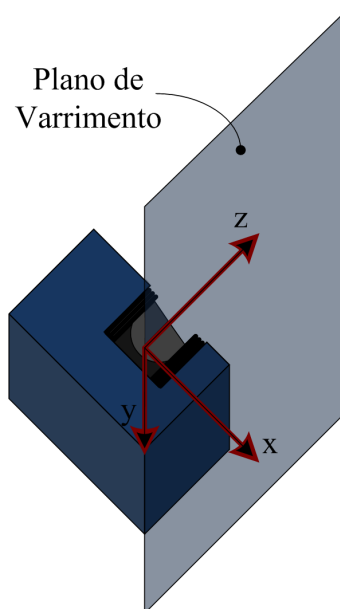


Figura 8.20: Disposição do sistema de eixos do sensor LRF lateral.

Os pontos em coordenadas cartesianas são sujeitos a uma sucessão de transformações, passando pelos referenciais intermédios da câmara direita e câmara esquerda, até atingir o referencial do corpo. As transformações em causa foram determinadas por intermédio da *toolbox* [120].

Uma vez definidos segundo o referencial do corpo, procede-se à determinação da posição e orientação de cada varrimento segundo o referencial de nível local, considerando a localização obtida pelo algoritmo de SLAM. A associação entre um determinado varrimento e o instante de localização respectivo é estabelecida de acordo com a proximidade entre

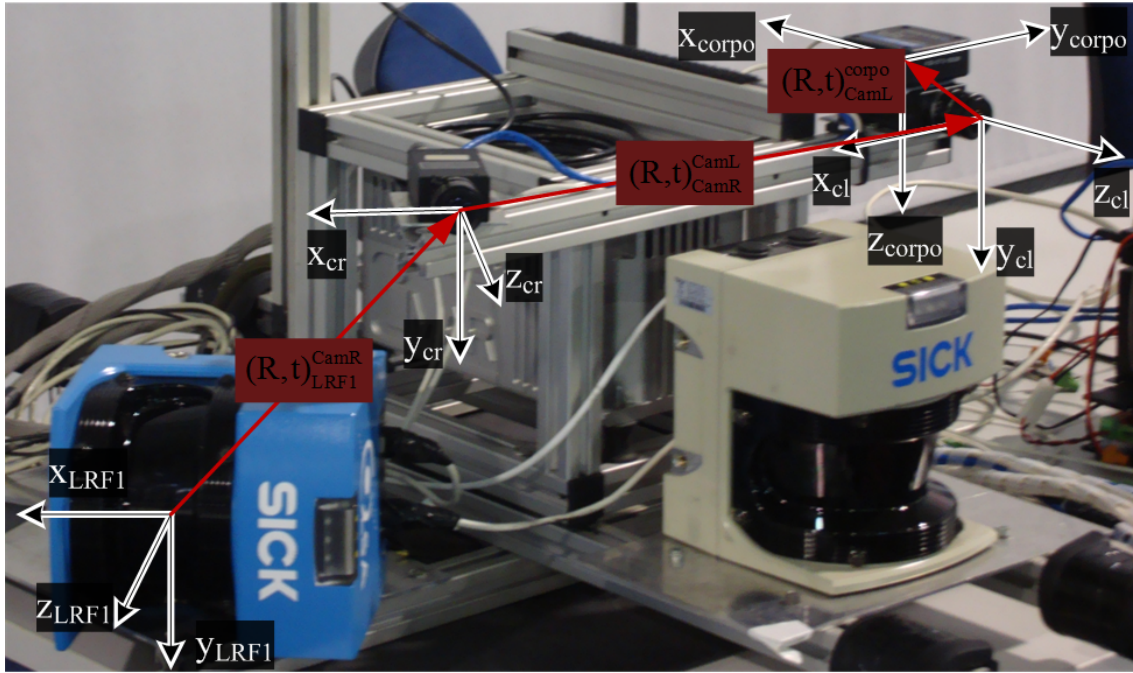


Figura 8.21: Sequência de transformações que, quando aplicada aos ponto no referencial do sensor LRF lateral, os define segundo o referencial do corpo.

os instantes de tempo de ambos. Dado que os instantes de tempo da localização não são exactamente coincidentes com os instantes de amostragem do sensor LRF, é calculada uma aproximação da localização para cada varrimento a partir das localizações cronologicamente mais próximas (fórmula 8.17).

$$x_{aprox}^n = x_{anterior}^n + (x_{posterior}^n - x_{anterior}^n) \frac{t_{x_{anterior}^n} - t_{varr}}{t_{x_{posterior}^n} - t_{x_{anterior}^n}} \quad (8.17)$$

Por fim, o registo do varrimento segundo o referencial de nível local é conseguido pela aplicação da transformação calculada a partir da localização aproximada.

$$\begin{bmatrix} x_{varr}^N \\ y_{varr}^E \\ z_{varr}^D \end{bmatrix} = C_b^n \begin{bmatrix} x_{varr}^b \\ y_{varr}^b \\ z_{varr}^b \end{bmatrix} + \begin{bmatrix} x_{aprox}^N \\ y_{aprox}^E \\ z_{aprox}^D \end{bmatrix} \quad (8.18)$$

O posicionamento dos varrimentos relativamente a um sistema referencial comum resulta numa nuvem de ponto, que representa de forma realista as características geométricas da estrutura.

Por fim, a nuvem de pontos é codificada num ficheiro segundo a linguagem VRML, com

vista à exibição num programa de modelação tridimensional. Neste trabalho foi utilizado o programa *open source* Meshlab [128], que, para além fomentar a interação com o modelo, disponibiliza algoritmos vocacionados para a construção de malhas poligonais e tratamento das mesmas.

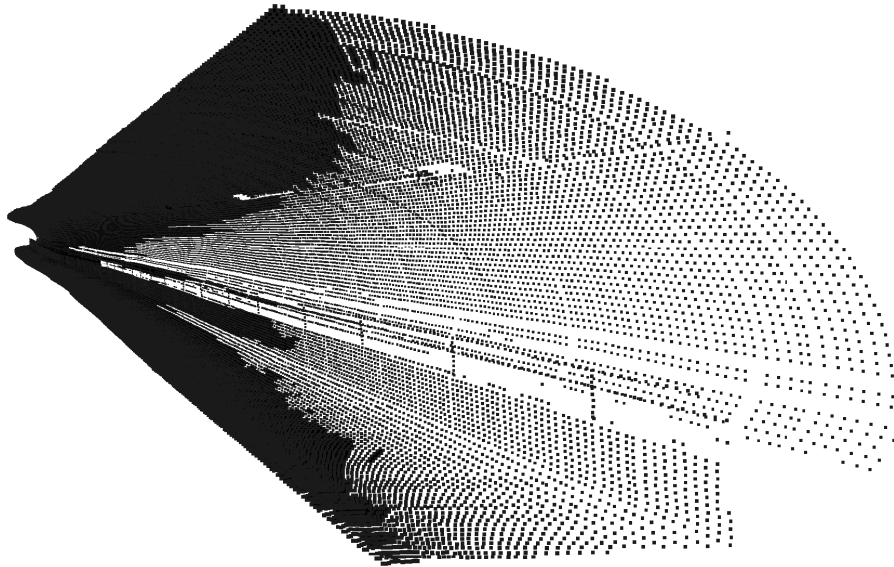


Figura 8.22: Nuvem de pontos do túnel visualizada a partir de uma das extremidades.

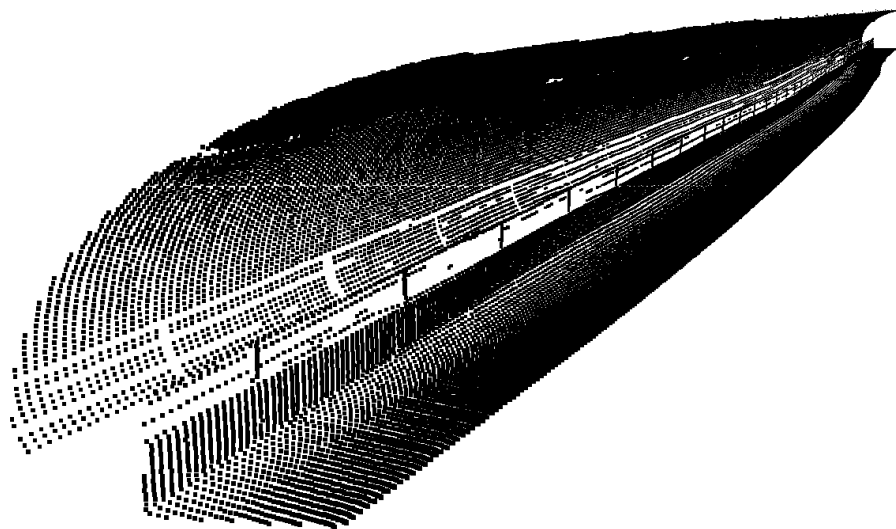


Figura 8.23: Nuvem de pontos vista de outra perspectiva.



Figura 8.24: Nuvem de pontos de toda a estrutura.

Como anteriormente referido, a representação de modelos tridimensionais sob a forma de nuvens de pontos pode causar dificuldades de interpretação, dada a ausência de superfícies. Este fenómeno acentua-se à medida que aumenta a ampliação da imagem (figura 8.25).

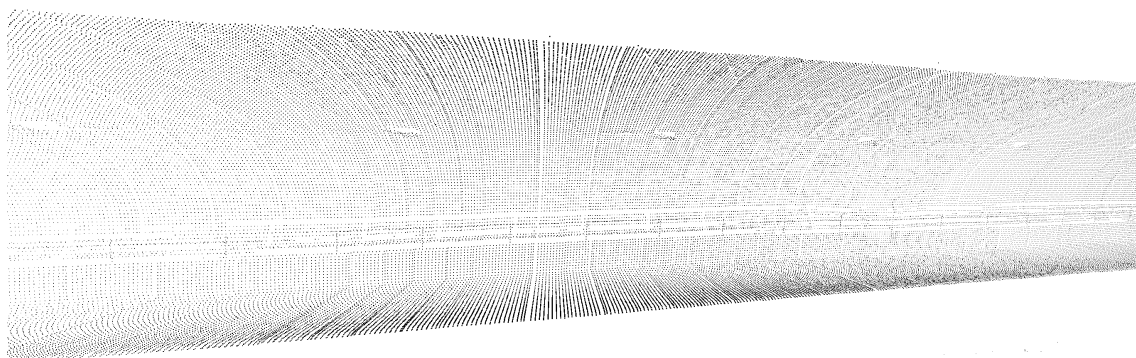


Figura 8.25: Secção ampliada da nuvem de pontos.

A representação através de uma malha poligonal permite uma interpretação mais imediata, como se pode verificar na imagem da figura (8.26).

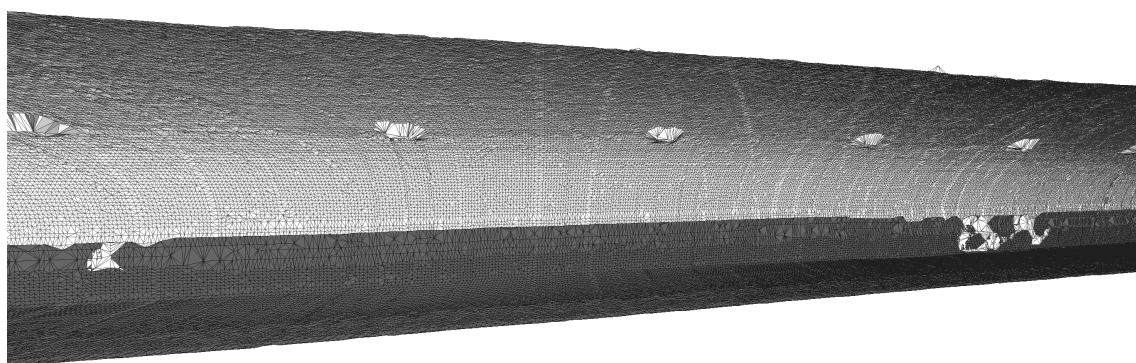


Figura 8.26: Secção convertida em malha poligonal.

As malhas poligonais apresentadas neste documento são gerada pelo algoritmo BPA, que opera sobre a nuvem de pontos inicialmente construída. É utilizada a implementação do algoritmo BPA disponível no programa Meshlab.

Numa representação em malha, a estrutura pode ser realçada através da incidência de luz sobre as faces opacas. A linguagem VRML permite definir pontos de luz, de modo a adequar a iluminação ao modelo. A atribuição de uma cor característica, de acordo com a orientação das faces de cada polígono, promove a diferenciação entre superfícies, facilitando a interpretação do modelo.

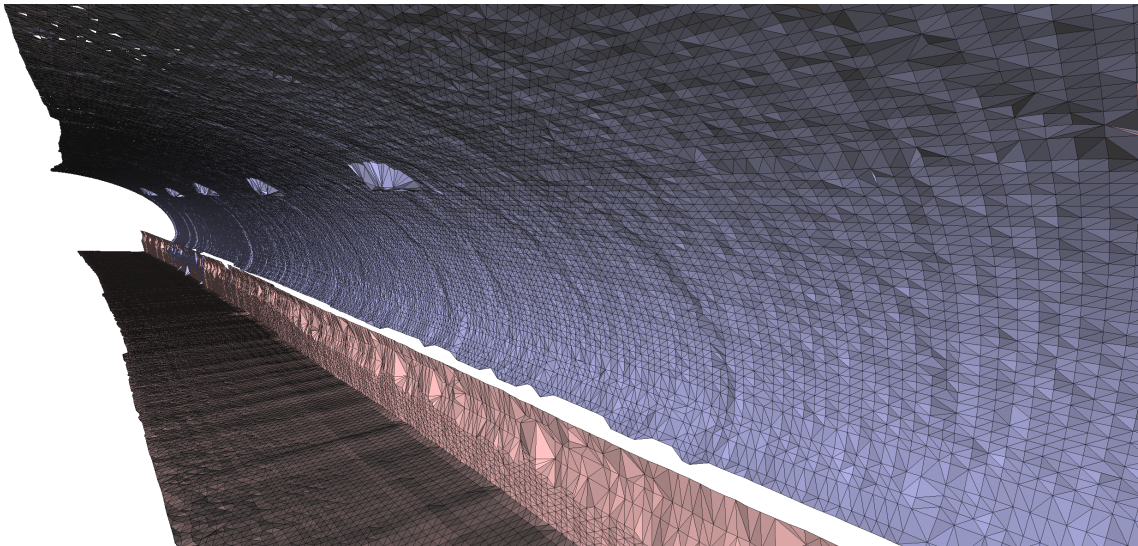


Figura 8.27: Malha poligonal com iluminação.

A iluminação contribui para o realismo da cena e possibilita a omissão das fronteiras dos polígonos, potenciando a constituição de superfícies contínuas.

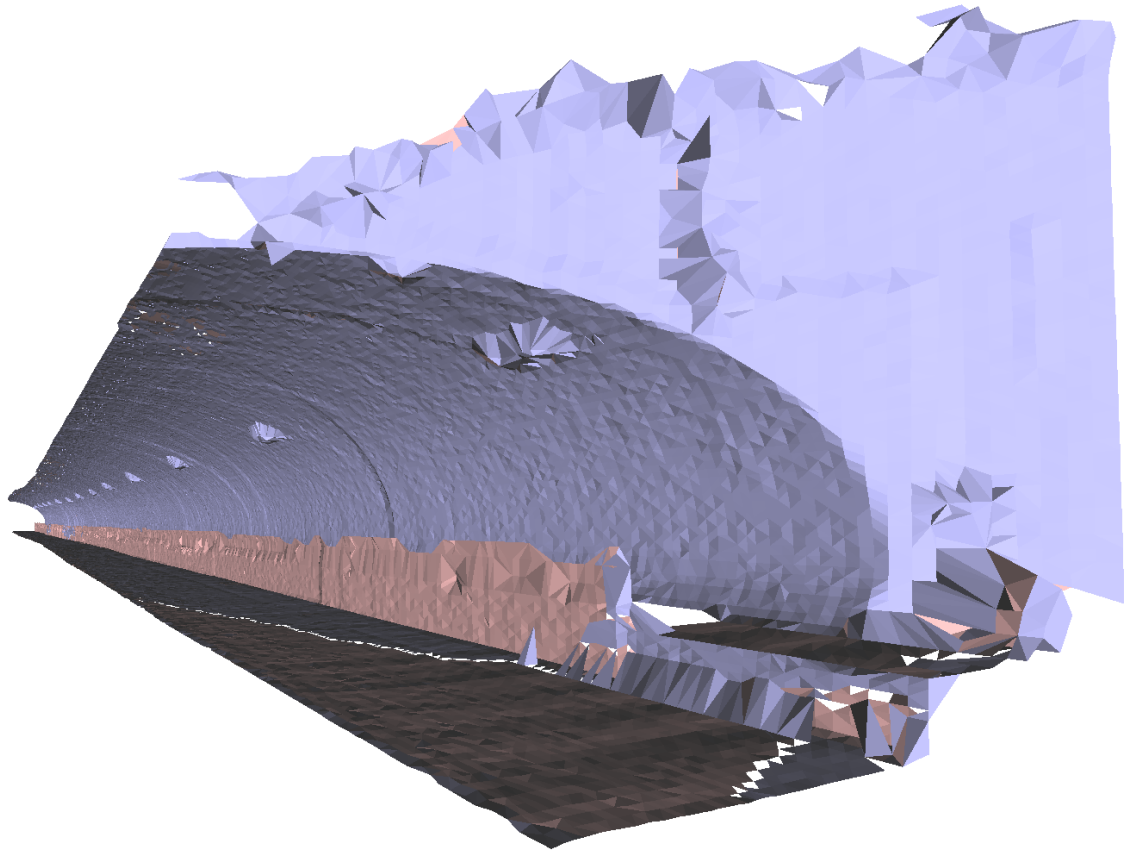


Figura 8.28: Superfícies resultantes da eliminação das fronteiras entre polígonos.

Finalmente, as irregularidades, originadas pela natureza poligonal do modelo, são atenuadas pela execução de sucessivas etapas de filtragem, resultando num cenário mais uniforme.

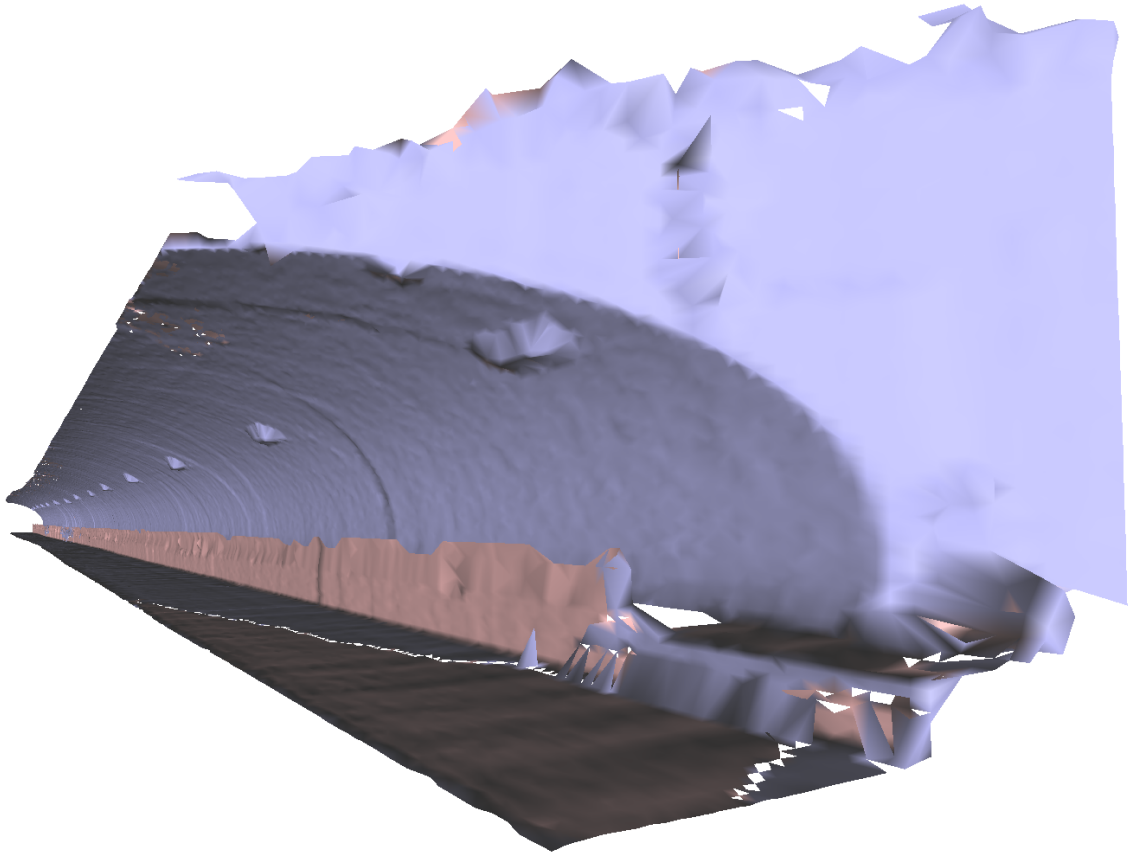


Figura 8.29: Modelo final após etapa de filtragem geométrica.

Esta página foi intencionalmente deixada em branco.

Capítulo 9

Conclusões e Trabalho Futuro

Conteúdo

9.1	Conclusões	153
9.2	Trabalho Futuro	154

9.1 Conclusões

Ao longo deste documento ficou patente a interdependência entre processos de localização e mapeamento. A construção de um modelo coerente e exacto, através de um conjunto de varrimentos adquiridos por sensores LRF 2D, implica uma localização exacta e precisa, pois esta serve de base ao processo de registo. Ambientes *underground* caracterizados por uma elevada monotonia estrutural, como é o caso dos túneis, originam medições sensoriais ambíguas, o que dificulta a implementação de processos robustos de localização autónoma.

Os sistemas de modelização analisados no capítulo do estado da arte, não se encontram preparados para operar num ambiente tão hostil como aquele a que nos propomos. Por isso, foi necessário estudar técnicas mais promissoras voltadas para a localização autónoma, pois com o problema de localização resolvido a tarefa de mapeamento torna-se trivial.

Nesse sentido foi identificada a técnica de SLAM, que pela fusão de informações sensoriais complementares permitiu obter uma boa estimativa da localização, ultrapassando as dificuldades impostas pelo cenário. Neste contexto, constatou-se a eficácia dos dados visuais na correcção dos desvios de trajectória característicos do sistema de navegação inercial. Por outro lado, ficou provado que a introdução de medidas inerciais previne a

degeneração do factor de escala, que normalmente sucede nos algoritmos de SLAM monoculares. Corroboramos assim os resultados apresentados em [123].

Ao optar pela parametrização de *features* em profundidade inversa foi possível introduzir directamente no algoritmo EKF *landmarks* com uma grande incerteza em profundidade.

As imagens adquiridas foram tratadas à margem do algoritmo principal. Esta estratégia foi fundamental para o sucesso do algoritmo de estimação, pois possibilitou a selecção das *features* com maior repetibilidade. A utilização do detector de *features* SIFT e a correspondência de pontos de interesse pela comparação de descritores revelaram-se bastante eficazes pois, apesar das severas perturbações verificadas nas imagens, foi possível identificar pontos de interesse com grande número de observações ao longo da sequência de frames.

Foi implementado um método de registo de varrimentos assente na localização estimada, com vista à construção de um modelo em nuvem de pontos. A conversão da nuvem de pontos numa representação em malha poligonal eleva o realismo do modelo e facilita a sua interpretação.

9.2 Trabalho Futuro

Apesar dos resultados satisfatórios conseguidos, o aperfeiçoamento de certos aspectos e a inclusão de mais informações sensoriais podem contribuir para o melhoramento de precisão da localização e do modelo e ao mesmo tempo acrescentar robustez ao sistema, permitindo o mapeamento de galerias mais extensas. Nesse sentido ficam registadas algumas propostas de trabalho futuro.

Um aspecto essencial passa pelo melhoramento do sistema de aquisição de imagens, pois a utilização de câmaras desprovidas de ganho automático compromete a qualidade e reduz a quantidade de *features* detectadas. Pode ainda ser ponderada a inclusão de um sistema de iluminação a bordo do veículo, proporcionando um acréscimo de robustez ao sistema, tornando-o apto a operar noutro tipo de cenário, como túneis ferroviários não iluminados por exemplo.

A exactidão das transformações entre os sistemas referenciais de sensores toma grande importância, pois produz um impacto directo na qualidade da trajectória estimada. Neste sentido, devem ser desenvolvidas metodologias de calibração capazes de fornecer uma estimativa rigorosa destes parâmetros.

Outra melhoria oportuna diz respeito à estimação dos erros *bias* dos sensores inerciais, que poderia ser conseguida pela introdução destes estados no filtro de Kalman estendido.

Podem ainda ser introduzidas no sistema de localização informações provenientes de outras fontes, nomeadamente visão *stereo* e registo de varrimentos LRF. A implementação de visão *stereo* iria permitir a determinação de distâncias relativas entre o veículo e os pontos de interesse num único instante de observação, ao passo que o registo de varrimentos poderá fornecer boas indicações sobre a orientação e até posição, caso as paredes da galeria possuam relevos.

A observação de linhas pelo sistema visual constitui igualmente uma ideia promissora, especialmente em túneis rodoviários, onde as marcações no asfalto são uma constante. As linhas podem ser inseridas no mapa de SLAM à semelhança do que acontece com as *landmarks*. Seria igualmente interessante a instalação de uma câmara orientada no sentido lateral do veículo, de modo a observar *features* com elevada paralaxe, introduzindo assim uma fonte de informação sobre o deslocamento segundo a direcção do movimento.

Como os valores de profundidade inversa definem pontos de linearização, uma má aproximação inicial causa erros de linearização que afectam as observações seguintes, levando a inconsistências no próprio filtro [123]. A direcção da *landmark*, obtida pelo sistema de visão no instante inicial, pode ser cruzada com as medidas do sensor LRF frontal, com o intuito de determinar uma melhor aproximação da profundidade inversa inicial para cada *landmark*.

Esta página foi intencionalmente deixada em branco.

Referências

- [1] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert D. Fox, D. Fox, D. Hähnel, N. Roy, D. Schulz, and C. Rosenberg. Probabilistic Algorithms and the Interactive Museum Tour-Guide Robot Minerva. *International Journal of Robotics Research*, 19:972–999, 2000.
- [2] Federico Bertolli, Patric Jensfelt, and Henrik I. Christensen. SLAM using Visual Scan-Matching with Distinguishable 3D Points. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS'06)*, 2006.
- [3] Wolfram Burgard, Armin B. Cremers, Dieter Fox, Dirk Hähnel, Gerhard Lakemeyer, Dirk Schulz, Walter Steiner, and Sebastian Thrun. Experiences with an interactive museum tour-guide robot. *Artif. Intell.*, 114(1-2):3–55, 1999.
- [4] Rudolph Triebel, Barbara Frank, Jörg Meyer, and Wolfram Burgard. First steps towards a robotic system for flexible volumetric mapping of indoor environments. In *In Proceedings of the 5th IFAC Symposium on Intelligent Autonomous Vehicles*, 2004.
- [5] Illah R. Nourbakhsh, Alvaro Soto, Judith Bobenage, Sebastien Grange, Roland Meyer, and Ron Lutz. An effective mobile robot educator with a full-time job. *Artificial Intelligence*, 114(1-2):95–124, 1999.
- [6] J. Borenstein and Y. Koren. Real-time Obstacle Avoidance for Fast Mobile Robots in Cluttered Environments. *IEEE Transactions on Systems, Man, and Cybernetics*, 19:1179–1187, 1990.
- [7] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping. pages 321–328, 2000.
- [8] Roland Siegwart and Illah R. Nourbakhsh. *Introduction to Autonomous Mobile Robots*. Bradford Company, Scituate, MA, USA, 2004.
- [9] S. F. El-Hakim, P. Boulanger, F. Blais, and J.-A. Beraldin. Sensor Based Creation of Indoor Virtual Environment Models. In *VSMM '97: Proceedings of the 1997 International Conference on Virtual Systems and MultiMedia*, page 50, Washington, DC, USA, 1997. IEEE Computer Society.
- [10] Kai Pervözl, Andreas Nüchter, Hartmut Surmann, and Joachim Hertzberg. Automatic Reconstruction of Colored 3D Models. In *Proceedings Robotik*, pages 215–222, 2004.

- [11] Tommaso Gramegna, Giovanni Attolico, and Arcangelo Distante. Planar Map and 3D Model Construction Using a Mobile robot. *22nd International Symposium on Automation and Robotics in Construction (ISARC)*, 2005.
- [12] Yiyong Sun, Christophe Dumont, and Mongi A. Abidi. Mesh-based integration of range and color images. In *Proceedings of SPIE, 4th Conference on Sensor Fusion: Architectures, Algorithms, and Applications*, volume 4051, pages 110–117, 2000.
- [13] D. Hähnel, W. Burgard, and S. Thrun. Learning Compact 3D Models of Indoor and Outdoor Environments with a Mobile Robot. *Robotics and Autonomous Systems*, 44(1):15–27, 2003.
- [14] Michael Montemerlo, Dirk Hähnel, David Ferguson, Rudolph Triebel, Wolfram Burgard, Scott Thayer, William Whittaker, and Sebastian Thrun. A system for three-dimensional robotic mapping of underground mines. Technical report, Carnegie Mellon University, Computer Science Department, 2002.
- [15] Dave Ferguson. Volumetric Mine Mapping with 3D Registration. Technical report, Carnegie Mellon University, Robotics Institute, 2002.
- [16] Sebastian Thrun, Dirk Hähnel, David Ferguson, Michael Montemerlo, Rudolph Triebel, Wolfram Burgard, Christopher Baker, Zachary Omohundro, Scott, Scott Thayer, and William Whittaker. A System for Volumetric Robotic Mapping of Abandoned Mines.
- [17] REIGL. RIEGL Laser Measurement System, 2010. <http://www.riegl.com/>.
- [18] Velodyne. High Definition Lidar, 2010. <http://www.velodyne.com/lidar/>.
- [19] Daniel Huber and Nicolas Vandapel. Automatic Three-dimensional Underground Mine Mapping. *The International Journal of Robotics Research*, 25(1):7–17, January 2006.
- [20] Hartmut Surmann, Andreas Nüchter, and Joachim Hertzberg. An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments, 2003.
- [21] M. Montemerlo and S. Thrun. A Multi-Resolution Pyramid for Outdoor Robot Terrain Perception. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 464–469, San Jose, CA, 2004. AAAI.
- [22] Paul J. Besl and Neil D. McKay. A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.
- [23] Joaquim Salvi, Carles Matabosch, David Fofi, and Josep Forest. A review of recent range image registration methods with accuracy evaluation. *Image and Vision Computing*, 25(5):578–596, 2007.
- [24] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 71–78, New York, NY, USA, 1992. ACM.

- [25] Angel Domingo Sappa and Miguel Angel Garcia. Incremental Multiview Integration of Range Images. *International Conference on Pattern Recognition*, 1:1546, 2000.
- [26] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The Ball-Pivoting Algorithm for Surface Reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359, 1999.
- [27] Hong Zhou, Yonghuai Liu, and Longzhuang Li. Incremental Mesh-Based Integration of Registered Range Images: Robust to Registration Error and Scanning Noise. Secaucus, NJ, USA, 2006. Springer-Verlag New York, Inc.
- [28] Andreas Nüchter, Hartmut Surmann, Kai Lingemann, Joachim Hertzberg, and S. Thrun. 6D SLAM with an Application in Autonomous Mine Mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1998–2003, 2004.
- [29] Pulli. Multiview Registration for Large Data Sets. In *Second International Conference on 3-D Digital Imaging and Modeling*, pages 160–168, 1999.
- [30] Gerhard Roth and Eko Wibowo. A Fast Algorithm for Making Mesh-Models from Multiple-View Range Data. In *Proceedings of the DND/CSA Robotics and Knowledge Based Systems Workshop*, pages 349–356, 1995.
- [31] Peter K. Allen, Ioannis Stamos, A. Troccoli, B. Smith, M. Leordeanu, and Y. C. Hsu. 3D Modeling of Historic Sites Using Range and Image Data. In *Proceedings of International Conference on Robotics and Animation (ICRA)*, pages 11–16, 2003.
- [32] Feng Lu and Evangelos Milios. Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans. *Journal of Intelligent and Robotic Systems*, 18(3):249–275, 1997.
- [33] Michael Erdmann, Eric Grimson Mit, David A. Simon, and David A. Simon. Fast and Accurate Shape-Based Registration. Technical report, Carnegie Mellon University, Pittsburgh, PA, USA, 1996.
- [34] Andreas Nüchter. Parallel and cached scan matching for robotic 3D mapping. *Journal of Computing and Information Technology*, 17(1), 2009.
- [35] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698–700, 1987.
- [36] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, 1987.
- [37] Michael W. Walker, Lejun Shao, and Richard A. Volz. Estimating 3-D location parameters using dual number quaternions. *CVGIP: Image Understanding*, 54(3):358–367, 1991.
- [38] Berthold K. P. Horn, H.M. Hilden, and Shariar Negahdaripour. Closed-Form Solution of Absolute Orientation using Orthonormal Matrices. *Journal of the Optical Society America*, 5(7):1127–1135, 1988.

- [39] A. Lorusso, D. W. Eggert, and R. B. Fisher. A comparison of four algorithms for estimating 3-D rigid transformations. In *BMVC '95: Proceedings of the 1995 British conference on Machine vision (Vol. 1)*, pages 237–246, Surrey, UK, UK, 1995. BMVA Press.
- [40] Szymon Rusinkiewicz and Marc Levoy. Efficient Variants of the ICP Algorithm. In *International Conference on 3-D Digital Imaging and Modeling*, 2001.
- [41] Timothée Jost and Heinz Hügli. A Multi-Resolution Scheme ICP Algorithm for Fast Shape Registration. In *1st International Symposium on 3D Data Processing Visualization and Transmission*, pages 540–543. IEEE Computer Society, 2002.
- [42] Esther Ezra, Micha Sharir, and Alon Efrat. On the ICP algorithm. In *SCG '06: Proceedings of the twenty-second annual symposium on Computational geometry*, pages 95–104, New York, NY, USA, 2006. ACM.
- [43] M. Greenspan and G. Godin. A Nearest Neighbor Method for Efficient ICP. *3D Digital Imaging and Modeling, International Conference on*, 0:161, 2001.
- [44] Timo Zinßer, Jochen Schmidt, and Heinrich Niemann. Performance Analysis of Nearest Neighbor Algorithms for ICP Registration of 3-D Point Sets. In *ICP Registration of 3-D Point Sets, In Vision, Modeling, and Visualization 2003*, pages 199–206, 2003.
- [45] Council Canada, M. A. Greenspan, G. Godin, J. Talbot, Michael Greenspan, Guy Godin, and Jimmy Talbot. Acceleration of Binning Nearest Neighbor Methods. In *Proceedings of Vision Interface*, 2000.
- [46] Jerome H. Friedman, Jon Louis Bentley, and Raphael Ari Finkel. An Algorithm for Finding Best Matches in Logarithmic Expected Time. *ACM Transactions on Mathematical Software*, 3(3):209–226, 1977.
- [47] Zhengyou Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994.
- [48] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, 1975.
- [49] Robert F. Sproull. Refinements to Nearest-Neighbor Searching in k-Dimensional Trees. *Algorithmica*, 6(4):579–589, 1991.
- [50] Andreas Nüchter, Kai Lingemann, and Joachim Hertzberg. Cached k-d tree search for ICP algorithms. In *3DIM '07: Proceedings of the Sixth International Conference on 3-D Digital Imaging and Modeling*, pages 419–426, Washington, DC, USA, 2007. IEEE Computer Society.
- [51] R. Madhavan and E. Messina. Iterative Registration of 3D LADAR Data for Autonomous Navigation. In *In Proceedings of the IEEE Intelligent Vehicles Symp*, pages 186–191, 2003.
- [52] T. Masuda, K. Sakaue, and N. Yokoya. Registration and Integration of Multiple Range Images for 3-D Model Construction. In *ICPR '96: Proceedings of the 1996 International Conference on Pattern Recognition (ICPR '96) Volume I*, page 879, Washington, DC, USA, 1996. IEEE Computer Society.

- [53] Bayu Slamet and Max Pfingsthorn. ManifoldSLAM : a Multi-Agent Simultaneous Localization and Mapping System for the RoboCup Rescue Virtual Robots Competition. Master's thesis, Universiteit van Amsterdam, 2006.
- [54] Javier Minguez, Florent Lamiroux, and Luis Montesano. Metric-Based Scan Matching Algorithms for Mobile Robot Displacement Estimation. In *International Conference on Robotics and Automation (ICRA)*, pages 3557–3563, 2005.
- [55] J. Minguez, L. Montesano, and F. Lamiroux. Metric-based iterative closest point scan matching for sensor displacement estimation. *IEEE Transactions on Robotics*, 22(5):1047–1054, oct. 2006.
- [56] Yangming Li and Edwin Olson. Extracting general-purpose features from LIDAR data. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2010.
- [57] I. Cox. Blanche - An experiment in guidance and navigation of an autonomous robot vehicle. In *IEEE Transactions on Robotics and Automation*, pages 193–204, 1991.
- [58] Jens steffen Gutmann and Christian Schlegel. AMOS: Comparison of Scan Matching Approaches for Self-Localization in Indoor Environments, 1996.
- [59] Kai Lingemann, Hartmut Surmann, Andreas Nüchter, and Joachim Hertzberg. Indoor and Outdoor Localization for Fast Mobile Robots. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-04)*, pages 2185–2190, 2004.
- [60] A. A. Aghamohammadi, H. D. Taghirad, A. H. Tamjidi, and E. Mihankhah. Feature-based laser scan matching for accurate and high speed mobile robot localization. In *3rd European Conference on Mobile Robots*, 1997.
- [61] Peter Biber and Wolfgang Straßer. The Normal Distribution Transform: A New Approach to Laser Scan Matching. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 3, 2003.
- [62] G. Weiss and E. von Puttkamer. A map based on laserscans without geometric interpretation. In *Proceedings of Intelligent Autonomous Systems 4 (IAS-4)*, pages 403–407, 1995.
- [63] C. Harris and M. Stephens. A Combined Corner and Edge Detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, 1988.
- [64] Hans Peter Moravec. *Obstacle avoidance and navigation in the real world by a seeing robot rover*. PhD thesis, Stanford, CA, USA, 1980.
- [65] David G. Lowe. Object Recognition from Local Scale-Invariant Features. In *ICCV '99: Proceedings of the International Conference on Computer Vision*, volume 2, pages 1150–1157, Washington, DC, USA, 1999. IEEE Computer Society.
- [66] Jan J. Koenderink. The Structure of Images. *Biological Cybernetics*, 50(5):363–370–370, August 1984.
- [67] Tony Lindeberg. Scale-space Theory: A Basic Tool for Analysing Structures at Different Scales. *Journal of Applied Statistics*, pages 224–270, 1994.

- [68] David A. Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference, 2002.
- [69] Tony Lindeberg and Lars Bretzner. Real-time Scale Selection in Hybrid Multi-scale Representations. pages 148–163. Springer-Verlag, 2003.
- [70] Luigi Di Stefano, Stefano Mattoccia, and Federico Tombari. ZNCC-based template matching using bounded partial correlation. *Pattern Recognition Letters*, 26(14):2129–2134, 2005.
- [71] Camera Calibration Toolbox for Matlab, 2010. http://www.vision.caltech.edu/bouguetj/calib_doc/.
- [72] Tsai Camera Calibration Software. <http://www.cs.cmu.edu/~rgw/TsaiCode.html>.
- [73] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 052154051, second edition, 2004.
- [74] Duane C. Brown. Close-Range Camera Calibration. *Photogrammetric Engineering*, 37(8):855–866, 1971.
- [75] Joss Knight. Robot Navigation by Active Stereo Fixation. Technical report, Active Vision Lab, University of Oxford, 1999.
- [76] Elan Dubrofsky and Robert J. Woodham. Combining Line and Point Correspondences for Homography Estimation. In *ISVC '08: Proceedings of the 4th International Symposium on Advances in Visual Computing, Part II*, pages 202–213. Springer-Verlag, 2008.
- [77] Richard I. Hartley. Theory and Practice of Projective Rectification. *Int. J. Comput. Vision*, 35(2):115–127, 1999.
- [78] Charles Loop and Zhengyou Zhang. Computing Rectifying Homographies for Stereo Vision. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:1125, 1999.
- [79] Andrea Fusiello, Emanuele Trucco, and Alessandro Verri. A compact algorithm for rectification of stereo pairs. *Mach. Vision Appl.*, 12(1):16–22, 2000.
- [80] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. pages 726–740, 1987.
- [81] K. Konolige, M. Agrawal, and Joan Solà. Large Scale Visual Odometry for Rough Terrain. In *Proc. International Symposium on Research in Robotics (ISRR)*, November 2007.
- [82] Bernd Kitt, Andreas Geiger, and Henning Lategahn. Visual Odometry Based on Stereo Image Sequences with RANSAC-based Outlier Rejection Scheme. June 2010.
- [83] David Nistér. An Efficient Solution to the Five-Point Relative Pose Problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(6):756–777, 2004.

- [84] D. Ortín and J. M. M. Montiel. Indoor robot motion based on monocular images. *Robotica*, 19:331–342, May 2001.
- [85] Davide Scaramuzza, Friedrich Fraundorfer, and Roland Siegwart. Real-time monocular visual odometry for on-road vehicles with 1-point RANSAC. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation, ICRA'09*, pages 488–494, 2009.
- [86] Javier Civera, Oscar G. Grasa, Andrew J. Davison, and J. M. M. Montiel. 1-Point RANSAC for EKF-based Structure from Motion. In *(IROS) International Conference on Intelligent Robots and Systems*, pages 3498–3504. IEEE, October 2009.
- [87] Javier Civera, Oscar G. Grasa, Andrew J. Davison, and J. M. M. Montiel. 1-Point RANSAC for EKF Filtering. Application to Real-Time Structure from Motion and Visual Odometry. *Journal of Field Robotics*, 27:609–631, September 2010.
- [88] Chris Engels, Henrik Stewénus, and David Nistér. Bundle Adjustment Rules. In *Photogrammetric Computer Vision*, 2006.
- [89] J. Borenstein, H. R. Everett, and L. Feng. Where am I? – Systems and Methods for Mobile Robot Positioning. Technical report, University of Michigan, 1996.
- [90] Tim Bailey. Mobile Robot Localization and Mapping in Extensive Outdoor Environments, 2002.
- [91] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. pages 167–193, 1990.
- [92] Kalman, Rudolph, and Emil. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D):35–45, 1960.
- [93] S. Thrun. Robotic Mapping: A Survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millennium*. Morgan Kaufmann, 2002.
- [94] Hugh Durrant-Whyte and Tim Bailey. Simultaneous Localization and Mapping (SLAM): Part I The Essential Algorithms. *IEEE Robotics and Automation Magazine*, 2, 2006.
- [95] Jose Guivant and Eduardo Nebot. Optimization of the Simultaneous Localization and Map Building Algorithm for Real Time Implementation. *IEEE Transactions on Robotics and Automation*, 17:242–257, 2001.
- [96] John J. Leonard, Hans Jacob, and S. Feder. A Computationally Efficient Method for Large-scale Concurrent Mapping and Localization. In *Proceedings of the Ninth International Symposium on Robotics Research*, pages 169–176. Springer-Verlag, 1999.
- [97] M. Sanjeev Arulampalam, Simon Maskell, and Neil Gordon. A Tutorial on Particle Filters for Online Nonlinear/non-Gaussian Bayesian Tracking. *IEEE Transaction on Signal Processing*, 50:174–188, 2002.

- [98] Arnaud Doucet, Nando de Freitas, Kevin P. Murphy, and Stuart J. Russell. Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks. In *UAI '00: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pages 176–183, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.
- [99] Michael Montemerlo. *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburg, PA, July 2003.
- [100] Michael Montemerlo and Sebastian Thrun, Daphne Koller, and Ben Wegbreit. FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, pages 593–598. AAAI, 2002.
- [101] *An efficient fastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements*, volume 1, October 2003.
- [102] G. Grisetti, C. Stachniss, and W. Burgard. Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling. In *ICRA*, pages 2443–2448, 2005.
- [103] Andreas Nüchter, Kai Lingemann, Joachim Hertzberg, and Hartmut Surmann. Heuristic-Based Laser Scan Matching for Outdoor 6D SLAM. In *Advances in artificial intelligence. 28th annual German Conf. on AI*, pages 304–319, 2005.
- [104] Andrew J. Davison. Real-Time Simultaneous Localisation and Mapping with a Single Camera. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 1403, Washington, DC, USA, 2003. IEEE Computer Society.
- [105] Ethan Eade and Tom Drummond. Scalable Monocular SLAM. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 469–476, Washington, DC, USA, 2006. IEEE Computer Society.
- [106] Andrew J. Davison, Javier Civera and J. M. M. Montiel. *IEEE Transactions on Robotics*, 24(5):932–945, October 2008.
- [107] J. Montiel, J. Civera, and A. Davison. Unified Inverse Depth Parametrization for Monocular SLAM. In *Proceedings of Robotics: Science and Systems*, Philadelphia, USA, August 2006.
- [108] Javier Civera. *Real-Time EKF-Based Structure from Motion*. PhD thesis, Universidad de Zaragoza, September 2009.
- [109] Andrew J. Davison, Javier Civera and J. M. M. Montiel. In *3rd Iberian Conference on Pattern Recognition and Image Analysis*.
- [110] J. Neira, J.D. Tardós, L.M. Paz, P. Piniés. 6DOF SLAM With Stereo Camera in Hand. In *Visual SLAM - An Emerging Technology, in IROS 2008*, San Diego, USA., 2008.
- [111] Carlo Tomasi and Takeo Kanade. Detection and Tracking of Point Features. Technical report, International Journal of Computer Vision, 1991.

- [112] R Kurazume, S. Nagata, and S. Hiiose. Cooperative Positioning with Multiple Robots. In *Proceedings IEEE International Conference on Robotics and Automation*, volume 2, pages 1250–1257, 1994.
- [113] Robert Grabowski, Luis Ernesto Navarro-Serment, Chris Paredis, and Pradeep Khosla. Heterogeneous Teams of Modular Robots for Mapping and Exploration. *Autonomous Robots – Special Issue on Heterogeneous Multirobot Systems*, 8:293–308, 2000.
- [114] F. Dunn and I. Parberr. *3D Math Primer for Graphics and Game Development: Theory and Practice in C++*. Wordware Publishing, 2002.
- [115] Linda G. Shapiro, George C. Stockman, Linda G. Shapiro, and George Stockman. *Computer Vision*. Prentice Hall, January 2001.
- [116] MicroStrain, Inclinómetro 3DM-GX, 2010. <http://www.microstrain.com/3dm-gx1.aspx>.
- [117] SICK – Sensor Intelligence, 2010. <http://www.sick.com/group/EN/home/Pages/Homepage1.aspx>.
- [118] JAI, Industrial CCD/CMOS cameras, 2010. <http://www.jai.com/EN/Pages/home.aspx>.
- [119] Qilong Zhand and R. Pless. Extrinsic Calibration of a Camera and Laser Range Finder (Improves Camera Calibration). In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004)*, volume 3, pages 2301–2306, 2004.
- [120] Toolbox para calibração de parâmetros extrínsecos entre câmaras e sensores laser range finder, 2010. <http://research.engineering.wustl.edu/~pless/fusion.html>.
- [121] Jorge Lobo and Jorge Dias. InerVis – Toolbox de calibração de parâmetros extrínsecos entre câmaras e sensores inerciais, 2010. http://www2.deec.uc.pt/~jlobo/InerVis_WebIndex/InerVis_Toolbox.html.
- [122] Jorge Lobo and Jorge Dias. Relative pose calibration between visual and inertial sensors. *International Journal of Robotics Research*, 26(6):561–575, 2007.
- [123] Pedro Pinies, Todd Lupton, Salah Sukkarieh, and Juan D. Tardós. Inertial Aiding of Inverse Depth SLAM using a Monocular Camera. In *International Conference on Robotics and Automation (ICRA)*, pages 2797–2802. IEEE, 2007.
- [124] Javier Civera and J. M. M. Montiel. 1-Point RANSAC Inverse Depth EKF Monocular SLAM Matlab Code. Version 1.0, 2010. <http://webdiis.unizar.es/~jcivera/code/1p-ransac-ekf-monoslam.html>.
- [125] Andrea Vedaldi. SIFT for Matlab, 2011. <http://www.vlfeat.org/~vedaldi/code/sift.html>.
- [126] Marco Zuliani. RANSAC Toolbox for Matlab, 2011. <http://vision.ece.ucsb.edu/~zuliani/Research/RANSAC/RANSAC.shtml>.

- [127] Hugh Durrant-Whyte. Multi Sensor Data Fusion. Technical report, Australian Centre for Field Robotics, University of Sydney, 2001.
- [128] Meshlab – 3D mesh processing software, 2011. <http://www.meshlab.org/>.

Anexo A

Cálculo dos jacobianos

Este anexo é dedicado à obtenção das matrizes jacobianas utilizadas no mecanismo do EKF.

A.1 Jacobianos do Modelo de Observação em Ordem aos Estados do Filtro

O jacobiano de modelo de observação em ordem aos estados do filtro pode ser segmentado no jacobiano em ordem aos estados da localização do corpo $\frac{\partial h_i}{\partial x_{loc}}$ e no jacobiano em ordem aos estados das *landmarks* do mapa $\frac{\partial h_i}{\partial x_{map}}$.

$$\nabla H = \frac{\partial h_i}{\partial x} = \left(\frac{\partial h_i}{\partial x_{loc}}, \frac{\partial h_i}{\partial x_{map}} \right) \quad (\text{A.1})$$

A.1.1 Jacobianos do Modelo de Observação em Ordem aos Estados da Localização do Corpo

Visto que os modelos de observação não envolvem a velocidade linear do corpo, o jacobiano em ordem a estes estados é nulo, restando somente as derivadas parciais em ordem à posição e orientação.

$$\frac{\partial h_i}{\partial x_{loc}} = \left(\frac{\partial h_i}{\partial r^n}, \frac{\partial h_i}{\partial \Theta^n}, \underbrace{\frac{\partial h_i}{\partial v^n}}_0 \right) \quad (\text{A.2})$$

As derivadas parciais em ordem à posição são calculadas recorrendo à regra da cadeia, ficando:

$$\frac{\partial h_i}{\partial r^n} = \frac{\partial h_i}{\partial h_d^c} \frac{\partial h_d^c}{\partial h_u^c} \frac{\partial h_u^c}{\partial r^n} \quad (\text{A.3})$$

O jacobiano de h_u^c relativamente a r^n para *landmarks* em *inverse depth* é:

$$\frac{\partial h_u^c}{\partial r^n} = -\rho_i C_n^b C_b^c \quad (\text{A.4})$$

Já para *landmarks* em coordenadas euclidianas é definido como:

$$\frac{\partial h_u^c}{\partial r^n} = -C_n^b C_b^c \quad (\text{A.5})$$

O jacobiano do modelo *pinhole* em ordem às coordenadas $(h_{d_x}^c, h_{d_y}^c)$ fica:

$$\frac{\partial h_i}{\partial h_d^c} = \begin{bmatrix} f_x & 0 \\ 0 & f_y \end{bmatrix} \quad (\text{A.6})$$

Desprezando a componente de distorção tangencial, a matriz jacobiana do modelo de distorção em ordem às componentes do vector direcção h_u^c tem a seguinte forma:

$$\frac{\partial h_d^c}{\partial h_u^c} = \begin{bmatrix} \frac{\partial h_{d_x}^c}{\partial h_{u_x}^c} & \frac{\partial h_{d_x}^c}{\partial h_{u_y}^c} & \frac{\partial h_{d_x}^c}{\partial h_{u_z}^c} \\ \frac{\partial h_{d_y}^c}{\partial h_{u_x}^c} & \frac{\partial h_{d_y}^c}{\partial h_{u_y}^c} & \frac{\partial h_{d_y}^c}{\partial h_{u_z}^c} \end{bmatrix} \quad (\text{A.7})$$

- Onde:

$$\frac{\partial h_{d_x}^c}{\partial h_{u_x}^c} = \frac{1 + k_1 r^2 + k_2 r^4 + k_5 r^6}{h_{u_z}^c} + \frac{2h_{u_x}^c{}^2}{h_{u_z}^c{}^3} \left(k_1 + \frac{2k_2 h_{u_x}^c{}^2}{h_{u_z}^c{}^2} + \frac{3k_5 h_{u_x}^c{}^4}{h_{u_z}^c{}^4} \right) \quad (\text{A.8a})$$

$$\frac{\partial h_{d_x}^c}{\partial h_{u_y}^c} = \frac{2h_{u_x}^c h_{u_y}^c}{h_{u_z}^c{}^3} \left(k_1 + \frac{2k_2 h_{u_y}^c{}^2}{h_{u_z}^c{}^2} + \frac{3k_5 h_{u_y}^c{}^4}{h_{u_z}^c{}^4} \right) \quad (\text{A.8b})$$

$$\frac{\partial h_{d_x}^c}{\partial h_{u_z}^c} = -\frac{h_{u_x}^c (1 + k_1 r^2 + k_2 r^4 + k_5 r^6)}{h_{u_z}^c{}^2} - 2 \frac{h_{u_x}^c{}^3 + h_{u_y}^c{}^2 h_{u_x}^c}{h_{u_z}^c{}^4} \left(k_1 + 2k_2 \frac{h_{u_x}^c{}^2 + h_{u_y}^c{}^2}{h_{u_z}^c{}^2} + 3k_5 \frac{h_{u_x}^c{}^4 + h_{u_y}^c{}^4}{h_{u_z}^c{}^4} \right) \quad (\text{A.8c})$$

$$\frac{\partial h_{d_y}^c}{\partial h_{u_x}^c} = \frac{2h_{u_x}^c h_{u_y}^c}{h_{u_z}^c{}^3} \left(k_1 + \frac{2k_2 h_{u_x}^c{}^2}{h_{u_z}^c{}^2} + \frac{3k_5 h_{u_x}^c{}^4}{h_{u_z}^c{}^4} \right) \quad (\text{A.8d})$$

$$\frac{\partial h_{d_y}^c}{\partial h_{u_x}^c} = \frac{1 + k_1 r^2 + k_2 r^4 + k_5 r^6}{h_{u_z}^c} + \frac{2h_{u_y}^c{}^2}{h_{u_z}^c{}^3} \left(k_1 + \frac{2k_2 h_{u_y}^c{}^2}{h_{u_z}^c{}^2} + \frac{3k_5 h_{u_y}^c{}^4}{h_{u_z}^c{}^4} \right) \quad (\text{A.8e})$$

$$\frac{\partial h_{d_y}^c}{\partial h_{u_z}^c} = -\frac{h_{u_y}^c (1 + k_1 r^2 + k_2 r^4 + k_5 r^6)}{h_{u_z}^c{}^2} - 2 \frac{h_{u_x}^c{}^2 h_{u_y}^c + h_{u_y}^c{}^3}{h_{u_z}^c{}^4} \left(k_1 + 2k_2 \frac{h_{u_x}^c{}^2 + h_{u_y}^c{}^2}{h_{u_z}^c{}^2} + 3k_5 \frac{h_{u_x}^c{}^4 + h_{u_y}^c{}^4}{h_{u_z}^c{}^4} \right) \quad (\text{A.8f})$$

Passemos agora ao jacobiano do modelo de observação em relação à orientação, que usando a regra da cadeia pode ser decomposto em:

$$\frac{\partial h_i}{\partial \Theta^n} = \frac{\partial h_i}{\partial h_d^c} \frac{\partial h_d^c}{\partial h_u^c} \frac{\partial h_u^c}{\partial \Theta^n} \quad (\text{A.9})$$

O cálculo dos jacobianos $\frac{\partial h_i}{\partial h_d^c}$ e $\frac{\partial h_d^c}{\partial h_u^c}$ respeita as equações já apresentadas (A.6) (A.7). Falta apenas detalhar $\frac{\partial h_u^c}{\partial \Theta^n}$.

$$\frac{\partial h_u^c}{\partial \Theta^n} = \begin{bmatrix} \frac{\partial h_{u_x}^c}{\partial \phi^n} & \frac{\partial h_{u_x}^c}{\partial \theta^n} & \frac{\partial h_{u_x}^c}{\partial \psi^n} \\ \frac{\partial h_{u_y}^c}{\partial \phi^n} & \frac{\partial h_{u_y}^c}{\partial \theta^n} & \frac{\partial h_{u_y}^c}{\partial \psi^n} \\ \frac{\partial h_{u_z}^c}{\partial \phi^n} & \frac{\partial h_{u_z}^c}{\partial \theta^n} & \frac{\partial h_{u_z}^c}{\partial \psi^n} \end{bmatrix} \quad (\text{A.10})$$

- Onde:

$$\frac{\partial h_{u_x}^c}{\partial \phi^n} = 0 \quad (\text{A.11a})$$

$$\frac{\partial h_{u_x}^c}{\partial \theta^n} = \Upsilon_x c(\theta) + s(\theta) c(\psi) \Upsilon_y - \Upsilon_z s(\theta) s(\psi) \quad (\text{A.11b})$$

$$\frac{\partial h_{u_x}^c}{\partial \psi^n} = \Upsilon_y c(\theta) s(\psi) + \Upsilon_z c(\theta) c(\psi) \quad (\text{A.11c})$$

$$\frac{\partial h_{u_y}^c}{\partial \phi^n} = -\Upsilon_x (c(\theta) c(\phi) - \Upsilon_y (s(\phi) s(\psi) - c(\psi) s(\theta) c(\psi))) + \Upsilon_z (-s(\phi) c(\psi) + c(\phi) s(\theta) s(\psi)) \quad (\text{A.11d})$$

$$\frac{\partial h_{u_y}^c}{\partial \theta^n} = \Upsilon_x s(\phi) s(\theta) - \Upsilon_y s(\phi) c(\theta) c(\psi) + \Upsilon_z s(\phi) c(\theta) s(\psi) \quad (\text{A.11e})$$

$$\frac{\partial h_{u_y}^c}{\partial \psi^n} = \Upsilon_y (c(\phi) c(\psi) + s(\phi) s(\theta) s(\psi)) + \Upsilon_z (-c(\phi) s(\psi) + s(\phi) s(\theta) c(\psi)) \quad (\text{A.11f})$$

$$\frac{\partial h_{u_z}^c}{\partial \phi^n} = \Upsilon_x s(\phi) c(\theta) - \Upsilon_y (c(\phi) s(\psi) - s(\phi) s(\theta) c(\psi)) - \Upsilon_z (c(\phi) c(\psi) - s(\phi) s(\theta) s(\psi)) \quad (\text{A.11g})$$

$$\frac{\partial h_{u_z}^c}{\partial \theta^n} = \Upsilon_x c(\phi) s(\theta) - \Upsilon_y c(\phi) c(\theta) c(\psi) + \Upsilon_z c(\phi) c(\theta) s(\psi) \quad (\text{A.11h})$$

$$\frac{\partial h_{u_z}^c}{\partial \psi^n} = -\Upsilon_y (s(\phi) c(\psi) - c(\phi) s(\theta) s(\psi)) + \Upsilon_z (s(\phi) s(\psi) + c(\phi) s(\theta) c(\psi)) \quad (\text{A.11i})$$

Nas expressões anteriores s e c representam respectivamente as funções trigonométricas *seno* e *co-seno* e:

$$\Upsilon_x = \rho_i (X_i - x^n) - \cos(\phi_i) \text{sen}(\theta_i) \quad (\text{A.12a})$$

$$\Upsilon_y = \rho_i (Y_i - y^n) \text{sen}(\phi_i) \quad (\text{A.12b})$$

$$\Upsilon_z = \rho_i (Z_i - z^n) + \cos(\phi_i) \text{cos}(\theta_i) \quad (\text{A.12c})$$

A.1.2 Jacobianos do modelo de observação em ordem aos estados das *landmarks*

O modelo de observação encontra-se relacionado apenas com os estados da respectiva *landmark* observada, por isso é nulo o jacobiano do modelo de observação em ordem aos estados das restantes *landmarks* no mapa.

$$\frac{\partial h_i}{\partial x_{map}} = \begin{pmatrix} 0 & \dots & 0 & \frac{\partial h_i}{\partial y_i} & 0 & \dots & 0 \end{pmatrix} \quad (\text{A.13})$$

A regra da cadeia permite decompor o jacobiano $\frac{\partial h_i}{\partial y_i}$ em três parcelas.

$$\frac{\partial h_i}{\partial y_i} = \frac{\partial h_i}{\partial h_d^c} \frac{\partial h_d^c}{\partial h_u^c} \frac{\partial h_u^c}{\partial y_i} \quad (\text{A.14})$$

Voltam a repetir-se os jacobianos $\frac{\partial h_i}{\partial h_d^c}$ (A.6) e $\frac{\partial h_d^c}{\partial h_u^c}$ (A.7) previamente apresentados. Quanto ao jacobiano $\frac{\partial h_u^c}{\partial y_i}$ é obtido pela fórmula seguinte:

$$\frac{\partial h_u^c}{\partial y_i} = \left(\rho_i C_b^c C_n^b, C_b^c C_n^b \frac{\partial m}{\partial \theta_i}, C_b^c C_n^b \frac{\partial m}{\partial \phi_i}, C_b^c C_n^b \left(\begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} - r^{WC} \right) \right) \quad (\text{A.15})$$

- Onde $m = [-\cos\phi_i \sin\theta_i \ \sin\phi_i \ \cos\phi_i \cos\theta_i]^T$ corresponde ao vector definido pelos ângulos de azimute e elevação $(\theta_i \ \phi_i)$. As derivadas parciais de m em ordem a θ_i e ϕ_i são as seguintes.

$$\frac{\partial m}{\partial \theta_i} = \begin{bmatrix} -\cos\phi_i \cos\theta_i \\ 0 \\ -\cos\phi_i \sin\theta_i \end{bmatrix} \quad (\text{A.16})$$

$$\frac{\partial m}{\partial \phi_i} = \begin{bmatrix} \sin\phi_i \sin\theta_i \\ \cos\phi_i \\ -\sin\phi_i \cos\theta_i \end{bmatrix} \quad (\text{A.17})$$

A.2 Jacobiano Utilizado na Conversão *Inverse-Depth*/Cartesianas

A conversão de uma *landmark* da representação em *inverse depth* para a representação em coordenadas cartesianas implica a actualização da matriz de covariância, pela sua multiplicação com a seguinte matriz jacobiana:

$$J = \text{diag} \left(I, \frac{\partial x_i}{\partial y_i}, I \right) \quad (\text{A.18})$$

As derivadas parciais da representação cartesiana em ordem à representação em *inverse depth* são as seguintes:

$$\frac{\partial x_i}{\partial y_i}_{[3 \times 6]} = \left[I_{[3 \times 3]} \ \frac{1}{\rho_i} \begin{pmatrix} s(\phi) s(\theta) \\ c(\phi) \\ -s(\phi) c(\theta) \end{pmatrix} \ \frac{1}{\rho_i} \begin{pmatrix} -c(\phi) c(\theta) \\ 0 \\ -c(\phi) s(\theta) \end{pmatrix} \ -\frac{1}{\rho_i^2} \begin{pmatrix} -c(\phi) s(\theta) \\ s(\phi) \\ c(\phi) c(\theta) \end{pmatrix} \right] \quad (\text{A.19})$$

A.3 Jacobiano para Cálculo da Covariância Inicial de uma *Landmark*

No instante de inicialização de uma *landmark*, o vector de estados e a matriz de covariância são expandidos, de modo a incluir no mapa as distribuições de probabilidade de um novo elemento. Como a observação de uma *landmark* se relaciona com a localização da câmara, no instante de inicialização, os estados da *landmark* encontram-se correlacionados com os estados associados à localização da câmara, no entanto não se verifica uma relação com o estado das *landmarks* já presentes no mapa. Assumindo que os estados das novas *landmarks* são introduzidos na base do vector de estados, a matriz de covariância é obtida da seguinte forma:

$$P(k|k)_{nova} = J \begin{bmatrix} P(k|k) & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & \sigma_{\rho_0}^2 \end{bmatrix} J^T \quad (\text{A.20})$$

- Onde R se relaciona com o ruído na imagem associado à detecção da *feature*, $\sigma_{\rho_0}^2$ corresponde à variância inicial do valor de profundidade inversa. J representa o jacobiano dos estados da *landmark* em ordem aos restantes estados do mapa (fórmula A.21).

$$J = \begin{bmatrix} & & & & 0 \\ & & & & \vdots \\ & I & & & 0 \\ & & & & \frac{\partial y}{\partial x_{cam}} \quad 0 \quad \dots \quad 0 \quad \frac{\partial y}{\partial h} \end{bmatrix} \quad (\text{A.21})$$

O processo de projecção inversa de *features* é independente da velocidade da câmara, por conseguinte as derivadas parciais em ordem a estes estados são nulas.

$$\frac{\partial y}{\partial x_{cam}} = \left(\frac{\partial y}{\partial r^n}, \frac{\partial y}{\partial \Theta^n}, \underbrace{\frac{\partial y}{\partial v^n}}_0 \right) \quad (\text{A.22})$$

O jacobiano em relação à posição da câmara é:

$$\frac{\partial y}{\partial r^n} = \begin{bmatrix} I_{[3 \times 3]} \\ 0_{[3 \times 3]} \end{bmatrix} \quad (\text{A.23})$$

Passemos às derivadas parciais associadas à orientação da câmara, que ficam:

$$\frac{\partial y}{\partial \Theta^n} = \begin{bmatrix} 0_{[3 \times 3]} \\ \frac{\partial \theta_i}{\partial \Theta^n}_{[1 \times 3]} \\ \frac{\partial \phi_i}{\partial \Theta^n}_{[1 \times 3]} \\ 0_{[1 \times 3]} \end{bmatrix} \quad (\text{A.24})$$

Ambos os jacobianos $\frac{\partial \theta_i}{\partial \Theta^n}$ e $\frac{\partial \phi_i}{\partial \Theta^n}$ são decompostos em duas parcelas (fórmulas A.25 e A.26).

$$\frac{\partial \theta_i}{\partial \Theta^n} = \frac{\partial \theta_i}{\partial h^n} \frac{\partial h^n}{\partial \Theta^n} \quad (\text{A.25})$$

$$\frac{\partial \phi_i}{\partial \Theta^n} = \frac{\partial \phi_i}{\partial h^n} \frac{\partial h^n}{\partial \Theta^n} \quad (\text{A.26})$$

- Onde:

$$\frac{\partial \theta_i}{\partial h^n} = \begin{bmatrix} \frac{-h_z^n}{h_z^{n2} - h_x^{n2}} & 0 & \frac{h_x^n}{h_z^{n2} - h_x^{n2}} \end{bmatrix} \quad (\text{A.27})$$

$$\frac{\partial \phi_i}{\partial h^n} = \begin{bmatrix} \frac{-h_x^n h_y^n}{(h_x^{n2} + h_y^{n2} + h_z^{n2}) \sqrt{h_x^{n2} + h_z^{n2}}} \\ \frac{\sqrt{h_x^{n2} + h_z^{n2}}}{h_x^{n2} + h_y^{n2} + h_z^{n2}} \\ \frac{-h_z^n h_y^n}{(h_x^{n2} + h_y^{n2} + h_z^{n2}) \sqrt{h_x^{n2} + h_z^{n2}}} \end{bmatrix}^T \quad (\text{A.28})$$

A matriz $\frac{\partial h^n}{\partial \Theta^n}$ tem dimensão 3×3 e cada elemento é obtido do seguinte modo:

$$\frac{\partial h^n}{\partial \Theta^n}(1, 1) = -h_{u_x}^c (s(\phi)s(\psi) + c(\phi)s(\theta)c(\psi)) + h_{u_y}^c (c(\phi)s(\psi) - s(\phi)s(\theta)c(\psi)) \quad (\text{A.29a})$$

$$\frac{\partial h^n}{\partial \Theta^n}(1, 2) = s(\theta)c(\psi) - h_{u_x}^c s(\phi)c(\theta)c(\psi) + h_{u_y}^c c(\psi)c(\theta)c(\psi) \quad (\text{A.29b})$$

$$\frac{\partial h^n}{\partial \Theta^n}(1, 3) = c(\theta)s(\psi) + h_{u_x}^c (c(\phi)c(\psi) + s(\phi)s(\theta)s(\psi)) + h_{u_y}^c (s(\phi)c(\psi) - c(\phi)s(\theta)s(\psi)) \quad (\text{A.29c})$$

$$\frac{\partial h^n}{\partial \Theta^n}(2, 1) = h_{u_x}^c (s(\phi)c(\psi) - c(\phi)s(\theta)s(\psi)) - h_{u_y}^c (c(\phi)c(\psi) + s(\phi)s(\theta)s(\psi)) \quad (\text{A.29d})$$

$$\frac{\partial h^n}{\partial \Theta^n}(2, 2) = s(\theta)s(\psi) - h_{u_x}^c s(\phi)c(\theta)s(\psi) + h_{u_y}^c c(\phi)c(\theta)s(\psi) \quad (\text{A.29e})$$

$$\frac{\partial h^n}{\partial \Theta^n}(2, 3) = -c(\theta)c(\psi) + h_{u_x}^c (c(\theta)s(\psi) - s(\phi)s(\theta)c(\psi)) + h_{u_y}^c (s(\phi)s(\psi) + c(\phi)s(\theta)c(\psi)) \quad (\text{A.29f})$$

$$\frac{\partial h^n}{\partial \Theta^n}(3, 1) = -h_{u_x}^c c(\phi)c(\theta) - h_{u_y}^c s(\phi)c(\theta) \quad (\text{A.29g})$$

$$\frac{\partial h^n}{\partial \Theta^n}(3, 2) = c(\theta) + h_{u_x}^c s(\phi)s(\theta) - h_{u_y}^c c(\phi)s(\theta) \quad (\text{A.29h})$$

$$\frac{\partial h^n}{\partial \Theta^n}(3, 3) = 0 \quad (\text{A.29i})$$

- Onde s e c indicam respectivamente as funções trigonométricas *seno* e *coseno* e $\Theta^n = [\phi \ \theta \ \psi]^T$.

Restam ainda as derivadas parciais em ordem ao inverso do modelo de observação.

$$\frac{\partial y}{\partial h} = \begin{bmatrix} \left(\frac{\partial y'}{\partial h} \right)_{[5 \times 2]} & 0_{[5 \times 1]} \\ 0_{[1 \times 2]} & 1 \end{bmatrix} \quad (\text{A.30})$$

- Onde $y' = (x_i, y_i, z_i, \theta_i, \phi_i)$ indica os estados da *landmark* excepto a profundidade inversa ρ_0 . As suas derivadas parciais podem ser decompostas do seguinte modo:

$$\frac{\partial y'}{\partial h} = \frac{\partial y'}{\partial h^n} \frac{\partial h^n}{\partial h_u^c} \frac{\partial h_u^c}{\partial h_d^c} \frac{\partial h_d^c}{\partial h} \quad (\text{A.31})$$

Sendo:

$$\frac{\partial y'}{\partial h^n} = \begin{bmatrix} 0_{[3 \times 3]} \\ \frac{\partial \theta_i}{\partial h^n} \\ \frac{\partial \phi_i}{\partial h^n} \end{bmatrix} \quad (\text{A.32})$$

As derivadas parciais $\frac{\partial \theta_i}{\partial h^n}$ e $\frac{\partial \phi_i}{\partial h^n}$ encontram-se em (A.27) e (A.28) respectivamente.

Por último:

$$\frac{\partial h^c}{\partial h_u^c} = C_b^m C_c^b \quad (\text{A.33})$$

$$\frac{\partial h_d^c}{\partial h} = \begin{bmatrix} 1 & 0 \\ \frac{1}{f_x k_u} & \\ 0 & \frac{1}{f_y k_v} \end{bmatrix} \quad (\text{A.34})$$

A.4 Jacobianos da Mecanização

Os jacobianos das equações da mecanização em ordem aos estados e às fontes de erro são necessários no cálculo da matriz de covariância, na fase de previsão do EKF. Antes de apresentar o cálculo dos jacobianos, recordemos as equações da mecanização, bem como as matrizes C_b^n e E_b^n , sobre as quais irão recair as operações de derivação:

$$\begin{bmatrix} r^n(k|k-1) \\ \Theta^n(k|k-1) \\ v^n(k|k-1) \end{bmatrix} = \begin{bmatrix} r^n(k-1|k-1) + v^n(k|k-1)\Delta t \\ \Theta^n(k-1|k-1) + E_b^n w^b(k)\Delta t \\ v^n(k-1|k-1) + (C_b^n a^b(k) + g^n)\Delta t \end{bmatrix} \quad (\text{A.35})$$

$$C_b^n = \begin{bmatrix} c(\psi)c(\theta) & c(\psi)s(\theta)s(\phi) - s(\psi)c(\phi) & c(\psi)s(\theta)c(\phi) + s(\psi)s(\phi) \\ s(\psi)c(\theta) & s(\psi)s(\theta)s(\phi) + c(\psi)c(\phi) & s(\psi)s(\theta)c(\phi) - c(\psi)s(\phi) \\ -s(\theta) & c(\theta)s(\phi) & c(\theta)c(\phi) \end{bmatrix} \quad (\text{A.36})$$

$$E_b^n = \begin{bmatrix} 1 & \text{sen}(\phi)\text{tan}(\theta) & \text{cos}(\phi)\text{tan}(\theta) \\ 0 & \text{cos}(\phi) & -\text{sen}(\phi) \\ 0 & \text{sen}(\phi)\text{sec}(\theta) & \text{cos}(\phi)\text{sec}(\theta) \end{bmatrix} \quad (\text{A.37})$$

A.4.1 Cálculo da matriz F

A matriz F consiste no jacobiano da mecanização em ordem aos estados do processo:

$$F = \begin{bmatrix} \frac{\partial r^n}{\partial r^n} [3 \times 3] & \frac{\partial r^n}{\partial \Theta^n} [3 \times 3] & \frac{\partial r^n}{\partial v^n} [3 \times 3] \\ \frac{\partial \Theta^n}{\partial r^n} [3 \times 3] & \frac{\partial \Theta^n}{\partial \Theta^n} [3 \times 3] & \frac{\partial \Theta^n}{\partial v^n} [3 \times 3] \\ \frac{\partial v^n}{\partial r^n} [3 \times 3] & \frac{\partial v^n}{\partial \Theta^n} [3 \times 3] & \frac{\partial v^n}{\partial v^n} [3 \times 3] \end{bmatrix} \quad (\text{A.38})$$

Muitas das derivadas parciais mencionadas em cima são nulas, pois a equação de base não depende dos estados relativamente aos quais se executa a derivada. As derivadas a calcular são as seguintes:

$$\frac{\partial r^n}{\partial r^n}, \frac{\partial r^n}{\partial v^n}, \frac{\partial \Theta^n}{\partial \Theta^n}, \frac{\partial v^n}{\partial \Theta^n}, \frac{\partial v^n}{\partial v^n}$$

Duas destas derivadas parciais resultam em matrizes identidade, nomeadamente :

$$\frac{\partial r^n}{\partial r^n} = \frac{\partial v^n}{\partial v^n} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{A.39})$$

O jacobiano da posição em ordem à velocidade consiste simplesmente na matrix identidade multiplicada pelo intervalo de tempo.

$$\frac{\partial r^n}{\partial v^n} = \begin{bmatrix} \Delta t & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \Delta t \end{bmatrix} \quad (\text{A.40})$$

As derivadas parciais da orientação em ordem a si própria ficam:

$$\frac{\partial \Theta^n}{\partial \Theta^n} = \begin{bmatrix} (\cos(\phi)tg(\theta)w_y^b - \text{sen}(\phi)tg(\theta)w_z^b)\Delta t + 1 & (\text{sen}(\phi)\text{sec}^2(\theta)w_y^b - \text{sen}(\phi)\text{sec}^2(\theta)w_z^b)\Delta t & 0 \\ -(\text{sen}(\phi)w_y^b + \cos(\phi)w_z^b)\Delta t & 1 & 0 \\ (\cos(\phi)\text{sec}(\theta)w_y^b - \text{sen}(\phi)\text{sec}(\theta)w_z^b)\Delta t & (\text{sen}(\phi)\text{sec}(\theta)tg(\theta)w_y^b + \cos(\phi)\text{sec}(\theta)tg(\theta)w_z^b)\Delta t & 1 \end{bmatrix} \quad (\text{A.41})$$

O jacobiano da velocidade em ordem à orientação obtém-se do seguinte modo:

$$\frac{\partial v^n}{\partial \Theta^n} = \begin{bmatrix} \frac{\partial v_x^n}{\partial \phi} [3 \times 3] & \frac{\partial v_x^n}{\partial \theta} [3 \times 3] & \frac{\partial v_x^n}{\partial \psi} [3 \times 3] \\ \frac{\partial v_y^n}{\partial \phi} [3 \times 3] & \frac{\partial v_y^n}{\partial \theta} [3 \times 3] & \frac{\partial v_y^n}{\partial \psi} [3 \times 3] \\ \frac{\partial v_z^n}{\partial \phi} [3 \times 3] & \frac{\partial v_z^n}{\partial \theta} [3 \times 3] & \frac{\partial v_z^n}{\partial \psi} [3 \times 3] \end{bmatrix} \quad (\text{A.42})$$

- Onde:

$$\frac{\partial v_x^n}{\partial \phi} = \left((s\phi s\psi + c\phi s\theta c\psi)a_y^b + (c\phi s\psi - s\phi s\theta c\psi)a_z^b \right) \Delta t \quad (\text{A.43a})$$

$$\frac{\partial v_x^n}{\partial \theta} = \left(-s\theta c\psi a_x^b + s\phi c\theta c\psi a_y^b + c\phi c\theta c\psi a_z^b \right) \Delta t \quad (\text{A.43b})$$

$$\frac{\partial v_x^n}{\partial \psi} = \left(-c\theta s\psi a_x^b - (c\phi c\psi + s\phi s\theta s\psi)a_y^b + (s\phi c\psi - c\phi s\theta s\psi)a_z^b \right) \Delta t \quad (\text{A.43c})$$

$$\frac{\partial v_y^n}{\partial \phi} = \left((-s\phi s\psi + c\phi s\theta s\psi)a_x^b - (c\phi c\psi - s\phi s\theta s\psi)a_z^b \right) \Delta t \quad (\text{A.43d})$$

$$\frac{\partial v_y^n}{\partial \theta} = \left(-s\theta s\psi a_x^b + c\phi s\theta s\psi a_y^b + c\phi c\theta s\psi a_z^b \right) \Delta t \quad (\text{A.43e})$$

$$\frac{\partial v_y^n}{\partial \psi} = \left(c\theta c\psi a_x^b + (-c\phi s\psi + s\phi s\theta c\psi)a_y^b + (s\phi s\psi - c\phi s\theta c\psi)a_z^b \right) \Delta t \quad (\text{A.43f})$$

$$\frac{\partial v_z^n}{\partial \phi} = c\phi s\theta a_y^b \Delta t - s\phi c\theta a_z^b \Delta t \quad (\text{A.43g})$$

$$\frac{\partial v_z^n}{\partial \theta} = -c\theta a_x^b \Delta t - s\phi s\theta a_y^b \Delta t - c\phi s\theta a_z^b \Delta t \quad (\text{A.43h})$$

$$\frac{\partial v_z^n}{\partial \psi} = 0 \quad (\text{A.43i})$$

A.4.2 Cálculo da matriz G

Considerando as fontes de erro $a^b(k+1)$ e $w^b(k+1)$, a matriz jacobiana dos estados em ordem às fontes de erro (matriz G) é constituída de acordo com a fórmula (A.44)

$$G = \begin{bmatrix} \frac{\partial r^n}{\partial a^b} & \frac{\partial r^n}{\partial w^b} \\ \frac{\partial \Theta^n}{\partial a^b} & \frac{\partial \Theta^n}{\partial w^b} \\ \frac{\partial v^n}{\partial a^b} & \frac{\partial v^n}{\partial w^b} \end{bmatrix} \quad (\text{A.44})$$

As derivadas parciais não nulas são $\frac{\partial \Theta^n}{\partial w^b}$ e $\frac{\partial v^n}{\partial a^b}$.

$$\frac{\partial \Theta^n}{\partial w^b} = \begin{bmatrix} \Delta t & \text{sen}(\phi) \text{tg}(\theta) \Delta t & \text{cos}(\phi) \text{tg}(\theta) \Delta t \\ 0 & \text{cos}(\phi) \Delta t & -\text{sen}(\phi) \Delta t \\ 0 & \text{sen}(\phi) \text{sec}(\theta) \Delta t & \text{cos}(\phi) \text{sec}(\theta) \Delta t \end{bmatrix} \quad (\text{A.45})$$

$$\frac{\partial \Theta^n}{\partial w^b} = \begin{bmatrix} c(\theta)c(\psi)\Delta t & (-c(\phi)s(\psi) + s(\phi)s(\theta)c(\psi))\Delta t & (s(\phi)s(\psi) + c(\phi)s(\theta)c(\psi))\Delta t \\ c(\theta)s(\psi)\Delta t & (c(\phi)c(\psi) + s(\phi)s(\theta)s(\psi))\Delta t & (-s(\phi)c(\psi) + c(\phi)s(\theta)s(\psi))\Delta t \\ -s(\theta)\Delta t & s(\phi)c(\theta)\Delta t & c(\phi)c(\theta)\Delta t \end{bmatrix} \quad (\text{A.46})$$

Anexo B

Representações de orientação

Existem várias representações matemáticas de rotações e orientação, sendo as mais comuns os Ângulos de Euler, quaterniões e matrizes de rotação.

B.1 Ângulos de Euler

Os Ângulos de Euler representam uma rotação tridimensional através de três parâmetros, que definem a magnitude das rotações sucessivas a efectuar segundo cada eixo de um sistema referencial ortogonal – o ângulo *roll* especifica a quantidade de rotação segundo o eixo x, o ângulo *pitch* sobre o eixo y e o ângulo *yaw* sobre o eixo z.

$$E = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} roll \\ pitch \\ yaw \end{bmatrix} \quad (\text{B.1})$$

Esta representação é a mais compacta das mencionadas anteriormente, no entanto está sujeita a singularidades, que ocorrem quando a rotação intermédia alinha os eixos da primeira e terceira rotações. Para sequências em que a primeira e terceira rotações são iguais – as chamadas Sequências Repetidas de Euler – a singularidade acontece caso a segunda rotação seja de 0° ou 180° . Para sequências não repetidas as singularidades passam a manifestar-se quando a segunda rotação é de $\pm 90^\circ$.

B.2 Quaterniões

Os quaterniões são números complexos constituídos por um elemento real e três componentes imaginárias. No contexto da representação de rotações no espaço tridimensional são utilizados quaterniões unitários, que podem ser encarados de forma diferente. A ideia por detrás dos quaterniões unitários é de que qualquer rotação no espaço tridimensional pode ser atingida pela rotação em torno de um vector (figura B.1).

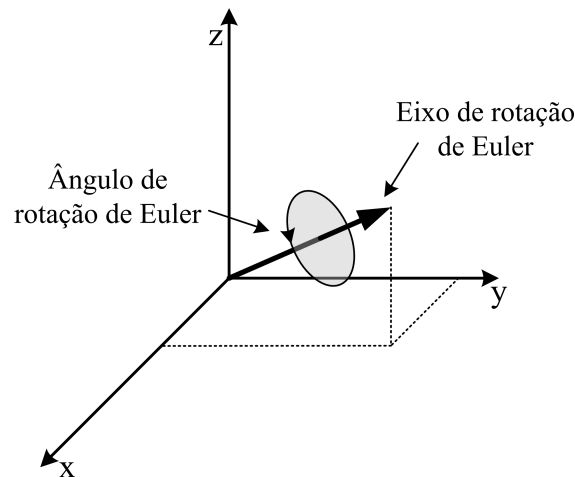


Figura B.1: Ilustração de uma rotação operada através de um quaternião.

O primeiro elemento de um quaternião unitário encontra-se associado à amplitude da rotação (ângulo de rotação de Euler), ao passo que os restantes se relacionam com o vector tridimensional em torno do qual se processa a rotação (eixo de rotação de Euler).

$$q = \begin{bmatrix} q_0 \\ q_1 \mathbf{i} \\ q_2 \mathbf{j} \\ q_3 \mathbf{k} \end{bmatrix} = \begin{bmatrix} \cos(\alpha/2) \\ e_x \text{sen}(\alpha/2) \\ e_y \text{sen}(\alpha/2) \\ e_z \text{sen}(\alpha/2) \end{bmatrix} \quad (\text{B.2})$$

- Onde α indica o ângulo de rotação em radianos e $E = [e_x \ e_y \ e_z]^T$ representa as componentes que especificam o eixo de rotação de Euler.

B.2.1 Operações sobre quaterniões

Considere-se o quaternião q :

$$q = \begin{bmatrix} q_0 \\ q_1 \mathbf{i} \\ q_2 \mathbf{j} \\ q_3 \mathbf{k} \end{bmatrix} \quad (\text{B.3})$$

O conjugado de um quaternião consiste no simétrico das suas componentes imaginárias:

$$\bar{q} = \begin{bmatrix} q_0 \\ -q_1 \mathbf{i} \\ -q_2 \mathbf{j} \\ -q_3 \mathbf{k} \end{bmatrix} \quad (\text{B.4})$$

A adição e subtracção de quaterniões são conseguidas respectivamente pela adição e subtracção directa das componentes correspondentes.

Do produto de quaterniões, indicado pelo símbolo \circ , resulta um terceiro quaternião. O produto entre as diversas componentes dos quaterniões é regido pelas seguintes propriedades:

$$\begin{aligned} \mathbf{i} \circ \mathbf{i} &= -1, & \mathbf{i} \circ \mathbf{j} &= \mathbf{k}, & \mathbf{i} \circ \mathbf{k} &= -\mathbf{j} \\ \mathbf{j} \circ \mathbf{j} &= -1, & \mathbf{j} \circ \mathbf{k} &= \mathbf{i}, & \mathbf{j} \circ \mathbf{i} &= -\mathbf{k} \\ \mathbf{k} \circ \mathbf{k} &= -1, & \mathbf{k} \circ \mathbf{i} &= \mathbf{j}, & \mathbf{k} \circ \mathbf{j} &= -\mathbf{i} \end{aligned}$$

O produto do quaternião q pelo quaternião $g = [g_0 \ g_1 \mathbf{i} \ g_2 \mathbf{j} \ g_3 \mathbf{k}]^T$, de acordo com a propriedade distributiva e as propriedades anteriores, fica:

$$\begin{aligned} q \circ g &= (q_0 g_0 - q_1 g_1 - q_2 g_2 - q_3 g_3) + (q_0 g_1 + q_1 g_0 + q_2 g_3 - q_3 g_2) \mathbf{i} + \\ &+ (q_0 g_2 - q_1 g_3 + q_2 g_0 + q_3 g_1) \mathbf{j} + (q_0 g_3 + q_1 g_2 - q_2 g_1 + q_3 g_0) \mathbf{k} \end{aligned} \quad (\text{B.5})$$

O produto expresso sob a forma matricial respeita a fórmula (B.17).

$$q \circ g = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \end{bmatrix} = \begin{bmatrix} g_0 & -g_1 & -g_2 & -g_3 \\ g_1 & g_0 & g_3 & -g_2 \\ g_2 & -g_3 & g_0 & g_1 \\ g_3 & g_2 & -g_1 & g_0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (\text{B.6})$$

É importante ter em conta que o produto de quaterniões não é cumulativo $q \circ g \neq g \circ q$, embora preserve a propriedade associativa $q \circ (g \circ b) = (q \circ g) \circ b$

A norma de um quaternião corresponde ao produto entre este e o seu próprio conjugado, como se mostra na fórmula (B.7).

$$\|q\| = q \circ \bar{q} = q_0^2 + q_1^2 + q_2^2 + q_3^2 \quad (\text{B.7})$$

A inversa de um quaternião é obtida pela divisão das componentes do conjugado pela norma, fórmula (B.8).

$$q^{-1} = \frac{\bar{q}}{\|q\|} \quad (\text{B.8})$$

B.2.2 Representação de quaterniões sob a forma de vector

Um quaternião pode ser representado sob a forma de vector:

$$q = q_0 + \vec{q} \quad (\text{B.9})$$

- Onde $\vec{q} = [q_1, q_2, q_3]^T$

A forma vectorial permite uma representação mais compacta das operações com quaterniões, por exemplo o conjugado (fórmula B.10) e o produto (fórmula B.11)

$$\bar{q} = q_0 - \vec{q} \quad (\text{B.10})$$

$$q \circ g = q_0 g_0 - \vec{q} \cdot \vec{g} + q_0 \vec{g} + g_0 \vec{q} + \vec{q} \times \vec{g} \quad (\text{B.11})$$

Partindo do quaternião q podem formar-se as seguintes matrizes:

$$Q_q = \begin{bmatrix} q_0 & -\vec{q}^T \\ \vec{q} & (q_0 I + [\vec{q} \times]) \end{bmatrix}, \quad \bar{Q}_q = \begin{bmatrix} q_0 & -\vec{q}^T \\ \vec{q} & (q_0 I - [\vec{q} \times]) \end{bmatrix} \quad (\text{B.12})$$

Constata-se que as matrizes anteriores possuem as seguintes propriedades: $Q_{\bar{q}} = Q_q^T$ e $\bar{Q}_{\bar{q}} = \bar{Q}_q^T$. Utilizando as matrizes Q_q e \bar{Q}_q , o produto de quaterniões fica:

$$q \circ g = (Q_q)g = (\bar{Q}_g)q \quad (\text{B.13})$$

As matrizes Q_q e \bar{Q}_q possuem igual constituição às apresentadas anteriormente em (B.17).

B.2.3 Rotações através de quaterniões

Considerem-se dois sistemas referenciais R_a e R_b cujo alinhamento entre ambos é conseguido pela rotação de ζ radianos em torno do vector unitário \mathbf{E} aplicada a R_a . O quaternião que produz a rotação de R_a para R_b é:

$$w = \begin{bmatrix} \cos(\zeta/2) \\ \mathbf{E} \sin(\zeta/2) \end{bmatrix} \quad (\text{B.14})$$

Seja $v_b = R_a^b v_a$, em que v_a define as coordenadas de um vector no referencial R_a enquanto v_b indica as coordenadas de v_a segundo o referencial R_b . Os vectores v_a e v_b podem ser expressos sob a forma de quaterniões do seguinte modo:

$$q_{v_a} = \begin{bmatrix} 0 \\ v_a \end{bmatrix}, \quad q_{v_b} = \begin{bmatrix} 0 \\ v_b \end{bmatrix} \quad (\text{B.15})$$

A transformação do vector v_a do referencial R_a para o referencial R_b utilizando quaterniões processa-se de acordo com a fórmula (B.16)

$$q_{v_b} = w \circ q_{v_a} \circ w^{-1} = w \circ q_{v_a} \circ \bar{w} \quad (\text{B.16})$$

A equação anterior pode ser reescrita da acordo com a relação da fórmula (B.13), ficando:

$$q_{v_b} = Q_w \bar{Q}_{\bar{w}} q_{v_a} = \begin{bmatrix} w_0 & -w_1 & -w_2 & -w_3 \\ w_1 & w_0 & -w_3 & w_2 \\ w_2 & w_3 & w_0 & -w_1 \\ w_3 & -w_2 & w_1 & w_0 \end{bmatrix} \begin{bmatrix} w_0 & w_1 & w_2 & w_3 \\ -w_1 & w_0 & -w_3 & w_2 \\ -w_2 & w_3 & w_0 & -w_1 \\ -w_3 & -w_2 & w_1 & w_0 \end{bmatrix} \begin{bmatrix} 0 \\ q_{v_a} \end{bmatrix} \quad (\text{B.17})$$

O resultado do produto matricial $Q_w \bar{Q}_{\bar{w}}$ é:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & w_0^2 + w_1^2 - w_2^2 - w_3^2 & 2(w_1w_2 - w_0w_3) & 2(w_0w_2 + w_1w_3) \\ 0 & 2(w_1w_2 + w_0w_3) & w_0^2 - w_1^2 + w_2^2 - w_3^2 & 2(-w_0w_1 + w_2w_3) \\ 0 & 2(-w_0w_2 + w_1w_3) & 2(w_0w_1 + w_2w_3) & w_0^2 - w_1^2 - w_2^2 + w_3^2 \end{bmatrix} \quad (\text{B.18})$$

A sub-matriz 3×3 do canto inferior direito corresponde à matriz rotação R_a^b . Generalizando, a construção da matriz rotação que roda vectores de um referencial para outro de acordo com o quaternião q segue a fórmula (B.19)

$$R(q) \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_0q_2 + q_1q_3) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(-q_0q_1 + q_2q_3) \\ 2(-q_0q_2 + q_1q_3) & 2(q_0q_1 + q_2q_3) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (\text{B.19})$$

B.2.4 Conversão de quaterniões noutras expressões de rotação

O quaternião equivalente a uma matriz rotação R_a^b pode ser construído pela fórmula (B.20).

$$q = \begin{bmatrix} \frac{1}{2} \sqrt{1 + R_a^b(1,1) + R_a^b(2,2) + R_a^b(3,3)} \\ \frac{1}{4b_0} (R_a^b(3,2) - R_a^b(2,3)) \\ \frac{1}{4b_0} (R_a^b(1,3) - R_a^b(3,1)) \\ \frac{1}{4b_0} (R_a^b(2,1) - R_a^b(1,2)) \end{bmatrix} \quad (\text{B.20})$$

A relação entre as componentes de um quaternião e os ângulos de Euler é:

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \cos(\phi/2)\cos(\theta/2)\cos(\psi/2) + \text{sen}(\phi/2)\text{sen}(\theta/2)\text{sen}(\psi/2) \\ \text{sen}(\phi/2)\cos(\theta/2)\cos(\psi/2) - \cos(\phi/2)\text{sen}(\theta/2)\text{sen}(\psi/2) \\ \cos(\phi/2)\text{sen}(\theta/2)\cos(\psi/2) + \text{sen}(\phi/2)\cos(\theta/2)\text{sen}(\psi/2) \\ \cos(\phi/2)\cos(\theta/2)\text{sen}(\psi/2) - \text{sen}(\phi/2)\text{sen}(\theta/2)\cos(\psi/2) \end{bmatrix} \quad (\text{B.21})$$

Inversamente, os ângulos de Euler podem ser derivados de um quaterni3o:

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \text{atan} \left(\frac{2(q_2q_3 + q_0q_1)}{1 - 2(q_1^2 + q_2^2)} \right) \\ \text{asen}(2(q_0q_2 - q_1q_3)) \\ \text{atan} \left(\frac{2(q_0q_3 + q_1q_2)}{1 - 2(q_2^2 + q_3^2)} \right) \end{bmatrix} \quad (\text{B.22})$$

- Esta rela33o s3o 3e v3lida quando $\theta \neq 90^\circ$.

B.3 Matrizes de rotação

O emprego de matrizes de rotação 3e bastante conveniente na aplica33o de rota33es entre sistemas referenciais. Existem tr3s matrizes de rotação distintas, cada qual associada 3a rota33o segundo um determinado eixo – em torno de x (B.23), em torno de y (B.24) e em torno de z (B.25).

$$C_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\text{sen}(\phi) \\ 0 & \text{sen}(\phi) & \cos(\phi) \end{bmatrix} \quad (\text{B.23})$$

$$C_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \text{sen}(\theta) \\ 0 & 1 & 0 \\ -\text{sen}(\theta) & 0 & \cos(\theta) \end{bmatrix} \quad (\text{B.24})$$

$$C_z(\psi) = \begin{bmatrix} \cos(\psi) & -\text{sen}(\psi) & 0 \\ \text{sen}(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{B.25})$$

Ao multiplicar as matrizes de rotação elementares atinge-se uma matriz que incorpora

a totalidade das operações de rotação, permitindo assim expressar uma rotação tridimensional através de uma única matriz.

$$C_{x,y,z}(\phi, \theta, \psi) = C_x(\phi)C_y(\theta)C_z(\psi) \quad (\text{B.26})$$

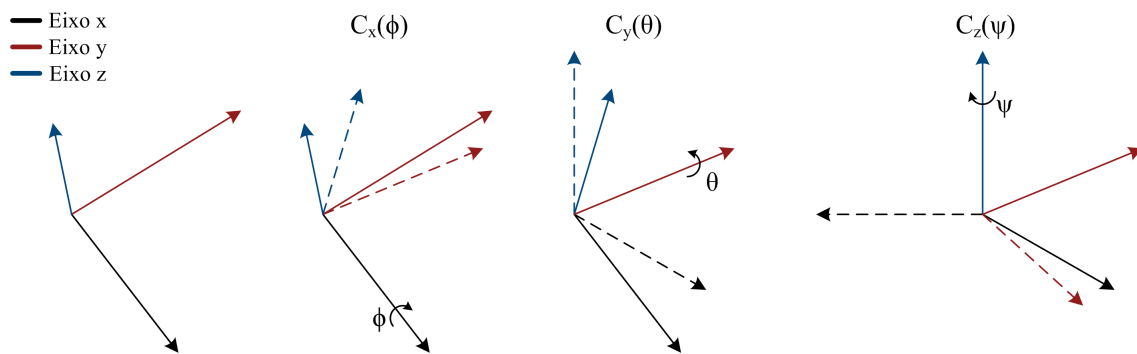


Figura B.2: Exemplo de uma sequência de rotações para alinhamento entre dois sistemas de coordenadas utilizando os Ângulos de Euler.

A rotação de um vetor tridimensional, utilizando uma matriz rotação, consiste na simples multiplicação da matriz pelo vetor expresso na forma de vector coluna.

$$v^a = C_b^a \times v^b \quad (\text{B.27})$$

Anexo C

Desvio Padrão da Trajectória de Referência

As seguintes figuras registam os valores do desvio padrão estimado nos estados de posição e atitude da trajectória de referência.

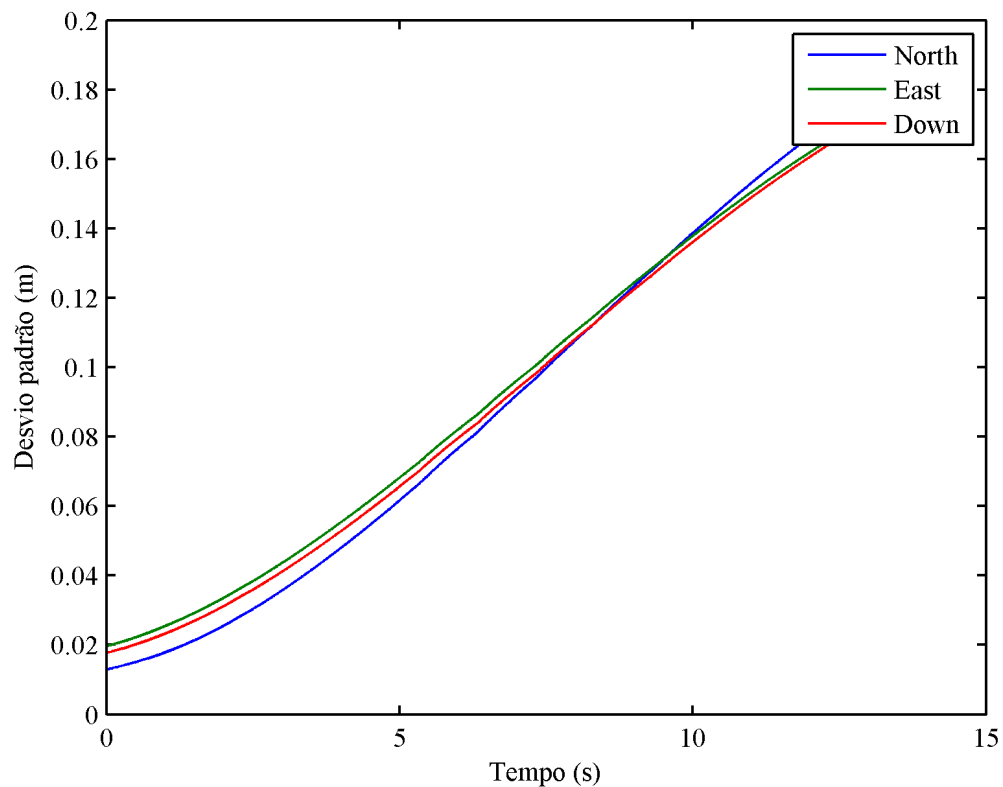


Figura C.1: Desvio padrão nos estados de posição.

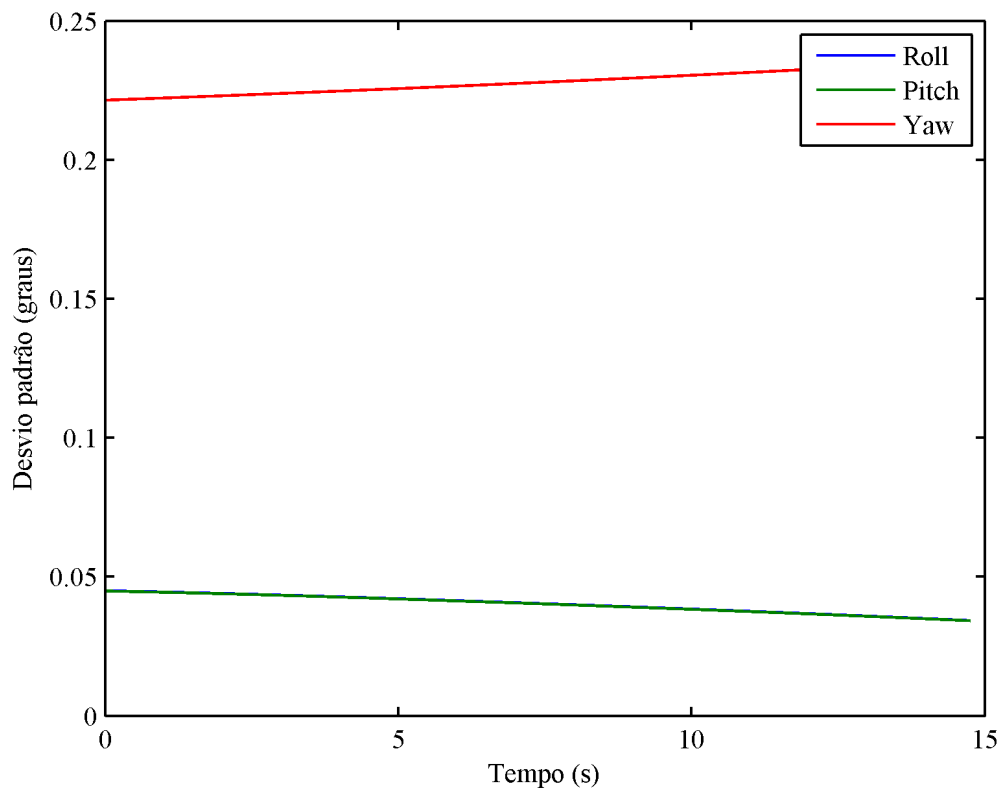


Figura C.2: Desvio padrão nos estados de orientação.