

Instituto Superior de Engenharia do Porto



Sistema de Visão para Percepção e Navegação de um UAV Quadrotor

Tiago Fernando Couto Sarmiento dos Santos

Tese submetida no âmbito do
Mestrado em Engenharia Electrotécnica e de Computadores
Ramo de Sistemas Autónomos

Orientador: José Miguel Soares de Almeida (Mestre)

Dezembro de 2009

Resumo

A inspecção visual, de infra-estruturas, de áreas de difícil acesso ou áreas em ambientes hostis à presença humana, assumem nos dias de hoje um carácter determinante, no processo de manutenção de infra-estruturas críticas, na monitorização de áreas de risco e/ou sinistradas, e, na vigilância de áreas de acesso restrito ou controlado. Para estas missões os veículos aéreos não tripulados (*UAV's*) na categoria dos *VTOL* (*Vertical Take-Off Landing*) e em especial quadrotors, apresentam-se como uma solução eficaz (ausência de riscos humanos e baixo custo sistemático) abrindo novas perspectivas para as aplicações de inspecção e monitorização.

Em missões de inspecção e monitorização remota nem sempre o *streaming* de vídeo em tempo real é a melhor representação da informação para o utilizador quando se pretende uma inspecção detalhada de uma zona ou estrutura. Consequentemente, será preferível a construção de uma representação única do cenário de inspecção, através da composição de um mosaico de imagens.

Nesta dissertação endereçamos os elementos necessários ao desenvolvimento de um sistema de inspecção visual em tempo real, capaz de servir como mapa de navegação para veículos autónomos aéreos, bem como em cenários de inspecção dotar o utilizador de uma melhor percepção das zonas e estruturas sob inspecção.

Foi efectuada uma análise dos projectos existentes de *UAV's* quadrotors mais relevantes com ênfase nas capacidades sensoriais e de percepção do ambiente que os rodeia, bem como das técnicas de mosaico para aplicações de inspecção e navegação em tempo real.

Neste trabalho são expostos conceitos necessários à compreensão da formação de um mosaico de imagens e os fundamentos de navegação que permitem compreender de que modo um sistema de visão pode apoiar outros sensores de navegação.

Foram implementadas técnicas de mosaico em *MathWorks Matlab* utilizando abordagens do estado da arte combinando detectores de pontos de interesse *SIFT* (*Scale-*

Invariant Feature Transform), com métodos de exclusão de falsas correspondências baseados em *RANSAC* (*Random Sample Consensus*), estimação de transformações recorrendo a técnicas de mínimos quadrados e com junção de imagens e pontos do mapa utilizando transformações *affine* e projectiva.

Tendo como objectivo a utilização das técnicas de mosaico em tempo real foi proposta e implementada uma variante da técnica anterior onde a informação de um sistema inercial é utilizada substituindo o método *RANSAC* na etapa de exclusão de falsas associações.

Ambos os métodos propostos foram validados experimentalmente apresentando resultados idênticos em termos de mosaico e informação para navegação, mas o novo método apresentou uma redução significativa em termos computacionais, o que traz uma mais valia na aplicação em tempo real.

O sistema desenvolvido permite obter mosaicos de imagens para inspecção e navegação. Para a criação de um mosaico de imagens é utilizado o algoritmo de detecção de pontos de interesse em imagens *SIFT*. Como os pontos de duas imagens nem sempre geram associações correctas existe uma fase de exclusão de correspondências. Nesta etapa são apresentadas duas soluções: uma com recurso a estimação robusta baseada no algoritmo *RANSAC* e a segunda com integração de informação do sistema inercial presente no veículo. Ambas permitem a determinação das associações correctas que servem de base para o cálculo dos parâmetros da transformação bidimensional (*affine* ou projectiva) que vai deformar a imagem, possibilitando assim o registo de imagens.

Foram estudados e implementados métodos para extracção da informação do movimento da câmara a partir das transformações estimadas da matriz de homografia entre imagens obtidas pelos métodos de mosaico implementados.

De forma a validar a abordagem e caracterizar os desempenhos dos algoritmos implementados foram feitos testes de criação de mosaicos de imagens, assim como da determinação da posição e orientação da câmara por forma a fornecer informação para a navegação.

Foi proposta uma arquitectura para o sistema de navegação e inspecção visual baseado num mosaico de imagens que integra estes resultados e que simultâneamente apoie a navegação (através da fusão com outros sensores de navegação como *GPS* e *INS*) e permita a inspecção e monitorização sob a forma de mosaico de imagens.

Palavras Chave: *UAV*, Quadrotor, Percepção, Navegação, Mosaico de Imagens, Inspeção, Visão Computacional, Robot

Esta página foi intencionalmente deixada em branco.

Abstract

A visual inspection, of infrastructure, in areas of difficult access or areas in hostile environments to human presence, assume today decisive importance in the process of maintaining critical infrastructure, monitoring of risk areas and/or affected, and the monitoring of restricted areas or controlled areas. For these missions, unmanned aerial vehicles (*UAV's*) in the category of *VTOL* (Vertical Take-Off Landing) and in particular quadrotors, presented as an effective solution (absence of human risk and systemic low-cost) opening new prospects for applications inspection and monitoring.

In the inspection missions and remote monitoring video streaming in real time is not always the best representation of the information to the user when we want a detailed inspection of an area or structure. Consequently, the construction of a single representation of the inspection scene by the composition of a mosaic of images is thus preferable.

In this dissertation we address the elements necessary to develop a visual inspection system in real time, capable to serve as navigation map for autonomous air vehicles, as well in inspection scenarios provide the user a better understanding of areas and structures under inspection.

At this work was made an analysis of existing projects of *UAV's* quadrotors more relevant with emphasis on skills and sensory perception of their environment, as well as the mosaic technique for inspection applications and real-time navigation.

This work expose concepts necessary for understanding the formation of a mosaic of images and the basics of navigation that allows to understand how a vision system can support other navigation sensors.

Have been implemented a mosaic technique using *MathWorks Matlab* using approaches to from state of the art that combine feature detector *SIFT* (*Scale-Invariant Feature Transform*) points of interest, with methods to exclude false matches based on *RANSAC*

(*Random Sample Consensus*), transformation estimation using least mean squares techniques and, image stitching and points of map using affine and projective transformations.

With the aim of using the techniques of mosaic in real time was proposed and implemented a variant of the previous technique where information from an inertial system is replacing the method used in the *RANSAC* step to exclude false associations.

Both proposed methods were validated experimentally by presenting identical results in terms of mosaic and information for navigation, but the new method showed a significant reduction in computing, which brings an added value in real time application.

The developed system allows get a mosaic of images for inspection and navigation. To create a mosaic of images is used to detect feature points in images *SIFT* algorithm. Since the points of two images do not always generate correct associations there is an exclusion phase of matches. In this step are presented two solutions: one using robust estimation algorithm based on *RANSAC* and the second with integration of information from inertial system present in this vehicle. Both allow the determination of correct associations that serve as the basis for calculating the parameters of two-dimensional transformation (affine or projective) that will warp the image, thus enabling the image registration.

Were studied and implemented methods for extraction of structure from motion of the camera using estimated homography matrix between images, obtained by the methods of mosaic of images implemented.

In order to validate the approach and characterize the performance of algorithms implemented tests were made to create a mosaic of images, as well as of determination of the positioning and orientation of the camera in order to provide information for navigation.

We proposed an architecture for the navigation system and visual inspection based on a mosaic of images that integrates these findings and to simultaneously support the navigation (through data fusion with other navigation sensors such as *GPS* and *INS*) and allow the inspection and monitoring in the form of mosaic of images.

Key Words: *UAV*, Quadrotor, Perception, Navigation, Mosaic Maps, Inspection, Computer Vision, Robot

Agradecimentos

Antes de mais é preciso agradecer aos meus pais que com esforço tornaram possível a concretização desta fase/etapa da minha vida.

Tenho de agradecer incondicionalmente aos meus amigos Guilherme Amaral, Ivo Costa, João Sousa, Miguel Magalhães, Pedro Pereira e Sérgio Coelho pela entreaajuda, apoio e momentos que passamos juntos durante a realização desta dissertação.

Como é de esperar agradeço também a todos os elementos do Laboratório de Sistemas Autónomos, pelo conhecimento transmitido e apoio na realização deste trabalho.

Fica também um agradecimento especial a todos os amigos e familiares que directa ou indirectamente ligados ao projecto sempre apoiaram, ajudaram, e sempre compreenderam a falta de tempo para estar com eles durante a elaboração deste projecto.

Muito Obrigado,
Tiago Fernando Couto Sarmiento dos Santos

Esta página foi intencionalmente deixada em branco.

“As respostas nem sempre foram fáceis mas acho que encontramos o caminho.”

Eddie Vedder

Esta página foi intencionalmente deixada em branco.

Conteúdo

1	Introdução	1
1.1	Âmbito da Dissertação	1
1.2	Enquadramento e Motivação	3
1.3	Objectivos	4
1.4	Organização da Dissertação	5
2	Estado da Arte	7
2.1	Projectos Relacionados de Quadrotors	8
2.1.1	<i>Starmac</i>	8
2.1.2	<i>Swarm</i>	8
2.1.3	<i>OS4</i>	9
2.1.4	<i>GRASP Laboratory</i>	10
2.1.5	<i>Team Swift</i>	10
2.1.6	<i>X4 - Flyer</i>	11
2.2	Técnicas de Navegação Aplicadas em <i>UAV's</i>	11
2.3	Sistemas de Visão em <i>UAV's</i>	12
3	Fundamentos de Visão	15
3.1	Câmara	16
3.1.1	Parâmetros Intrínsecos	17
3.1.2	Parâmetros Extrínsecos	19
3.1.2.1	Rotação 3D	20
3.2	Transformações	21
3.2.1	Transformações 2D	22
3.2.1.1	Transformação Euclidiana 2D	23
3.2.1.2	Transformação de Similaridade 2D	24
3.2.1.3	Transformação <i>Affine</i> 2D	25
3.2.1.4	Transformação Projectiva 2D	26
3.2.2	Transformações 3D	27
3.2.2.1	Transformação Euclideana 3D	28
3.2.2.2	Transformação de Similaridade 3D	28
3.2.2.3	Transformação <i>Affine</i> 3D	29
3.2.2.4	Transformação Projectiva 3D	29
3.3	Relação entre duas imagens	30
3.4	Geometria Epipolar	31
3.4.1	Matriz Fundamental	32
3.4.2	Decomposição da Matriz Essencial	32
3.5	Decomposição Matriz de Homografia	33

4	Técnica de Mosaico	37
4.1	<i>Scale-Invariant Feature Transform (SIFT)</i>	39
4.1.1	Detecção/Localização de Pontos Chave	40
4.1.2	Orientação dos Pontos Chave	40
4.1.3	Descritor do Ponto Chave	41
4.2	Correspondência de Pontos	43
4.3	Estimação Robusta em Visão Computacional	43
4.3.1	<i>RANSAC - Random Sample Consensus</i>	44
4.3.1.1	Construção do <i>MSS - Minimal Sample Set</i>	45
4.3.1.2	Número de Iterações	45
4.3.1.3	Cotação de um <i>CS - Consensus Set</i>	46
4.3.1.4	<i>Threshold</i>	47
4.3.1.5	<i>RANSAC</i> Aplicado a Homografia	48
4.3.2	Outros Algoritmos de Estimação Robusta	48
4.4	Junção de Imagens	49
4.4.1	Movimento Plano de Perspectiva	49
4.4.2	Panoramas	50
4.4.3	Coordenadas Esféricas e Cilíndricas	51
5	Fundamentos de Navegação	53
5.1	Técnicas de Navegação	54
5.1.1	Sistemas Baseados em Pontos de Referência	54
5.1.2	Sistemas “ <i>Dead Reckoning</i> ”	54
5.1.2.1	Sistema de Navegação Inercial - <i>INS</i>	55
5.2	Referenciais de Coordenadas	56
5.2.1	Referencial do Corpo	56
5.2.2	Referencial <i>NED</i>	57
5.3	Atitude do Veículo	57
5.3.1	Rotação em <i>Roll</i> (ϕ)	58
5.3.2	Rotação em <i>Pitch</i> (θ)	58
5.3.3	Rotação em <i>Yaw</i> (ψ)	58
5.4	Transformação de Referenciais	58
6	Implementação e Resultados	61
6.1	Arquitetura do Sistema de Navegação e Inspeção Visual	61
6.2	Mosaico de Imagens	63
6.2.1	Detecção e Caracterização de Pontos de Interesse	64
6.2.2	Correspondência de Pontos	67
6.2.3	Eliminação de Falsas Correspondências	67
6.2.3.1	Exclusão de Correspondências com <i>RANSAC</i>	68
6.2.3.2	Exclusão de Correspondências com <i>INS</i>	69
6.2.3.3	Comparação entre <i>RANSAC</i> e <i>INS</i>	72
6.2.4	Fusão de Imagens	73
6.2.4.1	Fusão de Imagens com Transformação <i>Affine</i>	74
6.2.4.2	Fusão de Imagens com Transformação Projectiva	76
6.3	Determinação da Posição e Orientação	78
6.3.1	Comparação Rotação <i>INS</i> com Rotação dos Parâmetros Extrínsecos	78
6.3.2	Decomposição da Matriz de Homografia	80

7 Conclusões e Trabalho Futuro

83

Referências

90

Esta página foi intencionalmente deixada em branco.

Lista de Figuras

2.1	<i>Starmac</i>	8
2.2	<i>Swarm</i>	9
2.3	<i>OS4</i>	10
2.4	<i>Team Swift</i>	11
2.5	<i>X4</i>	11
3.1	Modelo de uma câmara <i>pinhole</i>	16
3.2	Transformação de pixels para coordenadas normalizadas	18
3.3	Representação dos eixos XYZ numa câmara	20
3.4	Rotação em eixo Z	20
3.5	Rotação em eixo X	21
3.6	Rotação em eixo Y	21
3.7	Transformações 2D	24
3.8	Distorções de uma Transformação <i>Affine</i>	26
3.9	Relação existente entre duas imagens	30
3.10	Representação da geometria Epipolar	31
4.1	Dois pares de imagens para correspondência	38
4.2	Esquema de procura da localização de pontos chave	41
4.3	Exemplo gráfico da janela de interesse da imagem e representação de gradientes	42
4.4	Exemplo gráfico de um descritor	42
4.5	Problema da correspondência de Pontos	43
4.6	Representação ilustrativa do algoritmo <i>RANSAC</i>	45
4.7	Exemplo Algoritmo <i>RANSAC</i> aplicado à estimação de uma linha recta	45
4.8	Comparação de alguns algoritmos de estimação robusta	49
4.9	Exemplo do movimento de perspectiva plano	50
4.10	Exemplo de panorama para a criação de mapas para inspeção	51
5.1	Representação do referencial <i>NED</i>	57
6.1	Arquitectura sistema de visão e navegação e esquema da integração no sistema do veículo	62
6.2	Esquema das soluções para a criação de um mosaico de imagens.	63
6.3	Pontos detectados pelo algoritmo <i>SIFT</i>	65
6.4	Representação dos descritores dos pontos de interesse.	65
6.5	Invariância à rotação da imagem	66
6.6	Invariância à escala da imagem	66
6.7	Correspondência através de descritores.	67

6.8	Resultado da exclusão de pontos de interesse com <i>RANSAC</i>	69
6.9	Resultado da aplicação da rotação do <i>INS</i> aos pontos característicos da imagem	70
6.10	Análise da disparidade nos eixos dos $xx's$ e $yy's$	70
6.11	Análise da disparidade nos eixos dos $xx's$ e $yy's$ após exclusão de más associações	71
6.12	Pontos excluídos nas imagens.	71
6.13	Resultado final da descartação de pontos	72
6.14	Esquema das etapas para a fusão de imagens	73
6.15	Exemplo da junção de imagens com transformação <i>affine</i>	75
6.16	Imagens de base para o mosaico de imagens	77
6.17	Exemplo da junção de imagens com transformação projectiva	77
6.18	Atitude em <i>roll</i> , <i>pitch</i> e <i>yaw</i>	79
6.19	Esquema das etapas para a decomposição da matriz de homografia.	80
6.20	Disparidades provocadas por duas hipóteses de rotação da matriz H	81
6.21	Comparação dos resultados obtidos para a atitude através de <i>INS</i> e decomposição H	82

Lista de Tabelas

3.1	Tabela resumo das Transformações 2D	22
3.2	Tabela de hipóteses de decomposição da matriz essencial.	33
3.3	Hipóteses de solução da decomposição da matriz de homografia	35
6.1	Análise dos tempos de processamento.	72

Esta página foi intencionalmente deixada em branco.

Símbolos

X, Y, Z	-	Coordenadas de ponto no mundo
x, y, z	-	Coordenadas de ponto na imagem
x', y', z'	-	Coordenadas correspondente a x, y, z numa segunda imagem
M	-	Ponto no mundo
m	-	Ponto na imagem
m'	-	Ponto correspondente a m numa segunda imagem
f	-	Distância focal
P	-	Matriz projecção perspectiva
K	-	Matriz dos parâmetros intrínsecos
Π	-	Matriz dos parâmetros extrínsecos
O_i	-	Coordenadas do ponto principal no eixo i
s_θ	-	Factor de torção dos parâmetros intrínsecos
s_i	-	Número de pixels por unidade de distância no eixo i
R	-	Matriz de rotação
t	-	Vector translação
I	-	Matriz identidade
H_E	-	Matriz de homografia caracterizada pela transformação euclidiana
H_S	-	Matriz de homografia caracterizada pela transformação de similaridade
H_A	-	Matriz de homografia caracterizada pela transformação <i>affine</i>
H_P	-	Matriz de homografia caracterizada pela transformação projectiva
e	-	Epipólo
l	-	Linha Epipolar
F	-	Matriz Fundamental
E	-	Matriz Essencial
Σ	-	Matriz diagonal com constantes σ_1, σ_2 e σ_3
Z	-	Matriz que representa uma rotação de $\frac{\pi}{2}$
$m(x, y)$	-	Magnitude do ponto de interesse de coordenadas x, y
$\theta(x, y)$	-	Orientação do ponto de interesse de coordenadas x, y
D	-	Dados de entrada do algoritmo <i>RANSAC</i>
N	-	Número de elementos dos dados de entrada do algoritmo <i>RANSAC</i>
s	-	Conjunto de dados <i>MSS</i> (<i>Minimal Sample Set</i>)
δ	-	<i>Threshold</i> do algoritmo <i>RANSAC</i>
p	-	Probabilidade de pelo menos um conjunto s não conter dados falsos
C	-	Função mínimo custo do algoritmo <i>RANSAC</i>
ϕ	-	Ângulo de rotação <i>roll</i>
θ	-	Ângulo de rotação <i>pitch</i>
	-	Parâmetros do modelo do algoritmo <i>RANSAC</i>
ψ	-	Ângulo de rotação <i>yaw</i>

Esta página foi intencionalmente deixada em branco.

Acrónimos

2D	-	Bidimensional
3D	-	Tridimensional
AUV	-	<i>Autonomous Underwater Vehicle</i>
CS	-	<i>Consensus Set</i>
DGPS	-	<i>Diferential Global Positioning System</i>
ECEF	-	<i>Earth Centered Earth Fixed Frame</i>
ECI	-	<i>Earth Centered Inertial Frame</i>
GPS	-	<i>Global Positioning System</i>
IMU	-	<i>Inercial Measurement Unit</i>
INS	-	<i>Inertial Navigation System</i>
IR	-	<i>Infra-Red</i>
ISEP	-	Instituto Superior de Engenharia do Porto
KLT	-	<i>Kanade-Lucas-Tomasi Feature Tracker</i>
LIDAR	-	<i>Laser Infrared Detection and Ranging</i>
LMedS	-	<i>Least Median Squares</i>
LMS	-	<i>Least Mean Squares</i>
LSA	-	Laboratório de Sistemas Autónomos
MLSEAC	-	<i>Maximum Likelihood Estimation Sample and Consensus</i>
MSAC	-	<i>M-estimator Sample Consensus</i>
MSS	-	<i>Minimal Sample Set</i>
nDLT	-	<i>Normalized Direct Linear Transformation</i>
NED	-	<i>North East Down Frame</i>
P-RANSAC	-	<i>Preemptive RANSAC</i>
PD	-	<i>Proportional Derivative</i>
PID	-	<i>Proportional Integral Derivative</i>
PPM	-	<i>Perspective Projection Matrix</i>
PROSAC	-	<i>Progressive Sample Consensus</i>
R-RANSAC	-	<i>Randomized RANSAC</i>
RANSAC	-	<i>Random Sample Consensus</i>
RGFD	-	<i>Regional Gray Fitting Degree</i>
ROV	-	<i>Remote Operated Vehicle</i>
SIFT	-	<i>Scale-Invariant Feature Detector</i>
SLAM	-	<i>Simultaneous Localization and Mapping</i>
SUSAN	-	<i>Smallest Univalve Segment Assimilating Nucleus</i>
UAV	-	<i>Unmanned Aerial Vehicle</i>
UGV	-	<i>Unmanned Ground Vehicle</i>
USV	-	<i>Unmanned Surface Vehicle</i>
VTOL	-	<i>Vertical Take-Off Landing</i>
WGS-84	-	<i>World Geodetic System 1984 Frame</i>

Esta página foi intencionalmente deixada em branco.

Capítulo 1

Introdução

Conteúdo

1.1	Âmbito da Dissertação	1
1.2	Enquadramento e Motivação	3
1.3	Objectivos	4
1.4	Organização da Dissertação	5

1.1 Âmbito da Dissertação

A inspecção visual, de infra-estruturas, de áreas de difícil acesso ou áreas em ambientes hostis à presença humana assume nos dias de hoje um carácter determinante, no processo de manutenção de infra-estruturas críticas, na monitorização de áreas de risco e/ou sinistradas e na vigilância de áreas de acesso restrito ou controlado.

Nesta dissertação abordamos os elementos necessários ao desenvolvimento de um sistema de inspecção visual. Este caracteriza-se pela capacidade de: operação em ambientes hostis ou de difícil acesso; obtenção de imagens estáticas ou a partir de movimento de baixas dinâmicas; criação de mapas de áreas ou objectos para inspecção; e, navegação em tempo real. As linhas de trabalho abordadas são: mosaico de imagens; técnicas de mapeamento baseadas em mosaico de imagens¹; e, apoio à navegação relativa de aeronaves em

¹Mosaico de imagens refere-se à criação de um mapa para inspecção e/ou navegação, com base num conjunto indeterminado de imagens que são associadas de forma a obter uma só imagem.

tempo real em função dos objectos e áreas mapeadas permitindo o referenciamento das mesmas.

Frequentemente, nos cenários que endereçamos, o ponto e o sentido de observação mais favorável nem sempre pode ser acedido com os meios convencionais de observação. Nos casos em que as áreas a observar se encontrem paralelas á superfície terrestre e o sentido de observação é na vizinhança do sentido que aponta para o centro da terra, seria necessário obter imagens a partir de um ponto posicionado a uma altitude superior à área a ser observada. Por outro lado, o detalhe necessário à observação, nem sempre é compatível com a tomada de imagens de grandes distâncias, quer pela resolução da imagem e exactidão da referenciação da mesma, quer pela sua oclusão por outras infra-estruturas ou morfologia do terreno. Acresce que, em muitas aplicações, por questões de segurança a observação e vigilância não deve ser detectada. Os veículos aéreos não tripulados (*UAV's*) apresentam-se como uma solução eficaz (ausência de riscos humanos e baixo custo sistémico) abrindo novas perspectivas para as aplicações de inspecção e monitorização.

As recentes evoluções de sistemas computacionais e sistemas de micro-processamento, o aumento da densidade energética das baterias e a diminuição da dimensão dos actuadores, tem catalizado o desenvolvimento de UAV bem como a disseminação de aplicações, sobretudo na categoria dos *VTOL* (*Vertical Take-Off Landing*) e em especial os quadrotors. O interesse por estes veículos prende-se em parte pela sua versatilidade para aplicações nas mais diversas áreas, nomeadamente em operações de inspecção. Os *UAV* da classe *VTOL* destacam-se dos restantes *UAV* devido à sua capacidade de voar a baixa altitude, efectuar voo estacionário, descolar e aterrar na vertical. Estas características proporcionam vantagens em relação aos mais comuns *UAV* de asa fixa, tais como poderem descolar/aterrar em espaços reduzidos, a sua utilização em ambientes *indoor*, e permitir o voo estacionário, ou quase estacionário de grande relevância para as operações de inspecção. Na classe dos *VTOL* verifica-se recentemente uma preferência pelos quadrotors em relação aos comuns helicópteros, devido á menor complexidade mecânica e maior facilidade em executar o controlo da aeronave.

Associado ao crescente interesse pela utilização de quadrotors e também catalizada pelas evoluções dos sistemas computacionais está ligada a evolução de alguns sistemas percepção por visão, que permitem acrescentar funcionalidades ao quadrotor como: o seguimento de objectos; o apoio à navegação em ambientes *indoor*, ou em caso de falha de

sensores como *GPS*, a criação de mapas para navegação ou inspecção.

Nas operações de inspecção remota nem sempre o *streaming* de vídeo em tempo real é a melhor representação da informação para o utilizador, pois, por vezes os fenómenos a detectar dependem mais de relações espaciais dos elementos observados do que, da sua ordem de observação. Consequentemente, será preferível a construção de uma representação única do cenário de inspecção (com o detalhe necessário aos objectivos da aplicação em causa), através da composição de um mosaico de imagens e que nos permite o registo e a utilização posterior da informação em ambientes 4D².

De forma semelhante, a navegação através de um sistema de visão, pode ser obtida a partir da junção de pontos de interesse de imagens. Assim o processo de criação de mosaicos de imagens vai permitir simultaneamente disponibilizar uma melhor representação do cenário de inspecção e ainda fornecer informação para o sistema de navegação de veículos autónomos.

A obtenção de um mosaico de imagens pode ser feita com base em duas fontes: imagens capturadas de uma cena ou objecto, ou através dos *frames* de um vídeo de uma cena ou objecto. Como a junção de imagens não é uma tarefa trivial, abordamos conceitos de visão computacional, tais como alinhamento de imagens, detecção de pontos de interesse em imagens, correspondência de pontos entre imagens e estimação robusta aplicada a visão. Conceitos estes que também são necessários para determinar a posição e orientação da câmara relativa ao mapa no instante em que a imagem é capturada.

1.2 Enquadramento e Motivação

O Laboratório de Sistemas Autónomos (*LSA*) do Instituto Superior de Engenharia do Porto (*ISEP*), tem vindo a desenvolver ao longo dos últimos anos vários veículos autónomos e sub-sistemas com estes relacionados, enderçando os mais variados ambientes de operação. Existe também um esforço para desenvolver sistemas sensoriais, que permitam um aumento das capacidades de percepção e operação dos veículos desenvolvidos. No âmbito de inspecção e busca e salvamento em ambientes não estruturados está em curso o desenvolvimento de um quadrotor [4] que integrará uma equipa de robots autónomos. Um sistema de visão para percepção, que permita a obtenção de mapas para

²4D refere-se a quatro dimensões definidas pela posição nas coordenadas x, y, z e o tempo

inspecção detalhada de uma zona ou estrutura e o apoio à navegação, torna-se de grande interesse nestas aeronaves.

Em casos de aplicações civis os quadrotors podem ser usados, por exemplo, para funções de inspecção, monitorização, vigilância e mapeamento aéreo. Neste tipo de funções poderia destacar-se a sua utilização para inspecção de linhas de alta/média tensão, caminhos-de-ferro, pontes, edifícios, torres eólicas ou vítimas em cenários de catastrophe.

Para fins militares, o quadrotor pode ser usado em cenários de guerra, tendo como funções, por exemplo, fazer o reconhecimento de uma área hostil ou fazer a avaliação de estragos num combate. Outra aplicação ainda neste grupo, seria a sua utilização por parte da polícia para a vigilância de multidões, monitorização de tráfego ou o seguimento de veículos terrestres.

Por fim, as aplicações científicas que se podem dar a este tipo de veículo serão sempre no âmbito de desenvolver técnicas de controlo e navegação de modo a tornar mais simples a sua utilização em meios civis e militares. Assim, no mundo científico o quadrotor pode ser usado para manobras de voo estacionário (*hovering*), seguimento de objectos, navegação por *waypoints* e coordenação com outros veículos autónomos (*UAV*, *USV* - *Unmanned Surface Vehicle*, *UGV* - *Unmanned Ground Vehicle*).

1.3 Objectivos

Este trabalho aborda o problema da criação de mosaicos de imagens em tempo real, capazes de servir como mapas de navegação para *UAV's*, bem como em cenários de inspecção permitir dotar o operador de uma melhor percepção das zonas e estruturas sob inspecção. A criação destes mosaicos é concebida para cenários em que o *UAV* mapeia uma determinada área de interesse para inspecção. Na obtenção do mosaico em tempo real pode-se ainda obter informação sobre a posição e orientação da aeronave o que poderá ajudar na navegação, de forma a conseguir corrigir/determinar a atitude³ da aeronave.

Assim, os objectivos desta dissertação passam por:

- implementar e avaliar abordagens do estado da arte em técnicas de mosaico para aplicações de inspecção e navegação em tempo real;

³Orientação tridimensional definida pelos ângulos *roll*, *pitch* e *yaw*

- contribuir para a evolução das técnicas do estado da arte para a utilização do mosaico na navegação por forma a obter a atitude da aeronave em tempo real;
- propor uma arquitectura para o sistema de navegação e inspecção visual do quadrotor baseado em mosaico de imagens;
- definir estratégias para a criação de mapas de navegação através de mosaico

1.4 Organização da Dissertação

Nos capítulos seguintes são apresentadas técnicas e soluções para a criação de mosaicos de imagens e para a determinação da orientação e posição de uma câmara.

No segundo capítulo desta dissertação são apresentados alguns dos sistemas de quadrotors mais relevantes. É feita também uma análise aos sistemas de visão destes quadrotors, desde logo com o intuito de perceber que técnicas são utilizadas para criação de mapas que permitam a inspecção.

De seguida, no capítulo 3, são expostos os conceitos e fundamentos de visão computacional necessários sobre transformações geométricas e geometria aplicada a visão para relação de várias imagens.

A aplicação dos conceitos de visão de computacional ao problema de criação de mosaico de imagens pode ser vista no capítulo 4, onde são apresentados o algoritmo *SIFT* (*Scale-Invariant Feature Transform*) que permite a detecção de pontos em imagens e o algoritmo *RANSAC* (*Random Sample Consensus*) para determinar as correctas correspondências entre duas imagens. É também neste capítulo que é feita a referência ao modo como as imagens são fundidas.

No capítulo 5 é feita uma breve referência aos fundamentos de navegação, sobre referenciais e informação fornecida por sensores inerciais.

Nos dois últimos capítulos é feita a descrição da implementação e resultados obtidos (capítulo 6) e as respectivas conclusões (capítulo 7). Na implementação do projecto são apresentados todos os passos para a criação de um mosaico de imagens e a obtenção da posição e orientação da câmara. Por fim são apresentadas todas as conclusões retiradas da realização desta dissertação.

Esta página foi intencionalmente deixada em branco.

Capítulo 2

Estado da Arte

Conteúdo

2.1	Projectos Relacionados de Quadrotors	8
2.1.1	<i>Starmac</i>	8
2.1.2	<i>Swarm</i>	8
2.1.3	<i>OS4</i>	9
2.1.4	<i>GRASP Laboratory</i>	10
2.1.5	<i>Team Swift</i>	10
2.1.6	<i>X4 - Flyer</i>	11
2.2	Técnicas de Navegação Aplicadas em <i>UAV's</i>	11
2.3	Sistemas de Visão em <i>UAV's</i>	12

Neste capítulo vamos fazer uma breve apresentação dos projectos mais relevantes de quadrotors, focando os aspectos ligados aos sensores e métodos para a percepção e navegação.

Seguidamente vamos fazer um resumo das técnicas de navegação de *UAV's* mais utilizadas, terminando com uma descrição de projectos relacionados que usam visão para a criação de mosaico de imagens e para mapas para navegação.

2.1 Projectos Relacionados de Quadrotors

Os quadrotors cada vez mais se tornam uma classe de grande interesse na área dos *UAV's VTOL*, devido há maior facilidade de controlo destas aeronaves quando comparadas com outras da sua classe como por exemplo o helicóptero. Esta melhor manobrabilidade sobre o veículo permite que este possa ser utilizado num vasto leque de aplicações, que consequentemente levam ao desenvolvimento de enumeras técnicas e sistemas de navegação para estes *UAV's*.

Nesta secção são apresentados alguns dos projectos de desenvolvimento de quadrotors mais relevantes.

2.1.1 *Starmac*

A *Stanford University* apresenta um projecto de um quadrotor bastante avançado nos aspectos de navegação e controlo. Para efectuar o controlo do veículo *STARMAC* foram utilizados técnicas de controlo como o *Integral Sliding Mode* e o *Reinforcement Learning*. O processamento da informação de navegação da aeronave não é feito *on-board* mas sim numa estação de terra, libertando assim algum peso do *UAV Starmac* que seria necessário para fazer transportar um sistema computacional. Na estação de terra é feita toda a computação e fusão de dados provenientes dos sensores presentes no veículo (*IMU*, sonar e *GPS* diferencial), através de um filtro de *Kalman* [5].

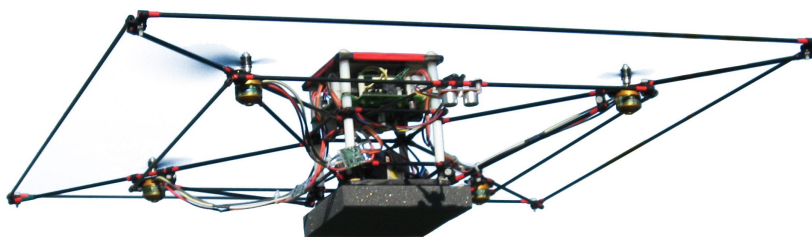


Figura 2.1: Quadrotor *Starmac* da *Stanford University*.

2.1.2 *Swarm*

A par da *Stanford University*, temos o projecto *SWARM* do *Massachusetts Institute of Technology (MIT)* também bastante desenvolvido. Este projecto tem trazido grandes progressos na área de coordenação de multi-robots *UAV's* quadrotors. A coordenação destes *UAV's* é feita autonomamente através do ajuste e comunicação de posições as

diversas aeronaves [6]. O controlo e navegação da aeronave são feitos através da técnica *Moire Patterns*, que utiliza dados de um sistema inercial (*IMU*) e um sistema de visão para a determinação da posição e orientação do veículo [7]. Aqui o sistema de visão não está presente a bordo mas sim em terra, colocado de forma a captar o espaço de operação dos quadrotors.



Figura 2.2: Quadrotor *Swarm* do MIT.

2.1.3 *OS4*

O *OS4* é um projecto da *École Polytechnique Fédérale de Lausanne (EPFL)*. A investigação e desenvolvimento feito sobre esta plataforma é bastante importante pois permitiu implementar e analisar vários tipos de controlo como *Backstepping*, *Sliding Mode* e o convencional *PID* [8, 9], sobre quadrotors. O trabalho neste veículo permitiu a *S. Bouabdallah* e *R. Siegwart*, construir e chegar a um modelo matemático bem descrito, de um quadrotor. Neste veículo existem os seguintes sensores, para navegação: *IMU*, sonar e sistema de visão [10]. O *IMU* permite saber a atitude do veículo em voo. O conjunto de sonares tem como função a percepção e detecção de obstáculos, a fim de os evitar. O sistema de visão permite saber o *heading* da aeronave (rotação em *yaw*) e obter medidas de disparidade linear. A informação destes sensores é fundida de forma a obter dados que permitam a navegação do *OS4*.

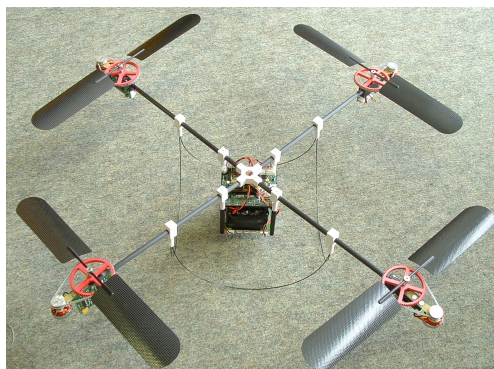


Figura 2.3: Quadrotor *OS4* da *EPFL*.

2.1.4 *GRASP Laboratory*

No *GRASP Laboratory* da *University of Pennsylvania*, foi usada uma plataforma de quadrotor para explorar metodologias de controlo e algoritmos de estimação de posição [11]. A estabilização deste veículo foi feita com recurso ao controlo através de *Backstepping*. Este projecto utiliza um sistema de visão, composto por duas câmaras: uma em terra e outra a bordo, aliado a um giroscópio para efectuar a percepção e navegação. O conjunto giroscópio mais câmara de bordo permite executar missões de seguimento de objectos. A câmara de terra tem a particularidade de ser *pan & tilt* e permite captar o espaço de operação do *UAV*. Esta segunda câmara permite ainda estimar a posição da aeronave.

2.1.5 *Team Swift*

A *Team Swift* do *Indian Institute of Technology Madras* (IIT Madras), é uma equipa composta por alunos de variadas áreas de engenharia que se juntaram com a finalidade de desenvolver um *UAV* para participar no *IARC* (*Internacional Aerial Robotics Competition*). Neste *UAV* estão presentes os seguintes sensores: *IMU*, *LIDAR* (*Laser Infrared Detection and Ranging*), sistema de visão, sonar e microfone. Com base na informação destes sensores é possível a aeronave voar estavelmente através de um controlo *PD* (*Proportional Derivative*). A navegação deste *UAV* é feita através de *SLAM* (*Simultaneous Localization and Mapping*), onde o *LIDAR* assume grande importância no mapeamento do ambiente em redor do veículo. O sistema de visão neste veículo apenas é utilizado para a detecção de objectos [12].



Figura 2.4: Quadrotor *Team Swift* do *IIT Madras*.

2.1.6 *X4 - Flyer*

O *X4 - Flyer* da *Australian National University (ANU)* é dos quadrotors mais simples a nível sensorial pois apenas utiliza um *IMU*. Com este projecto era pretendido fazer o estudo da dinâmica e força de impulsão do *X4 - Flyer* [13]. Para esta aeronave foi utilizado um controlo através de *PID*.

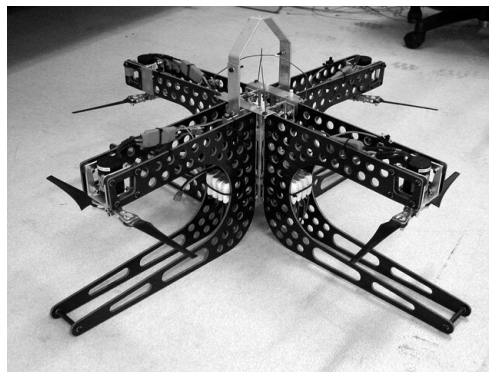


Figura 2.5: Quadrotor *X4 - Flyer* da *ANU*.

2.2 Técnicas de Navegação Aplicadas em UAV's

Uma das vertentes mais relevantes dos projectos acima mencionados são as metodologias de controlo e técnicas de navegação. Estes métodos estão muitas vezes relacionados com *UAV's* na sua classe geral, sendo depois alguns destes transferidos para projectos concretos, no nosso caso de interesse os quadrotors.

Uma das combinações de sensores mais utilizadas e mais descritas em quadrotors, é a utilização de visão em conjunto com *IMU* e *GPS (Global Positioning System)*. Estes dois

sensores só por si serviriam para efectuar a localização da aeronave, a utilização/apoio do sistema de visão prende-se com o facto da necessidade de estimação de altitude [14, 15], posição e velocidade [11, 16] da aeronave aquando da falha de sinal *GPS* ou para navegação em ambientes *indoor*. A utilização de um sistema de visão neste tipo de veículo para além de um sistema de apoio à localização deste, pode ainda servir como um sistema para a percepção do ambiente em seu redor com a finalidade de detectar e desviar o veículo de obstáculos [17]. Também para a detecção e desvio de obstáculos podem ser usados sensores acústicos [18]. Para estas aeronaves poderem navegar é necessário fundir a informação proveniente dos diversos sensores recorrendo a métodos de estimação. Estes métodos permitem estimar a posição, orientação, velocidade, atitude ou qualquer outra variável de navegação através da fusão de informação proveniente de diferentes sensores. O filtro mais aplicado que utiliza estes métodos para fusão sensorial, é o filtro de *Kalman* e as suas variantes *Extended (EKF)* [19] e *Unscent (UKF)* [20]. Existem também outros tipos de filtros como o complementar, utilizado para a estimação da atitude do veículo [21] e o filtro adaptativo [18].

Por vezes estes estimadores apresentam elevada sensibilidade a certos parâmetros do filtro (variância das fontes de ruído) levando a fracos desempenhos destes ou mesmo a divergências nos valores estimados, este motivo leva a que em alguns casos seja ainda anexado ao filtro um método controlo, como por exemplo o *Fuzzy Logic Controller*, que tem como função ajustar as variáveis de ruído, tornando a estimação mais robusta [19, 22].

2.3 Sistemas de Visão em *UAV's*

Como tem vindo a ser notório ao longo deste capítulo, muitos dos projectos de quadrotors e *UAV's* na sua generalidade utilizam sistemas de visão. Estes sistemas, podem assumir funções como a criação de mapas para inspecção e navegação, estimação de posição e orientação do veículo (odometria visual e estrutura do movimento).

Para a função de criar mapas para inspecção, destaca-se a técnica de mosaico de imagens. Na maioria dos projectos aqui apresentados este método é executado em três fases: na primeira são determinados os pontos de interesse em duas imagens da mesma cena ou objecto; na segunda são eliminadas as falsas correspondências, por forma a ser efectuada uma correcta correspondência entre as imagens; na terceira são unidas as imagens.

O projecto *DIVA* da Universidade de Coimbra, apresenta-se bastante desenvolvido na criação de mapas 3D e mapas construídos a partir de um mosaico de imagens. Neste projecto é explorada a fusão de técnicas de visão stereo com dados de um sistema de navegação inercial (*INS*), com o objectivo de construir um mapa 3D com imagens adquiridas de um observador (sistema de visão) em movimento [23]. Neste sistema o sensor inercial está fixo à câmara, formando assim um só objecto sólido, fornecendo deste modo os dados relativos à posição e movimento do sistema de visão. Através destes dados é possível relacionar a informação das imagens captadas, podendo assim construir um mapa a partir de várias imagens (mosaico de imagens). A nível de *software* são utilizados algoritmos que permitem determinar os pontos característicos (*features*) de imagens (algoritmo *SIFT*) e que permitem a exclusão de falsas correspondências entre imagens (algoritmo *RANSAC*). Para a obtenção do mapa final as imagens são depois unidas (*stitching*) pelos pontos resultantes da verificação das associações [24]. Com o domínio das técnicas de criação de mapas a partir de um mosaico de imagens, os investigadores do projecto *DIVA* fizeram avanços tecnológicos conseguindo fazer em simultâneo com a criação do mapa o seguimento de um objecto no solo [25].

Em [26], o *Beijing Institute of Technology*, apresenta um método de criação de um mosaico de imagens em três passos para aplicação em *UAV's*. No primeiro passo é executado o algoritmo *SUSAN* (*Smallest Univalued Segment Assimilating Nucleus*) que permite a detecção de pontos característicos na imagem (exemplo de cantos ou linhas), com bastante precisão em imagens portadoras de muito ruído. De seguida, é utilizado o conceito *RGFD* (*Regional Gray Fitting Degree*) que tem como função eliminar os pontos de interesse mal caracterizados, limitando assim a informação apenas aos pontos característicos com boa informação. Por fim, tendo os pontos de interesse é feita a coincidência de duas imagens, gerando assim um mapa como um mosaico de imagens.

Os mapas criados a partir de imagens não se restringem exclusivamente a fotos, podendo também ser obtidos de um vídeo. Nestes as imagens são originadas a partir dos *frames* de vídeo. Na *University of North Dakota* foi desenvolvido um sistema que permite a criação de um mapa através do vídeo obtido por uma câmara *IR* (*Infra-Red*) [27]. De igual modo ao que acontece com as fotos, também sobre os *frames* de vídeo são executados algoritmos para a detecção de pontos característicos e coincidência de dois *frames*. Os algoritmos utilizados neste projecto de um pequeno *UAV* são o *SIFT* (*Scale Invariant*

Feature Transform) e *RANSAC* (*Random Sample Consensus*). Neste sistema o algoritmo para a criação de mapas é executado remotamente e tempo real.

O *Tecnhion - Israel Institute of Technology*, apresenta um método de formação de mosaico de imagens *on-line*, aplicado num *UAV* que dispõe de uma câmara rotativa para capturar imagens [28]. Tal como na maioria dos trabalhos aqui apresentados, a formação do mosaico é conseguida através dos algoritmos *SIFT* e *RANSAC*. A informação do mapa gerado auxilia na navegação da aeronave quando esta perde o sinal *GPS*, com informação de posição e orientação.

Os sistemas de criação de mosaicos permitem também aos veículos efectuar a navegação com base em informações nos mapas criados. O *Aerospace Robotics Laboratory* da *Stanford University*, apresenta um sistema de navegação *dead-reckoning* baseado em visão [29]. Este sistema é aplicado num *AUV/ROV* (*Autonomous Underwater Vehicle / Remote Operated Vehicle*) e permite criar um mosaico de imagens do fundo do oceano remotamente em tempo real. Através de uma estação de terra, onde é processada toda a informação, é possível definir um objecto no mapa e colocar o veículo a seguir esse alvo. Ainda na área da robótica marinha, em [30] é apresentado um sistema de navegação baseado em visão para veículos autónomos subaquáticos com o uso de mosaico de imagens. O mapa neste sistema é criado *off-line* com base em *frames* de vídeo. Em cada *frame* é utilizado o detector *Harris corner*, para a detecção de pontos característicos na imagem. De seguida sobre a imagem é utilizado o método de estimação robusta *LMedS* (*Least Median Squares*) para efectuar a coincidência de duas imagens. Com o mapa gerado, o veículo é capaz de executar missões de seguimento de objectos e seguimento de *waypoints* com base na informação do mapa. Este sistema prevê ainda um algoritmo para evitar que o veículo de desloque para as zonas de fim do mapa (bordas), onde a coincidência das imagens é mais difícil, mantendo-o assim afastado dessa área.

Capítulo 3

Fundamentos de Visão

Conteúdo

3.1	Câmara	16
3.1.1	Parâmetros Intrínsecos	17
3.1.2	Parâmetros Extrínsecos	19
3.2	Transformações	21
3.2.1	Transformações 2D	22
3.2.2	Transformações 3D	27
3.3	Relação entre duas imagens	30
3.4	Geometria Epipolar	31
3.4.1	Matriz Fundamental	32
3.4.2	Decomposição da Matriz Essencial	32
3.5	Decomposição Matriz de Homografia	33

Os objectivos dos algoritmos de visão computacional são obter informação sobre geometria e fotogrametria ¹ a partir da imagem. Isto pode incluir, a detecção de um objecto e a sua posição na imagem, bem como a sua reconstrução; a determinação da posição e orientação da câmara; ou ainda as transformações necessárias para a criação de um mosaico de imagens [31].

¹A fotogrametria é uma técnica que tem por objectivo determinar as propriedades geométricas, dimensões e posicionamento, dos objectos no espaço, a partir de imagens

Todas estas aplicações necessitam extrair informação da imagem sob a forma de parâmetros que permitem descrever modelos de movimento, linhas, cantos, superfícies ou modelos de transformações.

Para este trabalho a informação relativa a transformações e a estimação da posição e orientação assumem grande importância. Esta informação vai permitir obter o alinhamento e relação entre imagens na criação de um mosaico de imagens. Quanto aos dados de posição e orientação da câmara podem ser relacionados com dados de navegação (ex.: *INS*), servindo assim para dar apoio na navegação do veículo.

3.1 Câmara

O modelo mais comum de uma câmara utilizado em visão por computador é o *pinhole*, também conhecido como projecção perspectiva. Esta é uma maneira simples de modelizar a maioria das câmaras de CCD modernas, considerando que projecção de raios de luz passa através de um pequeno buraco, sendo projectados numa superfície plana (imagem).

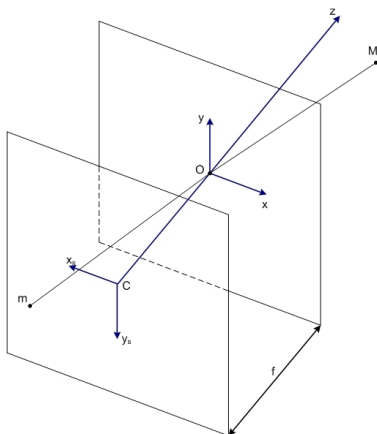


Figura 3.1: Modelo de uma câmara *pinhole* ou de projecção perspectiva.

A imagem do ponto 3D, M , sofre uma projecção perspectiva, através da passagem pelo centro óptico, O , e consequente projecção no plano da imagem. A distância entre o centro óptico e o plano da imagem é denominada de distância focal, f . A linha que liga o centro óptico e o ponto principal da imagem C , é o eixo óptico.

O uso de geometria projectiva permite descrever a projecção perspectiva através de uma equação linear (3.1), o que torna o modelo de fácil utilização.

$$\begin{bmatrix} x \\ y \end{bmatrix} = -\frac{f}{Z} \begin{bmatrix} X \\ Y \end{bmatrix} \quad (3.1)$$

onde X , Y , Z , são as coordenadas do ponto M . O sinal negativo na equação, indica que a imagem formada no plano da imagem está invertida. Este efeito pode ser resolvido invertendo a imagem, para isso $(x, y) \mapsto (-x, -y) = (x_s, y_s)$, o que corresponde a colocar o plano da imagem à frente do centro óptico [2]. Podendo então reformular 3.1:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{f}{Z} \begin{bmatrix} X \\ Y \end{bmatrix} \quad (3.2)$$

Em coordenadas homogéneas esta relação pode ser escrita da seguinte forma:

$$Z \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (3.3)$$

onde P é a matriz de projecção perspectiva (PPM):

$$P = K[R|t] \quad (3.4)$$

A matriz K depende dos parâmetros intrínsecos da câmara (que são descritos na secção 3.1.1) e R e t são as componentes dos parâmetros extrínsecos que são tratados na secção 3.1.2. A matriz PPM permite mapear aproximadamente um ponto do mundo num ponto de duas dimensões, através da seguinte relação:

$$m \approx PM \quad (3.5)$$

3.1.1 Parâmetros Intrínsecos

Os parâmetros intrínsecos de uma câmara servem para relacionar as coordenadas do plano da câmara com as do plano da imagem, ou seja, relacionar o referencial da imagem

(coordenadas de pixels) com o referencial da câmara (coordenadas normalizadas). Estes parâmetros estão presentes na matriz K da equação 3.4.

Na prática quando se captura uma imagem as medidas são obtidas em termos de pixels, com o sistema de coordenadas da imagem com origem normalmente no canto superior da imagem (i, j) . Os parâmetros intrínsecos vão permitir normalizar estas medidas.

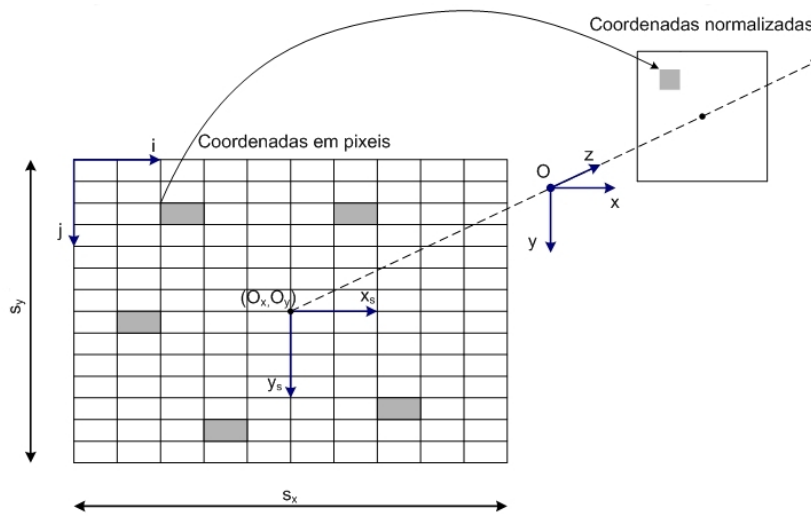


Figura 3.2: Transformação de pixels para coordenadas normalizadas.

O primeiro passo consiste em especificar as unidades dos eixos $xx's$ e $yy's$: se (x, y) estiverem em unidades métricas (por exemplo: milímetros), e (x_s, y_s) são escalados em relação às coordenadas de pixels (i, j) , então a transformação pode ser descrita por uma matriz de escalonamento (equação 3.6)

$$\begin{bmatrix} x_s \\ y_s \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (3.6)$$

Esta matriz, depende do tamanho dos pixels nas direcções do eixo dos $xx's$ e $yy's$, ou seja o número de pixels por unidade de distância (s_x e s_y). Quando $s_x = s_y$, significa que cada pixel é quadrado. No entanto nem sempre isto acontece e quando estes valores são diferentes os pixels vão ser rectangulares.

Em segundo lugar tem de se transferir a origem do referencial (x_s, y_s) para a origem

do referencial da imagem (i, j) .

$$\begin{aligned} i &= x_s + O_x \\ j &= y_s + O_y \end{aligned} \quad (3.7)$$

onde (O_x, O_y) são as coordenadas do ponto principal da imagem, em relação ao referencial da imagem.

O último aspecto a ter em conta é o factor de torção (s_θ), este valor relaciona o ângulo entre os eixos (x_s, y_s) . Para a maioria das câmaras, o ângulo é de 90° ou muito próximo deste valor o que implica $s_\theta = 0$.

Estes três passos podem ser todos resumidos numa matriz transformação:

$$K_s = \begin{bmatrix} s_x & s_\theta & O_x \\ 0 & s_y & O_y \\ 0 & 0 & 1 \end{bmatrix} \quad (3.8)$$

Combinando a matriz K_s com a distância focal é obtida a matriz K :

$$K = \begin{bmatrix} s_x & s_\theta & O_x \\ 0 & s_y & O_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} f s_x & s_\theta & O_x \\ 0 & f s_y & O_y \\ 0 & 0 & 1 \end{bmatrix} \quad (3.9)$$

3.1.2 Parâmetros Extrínsecos

Os parâmetros extrínsecos ou estimação da posição de uma câmara, permitem relacionar segundo uma rotação (R) e uma translação (t), o referencial do mundo e o referencial da câmara, ou seja, é possível determinar o movimento da câmara.

Os valores de rotação e translação podem ser extraídos, por exemplo: das transformações tridimensionais (secção 3.2.2), matrizes de homografia (secção 3.5) e essencial (secção 3.4.2). Qualquer um destes exemplos apontados está intrínsecamente caracterizado pela maneira como transforma os três eixos base, podendo dizer-se que com essas transformações já existe uma componente de rotação e de translação.

$$\Pi = [R|T] \quad (3.10)$$

3.1.2.1 Rotação 3D

A rotação 3D permite determinar a posição da câmara no espaço tridimensional, através da relação entre a projecção de um ponto 3D em duas imagens.

Uma rotação pura significa que dois dos eixos são fixos e o movimento rotacional é feito em volta do restante eixo. O processo de rotação pura permite obter os três ângulos de rotação: *roll*, *pitch* e *yaw*, também conhecidos por ângulos de *Euler*.

Assim os três eixos base podem ser definidos da seguinte forma: rotação em *roll* (eixo dos zz 's), rotação em *pitch* (eixo dos xx 's) e rotação em *yaw* (eixo yy 's).

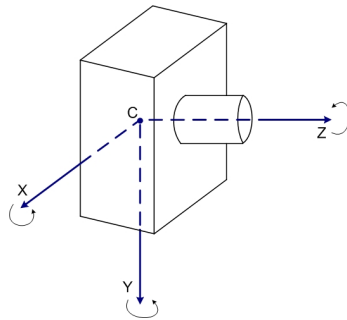


Figura 3.3: Representação dos eixos XYZ numa câmara.

A rotação em *roll* é feita através da rotação dos eixos xx 's e yy 's em torno do eixo dos zz 's. Esta é descrita algebricamente pela seguinte relação:

$$M = \begin{bmatrix} \cos\phi & -\text{sen}\phi & 0 \\ \text{sen}\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} M' \quad (3.11)$$

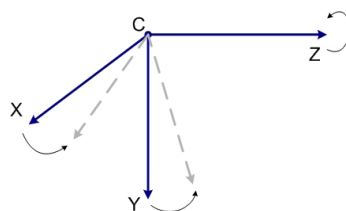


Figura 3.4: Rotação em torno do eixo dos zz 's.

A rotação em torno do eixo dos xx 's, também denominada por *pitch* é dada por:

$$M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\text{sen}\theta \\ 0 & \text{sen}\theta & \cos\theta \end{bmatrix} M' \quad (3.12)$$

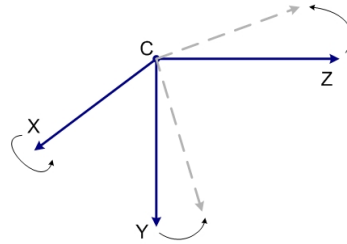


Figura 3.5: Rotação em torno do eixo dos xx 's.

Por fim, a rotação no eixo dos yy 's (*yaw*) matricialmente representa-se da seguinte forma:

$$M = \begin{bmatrix} \cos\psi & 0 & \text{sen}\psi \\ 0 & 1 & 0 \\ -\text{sen}\psi & 0 & \cos\psi \end{bmatrix} M' \quad (3.13)$$

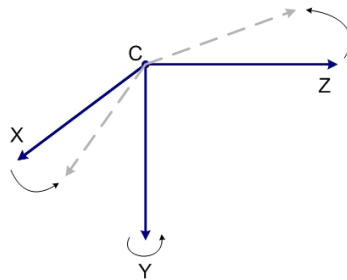


Figura 3.6: Rotação em torno do eixo dos yy 's.

3.2 Transformações

Uma transformação de coordenadas é uma operação matemática que leva as coordenadas de um ponto num sistema de coordenadas, para as coordenadas do mesmo ponto num segundo sistema de coordenadas.

Antes de unir duas ou mais imagens de uma cena de diferentes vistas ou câmaras, é necessário fazê-las coincidir através dos pontos de interesse da mesma região física da cena. É por isso, preciso estabelecer uma transformação entre os pontos de uma imagem e de outra, de forma a encontrar uma relação entre ambas. Esta relação é conseguida através das transformações 2D, transformações 3D e também a partir da estimação da posição da câmara (parâmetros extrínsecos). Uma transformação aplica uma deformação global à imagem, onde o comportamento da transformação é descrito apenas por um pequeno número de parâmetros.

Nesta secção são estudadas as transformações, que podem ser facilmente entendidas para o caso de duas dimensões (2D) pois nestas tem-se uma melhor visualização do problema. As transformações em três dimensões (3D) são apenas uma generalização do caso 2D. Sendo as transformações 2D normalmente utilizadas em aplicações de alinhamento de imagens, ou seja, permitem recolher parâmetros para relacionar duas imagens.

Nome	Matriz	Graus de Liberdade	Características invariantes
Euclidiana	$\begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix}$	3	Orientação Comprimentos Área
Similaridade	$\begin{bmatrix} sR & t \\ 0^T & 1 \end{bmatrix}$	4	Rácio dos comprimentos Ângulos
<i>Affine</i>	$\begin{bmatrix} A & t \\ 0^T & 1 \end{bmatrix}$	6	Paralelismo entre linhas Rácio de comprimentos entre linhas Rácio de áreas
Projectiva	$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$	8	Concorrência de pontos Colineariedade de pontos

Tabela 3.1: Resumo das funcionalidades das transformações bi-dimensionais.

3.2.1 Transformações 2D

As transformações 2D permitem relacionar/transformar geometricamente pontos, linhas e planos. As transformações mais elementares são a rotação e translação, que servem de base a todas as outras transformações.

Uma translação 2D é descrita por:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (3.14)$$

de forma compacta:

$$m' = \begin{bmatrix} I & t \\ 0^T & 1 \end{bmatrix} m \quad (3.15)$$

onde I é a matriz identidade 2x2 e $t \in \mathbb{R}^2$ é o vector translação expresso no referencial de m' .

Uma rotação 2D é definida como:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} \cos\theta & -\text{sen}\theta & 0 \\ \text{sen}\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (3.16)$$

de forma compacta:

$$m' = \begin{bmatrix} R & 0 \\ 0^T & 1 \end{bmatrix} m \quad (3.17)$$

onde R é a matriz de rotação.

A transformação projectiva generaliza todas as transformações, podendo dizer-se que as restantes (transformação euclidiana, similaridade e *affine*) são casos especiais da transformação projectiva [32].

3.2.1.1 Transformação Euclidiana 2D

A transformação euclidiana ou isométrica, preserva as distâncias euclidianas, ou seja mantém o tamanho da imagem permitindo apenas efectuar as duas transformações elementares: rotação e translação. Esta transformação é dada por:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} \epsilon \cos\theta & -\text{sen}\theta & t_x \\ \epsilon \text{sen}\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (3.18)$$

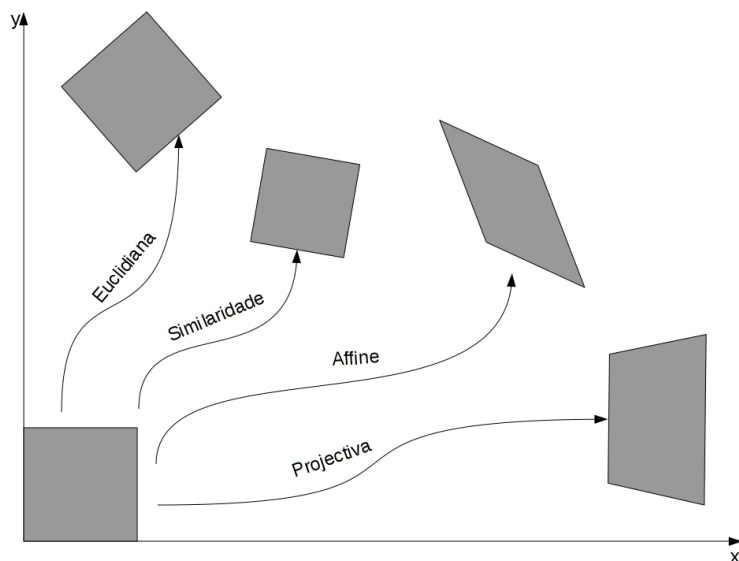


Figura 3.7: Deformações provocadas nas imagens através de transformações 2D.

onde, $\epsilon = \pm 1$. Se $\epsilon = 1$ a transformação preserva a orientação, ou seja é uma composição de rotação e translação. No caso de $\epsilon = -1$ a orientação é revertida, por exemplo um reflexo.

De forma compacta a transformação pode ser descrita pela matriz homografia da transformação de euclidiana (H_E da equação 3.19).

$$m' = H_E m = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} m \quad (3.19)$$

onde, R é uma matriz rotação ortogonal 2×2 ($R^T R = R R^T = I$) e t o vector translação.

Os casos especiais desta transformação são a rotação pura (quando $t = 0$) e a translação pura (quando $R = I$).

Uma transformação euclidiana tem três graus de liberdade, um para a rotação e dois para a translação. Assim a transformação pode ser obtida a partir da correspondência de dois pontos.

3.2.1.2 Transformação de Similaridade 2D

Uma transformação de similaridade não é mais que uma transformação euclidiana, mas com a diferença de nesta existir uma componente que permite alterar uniformemente a

escala da imagem (factor multiplicativo s). A sua representação em forma de matriz é dada por:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} s \cos\theta & -s \sin\theta & t_x \\ s \sin\theta & s \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (3.20)$$

que pode ser descrita de forma mais compacta como:

$$m' = H_s m = \begin{bmatrix} sR & t \\ 0^T & 1 \end{bmatrix} m \quad (3.21)$$

A transformação de similaridade tem quatro graus de liberdade, ou seja os mesmos três que a transformação euclidiana mais o grau de liberdade acrescentado pela alteração de escala da imagem. Com estas características é possível calcular a transformação a partir da correspondência de dois pontos. As características invariantes desta transformação é o rácio dos comprimentos.

3.2.1.3 Transformação *Affine* 2D

Uma transformação *affine* é uma transformação linear não singular seguida de uma translação. Esta transformação tem a particularidade de manter o paralelismo entre linhas, isto é, linhas que eram paralelas antes da transformação continuam a sê-lo depois. A transformação *affine* é representada matricialmente da seguinte forma:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (3.22)$$

e na sua forma mais compacta:

$$m' = H_A m = \begin{bmatrix} A & t \\ 0^T & 1 \end{bmatrix} m \quad (3.23)$$

com A a ser uma matriz não singular de tamanho 2x2:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad (3.24)$$

A transformação affine tem seis graus de liberdade, correspondendo cada um dos graus a um elemento da matriz 3.22. Para calcular todos os elementos desta matriz, tem de haver pelo menos a correspondência de três pontos.

A maneira mais fácil de compreender os efeitos geométricos causados pela matriz A é fazer a sua decomposição em duas transformações fundamentais: rotação e deformação 3.8. Assim a matriz pode ser definida como sendo:

$$A = R(\theta)R(-\phi)DR(\phi) \quad (3.25)$$

onde, $R(\theta)$ e $R(\phi)$ são as rotações por θ e ϕ respectivamente, e D a matriz diagonal:

$$D = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \quad (3.26)$$

Os parâmetros λ_1 e λ_2 são os factores de escala nas direcções x e y , respectivamente.

Comparando esta transformação com a de similaridade verifica-se que a diferença se prende com o facto de nesta transformação a alteração de escala não ser feita uniformemente mas sim em direcções ortogonais.

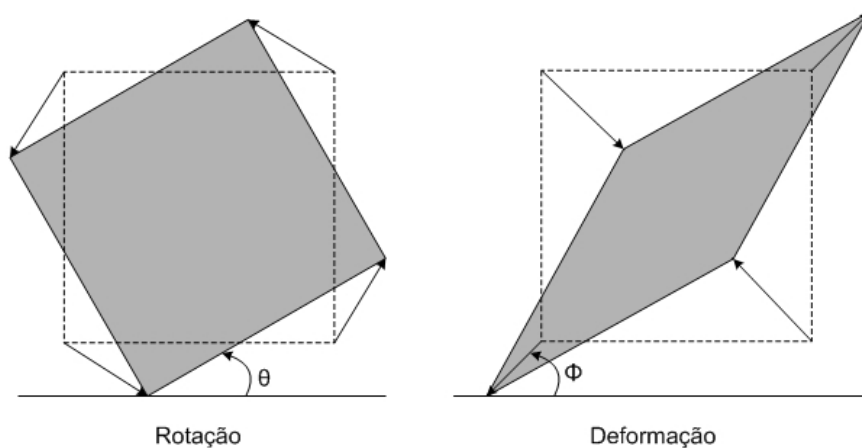


Figura 3.8: Distorções de uma Transformação *Affine*. Rotação por $R(\theta)$ e deformação por $R(-\phi)DR(\phi)$.

3.2.1.4 Transformação Projectiva 2D

A transformação projectiva é uma transformação linear de três vectores homogéneos representados por uma matriz não singular de dimensão 3x3, denominada de matriz de

homografia homogénea (H_p). Esta transformação aparece como sendo a mais importante já que de uma forma geral resume todas as apresentadas anteriormente, equação 3.27 (a transformação projectiva 2D pode ser decomposta em termos de todas as restantes transformações 2D).

$$H_p = H_E H_S H_A \quad (3.27)$$

onde H_E, H_S, H_A são respectivamente as matrizes de homografia que caracterizam as transformações euclidiana, similaridade e *affine*. Matricialmente a relação entre pontos de duas imagens dada pela transformação projectiva, representa-se da seguinte forma:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (3.28)$$

ou mais resumidamente:

$$m' = H_p m \quad (3.29)$$

Visto que o resultado da transformação H_p representa um ponto de coordenadas homogéneas, apenas importa a relação entre os elementos desta matriz. A multiplicação de um factor de escala diferente de zero por H_p não altera em nada a transformação projectiva. Assim, entre os nove elementos na matriz H_p , podem-se definir oito relações independentes, o que resulta em oito graus de liberdade. Esta transformação pode ser obtida a partir de pelo menos quatro correspondências de pontos.

3.2.2 Transformações 3D

Tal como já foi referido, as transformações 3D são muito semelhantes às transformações 2D. A maior diferença é de notar no número de graus de liberdade que como seria de esperar é superior nas transformações 3D. Também nestas, as transformações essenciais são a rotação e translação.

Uma translação 3D, na sua forma compacta, é descrita por:

$$M' = \begin{bmatrix} I & t \\ 0^T & 1 \end{bmatrix} M \quad (3.30)$$

onde I é a matriz identidade 3×3 e $t \in \mathbb{R}^3$ é o vector translação expresso no referencial de m' .

Uma rotação 3D, na sua forma compacta, é definida como:

$$M' = \begin{bmatrix} R & 0 \\ 0^T & 1 \end{bmatrix} M \quad (3.31)$$

onde R é a matriz rotação ortonormal 3×3 , com $RR^T = I$ e $|R| = 1$. Por vezes é mais conveniente descrever a transformação devida a uma rotação como sendo a diferença entre a rotação de um ponto (RM) e rotação no centro da imagem (Rc). Assim,

$$M' = R(M - c) = RM - Rc \quad (3.32)$$

onde c é o centro da imagem.

3.2.2.1 Transformação Euclideana 3D

Da mesma forma que em 2D também em 3D a transformação euclidiana é composta por uma rotação e translação. Matricialmente descrita por:

$$M' = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} M \quad (3.33)$$

Tratando-se de uma transformação 3D vamos ver o número de graus de liberdade aumentado para seis. A característica invariante nesta transformação continua a ser os comprimentos, ou seja com esta transformação os volumes vão ser mantidos.

3.2.2.2 Transformação de Similaridade 3D

Como foi visto no caso de 2D, também agora é multiplicado um factor de escala s , acrescentando assim uma alteração de escala à transformação euclidiana 3D. Esta transformação é descrita da seguinte forma:

$$M' = \begin{bmatrix} sR & t \\ 0^T & 1 \end{bmatrix} M \quad (3.34)$$

Esta transformação preserva os ângulos entre linhas e planos. Para a transformação de similaridade vão existir sete graus de liberdade, os seis da transformação anterior mais o grau adicionado pela alteração de escala.

3.2.2.3 Transformação *Affine* 3D

A transformação *affine* é representada matricialmente por:

$$M' = \begin{bmatrix} A & t \\ 0^T & 1 \end{bmatrix} M \quad (3.35)$$

onde A agora é uma matriz 3x3 invertível,

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad (3.36)$$

Também aqui se mantêm os paralelismos, mas agora entre linhas e planos. Com esta transformação conseguem-se doze graus de liberdade.

3.2.2.4 Transformação *Projectiva* 3D

A transformação projectiva 3D, tal como no caso bidimensional, é constituída pelas restantes transformações podendo ser definida como:

$$M' = H_E H_S H_A M = H_p M \quad (3.37)$$

com a matriz homogénea H_p 4x4,

$$H_p = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \\ h_{41} & h_{42} & h_{43} & h_{44} \end{bmatrix} \quad (3.38)$$

A transformação projectiva é definida por quinze graus de liberdade, sendo sete provenientes da transformação de similaridade, cinco devidos às alterações de escala provocados pela transformação *affine* e três para a parte da transformação projectiva.

3.3 Relação entre duas imagens

Como já foi visto na secção anterior existe uma relação entre um ponto no mundo (M na figura 3.1) e a sua projecção no plano da imagem (m na figura 3.1). Nesta secção vamos tratar agora da relação existente entre os pontos de duas imagens. A relação entre dois pontos de diferentes imagens que retratam o mesmo ponto do mundo é dada pelas transformações descritas anteriormente.

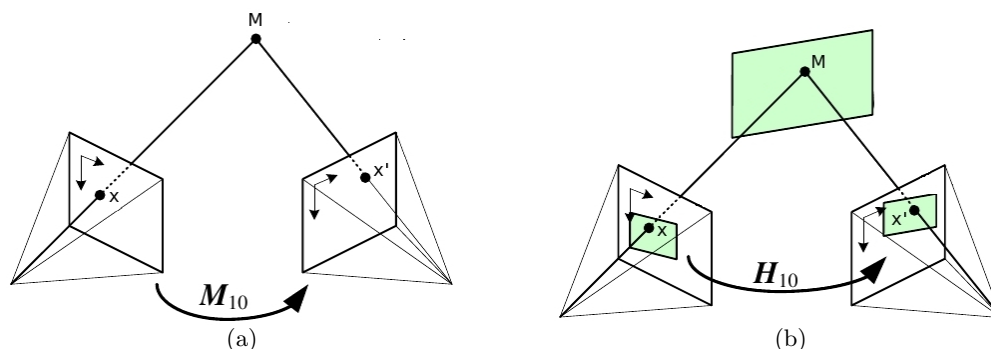


Figura 3.9: Relação entre duas imagens. a) Relação entre um ponto 3D do mundo e um ponto da imagem. b) Relação entre os pontos de duas imagens considerando o ponto do mundo num plano.

A relação dos pontos pode ser dividida em duas partes: uma em que é sabida a profundidade da imagem (figura 3.9(a)) e a segunda quando não se sabe a profundidade da imagem (figura 3.9(b)).

Assumindo que se sabe a profundidade da imagem, é possível calcular a localização do ponto no mundo usando a equação 3.5, e depois projectar este ponto na segunda imagem:

$$\begin{aligned}
 m' &\approx PM \\
 &\approx P'Pm \\
 &\approx M_{10}m
 \end{aligned}
 \tag{3.39}$$

onde P e P' são as matrizes projecção perspectiva (PPM) da primeira e segunda imagem respectivamente.

Como normalmente a profundidade de uma imagem não é conhecida, a relação entre os pontos de duas imagens é descrita de forma diferente. Pode-se então, considerar que o ponto no mundo se encontra numa cena plana e deste modo a profundidade pode ser definida como zero, o que leva a última coluna de M_{10} a ser ignorada. Podendo então

relacionar-se os pontos das duas imagens através de uma matriz de homografia (transformação).

$$m' = H_{10}m \quad (3.40)$$

Outra situação em que não é necessário saber é a profundidade acontece quando as imagens são capturadas em rotação pura ($t_0 = t_1$), neste caso podemos descrever a relação entre os pontos das duas imagens como:

$$m' \approx K_1 R_{10} K_0^{-1} m \quad (3.41)$$

onde, K_0 e K_1 são as matrizes dos parâmetros intrínsecos da primeira e segunda câmara respectivamente. Para o caso das duas imagens serem obtidas a partir da mesma câmara os parâmetros intrínsecos vão ser iguais e consequentemente $K = K_0 = K_1$.

3.4 Geometria Epipolar

A geometria epipolar existente entre duas imagens é essencialmente a geometria da intersecção dos planos de imagem com a *baseline*. A aplicação desta geometria é feita na rectificação de imagens, correspondência de pontos e determinação da orientação e posição de uma câmara [1].

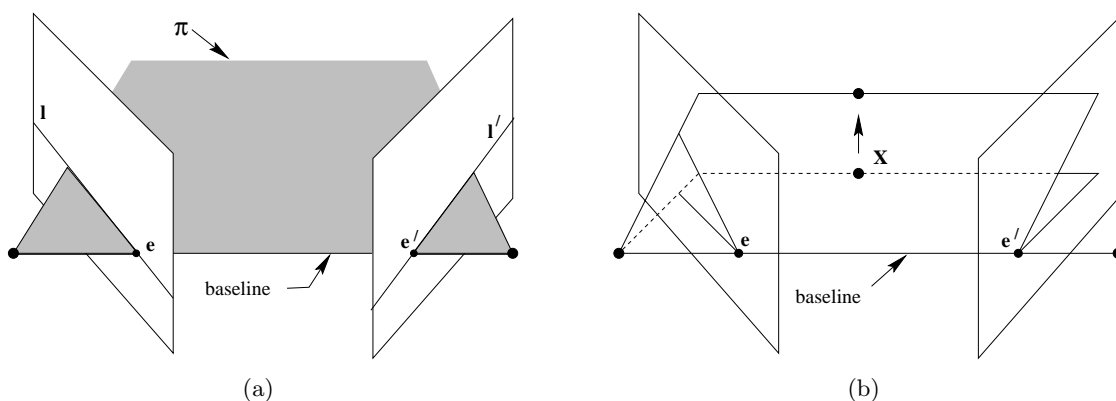


Figura 3.10: Representação da geometria Epipolar [1].

Na figura 3.10 estão representadas as componentes da geometria epipolar. A *baseline* é o eixo que une o centro das duas câmaras. Os epíolos (e e e' da figura 3.10) são os pontos de intersecção entre a *baseline* e os planos de imagem. O plano epipolar (plano π na figura 3.10(a)) é o plano que contém a *baseline* e um ponto no mundo (X). As linhas

epipolares (l e l' da figura 3.10(a)) são a intersecção dos planos epipolares com os planos de imagem. Pela figura 3.10(b), é possível verificar que com a variação das coordenadas de um ponto 3D (X), os planos epipolares para cada ponto vão ser rodados sobre a *baseline*.

Há duas formas de representar a geometria epipolar algebricamente, através da matriz fundamental ou através da matriz essencial que representa um caso particular da matriz fundamental.

3.4.1 Matriz Fundamental

A matriz fundamental descreve a geometria epipolar na sua generalidade, para casos em que as câmaras utilizadas não estão calibradas, ou seja os pontos encontram-se em coordenadas do referencial da imagem. Nesta situação a matriz fundamental é obtida da resolução da seguinte equação:

$$m'Fm = 0 \quad (3.42)$$

A determinação da matriz essencial é obtida de forma idêntica, mas como para este caso os pontos têm de se encontrar em coordenadas da câmara é necessário aplicar a matriz dos parâmetros intrínsecos (equação 3.9) aos pontos de forma a que estes fiquem em coordenadas normalizadas ($K^T m$ e $K^T m'$). Assim a matriz essencial é determinada através da seguinte relação:

$$(K^T m')^T E (K^T m) = 0 \quad (3.43)$$

A matriz essencial dispõe de cinco graus de liberdade, para a matriz de rotação e translação. Este baixo número de graus de liberdade, prende-se com o facto de se usarem coordenadas homogêneas, que provoca uma ambiguidade de escala. Esta característica permite que a matriz essencial seja decomposta em rotação e translação a menos de um factor de escala.

3.4.2 Decomposição da Matriz Essencial

A matriz essencial ganha ainda relevância devido a uma característica que lhe permite ser decomposta numa relação entre translação e rotação (equação 3.44).

$$E = tR \quad (3.44)$$

A obtenção destes dois factores, é feita através da decomposição em valores singulares (*SVD*) da matriz essencial, de onde o resultado obtido é o seguinte:

$$E = U\Sigma V^T \quad (3.45)$$

onde, $\Sigma = \text{diag}\{\sigma_1, \sigma_2, \sigma_3\}$ com $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq 0$ e $\sigma_1 \neq \sigma_2$ e $\sigma_3 \neq 0$.

Através da conjugação das três matrizes determinadas com a decomposição da matriz essencial é então possível definir duas hipóteses para a translação (equação 3.46) e duas para a rotação (equação 3.47). Pode-se concluir que irão existir quatro hipóteses para a matriz essencial, visto que para cada translação possível existem duas rotações.

Hipótese 1		Hipótese 2		Hipótese 3		Hipótese 4	
t_1	R_1	t_2	R_1	t_1	R_2	t_2	R_2

Tabela 3.2: Tabela de hipóteses de decomposição da matriz essencial.

Destas quatro apenas uma será a real, ou seja, apenas uma das hipóteses mapeia os pontos com profundidade positiva. Nas restantes hipóteses a translação e rotação vão provocar que os pontos sejam projectados atrás do plano da imagem, o que é impossível.

$$\begin{aligned} t_1 &= UZ\Sigma U^T \\ t_2 &= UZ^T\Sigma U^T \end{aligned} \quad (3.46)$$

$$\begin{aligned} R_1 &= UZU^T \\ R_2 &= UZ^T U^T \end{aligned} \quad (3.47)$$

com,

$$Z = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.48)$$

3.5 Decomposição Matriz de Homografia

Como se viu atrás na secção de transformações, uma matriz de homografia encapsula parâmetros de rotação e translação que permitem relacionar a informação entre duas imagens.

Após a determinação da matriz de homografia a sua decomposição pode ser feita em factores que permitem recolher informação sobre o movimento e estrutura (equação 3.49).

$$H = R + \frac{1}{d}tN^T \quad (3.49)$$

Na decomposição da matriz de homografia, começa-se por encontrar o vector N que indica a direcção do plano da cena (vector do plano). A matriz simétrica $H^T H$ tem três valores próprios $\sigma_1^2 \geq \sigma_2^2 \geq \sigma_3^2 \geq 0$ e $\sigma_2 = 1$. Como a matriz $H^T H$ é simétrica pode ser diagonalizada por uma matriz ortogonal V , tal que:

$$H^T H = V \Sigma V^T \quad (3.50)$$

onde, $\Sigma = \text{diag}(\sigma_1^2, \sigma_2^2, \sigma_3^2)$. Se $[v_1, v_2, v_3]$ forem os três vectores coluna de V , tem-se que:

$$\begin{aligned} H^T H v_1 &= \sigma_1^2 v_1 \\ H^T H v_2 &= v_2 \\ H^T H v_3 &= \sigma_3^2 v_3 \end{aligned} \quad (3.51)$$

Consequentemente v_2 é ortogonal ao vector N e t e o seu comprimento é mantido sobre o mapeamento de H . Pode-se então definir dois novos vectores unitários, como:

$$\begin{aligned} u_1 &= \frac{\sqrt{1 - \sigma_3^2} v_1 + \sqrt{\sigma_1^2 - 1} v_3}{\sqrt{\sigma_1^2 - \sigma_3^2}} \\ u_2 &= \frac{\sqrt{1 - \sigma_3^2} v_1 - \sqrt{\sigma_1^2 - 1} v_3}{\sqrt{\sigma_1^2 - \sigma_3^2}} \end{aligned} \quad (3.52)$$

estes dois vectores são também preservados sobre o mapeamento de H . Além disso é fácil de ver que H preserva todos os vectores unitários dentro de cada um dos seguintes sub-espacos:

$$\begin{aligned} S_1 &= \text{span}\{v_2, u_1\} \\ S_2 &= \text{span}\{v_2, u_2\} \end{aligned} \quad (3.53)$$

Uma vez que v_2 é ortogonal a u_1 e u_2 , $(v_2 \times u_1)$ é um vector unitário normal a S_1 e $(v_2 \times u_2)$ é um vector unitário normal a S_2 . O conjunto dos vectores $\{v_2, u_1, (v_2 \times u_1)\}$ e $\{v_2, u_2, (v_2 \times u_2)\}$ formam dois conjuntos de bases ortogonais para determinar a rotação.

Com isto, temos:

$$\begin{aligned} Rv_2 &= Hv_2 \\ Ru_1 &= Hu_1 \\ R(v_2 \times u_1) &= (H \times v_2)Hu_1 \end{aligned} \tag{3.54}$$

Com base nas equações anteriores podemos então definir as seguintes matrizes:

$$\begin{aligned} U_1 &= [v_2, u_1, (v_2 \times u_1)] \\ U_2 &= [v_2, u_2, (v_2 \times u_2)] \end{aligned} \tag{3.55}$$

$$\begin{aligned} W_1 &= [Hv_2, Hu_1, (H \times v_2)Hu_1] \\ W_2 &= [Hv_2, Hu_2, (H \times v_2)Hu_2] \end{aligned} \tag{3.56}$$

Temos então,

$$\begin{aligned} RU_1 &= W_1 \\ RU_2 &= W_2 \end{aligned} \tag{3.57}$$

Daqui podemos concluir que cada sub-espço (S_1 e S_2) vai contribuir com duas soluções para a decomposição da matriz de homografia (equação 3.3).

Hipótese 1	$R_1 = W_1 U_1^T$ $\vec{N}_1 = (v_2 \times u_1)$ $\frac{1}{d}t_1 = (H - R_1)N_1$	Hipótese 3	$R_3 = R_1$ $\vec{N}_3 = -\vec{N}_1$ $\frac{1}{d}t_3 = -\frac{1}{d}t_1$
Hipótese 2	$R_2 = W_2 U_2^T$ $\vec{N}_2 = (v_2 \times u_2)$ $\frac{1}{d}t_2 = (H - R_2)N_2$	Hipótese 4	$R_4 = R_2$ $\vec{N}_4 = -\vec{N}_2$ $\frac{1}{d}t_4 = -\frac{1}{d}t_2$

Tabela 3.3: Hipóteses de solução da decomposição da matriz de homografia.

De modo a reduzir o número de hipóteses fisicamente possíveis, deve ser imposta a restrição de profundidade positiva, uma vez que a câmara apenas consegue ver pontos que estão à sua frente. Para isto o terceiro elemento do vector \vec{N} terá de ser maior que zero (equação 3.58).

$$\vec{N}^T e_3 = n_3 > 0 \tag{3.58}$$

Esta página foi intencionalmente deixada em branco.

Capítulo 4

Técnica de Mosaico

Conteúdo

4.1	<i>Scale-Invariant Feature Transform (SIFT)</i>	39
4.1.1	Detecção/Localização de Pontos Chave	40
4.1.2	Orientação dos Pontos Chave	40
4.1.3	Descritor do Ponto Chave	41
4.2	Correspondência de Pontos	43
4.3	Estimação Robusta em Visão Computacional	43
4.3.1	<i>RANSAC - Random Sample Consensus</i>	44
4.3.2	Outros Algoritmos de Estimação Robusta	48
4.4	Junção de Imagens	49
4.4.1	Movimento Plano de Perspectiva	49
4.4.2	Panoramas	50
4.4.3	Coordenadas Esféricas e Cilíndricas	51

A técnica de mosaico não é mais que um conjunto de etapas para a obtenção de um mapa para inspecção e/ou navegação. A solução proposta para a criação destes mapas, baseados na junção de várias imagens, passa pelas seguintes etapas:

- detecção de pontos chave/interesse;
- correspondência entre pontos de interesse de duas imagens;

- estimação de uma transformação que relacione a informação de duas imagens;
- junção das imagens;

A detecção e correspondência de características ou pontos chave/interesse em imagens é uma componente essencial em muitas aplicações de visão computacional, tais como junção e alinhamento de imagens, estabilização de vídeo ou estimação da posição da câmara. Através da correspondência é possível saber quais os pontos de duas imagens que são a projecção da mesma cena ou objecto [33].

Existem pelo menos duas formas possíveis de abordar o problema de detecção de pontos chave ou características e as suas correspondências. A primeira é com recurso a técnicas de correlação e mínimos quadrados. Com esta técnica é possível por exemplo detectar os cantos de portas, janelas, telhado ou a linha entre as montanhas e céu na figura 4.1. A segunda é detectar as características independentemente em cada imagem (ex.: *SIFT* e *KLT*), como por exemplo zonas com grandes diferenças de contraste (gradiente), como é o caso de sombras na neve ou diferença de contraste entre uma zona de montanha e o céu (figura 4.1). Esta segunda forma, demonstra ser a mais indicada para aplicações onde exista um grande movimento ou mudança de aparência entre imagens (ex.: criação de mosaico de imagens) [3]. Com ambos os métodos a correspondência é feita com base na orientação e aparência local dos pontos de interesse.



Figura 4.1: Dois pares de imagens para correspondência de pontos de interesse [3].

Apesar de os algoritmos de detecção/localização serem bastante eficazes, nem sempre é possível fazer uma correspondência correcta entre os pontos de interesse de duas imagens, pelo que para aplicações como criação de um mosaico a partir de imagens é necessário

recorrer a estimação robusta. Os algoritmos de estimação robusta permitem fazer uma análise dos dados, descartando as falsas correspondências com grande eficiência. Apesar da grande eficiência estes algoritmos demonstram desvantagens a nível computacional. Esta desvantagem pode ser ultrapassada com a utilização de informação de um sistema inercial que permita encontrar a relação entre as duas imagens e excluir os dados que não sejam coerentes com essa relação.

Através da correcta correspondência de pontos de interesse de duas imagens consecutivas é possível fazê-las coincidir, pela área comum da mesma cena, alinhando as imagens através da utilização de geometria aplicada a visão.

Após a análise anterior, com base nos projectos existentes de quadrotors, sistemas visão neles implementados e técnicas de criação de mapas, é fácil delinear as etapas necessárias para a obtenção de um mosaico de imagens para inspecção. Deste modo as fases para a criação de um mapa de imagens podem ser as seguintes: detecção e correspondência de pontos de interesse, exclusão de falsas correspondências e por fim a união das imagens. A solução aqui apresentada tem: como detector de pontos de interesse o algoritmo *SIFT*; para exclusão de falsas correspondências a estimação robusta com *RANSAC*; para junção de imagens são usadas transformações bidimensionais que permitam relacionar duas imagens.

4.1 *Scale-Invariant Feature Transform (SIFT)*

O algoritmo *SIFT* (*Scale-Invariant Feature Transform*) resolve muitos dos problemas de detecção de pontos chave pois permite detectar com eficácia características distintas de imagens, como é o caso de pontos que são invariantes com a escala e rotação da imagem e parcialmente invariantes a variações na iluminação e adição de ruído [33].

Este método é executado em duas fases: a primeira para detecção de características/pontos chave na imagem, na segunda são gerados os descritores dos pontos chave. Pode então dividir-se o algoritmo *SIFT* em três etapas:

- Detecção/Localização de pontos chave
- Atribuição da orientação
- Descritor do ponto chave

A primeira fase corresponde à detecção de pontos característicos/chave na imagem sobre todas as escalas e as duas últimas etapas à fase de caracterização destes pontos.

4.1.1 Detecção/Localização de Pontos Chave

O primeiro passo da detecção de pontos chave é identificar as localizações e escalas dos pontos da mesma cena ou objecto que possam ser encontrados em diferentes imagens.

Detectar localizações que são invariantes à escala da imagem pode ser conseguido através da procura de características estáveis sobre todas as escalas, utilizando uma função espaço de escala [34]. Koenderik e Lindeberg ([35, 36]) mostraram que segundo uma variedade de pressupostos razoáveis a única possibilidade de um *kernel* espaço de escala é uma função gaussiana. Assim, o espaço de escala de uma imagem é definido como uma função (L), que é definida como a convolução de uma função gaussiana (G), com uma imagem (I):

$$L(x, y, \delta) = G(x, y, \delta) * I(x, y) \quad (4.1)$$

onde,

$$G(x, y, \delta) = \frac{1}{2\pi\delta^2} e^{-\frac{x^2+y^2}{2\delta^2}} \quad (4.2)$$

Para detectar eficazmente a localização dos pontos chave no espaço de escala, *David Lowe* [33] propõe a utilização da diferença de duas escalas separadas por um factor de multiplicação k , obtendo-se assim uma função de diferença de gaussiana, a convolver com a imagem:

$$\begin{aligned} D(x, y, \delta) &= (G(x, y, k\delta) - G(x, y, \delta)) * I(x, y) \\ &= L(x, y, k\delta) - L(x, y, \delta) \end{aligned} \quad (4.3)$$

4.1.2 Orientação dos Pontos Chave

É importante caracterizar os pontos chave quanto à sua orientação, para os tornar invariantes em relação à rotação da imagem. Esta característica é fundamental para efectuar uma correcta correspondência entre pontos de duas imagens que retratam uma cena de diferentes ângulos de visão.

A fim de determinar a orientação dos pontos chave de uma imagem, é usada a escala do ponto para seleccionar a imagem gaussiana (L) na escala mais próxima. Para cada uma

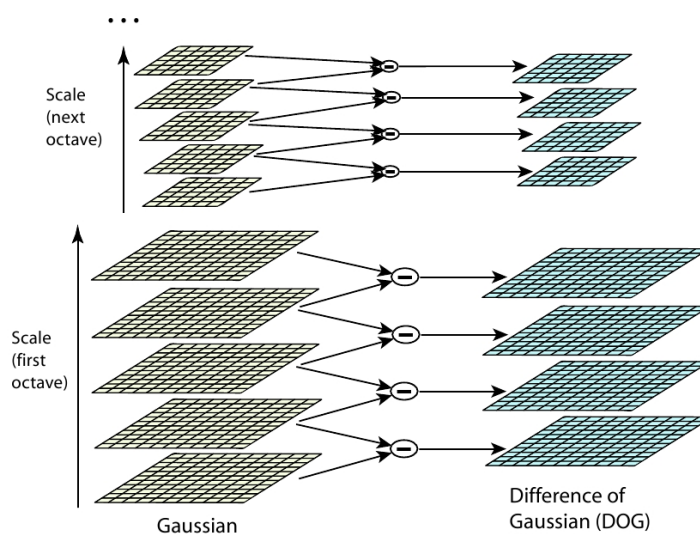


Figura 4.2: Esquema de procura da localização de pontos chave.

destas amostras da imagem, é calculada a magnitude (4.4) e orientação (4.5) do gradiente, através da diferença de pixels numa janela de 16x16 pixels em torno do ponto de interesse.

$$m(x, y) = \sqrt{[L(x + 1, y) - L(x - 1, y)]^2 + [L(x, y + 1) - L(x, y - 1)]^2} \quad (4.4)$$

$$\theta(x, y) = \arctan \left(\frac{L(x, y + 1) - L(x, y - 1)}{L(x + 1, y) - L(x - 1, y)} \right) \quad (4.5)$$

Com estes valores é construído um histograma de orientações com base na magnitude dos gradientes que estão compreendidos numa janela gaussiana de 1.5δ que define a região de interesse (linha azul na figura 4.3). Apenas são usados os valores da região de interesse por forma a diminuir a influência dos gradientes afastados do centro (localização do ponto). Antes de determinar o máximo do histograma, este é suavizado através de um filtro mediano. O pico mais elevado do histograma é detectado como a orientação do ponto chave. No entanto mais pontos chave podem ser gerados nesta localização, é o caso de existir uma ou mais orientações no histograma em que a sua magnitude é superior a 80% do pico máximo.

4.1.3 Descritor do Ponto Chave

As fases de localização e atribuição de orientação permitem caracterizar cada ponto de interesse com uma localização, uma escala e uma orientação. A etapa do descritor vai permitir caracterizar a imagem na região do ponto de interesse (mesma janela 16x16 da

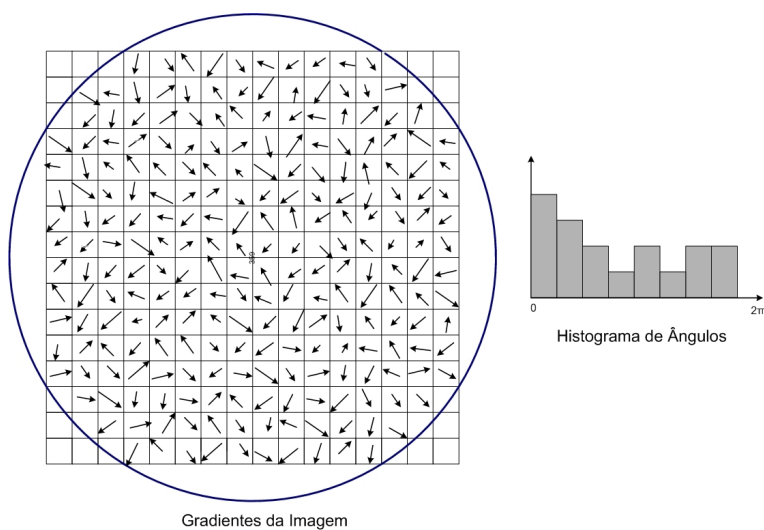


Figura 4.3: Exemplo gráfico da janela de interesse da imagem, e respectivos gradientes, e histograma de ângulos em função dos gradientes.

determinação da orientação), que é altamente distintiva e o mais invariante possível, a variações de luminosidade e diferentes ângulos de visão.

Para cada sub-região de tamanho 4x4 pixels, é determinado um histograma de orientações com oito direcções, com base nos gradientes de cada pixel. O descriptor é então caracterizado por estas oito direcções, em que cada seta representa a magnitude do gradiente na sua direcção. Obtendo-se assim o descriptor na forma de um vector com 128 valores não negativos correspondentes à magnitude de cada direcção.

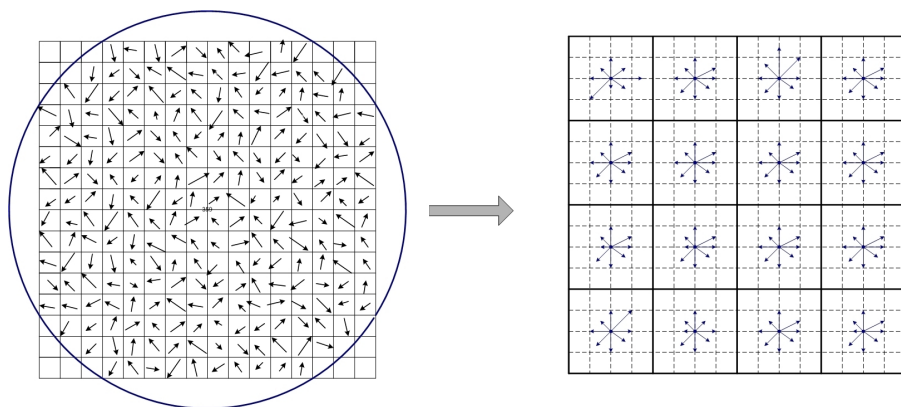


Figura 4.4: Exemplo gráfico de um descriptor do algoritmo *SIFT*.

4.2 Correspondência de Pontos

Uma vez extraídos das imagens os pontos de interesse e devidamente caracterizados pelo seu descritor, o próximo passo é estabelecer uma primeira correspondência entre estes pontos. A aproximação que se tem na correspondência de pontos depende da aplicação em que vai ser usada, por exemplo, se for para estabelecer a correspondência entre pontos de duas imagens sobrepostas, já se sabe à partida que existirão pontos de interesse comuns; no entanto se for usada para reconhecimento de objectos numa imagem não temos a garantia de que vão existir correspondências.

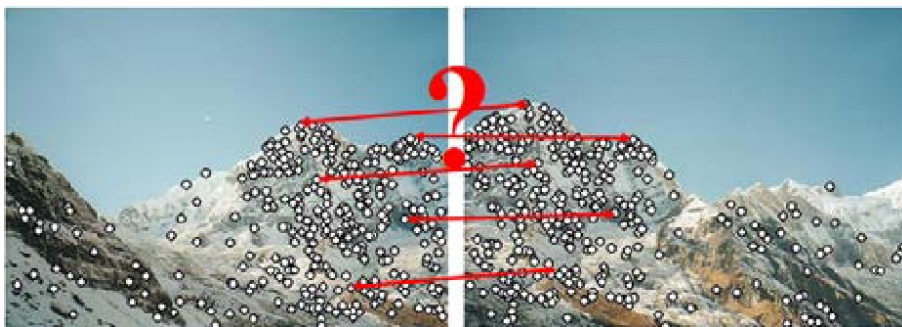


Figura 4.5: Problema da correspondência de Pontos.

Então para esta primeira correspondência é feita a comparação dos descritores dos pontos de interesse de duas imagens. Assim, um descritor $Desc$ corresponde a um descritor $Desc'$, na segunda imagem, se a diferença entre eles multiplicada por um *threshold* não for superior à distância do $Desc$ a qualquer outro descritor da segunda imagem $Desc_n$. No caso de dois ou mais descritores serem aceites como correspondência de um descritor apenas é aceite o melhor, ou seja o descritor com o valor de distância mais baixo.

4.3 Estimação Robusta em Visão Computacional

Uma das tarefas chave no campo da visão computacional é estabelecer uma relação entre a informação extraída de imagens e um modelo matemático. A relação entre os pontos correspondentes de duas imagens pode ser considerada de dois tipos: falsos e verdadeiros. As correspondências verdadeiras referem-se a informação viável e que irá gerar bons resultados na sua utilização. As falsas correspondências serão todos aqueles dados que induzirão erro na estimação, por exemplo dados que contêm ruído. Estas dificuldades são ultrapassadas com a utilização de algoritmos de estimação robusta [37],

que permitem excluir as falsas correspondências, ou através do uso de informação de sensores presentes no veículo (ex.: *INS*) e que permitam a validação das correspondências entre as imagens (capítulo 5).

Na descrição de estimação robusta importa definir claramente o significado de robustez. Na estimação um algoritmo é definido como robusto através do seu ponto de quebra. Este ponto é dado pela percentagem de falsas correspondências entre duas imagens e que causam uma divergência na estimação para longe do valor real. Em teoria o valor de quebra máximo é de 50%, limite para algoritmos como por exemplo *LMS* e *M-estimators*, podendo atingir em prática valores superiores a 50% através de algoritmos como *RANSAC* e transformada de *Hough* [38].

4.3.1 *RANSAC - Random Sample Consensus*

O algoritmo, *RANSAC* (*Random Sample Consensus*) foi introduzido pela primeira vez em 1981 por *Fischler* e *Bolles* [39] como sendo um método para a estimação de parâmetros de um certo modelo (linha recta, plano, homografia ou pontos relacionados por uma rotação, translação e escala), partindo de um conjunto de dados contaminados por uma grande quantidade de dados falsos (normalmente mais de 50%).

O algoritmo *RANSAC* segue a seguinte estratégia: os dados obtidos (D) têm um total de N elementos, com uma percentagem desconhecida de dados falsos. Para estimar os parâmetros do modelo pretendido, tem de se separar a informação em falsa e verdadeira e fazer a estimação com base apenas nos dados verdadeiros. Para seguir a estratégia definida, o algoritmo *RANSAC* é composto por dois passos: hipótese e teste/verificação, repetidos iterativamente. No passo de hipótese são geradas soluções para os parâmetros do modelo pretendido, a partir do *MSS* (*minimal sample set*). Na fase de teste/verificação é cotado um conjunto *CS* (*consensus set*), originado de um *MSS*, de acordo com o número de elementos do conjunto de dados inicial (D) que são consistentes com os parâmetros estimados do modelo pretendido.

Um exemplo simples da aplicação do algoritmo *RANSAC*, é a sua utilização para a estimação de uma linha recta com base num conjunto de pontos 2D. Deste conjunto são seleccionados aleatoriamente dois pontos (*MSS*) que definem o modelo de uma linha recta. Os pontos do conjunto de dados próximos desta linha segundo um *threshold* é o *CS*. O *consensus set* que consiga um maior número dados, melhor correspondência dos dados face

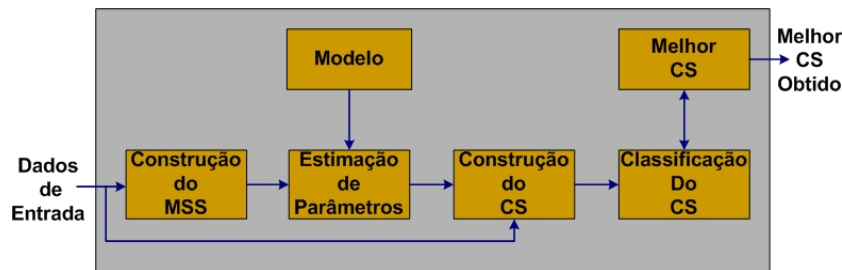


Figura 4.6: Representação ilustrativa do algoritmo *RANSAC*.

à linha definida, é aceite como a melhor estimacão (linha verde representada na figura 4.7).

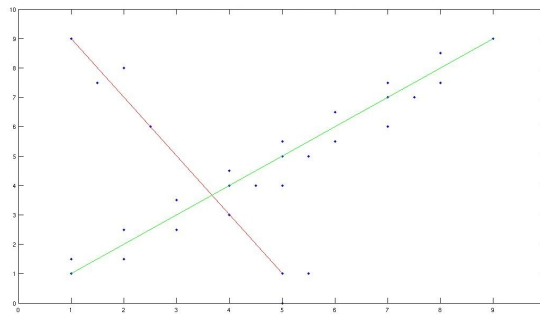


Figura 4.7: Exemplo Algoritmo *RANSAC* aplicado à estimacão de uma linha recta. Linha verde melhor estimacão para o modelo de uma linha recta. Linha vermelha, hipótese de estimacão rejeitada.

4.3.1.1 Construcão do *MSS* - *Minimal Sample Set*

Se os pontos do conjunto de dados de entrada (D) estiverem livres de ruído, então qualquer conjunto *MSS* (s) é composto apenas por dados verdadeiros, pelo que os parâmetros do modelo vão ser bem estimados. Assim, no primeiro passo do algoritmo *RANSAC*, é seleccionado aleatoriamente um conjunto s , a partir do conjunto dos dados de entrada D , de onde são estimados os parâmetros do modelo (θ), com base apenas nos dados do conjunto s .

4.3.1.2 Número de Iterações

Não é necessário tentar arranjar todas as combinações de conjuntos *MSS* (s) possíveis, o que também é computacionalmente impensável. Assim sendo, o número de iterações (h)

deverá ser suficientemente grande, para assegurar que com uma probabilidade p pelo menos um dos conjuntos s está livre de dados falsos.

Se p for a probabilidade de obter correctamente os parâmetros de um modelo, a partir de um conjunto s obtido dos dados de entrada D . Consequentemente, a probabilidade de achar um s com pelo menos um dado falso é:

$$1 - p \tag{4.6}$$

Se agora construirmos h diferentes de conjuntos s , então a probabilidade de todos estarem contaminados por dados falsos é dada por:

$$(1 - p)^h \tag{4.7}$$

Este valor tende para zero quando h crescer para infinito, ou seja, mais cedo ou mais acabaremos por encontrar um bom conjunto s . O número de iterações desejado terá de ser grande o suficiente para que a probabilidade na equação 4.7, esteja abaixo de um certo *threshold* (δ).

$$(1 - p)^h \leq \delta \tag{4.8}$$

A relação anterior pode ser invertida a fim de determinar o número de iterações, podendo escrever-se da seguinte forma:

$$h \geq \frac{\log \delta}{\log(1 - p)} \tag{4.9}$$

4.3.1.3 Cotação de um *CS* - *Consensus Set*

Como já foi mencionado cada *MSS* dá origem a um *CS*, este conjunto tem de ser cotado para avaliar o melhor conjunto *CS*. O algoritmo *RANSAC* na sua formulação original [39], avalia o conjunto *CS* com base no número de elementos presentes no conjunto (cardinalidade), ou seja, um conjunto *CS* com mais elementos era classificado melhor que um que tivesse contido poucos elementos.

Deste modo, o algoritmo *RANSAC* pode ser visto como um algoritmo de optimização,

que minimiza a função de custo mínimo C , definida como:

$$C = \sum_i p(e_i^2) \quad (4.10)$$

onde e^2 é a discrepância entre um ponto do conjunto de entrada (D) e os parâmetros estimados, e $p(e^2)$ pode assumir os seguintes valores:

$$p(e^2) = \begin{cases} 0 & e^2 < \delta^2 \\ \text{constante} & e^2 \geq \delta^2 \end{cases} \quad (4.11)$$

Ou seja, dados verdadeiros contam zero para a avaliação e os dados falsos contam uma penalização constante. Assim quanto maior o *threshold* (δ), maior será o número de soluções (CS) com o valor da função de mínimo custo (C) igual, fazendo com que a estimação fique má. No caso de o *threshold* ser um valor demasiado pequeno os parâmetros estimados tendem a ser instáveis. Por exemplo, se um *threshold* for suficientemente grande, então todas as soluções (CS) vão ter custos iguais, isto é, todos os dados vão ser aceites como verdadeiros. Definida desta maneira a cotação do conjunto CS pode ser levada para uma abordagem baseada em M -estimators, tornando a função de probabilidade 4.11 mais sensível.

$$p(e^2) = \begin{cases} e^2 & e^2 < \delta^2 \\ \delta^2 & e^2 \geq \delta^2 \end{cases} \quad (4.12)$$

Pode-se verificar que da mesma maneira que em 4.11 dados falsos contribuem com um valor constante também em 4.12 isso acontece, a diferença está no facto de agora informação verdadeira ser pesada pela forma como se ajusta aos dados. Torr et al. referem-se a esta modificação do algoritmo *RANSAC* original, com o nome de *MSAC* [40].

4.3.1.4 *Threshold*

A distância de *threshold* (δ), é uma distância que deverá ser tal que com uma probabilidade α um ponto do conjunto de dados de entrada D é considerado como dado verdadeiro. Por exemplo, para um $\alpha = 0.95$ existe 95% de probabilidade de um ponto ser considerado verdadeiro. Para o cálculo desta distância é preciso saber a distribuição de probabilidade para considerar que um dado é verdadeiro, para um determinado modelo, o que nem sempre é possível obter de forma trivial. Na prática o valor de *threshold* é

escolhido empiricamente. Assim, um ponto é considerado verdadeiro ou falso se a sua distância for inferior ou superior ao *threshold*, respectivamente.

$$\begin{cases} \textit{verdadeiro} & d^2 < \delta^2 \\ \textit{falso} & d^2 \geq \delta^2 \end{cases} \quad (4.13)$$

4.3.1.5 RANSAC Aplicado a Homografia

Dado um conjunto de correspondências entre duas imagens, através do algoritmo *RANSAC* é possível estimar os parâmetros de uma homografia, e daí descartar falsas correspondências. Uma homografia é caracterizada por ter nove termos (exemplo de uma matriz de transformação 2D), mas apenas oito graus de liberdade, uma vez que escala é imaterial. Estes parâmetros são estimados através do algoritmo *nDLT* (*Normalized Direct Linear Transform*). Tratando-se de uma homografia os parâmetros têm de ser estimados a partir de um *MSS* com cardinalidade de pelo menos quatro. Ou seja, quatro pontos de cada imagem são necessários para definir uma homografia.

$$MSS = \left\{ \left[\begin{array}{c} x_1 \\ x'_1 \end{array} \right], \dots, \left[\begin{array}{c} x_4 \\ x'_4 \end{array} \right] \right\} \quad (4.14)$$

4.3.2 Outros Algoritmos de Estimação Robusta

Partindo deste algoritmo como base, através de pequenas modificações, é possível obter outras técnicas de estimação como é o caso do *MLE SAC* (*Maximum Likelihood Estimation SAmple and Consensus*), *PROSAC* (*PROgressive SAmple Consensus*), *R-RANSAC* (*Randomized RANdom SAmple Consensus*), *P-RANSAC* (*Preemptive RANdom SAmple Consensus*) e *LMS* (*Least Median of Squares*).

O algoritmo *MLE SAC* tal como o *MSAC*, permite compensar a sensibilidade do *RANSAC* melhorando a qualidade do *CS*, através do cálculo da sua probabilidade [40]. Em [41], é proposto o método *PROSAC*, no qual a modificação para o *RANSAC* fica-se com o facto de se esperar à partida que alguma informação seja conhecida, podendo assim mais facilmente definir os dados como falsos ou verdadeiros. Uma versão aleatória do *RANSAC*, chamada *R-RANSAC* é proposta em [42], a ideia principal desta variante é reduzir a carga computacional necessária para encontrar um bom *consensus set* (*CS*), para isso é utilizado apenas um conjunto de dados de entrada, em vez do conjunto inteiro. Por fim, o

P-RANSAC permite estimativa robusta em tempo real. A ideia central desta modificação do *RANSAC*, consiste na geração de um número fixo de hipóteses que são comparadas no que diz respeito à qualidade da hipótese gerada.

No algoritmo *RANSAC* um modelo instânciado por um pequeno conjunto de dados é avaliado pelo número de dados contidos num intervalo de distância. Uma alternativa é avaliar o modelo pela média das distâncias a todos os pontos, sendo seleccionado o modelo com a média mais baixa. O *LMS* tem vantagem sobre o *RANSAC* na medida em que não é necessário definir um *threshold*, nem é preciso ter conhecimento prévio da variação do erro. A desvantagem prende-se com o facto deste algoritmo falhar para um número de dados falsos superior ao número de dados verdadeiros [1].

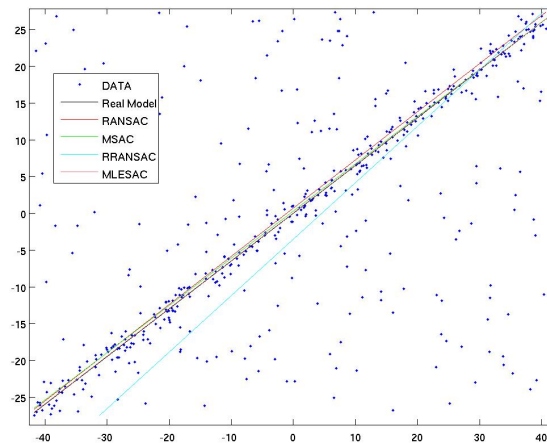


Figura 4.8: Comparação de alguns algoritmos de estimação robusta.

4.4 Junção de Imagens

Através das transformações geométricas descritas no capítulo anterior, é possível efectuar o alinhamento das imagens capturadas de diferentes ângulos de visão. Para juntar estas imagens é preciso agora fêzê-las coincidir, pelas áreas das imagens que são a representação da mesma cena física ou objecto.

4.4.1 Movimento Plano de Perspectiva

O modelo de movimento plano de perspectiva é o mais simples de utilizar com imagens pois neste apenas é aplicada uma transformação 2D simples (rotação e translação) nas

imagens de forma a fazê-las coincidir.

A partir da correspondência de pontos entre duas imagens são determinados os parâmetros de rotação (R) e translação (t) de uma transformação 2D a aplicar. Com estes parâmetros determinados as imagens estão agora em condições de serem alinhadas e fundidas. Este passo é feito através da aplicação da transformação a uma das imagens, mantendo a outra imagem como base.

As transformações bidimensionais que se adequam a este método de união de imagens são: a transformação euclidiana e a transformação de similaridade.



Figura 4.9: Exemplo do movimento de perspectiva plano, para a junção de duas imagens [3].

Um exemplo prático deste movimento, é o que fazemos se tentarmos juntar algumas fotografias impressas ou um documento digitalizado. A rotação e translação são também normalmente modelos adequados para compensar pequenos movimentos da câmara como em aplicações de estabilização de vídeo ou mosaico de imagens.

4.4.2 Panoramas

Também para a obtenção de mapas de inspecção é importante o conceito de panorama. Estes permitem recriar o cenário com imagens obtidas em torno de um ponto, ou seja, quando existe um caso de rotação pura. Isto pode acontecer por dois motivos: o centro da câmara é fixo e todas as imagens são capturadas efectuando apenas uma rotação da câmara; ou quando são capturadas imagens com a geometria da cena longe da câmara, nesta situação os pontos na imagem consideram-se no infinito pelo que a translação pode ser desprezada, restando apenas o factor de rotação.

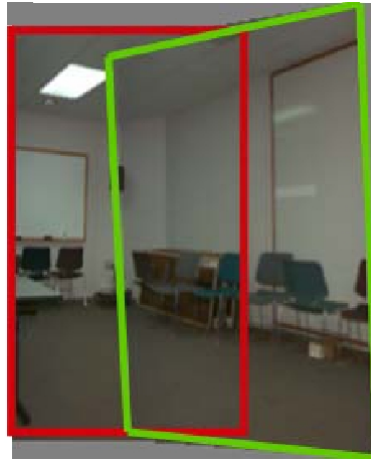


Figura 4.10: Exemplo de panorama para a criação de mapas para inspecção [3].

Na criação de mapas com o conceito de panorama, a correspondência entre pontos de interesse em duas imagens é usada para determinar os parâmetros de uma transformação 2D *affine* ou projectiva. Sendo posteriormente aplicada a transformação a uma das imagens usando a segunda como base, tal como o descrito para o caso anterior.

4.4.3 Coordenadas Esféricas e Cilíndricas

A transformação para coordenadas esféricas e cilíndricas é uma alternativa ao uso de homografias (transformações) para o alinhamento e união das imagens, mas pouco utilizada. O objectivo deste método consiste em deformar as imagens de forma a fazer com que estas fiquem em coordenadas cilíndricas. Com isto será apenas necessário utilizar a translação entre as imagens para as alinhar. Infelizmente este método apenas pode ser utilizado se for conhecido o ângulo entre imagens.

Esta página foi intencionalmente deixada em branco.

Capítulo 5

Fundamentos de Navegação

Conteúdo

5.1	Técnicas de Navegação	54
5.1.1	Sistemas Baseados em Pontos de Referência	54
5.1.2	Sistemas “ <i>Dead Reckoning</i> ”	54
5.2	Referenciais de Coordenadas	56
5.2.1	Referencial do Corpo	56
5.2.2	Referencial <i>NED</i>	57
5.3	Atitude do Veículo	57
5.3.1	Rotação em <i>Roll</i> (ϕ)	58
5.3.2	Rotação em <i>Pitch</i> (θ)	58
5.3.3	Rotação em <i>Yaw</i> (ψ)	58
5.4	Transformação de Referenciais	58

É possível definir navegação como sendo o processo de planejar, identificar e controlar o movimento do veículo, seja este terrestre, aéreo ou marítimo, desde o seu ponto de partida até ao seu destino.

Neste capítulo são apresentados alguns conceitos de navegação necessários para a realização deste trabalho. Destas noções destaca-se a importância da relação entre diferentes referenciais, aos quais os dados são obtidos.

5.1 Técnicas de Navegação

Em termos de técnicas de navegação pode-se considerar que estas se dividem em dois grupos: sistemas baseados em pontos de referência e sistemas “*dead reckoning*”.

5.1.1 Sistemas Baseados em Pontos de Referência

Os sistemas baseados em pontos de referência são os mais utilizados para a navegação de veículos autónomos, sejam eles marítimos, aéreos ou terrestres. Estes sistemas proporcionam informação relativa à posição e orientação, bastante precisa com um processamento mínimo.

Os sistemas baseados em pontos de referência utilizam sensores de: ultra-sons, rádio frequência ou fontes luminosas (por exemplo: *GPS*, sistema de visão, sonar, laser), por forma a determinar a posição e orientação do veículo relativamente a um ponto de referência. A posição deste ponto no mundo pode ser conhecida tratando-se então de um sistema de navegação *active beacon*, neste método o cálculo da posição é feito utilizando conceitos de trilateração e triangulação. No caso, desta posição não ser conhecida no mundo, mas sim de forma relativa estamos perante um sistema de navegação “*dead reckoning*”, em que a informação é referenciada em relação a um referencial local.

Em ambos os métodos apresentados atrás está muitas vezes associada a sua aplicação para a criação de mapas para navegação. Estes mapas utilizam a informação de posição e orientação do veículo, e dos pontos de referência, para representar o mundo e atitude do veículo nesse local. A técnica de *SLAM* (*Simultaneous Localization and Mapping*) permite a integração de todas estas características de navegação. Com o *SLAM* é possível localizar e gerar o mapa do mundo envolvente do robot em simultâneo, permitindo ao robot navegar nesse ambiente.

5.1.2 Sistemas “*Dead Reckoning*”

“*Dead Reckoning*” é uma técnica de cálculo da posição a partir de medições de velocidade [43]. Isto significa que a navegação é feita na ausência de posições fixas e consiste em calcular a posição do veículo através da integração da velocidade (estimada ou medida directamente), assim as posições obtidas são sempre em relação à posição inicial que é conhecida. Este tipo de navegação é feito com base em informações de sensores como por exemplo: encoders, tacómetros, bússola ou *IMU* (*Inercial Measurement Unit*).

As vantagens destes sistemas prendem-se com o facto dos sensores utilizados estarem contidos no veículo e não necessitarem de nenhum apoio exterior (ao contrário dos sistemas *active beacon*) e possibilitarem navegação de baixo custo. Como desvantagem, nestes sistemas as medidas vão-se degradando ao longo do tempo devido à acumulação de erros intrínsecos dos sensores.

5.1.2.1 Sistema de Navegação Inercial - *INS*

Um sistema de navegação inercial (*INS*) não é mais que um sistema de navegação *dead reckoning* que fornece informação dinâmica sobre a posição e velocidade do veículo, tendo como base medições obtidas de um *IMU* (*Inertial Measurement Unit*). Um *IMU* é constituído por três acelerómetros e três giroscópios montados numa tríade ortogonal, podendo ainda em alguns casos dispor de um magnetómetro.

O princípio básico de operação da navegação inercial é baseado na lei de movimento de *Newton*, que diz que um objecto se mantém num estado de repouso ou movimento uniforme, até ser perturbado por uma força exterior. A aplicação de uma força exterior gera a aceleração, que é medida pelos acelerómetros presentes no *IMU*. Esta aceleração quando integrada duas vezes dá a informação da alteração da posição do veículo em relação às condições iniciais. Deste modo, o *INS* fornece informação relativa à alteração do estado do veículo, pelo que as condições iniciais (posição, velocidade e atitude) do veículo têm de estar bem definidas para uma correcta navegação.

O processo de cálculo dos valores de posição, velocidade e atitude, nem sempre é muito simples devido ao facto de que o *IMU* fornece as medidas relativas a um dado referencial que muitas das vezes é diferente do referencial em que a posição e velocidade estão expressas. Por esta razão o *INS* contém giroscópios, que servem para fornecer medições angulares, para que seja possível transformar a aceleração medida para o referencial adequado, antes da integração. Assim a precisão dos giroscópios é fundamental para um *INS* porque qualquer erro na transformação da aceleração irá provocar um erro no cálculo da posição e velocidade do veículo.

O desempenho do sistema inercial, tanto dos giroscópios como dos acelerómetros, é afectado por uma variedade de erros. A maioria dos erros pode ser classificada no *bias*, factor de escala, desalinhamento de eixos e ruído. Destes o *bias drift* é aquele que apresenta maior interesse para este trabalho. O *bias* é uma componente presente quer nos giroscópios

quer nos acelerómetros e que consiste em duas partes: uma determinista chamada de *bias offset* e uma parte aleatória chamado de *bias drift*. O *bias offset* refere-se ao *offset* presente nas medidas do sistema inercial, este *offset* pode ser determinado através de uma calibração do sistema inercial. Já o *bias drift* refere-se à taxa com que o erro no sensor inercial se acumula com o tempo.

5.2 Referenciais de Coordenadas

Um referencial é um sistema de coordenadas. Para a navegação é necessário utilizar pelo menos dois referenciais. Um para a representação do corpo/inercial e outro para representação da navegação (mapa).

Alguns dos referenciais mais utilizados em sistemas de navegação são:

- Referencial *ECEF* (*Earth Centered Earth Fixed*)
- Referencial *WGS-84* (*World Geodetic System 1984*)
- Referencial *NED* (*North East Down*)
- Referencial do Corpo (*Body Frame*)
- Referencial *ECI* (*Earth Centered Inercial*)

Nesta secção apenas serão focados os referenciais de interesse para este trabalho (corpo e *END*).

5.2.1 Referencial do Corpo

O referencial do corpo neste caso, contém os seus eixos coincidentes com os do veículo, podendo estes ter sido arbitrados de outra forma. Sendo este o referencial base para os sensor inercial, estará coincidente com o referencial da aeronave. Desta forma, o eixo dos xx' aponta para a frente da aeronave (eixo de *roll*), o eixo de yy' aponta para a direita da aeronave (eixo de *pitch*) e por fim o eixo de zz' aponta para baixo (eixo de *yaw*), completando um sistema de coordenadas ortogonal. A origem deste referencial é o centro de massa do quadrotor.

5.2.2 Referencial *NED*

O referencial *NED* (*North East Down*), também conhecido como referencial local, é usado para navegação. A utilização deste referencial faz-se para navegação próxima da sua origem. Este é constituído por três vectores: *N* (*north*), *E* (*east*) e *D* (*down*). O vector *N*, aponta para o pólo magnético norte da terra. O vector *E*, aponta para o Este geodésico. O último, vector *D*, aponta no sentido do vector de gravidade local. O conjunto destes três vectores forma um sistema de coordenadas ortogonal.

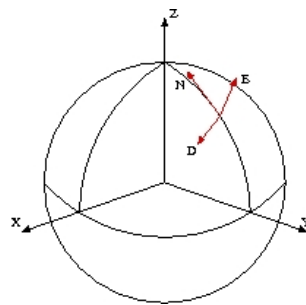


Figura 5.1: Representação do referencial *NED*.

5.3 Atitude do Veículo

Em navegação é necessário transformar eficazmente coordenadas entre diferentes referenciais, porque para a fusão da informação dos diversos sistemas de navegação, estes têm de estar no mesmo referencial.

A atitude de uma aeronave no espaço (referencial do corpo), pode ser descrita através de três rotações (*roll*, *pitch* e *yaw*), normalmente denominadas de ângulos de *Euler*. Os ângulos de *Euler* estão associados ao referencial do corpo pelo que para serem usados na navegação ou em aplicações de visão, terão de ser transformados para um referencial diferente.

Os ângulos de Euler são definidos da seguinte maneira:

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} roll \\ pitch \\ yaw \end{bmatrix}$$

5.3.1 Rotação em *Roll* (ϕ)

O ângulo de roll, indica a inclinação lateral do quadrotor, rotação em torno do eixo dos xx' . Quando o ângulo de roll é $\phi = 0^\circ$, o eixo dos xx' do referencial do corpo encontra-se numa posição horizontal. Quando este é positivo, provoca uma rotação no sentido dos ponteiros do relógio.

$$R_X(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}$$

5.3.2 Rotação em *Pitch* (θ)

Em aeronáutica o ângulo de pitch indica se o nariz da aeronave (frente da aeronave), está a apontar para cima ou para baixo. Este é o ângulo de rotação sobre o eixo dos yy' , o que significa que para $\theta = 0^\circ$ o veículo encontra-se em posição horizontal.

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

5.3.3 Rotação em *Yaw* (ψ)

O ângulo de yaw, representa uma rotação da aeronave para a esquerda ou direita. Ou seja, uma rotação sobre o vector de gravidade (eixo dos zz'). Este ângulo tem valor nulo, $\psi = 0^\circ$ quando o veículo se encontra alinhado (eixo dos xx') com o vector norte do referencial de navegação.

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

5.4 Transformação de Referenciais

Uma vez que os vários sensores utilizados em sistemas de navegação poderão fornecer resultados em referenciais diferentes, será necessária a aplicação de métodos de transformação de coordenadas de modo a que todas as medidas se refiram a um referencial único.

A transformação entre o referencial do corpo e o referencial local (*NED*) pode ser realizada através de três rotações consecutivas sobre os eixos do corpo Y_{corpo} , X_{corpo} e Z_{corpo} , respectivamente. A transformação é assumida pela equação seguinte:

$$R_{corpo}^{NED} = (R_{NED}^{corpo})^T = R_Z(\psi)R_X(\phi)R_Y(\theta) \quad (5.1)$$

Tendo ainda em vista a utilização de informação do sistema inercial no sistema de visão para a exclusão de falsas correspondências, é necessário relacionar a informação entre os dois sensores. Como o sistema inercial tem a sua informação no referencial do corpo e os dados de visão são tratados no referencial da câmara é preciso determinar uma rotação que coloque os dados de ambos no mesmo referencial.

A matriz de rotação que relaciona a informação do sistema inercial e o sistema de visão, no caso de os dados deste sistema serem recolhidos num plano perpendicular ao veículo (ex.: inspeção de um plano vertical) é a seguinte:

$$R_{CAM} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} R_{INS} \quad (5.2)$$

O caso contrário de inspeção dá-se quando se pretende inspeccionar um plano paralelo ao veículo (ex.: plano horizontal). Nesta configuração a relação entre sistema visão e sistema inercial é dada por:

$$R_{CAM} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} R_{INS} \quad (5.3)$$

Esta página foi intencionalmente deixada em branco.

Capítulo 6

Implementação e Resultados

Conteúdo

6.1	Arquitectura do Sistema de Navegação e Inspecção Visual . . .	61
6.2	Mosaico de Imagens	63
6.2.1	Detecção e Caracterização de Pontos de Interesse	64
6.2.2	Correspondência de Pontos	67
6.2.3	Eliminação de Falsas Correspondências	67
6.2.4	Fusão de Imagens	73
6.3	Determinação da Posição e Orientação	78
6.3.1	Comparação Rotação <i>INS</i> com Rotação dos Parâmetros Extrínsecos	78
6.3.2	Decomposição da Matriz de Homografia	80

6.1 Arquitectura do Sistema de Navegação e Inspecção Visual

O sistema de percepção/navegação desenvolvido para este projecto pode ser dividido em duas fases: a percepção e a navegação. A fase de percepção contempla a criação de um mosaico de imagens para formação de mapas. A navegação irá consistir na obtenção de informação a partir das imagens que integram o mosaico de forma a que seja possível

determinar a orientação e posição da câmara e consequentemente obter informação que permita a navegação da aeronave.

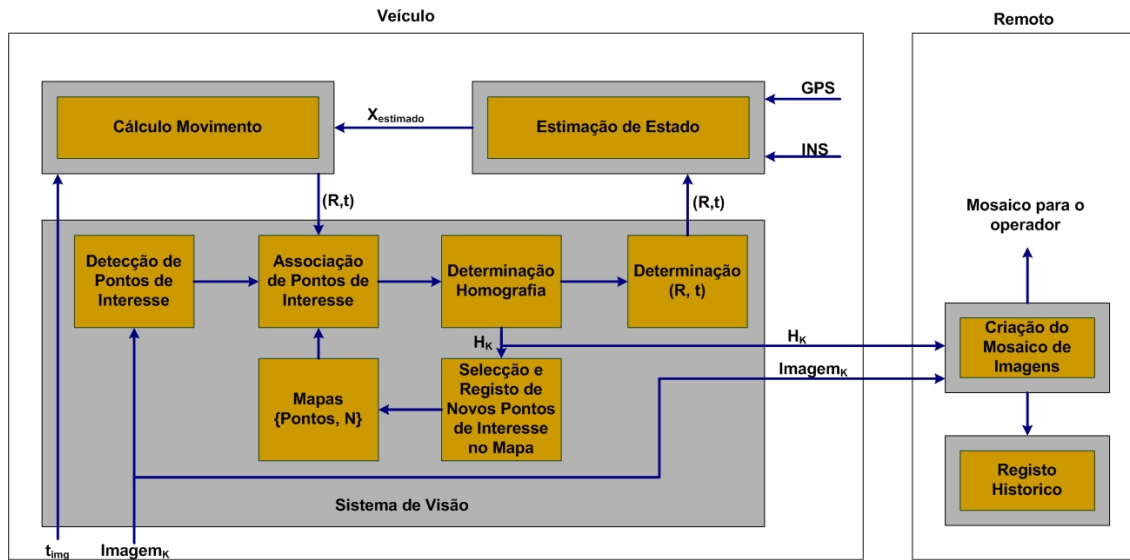


Figura 6.1: Arquitectura sistema de visão e navegação e esquema da integração no sistema do veículo.

A figura 6.1 pretende representar o modo como o sistema de visão pode ser integrado no sistema de navegação da aeronave. Os aspectos a focar para a criação de um mosaico de imagens em tempo real são a utilização da informação de rotação e translação e a conservação dos pontos de interesse das imagens. Ao serem guardados os pontos de interesse numa base de dados perde-se a necessidade de ter de procurar novos pontos nas imagens formadas (mosaico), sendo assim preciso apenas procurar na imagem a adicionar ($Imagem_k$). Esta base de dados tem de ser actualizada sempre que se funde duas imagens, ou seja quando os pontos de interesse estão validados com associações correctas.

O vector \vec{N} é um parâmetro que define a perpendicular ao plano a mapear e é dependente da aplicação, ou seja este vector define o plano do mapa a ser criado.

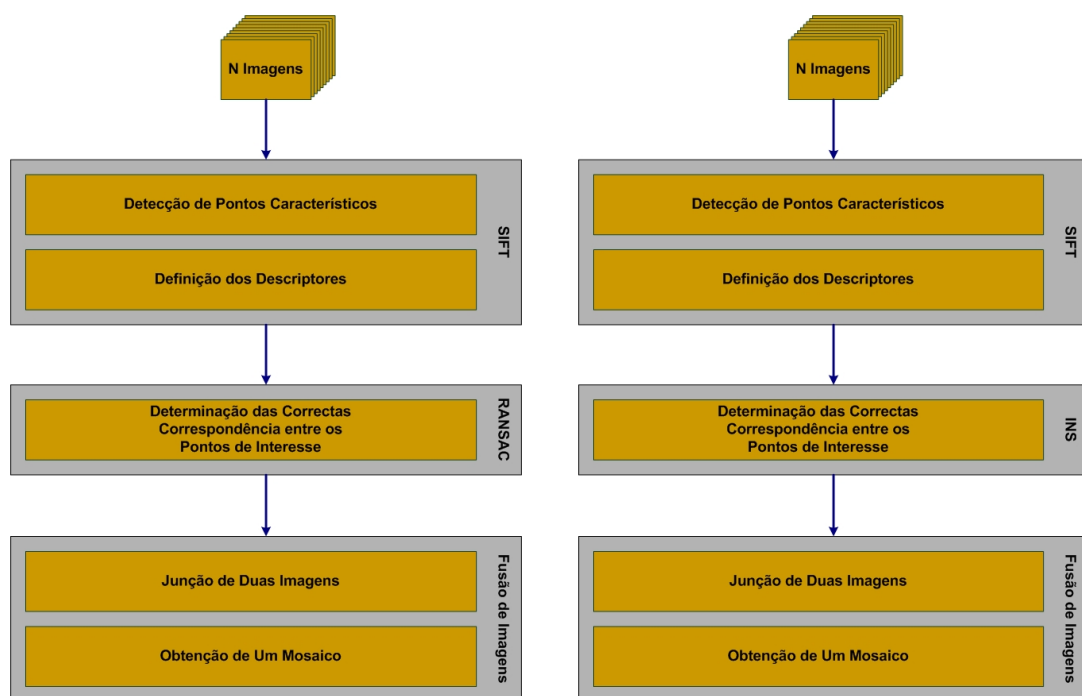
O sistema de visão constituiu ainda parte dos dados de um bloco de estimacão de posição do veículo. Neste bloco pode ser feita a fusão da informação entre vários sensores desde logo *GPS* e *INS*. Aqui o sistema de visão poderá contribuir com a estimacão da orientação e posição obtida da decomposicão da matriz de homografia, valores estes que podem servir de base para uma estimacão dos *bias* dos giroscópios e acelerómetros do *INS*, melhorando o desempenho quando não há informação disponível do estado do *GPS*.

6.2 Mosaico de Imagens

Ao longo dos capítulos anteriores tem vindo a ser apresentados os conceitos teóricos e ferramentas necessárias para efectuar um mapa com base em imagens ou *frames* de um vídeo. Toda esta matéria surge de um estudo do estado da arte (capítulo 2) sobre técnicas/algoritmos de criação de um mosaico de imagens.

Aqui são apresentadas duas soluções idênticas para a criação de um mosaico de imagens, diferindo no método de exclusão de falsas correspondências. Para uma aplicação que seja executada *offline* ou *online* num bom sistema computacional (normalmente difícil de transportar a bordo), é proposta a utilização do algoritmo *RANSAC* para a descartação de falsas correspondências. A segunda alternativa fixa-se na utilização da informação de um sistema inercial, que traz vantagens na aplicação em tempo real porque requer menos poder computacional.

Assim os passos para obter um mosaico de imagens podem ser resumidos no seguinte esquema:



(a) Sequência para obtenção de mosaico de imagens através do algoritmo *RANSAC*.

(b) Sequência para obtenção de mosaico de imagens através da informação do *INS*.

Figura 6.2: Esquema das soluções para a criação de um mosaico de imagens.

Na sequência das fases de obtenção de um mosaico de imagens (figura 6.2), à primeira

imagem é efectuada apenas uma rotação ($H = R$). A informação para determinar essa rotação vem do sistema de navegação ou directamente do *INS*. Após a primeira imagem, a imagem de base será sempre o resultado da fusão das imagens, de modo a permitir adicionar novas imagens ao mosaico existente. A procura de pontos de interesse com o algoritmo *SIFT* é feita apenas na nova imagem a adicionar, sendo os pontos de interesse da imagem anterior guardados para comparação com os da nova imagem. Com os pontos de duas imagens consecutivas é determinada a homografia para relacionar as duas imagens fazendo-as coincidir.

Nesta secção é então apresentada a aplicação e resultado de cada uma destas etapas para a criação de um mosaico de imagens.

6.2.1 Detecção e Caracterização de Pontos de Interesse

A detecção e caracterização de pontos chave/interesse nas imagens é feita com a aplicação do algoritmo *SIFT*, deste modo é possível detectar características invariantes à escala e rotação da imagem. Podemos assim dizer que os pontos detectados são únicos e estão bem definidos em cada imagem.

A aplicação do algoritmo *SIFT* para a detecção e caracterização de pontos de interesse em imagens é feita com recurso a uma ferramenta desenvolvida para *MathWorks Matlab*, por Andrea Vedaldi [44]. Esta função implementa todas as etapas do algoritmo *SIFT*, desde a aplicação de uma função de diferença de gaussiana para pesquisa de pontos em várias escalas, ao cálculo das magnitudes e orientações dos pontos de interesse.

Esta função encontra-se ainda otimizada de forma a permitir descartar os pontos de interesse próximos das margens da imagem. O facto de se ignorar esta informação é justificada pela fraca qualidade dos descritores destes pontos, quando comparados com os descritores provenientes de pontos em zonas centrais da imagem. A pior qualidade destes descritores fica a dever-se à sua janela, em redor do ponto de interesse, se situar fora da imagem o que provoca que o descritor determinado não seja o mais correcto.

Então, na detecção de pontos é devolvida a posição (x, y) do ponto de interesse no sistema de coordenadas da imagem (origem do referencial é o canto superior esquerdo da imagem).

Para completar, na caracterização dos pontos de interesse é obtida a informação da escala e orientação do ponto de interesse o que permite saber quando é definido mais do

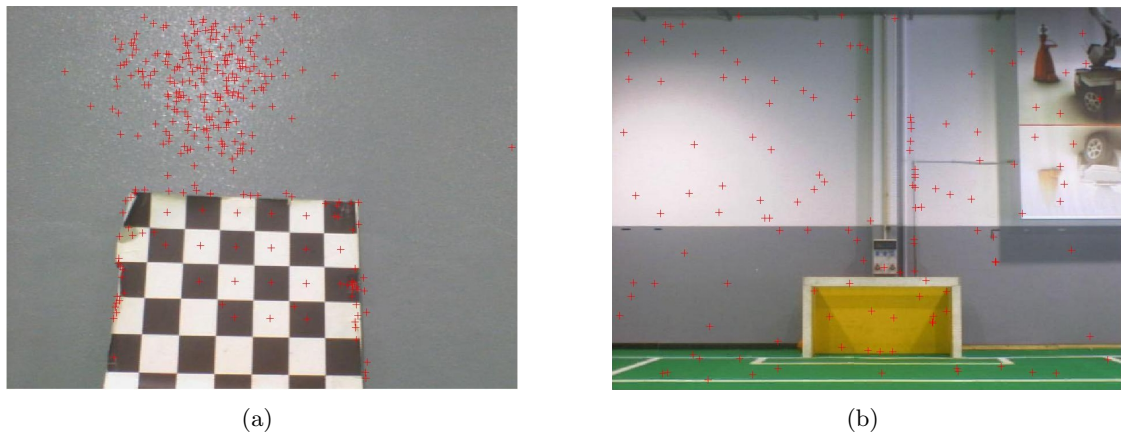


Figura 6.3: Exemplo de duas imagens com a localização dos pontos detectados pelo algoritmo *SIFT*.

que um ponto de interesse na mesma localização.

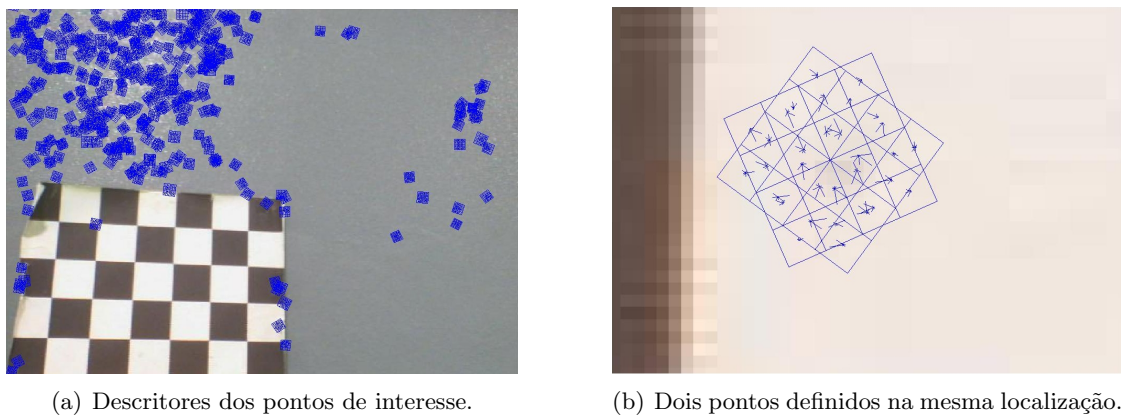


Figura 6.4: Representação dos descritores dos pontos de interesse.

Com toda esta informação proveniente da imagem é então possível definir eficazmente a localização de um ponto através do seu descritor. As duas figuras seguintes (figura 6.5 e figura 6.6) são exemplos da invariância a rotações e alterações de escala na imagem, conseguidas com o algoritmo *SIFT*.

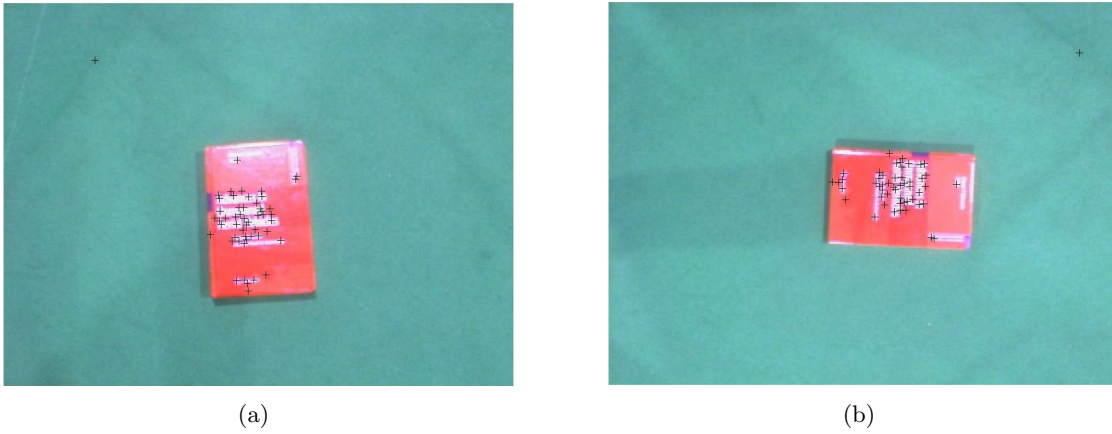


Figura 6.5: Invariância do algoritmo *SIFT*, à rotação da imagem.

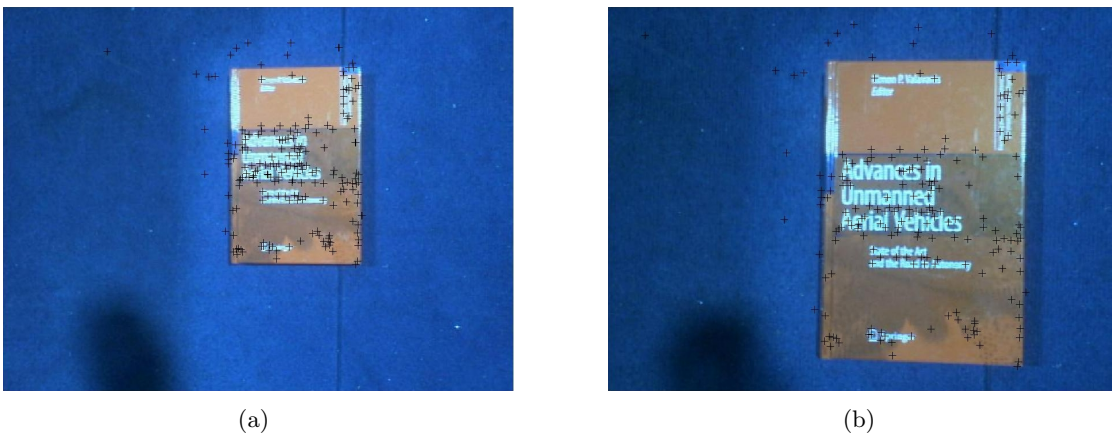


Figura 6.6: Invariância do algoritmo *SIFT*, à escala da imagem.

6.2.2 Correspondência de Pontos

Na criação de um mosaico de imagens a correspondência de pontos de interesse entre duas imagens assume uma grande importância, pois todo o trabalho vai depender da correcta associação destes pontos. Anexado à importância da correspondência de pontos está a boa qualidade dos descritores dos pontos de interesse.

Os descritores assumem grande relevância pois é neles que está contida toda a informação que permite comparar dois pontos de imagens distintas (secção 4.2), como tal quanto melhor a definição do descritor melhor é a associação feita.

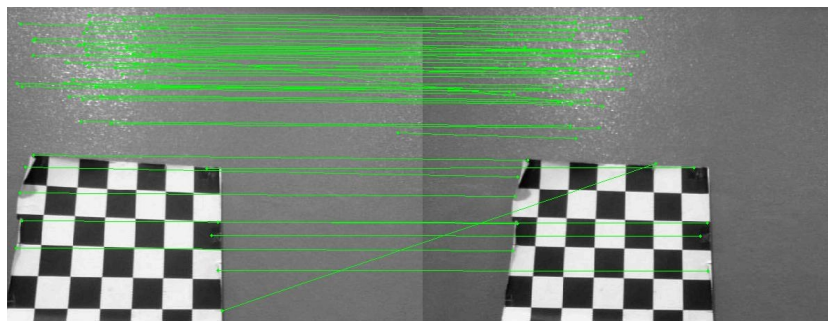


Figura 6.7: Correspondência através de descritores.

Contudo e apesar da boa qualidade dos descritores, nem todas as associações são feitas correctamente como é visível na figura 6.7, porque dois pontos bem descritos em zonas diferentes das duas imagens podem ser associados como sendo o mesmo ponto na cena ou objecto, devido à margem de erro que é dada para a associação.

6.2.3 Eliminação de Falsas Correspondências

Como vimos no passo antecedente para a criação de um mosaico de imagens, a correspondência de pontos é falível. Para a fusão de imagens é necessário ter o maior número correcto de correspondências com a finalidade de serem estimados acertadamente os parâmetros que relacionam as duas imagens.

Nesta secção serão apresentadas duas soluções para a resolução do problema da correspondência de pontos de interesse entre imagens. A primeira tem como base o algoritmo de estimação robusta *RANSAC* e é pensada para uma utilização numa aplicação *offline* e numa estação de terra (execução remota). Com a segunda hipótese pretende-se descartar as falsas correspondências através da informação obtida de um sensor inercial. Esta

alternativa visa colmatar a desvantagem a nível computacional imposta pela aplicação da estimação robusta.

6.2.3.1 Exclusão de Correspondências com *RANSAC*

Com o intuito de eliminar as falsas correspondências entre pontos de interesse das imagens é utilizado o algoritmo *RANSAC*, aplicado à determinação de homografia, usufruindo do trabalho desenvolvido por Marco Zuliani [45]. A aplicação de Zuliani permite a utilização do algoritmo *RANSAC* nas suas diversas variantes, para estimação de linhas, planos, homografias ou *RST* (*Rotation Scale Translation*).

Na utilização desta aplicação é necessário especificar que tipo de estimação pretendemos fazer, que neste caso será a estimação da homografia. A homografia permite fazer o registo¹ de uma imagem em relação à outra, através de pontos das duas imagens. Ou seja, através das correspondências feitas atrás é possível estimar a relação entre os pontos das duas imagens, podendo assim eliminar as correspondências que geram uma má estimação.

Para o algoritmo realizar a estimação é então indispensável fornecer os dados relativos a uma primeira correspondência (correspondência de pontos de interesse através dos descritores, secção 6.2.2). A execução da estimação da homografia vai ter como resultado final um *consensus set* (*CS*), que relembre-se contém todos os dados consistentes com o modelo, com a particularidade de que nesta aplicação o *CS* não contém os dados mas sim um valor lógico que permite saber se uma determinada correspondência é verdadeira (valor lógico 1) ou falsa (valor lógico 0).

$$\text{Correspondências} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} \\ x'_1 & x'_2 & x'_3 & x'_4 & x'_5 & x'_6 & x'_7 & x'_8 & x'_9 & x'_{10} \end{bmatrix} \quad (6.1)$$

$$\text{Consensus Set} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (6.2)$$

De acordo com o que foi dito atrás, das equações 6.1 e 6.2 pode-se concluir que quatro das correspondências estão correctas e as restantes são falsas.

¹O registo de uma imagem em relação a outra, consiste na determinação de parâmetros que as faça coincidir pela mesma zona comum.

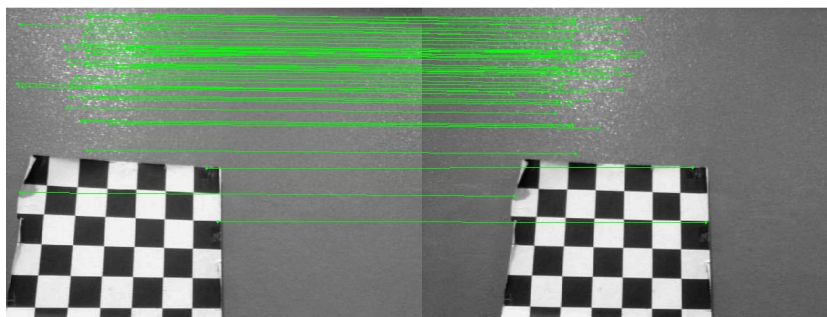


Figura 6.8: Resultado da exclusão de pontos de interesse com *RANSAC*.

Na figura 6.8 temos o resultado da aplicação do algoritmo *RANSAC* aos pontos de interesse da figura 6.7, onde é possível ver que as más associações deixaram de existir, tendo sido excluídas na execução do algoritmo. Estas novas correspondências consideradas como correctas serão as usadas na transformação/deformação da imagem de modo a construir o mosaico.

6.2.3.2 Exclusão de Correspondências com *INS*

Apesar dos excelentes resultados obtidos para a associação de dados, através da estimação robusta e neste caso particular com a utilização do algoritmo *RANSAC*, este revela-se bastante pesado computacionalmente. Pelo que o seu emprego em aplicações que são executadas em tempo real traz uma desvantagem.

Aproveitando o facto de existir neste veículo um sistema inercial (*INS*), pode ser usada a informação proveniente deste para a exclusão de falsas correspondências.

O sistema inercial é capaz de fornecer os dados da atitude da câmara no momento em que uma imagem é capturada. Deste modo, através da atitude de duas imagens consecutivas (R_1 e R_2 da equação 6.3) é possível determinar a rotação (R_{21} da equação 6.3) que transforma os pontos da segunda imagem (P_{img_2} na equação 6.4) para a primeira imagem (P_{img_1} na equação 6.4).

$$R_{21} = R_1 R_2^T \quad (6.3)$$

$$P_{img_1} = R_{21} P_{img_2} \quad (6.4)$$

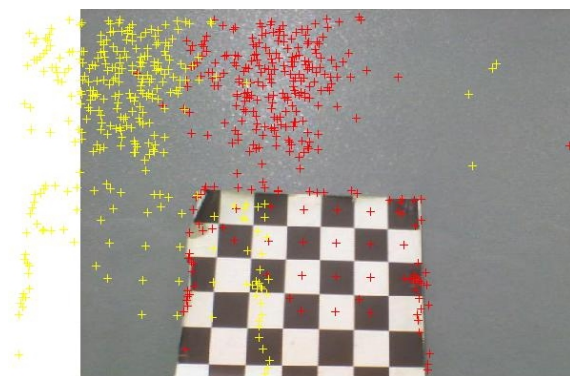
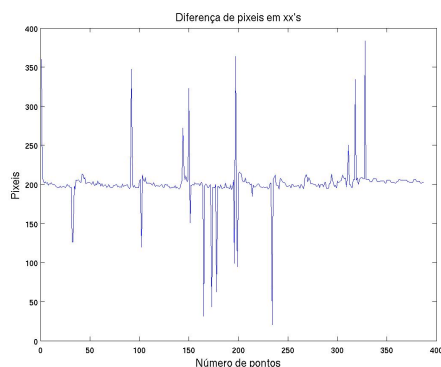


Figura 6.9: Resultado da aplicação da rotação do *INS* aos pontos característicos da imagem 2 (pontos a amarelo) colocando-os na imagem 1 (pontos a vermelho).

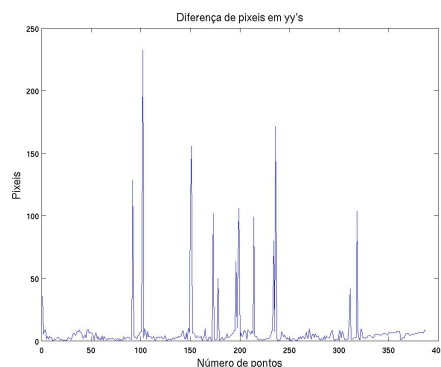
Assim para os pontos de duas imagens coincidirem falta apenas encontrar a translação que existe entre as mesmas. A translação pode ser determinada através da análise do erro em coordenadas de $xx's$ e $yy's$ (equação 6.5).

$$\begin{bmatrix} \Delta_x \\ \Delta_y \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} - R \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} \quad (6.5)$$

Isto é, como existe uma pré-correspondência feita através dos descritores produzidos pelo algoritmo *SIFT* (secção 6.2.2), será fácil perceber que más associações vão gerar erros maiores.



(a) Disparidade em x



(b) Disparidade em y

Figura 6.10: Análise da disparidade nos eixos dos $xx's$ e $yy's$, referente aos dados da figura 6.9.

Das figuras 6.10(a) e 6.10(b) pode-se concluir que a translação é aproximadamente igual há média do erro em x e y , respectivamente para os eixos de $xx's$ e $yy's$, já que

pontos que provêm de más associações geram picos de disparidade fora da média e têm de ser descartados. Será então possível definir uma faixa do erro (*threshold*) na qual os pontos são considerados como boas associações, ficando os pontos que geram erros superiores descartados (figura 6.12).

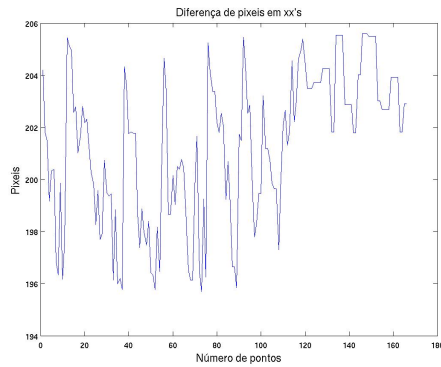
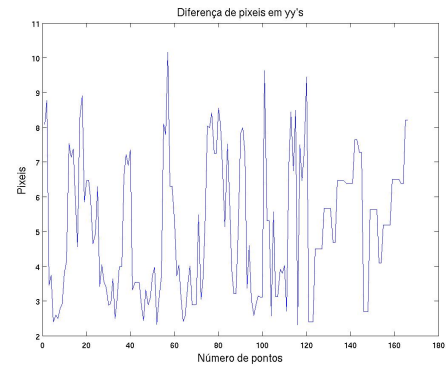
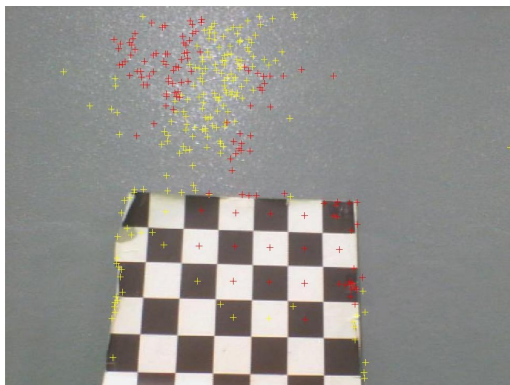
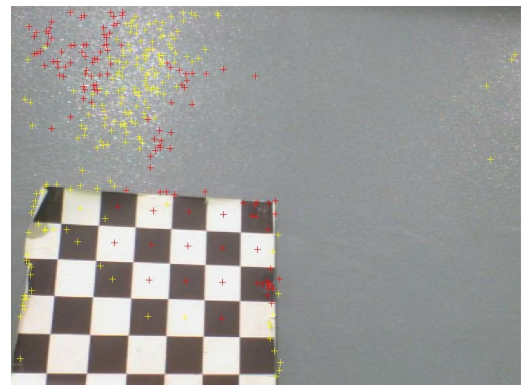
(a) Disparidade em x .(b) Disparidade em y .

Figura 6.11: Análise da disparidade nos eixos dos $xx's$ e $yy's$, após exclusão de más associações.



(a) Pontos excluídos na primeira imagem.



(b) Pontos excluídos na segunda imagem.

Figura 6.12: Pontos excluídos nas imagens. A amarelo estão representados os pontos descartados, a cor vermelha representa os pontos cuja associação foi classificada como correcta.



Figura 6.13: Resultado final da descartação de pontos. Pontos a vermelho são pontos da imagem 1, os pontos amarelos representam os pontos da imagem 2 na imagem 1, já com o resultado final da rotação e translação.

6.2.3.3 Comparação entre *RANSAC* e *INS*

Como tem vindo a ser anunciado ao longo dos passos para a criação de um mosaico de imagens a utilização do algoritmo de estimação robusta *RANSAC* para a descartação de falsas correspondências requer bastante poder computacional, em parte devido ao modo como é executado (iterativamente). A utilização da informação do sistema inercial apresenta-se como uma boa solução para fazer baixar os recursos computacionais necessários.

Nesta secção é feita uma breve comparação dos resultados obtidos pelas duas soluções, de forma a justificar a utilização quer de um método quer de outro, nas aplicações já mencionadas. A análise dos tempos de processamento computacional dos algoritmos é feita em através de *MathWorks Matlab*.

	Método I				Método II		
	Número de Pontos				Número de Pontos		
	80	320	430		80	320	430
SIFT	66.01	75.40	90.70	SIFT	99.76	99.79	99.98
RANSAC	33.90	24.50	9.27	INS	0.108	0.073	0.014
Junção de Imagens	0.09	0.10	0.03	Junção de Imagens	0.134	0.135	0.028

Tabela 6.1: Análise da percentagem de processamento.

Na tabela 6.1 está presente uma análise ao peso que cada grande bloco tem na criação de mosaico de imagens, para diferentes números de pontos de interesse detectados. Desde logo a primeira conclusão a tirar é que ainda mais pesado computacionalmente que a descartação de pontos é a detecção de pontos de interesse com o algoritmo *SIFT*, função

que ocupa praticamente todo o tempo de processamento. Voltando a análise para o que se pode ganhar com a utilização da informação do sensor inercial (*INS*) numa aplicação em tempo real, os ganhos são claros face à utilização do algoritmo de estimação robusta *RANSAC*. Principalmente com um número reduzido de pontos de interesse.

6.2.4 Fusão de Imagens

A etapa fulcral onde o mosaico de imagens vai adquirir o seu aspecto final é na fusão de imagens. Nesta fase são determinados os parâmetros que permitem relacionar as duas imagens havendo um mínimo de quatro correspondências correctas, em qualquer outro caso a imagem terá de ser descartada e todo o processo reiniciado.

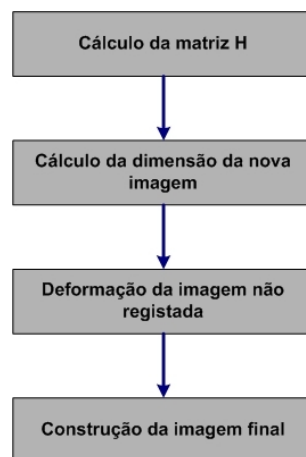


Figura 6.14: Esquema das etapas para a fusão de imagens.

A matriz de homografia (H) pela qual os pontos das duas imagens vão ser relacionados é dada pela transformação *affine* 2D ou transformação projectiva 2D. A transformação a utilizar fica dependente da aplicação para a qual vai ser usada a criação do mosaico. Pode-se então dividir em dois tipos: aplicações de mosaico para cenas onde é perceptível a profundidade das imagens (cenas ou objectos próximos da câmara) e aplicações onde a cena ou objecto estão afastados da imagem e a profundidade é desprezável. No primeiro caso é necessário uma transformação com maior número graus de liberdade para deformar a imagem o suficiente, pelo que a transformada aplicada será projectiva. Na segunda hipótese são precisos menos graus de liberdade para a deformação logo é aplicada a transformação bidimensional *affine*.

Neste momento é importante definir qual a imagem de base e qual a imagem não registada, pois para um caso vão ser calculados os parâmetros da matriz homografia directa

e no caso contrário é determinada a matriz homografia na sua forma inversa. Assim a imagem de base será sempre a última imagem produzida pelo mosaico de imagens e a imagem não registada será sempre a imagem proveniente do lote de novas imagens (figura 6.2).

6.2.4.1 Fusão de Imagens com Transformação *Affine*

Após a execução da etapa de exclusão de falsas correspondências entre pontos característicos de duas imagens, têm-se todas as variáveis descritas para a determinação da transformação *affine* 2D. Os parâmetros desta transformação são calculados através da resolução do sistema linear $Ax = b$ (equação 6.11) obtido da reformulação da equação 3.22 na seguinte forma:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (6.6)$$

$$\begin{bmatrix} x_1 & y_1 & 0 & 0 & 1 & 0 \\ 0 & 0 & x_1 & y_1 & 0 & 1 \\ x_2 & y_2 & 0 & 0 & 1 & 0 \\ 0 & 0 & x_2 & y_2 & 0 & 1 \\ x_3 & y_3 & 0 & 0 & 1 & 0 \\ 0 & 0 & x_3 & y_3 & 0 & 1 \\ x_4 & y_4 & 0 & 0 & 1 & 0 \\ 0 & 0 & x_4 & y_4 & 0 & 1 \\ \dots & & & & & \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_{21} \\ a_{22} \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ x'_3 \\ y'_3 \\ x'_4 \\ y'_4 \\ \dots \end{bmatrix} \quad (6.7)$$

Da resolução da equação linear 6.11 são então encontrados os parâmetros da transformação *affine* 2D, pode então deduzir-se a rotação e escalonamento (matriz A) e translação (matriz t), componentes que constituem a matriz homografia, com base na informação da secção 3.2.1.3:

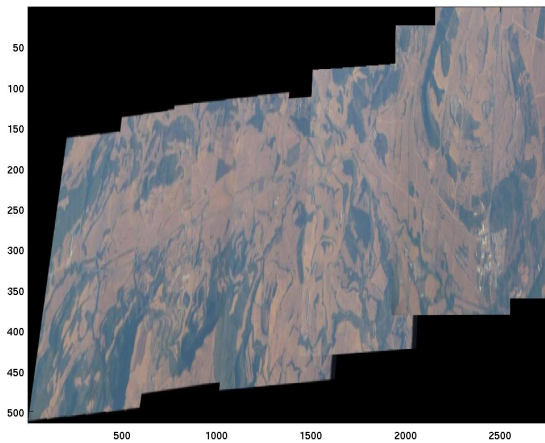
$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad (6.8)$$

$$t = \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (6.9)$$

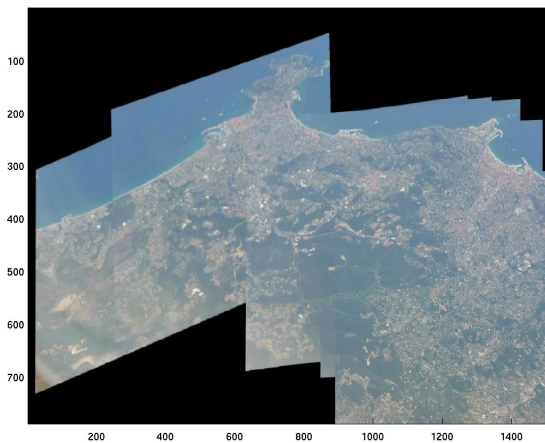
A aplicação destes parâmetros à imagem não registada provoca uma deformação na

imagem, de modo a que esta fique coincidente com a imagem de base pelos pontos de interesse que geraram boas correspondências. O resultado desta operação é a imagem final de um mosaico de imagens.

Os dois exemplos de mosaico de imagens apresentados na figura 6.15, são construídos com base em imagens captadas de um avião comercial com uma máquina fotográfica *Lumix TZ5*. Neste exemplo a exclusão dos pontos de interesse é feito com recurso ao algoritmo *RANSAC* pois não havia dados do sistema inercial.



(a)



(b)

Figura 6.15: Dois exemplos da junção de imagens com transformação *affine*.

6.2.4.2 Fusão de Imagens com Transformação Projectiva

No caso da aplicação de uma transformação projectiva bidimensional o processo é idêntico ao anterior, apenas havendo variações nas matrizes que constituem o sistema linear que permite a obtenção dos parâmetros.

Reescrevendo a equação 3.28 que permite encontrar os parâmetros da matriz homografia projectiva, é obtido o seguinte sistema:

$$\begin{cases} x' = \frac{xh_{11} + yh_{12} + h_{13}}{xh_{31} + yh_{32} + h_{33}} \\ y' = \frac{xh_{21} + yh_{22} + h_{23}}{xh_{31} + yh_{32} + h_{33}} \end{cases} = \begin{cases} xh_{11} + yh_{12} + h_{13} - xx'h_{31} - yy'h_{32} = x'h_{33} \\ xh_{21} + yh_{22} + h_{23} - xy'h_{31} + yy'h_{32} = y'h_{33} \end{cases} \quad (6.10)$$

Neste caso a equação linear $Ax = b$, é obtida da análise do sistema de equações anterior assumindo que $h_{33} = 1$, e tem a seguinte forma:

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1x_1 & -x'_1y_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & y'_1x_1 & y'_1y_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x'_2x_2 & -x'_2y_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & y'_2x_2 & y'_2y_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x'_3x_3 & -x'_3y_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & y'_3x_3 & y'_3y_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x'_4x_4 & -x'_4y_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & y'_4x_4 & y'_4y_4 \\ & & & & & \dots & & \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = \begin{bmatrix} x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ x'_3 \\ y'_3 \\ x'_4 \\ y'_4 \\ \dots \end{bmatrix} \quad (6.11)$$

Tal como no caso anterior a aplicação destes parâmetros há imagem não registada vai provocar uma deformação, que faz com que este fique alinhada com a imagem de base, criando assim um mosaico de imagens.

Na figura 6.17 são apresentados os resultados deste método para a junção de imagens. Para o primeiro mosaico, figura 6.17(a), as imagens foram capturadas com uma webcam *Philips SPC900*, nesta podemos verificar a correcta junção de imagens para cenas com geometria próxima da câmara. O segundo exemplo é novamente uma imagem obtida a partir de um avião comercial com um máquina fotográfica comum (*Lumix TZ5*).

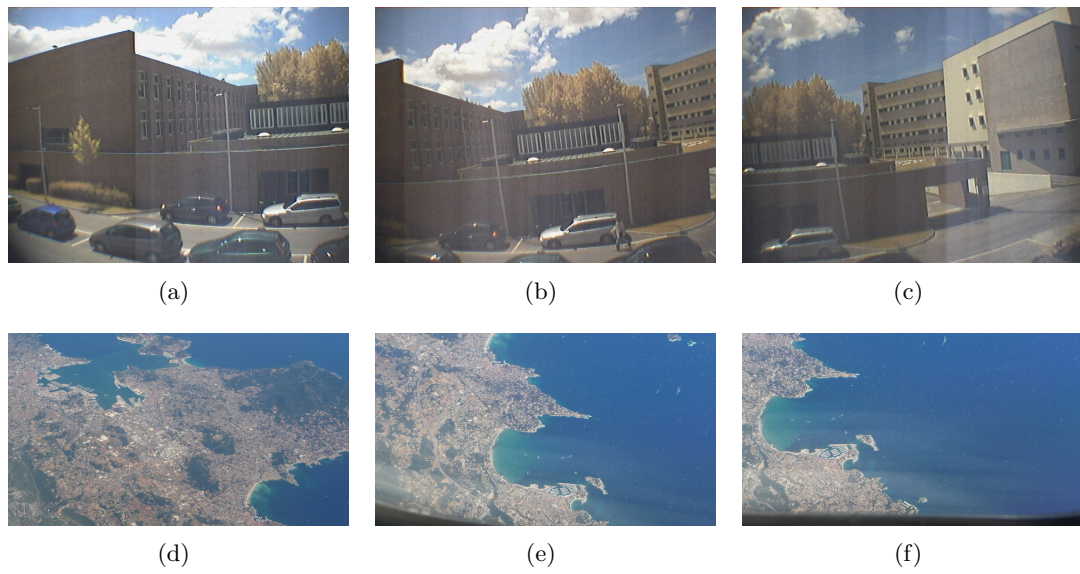


Figura 6.16: Imagens de base para os mosaicos de imagens apresentados na figura 6.17.

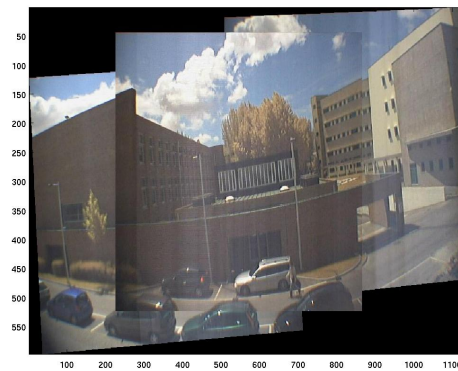


Figura 6.17: Dois exemplos da junção de imagens com transformação projectiva, neste caso já é possível unir correctamente fotos com cenas/objectos próximos da câmara.

6.3 Determinação da Posição e Orientação

O sistema de visão desenvolvido prevê ainda a determinação da posição e orientação da câmara², com a finalidade de obter informação que possa servir de apoio à navegação.

Como ficou demonstrado na exposição teórica (capítulo 3) é possível determinar a posição e orientação das imagens e consequente transformação necessária para chegar de uma imagem a outra. As características deste movimento podem ser encontradas, por exemplo, nos parâmetros extrínsecos da câmara, na matriz de homografia definida por uma transformação projectiva ou ainda na matriz essencial da geometria epipolar. As decomposições das matrizes essencial e de homografia revelam-se importantes pois podem ser obtidas através da correspondência de pontos, ou seja, será possível obter a rotação e translação com base nestas correspondências.

Nesta secção pretende-se demonstrar a possibilidade de obter a rotação e translação entre duas imagens, através quer da decomposição da matriz de homografia, quer da decomposição da matriz essencial. Destas variáveis a translação é a que assumirá maior importância, visto ser a que permite corrigir o *bias drift* do sistema inercial.

6.3.1 Comparação Rotação *INS* com Rotação dos Parâmetros Extrínsecos

Numa primeira análise foi feita uma comparação entre a rotação fornecida pelo *INS* e a rotação determinada na obtenção dos parâmetros extrínsecos. Com esta comparação pretende-se validar que os valores de rotação fornecidos pelos dois métodos são idênticos, deixando assim aberta a possibilidade para calcular a rotação e translação com base na decomposição da matriz essencial ou da matriz de homografia.

A determinação da rotação dada pelo *INS* no instante em que a imagem é capturada é calculada com base na medida dos ângulos de *euler*.

Os parâmetros extrínsecos e intrínsecos são determinados através da *toolbox* de calibração de câmaras desenvolvida por *Jean-Yves Bouguet* [46]. Desta ferramenta é possível retirar valores de translação e rotação da câmara para cada imagem relativamente a um alvo de calibração, podendo daí obter a rotação entre duas imagens com a equação 6.3.

²A determinação da posição e orientação de uma câmara é também denominada na maioria da literatura de *Structure from Motion*

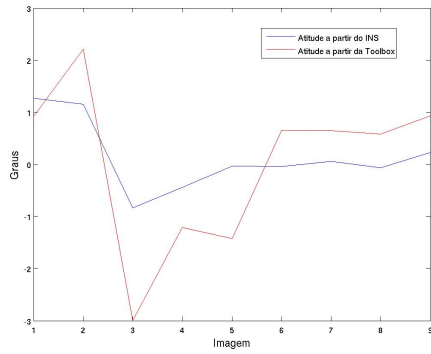
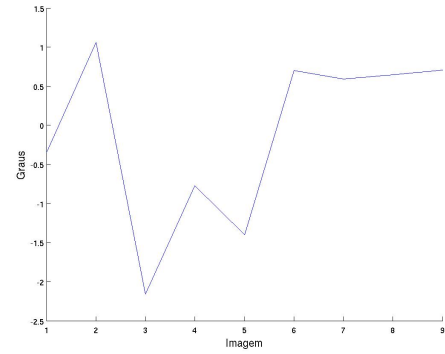
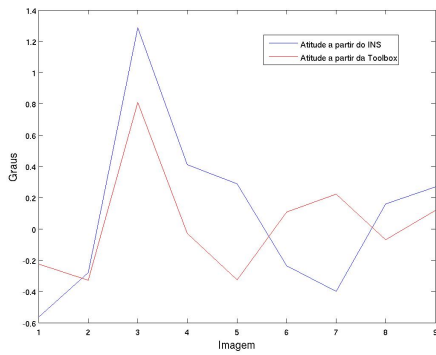
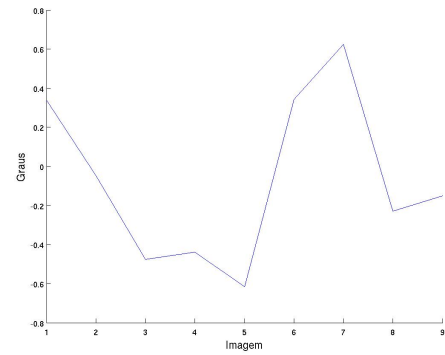
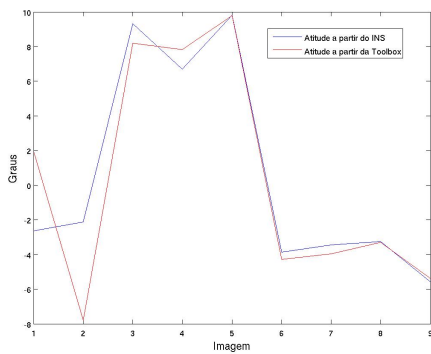
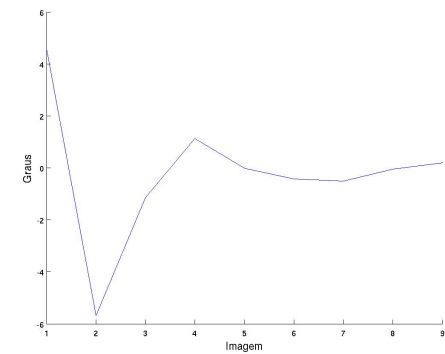
(a) Atitude em *roll*.(b) Erro em *roll*.(c) Atitude em *pitch*.(d) Erro em *pitch*.(e) Atitude em *yaw*.(f) Erro em *yaw*.

Figura 6.18: Comparação entre a atitude nos ângulos de *roll*, *pitch* e *yaw* obtidos a partir da toolbox de calibração de câmaras e da informação do *INS*.

Da figura 6.18 constata-se que os valores da atitude são idênticos pelo que é possível dizer que um sistema de visão consegue obter a atitude da câmara e consequentemente a atitude da aeronave, onde está acoplada.

6.3.2 Decomposição da Matriz de Homografia

Tirando partido de se ter calculado previamente uma matriz de homografia na etapa de junção de imagens e com o suporte da conclusão da comparação anterior é então feita a decomposição da matriz de homografia (secção 3.5). Desta forma obtêm-se os valores de rotação e translação que permitem determinar a posição e orientação da câmara e portanto da aeronave.

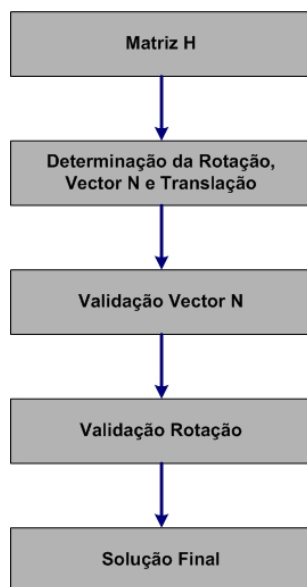
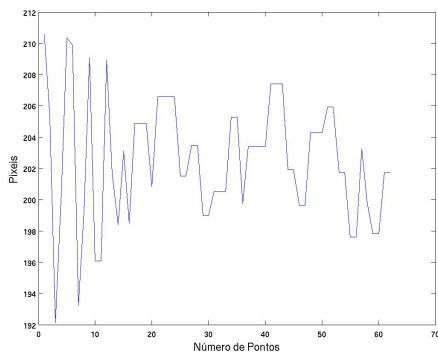


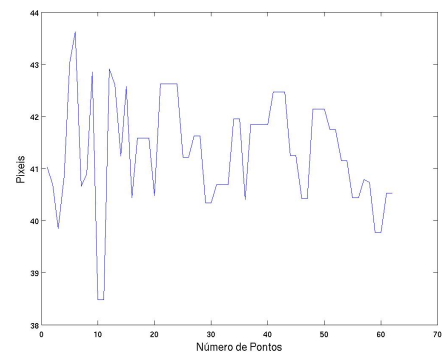
Figura 6.19: Esquema das etapas para a decomposição da matriz de homografia.

A decomposição da matriz de homografia gera quatro soluções, como apenas uma destas é a correcta será preciso eliminar as restantes três. Assim, começa-se desde logo por verificar qual o vector (N) que define correctamente o plano, pela análise do seu terceiro elemento ($n_3 > 0$).

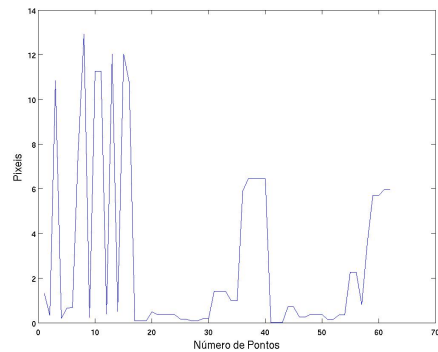
A veracidade das hipóteses para a rotação (R_1 e R_2) pode ser vista de forma análoga ao que foi feito na aplicação da rotação do *INS* a pontos de uma imagem. Isto é, a hipótese de rotação correcta tem valores de disparidade menores que os valores de disparidade de hipótese errada, este facto deve-se a esta segunda colocar os pontos atrás do plano da imagem.



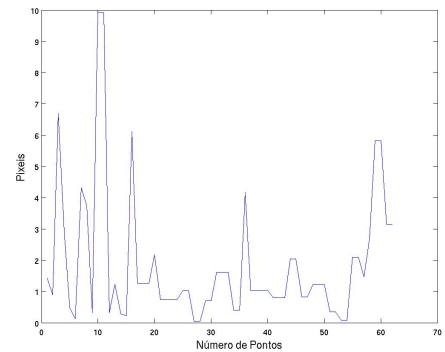
(a) Disparidade em x provocada pela primeira hipótese de rotação.



(b) Disparidade em y provocada pela primeira hipótese de rotação.



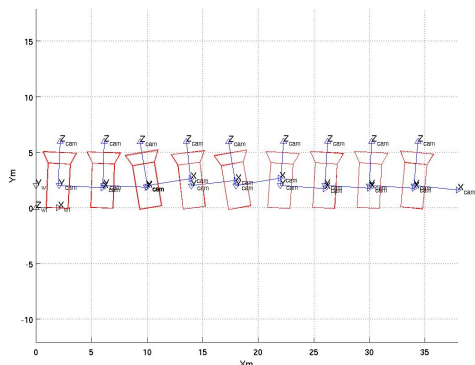
(c) Disparidade em x provocada pela segunda hipótese de rotação.



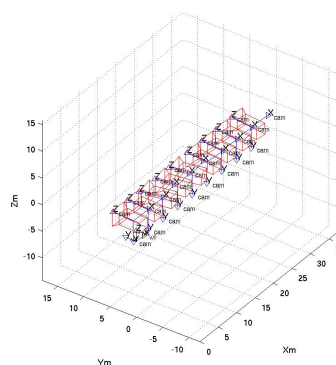
(d) Disparidade em y provocada pela segunda hipótese de rotação.

Figura 6.20: Disparidades provocadas por duas hipóteses de rotação da decomposição da matriz de homografia.

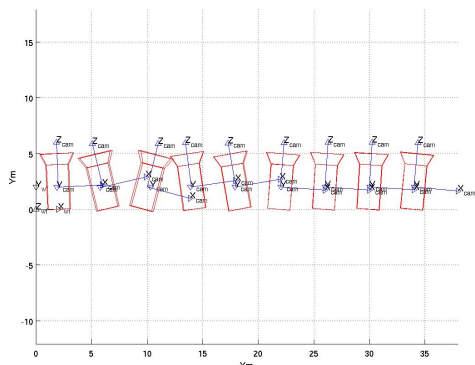
Com o vector plano e rotação correctamente escolhidos de entre as hipóteses apresentadas é agora possível chegar há solução final da decomposição, visto só haver uma translação capaz de satisfazer os restrimentos impostos pelas opções anteriores. Na translação não se encontra o valor real do deslocamento pois esta é determinada a menos de um factor de escala, podendo apenas concluir a relação entre os deslocamentos em cada eixo e o sentido em que o deslocamento se efectuou.



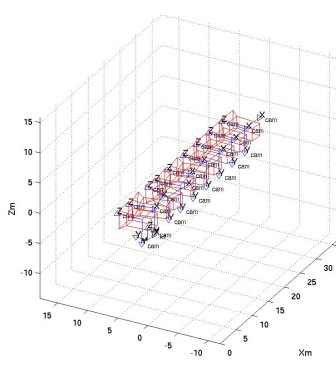
(a) Representação da atitude do INS



(b) Representação da atitude do INS



(c) Representação da atitude pela decomposição da matriz de homografia



(d) Representação da atitude pela decomposição da matriz de homografia

Figura 6.21: Comparação dos resultados obtidos para a atitude através de INS e decomposição da matriz de homografia. A representação da atitude por H, deve-se apenas há sua hipótese de rotação correcta em ambos os casos os valores de translação não são os reais.

Capítulo 7

Conclusões e Trabalho Futuro

Ao longo desta dissertação foram abordados conceitos de visão computacional e fundamentos de navegação que permitiram a obtenção de mosaicos de imagens para operações de monitorização e inspecção, bem como retirar informação que dê apoio à navegação da aeronave.

Foram implementadas técnicas de mosaico em *MathWorks Matlab* utilizando abordagens do estado da arte combinando detectores de pontos de interesse *SIFT*, com métodos de exclusão de falsas correspondências baseados em *RANSAC*, estimação de transformações recorrendo a técnicas de mínimos quadrados e com junção de imagens e pontos do mapa utilizando transformações affine e projectiva.

Na etapa de junção de imagens são apresentadas duas transformações bi-dimensionais (transformação *affine* e transformação projectiva) para fazer o registo de imagens. A transformação *affine* tem um bom desempenho para imagens onde a cena geométrica está bastante afastada da câmara, ou seja imagens onde a profundidade não é perceptível. Isto não acontece com cenas geométricas próximas da câmara pelo que o desempenho desta transformação é afectado, nestes casos a aplicação da transformação projectiva vem resolver esta questão, visto que esta tem mais graus de liberdade sendo possível deformar a imagem de forma a fazer o correcto registo de imagens.

A estimação robusta e em particular a aplicação do algoritmo *RANSAC* a visão computacional permite excelentes resultados na exclusão de falsas correspondências entre pontos de interesse. O ponto fraco da aplicação do *RANSAC* prende-se com o facto de o seu

processo de determinação de um conjunto de boas associações ser feito iterativamente.

Tendo como objectivo a utilização das técnicas de mosaico em tempo real, foi proposta uma variante da técnica previamente implementada onde a informação de um sistema inercial é utilizada substituindo o método *RANSAC* na etapa de exclusão de falsas associações. O método proposto foi implementado e validado experimentalmente apresentando resultados idênticos aos do método inicial mas fazendo o poder computacional necessário diminuir, o que traz uma mais valia na aplicação em tempo real.

Apesar do ganho computacional que se obtém com a integração de informação do sistema inercial na associação de pontos de interesse entre imagens, verifica-se que o detector de features *SIFT* impõe requisitos computacionais elevados, para o sistema computacional do *UAV*. A utilização deste detector foi sendo sustentada em parte pelo modo como consegue descrever um ponto de interesse, com bastante informação.

Os valores de rotação obtidos pelo sistema inercial no acto de captura das imagens demonstram-se consistentes com os valores encontrados nos parâmetros extrínsecos da câmara, obtidos de uma *toolbox* de calibração de câmaras, concluindo-se que de um sistema de visão é possível determinar a posição e orientação da câmara. Esta comparação abre então caminho há utilização da matriz de homografia, que contém os parâmetros da transformação projectiva e que relaciona duas imagens, podendo ser decomposta em factores de rotação e translação.

Foram estudados e implementados métodos para extracção da informação do movimento da câmara a partir das transformações estimadas da matriz de homografia entre imagens. Foi validado experimentalmente que a rotação assim obtida apresentava valores estimados confiáveis e idênticos quer aos obtidos do sistema inercial, quer aos obtidos na determinação dos parâmetros extrínsecos. A translação é obtida a menos de um factor de escala, pelo que a direcção fica bem definida e o módulo está relacionado pela distância entre a câmara e plano da cena geométrica. Assim da decomposição é possível concluir a orientação da câmara e respectivamente do veículo.

Com toda esta informação foi proposta uma arquitectura para o sistema de navegação e inspecção visual baseado num mosaico de imagens que integra estes resultados e apoie a navegação. Nesta arquitectura pretende-se ainda que os pontos guardados na base de dados abordo do veículo, constituam a informação do mapa para a navegação actuando como pontos de referência. Todos os cálculos para determinação da matriz de homografia

serão efectuados a bordo podendo estes ser enviados para terra em conjunto com a imagem a registar, onde seria adicionada ao mosaico de imagens para efeitos de inspecção e monitorização.

Como trabalho futuro será importante identificar outros métodos de extracção de pontos de interesse, menos exigentes computacionalmente mesmo que estes métodos forneçam menos informação como a que é fornecida pelos descritores do algoritmo *SIFT*. Uma solução poderia passar por detectores como *KLT* (*Kanade-Lucas-Tomasi*) ou *Harris Corner* que necessitam de um menor poder computacional, embora não descrevam tão eficazmente os pontos de interesse quanto o algoritmo *SIFT*. Esta ineficiência poderá ser ultrapassada pela utilização de informação do sistema inercial que permite uma boa exclusão de falsas associações de pontos de interesse.

Para trabalho futuro pode-se ainda apontar o desenvolvimento e implementação de um sistema de navegação que siga a arquitectura proposta e integre a informação visual com sensores como *GPS* e *INS*. Na fusão de informação destes sistemas pode-se destacar o apoio que o sistema de visão dá ao *INS* e a informação de translação vinda do *GPS* que torna possível determinar a distância a objectos detectados nas imagens ou mapas de inspecção.

Esta página foi intencionalmente deixada em branco.

Referências

- [1] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.
- [2] Yi Ma, Stefano Soatto, Jana Košecká, and S. Shankar Sastry. *An Invitation to 3-D Vision, from images to geometric models*. Springer, 2006.
- [3] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Draft Version, 2009.
- [4] Guilherme Amaral. *Desenvolvimento e Controle de um Veículo Autônomo Aéreo Quadrotor*. A publicar, 2009.
- [5] Gabe Hoffmann, Dev Gorur Rajnarayan, Steven L. Waslander, David Dostal, Jung Soon Jang, and Claire J. Tomlin. The Stanford Testbed Of Autonomous Rotorcraft For Multi Agent Control (STARMAC). *The 23rd Digital Avionics Systems Conference, 2004. DASC 04.*, Volume 2:12.E.4–121–10, 2004.
- [6] Mario J. Valenti. *Approximate Dynamic Programming with Applications in Multi-Agent Systems*. PhD thesis, Massachusetts Institute of Tecnology, 2007.
- [7] Glenn P. Tournier. *Six Degree of Freedom Estimation Using Monocular Vision and Moiré Patterns*. PhD thesis, Massachusetts Institute of Tecnology, 2006.
- [8] Samir Bouabdallah, Roland Siegwart, and André Noth. PID vs LQ Control Techniques Applied to an Indoor Micro Quadrotor. *In Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004. (IROS 2004).*, Volume 3:2451– 2456, 2004.
- [9] Samir Bouabdallah and Roland Siegwart. Backstepping and Sliding Mode Techniques Applied to an Indoor Micro Quadrotor. *In Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005. ICRA 2005.*, pages 2247– 2252, 2005.
- [10] Samir Bouabdallah, Roland Siegwart, and Pierpaolo Murriero. Design and Control of an Indoor Micro Quadrotor. *In Proceedings of ICRA '04. IEEE International Conference on Robotics and Automation, 2004.*, Volume 5:4393– 4398, 2004.
- [11] Erdinç Altuğ, James P. Ostrowski, and Camillo J. Taylor. Quadrotor Control Using Dual Camera Visual Feedback. *In Proceedings of ICRA '03. IEEE International Conference on Robotics and Automation, 2003.*, Volume 3:4294–4299, 2003.
- [12] A. Ravikanth amd C. R. Raviteja, N. N. V. S. Pavan Kumar, Vamsimohan Ch., Vikram R. Shah, Hem Rampal, Kashyap G., Pradeep M., and Kedar Kulkarni. Development of An Autonomous Aerial Vehicle Capable of Indoor Navigation. *The First Symposium on Indoor Flight Issues, 2009*, 2009.

- [13] Peter Corke, Jonathan Roberts, Paul Pounds, Robert Mahony, and Joel Gresham. Towards Dynamically-Favourable Quad-Rotor Aerial Robots. 2004.
- [14] Srikanth Saripalli, Jonathan M. Roberts, Peter Corke, Gregg Buskey, and Gaurav S. Sukhatme. A Tale of Two Helicopters. *In Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003. (IROS 2003).*, Volume 1:805–810, 2003.
- [15] Courtney S. Sharp, Omid Shakernia, and S. Shankar Sastry. A Vision System for Landing an Unmanned Aerial Vehicle. *In Proceedings of 2001 IEEE International Conference on Robotics and Automation*, pages 1720–1727, 2001.
- [16] Allen D. Wu, Eric N. Johnson, and Alison A. Proctor. Vision-Aided Inertial Navigation for Flight Control. *AIAA Guidance Navigation and Control Conference and Exhibit*, 2005.
- [17] Stephen Griffiths, Jeff Saunders, Andrew Curtis, Tim McLain, and Randy Beard. Obstacle and Terrain Avoidance for Miniature Aerial Vehicles. *Chapter 7 in Advances in Unmanned Aerial Vehicles: State of the art and the road to autonomy*, edited by Kimon P. Valavanis, Springer Verlag, 2007.
- [18] Jean Michel Pflimlin, Philippe Souères, and Tarek Hamel. Waypoint Navigation Control of a VTOL UAV Amidst Obstacles. *In Proceedings of 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006*, pages 3544–3549, 2006.
- [19] Roger Johnson, Jerzy Sasiadek, and Janusz Zalewski. Kalman Filter Enhancement for UAV Navigation. 2002.
- [20] Pifu Zhang, Jason Gu, Evangelos E. Miliotis, and Peter Huynh. Navigation with IMU/GPS/Digital Compass with Unscent Kalman Filter. *IEEE International Conference on Mechatronics and Automation, 2005*, Volume 3:1497–1502, 2005.
- [21] J. F. Vasconcelos, J. Calvário, P. Oliveira, and C. Silvestre. GPS Aided IMU for Unmanned Air Vehicles. *5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles*, 2004.
- [22] L. Doitsidis, Kimon P. Valavanis, N. C. Tsourveloudis, and M. Kontitsis. A Framework for Fuzzy Logic Based UAV Navigation and Control. *In Proceedings of ICRA '04. IEEE International Conference on Robotics and Automation, 2004.*, Volume 4:4041– 4046, 2004.
- [23] Luiz G. B. Mirisola and Jorge M. M. Dias. Stereo Vision 3D Map Registration for Airships using Vision-Inertial Sensing. *3rd European Conference on Mobile Robots*, 2007.
- [24] Luiz G. B. Mirisola and Jorge M. M. Dias. Exploiting Inertial Sensing in Mosaicing and Visual Navigation. *In Proceedings of the 6th IFAC Symposium on Intelligent Autonomous Vehicles*, 2007.
- [25] Luiz G. B. Mirisola and Jorge M. M. Dias. Tracking from a Moving Camera with Attitude Estimates.

- [26] Yingdong Huang, Jie Li, and Ningjun Fan. Image Mosaicing for UAV Application. *International Symposium on Knowledge Acquisition and Modeling, 2008. KAM '08.*, pages 663–667, 2008.
- [27] Yi Wang, Aldo Camargo, Ronald Fevig, Florent Martel, and Richard R. Shultz. Image Mosaicing from Uncooled Thermal IR Video Captured by a Small UAV. *In Proceedings of the 2008 IEEE Southwest Symposium on Image Analysis and Interpretation*, pages 161–164, 2008.
- [28] Valdim Indelman, Pini Gurfil, Ehud Rivlin, and Hector Rotstein. Navigation Aiding Using On-Line Mosaicking. *IEEE/ION Position, Location and Navigation Symposium, 2008*, pages 779–791, 2008.
- [29] Andreas Huster, Stephen D. Fleischer, and Stephen M. Rock. Demonstration of a Vision-Based Dead-Reckoning System for Navigation of an Underwater Vehicle. *In Proceedings Of The 1998 Workshop on Autonomous Underwater Vehicles, 1998. AUV'98.*, pages 185–189, 1998.
- [30] Nuno Gracias, Sjoerd van der Zwaan, Alexandre Bernardino, and José Santos Victor. Mosaic Based Navigation for Autonomous Underwater Vehicles. *IEEE Journal of Oceanic Engineering*, Volume 23:609– 624, 2003.
- [31] Charles V. Stewart. Robust Parameter Estimation in Computer Vision. *Society for Industrial and Applied Mathematics*, July 1999.
- [32] Olivier Faugeras and Quang-Tuan Luong. *The Geometry of Multiple Images*. Massachusetts Institute of Technology, 2001.
- [33] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. Technical report, Computer Science Department of University of British Columbia, January 2004.
- [34] Andrew P. Witkin. Scale-Space Filtering. *In Proceedings of the Eighth international joint conference on Artificial intelligence*, Volume 2:1019–1022, 1983.
- [35] Jan J. Koenderink. The Structure of Images. *International Journal of Computer Vision*, Volume 31:159 – 168, 1999.
- [36] Tony Lindeberg. Scale-Space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, Volume 21:225–270, 1994.
- [37] Anton Konouchine, Victor Gaganov, and Vladimir Veznevets. AMLSEAC: A New Maximum Likelihood Robust Estimator.
- [38] Marco Zuliani. *RANSAC for Dummies, With examples using RANSAC toolbox for Matlab and more...* 2008.
- [39] Martin A. Fischler and Robert C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Readings in computer vision: issues, problems, principles, and paradigms*, pages 726–740, 1987.

- [40] P. H. S. Torr and A. Zisserman. MLSEAC: A New Roboust Estimator With Application to Estimating Image Geometry. *Journal of Computer Vision and Image Understanding*, pages 138–156, July 1995.
- [41] Ondřej Chum and Jiří Matas. Matching with PROSAC - progressive sample consensus. *In Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 220–226, June 2005.
- [42] Ondřej Chum and Jiří Matas. Randomized RANSAC with $T_{d,d}$ test. *British Machine Vision Conference 2002*, pages 837 – 842, September 2002.
- [43] Myron Kayton and Walter R. Fried. *Avionics Navigation Systems*. Wiley Interscience, 1997.
- [44] Andrea Vedaldi. <http://www.vlfeat.org/~vedaldi>, 2009.
- [45] Marco Zuliani. <http://vision.ece.ucsb.edu/~zuliani>, 2009.
- [46] Jean-Yves Bouguet. <http://www.vision.caltech.edu/bouguetj>, 2009.