

Universidade de Coimbra
Faculdade de Ciências e Tecnologia
Departamento de Engenharia Electrotécnica

***TÉCNICAS DE VISÃO APLICADAS À
NAVEGAÇÃO EM ROBÓTICA MÓVEL***

TESE DE MESTRADO

Inácio de Sousa Adelino da Fonseca
Licenciado em Engenharia Electrotécnica

Coimbra
Março de 1998

Dissertação
submetida ao Departamento de Engenharia Electrotécnica
da Universidade de Coimbra
satisfazendo parcialmente os requisitos
para a obtenção do grau de
mestre

Trabalho desenvolvido sob orientação do Dr. Jorge Manuel Miranda Dias - professor
Auxiliar do Departamento de Engenharia Electrotécnica da Universidade de Coimbra

Agradecimentos

Desejo dar os meus agradecimentos:

- Dr. Jorge Dias, que me disponibilizou o seu apoio, orientação, e o material necessário para efectuar este trabalho.
- Eng. Paulo Peixoto pelo apoio e discussão sobre o problema da determinação do fluxo óptico, e ajudas sobre o sistema FreeBSD.
- Eng. Paulo Menezes, pela sua ajuda com alguns problemas de instalação do FreeBSD.
- a todo o pessoal do ISR, em especial ao Eng. Jorge Batista.
- Eng. Luís Almeida, pelo seu apoio e interesse, e pela sua ajuda nos problemas de hardware que foram aparecendo.
- a todo o pessoal do laboratório, por partilharem muitas horas de trabalho.
- Eng. Rui Cortesão, pelo seu apoio e amizade ao longo deste tempo. E pela sua ajuda na teoria de filtros de Kalman, \LaTeX , FreeBSD.
- ao ISEC, onde eu trabalho, que me deu o suporte financeiro para efectuar esta tese.
- ao PRODEP, que me financiou de 20 Dezembro de 1995 a 20 de Setembro de 1996.
- ao meu amigo Luís Marques e esposa, por aqueles pequenos detalhes.
- ao meu amigo Emílio, esposa e à pequena filhota Inês.

- à Fátima, Susana, Cristina, Nuno Cid, Pedro, ...
- a todos os meus amigos, e à minha família.

Resumo

O desenvolvimento de sistemas robóticos de elevada flexibilidade está fortemente dependente do estado de desenvolvimento dos sistemas sensoriais. Actualmente está a ser efectuado um grande esforço de investigação para que os sistemas robóticos sejam autónomos a nível das decisões de alto nível a executar. No campo dos sistemas sensoriais, a visão por computador assume papel preponderante, pois permite melhorar a autonomia do sistema tal como se prova pelos resultados experimentais apresentados neste trabalho.

Nesta tese, pretende-se apenas estudar o problema da navegação com um robot móvel, utilizando visão.

O problema da navegação pode ser estudado de diferentes perspectivas. Uma passa pelo reconhecimento de pontos de referência (“landmarks”), que definem locais a atingir ou a evitar. Eventualmente, estes pontos de referência podem ajudar à localização do robot num mapa que represente o meio ambiente. Outra das perspectivas, passa pela utilização de informação dinâmica, i.e., utilização de medidas da velocidade em imagens.

Neste trabalho optou-se pela segunda alternativa, baseando o processo de navegação no controlo das velocidades de rotação e translação do veículo, usando a informação visual. As soluções apresentadas são baseadas no fluxo óptico medido em duas câmaras, e a diferença entre ambos os fluxos medidos permite controlar a atitude do robot móvel. O objectivo principal é a obtenção de algoritmos que possam ser realizados em programas de computador mas cujo tempo de execução seja pequeno, e podendo ser utilizado na malha de controlo de um sistema de controlo. Para obter um processo ainda mais rápido, são utilizadas imagens log-polar, onde é calculado o fluxo óptico.

Uma das vantagens da utilização de imagens log-polar é a facilidade da detecção de campos de velocidades divergentes, cujo módulo é directamente proporcional ao raio me-

dido relativamente ao centro da imagem. O método de navegação descrito neste trabalho explora uma técnica de cálculo de fluxo óptico, baseado num esquema multi-resolução, de modo a obter um campo de fluxo de vectores velocidade com coerência local. São apresentados os resultados obtidos através de algoritmos baseados na restrição do brilho (“brightness constraint”) e no modelo de velocidade afim, ambos aplicados a imagens log-polar. Os algoritmos de detecção de fluxo óptico geram o sinal de erro que permite controlar a trajectória de um robot móvel dispondo de um sistema de visão activa, e que já foi utilizado em outros trabalhos [J. Dias e C. Paredes 95] e [J. Dias e C. Paredes 96].

Outro dos assuntos abordados neste trabalho, ainda no campo da navegação, passa pela detecção da posição do foco de expansão *FOE* (“Focus of Expansion”), o que permite explorar a possibilidade de controlar a posição do sensor visual segundo uma direcção de translação preferida (“Navigation by preferred direction”). Esta técnica de alinhamento do sensor permite recuperar de uma forma mais eficaz, a informação visual sobre o meio ambiente [Alberto Elfes 95]. Note-se que tipicamente, no caso das pessoas, os olhos mantêm-se alinhados com a direcção do seu movimento, e neste caso esta técnica simula este comportamento.

Conteúdo

	v
Agradecimentos	vii
Resumo	ix
1 Introdução	1
1.1 O enquadramento do trabalho	2
1.1.1 Navegação baseada em fluxo óptico normal	3
1.1.2 Medida de desempenho de algoritmos de cálculo de fluxo óptico	5
1.1.3 Transformação Log-polar	6
1.1.4 Detecção do foco de expansão - <i>FOE</i>	7
1.1.5 Detecção de obstáculos e tempo até colisão	8
1.2 A estrutura da tese	9
2 Transformação Log-polar	11
2.1 Perspectiva geométrica	11
2.2 Transformação Log-Polar	12
2.2.1 Transformações Conformais	12
2.2.2 Transformação Log-Polar genérica	15
2.2.3 Uma transformação Log-polar mais elaborada	16
2.2.4 A transformação log-polar discreta	19
2.3 Retina dinâmica (ou transformação log-polar dinâmica)	21

2.4	Campo de movimento induzido numa câmara	23
2.5	Algumas propriedades da transformação Log-Polar	25
2.5.1	Campo de movimento no plano Log-polar	26
2.6	O efeito de sub-amostragem produzido pelo log-polar	29
3	Detecção de Movimento nas Imagens	33
3.1	Campo de movimento induzido numa câmara por uma superfície planar . .	34
3.2	Cálculo do fluxo óptico	35
3.2.1	Filtragem	38
3.2.2	Métodos que utilizam o gradiente	38
3.2.3	Algoritmo de fluxo óptico normal	40
3.2.4	Algoritmo de fluxo óptico usando o modelo afim	44
3.2.5	Algoritmo de fluxo óptico de Horn e Schunck Revisto	48
3.2.6	Cálculo de fluxo óptico usando correlação	49
3.2.7	Resultados experimentais com imagens sintéticas	51
3.2.8	Medida do desempenho de algoritmos de cálculo de fluxo óptico em imagens log-polar e cartesianas	56
4	Determinação de Parâmetros do Movimento	65
4.1	Introdução	66
4.2	Estimação do foco de expansão	68
4.2.1	Equações de velocidade expressas em coordenadas polares	70
4.2.2	Detecção da direcção do foco de expansão	73
4.2.3	Detecção da posição do foco de expansão	73
4.2.4	Resultados de simulação	77
4.2.5	Resultados com o sistema experimental	78
4.2.6	Principais conclusões sobre a detecção do foco de expansão	78
4.3	Controlo do alinhamento de uma câmara	85
5	Navegação utilizando visão artificial	93
5.1	Introdução	93
5.2	Motivação	94

5.2.1	Sensores de imagens esféricos	95
5.2.2	Caso especial da utilização de um sistema de visão activa	102
5.3	Navegação com duas câmaras	109
5.4	Navegação com duas câmaras e espelhos	112
5.4.1	Espelhos como simuladores de várias câmaras	112
5.5	Resultados Simulados e Experimentais	117
5.5.1	Resultados simulados	117
5.5.2	Resultados experimentais	120
6	Conclusões e trabalho Futuro	127
6.1	Conclusões	127
6.2	Trabalho futuro	129
A	Definições e convenções	131
B	O Sistema do Robot Móvel	133
C	Cinemática dos sistemas autónomos utilizados	137
C.1	Introdução	137
C.1.1	Transformações entre referenciais	138
C.1.2	Relações de velocidade	139
C.2	Cinemática (directa e inversa)	142
C.2.1	Caso do sistema de visão activa	142
C.2.2	Caso dos Pan & Tilt	146
C.3	Matrizes “skew symmetric”	150
C.4	A influência da velocidade angular	153
D	Pormenores da realização prática	155
E	Filtros α-β e α-β-γ	163
E.1	Introdução	163
E.2	Filtro α - β	164
E.2.1	Filtro α - β - γ	165

Lista de Tabelas

2.1	Passos necessários para estudar os parâmetros dos filtros necessários à filtragem das imagens, antes da transformação log-polar.	31
3.1	Resultados para a sequência “Translating Tree” usando imagens Log-Polar	57
3.2	Resultados para a sequência “Translating Tree” usando imagens cartesianas	58
3.3	Resultados para a sequência “Diverging Tree” usando imagens Log-Polar .	59
3.4	Resultados para a sequência “Diverging Tree” usando imagens cartesianas	60
3.5	Resultados da estimação do fluxo óptico usando imagens log-polar para as sequências de translação e divergente, usando técnicas descritas em [J. Barron e D. Fleet 94]	62
3.6	Aplicação do método de Fleet à sequência divergente e de translação da árvore, usando diferentes bases para a transformação log-polar (mantendo a mesma resolução angular)	63
B.1	Parâmetros para controlar a velocidade da plataforma móvel	133

Capítulo 1

Introdução

O desenvolvimento de sistemas robóticos de elevada flexibilidade está fortemente dependente do estado de desenvolvimento dos sistemas sensoriais. Actualmente está a ser efectuado um grande esforço de investigação para que os sistemas robóticos sejam autónomos a nível das decisões de alto nível a executar. No campo dos sistemas sensoriais, a visão por computador assume papel preponderante, pois permite melhorar a autonomia do sistema tal como se prova pelos resultados experimentais apresentados neste trabalho.

Nesta tese, pretende-se apenas estudar o problema da navegação com um robot móvel, utilizando visão.

O problema da navegação pode ser estudado de diferentes perspectivas. Uma passa pelo reconhecimento de pontos de referência (“landmarks”), que definem locais a atingir ou a evitar. Eventualmente, estes pontos de referência podem ajudar à localização do robot num mapa que represente o meio ambiente. Outra das perspectivas, passa pela utilização de informação dinâmica, i.e., utilização de medidas da velocidade em imagens.

Neste trabalho optou-se pela segunda alternativa, baseando o processo de navegação no controlo das velocidades de rotação e translação do veículo, usando a informação visual. As soluções apresentadas são baseadas no fluxo óptico medido em duas câmaras, e a diferença entre ambos os fluxos medidos permite controlar a atitude do robot móvel. O objectivo principal é a obtenção de algoritmos que possam ser realizados em programas de computador mas cujo tempo de execução seja pequeno, e podendo ser utilizado na malha de controlo de um sistema de controlo. Para obter um processo ainda mais rápido,

são utilizadas imagens log-polar, onde é calculado o fluxo óptico.

Uma das vantagens da utilização de imagens log-polar é a facilidade da detecção de campos de velocidades divergentes, cujo módulo é directamente proporcional ao raio medido relativamente ao centro da imagem. O método de navegação descrito neste trabalho explora uma técnica de cálculo de fluxo óptico, baseado num esquema multi-resolução, de modo a obter um campo de fluxo de vectores velocidade com coerência local. São apresentados os resultados obtidos através de algoritmos baseados na restrição do brilho (“brightness constraint”) e no modelo de velocidade afim, ambos aplicados a imagens log-polar. Os algoritmos de detecção de fluxo óptico geram o sinal de erro que permite controlar a trajectória de um robot móvel dispondo de um sistema de visão activa, e que já foi utilizado em outros trabalhos [J. Dias e C. Paredes 95] e [J. Dias e C. Paredes 96].

Outro dos assuntos abordados neste trabalho, ainda no campo da navegação, passa pela detecção da posição do foco de expansão *FOE* (“Focus of Expansion”), o que permite explorar a possibilidade de controlar a posição do sensor visual segundo uma direcção de translação preferida (“Navigation by preferred direction”). Esta técnica de alinhamento do sensor permite recuperar de uma forma mais eficaz, a informação visual sobre o meio ambiente [Alberto Elfes 95]. Note-se que tipicamente, no caso das pessoas, os olhos mantêm-se alinhados com a direcção do seu movimento, e neste caso esta técnica simula este comportamento.

1.1 O enquadramento do trabalho

Nesta secção será dada uma breve panorâmica de algumas aplicações da visão na navegação de “robots” e o enquadramento deste trabalho neste domínio de aplicação. Concretamente, serão referenciados trabalhos que utilizam a informação visual como sinal de erro no controlo do processo de navegação, técnicas de detecção de fluxo óptico e seu desempenho, transformações *log-polar*, e a detecção do foco de expansão normalmente designado por *FOE*.

1.1.1 Navegação baseada em fluxo óptico normal

O fluxo óptico normal é uma técnica de estimação de movimento nas imagens, de tal modo que Cornelia Fermuller e Yiannis Aloimonos [C. Fermuller e Y. Aloimonos 92] usaram-na para estimar movimentos 3-D e inferir da posição do *FOE* de modo a efectuar acções de seguimento (“tracking”). No nosso caso utilizamos fluxo afim de modo a detectar a posição do *FOE* para estimar a direcção preferida de posicionamento do sensor visual.

V. Sundareswaran e P. Bouthemy [V. Sundareswaran e P. Bouthemy 94] apresentaram duas técnicas de alinhamento do eixo óptico de uma câmara, utilizando para isso informação visual. Uma das técnicas usa o foco de expansão e a outra incorpora os parâmetros do modelo afim nas equações de controlo. As técnicas permitem alinhar o eixo óptico com sucesso, sem conhecimento prévio do movimento efectuado pelo sistema onde a câmara se encontra. No nosso caso a técnica de alinhamento do sensor visual utiliza também a posição do *FOE* na imagem para inferir um melhor posicionamento do sensor visual, no entanto a nossa técnica de estimação do *FOE* é diferente da descrita neste artigo, e apresenta resultados promissores na presença de algumas restrições ao movimento do sensor visual.

David Coombs e outros [D. Coombs e H. Herman 95] descrevem um processo de detecção de obstáculos em tempo real baseado em fluxo óptico divergente e periférico, calculado através do fluxo óptico normal. Neste trabalho o fluxo óptico periférico foi usado para controlar a atitude do veículo relativamente às paredes de um corredor, e o fluxo óptico divergente para estimar o tempo até impacto, no caso de um obstáculo. No artigo os autores relatam experiências, com o robot a uma velocidade nominal de 30cm/s, efectuando trajectórias durante 20 minutos sem colidir com obstáculos. No nosso trabalho explora-se a forma de controlar a atitude do veículo através do fluxo óptico medido em imagens log-polar, apresentando resultados práticos pouco satisfatórios na presença de movimentos de rotação.

O cálculo do fluxo óptico usando apenas a restrição do brilho, conduz normalmente a algoritmos cujas soluções podem ser instáveis, uma vez que esta restrição por si só, não verifica a coerência local do movimento. Soluções mais estáveis podem ser obtidas através de algum processo que controle localmente os valores obtidos para o fluxo óptico.

Um exemplo da aplicação destas técnicas é apresentado no trabalho efectuado por Peter Nordlund e Tomas Uhlin [P. Nordlund e T. Uhlin 95]. Nesse trabalho é apresentado um algoritmo baseado em fluxo óptico que permite estimar e seguir um objecto em movimento. O sistema construído é capaz de manter o objecto no centro da imagem através do controlo de um sistema de visão activa. O método de detecção de movimento utiliza um algoritmo baseado no modelo afim para a velocidade projectada na imagem. No cálculo do movimento, a imagem é previamente estabilizada. No nosso caso não utilizamos visão activa, uma vez que o processo seria demasiado complicado de estabilizar, devido à dificuldade acrescida da utilização do movimento do robot móvel, i.e., seria neste caso bastante difícil anular o fluxo induzido devido aos movimentos do robot móvel.

Um exemplo de navegação também com alguma relevância, foi apresentado pelos autores [Ricardo e Michel 91], onde foram implementadas as seguintes tarefas: seguir uma parede, chegar perto dum objecto, virar numa esquina, etc. Nesta abordagem, é utilizado fluxo óptico, extracção de contornos de objectos, etc, e o campo potencial para realizar o controlo do veículo.

Neste momento, interessa referir que para além da abordagem proposta nesta tese, existem outras diferentes, que pretendem resolver o mesmo problema da navegação local através do uso de “landmarks”. Tais exemplos podem ser consultados nos trabalhos dos autores [Davison e Murray 95] e [Davison e Murray 97].

No controlo moderno, os algoritmos de controlo são implementados em computadores digitais, o que implica sempre uma amostragem de entidades analógicas. A teoria de sistemas de eventos discretos (“Discrete Event System”), que permite representar um sistema e a sua dinâmica com um autómato finito não determinístico apresenta uma alternativa a outros métodos de controlo discreto. Um desses exemplos é descrito em [Luca Bogoni e Ruzena Bajcsy 94], onde são apresentadas metodologias para investigar a funcionalidade em tarefas de manipulação de objectos. A descrição da funcionalidade da tarefa é feita através da teoria dos sistemas de eventos discretos, permitindo descrever várias tarefas a implementar, sendo a sua transição assegurada pelo “autómato discreto”. No nosso caso optámos por um controlo deste género através da especificação de um autómato discreto com vários estados que permite definir em concreto o controlo a efectuar, na presença de determinadas entradas do sistema.

José Santos Victor e G. Sandini [J. Santos-Victor e G. Sandini 95] propõem uma técnica de fluxo óptico divergente para efectuar navegação. Neste trabalho, o robot encontra-se equipado com duas câmaras montadas lateralmente e o controlo da atitude do robot é efectuado com base no sinal de erro obtido pela comparação dos fluxos obtidos em cada câmara. O trabalho inspira-se em estudos efectuados sobre insectos voadores que "adoptam" um esquema semelhante de navegação.

1.1.2 Medida de desempenho de algoritmos de cálculo de fluxo óptico

É importante conhecer o desempenho das técnicas de cálculo de fluxo óptico, na medida em que esse conhecimento facilita a escolha do algoritmo a utilizar. Devido ao fluxo óptico ser por natureza uma medida com ruído, convém de certa forma saber até que ponto pode ser ou não viável a sua utilização em algoritmos de controlo. Nesta ordem de ideias optámos por realizar um estudo do seu comportamento tanto em imagens cartesianas como em imagens log-polar.

Na literatura, já existiam alguns trabalhos nesta área, tais como os apresentados por J.L. Barron e outros [J. Barron e D. Fleet 94] que testaram diferentes técnicas de fluxo óptico, usando sequências de imagens sintéticas e reais, relatando diferentes medidas comparativas. As técnicas de fluxo óptico testadas, baseiam-se em métodos diferenciais, métodos de correspondência de regiões, métodos de detecção de energia e fase. Os métodos baseados em equações diferenciais de primeira ordem testados foram: *Horn and Schunck* [Horn e Schunck 81] e *Lucas and Kanade* [B. Lucas e T. Kanade 81]. Os métodos testados, que utilizam técnicas baseadas em equações diferenciais de segunda ordem foram: *Nagel* [H.H. Nagel 83, H.H. Nagel 87] e *Uras, Giroso, Verri, and Torre* [S. Uras e F. Giroso 88]. As outras técnicas testadas foram: *Anandan, Heeger, Waxman, Wu, and Bergholm* e *Fleet and Jepson* (ver [J. Barron e D. Fleet 94] para mais referências). No nosso caso utilizamos as definições apresentadas neste artigo, para entender a sua aplicação a imagens log-polar, podendo assim analisar-se o desempenho dos algoritmos que utilizem fluxo óptico medido neste tipo de imagens. Os resultados demonstram que o fluxo óptico é na realidade uma medida bastante ruidosa quer em imagens

cartesianas quer em imagens log-polar. No entanto chegamos à conclusão que em imagens log-polar esses erros podem ser atenuados através de uma boa especificação, como por exemplo, saber qual o tipo de fluxo que se pretende medir, e através deste conhecimento escolher de uma forma correcta os parâmetros da amostragem log-polar.

1.1.3 Transformação Log-polar

Inspirados no homem, bem como em outros mamíferos, muitos investigadores recorrem à visão artificial como uma forma potencial de apreender o meio ambiente, uma vez que a informação presente em imagens digitais é muito rica. No entanto para se levar a cabo determinadas tarefas, muita dessa informação é supérflua, e de modo a se eliminar esses dados muitas das vezes recorre-se a determinadas técnicas. Pode-se salientar de entre muitas, por exemplo a diminuição da resolução das imagens (pirâmides de imagens), bem como a transformação log-polar. Qualquer destas duas permite reduzir de certa forma o número de dados a processar.

Nesta ordem de ideias, nós optamos neste trabalho por estudar as propriedades da transformação log-polar para permitir suprimir alguma da informação que poderia não ter grande influência para o fim em vista.

De entre muitos autores que estudaram este tipo de transformação salientamos os que se seguem, por ser aqueles que a nosso ver apresentam maior relevância.

Carl F. R. Weiman [Carl Weiman 94], apresentaram experiências em tempo real, onde usaram um sistema baseado em visão activa binocular. Para efectuar o processamento da imagem utilizaram filtros de Gabor aplicados a imagens log-polar. Este documento é uma excelente referência sobre técnicas de transformação log-polar.

G. Sandini, C.Capurro, F.Panerai [G. Sandini e C. Capurro 95], usaram imagens log-polar para controlar activamente um sistema equipado com sensores visuais. As soluções apresentadas usam medidas de correlação, que servem para controlar os ângulos de vergência das câmaras.

No nosso caso pretendemos medir o fluxo óptico, mas através da utilização da transformação log-polar a imagens cartesianas, para permitir uma maior velocidade de processamento.

1.1.4 Detecção do foco de expansão - *FOE*

Em navegação através de visão, o conhecimento do parâmetro *FOE*, pode ser utilizado para determinar e controlar o nosso próprio movimento no espaço. Esta detecção é importante, porque assim é possível estimar alguns dos parâmetros de velocidade da câmara. Outra possibilidade é a utilização do conhecimento da posição do *FOE* para alinhar o sensor visual com o vector de velocidade de translação, optimizando assim a capacidade de detecção de fenómenos transitórios na frente da câmara.

M. Tistarelli, E. Grosso and G. Sandini [M. Tistarelli e E. Grosso 90], apresentaram um método para determinar a posição do *FOE* usando uma técnica de mínimos quadrados para determinar a intercepção das rectas obtidas da elongação dos vectores de fluxo óptico. Neste trabalho não foram apresentados resultados experimentais.

Rami Guissin, Shimon Ullman [R. Guissin e S. Ullman 91], apresentaram um método para determinar a posição do *FOE* onde é utilizada uma técnica de procura directa no campo de fluxo óptico obtido das imagens. O método utiliza uma técnica de procura 1D para seleccionar a direcção da recta que passa pelo centro da imagem e pelo *FOE*. É possível utilizar o algoritmo para movimentos com/sem restrições da câmara. No caso de movimentos sem restrições é necessário des-rotacionar o campo de velocidades de modo a cancelar a velocidade observada no centro da imagem. Esta técnica aparentemente obtém boas performances para casos simulados.

Michal Irani, Benny Rousso e Shmuel Peleg [M. Irani e B. Rousso 94], apresentaram um método para determinar a posição do *FOE* que se baseia no cálculo do movimento efectuado por uma região da imagem. O movimento calculado nessa região é então usado para anular o fluxo de rotação da imagem, de tal forma que as regiões detectadas aparecem então como estacionárias. As imagens obtidas, teoricamente, estão apenas afectadas de movimentos de translação. O método usado utiliza algoritmos baseados em modelos de velocidade afim, e não existe à partida, conhecimento do movimento efectuado.

Yiannis Aloimonos e Zoran Duric [Y. Aloimonos e Z. Duric 94], apresentaram um método para determinar o *FOE* utilizando fluxo óptico normal para restringir a translação do observador. Para resolver o problema das rotações, foi apresentado neste artigo uma versão modificada do algoritmo apresentado por Horn and Weldon, 1987.

No nosso caso a detecção do *FOE* é feita através do fluxo afim, utilizando um método de pesquisa no campo de fluxo óptico. Não foi resolvido o problema das rotações na totalidade, uma vez que o algoritmo funciona sobre o efeito duma restrição, que no nosso caso é verificada devido ao modo de locomoção do robot móvel.

1.1.5 Detecção de obstáculos e tempo até colisão

No nosso trabalho não abordamos este tema, no entanto fica em aberto uma futura implementação de algoritmos que permitam detectar obstáculos e o tempo até colisão através de informação visual. No entanto estes dois conceitos têm a nosso ver alguma relevância no desenvolvimento de tarefa de navegação, pelo que optámos por descrever brevemente alguns dos trabalhos já realizados, para uma maior informação do leitor.

Massimo Tistarelli e Giulio Sandini [M. Tistarelli e G. Sandini 93], apresentaram um artigo onde estimam o tempo até impacto utilizando imagens log-polar. No entanto esta técnica apresenta alguma instabilidade devido ao ruído introduzido no cálculo do fluxo óptico, que advém do uso de uma técnica diferencial de segunda ordem.

Nicola Ancona e Tomaso Poggio [N. Ancona e T. Poggio 93], usaram um esquema baseado em fluxo óptico para detectar a expansão e rotação entre imagens, de modo a obter assim uma estimativa do tempo até colisão. O método foi utilizado em imagens reais com sucesso.

Thomas Cord e Dirk Pallmer [T. Cord e D. Pallmer 94], apresentaram um método para estimação da profundidade que requer um movimento controlado da câmara ao longo do seu eixo óptico. As sequências de imagens são analisadas de modo a calcular os vectores translação de pontos característicos.

P. Fornland [P. Fornland 95], apresentou um método de detecção de obstáculos no qual a câmara movimenta-se paralelamente ao chão. Os parâmetros da câmara estão linearmente relacionados com as derivadas de primeira ordem da sequência de imagens. O movimento é estimado através do método RANSAC [Roth, Gerhard e Levine 92].

1.2 A estrutura da tese

A presente dissertação é composta por seis capítulos, incluindo este, onde é descrito o trabalho efectuado, bem como o conteúdo dos capítulos que se seguem.

No **capítulo 2**, são apresentados os modelos de projecção utilizados para representar as câmaras, e as técnicas de transformação log-polar. Será efectuada uma descrição das propriedades do log-polar, bem como a relação entre as coordenadas utilizadas. Este capítulo introduz os fundamentos necessários à compreensão do trabalho apresentado nos capítulos que se seguem.

No **capítulo 3**, são descritos os algoritmos de detecção de movimento, especialmente aqueles que se baseiam em modelos de gradiente. É introduzido neste capítulo um algoritmo baseado no fluxo óptico normal, através da introdução de um pré- e pós- processamento e dum sistema de votação que melhora os resultados obtidos através do algoritmo de fluxo óptico normal. São também apresentados os algoritmos de Horn e Schunck [Horn e Schunck 81], os algoritmos baseados no modelo afin e um algoritmo de cálculo de fluxo óptico baseado em técnicas de correlação cujos resultados são mais robustos, mas cujo processo em si é mais lento. Neste capítulo é também introduzido o modelo afin para o plano log-polar. De seguida, é efectuado um estudo do desempenho de alguns algoritmos de cálculo de fluxo óptico no plano log-polar e cartesiano. As medidas efectuadas são idênticas às utilizadas por J.L. Barron e outros [J. Barron e D. Fleet 94].

No **capítulo 4** é apresentado um método para determinar alguns parâmetros do movimento, tal como a detecção da posição do *FOE*, que permite identificar a direcção de translação do sensor visual.

Finalmente com base na detecção da posição do *FOE* é apresentado um método para efectuar um processo de alinhamento e estabilização duma câmara montada num robot móvel em movimento puramente translacional - detecção da direcção preferida -.

No **capítulo 5**, é dada particular ênfase à implementação/construção de algoritmos de navegação, à dinâmica do sistema, e ao uso do fluxo óptico como medida para cálculo do erro usado para controlo da dinâmica do sistema. É também aqui discutida a realização da aplicação em tempo real.

Finalmente, no **capítulo 6** serão apresentadas as conclusões deste trabalho.

Capítulo 2

Transformação Log-polar

Neste capítulo, descreve-se a forma de aplicar a transformação log-polar a imagens contínuas e discretas, fornecendo assim os fundamentos necessários para uma correcta compreensão do trabalho que é apresentado nos restantes capítulos.

2.1 Perspectiva geométrica

Neste modelo a imagem obtida pelas câmaras é representada pelo plano I e o ponto de perspectiva central pelo ponto O (ver figura 2.1). Por definição, a projecção de um ponto 3D $P(X, Y, Z)$ no plano I, é expressa matematicamente por:

$$\begin{aligned} u &= f_x \frac{X}{Z} \\ v &= f_y \frac{Y}{Z} \end{aligned} \tag{2.1}$$

onde f_x e f_y correspondem à distância focal afectada do factor de escala horizontal e vertical. As imagens representadas na fig. 2.2 são exemplos de imagens geradas por alguns modelos de perspectiva e correspondentes a variantes do modelo expresso pelas equações (2.1).

2.2 Transformação Log-Polar

Os sistemas de visão baseados em imagens com resolução espacial homogêneas, têm dificuldades em fornecer respostas em tempo real, quando se encontram na presença de um meio ambiente mutável. A realização de um sistema visual com um comportamento reactivo, necessita de alguma selectividade espacial nas imagens. Esta selectividade pode ser obtida através duma amostragem espacialmente variável. Nessa situação, a imagem é “separada” numa zona central de grande informação espacial - a fovea - e numa zona cuja resolução espacial decresce à medida que o raio ao centro aumenta. O modelo mais convincente, para este tipo de comportamento, é a transformação log-polar.

Basicamente, um modelo deste género pode ser obtido através duma transformação que transforma os pontos medidos relativamente ao centro da imagem cartesiana, de uma forma proporcional à sua distância a esse centro, e também proporcional à resolução angular medida relativamente a um eixo. Esta transformação na sua forma mais simples resume-se a uma transformação entre coordenadas cartesianas para coordenadas polares, seguida da aplicação da função logaritmo à coordenada radial ρ ($\rho \in \mathbb{R}^+$, $\theta \in [0, 2\pi[$, $a, b > 1$). Este tipo de transformação reduz significativamente, a quantidade de dados a tratar, e pode desta forma ser um meio de gerar sinais de “feedback” visuais, que possam ser usados em navegação (ver o Anexo A para as notações utilizadas nas secções seguintes).

2.2.1 Transformações Conformais

Muitos dos problemas de engenharia, podem ser tratados e resolvidos através de métodos de análise complexa. A transformação conformal insere-se neste domínio da matemática e é um método bastante utilizado na engenharia em geral para resolver problemas da teoria de potencial em duas dimensões, pois permite simplificar as resoluções.

Segundo as definições de [Erwin Kreyszig 79], uma *transformação diz-se conformal* ou que *preserva ângulos*, se esta *mantiver o valor dos ângulos entre curvas orientadas, quer em valor, quer em direcção, isto é, as imagens de duas quaisquer curvas orientadas que se intersectem num ponto, mantêm a mesma relação geométrica angular após a*

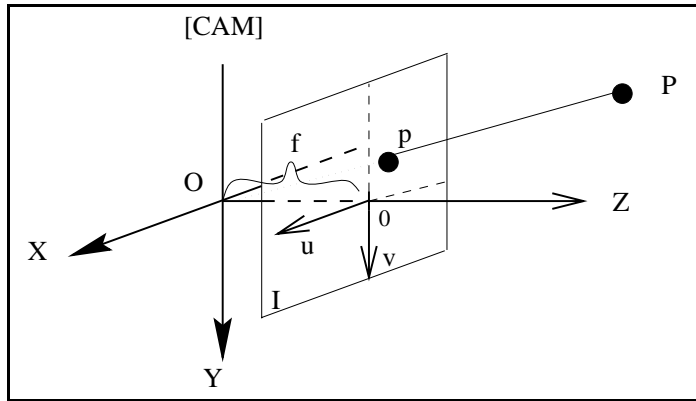


Figura 2.1: Modelo de perspectiva geométrica



Figura 2.2: Imagens sintéticas geradas por (da esquerda para a direita): modelo geométrico, modelo ortográfico, modelo semi-esférico. As respectivas equações do modelo são dadas por: $u = X.S_u$, $v = Y.S_v$ para o modelo ortográfico e por $u = X_s.S_u$, $v = Y_s.S_v$ para o modelo semi-esférico, onde $X_s = X \frac{r}{\sqrt{(X^2+Y^2+Z^2)}}$, $Y_s = Y \frac{r}{\sqrt{(X^2+Y^2+Z^2)}}$ e $Z_s = Z \frac{r}{\sqrt{(X^2+Y^2+Z^2)}}$, sendo r o raio da esfera.

transformação.

Esta transformação contém propriedades interessantes para o domínio deste trabalho, na medida em que nos permite verificar que tipo de informação pode ser preservada pela transformação, e que se expressa pelos seguintes teoremas:

Teorema 2.1 *Se uma função complexa $\omega = f(z) = u(x, y) + i.v(x, y)$ onde $z = x + i.y$ está definida num domínio D do plano complexo Z , então para cada ponto de D existe um ponto correspondente no plano W . Desta forma, existe uma transformação do plano Z para o plano W . A função $f(z)$ define uma transformação conformal, excepto nos pontos onde $f'(z)$ é zero, se e só se $f(z)$ é uma função analítica.*

Teorema 2.2 *A função $f(z) = u(x, y) + i.v(x, y)$ onde $z = x + i.y$ é analítica num domínio D , se as funções $u(x, y)$ e $v(x, y)$ de variáveis reais x e y tiverem derivadas contínuas de primeira ordem, que satisfaçam as equações de Cauchy-Riemann num domínio D , isto é,*

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} \quad e \quad \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x} \quad (2.2)$$

ou se a função estiver expressa em coordenadas polares, $z = \rho e^{i\theta}$, e $f(z) = u(\rho, \theta) + i.v(\rho, \theta)$, então as equações de Cauchy-Riemann serão dadas por:

$$\frac{\partial u}{\partial \rho} = \frac{1}{\rho} \frac{\partial v}{\partial \theta} \quad e \quad \frac{\partial u}{\partial \theta} = -\rho \frac{\partial v}{\partial \rho} \quad (2.3)$$

A demonstração destes teoremas sai do domínio deste tese, mas poderá ser consultada em [Erwin Kreyszig 79].

Um exemplo de transformação muito utilizada em visão por computador (transformações $2D$ para $2D$) corresponde à transformação log-polar. A transformação log-polar utilizada no desenvolvimento desta tese, pela forma como foi definida não apresenta a propriedade de transformação conformal, mas este facto não é muito preocupante, porque não se pretende medir ângulos entre objectos. A sua definição pode ser dada como ($\omega = \eta + i\theta$):

$$z = x + iy = f^{-1}(\omega) = a^\eta \cdot \cos(\theta) + i \cdot a^\eta \cdot \sin(\theta). \quad (2.4)$$

e na próxima secção abordaremos mais em pormenor este assunto. No entanto, pode-se definir uma transformação log-polar com a propriedade conformal, i.e., se $z = \rho e^{i\theta}$, então

a sua transformação ω no domínio log-polar é

$$\omega = \eta + i.\alpha = f(z) = \log_a(z) = \log_a(\rho) + i.\frac{\theta}{\ln a} \quad (2.5)$$

e a sua transformação inversa é descrita por,

$$z = x + i.y = f^{-1}(\omega) = a^\eta.\cos(\alpha.\ln a) + i.a^\eta.\sin(\alpha.\ln a). \quad (2.6)$$

2.2.2 Transformação Log-Polar genérica

Assuma-se então um ponto (u, v) da imagem, expressa em coordenadas cartesianas. O par (ρ, θ) especifica o mesmo ponto, mas em coordenadas polares ¹. O ponto (u, v) no plano cartesiano, pode ser representado por um número complexo z , e pelas suas coordenadas polares (ρ, θ) , através de:

$$z = u + i.v = \rho.e^{i.\theta} \quad (2.7)$$

com

$$\begin{aligned} \rho &= \sqrt{u^2 + v^2} \\ \theta &= \tan^{-1}\left(\frac{v}{u}\right) \end{aligned} \quad (2.8)$$

A transformação logarítmica complexa (ou transformação log-polar) é, desta forma, definida como

$$w = \ln(z) \quad (2.9)$$

para cada $z \neq 0$, onde $\Re(w) = \ln(\rho)$ e $\Im(w) = \theta + 2k\pi$. Para $z \neq 0$ existe uma singularidade que fisicamente num sensor de imagem corresponderia a uma zona em que os “pixels” teriam de ser infinitesimalmente pequenos e infinitamente numerosos no centro do sensor - uma configuração que fisicamente não é realizável - . Para excluir a periodicidade da parte imaginária, é assumido $k = 1$ de modo a restringir os valores possíveis de $\Im(w)$ ao intervalo $[0, 2\pi[$. Ao plano $(\ln(\rho), \theta)$ também se chama de plano w . Facilmente se demonstra que uma rotação centrada na origem do plano z resulta num deslocamento na direcção θ no plano w e que uma multiplicação por um factor de escala no plano z , resulta num deslocamento na direcção $\ln(\rho)$ no plano w .

¹Assumindo as notações indicadas no Apêndice A.

Esta definição pode ser extensível a outros casos, de modo a permitir trabalhar imagens com diferentes “aspect ratio”, ou quando é necessário efectuar amostragens com diferentes espaçamentos nos eixos horizontal e vertical. Esta transformação poderá ser extremamente útil para imagens com diferentes resoluções nos dois eixos, caso se pretenda transformar toda a imagem cartesiana para o plano log-polar. Neste caso a fórmula geral para a transformação é dada por (não se trata de uma transformação conformal):

$$\begin{cases} u(\eta, \theta) = a^\eta \cos(\theta) \\ v(\eta, \theta) = b^\eta \sin(\theta) \end{cases} \quad (2.10)$$

As constantes a e b controlam a evolução da transformação na direcção radial, e para valores particulares podem ser obtidos respectivamente transformações em círculos ou em elipses para os casos de $a = b$ ou $a \neq b$. Esta é a transformação log-polar utilizada nas implementações práticas descritas neste trabalho.

Como a transformação log-polar vai corresponder a uma sub-amostragem da imagem é muito comum simular transformações log-polar com o uso de pequenas células receptoras de forma circular centradas no ponto de amostragem log-polar. O valor do pixel no plano log-polar é neste caso dado através da média dos “pixels” que pertencem à respectiva célula circular. Na próxima secção será apresentado um processo semelhante ao descrito. Na maioria dos casos o cálculo desta média é omitido para permitir um maior velocidade de processamento.

2.2.3 Uma transformação Log-polar mais elaborada

Um dos métodos propostos por [M. Blackburn e H. Nguyen 94] e [Carl Weiman 94] permite que a transformação log-polar seja realizada através de círculos concêntricos radiais. Em cada círculo concêntrico existem círculos de amostragem angular, sendo o valor do pixel de cada amostra no plano log-polar dado pela média dos “pixels” que se encontram dentro do respectivo círculo para a respectiva amostra angular. Por outras palavras, a posição dos pontos, espacialmente variante é obtida através duma amostragem regular e duma grelha de círculos concêntricos com N_θ amostras em cada círculo radial. Para cada um destes círculos concêntricos com distância constante ao centro da imagem cartesiana, são efectuadas N_θ amostras angulares, e para cada uma destas amostras, o valor

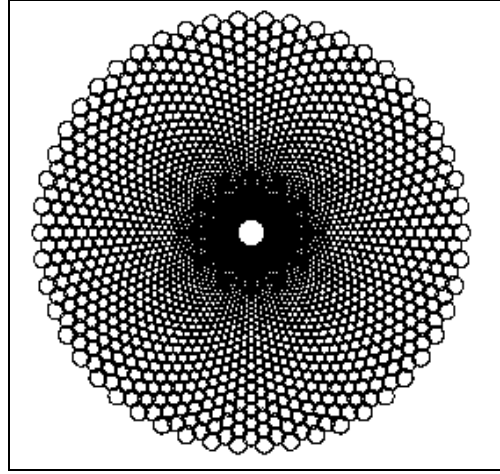


Figura 2.3: Representação gráfica de uma transformação log-polar, através do uso de uma técnica de círculos concêntricos. Neste caso o número de amostras angulares é de 50.

de amostragem é obtido dos “pixels” que se encontram num círculo centrado na posição da amostra angular. A figura 2.3, representa graficamente este tipo de amostragem log-polar.

O número de amostras em cada círculo é constante, conforme ilustrado na Fig. 2.3. Para um dado N_θ , a componente radial ρ_ζ do ζ -ésimo círculo é dada por:

$$\rho_\zeta = \rho_{fovea} \left(1 + \frac{\pi}{\sqrt{3}N_\theta} \right)^\zeta = \rho_{fovea} a^\zeta \quad (2.11)$$

onde $\zeta = 0, \dots, N_\rho \Leftrightarrow 1$ e ρ_{fovea} indica o raio mínimo do círculo quando $\zeta = 0$, e é escolhido de forma a evitar a sobreposição dos círculos de amostras angulares consecutivas (neste caso é uma condição universal), e o raio dos círculos $R_\zeta \geq 1$ (para o caso $\zeta = 0$). O valor de ρ_{fovea} é assim dado por:

$$\rho_{fovea} \geq 3 \frac{N_\theta}{2\pi} \quad (2.12)$$

Sobre o mesmo círculo concêntrico são tiradas N_θ amostras espaçadas periodicamente por:

$$P_\zeta = \frac{2\pi\rho_\zeta}{N_\theta} \quad (2.13)$$

e para cada uma destas amostras está associado um círculo cujo raio é dado por:

$$R_\zeta = \frac{2\pi\rho_\zeta}{3N_\theta} \quad (2.14)$$

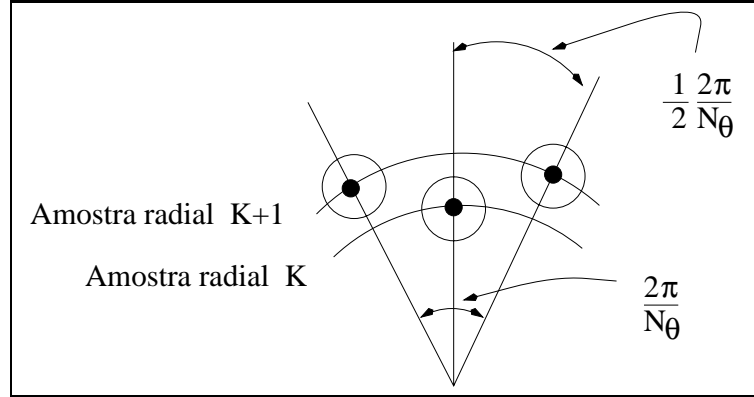


Figura 2.4: Entre dois círculos concêntricos consecutivos as amostras angulares são deslocadas de meio período.

que é igual para as N_θ amostras (para o mesmo ρ). Entre dois círculos concêntricos consecutivos as amostras angulares são deslocadas de meio período (veja-se Fig. 2.4), porque com este esquema, mesmo na hipótese de se usar um número de amostras angulares baixo, conseguir-se-á mesmo assim uma boa representação da imagem original (pode-se considerar que seria como se tivesse usado o dobro das amostras angulares), onde a fórmula matemática é dada por:

$$\theta_\zeta = \frac{2\pi}{N_\theta} \left(\gamma + \frac{\text{odd}(\zeta)}{2} \right) \quad (2.15)$$

com $\gamma = 0, \dots, N_\theta \Leftrightarrow 1$.

A transformação para os índices do plano Log-Polar é assim dada por:

$$\begin{aligned} \rho_{\log_\zeta} &= \log_a \frac{\rho_\zeta}{\rho_{fovea}} \\ \theta_{\log_\zeta} &= \frac{\theta_\zeta}{\frac{2\pi}{N_\theta} \left(1 + \frac{\text{odd}(\zeta)}{2} \right)} \end{aligned} \quad (2.16)$$

Alguns investigadores, propuseram diferentes soluções para efectuar a transformação log-polar, de modo a conseguir eliminar a singularidade presente na origem. Por exemplo, [A.S. Roger e and E.L. Schwartz 90], propuseram uma transformação log-polar sem a inclusão da fovea.

Outros investigadores, como por exemplo [Carl Weiman 94], propuseram uma forma eficiente de ultrapassar o problema da singularidade. Segundo esses investigadores, a zona da fovea, representada na Fig. 2.3 pelo círculo central, seria transformada através duma

estrutura rectangular uniformemente preenchida com células, cujas áreas seriam iguais à área da célula log-polar mais próxima (o que neste caso seria para $\zeta = 0$). A este processo eles chamaram de “*Isotropic central fill*”. Um dos benefícios, desta abordagem, advém do facto, de assim se poder utilizar coordenadas lineares no centro da imagem, com igual espaçamento, permitindo também a efectuação de reconhecimento de padrões através dum processo invariante à translação. No entanto a descontinuidade, presente na transição entre a zona de coordenadas (u, v) e a zona de coordenadas $(\log_a(\rho), \theta)$, levanta um problema à transformação da imagem nessa transição, por exemplo a translação e a rotação. As equações para este processo são semelhantes às equações anteriormente apresentadas na zona periférica, e para a região central definida pela restrição $\rho < \rho_{fovea}$ são utilizadas as coordenadas cartesianas.

2.2.4 A transformação log-polar discreta

Para os casos práticos da aplicação do log-polar em visão por computador, o plano W será referido como o plano imagem log-polar. As variáveis N_θ e N_ρ definem a resolução radial e angular da imagem log-polar. A transformação log-polar, será apenas aplicada a toda a imagem excepto à zona da fovea. Mas primeiro comecemos com a transformação geral. No caso discreto, a cada pixel (ζ, γ) no plano log-polar, corresponde um pixel (u, v) no plano imagem. O valor da intensidade do pixel (ζ, γ) no plano log-polar é obtido através da aplicação de uma máscara gaussiana, cuja máscara de dimensão $n \times n$, deve ser centrada no pixel (u, v) (veja-se Apêndice D). Definindo o valor incremental angular:

$$\Delta\theta = \frac{2\pi}{N_\theta} \quad (2.17)$$

a transformada é baseada nas seguintes equações:

$$\begin{cases} u = \rho_{fovea} \cdot a^\zeta \cdot \cos(\Delta\theta \cdot \gamma) \\ v = \rho_{fovea} \cdot b^\zeta \cdot \sin(\Delta\theta \cdot \gamma) \end{cases} \begin{cases} \theta = \arctan\left(\left(\frac{b}{a}\right)^\zeta \tan(\Delta\theta \cdot \gamma)\right) \\ \rho = \rho_{fovea} \sqrt{a^{2\zeta} \cos^2(\Delta\theta \cdot \gamma) + b^{2\zeta} \sin^2(\Delta\theta \cdot \gamma)} \end{cases} \quad (2.18)$$

O valor de ρ_{fovea} é uma constante para permitir que não exista sobreposição de “pixels” para $\zeta = 0$ (neste caso ζ, γ apenas assumem valores inteiros). Este valor deve ser escolhido de forma a que o perímetro do círculo para $\zeta = 0$ tenha no mínimo N_θ amostras, pois só

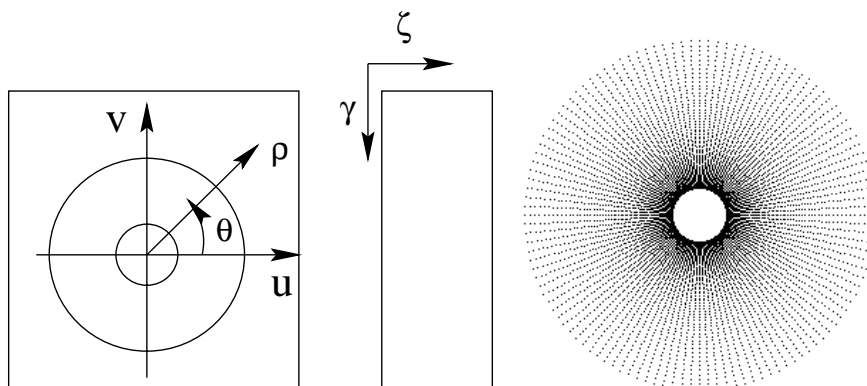


Figura 2.5: Transformação Log-polar e a representação da estrutura de amostragem para um caso em que $a = b$.

assim se garante que existem N_θ “pixels” diferentes para N_θ amostras angulares (no caso em que $a = b$). Este valor limiar é dado por:

$$\rho_{fovea} \geq \frac{N_\theta}{2\pi}. \quad (2.19)$$

A aplicação desta restrição, resulta num círculo central, na imagem do plano cartesiano, cuja informação lá contida não é transformada para o plano log-polar, tal como pode ser visto através da Fig. 2.5. As dimensões deste círculo podem ser controladas através do valor do número de amostras segundo a direcção θ (N_θ), ou através dos valores da base da função logaritmo. As constantes a , b controlam a evolução das amostras segundo a direcção radial ρ . Da aplicação das equações 2.18 para $\gamma = 0, \dots, N_\theta \Leftrightarrow 1$ e $\zeta = 0, \dots, N_\rho \Leftrightarrow 1$, resulta a estrutura de amostragem representada na Fig. 2.5 (para $a = b$). Outro pormenor importante, é o facto de a resolução na direcção θ do log-polar ser sempre constante para todos os valores radiais, o que resulta numa imagem no plano log-polar com mais informação sobre a zona central da imagem cartesiana, mas decrescendo para a periferia. No caso descrito acima, são usadas as variáveis N_θ , a , b , para controlar a amostragem.

Para o caso de uma transformação log-polar com $a = b$, pode-se definir $\rho_{fovea} \leq \rho \leq \frac{\rho_{max}}{2}$, que permite indicar qual a zona da imagem cartesiana que será transformada, onde ρ_{fovea} é o raio da zona central não amostrada (fovea) e ρ_{max} a dimensão da imagem cartesiana em “pixels” (assumindo uma imagem quadrangular). A transformação de

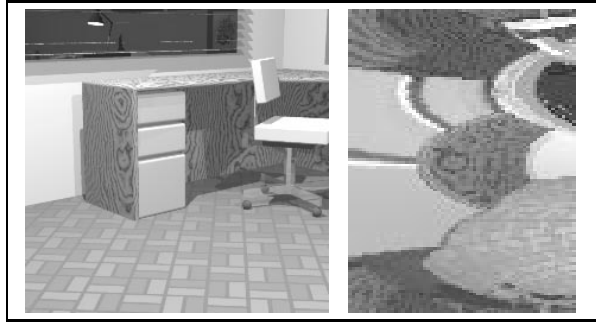


Figura 2.6: Exemplo duma imagem log-polar. Na esquerda encontra-se a imagem cartesiana, e à direita a imagem log-polar resultante da amostragem pela estrutura representada na Fig. 2.5

coordenadas polares (ρ, θ) para coordenadas log-polares (ζ, γ) é dada por:

$$\zeta = \log_a \left(\frac{\rho}{\rho_{fovea}} \right) \quad (2.20)$$

conforme pode ser visto na Fig. 2.5. A base a do logaritmo é dada na dependência de N_ρ por

$$\rho_{fovea} \cdot a^{N_\rho - 1} = \frac{\rho_{max}}{2} \Leftrightarrow a = \exp \left(\frac{1}{N_\rho - 1} \ln \left(\frac{\rho_{max}}{2\rho_{fovea}} \right) \right) \quad (2.21)$$

Dada uma imagem digital quadrada $I(u, v)$ contendo $\rho_{max} \times \rho_{max}$ “pixels”, pode-se verificar que para um dado $\rho_{fovea} \leq \rho \leq \frac{\rho_{max}}{2}$, a transformação não tem uma correspondência unívoca. A Fig. 2.6 mostra um exemplo da transformação log-polar para o que foi descrito anteriormente. Na Fig. 2.7 apresenta-se também exemplos da amostragem log-polar quando $a = b$, $a > b$ e $a < b$.

2.3 Retina dinâmica (ou transformação log-polar dinâmica)

Na secção 2.2.2 foi definida a transformação log-polar (eq. 2.10). Suponha-se agora, que é necessário efectuar a transformação, não centrada no ponto $(0, 0)$, mas sim, centrada num ponto genérico (u_i, v_i) . À primeira vista, o problema parece simples de resolver, uma vez que apenas é necessário efectuar a transformação centrada no ponto dado. No

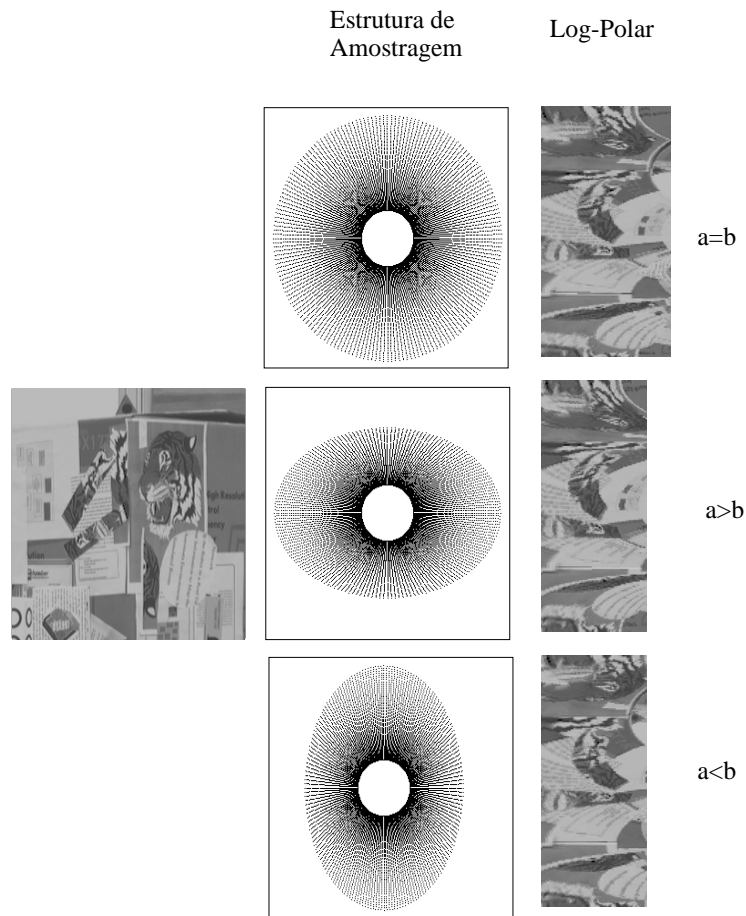


Figura 2.7: Exemplos de transformações Log-Polar, quando $a = b$, $a > b$, e $a < b$ (veja-se eq. 2.10).

entanto, o que nos interessa, é obter a nova imagem log-polar utilizando apenas a imagem log-polar obtida da transformação centrada num ponto (u_0, v_0) , para isso é necessário desenvolver a transformação entre os dois planos log-polar.

Definição 2.1 *Uma retina centrada num ponto (u_0, v_0) , é equivalente a uma transformação log-polar centrada em (u_0, v_0) . O termo “retina dinâmica” representa a mudança do ponto central, onde a transformação log-polar é efectuada.*

Suponha-se, então, que se tem a retina original centrada no ponto (u_0, v_0) , com variáveis independentes (η, θ) , e que se pretende obter uma retina centrada no ponto (u_1, v_1) com variáveis independentes (η', θ') . Para a primeira retina (ou transformação log-polar)

tem-se:

$$\begin{aligned} u(\eta, \theta) &= a^\eta \cos(\theta) + u_0 \\ v(\eta, \theta) &= a^\eta \sin(\theta) + v_0 \end{aligned} \quad (2.22)$$

e para a segunda retina (ou transformação log-polar):

$$\begin{aligned} u(\eta', \theta') &= a^{\eta'} \cos(\theta') + u_1 \\ v(\eta', \theta') &= a^{\eta'} \sin(\theta') + v_1 \end{aligned} \quad (2.23)$$

Tendo em conta que o novo ponto central (u_1, v_1) pode ser descrito através da equação (2.22), para um determinado (η_0, θ_0) através de:

$$\begin{aligned} a^{\eta_0} \cos(\theta_0) &= u_1 \Leftrightarrow u_0 \\ a^{\eta_0} \sin(\theta_0) &= v_1 \Leftrightarrow v_0 \end{aligned} \quad (2.24)$$

a partir de (2.22) e (2.23) e tendo em conta que se pretende obter a nova imagem log-polar (η', θ') em função da imagem log-polar (η, θ) , resolvendo para (η, θ) obtemos:

$$\begin{aligned} \eta &= \frac{1}{2} \log_a \left[\left(a^{\eta'} \cos(\theta') + a^{\eta_0} \cos(\theta_0) \right)^2 + \left(a^{\eta'} \sin(\theta') + a^{\eta_0} \sin(\theta_0) \right)^2 \right] \\ \theta &= \tan^{-1} \left(\frac{a^{\eta'} \sin(\theta') + a^{\eta_0} \sin(\theta_0)}{a^{\eta'} \cos(\theta') + a^{\eta_0} \cos(\theta_0)} \right) \end{aligned} \quad (2.25)$$

Em conclusão, no domínio contínuo, esta propriedade permite transformar uma imagem log-polar que tenha sido obtida através de um determinado ponto central da imagem cartesiana numa outra imagem log-polar com outro ponto central, sem ser necessário reutilizar a imagem cartesiana. Em termos práticos sendo as imagens discretas, a utilização da retina dinâmica oferece alguns problemas de difícil resolução.

2.4 Campo de movimento induzido numa câmara

Neste momento é importante estudar o efeito resultante nas imagens a nível da velocidade projectada, quando uma câmara se movimenta num meio ambiente estático. Para isso descreve-se de seguida o comportamento do modelo para a projecção da velocidade no plano imagem, dados os parâmetros de movimento numa câmara.

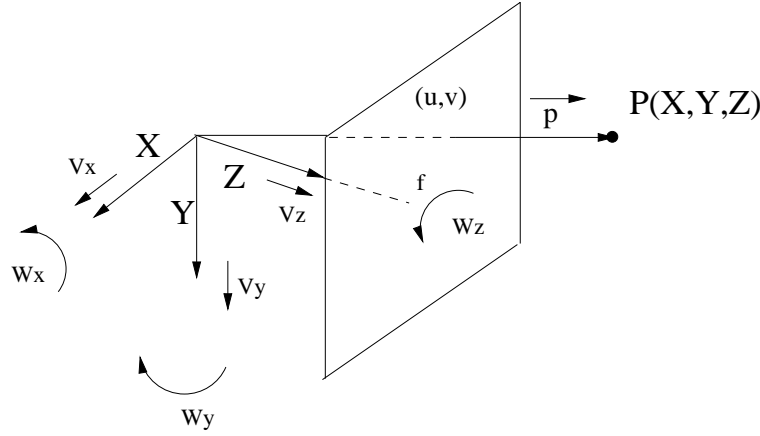


Figura 2.8: Orientação das velocidades 3D da câmara

Assuma-se assim, um modelo de perspectiva geométrica conforme descrito na secção 2.1, com a equação (2.1) que relaciona os pontos 3D com os pontos 2D. A câmara possui duas componentes de movimento: a translacional $\vec{v}_t = [v_x, v_y, v_z]^T$ e a rotacional $\vec{\omega} = [\omega_x, \omega_y, \omega_z]^T$ (veja-se a Fig. 2.8). Devido ao movimento da câmara, o ponto P da cena com um vector $\vec{p} = [X, Y, Z]^T$ associado, aparenta um movimento relativo à câmara com translação $\Leftrightarrow \vec{v}_t$ e com rotação $\Leftrightarrow \vec{\omega}$. Neste caso a velocidade 3D, $\vec{v} = [\dot{X}, \dot{Y}, \dot{Z}]^T$ pode ser expressa por:

$$\vec{v} = \Leftrightarrow \vec{v}_t \Leftrightarrow \vec{\omega} \times \vec{p} \quad (2.26)$$

Este é o vector de movimento 3D, que gera o vector de movimento 2D no plano imagem. Na realidade, estes vectores serão projectados no plano imagem dando origem a uma imagem do campo de movimento (o fluxo óptico teórico). Para projectar os vectores de velocidade 3D é necessário utilizar a equação 2.1 diferenciada em ordem a t :

$$\begin{aligned} \dot{u} &= \frac{\dot{X} f_x}{Z} \Leftrightarrow \frac{f_x \dot{Z} X}{Z^2} \\ \dot{v} &= \frac{\dot{Y} f_y}{Z} \Leftrightarrow \frac{f_y \dot{Z} Y}{Z^2} \end{aligned} \quad (2.27)$$

e efectuando a substituição de \vec{v} na equação, obtêm-se o fluxo induzido na imagem devido ao próprio movimento da câmara:

$$\begin{bmatrix} \dot{u} & \dot{v} \end{bmatrix}^T = \frac{1}{Z} \cdot A \cdot \vec{v}_t + B \cdot \vec{\omega} \quad (2.28)$$

onde a sua expansão é dada por:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} \Leftrightarrow f_x & 0 & u \\ 0 & \Leftrightarrow f_y & v \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} + \begin{bmatrix} \frac{vu}{f_y} & \Leftrightarrow \left(f_x + \frac{u^2}{f_x} \right) & v \frac{f_x}{f_y} \\ \left(f_y + \frac{v^2}{f_y} \right) & \Leftrightarrow \frac{vu}{f_x} & \Leftrightarrow u \frac{f_y}{f_x} \end{bmatrix} \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} \quad (2.29)$$

Obviamente, esta expressão representa o fluxo óptico analítico no plano imagem, onde o primeiro termo representa o efeito do movimento de translação, enquanto que o segundo é o fluxo induzido devido ao movimento de rotação da câmara. É importante referir, que o efeito associado à translação depende da profundidade Z (apenas pode ser obtida a direcção do movimento de translação em vez das suas componentes), e portanto, no caso da velocidade ser conhecida, pode-se estimar a profundidade relativamente ao referencial da câmara.

2.5 Algumas propriedades da transformação Log-Polar

A transformação log-polar apresenta algumas propriedades interessantes, que torna este processo numa ferramenta bastante útil para aplicar no campo da visão por computador. Algumas dessas propriedades estão ligadas ao efeito de rotação, e à mudança do factor de escala (relativamente ao centro da imagem). As imagens da Fig. 2.9 ilustram a propriedade da rotação, que no plano log-polar corresponde a um deslocamento vertical.

Mas, no entanto, a mais importante é a projecção do campo de movimento no plano log-polar, quando o observador efectua movimentos de rotação ou de translação. O que nos interessa aqui, é observar o que acontece ao campo de fluxo óptico para alguns tipos de movimento. É importante referir que através das equações obtidas na secção 2.4 facilmente se prova as afirmações que se seguem, através da restrição dos pontos tridimensionais ao tipo de superfície em questão.

Note-se que o fluxo óptico gerado no plano imagem numa câmara, quando esta efectua movimentos de translação na direcção do seu eixo óptico, é divergente, e o seu módulo aumenta com a distância ao centro da imagem. Neste caso, existe um ponto designado de foco de expansão (FOE), que está situado no centro do fluxo óptico divergente na Fig. 2.10 (coincidente com o centro da imagem). Este campo de vectores, no plano

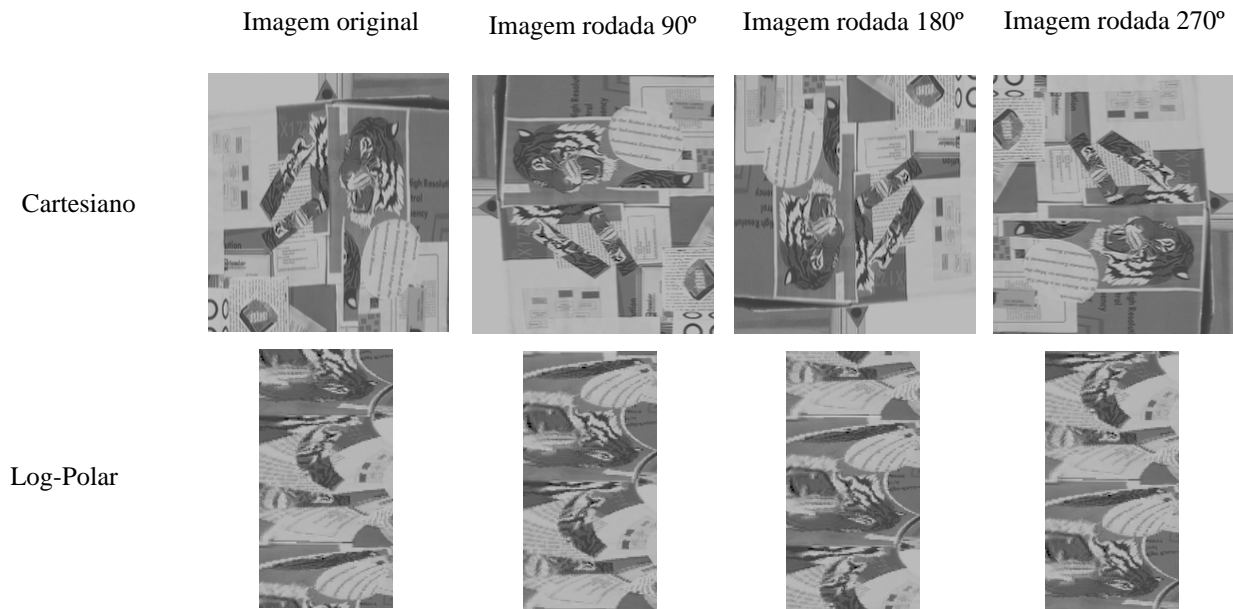


Figura 2.9: O efeito da rotação. A imagem cartesiana é rodada, e o conseqüente efeito no plano log-polar é uma translação na direcção γ .

log-polar será constituído por um campo de vectores com módulo variável, mas com a mesma orientação angular, tal como pode ser visto na Fig. 2.10, e se for efectuado um histograma na direcção ζ resultará numa função horizontal constante para cada um dos valores de γ .

Campos de vectores de fluxo óptico deste tipo, são bastante importantes para este trabalho.

2.5.1 Campo de movimento no plano Log-polar

Os métodos de amostragem com resolução espacial variável, possuem algumas características interessantes. Uma dessas, em particular, é o estudo do cálculo do fluxo óptico (veja-se Cap. 3). Na realidade, com a amostragem log-polar, é possível detectar de uma forma mais eficaz algumas formas de fluxo óptico que com outras técnicas normais não é possível. Portanto, nesta secção, vai-se efectuar o estudo das relações entre o fluxo no plano cartesiano e no plano log-polar.

Para relacionar o fluxo óptico $(\frac{\partial \eta(t)}{\partial t}, \frac{\partial \theta(t)}{\partial t})$ no plano log-polar com o fluxo no plano

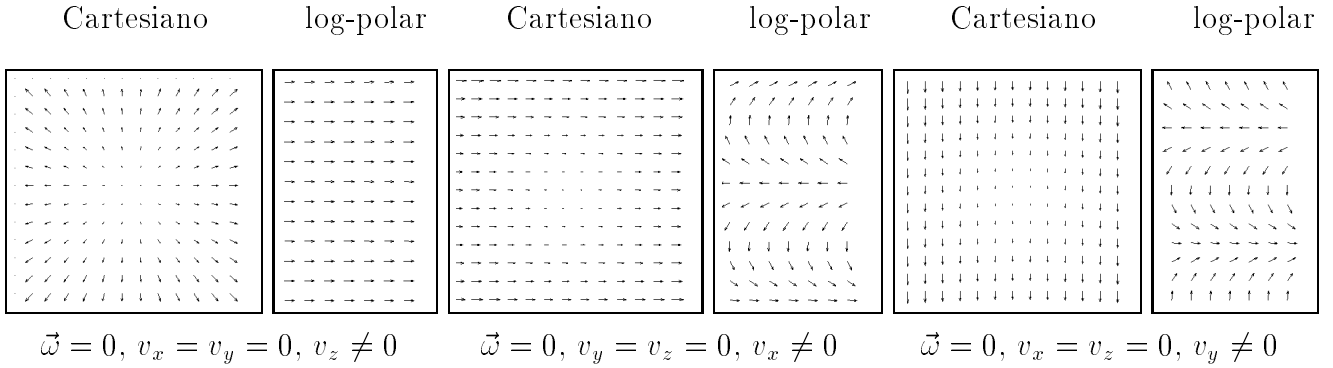


Figura 2.10: Relação entre os fluxos ópticos dos dois planos, para diferentes tipos de movimento. Da esquerda para a direita, é representado o fluxo no plano imagem seguido do seu equivalente no plano log-polar. As duas imagens da esquerda representam o fluxo óptico divergente, que no plano log-polar é constante. As imagens do centro, e da direita, correspondem a um fluxo horizontal e vertical crescente respectivamente, que no plano log-polar dá origem a vectores com o mesmo módulo, mas com orientações diferentes.

cartesiano $(\frac{\partial u(\eta, \theta, t)}{\partial t}, \frac{\partial v(\eta, \theta, t)}{\partial t})$, deve-se diferenciar a equação 2.10 em ordem a t , que resulta na seguinte equação, que representa a relação entre os dois campos de vectores

$$\begin{bmatrix} \frac{\partial u(\eta, \theta, t)}{\partial t} \\ \frac{\partial v(\eta, \theta, t)}{\partial t} \end{bmatrix} = \begin{bmatrix} a^\eta \ln(a) \cos(\theta) & \Leftrightarrow a^\eta \sin(\theta) \\ b^\eta \ln(b) \sin(\theta) & b^\eta \cos(\theta) \end{bmatrix} \begin{bmatrix} \frac{\partial \eta(t)}{\partial t} \\ \frac{\partial \theta(t)}{\partial t} \end{bmatrix} \quad (2.30)$$

na realidade a matriz J representa o Jacobiano da matriz da equação 2.10:

$$J = \begin{bmatrix} \frac{\partial u(\eta, \theta, t)}{\partial \eta} & \frac{\partial u(\eta, \theta, t)}{\partial \theta} \\ \frac{\partial v(\eta, \theta, t)}{\partial \eta} & \frac{\partial v(\eta, \theta, t)}{\partial \theta} \end{bmatrix} \quad (2.31)$$

Usando a notação $\dot{u} = \frac{\partial u(\eta, \theta, t)}{\partial t}$, e efectuando a transformação inversa da eq. 2.30 resulta na relação do campo de vectores do plano cartesiano com o plano log-polar:

$$\begin{bmatrix} \dot{\eta} \\ \dot{\theta} \end{bmatrix} = \frac{1}{\cos(\theta)^2 \ln(\frac{a}{b}) + \ln(b)} \begin{bmatrix} \frac{\cos(\theta)}{a^\eta} & \frac{\sin(\theta)}{b^\eta} \\ -\frac{\sin(\theta) \ln(b)}{a^\eta} & \frac{\cos(\theta) \ln(a)}{b^\eta} \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} \quad (2.32)$$

e como pode ser visto, a detecção do fluxo óptico no plano log-polar é igual ao fluxo do plano imagem, mas dividido pela distância radial (a^η e b^η) relativamente ao centro da

imagem (a matriz representa apenas uma transformação de rotação). Esta propriedade é de grande utilidade, porque permite detectar campos de fluxos crescentes radialmente a partir do centro da imagem, enquanto que com outras técnicas será muito difícil.

No caso em que os parâmetros que controlam a evolução da componente radial são iguais ($a = b$), obtemos as seguintes relações:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = a^\eta \begin{bmatrix} \ln(a) \cos(\theta) & \Leftrightarrow \sin(\theta) \\ \ln(a) \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} \dot{\eta} \\ \dot{\theta} \end{bmatrix} \quad (2.33)$$

e

$$\begin{bmatrix} \dot{\eta} \\ \dot{\theta} \end{bmatrix} = \frac{1}{a^\eta} \begin{bmatrix} \frac{\cos(\theta)}{\ln(a)} & \frac{\sin(\theta)}{\ln(a)} \\ \Leftrightarrow \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} \quad (2.34)$$

É importante verificar a diferença entre uma transformação polar e log-polar. Rescrevendo as equações indicadas acima, para o caso polar:

$$\begin{bmatrix} \dot{\rho} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ \Leftrightarrow \frac{\sin(\theta)}{\rho} & \frac{\cos(\theta)}{\rho} \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} \quad (2.35)$$

Analisando a equação (2.35), verifica-se que o fluxo óptico radial apresenta algumas diferenças, quando comparado com o caso log-polar. Certamente a utilização da função logaritmo na direcção radial é a origem deste facto. Na secção 4.2 veremos esta diferença, de uma forma mais explícita.

Como em situações práticas, se utiliza a versão discreta das equações (2.33) e (2.34) (para $a = b$) então teremos:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \rho_{fovea} \cdot a^\zeta \begin{bmatrix} \ln(a) \cos(\Delta\theta \cdot \gamma) & \Leftrightarrow \Delta\theta \cdot \sin(\Delta\theta \cdot \gamma) \\ \ln(a) \sin(\Delta\theta \cdot \gamma) & \Delta\theta \cdot \cos(\Delta\theta \cdot \gamma) \end{bmatrix} \begin{bmatrix} \dot{\zeta} \\ \dot{\gamma} \end{bmatrix} \quad (2.36)$$

e

$$\begin{bmatrix} \dot{\zeta} \\ \dot{\gamma} \end{bmatrix} = \frac{1}{\rho_{fovea} a^\zeta} \begin{bmatrix} \frac{\cos(\Delta\theta \cdot \gamma)}{\ln(a)} & \frac{\sin(\Delta\theta \cdot \gamma)}{\ln(a)} \\ \Leftrightarrow \frac{\sin(\Delta\theta \cdot \gamma)}{\Delta\theta} & \frac{\cos(\Delta\theta \cdot \gamma)}{\Delta\theta} \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} \quad (2.37)$$

2.6 O efeito de sub-amostragem produzido pelo log-polar

Nesta secção, analisa-se a transformação log-polar no domínio da frequência e pretende-se conhecer à partida as especificações dos eventuais filtros que serão necessários para efectuar a transformação log-polar discreta de uma forma conveniente. Assume-se que a forma conveniente é evitar fenómenos de “aliasing”.

Para efectuar o estudo, assumam-se à partida dois sinais bidimensionais contínuos. O primeiro sinal $I(u, v)$, que é o sinal original, encontra-se representado no domínio cartesiano. O segundo sinal $I_2(\eta, \theta)$ foi obtido do sinal contínuo $I(u, v)$ através da transformação log-polar contínua definida pela equação 2.10 (com $b = a > 1$).

Para o sinal $I(u, v)$ o seu espectro $FT\{I\}(\omega_u, \omega_v)$ é dado por:

$$FT\{I\}(\omega_u, \omega_v) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} I(u, v) \cdot e^{-j\omega_u u} \cdot e^{-j\omega_v v} du dv. \quad (2.38)$$

onde $\omega_u = 2\pi f_u$ e $\omega_v = 2\pi f_v$. Utilizando a relação entre $I(u, v)$ e $I_2(\eta, \theta)$ e efectuando-se uma mudança de variável, a mesma transformada pode ser expressa por:

$$FT\{I\}(\omega_u, \omega_v) = \int_{-\infty}^{+\infty} \int_0^{2\pi} I_2(\eta, \theta) \cdot e^{-j\omega_u a^\eta \cos(\theta)} \cdot e^{-j\omega_v a^\eta \sin(\theta)} |J| d\theta d\eta \quad (2.39)$$

onde

$$|J| = \left| \begin{bmatrix} a^\eta \ln(a) \cos(\theta) & \Leftrightarrow a^\eta \sin(\theta) \\ a^\eta \ln(a) \sin(\theta) & a^\eta \cos(\theta) \end{bmatrix} \right| = a^{2\eta} \ln a > 0 \quad e \quad I_2(\eta, \theta) = I(a^\eta \cos(\theta), a^\eta \sin(\theta))$$

que após as respectivas simplificações, pode ser representada por:

$$FT\{I\}(\omega_u, \omega_v) = \ln a \int_{-\infty}^{+\infty} a^{2\eta} \left(\int_0^{2\pi} I_2(\eta, \theta) \cdot e^{-j\omega_u a^\eta \cos(\theta)} \cdot e^{-j\omega_v a^\eta \sin(\theta)} d\theta \right) d\eta \quad (2.40)$$

que corresponde a um espectro do sinal com coordenadas cartesianas.

Por outro lado, suponham-se agora o sinal $I_2(\eta, \theta)$ que foi obtido do sinal contínuo $I(u, v)$ através da transformação log-polar contínua. Aplicando a transformada de Fourier indicada na eq. (2.38) ao sinal $I_2(\eta, \theta)$, sabendo que $\eta \in]-\infty, +\infty[$ e $\theta \in [0, 2\pi[$, o espectro do sinal em coordenadas polares é dado por:

$$FT\{I_2\}(\omega_\eta, \omega_\theta) = \int_{-\infty}^{+\infty} \int_0^{2\pi} I_2(\eta, \theta) \cdot e^{-j\omega_\eta \eta} \cdot e^{-j\omega_\theta \theta} d\theta d\eta \quad (2.41)$$

Comparando as expressões 2.41 e 2.40, verifica-se que existe um factor de escala $a^{2\eta} \ln a$ que advém da amostragem logarítmica.

Como o objectivo é pretender conhecer qual o filtro que se deve aplicar de modo a evitar “aliasing” e tendo em conta que de facto o processo de amostragem logarítmica possui papel preponderante, vamos estudar em separado o efeito da amostragem logarítmica e angular.

Por simplicidade, para o estudo da amostragem logarítmica, considere-se um sinal unidimensional discreto. Assuma-se desta forma um sinal discreto original $x_{original}[n]$ com $n = 0, \dots, N \Leftrightarrow 1$ elementos, amostrado através duma função logarítmica $n = \rho_{fovea} a^\eta$ com $\rho_{fovea} > 0$ e $\eta = 0, \dots, N_\rho \Leftrightarrow 1$. Após a aplicação desta transformação obtém-se o sinal pretendido $x[\eta] = x_{original}[\rho_{fovea} a^\eta]$.

No entanto, para uma melhor compreensão, considere-se ainda sem perda de generalidade, que $x_{original}[n]$ foi obtido dum sinal periódico contínuo $x(t)$ com período T_0 e que permite representar completamente o sinal $x(t)$, i.e.,

$$x_{original}[n] = \int_0^{T_0} x(t) \delta \left(t \Leftrightarrow \frac{T_0 n}{N} \right) dt \quad (2.42)$$

onde $\delta()$ representa a função delta de Dirac. O sinal discreto e equivalente a uma amostragem logarítmica terá deste modo a seguinte forma:

$$x[\eta] = \int_0^{T_0} x(t) \delta \left(t \Leftrightarrow \frac{T_0 \rho_{fovea} a^\eta}{N} \right) dt \quad (2.43)$$

Este novo sinal pode não representar correctamente o sinal $x(t)$ devido ao espaçamento das amostras não cumprir o teorema da amostragem [Alan Oppenheim 79]. Este facto terá maior probabilidade de ocorrência quando η tende para $N_\rho \Leftrightarrow 1$, pois para as duas amostras finais é que existe um maior espaçamento. Este espaçamento define a frequência de amostragem de valor mais baixo e portanto a mais crítica. Por outro lado, dado o espaçamento entre amostras ser variável, pode-se considerar que a amostragem logarítmica possui uma frequência de amostragem variável com a posição da amostra.

Para determinar as especificações do filtro vai-se considerar inicialmente um novo sinal $x_1[m]$, obtido de $x_{original}[n]$ através de uma amostragem logarítmica, mas que contenha na totalidade todos os elementos de $x_{original}[n]$. Para tal é necessário definir os parâmetros da nova função de amostragem. Estes parâmetros estão directamente relacionados com

<i>Tipo de sinal</i>	<i>Expressão do sinal</i>	<i>Número de amostras</i>	<i>Variável de controlo</i>	<i>Filtro</i>
original	$x_{original}[n] = \int_0^{T_0} x(t) \delta \left(t \Leftrightarrow \frac{T_0 n}{N} \right) dt$	N	n	
Contém na totalidade o sinal original	$x_1[m] = \int_0^{T_0} x(t) \delta \left(t \Leftrightarrow \frac{T_0 \rho_{fovea} b^m}{N} \right) dt$	M	m	
Desejado, mas obtido de $x_1[m]$	$x_2[\eta] = \int_0^{T_0} x(t) \delta \left(t \Leftrightarrow \frac{T_0 \rho_{fovea} b^{\eta \rho}}{N} \right) dt$	N_ρ	η	$\frac{\pi}{M}$
desejado	$x[\eta] = \int_0^{T_0} x(t) \delta \left(t \Leftrightarrow \frac{T_0 \rho_{fovea} a^\eta}{N} \right) dt$	N_ρ	η	$\frac{\pi}{M}$

Tabela 2.1: Passos necessários para estudar os parâmetros dos filtros necessários à filtração das imagens, antes da transformação log-polar.

o espaçamento das duas amostras finais, uma vez que estas devem estar presentes no sinal $x_1[m]$. Considere-se então uma nova função de amostragem dada por exemplo, por $n = \rho_{fovea} b^m$ onde $m = 0, \dots, M \Leftrightarrow 1$, isto para um determinado b e M . Os valores de b e M podem ser obtidos através da restrição na periferia da função $\rho_{fovea} b^m$, i.e.,

$$\begin{aligned} \rho_{fovea} b^{M-1} &= N \Leftrightarrow 1 \\ \rho_{fovea} b^{M-2} &= N \Leftrightarrow 2 \end{aligned} \quad (2.44)$$

obtendo-se assim para b e M respectivamente $b = \frac{N-1}{N-2}$ e $M = \log_b \left(\frac{N-1}{\rho_{fovea}} \right) + 1$. Após a aplicação desta transformação obtém-se o sinal pretendido $x_1[m] = x_{original}[\rho_{fovea} b^m]$, que representa na totalidade $x(t)$.

Desta forma o sinal $x_1[m]$ pode ser dado por:

$$x_1[m] = \int_0^{T_0} x(t) \delta \left(t \Leftrightarrow \frac{T_0 \rho_{fovea} b^m}{N} \right) dt \quad (2.45)$$

Para obter o sinal $x[\eta]$ a partir de $x_1[m]$ é necessário efectuar uma interpolação de N_ρ seguida duma decimação de M , isto porque o sinal $x_1[m]$ tem M amostras e pretende-

se obter no final um sinal com N_ρ amostras. Por esta razão deve-se substituir m pela seguinte relação:

$$m = \eta \frac{M}{N_\rho} \Leftrightarrow m = \eta p \quad \eta = 0 \cdots N_\rho \Leftrightarrow 1 \quad (2.46)$$

pelo que o sinal $x_2[\eta]$ será assim dado por:

$$x_2[\eta] = \int_0^{T_0} x(t) \delta \left(t \Leftrightarrow \frac{T_0 \rho_{fovea} b^{\eta p}}{N} \right) dt \quad (2.47)$$

Através das equações 2.43 e 2.47 verificamos que ambas representam o mesmo sinal se a condição $a^\eta = b^{p\eta}$ se verificar. Nessa situação o filtro que se deve aplicar depende do factor de decimação, que neste caso era de M , i.e., a frequência de corte do filtro deve ser de $\frac{\pi}{M}$. A tabela 2.1 apresenta os passos tomados para se obter esta conclusão.

Neste momento, é importante voltar agora ao caso log-polar em 2D, mas primeiro deve-se estudar a amostragem angular da transformação log-polar.

Para a amostragem angular, o maior factor de decimação acontece na periferia. Nessa zona a distância entre amostras angulares é de $\simeq \frac{2\pi}{N_\theta} \rho_{fovea} a^{N_\rho - 1}$ “pixels”, o que origina um factor de decimação de:

$$K = \frac{2\pi}{N_\theta} \rho_{fovea} a^{N_\rho - 1} \Leftrightarrow 1 \quad (2.48)$$

e para se evitar “aliasing”, a frequência de corte do filtro deve ser de $\frac{\pi}{K}$.

Deste modo, para o caso log-polar, a frequência de corte é dada por $\frac{\pi}{N}$, onde $N = \max(K, M)$. Esta é uma das restrições mais severas, devido à amostragem espacial variante, no entanto é possível desenhar vários filtros para diferentes pontos de amostragem radial (i.e. a frequência de corte do filtro depende da distância radial).

É importante referir, que o sinal resultante terá componentes de frequência acima de $\frac{\pi}{2}$, por isso deve-se ter cuidado, quando se utilizam técnicas de diferenciação. Neste caso deve-se filtrar o sinal final com um filtro com frequência de corte $\frac{\pi}{2}$, perdendo-se assim alguma informação.

Capítulo 3

Detecção de Movimento nas Imagens

Neste capítulo, é abordado o cálculo do fluxo óptico numa sequência de imagens temporais apresentando um algoritmo para o seu cálculo, bem como a sua extensão para imagens log-polar.

Como o tema principal desta tese é a navegação num espaço específico utilizando visão dinâmica para testar e desenvolver o método de navegação, é necessário ter um estimador do sinal que é resultado do fluxo óptico na sequência temporal de imagens que se vão adquirindo. Interessa portanto estudar quais os métodos mais adequados e expeditos para este objectivo. Esta foi a principal motivação para se efectuar este estudo, tendo também em conta que se pretende explorar imagens log-polar. Este último ponto é decisivo, porque o cálculo do fluxo óptico através de imagens log-polar é mais rápido do que com imagens cartesianas, e consequentemente podem ser obtidos algoritmos com tempos de execução muito pequenos. Por conseguinte, é importante abordar o problema tendo em conta a necessidade de encontrar respostas para:

- Das técnicas de cálculo de fluxo óptico, qual é a mais indicada para usar em imagens log-polar?
- Será necessário utilizar algum processo adicional para compensar o facto do cálculo do fluxo ser baseado em técnicas diferenciais?
- Será possível estimar o movimento usando apenas fluxo óptico normal?

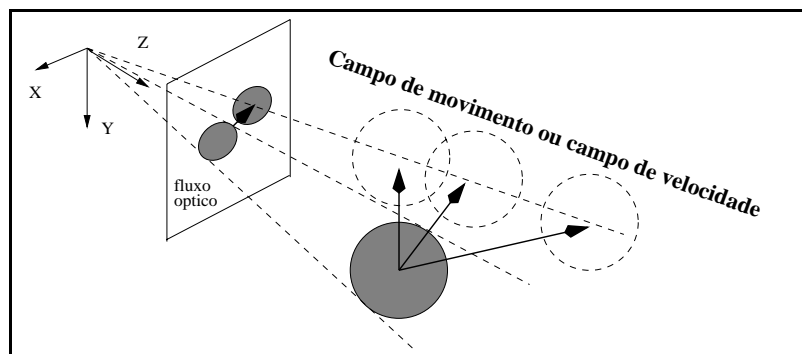


Figura 3.1: Relação entre o campo de movimento e o fluxo óptico

- Como serão os resultados influenciados pelos parâmetros do log-polar?

Nas secções deste capítulo tentaremos dar respostas a estas questões.

Consideráveis esforços têm sido efectuados no desenvolvimento de técnicas diferenciais para o cálculo de fluxo óptico (ou estimação da velocidade projectada usando imagens). Com base no conhecimento deste fluxo é possível desenvolver técnicas para o seguimento de objectos que se movimentam no espaço, determinação de parâmetros de movimento (“egomotion”) ou de navegação com base na informação dinâmica obtida a partir de imagens. Estes problemas são difíceis de resolver, uma vez que a velocidade projectada no plano imagem é a projecção da velocidade tridimensional do objecto, que em alguns casos não pode ser totalmente estimada, como podemos verificar pela Fig. 3.1.

3.1 Campo de movimento induzido numa câmara por uma superfície planar

Reporte-se o leitor à secção 2.4, onde foi previamente introduzida a relação entre o campo de movimento projectado numa imagem e o induzido por uma câmara em movimento. Nesta secção é feita uma extensão a essa relação para um conjunto de pontos tridimensionais que pertencem a uma superfície planar.

Os resultados obtidos são interessantes para o cálculo de parâmetros de movimento e navegação, pois as muitas superfícies podem ser representadas como conjunto de pequenos

planos. Por exemplo, a superfície das paredes do corredor, onde se pretende efectuar a navegação, podem modelar-se como planos, ou conjunto de planos.

Assuma-se então que alguns dos pontos projectados na imagem pertencem a um plano. Todos esses pontos satisfazem uma restrição geométrica correspondente à equação dum plano $Z = C + D.X + E.Y$, que pode ser expressa em termos das coordenadas da imagem, usando a equação 2.1 como:

$$\frac{1}{Z} = \alpha + \beta.u + \gamma.v \quad (3.1)$$

e substituindo este resultado na equação 2.29 obtêm-se:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} a_1 + a_2.u + a_3.v + a_c.u^2 + b_c.u.v \\ b_1 + b_2.u + b_3.v + a_c.u.v + b_c.v^2 \end{bmatrix} \quad (3.2)$$

onde

$$\begin{aligned} a_1 &= \Leftrightarrow \alpha f_x V_x \Leftrightarrow f_x W_y & b_1 &= \Leftrightarrow \alpha f_y V_y + f_y W_x \\ a_2 &= \Leftrightarrow \beta f_x V_x + \alpha V_z & b_2 &= \Leftrightarrow \beta f_y V_y \Leftrightarrow \frac{f_y}{f_x} W_z \\ a_3 &= \Leftrightarrow \gamma f_x V_x + \frac{f_x}{f_y} W_z & b_3 &= \Leftrightarrow \gamma f_y V_y + \alpha V_z \\ a_c &= \Leftrightarrow \frac{W_y}{f_x} + \beta V_z & b_c &= \gamma V_z + \frac{W_x}{f_y} \end{aligned} \quad (3.3)$$

A equação 3.2 descreve o fluxo de movimento na imagem representado parametricamente através de oito parâmetros, e gerado por um movimento tridimensional dum plano na cena. Estes parâmetros são o suporte para os métodos “afim”, “translacional” e “movimento de uma superfície plana” para a determinação do fluxo óptico no plano imagem, como será apresentado nas secções que se seguem. O modelo descrito pela equação 3.2, pode ser utilizado com um método de mínimos quadrados para estimar o fluxo óptico, e assim o problema da abertura pode ser ultrapassado através deste modelo específico. Este assunto será explicado mais tarde.

3.2 Cálculo do fluxo óptico

Para estimar o fluxo óptico em imagens discretas é necessário ter alguns cuidados como veremos mais à frente nesta secção. Existem já alguns trabalhos neste campo, tal como o trabalho feito por Peixoto [Peixoto 95], onde através do cálculo do fluxo afim se estuda a coerência do movimento em sequências de imagens. Este trabalho tem como ponto alto, a

detecção de dois tipos de movimento (movimento dum objecto na presença de movimento de fundo), e a sua possível remoção. No nosso caso não pretendemos detectar vários movimentos distintos, mas sim utilizar o modelo afim para estimar o foco de expansão como veremos na secção 4.2.

Convém neste momento referir que existem pelo menos quatro grandes estratégias algorítmicas para a determinação do fluxo óptico, que são baseados:

1. ***no gradiente*** : as técnicas baseadas no gradiente assumem que a intensidade da imagem se conserva entre imagens. Esta técnica é a utilizada neste trabalho, e será abordada em detalhe nesta secção.
2. ***na correlação*** : considera que a distribuição local da intensidade das imagens se mantêm entre estas. Normalmente este cálculo é efectuado sobre duas imagens consecutivas separadas temporalmente por um intervalo de tempo Δt muito pequeno. A técnica consiste na determinação da posição relativa de um pixel, ponto ou característica entre duas imagens temporalmente consecutivas. O fluxo óptico será o vector deslocamento entre as posições consecutivas entre as duas imagens. Para achar este vector é necessário efectuar uma procura de entre alguns candidatos da segunda imagem e achar o melhor de todos. Alguns investigadores [D. Marr e Poggio 79], [Barnard e Thompson 80] e [Burt, Xen e Xu 83], usaram as seguintes estratégias para achar esse candidato:
 - os candidatos pertencem a uma área onde se considera que no máximo existiu um deslocamento de x e y “pixels” (velocidades máximas admissíveis). A esta região chama-se a janela de procura.
 - O melhor candidato é dado através da máxima correlação que usa uma área à volta do pixel da imagem no tempo t , e a área correspondente em redor do pixel candidato da imagem $t + \Delta t$. Pelo facto do uso de uma janela este método considera que a distribuição local da intensidade das imagens se mantêm constante.
3. ***em energia espaço temporal*** : neste processo o cálculo do fluxo óptico é baseado na resposta de filtros sintonizados para determinadas velocidades, considerando-se

assim que as frequências espaço temporais, estão relacionadas com a velocidade dos “pixels” através da seguinte fórmula:

$$\omega_t = \omega_u \dot{u} + \omega_v \dot{v}. \quad (3.4)$$

onde as variáveis \dot{u} , \dot{v} , referem as duas componentes ortogonais da velocidade 2D, ω_u , ω_v são as componentes ortogonais da frequência espacial do estímulo visual, e ω_t é a frequência temporal correspondente. A equação 3.4 foi deduzida através da transformada de Fourier aplicada a duas imagens numa sequência temporal, considerando que existe conservação da intensidade dos “pixels” entre as imagens [D. Hegger 87].

Segunda a equação 3.4, se o sinal for visto no espaço ω_u , ω_v , ω_t , temos a equação de um plano. A velocidade (\dot{u}, \dot{v}) pode ser determinada se o plano for determinado. Heeger [D. Hegger 87] utiliza doze filtros de Gabor sintonizados em frequências centrais que estão contidas dentro de um cilindro no espaço ω_u , ω_v , ω_t , onde o eixo do cilindro é o eixo ω_t . Cada filtro está sintonizado para dar a sua máxima resposta para um determinado valor do fluxo óptico, e desta forma através do uso de diferentes filtros podem-se obter respostas diferentes, que dependem da frequência espaço temporal das imagens. O melhor dos doze filtros, dá origem ao resultado escolhido, definindo assim os parâmetros do plano.

4. **na fase :** é um processo semelhante à técnica da energia espaço temporal, pelo facto de utilizar filtros sintonizáveis (neste caso passa-banda), no entanto a velocidade (u, v) é definida em termos do valor da fase para a saída do filtro (normalmente técnicas de passagem por zero).

No nosso caso interessa-nos em especial os métodos baseados no gradiente, pois permitem à partir algoritmos com um menor tempo de execução, se bem que, eventualmente, com um maior erro de cálculo. Este último ponto pode ser decisivo pois pode tornar instável a solução encontrada para o problema em causa. Para responder a esta questão, no final desta secção será apresentado um estudo realizado para conhecer o desempenho dos algoritmos de fluxo óptico em imagens cartesianas e log-polar. Neste estudo são apresentados resultados de comparação entre quatro métodos de cálculo do fluxo óptico.

3.2.1 Filtragem

Quando se pretende estimar o fluxo óptico numericamente, é extremamente importante que se efectue uma filtragem da imagem discreta para eliminar algum do ruído. O processo de filtragem pode ser realizado para seleccionar uma parte do espectro do sinal e/ou para remover o ruído de alta frequência que normalmente existe nas imagens discretas. Considerando que a imagem contém a densidade espectral concentrada nas baixas frequências, pode-se remover a maior parte do ruído efectuando uma filtragem passa-baixo. Conforme explicado na secção 2.6, quando é necessário decimar um sinal, deve-se evitar o fenómeno de “aliasing” através da limitação do espectro do sinal. Como a intenção é utilizar imagens log-polar, neste caso apenas será efectuada a filtragem espacial para garantir que quando se faz a amostragem log-polar o espectro do sinal é de banda limitada. No entanto, efectuar uma filtragem espacial consome muito tempo, para se reduzir este tempo, pode-se verificar que apenas é necessário filtrar uma zona da imagem à volta do pixel que vai ser amostrado pelo processo log-polar.

3.2.2 Métodos que utilizam o gradiente

Dos métodos de cálculo de fluxo óptico existentes, os métodos diferenciais são os mais ruidosos, mas permitem obter algoritmos com tempos de execução muito pequenos. Por conseguinte, a maioria dos algoritmos propostos e utilizados em processos de tempo real, exploram a diferenciação dos valores de cinzento para cada pixel, de modo a efectuar a estimação do campo de velocidade. Estes métodos baseiam-se na observação da mudança dos valores de cinzento da imagem $I(u, v, t)$, e são chamados de métodos baseados no gradiente. A solução para as equações resultantes, fornecem a estimação para o campo de velocidades do plano imagem, também chamado de fluxo óptico do campo de velocidade da imagem.

Suponha-se que (u, v) é a posição na imagem dum ponto da cena no instante t , e que (\dot{u}, \dot{v}) é a velocidade desse ponto projectada na imagem. O mesmo ponto (u, v) após um tempo Δt mover-se-á para uma nova posição $(u + \dot{u}.\Delta t, v + \dot{v}.\Delta t)$. Assumindo a

conservação do brilho da imagem nesse ponto implica que

$$I(u, v, t) = I(u + \dot{u} \cdot \Delta t, v + \dot{v} \cdot \Delta t, t + \Delta t) \quad (3.5)$$

Expandindo o termo da direita através da série de Taylor e representando $I_u = I_u(u, v, t) = \frac{\partial I(u, v, t)}{\partial u}$, $I_v = I_v(u, v, t) = \frac{\partial I(u, v, t)}{\partial v}$, $I_t = I_t(u, v, t) = \frac{\partial I(u, v, t)}{\partial t}$, a equação 3.5 pode ser reduzida a:

$$I_u \dot{u} \Delta t + I_v \dot{v} \Delta t + I_t \Delta t \Leftrightarrow Err(\Delta t^2) = 0 \quad (3.6)$$

Esta restrição assume que o brilho da imagem $I(u, v, t)$ “sofrerá” uma pequena variação durante o intervalo de tempo Δt entre as duas imagens consecutivas, e que $I(u, v, t)$ é uma função diferenciável - ou seja obedece às condições de expansibilidade em série de Taylor. O termo $Err(\Delta t^2)$ representa o erro existente neste cálculo, pois apenas se utilizam termos até à primeira derivada na expansão da série de Taylor. Assumindo que este erro é desprezável, então obteremos a equação 3.7 que representa a designada “restrição do brilho” (brightness constraint):

$$\vec{\nabla} I \cdot \vec{v} + I_t = 0 \quad (3.7)$$

onde $\vec{v} = [\dot{u}, \dot{v}]^T$ é o fluxo óptico no instante de tempo t no ponto (u, v) e $\vec{\nabla} I = [I_u, I_v]^T$ é o gradiente no mesmo ponto.

A equação 3.7 representa a versão vectorial da restrição do brilho e, no caso de apenas se utilizar esta restrição para a estimação do fluxo óptico, apenas a projecção do fluxo óptico na direcção do gradiente poderá ser calculada. Este facto está relacionado com o designado problema da abertura e ilustrado na Fig. 3.2. Da figura, conclui-se que apenas a componente da velocidade que é normal as arestas dos objectos poderá ser detectada. A esta componente chama-se de fluxo óptico normal e é dada pela equação:

$$\| \vec{v}_{normal} \| = \Leftrightarrow \frac{I_t}{\sqrt{I_u^2 + I_v^2}} = \Leftrightarrow \frac{I_t}{\| \vec{\nabla} I \|} \quad (3.8)$$

que é obtida através da eq. 3.7.

Pode-se estabelecer uma relação entre o fluxo óptico normal com o campo de velocidade normal. Para isso assumamos um ponto (u, v) com uma velocidade projectada dada por

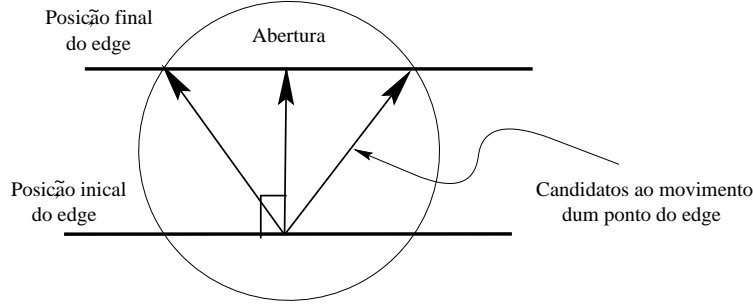


Figura 3.2: Representação do problema da abertura. O contorno da linha é observado através duma pequena janela no instante t e move-se para outra posição no instante $t+1$. O movimento efectuado pode ser uma translação vertical, ou uma translação mais um deslocamento na direcção da linha. No entanto, como a linha está a ser observada através duma pequena janela, apenas se pode detectar a componente do fluxo óptico na direcção perpendicular às arestas.

$\vec{v} = [\dot{u}, \dot{v}]^T$ e o respectivo gradiente $\vec{\nabla}I$. O gradiente local normalizado é dado por:

$$\vec{n} = \frac{\vec{\nabla}I}{\|\vec{\nabla}I\|} = \frac{1}{\|\vec{\nabla}I\|} \cdot \begin{bmatrix} I_u \\ I_v \end{bmatrix}. \quad (3.9)$$

A velocidade normal no ponto (u, v) é igual a:

$$u_n = \vec{n}^T \cdot \vec{v} = \frac{1}{\|\vec{\nabla}I\|} (I_u \cdot \dot{u} + I_v \cdot \dot{v}). \quad (3.10)$$

As duas velocidades são iguais se a subtracção da eq. 3.10 e eq. 3.8 for zero:

$$u_n \Leftrightarrow \|\vec{v}_{normal}\| = \frac{1}{\|\vec{\nabla}I\|} (I_u \cdot \dot{u} + I_v \cdot \dot{v} + I_t) = \frac{1}{\|\vec{\nabla}I\|} \frac{dI}{dt}. \quad (3.11)$$

Da eq. 3.11 pode-se concluir que as velocidades são iguais se o gradiente localmente for diferente de zero e de valor elevado, ou se a soma das derivadas locais da função I for pequena ou nula. Esta é uma restrição prática a efectuar a quando do cálculo do fluxo óptico, e é aplicado também no cálculo do fluxo óptico normal conforme explicado na secção 3.2.3.

3.2.3 Algoritmo de fluxo óptico normal

É do conhecimento geral, que o fluxo óptico normal em geral não é igual ao fluxo óptico

real porque o fluxo óptico normal depende do problema da abertura. Por este facto, é compreensível que existam erros quer na amplitude quer na orientação dos vectores de fluxo normal calculados. A questão que se põe é a seguinte, poderemos usar algum processo para corrigir ou atenuar estes erros? O próximo algoritmo é a nossa resposta a esta pergunta. Através de experiências realizadas descobriu-se que para obter melhores resultados a nível da orientação dos vectores de fluxo, era necessário utilizar três estados de processamento no cálculo do fluxo óptico normal. No entanto, este processo não introduz acréscimos significativos a nível do módulo dos vectores. Os três estados de processamento são executados sequencialmente e consistem em:

- *Estado de pré-processamento* - neste estado alguns dos pontos numericamente instáveis são excluídos do cálculo do fluxo óptico normal
- *Estado do cálculo do fluxo óptico normal* - cálculo do fluxo óptico tal como definido
- *Estado de pós-processamento* - um mecanismo de votação baseado em três níveis numa pirâmide de fluxo óptico normal, para testar a coerência dos vectores de fluxo, e o uso de um filtro aplicado a cada ponto para melhorar a estimação do fluxo.

Estes três estados são fundamentais, porque o cálculo do fluxo óptico normal é efectuado sobre imagens log-polar, onde a diferenciação numérica tende a ser mais instável. Se isso é um facto então qual é a justificação para usar imagens log-polar e fluxo óptico normal, e não usar um outro processo ou técnica de determinação de fluxo? Bom, não nos devemos esquecer que o nosso objectivo é obter resultados em tempo real, ou caso contrário poder-se-ia utilizar uma outra técnica mais elaborada, e eventualmente mais robusta, mas com o conseqüente aumento do tempo de processamento.

Estado de pré-processamento

Para cada ponto da imagem, o gradiente é calculado através dum operador espacial de dimensão 3×3 (definido na secção 3.2.8) para obter a estimação de $I_\zeta(\zeta, \gamma, t)$, $I_\gamma(\zeta, \gamma, t)$ e $I_t(\zeta, \gamma, t)$.

Na prática, apenas os “pixels” com gradiente e derivada temporal superior a um limiar serão considerados, de modo a evitar os erros causados pelo ruído associado ao cálculo diferencial. Este processo corresponde ao estado de pré-processamento e é apenas utilizado no algoritmo do fluxo óptico normal.

Estado do cálculo do fluxo óptico normal

O fluxo óptico normal é determinado através da exploração da restrição do brilho e é definido como:

$$\vec{v}_{normal}(\zeta, \gamma, t) = \frac{I_t}{\|\vec{\nabla}I\|^2} [I_\zeta, I_\gamma]^T. \quad (3.12)$$

Este vector é calculado após o estado de pré-processamento.

Estado de pós-processamento

O estado de pós-processamento, serve para garantir a coerência do movimento detectado através do uso duma pirâmide de imagens com três níveis. Subir um nível na pirâmide resulta numa imagem com metade das dimensões da imagem do nível anterior, onde cada pixel é obtido através da média dos “pixels” duma região de 2×2 da imagem no nível anterior. A imagem resultante é filtrada com uma função gaussiana, de modo a suavizar a imagem. Este filtro é um filtro passa-baixo para evitar fenómenos de “aliasing” (ver apêndice D na expressão D.1). Para cada imagem da pirâmide, o fluxo óptico normal é calculado usando os estados de *pré-processamento* e o estado do *cálculo do fluxo óptico normal*. A consistência dos vectores de movimento é efectuada através da análise do fluxo óptico normal nos três níveis da pirâmide.

No entanto, em primeiro lugar, a consistência do fluxo óptico normal é testada em cada nível separadamente, através do uso de regiões de $n \times n$ “pixels” (a região depende da estrutura de amostragem do log-polar e do nível da pirâmide). O fluxo $(\dot{\zeta}, \dot{\gamma})$ é o mais votado da região considerada, calculado nos vários níveis separadamente. As componentes radiais e angulares do fluxo $(\dot{\zeta}, \dot{\gamma})$ são dadas pelas células $\dot{\zeta}(\zeta_l, \gamma_l)$ e $\dot{\gamma}(\zeta_l, \gamma_l)$, onde ζ_l e γ_l são calculados como:

$$\begin{aligned}\gamma_l &= \text{hist}_{radial}^{-1}(\max\{\text{hist}_{radial}(\gamma)\}) \\ \zeta_l &= \text{hist}_{angular}^{-1}(\max\{\text{hist}_{angular}(\zeta)\})\end{aligned}\quad (3.13)$$

onde $\text{hist}_{radial}(\gamma) = \sum_{\zeta=0}^3 \dot{\zeta}(\zeta, \gamma)$ e $\text{hist}_{angular}(\zeta) = \sum_{\gamma=0}^3 \dot{\gamma}(\zeta, \gamma)$. Este processo é ilustrado na Fig. 3.3. Em cada nível da pirâmide um processo semelhante ao descrito pela eq. 3.13 é efectuado, mas com a diferença nas dimensões da região (a região terá metade das dimensões da região do nível da pirâmide anterior). Para o primeiro nível da pirâmide (imagem com maior resolução), a região é de 4×4 “pixels”. Um processo de votação similar a este foi proposto por J. Little e outros, [J. Little, H. Bulthof e T. Poggio 88] para o cálculo do fluxo óptico em imagens cartesianas.

A verificação da consistência do fluxo óptico normal nos três níveis da pirâmide de modo a estimar o fluxo na imagem de maior resolução, é efectuada da seguinte forma: todos os “pixels” são verificados nos três níveis, em regiões correspondentes, de modo a verificar a consistência angular e de amplitude. O valor resultante para o fluxo da região é calculado através dum processo de votação, cujo critério consiste em verificar se dois dos três vectores apresentam a mesma fase, caso contrário será assumido um valor nulo para o fluxo na respectiva região. É importante referir que a amplitude entre dois níveis duma pirâmide tem um factor de escala de 2, e que os vectores de fluxo normal são retirados de regiões correspondentes em cada nível. Estes factos são também levados em conta a quando da comparação das fases dos três vectores, uma vez que pode-se detectar fluxo no nível mais alto sem este estar presente nos níveis anteriores.

Após a aplicação dos três estados de processamento do fluxo óptico normal, as componentes dos vectores resultantes, são continuamente incluídos num filtro α - β . O uso deste filtro permite melhorar a estimação dos vectores de fluxo óptico, removendo à partida algum do ruído presente devido ao cálculo numérico e para garantir uma certa coerência e consistência a nível temporal. O uso do filtro é útil também para estimar os valores de fluxo em regiões sem textura, caso nesse local, em tempos passados, tenha possuído fluxo não nulo. Neste caso o fluxo para esses “pixels” é dado pela predição do filtro α - β (apenas para 3 amostras consecutivas com fluxo nulo). Para além deste carácter preditivo o uso do filtro previne uma variação abrupta nos vectores de fluxo detectados, enquanto tende a suavizar a velocidade em termos temporais.

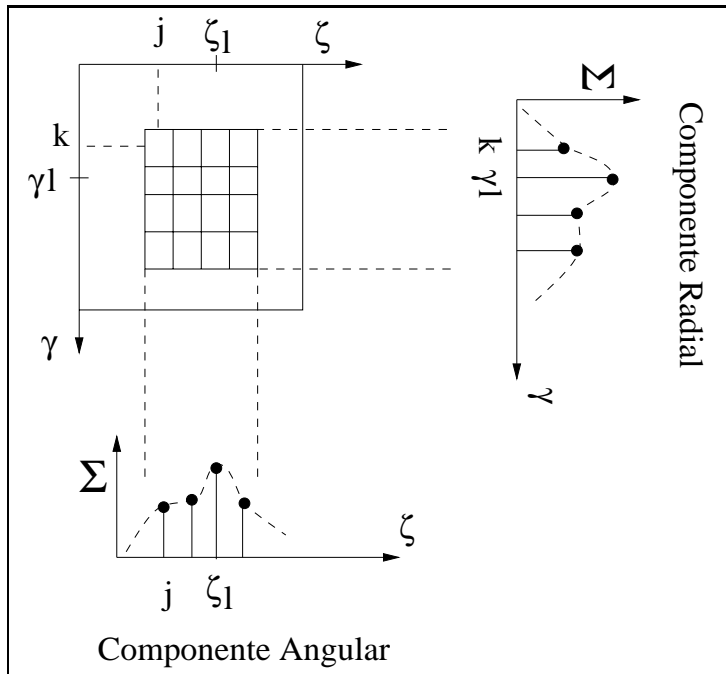


Figura 3.3: Verificação da consistência dos vectores de fluxo, através do uso de um esquema de votação, tal como é usado para o 1º nível da pirâmide

3.2.4 Algoritmo de fluxo óptico usando o modelo afim

O segundo método explorado, baseia-se também na restrição do brilho, em conjunção com o modelo afim (ou outro) entre duas imagens consecutivas. Esta técnica foi explorada por Nordlund & Uhlin [P. Nordlund e T. Uhlin 95], mas para imagens cartesianas. No nosso caso será estendida a sua aplicação a imagens log-polar.

Tal como explicado na secção 3.2.2, a equação 3.7, assume que a restrição de brilho é verificada, o que implica condições de iluminação uniforme.

Uma forma eficiente de ultrapassar este problema, pode ser a utilização de um modelo de velocidade descrito em função das coordenadas da imagem. Devido ao tempo Δt entre imagens não ser infinitesimal, é natural que a equação 3.7 não se verifique em alguns pontos. No entanto pode-se utilizar um método de mínimos quadrados de modo a minimizar o erro global cometido numa certa região da imagem (note-se que se está a

usar imagens log-polar):

$$\min \left\{ \sum_{\zeta=\zeta_0}^{\zeta_1} \sum_{\gamma=\gamma_0}^{\gamma_1} (I_\zeta \dot{\zeta} + I_\gamma \dot{\gamma} + I_t)^2 \right\} \quad (3.14)$$

Utilizando a expressão (3.2) a velocidade pode ser descrita de três formas distintas, e para cada uma delas, pode-se obter uma solução fechada:

- modelo puramente translacional: $\begin{bmatrix} \dot{\zeta} \\ \dot{\gamma} \end{bmatrix} = \begin{bmatrix} a_1 \\ b_1 \end{bmatrix}$,
- modelo afim: $\begin{bmatrix} \dot{\zeta} \\ \dot{\gamma} \end{bmatrix} = \begin{bmatrix} a_1 + a_2 \cdot \zeta + a_3 \cdot \gamma \\ b_1 + b_2 \cdot \zeta + b_3 \cdot \gamma \end{bmatrix}$,
- modelo duma superfície planar em movimento: $\begin{bmatrix} \dot{\zeta} \\ \dot{\gamma} \end{bmatrix} = \begin{bmatrix} a_1 + a_2 \cdot \zeta + a_3 \cdot \gamma + a \cdot \zeta^2 + b \cdot \zeta \cdot \gamma \\ b_1 + b_2 \cdot \zeta + b_3 \cdot \gamma + a \cdot \zeta \cdot \gamma + b \cdot \gamma^2 \end{bmatrix}$,

Nestes modelos temos 2, 6 e 8 parâmetros a estimar, respectivamente. Para o modelo translacional a solução é dada pela equação:

$$\begin{bmatrix} \sum I_\zeta^2 & \sum I_\gamma I_\zeta \\ \sum I_\gamma I_\zeta & \sum I_\gamma^2 \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ b_1 \end{bmatrix} = \begin{bmatrix} \Leftrightarrow \sum I_t I_\zeta \\ \Leftrightarrow \sum I_t I_\gamma \end{bmatrix} \quad (3.15)$$

Para o modelo afim, podemos inicialmente definir a seguinte matriz simétrica:

$$\mathbf{A} = \begin{bmatrix} \sum I_\zeta^2 & \sum I_\zeta^2 \zeta & \sum I_\zeta^2 \gamma & \sum I_\gamma I_\zeta & \sum I_\gamma I_\zeta \zeta & \sum I_\gamma I_\zeta \gamma \\ & \sum I_\zeta^2 \zeta^2 & \sum I_\zeta^2 \zeta \gamma & \sum I_\gamma I_\zeta \zeta & \sum I_\gamma I_\zeta \zeta^2 & \sum I_\gamma I_\zeta \zeta \gamma \\ & & \sum I_\zeta^2 \gamma^2 & \sum I_\gamma I_\zeta \gamma & \sum I_\gamma I_\zeta \zeta \gamma & \sum I_\gamma I_\zeta \gamma^2 \\ & & & \sum I_\gamma^2 & \sum I_\gamma^2 \zeta & \sum I_\gamma^2 \gamma \\ & & & & \sum I_\gamma^2 \zeta^2 & \sum I_\gamma^2 \zeta \gamma \\ & & & & & \sum I_\gamma^2 \gamma^2 \end{bmatrix} \quad (3.16)$$

simetrica

e os vectores:

$$\vec{\mathbf{x}} = \left[a_1, a_2, a_3, b_1, b_2, b_3 \right]^T$$

$$\vec{\mathbf{k}} = \left[\Leftrightarrow \sum I_t I_\zeta, \Leftrightarrow \sum I_t I_\zeta \zeta, \Leftrightarrow \sum I_t I_\zeta \gamma, \Leftrightarrow \sum I_t I_\gamma, \Leftrightarrow \sum I_t I_\gamma \zeta, \Leftrightarrow \sum I_t I_\gamma \gamma \right]^T$$

neste caso o número de parâmetros a estimar é de 6 e podem ser obtidos através da seguinte relação:

$$\mathbf{A} \cdot \vec{\mathbf{x}} = \vec{\mathbf{k}} \quad (3.17)$$

Para o modelo numa superfície planar em movimento, definindo as entidades:

$$q = I_\zeta \zeta + I_\gamma \gamma$$

$$\vec{\mathbf{x}}' = \left[a \quad b \right]^T$$

$$\vec{\mathbf{k}}' = \left[\Leftrightarrow I_t \zeta \cdot q \quad \Leftrightarrow I_t \gamma \cdot q \right]^T$$

$$\mathbf{C} = \begin{bmatrix} \sum q^2 \cdot \zeta^2 & \sum q^2 \cdot \zeta \gamma \\ \sum q^2 \cdot \zeta \gamma & \sum q^2 \cdot \gamma^2 \end{bmatrix}$$

$$\mathbf{D} = \begin{bmatrix} \sum q \cdot I_\zeta \zeta & \sum q \cdot I_\zeta \zeta^2 & \sum q \cdot I_\zeta \zeta \gamma & \sum q \cdot I_\gamma \zeta & \sum q \cdot I_\gamma \zeta^2 & \sum q \cdot I_\gamma \zeta \gamma \\ \sum q \cdot I_\zeta \gamma & \sum q \cdot I_\zeta \zeta \gamma & \sum q \cdot I_\zeta \gamma^2 & \sum q \cdot I_\gamma \gamma & \sum q \cdot I_\gamma \zeta \gamma & \sum q \cdot I_\gamma \gamma^2 \end{bmatrix}$$

a solução pode ser calculada através de:

$$\begin{bmatrix} \mathbf{A} & \mathbf{D}^T \\ \mathbf{D} & \mathbf{C} \end{bmatrix} \cdot \begin{bmatrix} \vec{\mathbf{x}} \\ \vec{\mathbf{x}}' \end{bmatrix} = \begin{bmatrix} \vec{\mathbf{k}} \\ \vec{\mathbf{k}}' \end{bmatrix} \quad (3.18)$$

Para usar este método, de início, descreve-se a velocidade através do modelo afim, e se o determinante da matriz A for nulo ou abaixo dum limiar, então será assumido o modelo translacional. Se ainda para este modelo o determinante da matriz A estiver abaixo dum limiar ou for nulo então será assumido um fluxo nulo para a região considerada.

Nas equações indicadas anteriormente, a soma é efectuada numa região de $n \times m$ “pixels” de modo a estimar os 6 ou 2 parâmetros (no nossos testes práticos esta região

possuí as dimensões de 23×36 “pixels” para dividir a imagem log-polar no mesmo número de regiões para as duas dimensões da imagem).

Estes modelos usam a imagem log-polar, tal como se esta fosse uma imagem cartesiana. Contudo na próxima secção poderá ser visto, que se pode definir os mesmos modelos, mas usando o facto de que as imagens utilizadas são imagens log-polar.

Refinamento do Algoritmo

Como neste caso são utilizadas imagens log-polar, será de esperar que se possa usar os resultados diferenciais, tal como se estes tivessem sido tirados da imagem cartesiana no mesmo ponto onde a amostragem log-polar é feita, mas com a diferença, de que neste caso o log-polar é utilizado apenas para aumentar a velocidade de cálculo. Este processo é fácil de implementar, dadas as derivadas espaciais (I_ζ, I_γ) e usando a eq. 2.36 pode-se obter (I_u, I_v), e com a eq. 2.18, gerar as coordenadas (u, v) dadas as coordenadas (ζ, γ). Usando o método desta forma, os parâmetros calculados podem ser relacionados com os parâmetros de velocidade da câmara, tal como foi explicado anteriormente. Contudo com este modelo, a profundidade da cena não pode possuir grandes discontinuidades porque se assume que os pontos tridimensionais estão restringidos a um plano.

Verificação do modelo

Após a solução do sistema, pode-se calcular o fluxo óptico para cada ponto da imagem, e desta forma verificar quais os pontos da imagem que verificam o modelo. Este facto pode ser verificado através da construção de uma imagem erro usando a equação:

$$Err = (I_\zeta \cdot \dot{\zeta} + I_\gamma \cdot \dot{\gamma} + I_t)^2 \quad (3.19)$$

O fluxo óptico pode então ser novamente determinado para as regiões da imagem que não verificaram o modelo, dando assim a possibilidade de detectar diferentes objectos (caso estes se encontrem a diferentes profundidades).

3.2.5 Algoritmo de fluxo óptico de Horn e Schunck Revisto

O terceiro método implementado, usa um processo semelhante ao algoritmo proposto por Horn e Schunck [J. Barron e D. Fleet 94], mas aplicado neste caso ao plano log-polar. Este processo combina a restrição do brilho com uma restrição que garante uma certa suavidade ao campo de velocidade estimado. O método define as grandezas $s(\zeta, \gamma)$ e $c(\zeta, \gamma)$, respectivamente de medida de contraste (transições abruptas) e erro na restrição do fluxo óptico como:

$$s(\zeta, \gamma) = \frac{1}{4}([\dot{\zeta}(\zeta + 1, \gamma) \Leftrightarrow \dot{\zeta}(\zeta, \gamma)]^2 + [\dot{\zeta}(\zeta, \gamma + 1) \Leftrightarrow \dot{\zeta}(\zeta, \gamma)]^2 + [\dot{\gamma}(\zeta + 1, \gamma) \Leftrightarrow \dot{\gamma}(\zeta, \gamma)]^2 + [\dot{\gamma}(\zeta, \gamma + 1) \Leftrightarrow \dot{\gamma}(\zeta, \gamma)]^2)$$

$$c(\zeta, \gamma) = [I_\zeta \dot{\zeta} + I_\gamma \dot{\gamma} + It]^2$$

O objectivo principal é o de encontrar os valores que minimizam a seguinte equação numa determinada região

$$\sum_{\zeta, \gamma} [s(\zeta, \gamma) + \lambda c(\zeta, \gamma)] \quad (3.20)$$

O $\lambda \in \mathfrak{R}^+$, e é usado como uma forma de balancear a importância entre as funções $s(\zeta, \gamma)$ e $c(\zeta, \gamma)$. Se o λ se aproxima de zero, a função $s(\zeta, \gamma)$ terá mais peso no resultado, caso contrário será a função $c(\zeta, \gamma)$.

A principal vantagem deste método advém do facto da minimização poder ser resolvida através duma solução iterativa. A solução iterativa para obter a velocidade é assim dada por:

$$\begin{aligned} \dot{\zeta}^{k+1} &= \bar{\zeta}^k \Leftrightarrow \frac{I_\zeta \cdot [I_\zeta \bar{\zeta}^k + I_\gamma \bar{\gamma}^k + It]}{1 + \lambda \cdot (I_\zeta^2 + I_\gamma^2)} \\ \dot{\gamma}^{k+1} &= \bar{\gamma}^k \Leftrightarrow \frac{I_\gamma \cdot [I_\zeta \bar{\zeta}^k + I_\gamma \bar{\gamma}^k + It]}{1 + \lambda \cdot (I_\zeta^2 + I_\gamma^2)} \end{aligned} \quad (3.21)$$

onde k indica o número de iterações, $\dot{\zeta}^0$ e $\dot{\gamma}^0$ indicam estimativas iniciais da velocidade iniciadas a zero, $\bar{\zeta}^k$ e $\bar{\gamma}^k$ indicam médias locais na vizinhança dos pontos considerados. Nas experiências realizadas, estas médias foram tiradas dos 8 “pixels” à volta do pixel (ζ, γ) . O valor de λ e o número de iterações foram obtidos através das experiências efectuadas e são um pouco heurísticos. No nosso caso usaram-se 5 iterações com $\lambda = 3.5$.

Na prática este método apresenta alguns problemas, uma vez que o fluxo tem tendência a ser propagado para outras regiões devido às médias da velocidade, e nalguns casos, as zonas da imagem sem movimento podem assim apresentar velocidade não nula. Este processo pode ser controlado através duma escolha adequada para o número de iterações. Outro problema importante, neste método, prende-se com as atribuições iniciais na periferia da imagem, pois este fluxo também será propagado para outras regiões da imagem.

3.2.6 Cálculo de fluxo óptico usando correlação

Uma técnica bastante importante de cálculo de fluxo óptico, pode ser obtida através do uso de correlação, no entanto este método apenas foi testado em imagens cartesianas. O método que se apresenta em baixo, assume que a cena visualizada não possui grandes descontinuidades em termos de profundidade.

A ideia é em cada imagem da sequência, identificar pontos de interesse (estes pontos vêm de zonas da imagem com alto contraste). Caso não seja possível identificar estes pontos, o método falha. O operador de selecção dos pontos de interesse tem por objectivo seleccionar zonas de alto contraste como por exemplo, zonas de contornos, objectos, etc, de modo à área em redor do ponto ser facilmente identificável em imagens futuras. Estes objectivos podem ser concretizados através do uso duma função que retorne medidas da variância direccional, retornando os pontos com máxima variância. A variância direccional é medida em pequenas janelas quadrangulares, e neste caso é implementada através da soma do quadrado da diferença entre “pixels” adjacentes em cada uma das três direcções (horizontal, vertical e diagonal). O operador de interesse retorna assim o valor mínimo de entre os três valores calculados.

Em cada ponto de interesse, é definido um padrão de tamanho $n \times n$ “pixels” na imagem $I(t)$, e a correlação é efectuada numa janela de pesquisa com tamanho $x \times y$ e centrada no mesmo ponto da imagem $I(t + \Delta t)$. O deslocamento entre o ponto original e o ponto onde a correlação é maior define o fluxo óptico desse ponto.

Para extrapolar o fluxo óptico para toda a imagem, considera-se um modelo de fluxo afim, e neste caso apenas são necessários 3 pontos para resolver o sistema de incógnitas. No entanto usando apenas 3 pontos pode resultar em soluções instáveis, pois a correlação pode

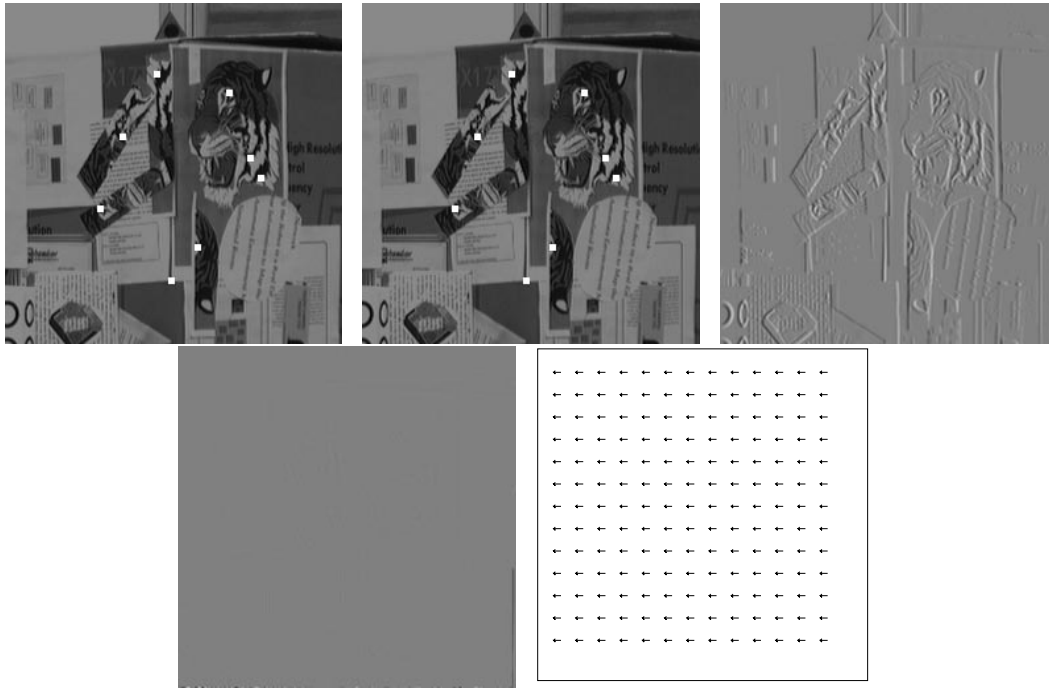


Figura 3.4: No topo esquerdo: a imagem com os 8 pontos de interesse usados para estimar o fluxo, a imagem central mostra os pontos identificados pelo processo de correlação, a imagem à direita mostra a diferença entre as duas imagens anteriores. Em baixo, à esquerda encontra-se a diferença entre a segunda imagem, e a imagem original deslocada pelo campo de fluxo estimado pelo processo de *Cálculo de fluxo óptico usando correlação*.

falhar em algum ponto. Para resolver este problema são usados m pontos de m regiões diferentes da imagem. O vector de medidas efectuado para cada ponto é $[\dot{u}_i, \dot{v}_i, u_i, v_i]^T$ e usando um método de mínimos quadrados, pode-se obter a solução do sistema. Para cada ponto considerado obtém-se a seguinte relação matricial:

$$\begin{bmatrix} \dot{u}_i \\ \dot{v}_i \end{bmatrix} = \begin{bmatrix} 1 & u_i & v_i & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & u_i & v_i \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad (3.22)$$



Figura 3.5: Evolução do algoritmo de fluxo óptico normal. Da esquerda para a direita, encontra-se representada a evolução da cena, e o respectivo fluxo no plano log-polar.

Em geral, dado um sistema linear de equações

$$\mathbf{A} \cdot \vec{\mathbf{x}} = \vec{\mathbf{k}} \quad (3.23)$$

onde \mathbf{A} é uma matriz de $m \times n$ elementos ($m > n$), $\vec{\mathbf{k}}$ é um vector de m elementos e $\vec{\mathbf{x}}$ é um vector de n incógnitas, a pseudo-inversa, ou solução de mínimos quadrados é dada por:

$$\vec{\mathbf{x}} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \vec{\mathbf{k}} \quad (3.24)$$

Nos testes foi usado $n = 6$ e $m = 16$. Após a solução pode-se gerar o fluxo óptico para cada pixel da imagem. A Fig. 3.4 mostra um exemplo desta técnica.

3.2.7 Resultados experimentais com imagens sintéticas

Nesta secção, serão apresentados alguns resultados experimentais, para o fluxo óptico, e fluxo óptico normal descritos anteriormente.

Os primeiros exemplos nas Fig. 3.5, 3.6 e 3.7, são os resultados dos algoritmos de fluxo óptico normal, fluxo óptico usando os modelos afins, e Horn & Schunck (usam todos uma pirâmide com três níveis de imagens). As imagens são sintéticas e simulam a translação de uma câmara na direcção do eixo óptico. Das figuras, pode-se verificar que o movimento foi detectado por qualquer um dos métodos, sendo o resultado mais coerente, o dado pelo algoritmo afim. Contudo os outros dois, também apresentam um bom comportamento, no entanto dos três métodos, o fluxo óptico normal é o *mais rápido*. A performance obtida pelo algoritmo de *fluxo óptico normal* deve-se à utilização dos estados de pré- e pós-processamento.

Uma das formas de testar a robustez dos algoritmos na detecção de movimento, é a utilização de situações onde a velocidade na imagem aumente ou seja, que esteja sujeita



Figura 3.6: Evolução do algoritmo de fluxo óptico baseado no modelo afim



Figura 3.7: Evolução do algoritmo de Horn & Schunck

a uma aceleração. Neste caso será possível detectar se os algoritmos conseguiram seguir perfeitamente o campo de velocidades. Para isso, a mesma situação foi colocada a cada algoritmo separadamente, onde era simulada uma velocidade crescente. A velocidade para dois “pixels” consecutivos foi medida, enquanto a sequência decorria. Neste caso a cena era oblíqua relativamente à câmara, e a Fig 3.8 mostra a evolução temporal da velocidade ζ para os três métodos. Conforme pode ser visto, na 9ª amostra, onde o pixel apresenta velocidade não nula, o método afim detectou velocidade zero, o Horn & Schunck detectou velocidade diferente de zero, devido ao fenómeno de propagação, e finalmente, o método do fluxo óptico normal, com a ajuda do filtro α - β conseguiu detectar a velocidade crescente.

As experiências que se seguem demonstram os resultados para o método de refinamento do modelo afim conforme explicado na secção 3.2.4, neste caso no plano log-polar, de modo a ser utilizado na detecção da posição do *FOE*.

Na imagem representada na Fig. 3.9 o observador movimenta-se para a frente em direcção ao canto da parede. Os parâmetros do modelo afim para a imagem cartesiana (256×256) são:

$$(a_1, a_2, a_3, b_1, b_2, b_3) = (\Leftrightarrow 0.080501, 0.001473, \Leftrightarrow 0.000403, 0.038492, 0.000294, 0.005596)$$

enquanto para o modelo afim da imagem log-polar (usando 180 amostras angulares e

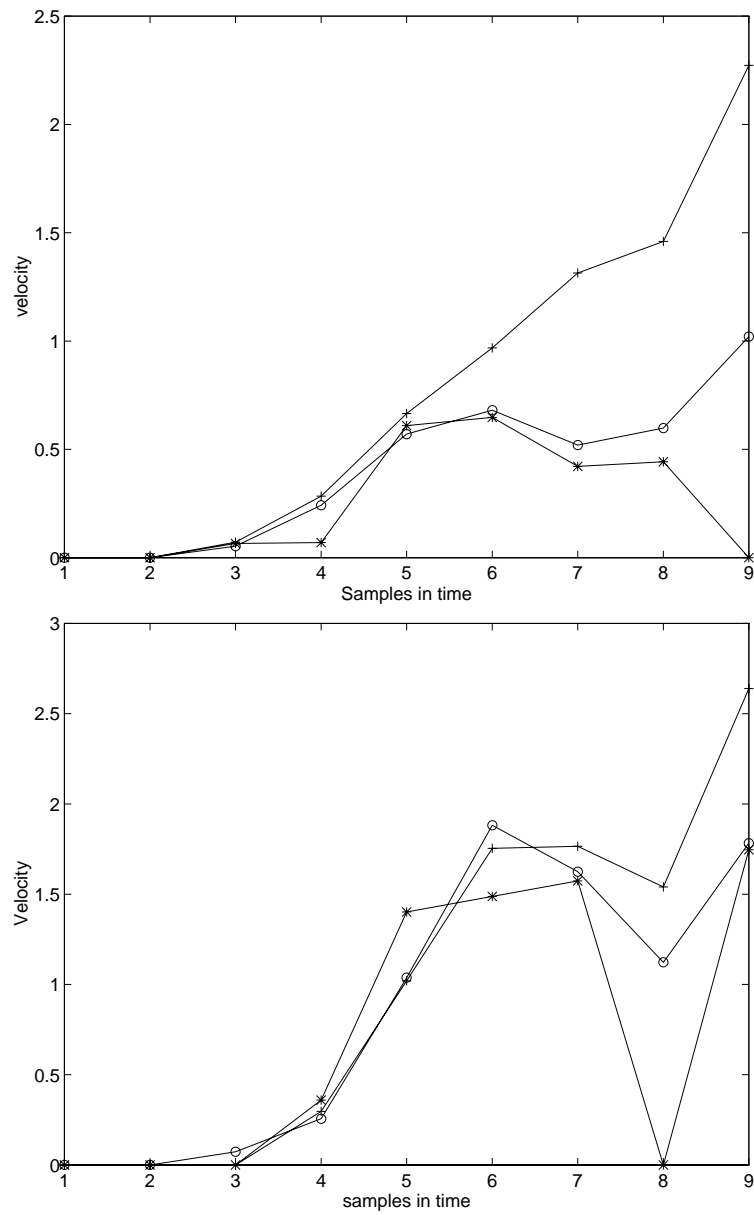


Figura 3.8: Evolução temporal da velocidade ζ para dois “pixels” consecutivos(+ -fluxo óptico Normal * -Modelo afim o -Horn & Schunck)

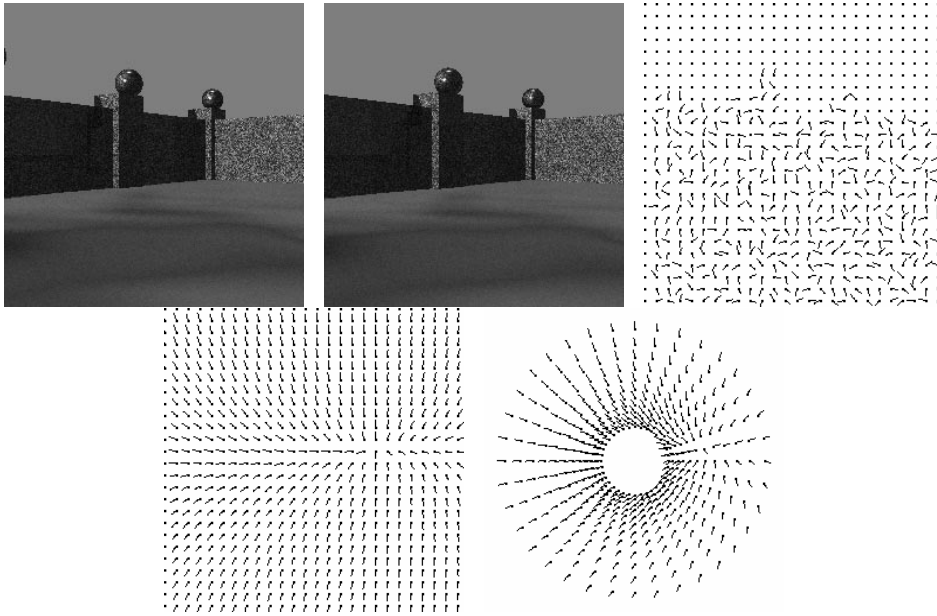


Figura 3.9: No topo esquerdo, está representada a sequência. No topo direito o respectivo fluxo óptico normal. Em baixo à esquerda o modelo afim para a imagem cartesiana, e em baixo à direita o modelo afim para a respectiva imagem log-polar.

$a = b = 1.025$), foi detectado:

$$(a_1, a_2, a_3, b_1, b_2, b_3) = (\Leftrightarrow 0.057809, 0.000917, \Leftrightarrow 0.000409, 0.015966, 0.000031, 0.001717)$$

a maior diferença pode ser vista em termos dum valor de escalonamento entre os dois tipos de parâmetros, e este facto acontece devido ao número de pontos envolvidos no cálculo ser diferente, no entanto ambos os parâmetros mantêm a tendência do movimento detectado (com a vantagem do log-polar necessitar de menor informação).

No segundo exemplo, foram utilizadas imagens reais, e a câmara roda para a esquerda, o modelo afim para a imagem cartesiana detectou:

$$(a_1, a_2, a_3, b_1, b_2, b_3) = (\Leftrightarrow 1.053742, \Leftrightarrow 0.000685, 0.000571, \Leftrightarrow 0.007082, 0.000003, \Leftrightarrow 0.000006)$$

enquanto para a imagem log-polar:

$$(a_1, a_2, a_3, b_1, b_2, b_3) = (\Leftrightarrow 0.173429, \Leftrightarrow 0.000018, 0.000218, \Leftrightarrow 0.011265, \Leftrightarrow 0.000040, \Leftrightarrow 0.000049)$$

e novamente os parâmetros não são iguais, mas mantêm a tendência do movimento efectuado.

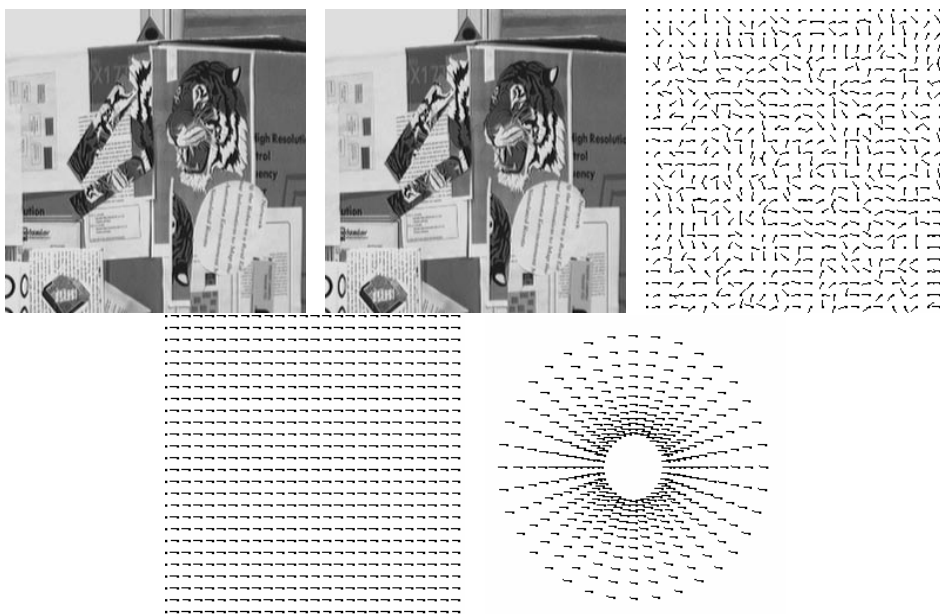


Figura 3.10: Em cima à esquerda encontra-se a sequência. Em cima à direita o respectivo fluxo óptico normal. Em baixo à esquerda, o resultado do modelo afim nas imagens cartesianas, e à direita o resultado do modelo afim aplicado às respectivas imagens log-polar.

3.2.8 Medida do desempenho de algoritmos de cálculo de fluxo óptico em imagens log-polar e cartesianas

Barron e outros [J. Barron e D. Fleet 94] implementaram uma série de técnicas de cálculo de fluxo óptico de modo a comparar as suas performances. Nesta secção vamos usar as mesmas definições para analisar os fluxos obtidos, mas usando imagens log-polar. Os resultados apresentados usam as sequências de imagens “Translating and Diverging Tree sequences”, no plano cartesiano e no plano log-polar (para cada caso será apresentado o **Erro médio** (Ave. Err.), **Desvio padrão** (Sta. Dev.) e **Densidade** (Dens.)).

O algoritmo de fluxo óptico normal foi implementado, usando um pirâmide com dois níveis (Nflow 2L) e usando uma pirâmide com um nível (Nflow 1L).

As imagens foram filtradas com três tipos diferentes de filtros, um filtro gaussiano com $\sigma = 1.5$, um filtro FIR com máscara

$$[0.051, 0.0, \Leftrightarrow 0.087, 0.0, 0.298, 0.475, 0.298, 0.0, \Leftrightarrow 0.087, 0.0, 0.051]$$

aplicado na direcção u e de seguida na direcção v , e um filtro de Wechsler [H. Wechsler 90] definido como:

$$\begin{bmatrix} c & c & c & c & c \\ c & b & b & b & c \\ c & b & a & b & c \\ c & b & b & b & c \\ c & c & c & c & c \end{bmatrix} \quad (3.25)$$

onde $0 < a < 0.5$, $b = 1/16$ e $c = 1/32 \Leftrightarrow a/16$.

O processo de estimação das derivadas temporais e espaciais é extremamente importante tal como pode ser visto através de estudos realizados por F. Bergholm [F. Bergholm 93]. Devido a esse facto, para estimar as derivadas temporais e espaciais foram usados cinco métodos diferentes:

1. Para as derivadas espaciais: a matriz $[[\Leftrightarrow 1, 0, 1], [\Leftrightarrow 2, 0, 2], [\Leftrightarrow 1, 0, 1]]$ para a direcção u e na direcção v pela sua transposta, em cada caso, o resultado é dividido por 8.0.

		Gauss com $\sigma = 1.5$			filtro FIR			filtro de Wechsler		
Dif.	Método	Ave. Err.	Sta. Dev.	Den.	Ave. Err.	Sta. Dev.	Den.	Ave. Err.	Sta. Dev.	Den.
1	Nflow 2L	44.32	19.26	27%	45.15	18.96	27%	44.56	19.18	27%
	Nflow 1L	37.05	16.22	19%	38.33	16.44	19%	37.42	16.25	19%
	Horn	37.63	14.97	28%	37.92	14.70	28%	37.90	14.87	28%
	Affine	26.96	9.83	28%	27.37	10.13	28%	27.09	10.06	28%
	Second Affine	36.19	11.82	28%	36.15	11.61	28%	36.17	11.72	28%
2	Nflow 2L	45.19	17.12	27%	45.59	17.15	27%	45.27	17.17	27%
	Nflow 1L	43.76	16.41	24%	44.37	16.60	24%	44.05	16.41	24%
	Horn	44.81	16.56	28%	45.22	16.56	28%	44.88	16.68	28%
	Affine	41.31	15.24	28%	41.91	15.51	28%	41.54	15.38	28%
	Second Affine	45.47	16.74	28%	45.72	16.83	28%	45.59	16.78	28%
3	Nflow 2L	47.49	19.50	28%	47.17	19.66	27%	47.24	19.32	27%
	Nflow 1L	39.70	18.45	16%	41.31	18.78	16%	40.35	18.54	16%
	Horn	36.16	15.25	28%	37.19	15.39	28%	36.59	15.13	28%
	Affine	26.79	14.37	28%	26.90	14.74	28%	26.83	14.51	28%
	Second Affine	30.34	8.97	28%	30.54	8.15	28%	30.40	8.56	28%
4	Nflow 2L	44.29	19.42	28%	45.71	18.79	27%	45.28	19.06	27%
	Nflow 1L	38.66	16.12	19%	40.13	16.95	19%	39.07	16.35	19%
	Horn	38.30	15.12	28%	39.28	15.72	28%	38.82	15.21	28%
	Affine	28.63	9.55	28%	29.80	9.87	28%	29.13	9.69	28%
	Second Affine	37.41	12.48	28%	37.93	12.71	28%	37.62	12.55	28%
5	Nflow 2L	44.22	19.36	28%	45.37	18.80	27%	44.88	19.03	27%
	Nflow 1L	37.85	16.08	19%	39.43	16.52	19%	38.28	16.19	19%
	Horn	37.90	15.03	28%	38.67	15.26	28%	38.26	14.97	28%
	Affine	27.87	9.71	28%	28.85	10.06	28%	28.23	9.90	28%
	Second Affine	36.29	11.78	28%	36.45	11.75	28%	36.33	11.72	28%

Tabela 3.1: Resultados para a sequência “Translating Tree” usando imagens Log-Polar

		Gauss com $\sigma = 1.5$			filtro FIR			filtro de Wechsler		
Dif.	Método	Ave. Err.	Sta. Dev.	Den.	Ave. Err.	Sta. Dev.	Den.	Ave. Err.	Sta. Dev.	Den.
Dif. 1	Nflow 2L	54.97	13.80	99%	56.33	12.93	99%	55.51	13.49	98%
	Nflow 1L	45.62	15.91	61%	47.26	15.55	61%	45.78	15.68	60%
	Horn	46.34	11.25	100%	45.49	10.53	100%	45.74	10.95	100%
	Affine	20.59	8.28	100%	19.20	7.89	100%	19.54	7.37	100%
	Second Affine	18.30	0.94	100%	17.16	0.66	100%	17.62	0.76	100%
Dif. 2	Nflow 2L	59.30	4.40	97%	59.95	4.22	98%	59.40	4.30	97%
	Nflow 1L	59.01	4.33	86%	59.74	4.59	89%	59.13	4.38	87%
	Horn	59.47	3.12	100%	59.93	3.06	100%	59.44	3.16	100%
	Affine	53.84	2.96	100%	54.82	2.75	100%	53.95	2.98	100%
	Second Affine	53.43	0.72	100%	54.09	0.52	100%	53.54	0.74	100%
Dif. 3	Nflow 2L	57.74	13.68	99%	58.50	12.42	99%	57.73	13.61	99%
	Nflow 1L	41.84	20.20	49%	44.17	19.57	50%	41.80	19.89	49%
	Horn	39.71	16.79	100%	37.72	16.29	100%	38.42	15.96	100%
	Affine	10.31	18.31	100%	9.48	15.12	100%	7.56	12.24	100%
	Second Affine	1.09	0.58	100%	1.87	0.07	100%	1.16	0.14	100%
Dif. 4	Nflow 2L	55.32	13.55	99%	56.13	12.87	98%	55.84	13.19	98%
	Nflow 1L	46.08	15.90	60%	48.51	15.39	64%	46.23	15.58	60%
	Horn	46.86	11.34	100%	47.21	10.60	100%	46.50	10.94	100%
	Affine	23.89	11.05	100%	22.99	9.14	100%	21.84	8.41	100%
	Second Affine	19.96	1.38	100%	19.00	1.20	100%	18.79	0.81	100%
Dif. 5	Nflow 2L	55.06	13.76	99%	56.35	12.87	98%	55.59	13.48	98%
	Nflow 1L	45.64	16.02	60%	47.63	15.76	61%	45.75	15.77	59%
	Horn	46.55	11.36	100%	46.29	10.81	100%	46.07	11.10	100%
	Affine	22.61	10.87	100%	21.34	9.50	100%	20.55	8.47	100%
	Second Affine	18.65	1.19	100%	17.48	0.96	100%	17.66	0.85	100%
	Correla.	1.55	0.93	100%	1.55	0.93	100%	6.98	5.07	100%

Tabela 3.2: Resultados para a sequência “Translating Tree” usando imagens cartesianas

		Gauss com $\sigma = 1.5$			filtro FIR			filtro de Wechsler		
Dif.	Método	Ave. Err.	Sta. Dev.	Den.	Ave. Err.	Sta. Dev.	Den.	Ave. Err.	Sta. Dev.	Den.
1	Nflow 2L	30.10	9.30	28%	30.37	9.06	27%	30.37	9.21	27%
	Nflow 1L	26.41	10.69	19%	27.33	10.67	19%	26.77	10.71	19%
	Horn	27.36	10.27	28%	27.51	10.34	28%	27.49	10.19	28%
	Affine	18.33	10.38	28%	19.12	10.53	28%	18.61	10.45	28%
	Second Affine	23.23	8.63	28%	23.47	8.64	28%	23.31	8.64	28%
2	Nflow 2L	32.88	8.84	26%	33.29	8.85	27%	32.81	8.80	27%
	Nflow 1L	32.35	8.32	24%	32.85	8.41	24%	32.56	8.41	24%
	Horn	32.60	8.95	28%	33.00	8.94	28%	32.69	8.99	28%
	Affine	31.06	9.15	28%	31.54	9.24	28%	31.19	9.30	28%
	Second Affine	32.06	8.71	28%	32.38	8.70	28%	32.19	8.71	28%
3	Nflow 2L	29.72	9.56	28%	30.24	9.76	28%	30.20	9.49	28%
	Nflow 1L	25.16	11.80	16%	26.47	11.81	16%	25.67	11.80	16%
	Horn	23.78	10.94	28%	23.56	10.93	28%	23.80	10.96	28%
	Affine	10.16	10.22	28%	10.68	10.37	28%	10.46	10.26	28%
	Second Affine	14.62	8.57	28%	14.99	8.60	28%	14.72	8.58	28%
4	Nflow 2L	30.43	9.29	28%	31.02	9.82	27%	30.52	9.30	27%
	Nflow 1L	27.66	10.37	18%	29.84	11.37	20%	28.45	10.59	19%
	Horn	27.96	10.42	28%	29.42	10.91	28%	28.31	10.44	28%
	Affine	20.69	9.99	28%	22.68	10.68	28%	21.46	10.25	28%
	Second Affine	24.83	8.64	28%	26.54	8.65	28%	25.46	8.65	28%
5	Nflow 2L	30.18	9.38	28%	30.86	9.88	27%	30.34	9.15	27%
	Nflow 1L	26.88	10.54	18%	28.70	11.35	19%	27.50	10.64	18%
	Horn	27.66	10.33	28%	28.29	10.76	28%	27.94	10.39	28%
	Affine	19.49	10.18	28%	21.11	10.83	28%	20.14	10.36	28%
	Second Affine	24.02	8.63	28%	25.26	8.65	28%	24.44	8.64	28%

Tabela 3.3: Resultados para a sequência “Diverging Tree” usando imagens Log-Polar

		Gauss com $\sigma = 1.5$			filtro FIR			filtro de Wechsler		
Dif.	Método	Ave. Err.	Sta. Dev.	Den.	Ave. Err.	Sta. Dev.	Den.	Ave. Err.	Sta. Dev.	Den.
Dif. 1	Nflow 2L	37.47	13.66	97%	37.55	13.44	97%	37.52	13.67	97%
	Nflow 1L	28.60	12.50	61%	29.70	12.43	63%	28.52	12.14	59%
	Horn	30.68	11.59	100%	30.60	11.04	100%	30.47	11.20	100%
	Affine	19.40	8.12	100%	19.48	7.59	100%	19.13	8.19	100%
	Second Affine	18.67	4.60	100%	18.68	4.63	100%	18.23	4.53	100%
Dif. 2	Nflow 2L	39.09	11.71	93%	39.52	12.18	97%	39.14	11.67	93%
	Nflow 1L	37.64	11.41	85%	38.66	11.81	90%	37.88	11.40	87%
	Horn	38.49	11.75	100%	39.23	11.85	100%	38.58	11.73	100%
	Affine	36.81	11.22	100%	37.89	11.39	100%	36.98	11.10	100%
	Second Affine	36.43	10.44	100%	37.09	10.69	100%	36.45	10.46	100%
Dif. 3	Nflow 2L	38.02	14.65	99%	37.93	14.68	98%	37.86	14.59	99%
	Nflow 1L	24.06	15.72	49%	24.85	15.43	49%	24.03	15.26	48%
	Horn	24.27	13.12	100%	22.51	12.07	100%	23.56	12.78	100%
	Affine	8.38	10.37	100%	6.09	9.74	100%	7.70	11.20	100%
	Second Affine	8.59	4.90	100%	5.81	3.32	100%	7.60	4.40	100%
Dif. 4	Nflow 2L	37.51	13.60	97%	37.40	13.20	97%	37.46	13.57	97%
	Nflow 1L	29.04	12.77	60%	31.05	12.51	67%	29.08	12.23	59%
	Horn	31.11	11.75	100%	31.87	11.15	100%	30.96	11.44	100%
	Affine	20.90	8.38	100%	21.29	6.95	100%	20.61	7.83	100%
	Second Affine	19.13	4.48	100%	19.74	4.76	100%	18.69	4.48	100%
Dif. 5	Nflow 2L	37.54	13.66	97%	37.47	13.41	97%	37.41	13.66	97%
	Nflow 1L	28.45	12.45	59%	29.79	12.45	64%	28.36	11.96	58%
	Horn	30.80	11.66	100%	30.72	11.04	100%	30.49	11.18	100%
	Affine	19.19	6.87	100%	19.03	5.79	100%	18.76	6.31	100%
	Second Affine	17.65	4.11	100%	18.00	4.29	100%	17.44	4.16	100%
	Correla.	17.37	9.16	100%	22.15	8.54	100%	44.64	23.39	100%

Tabela 3.4: Resultados para a sequência “Diverging Tree” usando imagens cartesianas

Para a derivada temporal: através da diferença das imagens numa região de 3×3 “pixels” e o resultado é dividido por 9.

2. Para as derivadas espaciais: o vector $\frac{1}{12} [1, \Leftrightarrow 8, 0, 8, 1]$, é usado para preencher uma matriz de 5×5 onde cada linha é uma cópia do vector. Para a derivada temporal: através da diferença das imagens numa região de 5×5 “pixels” e o resultado é dividido por 25.
3. Para as derivadas espaciais: a matriz $[[\Leftrightarrow 1, 1], [\Leftrightarrow 1, 1]]$ para a direcção u e na direcção v pela sua transposta, em cada caso, o resultado é dividido por 4.0. Para a derivada temporal: através da diferença das imagens numa região de 2×2 “pixels” e o resultado é dividido por 4.
4. Para as derivadas espaciais: o vector $\frac{1}{12} [1, \Leftrightarrow 8, 0, 8, 1]$ para a direcção u e na direcção v pelo seu vector transposto. Para a derivada temporal: através da diferença das imagens numa região de 2×2 “pixels” definida pela matriz $[[0, 1, 0], [1, 0, 1], [0, 1, 0]]$ e o resultado é dividido por 4.
5. Para as derivadas espaciais: o vector $[\Leftrightarrow 1, 0, 1]$ para a direcção u e na direcção v pelo seu vector transposto, o resultado é dividido por 2. Para a derivada temporal: através da diferença das imagens numa região de 2×2 “pixels” definida pela matriz $[[0, 1, 0], [1, 0, 1], [0, 1, 0]]$ e o resultado é dividido por 4.

As tabelas 3.1, 3.2, 3.3, 3.4 apresentam os resultados para as sequências de imagens “translational and diverging tree sequences”. Através dos resultados apresentados, verifica-se que o método para estimar as derivadas desempenha um papel bastante importante na performance da técnica utilizada, enquanto para os filtros experimentados os resultados apresentam performances semelhantes [F. Bergholm 93]. Em geral, pode-se concluir que a performance da detecção do fluxo óptico feito sobre imagens log-polar é mais baixa do que em imagens cartesianas, no entanto os cálculos são mais rápidos (em média, para as imagens testadas apenas é necessário efectuar 28% dos cálculos relativamente ao caso cartesiano). Os resultados para a sequência de imagens divergente em log-polar são melhores quando comparados com a sequência de translação. Este facto

<i>Método</i>	Sequência Divergente			Sequência de translação		
	<i>Ave.</i> Err.	<i>Sta.</i> Dev.	<i>Den.</i>	<i>Ave.</i> Err.	<i>Sta.</i> Dev.	<i>Den.</i>
Fleet ($\tau = 1.25$)	2.43	1.84	83%	28.01	14.38	82%
Anandan	24.20	14.66	100%	30.94	15.52	100%
Horn (modificado) $\ \nabla I\ \geq 5.0$	4.98	4.01	100%	28.38	13.21	100%
Nagel $\ \nabla I\ _2 \geq 5.0$	5.74	5.72	100%	37.52	30.17	100%
Lucas ($\lambda_2 \geq 1.0$)	4.60	3.20	80%	28.00	13.75	80%
Singh (<i>passo</i> = 2, <i>n</i> = 2, <i>w</i> = 2, <i>N</i> = 4)	18.57	7.21	100%	51.48	26.37	100%

Tabela 3.5: Resultados da estimação do fluxo óptico usando imagens log-polar para as sequências de translação e divergente, usando técnicas descritas em [J. Barron e D. Fleet 94]

confirma que as imagens log-polar são uma boa ferramenta para detectar campos de velocidades dessa natureza (em alguns algoritmos a performance é superior ao respectivo caso cartesiano). É importante referir que a transformação log-polar foi efectuada com $N_\theta = 150$, e $a = b = 1.025$, o que representa neste caso 28% da informação contida na imagem cartesiana. Entre os algoritmos afim, verifica-se também que o mais consistente é o “*método afim por regiões*”, devido às imagens visualizadas apresentarem descontinuidades em profundidade.

Os resultados indicados na tabela 3.5, foram obtidos através da implementação descrita em [J. Barron e D. Fleet 94], mas usando a representação log-polar da imagem. O respectivo fluxo correcto foi convertido também para log-polar através da equação 2.35, e serve de base de comparação com os fluxos obtidos dos respectivos algoritmos, utilizando as medidas estatísticas indicadas anteriormente. Através dos resultados, pode-se constatar que estes métodos “off-line” no caso da sequência de translação apresentam também um erro elevado, mas para o caso da sequência divergente, os resultados são na realidade bastante bons (deve-se salientar que todos os métodos implementam a filtragem usando mais do que duas imagens para estimar a derivada temporal, enquanto os métodos apresentados nas tabelas 3.1, 3.2, 3.3, 3.4 apenas usam duas imagens, e as imagens não são filtradas temporalmente).

Método de FLEET	Sequência Divergente			Sequência de Translação		
	<i>Ave.</i> Err.	<i>Sta.</i> Dev.	<i>Den.</i>	<i>Ave.</i> Err.	<i>Sta.</i> Dev.	<i>Den.</i>
1.025	2.43	1.84	83%	28.01	14.38	82%
1.020	5.41	3.83	68%	27.18	13.99	68%
1.015	12.02	7.39	51%	25.80	12.53	54%
1.010	21.16	9.47	38%	30.75	21.58	33%

Tabela 3.6: Aplicação do método de Fleet à sequência divergente e de translação da árvore, usando diferentes bases para a transformação log-polar (mantendo a mesma resolução angular)

Os resultados indicados na tabela 3.6, foram obtidos através do método de Fleet, aplicado às sequências de translação e divergente em log-polar, mas variando a base logarítmica da transformação log-polar, mantendo a mesma resolução angular. Dos resultados torna-se evidente que a escolha da base do log-polar é decisiva para a performance obtida para o fluxo calculado. Pode-se de certa forma afirmar que a base do log-polar desempenha o papel de sintonizar o fluxo divergente que se pretende detectar coerentemente.

Em conclusão, estes resultados indicam que em geral o cálculo do fluxo óptico sobre imagens log-polar é mais ruidoso do que sobre imagens cartesianas, no entanto para fluxos divergentes, o log-polar apresenta boas performances com uma quantidade de cálculos inferior, o que neste caso é útil, pois é esta a configuração de fluxo que pretendemos detectar.

Capítulo 4

Determinação de Parâmetros do Movimento

- Determinação da direcção de translação

Este capítulo reflecte o nosso esforço para implementar uma solução que resolva o problema da navegação de um observador móvel, de modo a detectar a sua direcção de translação e manter a câmara orientada para essa direcção apesar das eventuais alterações de movimento do observador. O algoritmo que detecta a direcção do movimento explora as propriedades geométricas da transformação log-polar. Os ensaios experimentais foram efectuados numa plataforma móvel equipada com um sistema de visão activa descrito no Apêndice B.

Um observador móvel que se movimenta num meio ambiente desconhecido necessita de obter a informação visual mais rica possível, posicionando o sensor de forma a captar a informação necessária à actividade que o observador está a realizar. Considera-se que a melhor informação visual é por exemplo, possuir as melhores condições de iluminação, de focagem e o melhor alinhamento para o sensor visual. Neste capítulo, apresenta-se um algoritmo que permite controlar o sensor visual de modo que a informação visual captada seja útil, i.e., ajusta-o para uma direcção preferida enquanto em movimento.

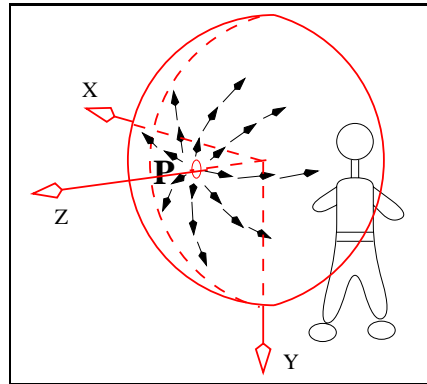


Figura 4.1: Campo de fluxo óptico visualizado por um observador que se movimenta na direcção do pólo \mathbf{P} (neste caso direcção do eixo \mathbf{Z}).

Neste trabalho considera-se como direcção preferida a direcção definida pelo centro de perspectiva e o foco de expansão.

Nas tarefas de navegação através de visão, o conhecimento do foco de expansão pode ser usado para determinar e controlar o nosso próprio movimento translacional no espaço. Este facto é importante, porque com a posição do foco de expansão é possível estimar alguns dos parâmetros de velocidade do agente ou observador em movimento. Outra possibilidade pode ser a utilização desse mesmo parâmetro para alinhar o sensor visual com a direcção de translação do sistema.

Num sistema de visão activa podemos definir uma direcção preferencial de forma a permitir captar informação importante para a navegação. No nosso caso esta direcção corresponde à direcção de translação do observador. Um sistema visual que, activamente tente manter essa direcção é equivalente a manter constante a posição do foco de expansão no centro da imagem obtida do sensor visual.

4.1 Introdução

Um observador através da sua visão (ou câmaras no caso dum observador artificial), quando se movimenta com uma velocidade de translação num meio ambiente estático, observa a projecção dos deslocamentos nas imagens de uma forma similar ao apresentado na Fig. 4.1. Se o observador mantiver constante a sua velocidade de translação, o

pólo frontal, equivalente ao ponto central do fluxo radial, será estacionário. Também se verifica, que para um objecto estacionário que se encontre numa direcção diferente da direcção do pólo frontal, as imagens retiniais desses objectos movem-se mais rapidamente em função do desvio angular relativo à direcção definida pelo pólo frontal e pelo centro de coordenadas (veja-se Fig. 4.1), se estiverem à mesma distância.

Contudo, se num observador os olhos, corpo ou cabeça (respectivamente câmaras, robot móvel e sistema de visão num sistema artificial) induzem um movimento rotacional, o pólo \mathbf{P} não mantêm a mesma estabilidade posicional.

Uma forma de representar o efeito de projecção do movimento de um observador é através de uma esfera [F. Bergholm 90]. A esfera que representa o mundo visível é apenas uma elegante idealização matemática que à partida é suficiente para representar os campos de fluxo que podem ser gerados pelos movimentos do observador. Na prática, esta estrutura de representação pode ser simulada parcialmente através do uso de câmaras normais. A imagem capturada pelas câmaras é equivalente a efectuar uma amostra da superfície esférica. Normalmente esta amostra possui informação visual suficiente para ser utilizada no controlo da actividade do observador. Neste e no próximo capítulo apresentam-se dois algoritmos para navegação com base neste princípio.

Quando um observador se movimenta no espaço, para se obter a máxima informação possível, os olhos devem estar alinhados com a direcção do movimento. Só desta forma é possível detectar obstáculos, corrigir trajectórias de navegação, etc. Todos os observadores humanos no seu dia a dia, efectuam comportamentos deste género sem se aperceberem.

Uma das direcções preferenciais do alinhamento do sensor visual é a direcção de translação. Desta forma o conhecimento do pólo frontal é crucial para efectuar este género de comportamento. Neste caso o objectivo é manter o pólo frontal no centro do campo visual. Este capítulo reflecte o estudo da implementação deste comportamento específico, através de experiências simuladas e reais. Descreve-se neste capítulo um algoritmo para detectar a direcção preferida do sensor visual dum observador móvel e manter essa mesma direcção. O algoritmo detecta a direcção do movimento translacional e explora as propriedades geométricas da transformação log-polar para detectar o pólo frontal \mathbf{P} - veja-se Fig. 4.1.

Nas próximas secções será então introduzido em maior pormenor o processo que foi

brevemente descrito nesta introdução.

4.2 Estimação do foco de expansão

Esta secção trata do problema da estimação da direcção do vector de velocidade de translação, através da medida das velocidades sobre imagens, quando um observador se movimenta num ambiente estático. A intercepção do vector de movimento translacional, com o plano imagem gera um ponto que se denomina como foco de expansão ou abreviadamente por ponto *FOE* [R. Guissin e S. Ullman 91], [C. Fermuller e Y. Aloimonos 91].

Em alguns casos especiais (baixa velocidade de rotação) os vectores de fluxo induzidos na imagem devido ao próprio movimento do observador, apresentam algumas propriedades geométricas, que podem ser úteis para detectar na imagem o local onde se encontra o foco de expansão. Estas propriedades geométricas, baseiam-se no facto de que os vectores de fluxo são radiais a partir da posição do foco de expansão, e que a linha que passa pelo foco de expansão e pelo centro da imagem apenas possui fluxo radial conforme pode ser visto na Fig. 4.2.

Usando estes factos, pode-se estabelecer um mecanismo de procura da posição do *FOE*. O processo baseia-se na detecção da orientação angular (direcção θ_{FOE}) da linha que passa pelo centro da imagem e pelo *FOE*, através dum processo de procura para cada orientação possível, escolhendo-se a orientação que apresenta menor fluxo circular. De seguida para detectar a posição do *FOE*, faz-se uma translação do sistema de coordenadas para cada ponto pertencente à linha encontrada, até se encontrar um ponto, do qual os vectores de fluxo são radiais (a este método chamamos de retina dinâmica).

Em navegação através de visão, o conhecimento do parâmetro *FOE*, pode ser utilizado para determinar e controlar o nosso próprio movimento no espaço. No nosso caso iremos demonstrar que é possível detectar a direcção de translação do nosso próprio movimento, através de testes efectuados num robot móvel simulado e real e com câmaras com graus de liberdade de “pan”, “tilt” e vergência (veja-se Apêndice C). No caso do robot móvel simulado, a câmara encontra-se perto do eixo de coordenadas de rotação do robot, de modo a simplificar a detecção do *FOE*, ou caso contrário será impossível medir com

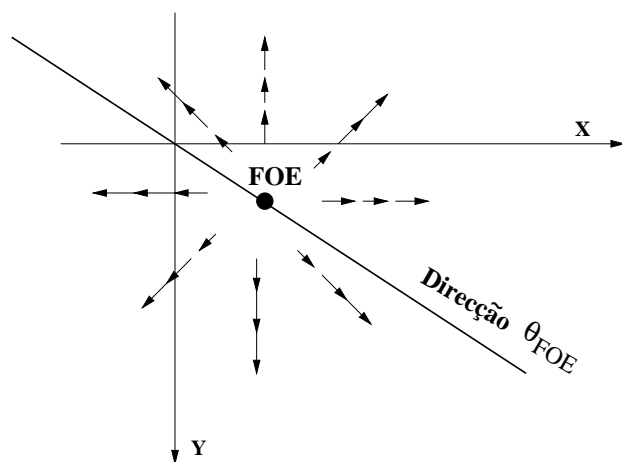


Figura 4.2: Propriedade geométrica dos vetores de fluxo, quando o observador apresenta uma velocidade de translação.

precisão essa mesma posição. O facto de se poder considerar que o robot se movimentava num plano horizontal é importante, porque a velocidade translacional é perpendicular ao vector velocidade rotacional, o que simplifica a detecção da direcção θ_{FOE} , tal como será descrito nas próximas secções. O robot simulado é idêntico ao robot descrito no Apêndice B.

Em primeiro lugar, será dada uma apresentação das equações de velocidade no plano cartesiano, através do uso da transformação de coordenadas cartesianas para coordenadas polares de modo a adquirir o conhecimento de como se comporta o fluxo induzido na direcção θ_{FOE} , quando a restrição descrita atrás é utilizada para detectar a linha (restrição: vector velocidade translacional é perpendicular ao vector velocidade rotacional).

De seguida, mostra-se que existe uma relação muito próxima entre os fluxos induzidos descritos em coordenadas polares e em coordenadas log-polares, o que permite expandir o método ao caso de imagens log-polar. Este facto é importante, porque através do uso de imagens log-polar, o processo é mais rápido do que com imagens cartesianas (um exemplo de um trabalho com algumas semelhanças pode ser visto em [R. Guissin e S. Ullman 91]).

Finalmente, serão apresentados os resultados experimentais, no final desta secção 4.2.

4.2.1 Equações de velocidade expressas em coordenadas polares

Para obter a expressão da velocidade do observador em coordenadas polares é necessário conhecer os seus parâmetros de movimento: velocidade rotacional e translacional. À medida que o observador se movimenta no espaço tridimensional não homogéneo, as imagens adquiridas sofrem alterações. Estas alterações do fluxo óptico induzido na imagem podem ser formuladas através da equação 2.29. As equações de velocidade no plano cartesiano expressas em coordenadas polares, podem ser encontradas através da transformação 2.35 aplicada à equação de movimento 2.29. Após essa transformação, obtém-se a seguinte relação (onde T_c^p representa a transformação):

$$\begin{bmatrix} \dot{\rho} \\ \dot{\theta} \end{bmatrix} = \frac{1}{Z} T_c^p A \vec{v}_t + T_c^p B \vec{\omega} \quad (4.1)$$

Quando o observador se movimenta com uma velocidade de translação não nula e uma velocidade de rotação nula, a velocidade (fluxo óptico teórico) (\dot{u}, \dot{v}) no ponto do foco de expansão é nula. Por definição o FOE corresponde ao ponto de intercepção do vector de movimento translacional com o plano imagem, e tem por coordenadas:

$$(FOE_x, FOE_y) = \left(f_x \frac{V_x}{V_z}, f_y \frac{V_y}{V_z} \right) \quad (4.2)$$

Com esta definição, a linha que passa pelo centro da imagem e pelo ponto de FOE apresenta as seguintes relações:

$$\begin{cases} \sin(\theta) = \frac{f_y V_y}{\sqrt{(f_x V_x)^2 + (f_y V_y)^2}} \\ \cos(\theta) = \frac{f_x V_x}{\sqrt{(f_x V_x)^2 + (f_y V_y)^2}} \end{cases} \quad (4.3)$$

Chama-se a atenção do leitor para o seguinte facto: aquilo que se pretende numa primeira fase é detectar a orientação da recta definida pelo foco de expansão e pelo centro da imagem (recta que contém a origem e o FOE). O algoritmo para determinar essa orientação efectua uma pesquisa em todas as orientações possíveis, até encontrar a orientação onde o fluxo circular é menor. A equação do fluxo circular encontra-se na segunda linha do sistema matricial indicado na equação 4.1. No entanto, esta equação

depende da profundidade Z de cada ponto considerado, o que à partida inviabiliza o processo de minimização, pois não existe forma de conhecer a profundidade de cada ponto. Tal facto pode ser ultrapassado através de restrições impostas aos vectores velocidade de modo que a equação para $\dot{\theta}$ obtida a partir de 4.1 não dependa de Z . Uma das restrições que verifica tal facto é:

$$v_x \cdot w_x = \Leftrightarrow v_y \cdot w_y \quad (4.4)$$

Esta equação pode também ser escrita na forma vectorial através de:

$$\begin{bmatrix} v_x & v_y \end{bmatrix} \cdot \begin{bmatrix} w_x \\ w_y \end{bmatrix} = 0 \quad (4.5)$$

o que é equivalente a afirmar que estes dois vectores devem ser perpendiculares¹. Note-se que os vectores encontram-se no mesmo plano.

Utilizando as equações 4.4 e 4.3, em conjugação com a equação 4.1 obtemos (na direcção θ_{FOE}):

$$\rho \cdot \dot{\theta} = \rho \cos^2(\theta) w_z \left[\frac{f_x}{f_y} \Leftrightarrow \frac{f_y}{f_x} \right] \Leftrightarrow \rho w_z \frac{f_x}{f_y} \quad (4.6)$$

e (na direcção perpendicular à direcção θ_{FOE}):

$$\dot{\rho} = \rho^2 \left(\sin(\theta) \frac{w_x}{f_y} \Leftrightarrow \cos(\theta) \frac{w_y}{f_x} \right) + \rho \cos(\theta) \sin(\theta) w_z \left[\frac{f_x}{f_y} \Leftrightarrow \frac{f_y}{f_x} \right] + \frac{\rho v_z}{Z} \quad (4.7)$$

Verifica-se assim, que a equação 4.6 não depende de Z e apenas apresenta dependências de w_z , sendo os outros parâmetros fixos para um determinado θ e ρ .

De momento, interessa também constatar que se pode obter semelhantes resultados para o log-polar, no entanto na secção 4.2.2 será descrito a forma de utilizar as equações 4.6 e 4.7, para efectuar a detecção da direcção θ_{FOE} .

Como será visto na secção que se segue, pode-se efectuar a mesma análise para o caso do plano log-polar e as equações obtidas são muito semelhantes às equações 4.6 e 4.7, a menos de determinados factores constantes. Tal facto permite pensar que será viável utilizar o fluxo óptico medido em imagens log-polar para estimar a posição do foco de expansão.

¹No caso do robot móvel, esta restrição é desnecessária, porque os movimentos possíveis do robot são descritos por $\vec{v}_i = [0, 0, v_z]^T$ e $\vec{\omega} = [0, \omega_y, 0]^T$ (veja-se Apêndice B)

Extensão para o plano log-polar

Para o caso do plano log-polar, usando a equação de fluxo induzido 2.29, mas agora através da transformação dada pela equação 2.37, permite obter uma equação semelhante à equação 4.1. Esta equação pode ser expressa por (onde T_c^{lp} representa a transformação):

$$\begin{bmatrix} \dot{\zeta} \\ \dot{\gamma} \end{bmatrix} = \frac{1}{Z} T_c^{lp} A \vec{v}_t + T_c^{lp} B \vec{w} \quad (4.8)$$

Usando novamente os resultados das equações 4.4, 4.3, mas agora na equação 4.8, chega-se à seguinte expressão (na direcção θ_{FOE}):

$$\rho_{fovea} a^\zeta \cdot \Delta\theta \cdot \dot{\gamma} = \rho_{fovea} a^\zeta \cos^2(\Delta\theta \cdot \gamma) w_z \left[\frac{f_x}{f_y} \Leftrightarrow \frac{f_y}{f_x} \right] \Leftrightarrow \rho_{fovea} a^\zeta w_z \frac{f_x}{f_y} \quad (4.9)$$

e a (na direcção perpendicular à direcção θ_{FOE}):

$$\begin{aligned} \rho_{fovea} a^\zeta \cdot \ln(a) \cdot \dot{\zeta} &= (\rho_{fovea} a^\zeta)^2 \left(\sin(\Delta\theta \cdot \gamma) \frac{w_x}{f_y} \Leftrightarrow \cos(\Delta\theta \cdot \gamma) \frac{w_y}{f_x} \right) \\ &+ \rho_{fovea} a^\zeta \cos(\Delta\theta \cdot \gamma) \sin(\Delta\theta \cdot \gamma) w_z \left[\frac{f_x}{f_y} \Leftrightarrow \frac{f_y}{f_x} \right] + \frac{\rho_{fovea} a^\zeta v_z}{Z} \end{aligned} \quad (4.10)$$

Estas duas equações são semelhantes às equações do caso polar, e como para ambos os casos, para efectuar a procura da direcção angular é necessário utilizar um determinado valor angular incremental, as equações na realidade, representam a mesma entidade (a diferença reside nos valores de escalonamento, e na amostragem radial).

Para um melhor esclarecimento dos algoritmos propostos, nas próximas secções define-se a variável r como $r = \rho$ para o caso polar e $r = \rho_{fovea} a^\zeta$ para o caso log-polar e também a função $F_\theta(r, \theta)$ como $F_\theta(r, \theta) = r \cdot \dot{\theta}$ para o caso polar e $F_\theta(r, \theta) = r \cdot \Delta\theta \cdot \dot{\gamma}$ para o log-polar.

A estimação da posição do FOE é feita em duas etapas em sequência:

- a detecção da orientação da linha que passa na direcção θ_{FOE} , através do uso de um algoritmo de procura que minimiza uma função $F_\theta(r, \theta)$, processo este que se denomina de “*Detecção da direcção θ_{FOE}* ”
- a procura em todos os pontos da linha encontrada, do ponto a partir do qual, o fluxo é radial, processo este que se denomina por “*Detecção da posição do FOE* ”

4.2.2 Detecção da direcção do foco de expansão

Nesta secção aborda-se o problema de detectar a direcção θ_{FOE} , que basicamente é conseguida através dum algoritmo de procura unidimensional, de entre o espaço de todas as orientações possíveis dentro dum círculo unitário (usando um valor incremental $\Delta\theta$ para o ângulo), usando a equação 4.6 ou a equação 4.9.

Para cada uma das direcções, através do uso de n pontos arbitrários, estima-se a seguinte quantidade:

$$M_\theta = \cos^2(\Delta\theta \cdot \gamma) w_z \left[\frac{f_x}{f_y} \Leftrightarrow \frac{f_y}{f_x} \right] \Leftrightarrow w_z \frac{f_x}{f_y}$$

através do uso da seguinte relação:

$$M_\theta = \frac{\sum_{k=0}^{n-1} F_\theta(r_k, \theta)}{\sum_{k=0}^{n-1} r_k} \quad (4.11)$$

e de seguida acumulando uma estimativa de erro E_θ para cada direcção:

$$E_\theta = \sum_{k=0}^{n-1} [F_\theta(r_k, \theta) \Leftrightarrow r_k M_\theta]^2 \quad (4.12)$$

Após a construção desta função, a direcção θ_{FOE} é seleccionada como sendo a posição onde a função E_θ é mínima. No caso do FOE estar presente no centro da imagem, a função E_θ deveria ser constante, logo sem mínimos, este problema pode ser resolvido através do uso dum limiar aplicado à função E_θ . No entanto os casos ambíguos poderão ser resolvidos, se à partida se tiver uma estimativa do movimento (i.e. ter o conhecimento em que quadrante se encontra o FOE).

4.2.3 Detecção da posição do foco de expansão

Após a detecção da direcção da recta que passa pela posição do FOE , é necessário agora utilizar um algoritmo que permita estimar a posição do FOE na recta previamente identificada. Para efectuar esta estimação realizaram-se testes experimentais com os seguintes algoritmos:

- *Algoritmo da linha* - este processo apenas usa os vectores de fluxo pertencentes à linha θ_{FOE} de modo a detectar o ponto onde o fluxo radial muda de sentido. Se

algum ponto for encontrado, esse é considerado como a posição do *FOE*. Noutra situação o processo falha.

- *Algoritmo da FFT* - explora o conceito da “Retina Dinâmica” através da procura em cada ponto da linha θ_{FOE} , por aquele onde o respectivo histograma radial é constante. Esse ponto é o *FOE*. A FFT é usada para detectar a forma do histograma.
- *Algoritmo do Desvio Padrão* - este algoritmo é semelhante ao algoritmo da FFT, mas, usa neste caso, o desvio padrão do histograma de fluxos radiais, o que permite detectar a forma do gráfico do histograma. Interessa-nos tal como no *Algoritmo da FFT* detectar gráficos que apenas contenham uma componente *DC*.

Algoritmo da linha

Este algoritmo baseia-se na constatação de que os vectores de fluxo divergem do *FOE*, sendo assim possível utilizar um processo de procura na linha θ_{FOE} , de modo a encontrar o local onde se verifica a inversão do sentido dos vectores de fluxo (basta apenas verificar esta condição, a nível do fluxo radial, uma vez que a linha θ_{FOE} minimiza o fluxo angular).

Efectuando as convenções que se indicam de seguida, é possível construir uma função unidimensional de procura, onde o *FOE* se encontra no ponto associado à passagem por zero da função. Considere-se para isso as seguintes convenções:

- a distância radial dos pontos na semi-linha θ_{FOE} é positiva e a distância radial dos pontos da semi-linha $\theta = \theta_{FOE} + \pi$ é negativa. Esta definição pode ser utilizada para definir uma variável que constituirá a abcissa da função de procura e que é utilizada para indicar qual é o ponto da recta sob análise.
- o sentido positivo do fluxo radial é no sentido da semi-linha θ_{FOE} , e negativo no sentido $\theta = \theta_{FOE} + \pi$. Pode-se assim definir a ordenada da função de procura, com a particularidade de se obter um gráfico muito próximo duma recta.

No caso da recta do gráfico da função de procura passar por zero dentro dos limites das abcissas da função, o *FOE* encontra-se dentro da imagem, caso contrário poderá ser atribuído ao respectivo limite, dependendo do declive.

Como é óbvio, na presença de rotações, o que foi descrito anteriormente é inválido, mas nesse caso em vez duma linha teremos uma parábola, tal como foi constatado experimentalmente. Nesta situação, apenas é necessário encontrar o ponto mínimo ou máximo da parábola, conforme a concavidade desta, ou então o local onde esta intercepta o eixo do x . Este pressuposto apenas foi validado experimentalmente para pequenos valores de w_x , w_y e w_z , não sendo verdadeiro noutros casos.

Para estimar o FOE , foi assim utilizado um algoritmo de mínimos quadrados que permite identificar a forma que mais se aproxima do gráfico da função de procura construída, usando o modelo

$$y = a.x^2 + b.x + c \quad (4.13)$$

cuja solução é dada por (usando m pontos):

$$\begin{bmatrix} \sum_{k=0}^{m-1} x^4[k] & \sum_{k=0}^{m-1} x^3[k] & \sum_{k=0}^{m-1} x^2[k] \\ \sum_{k=0}^{m-1} x^3[k] & \sum_{k=0}^{m-1} x^2[k] & \sum_{k=0}^{m-1} x[k] \\ \sum_{k=0}^{m-1} x^2[k] & \sum_{k=0}^{m-1} x[k] & m \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \sum_{k=0}^{m-1} y[k].x^2[k] \\ \sum_{k=0}^{m-1} y[k].x[k] \\ \sum_{k=0}^{m-1} y[k] \end{bmatrix} \quad (4.14)$$

onde a, b, c são encontrados através da resolução do sistema (a função parabólica representa também uma linha, quando necessário, pelo que é suficiente usar este modelo, para contemplar os dois casos descritos anteriormente).

Algoritmo da FFT

Este algoritmo utiliza o conceito da “Retina Dinâmica” introduzido na secção 2.3. Para estimar a posição do FOE , a retina é centrada em cada ponto pertencente à linha θ_{FOE} , e para cada um destes pontos, é efectuado um histograma dos fluxos radiais. Note-se que o fluxo radial para cada ponto inspeccionado é definido relativamente ao centro da retina, e por isso diferente do fluxo radial definido relativamente ao centro da imagem (fluxo radial detectado pelo algoritmo de cálculo do fluxo óptico). Esta diferença pode ser vista através da Fig. 4.3. Para efectuar o histograma é necessário projectar o fluxo óptico segundo a direcção radial da retina que está a ser inspeccionada. Se a retina se encontrar centrada no FOE , o fluxo óptico será radial, e apenas para este caso, o histograma apresentará um valor constante para qualquer das direcções envolvidas no cálculo. Caso contrário,

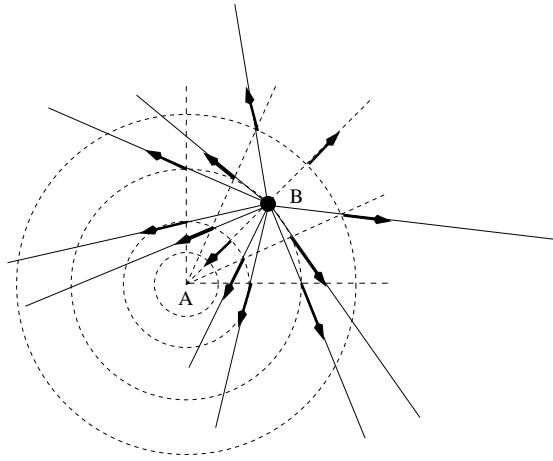


Figura 4.3: Exemplo do conceito da “Retina Dinâmica” aplicada ao caso log-polar, através da utilização de 8 direcções distintas (o ponto *A* indica o centro do log-polar e o ponto *B* o centro da retina dinâmica).

o histograma apresentará termos modulados. Estes termos, aparecem no histograma, porque quando a retina não se encontra centrada no *FOE*, a componente radial utilizada no histograma é diferente para diferentes direcções angulares, mesmo que o módulo do fluxo seja igual. A sugestão para este processo, é neste caso, utilizar o algoritmo da *FFT*, para detectar qual dos histogramas apresenta apenas um termo constante.

O conceito da retina dinâmica é facilmente aplicável a imagens cartesianas discretas, ou a imagens log-polar contínuas, no entanto, quando se utilizam imagens log-polar discretas, a descrição apresentada na secção 2.3 não pode ser usada devido à falta de amostras, quando se pretende mudar o ponto de aplicação da retina. Qual será então a solução? De modo a evitar efectuar uma amostragem log-polar para cada retina, é neste caso efectuada a seguinte aproximação: para realizar o histograma são apenas inspeccionadas 8 direcções angulares ($0, 45^{\circ}, 90^{\circ}, \dots$) da imagem log-polar (usando apenas 4 pontos para cada direcção radial). Para cada direcção inspeccionada, é necessário converter o fluxo óptico para a respectiva direcção radial da retina. Tal processo passa-se como se a amostragem log-polar tivesse sido efectuada no local onde se encontra o centro da retina dinâmica (isto apenas é válido, se o número de amostras angulares da imagem log-polar for elevado). A Fig. 4.3 exemplifica o que foi referido.

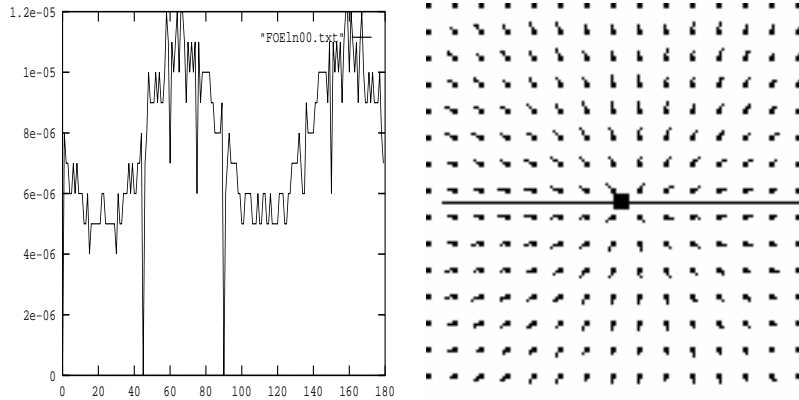


Figura 4.4: Neste exemplo, $\vec{v} = [0, 0, 1]^T$ (cm/s) e $\vec{\omega} = 0$. À esquerda encontra-se o resultado da detecção da direcção θ_{FOE} , e à direita o campo de fluxo óptico com a linha θ_{FOE} , onde o quadrado a preto indica a posição detectada para o FOE . As distâncias focais são $f_x = f_y = 15.0\text{cm}/\text{pixel}$. Neste caso foram utilizadas 180 amostras angulares.

Devido à zona da fovea, que não é transformada para o plano log-polar, é neste caso bastante difícil detectar a posição do FOE , se este se encontrar nessa zona (é bastante difícil neste caso usar um esquema de interpolação, para obter informação na zona da fovea).

Algoritmo do Desvio Padrão

Este algoritmo é bastante semelhante ao anterior, no entanto, em vez de usar uma FFT para detectar a forma do gráfico dos histogramas, utiliza o desvio padrão de modo a seleccionar qual o histograma que possui um gráfico constante. O FOE encontra-se assim no ponto que deu origem ao histograma com um menor valor de desvio padrão. Nas experiências realizadas, em geral, é utilizado um método de interpolação de modo a extrapolar o valor da variância na zona da fovea. No entanto a utilização da interpolação normalmente conduz a resultados mais instáveis.

4.2.4 Resultados de simulação

Neste secção, são apresentados os resultados da detecção do FOE , através do uso de fluxo óptico simulado obtido directamente da equação 2.29. No primeiro exemplo, apenas se

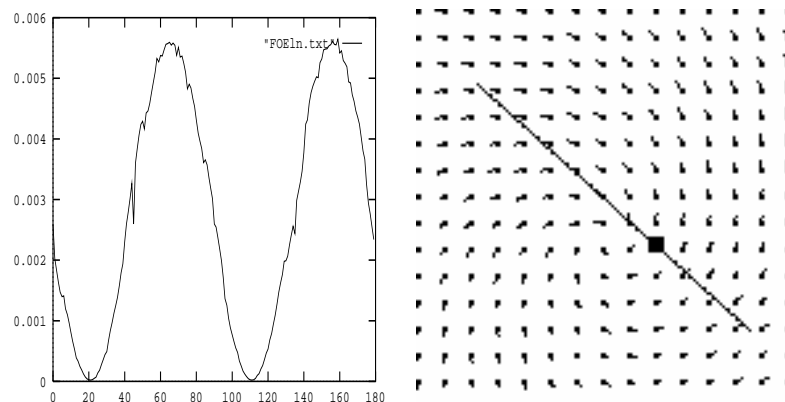


Figura 4.5: Neste exemplo, $\vec{v} = [1, 1, 1]^T$ e $\vec{\omega} = [0, 0, 0.1]^T$ (graus/s). A direcção θ_{FOE} detectada foi de 42° , uma diferença de 3° , e mesmo na presença de uma rotação de elevado valor.

usa uma translação (Fig. 4.4), na Fig. 4.5 é efectuada uma rotação em torno do eixo z e na Fig. 4.6, a detecção torna-se mais difícil porque a rotação é no eixo x , e para este caso o FOE não foi detectado correctamente, porque a restrição 4.4 não é verificada (note-se que se usa aqui fluxo óptico sem ruído).

4.2.5 Resultados com o sistema experimental

Em geral, a detecção do FOE depende da técnica de detecção do fluxo óptico. Assim sendo, uma das chaves para um sucesso na detecção do FOE é a performance da técnica de fluxo óptico utilizada. De modo aos resultados da posição do FOE serem coerentes, a percentagem de vectores de fluxo deve ser elevada, e nas experiências realizadas, os melhores resultados foram obtidos através da utilização do método afim e afim por regiões. Nas Figuras 4.8, 4.9, 4.10, 4.11 e 4.12 apresentam-se os resultados com imagens sintéticas para os métodos afins, no plano cartesiano e log-polar.

4.2.6 Principais conclusões sobre a detecção do foco de expansão

Dos algoritmos propostos para detectar a posição do FOE , as seguintes conclusões podem ser retiradas:

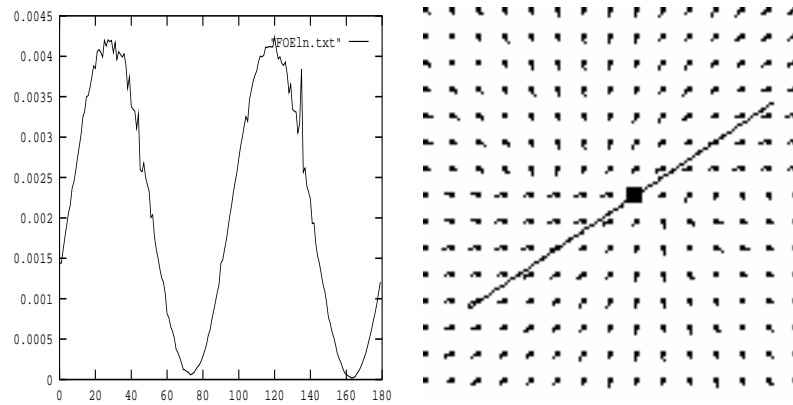


Figura 4.6: Neste exemplo, $\vec{v} = [1, 1, 1]^T$ e $\vec{\omega} = [0.1, 0, 0]^T$ (graus/s). A direcção θ_{FOE} detectada foi de 326° , o que é bastante erróneo, no entanto a restrição $v_x \cdot w_x = \Leftrightarrow v_y \cdot w_y$ não foi verificada.

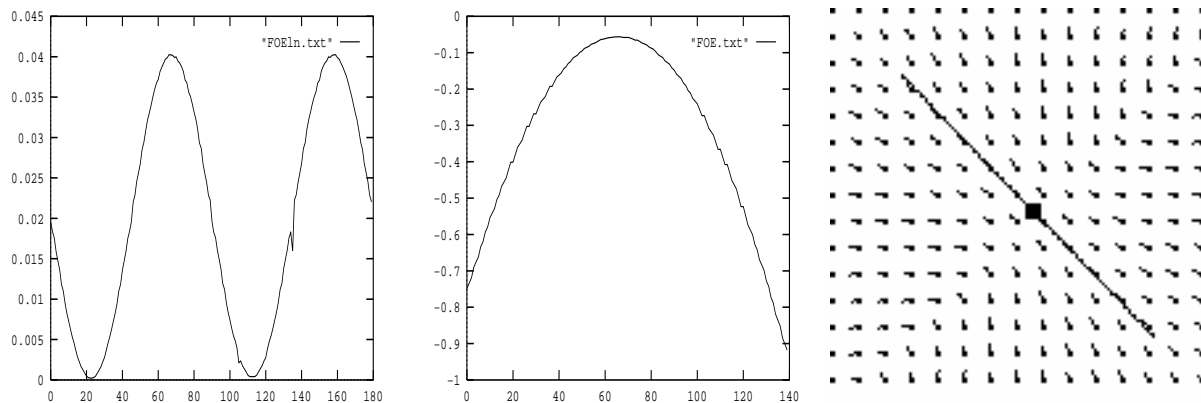


Figura 4.7: Neste exemplo, $\vec{v} = [1, 1, 1]^T$ e $\vec{\omega} = [\Leftrightarrow 0.1, 0.1, 0]^T$. A direcção θ_{FOE} detectada foi de 46° , (um erro de 1°), porque a restrição $v_x \cdot w_x = \Leftrightarrow v_y \cdot w_y$ é verificada neste caso.

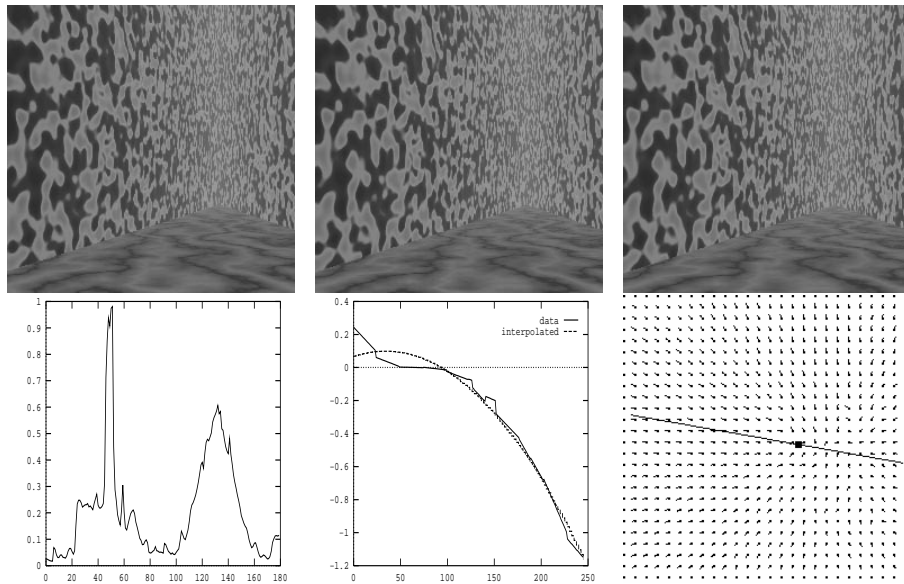


Figura 4.8: Experiência com o modelo afim por regiões. Neste exemplo, $\vec{v} = [1.0, 0, 0.2]^T$ e $\vec{\omega} = [0, 0.04, 0]^T$. A direcção θ_{FOE} detectada foi 10.0° , e o ponto FOE em $(28.852341, 5.087448)$. Este resultado assume $f_x = f_y = 10.0$

- a detecção do FOE depende fortemente da estabilidade do algoritmo de fluxo óptico usado.
- para valores de rotação elevados, apesar de se verificar a restrição imposta pela equação 4.4, a detecção da orientação θ_{FOE} é bastante instável.
- o algoritmo da linha é o que apresenta resultados menos satisfatórios, porque apenas usa o fluxo da linha θ_{FOE} .
- o algoritmo da FFT não pode ser utilizado no plano log-polar, porque o histograma apresenta uma resolução angular baixa.
- o algoritmo do desvio padrão é de todos, o melhor, não só a nível da estabilidade, mas também a nível da interpolação que permite obter o FOE caso ele se encontre na zona da fovea.
- no geral, os algoritmos propostos apresentam melhores resultados quando aplicados a imagens cartesianas, no entanto o tempo de processamento é mais elevado do que

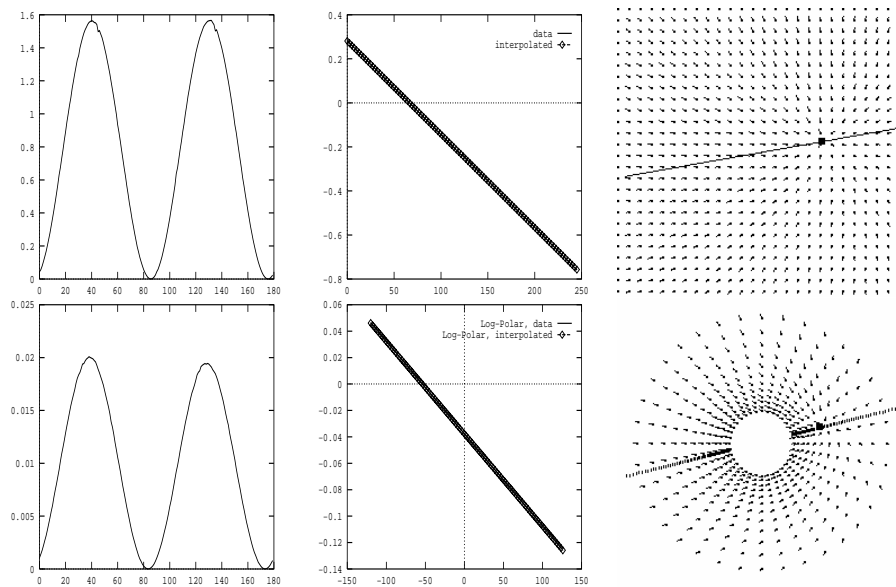


Figura 4.9: Experiência com o modelo afim. O movimento efectuado é idêntico ao da Fig 4.8. As imagens do topo representam a detecção no plano cartesiano. A direcção θ_{FOE} detectada foi 350.0° , e o $FOE=(54.810425, \Leftrightarrow 10.335442)$. As imagens em baixo referem-se ao caso log-polar, onde a direcção θ_{FOE} detectada foi 346.0° , e o $FOE=(52.330307, \Leftrightarrow 14.952591)$.

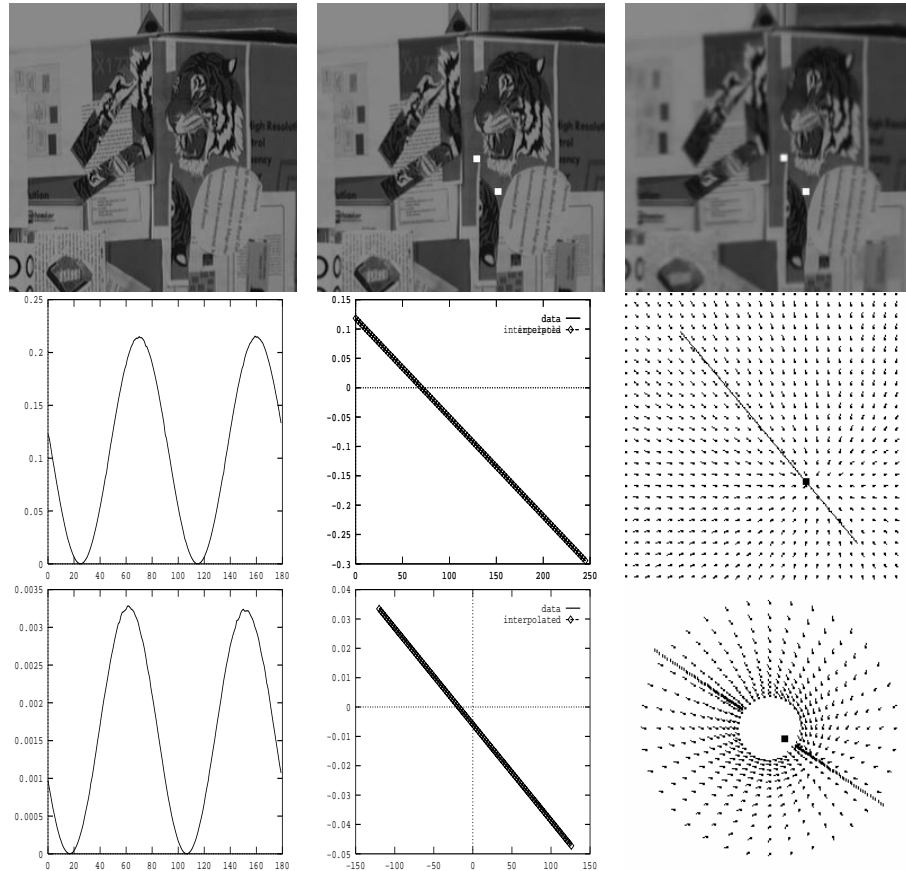


Figura 4.10: Experiência com o modelo afim. O movimento efectuado é $\vec{v} = [10.0, 10.0, 4.0]^T$ e $\vec{\omega} = [0, 0, 0]^T$, $f_x = f_y = 15.0$. O FOE real neste caso encontra-se em $(37.5, 37.5)$. Na sequência do topo os pontos a branco indicam a posição do FOE detectada. As imagens do centro representam a detecção efectuada sobre imagens cartesianas, onde a direcção θ_{FOE} detectada foi de 50.0° , e o $FOE = (33.318420, 39.707352)$. Em baixo encontra-se o resultado para o caso log-polar. A direcção θ_{FOE} detectada foi de 34.0° , e o $FOE = (14.960556, 10.091019)$. Neste caso o log-polar não detectou de forma correcta a posição do FOE .

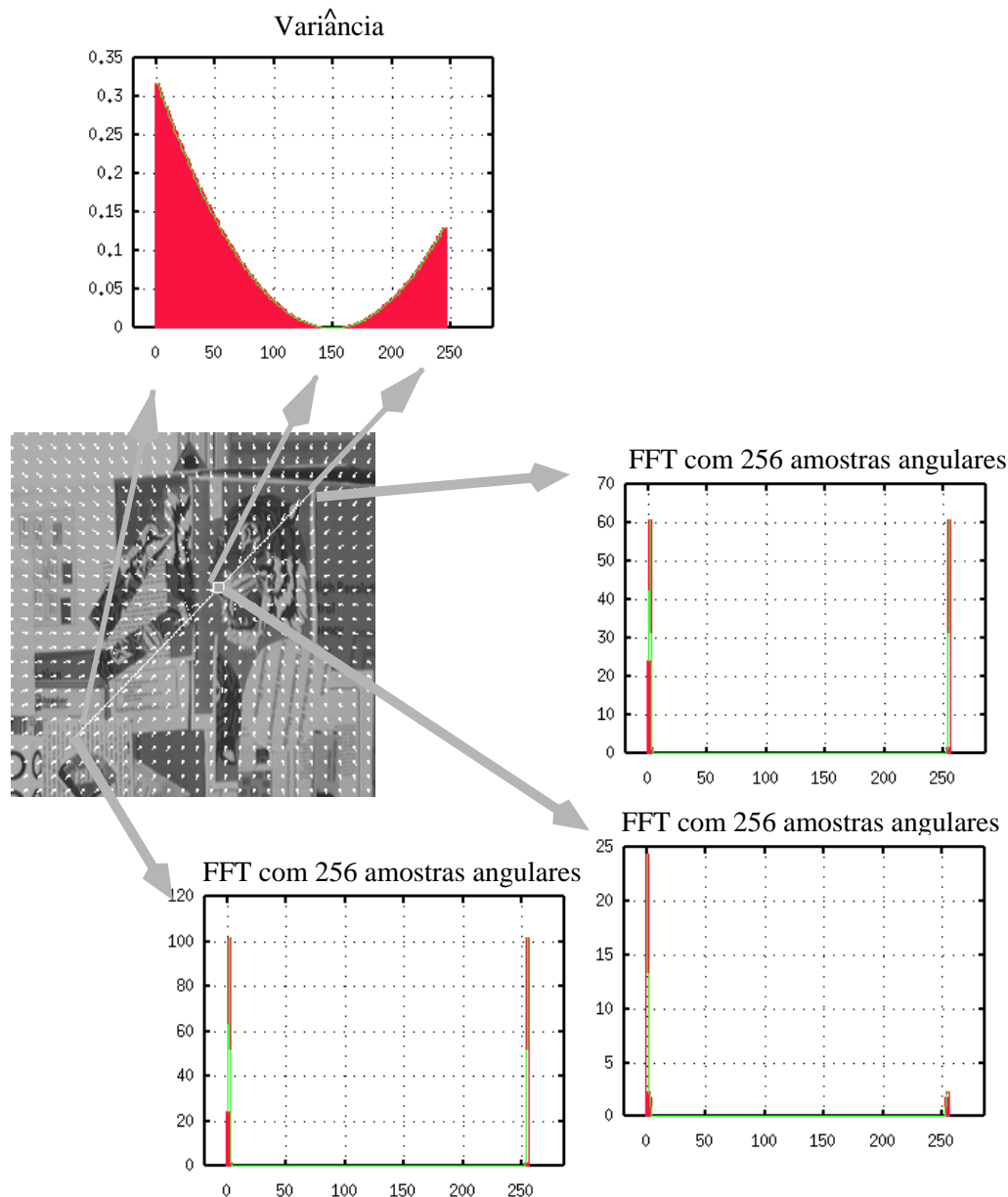


Figura 4.11: O movimento é $\vec{v} = [2.0, \Leftrightarrow 2.0, 1.0]^T$, $\vec{\omega} = [0, 0, 0]^T$, $f_x = f_y = 10.0$. a posição real do FOE é (20,-20). Os resultados para o plano cartesiano são: algoritmo da linha (19.292419,-19.977890), algoritmo da FFT (19,-21), algoritmo do desvio padrão (18.000000,-19.000000). Usando imagens log-polar obtêm-se: algoritmo da linha (8.006317,-7.731613), algoritmo do desvio padrão sem interpolação (24.000000,-23.000000) e com interpolação (7.955401,-8.238059).

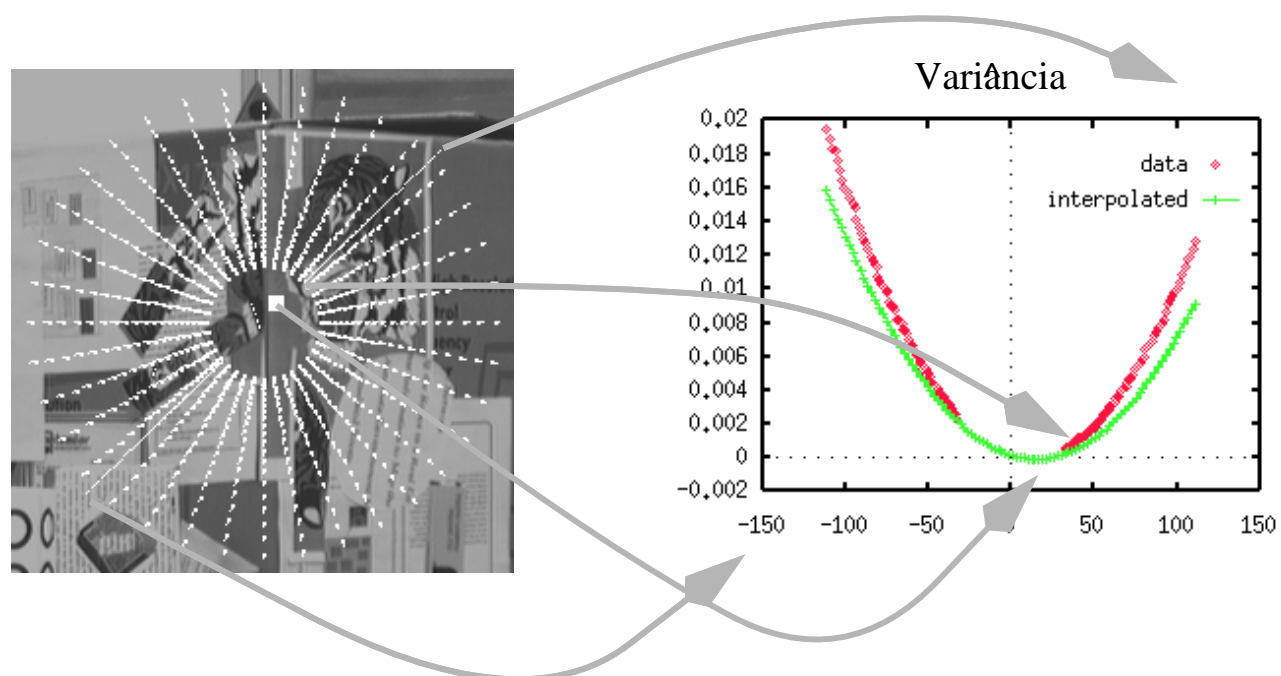


Figura 4.12: O algoritmo do desvio padrão para o caso do exemplo apresentado na Fig. 4.11

no caso do log-polar, e em geral a precisão dos resultados em imagens log-polar é suficiente.

4.3 Controlo do alinhamento de uma câmara

Nesta secção apresenta-se o algoritmo de alinhamento do sensor visual com o vector de movimento translacional - detecção da direcção preferida de visualização. Para conseguir tal comportamento, utilizou-se um dos algoritmos de detecção do *FOE* proposto na secção anterior e através dum controlador tenta-se manter a posição do *FOE* no centro das imagens (Fig. 4.14). O algoritmo de detecção utiliza imagens log-polar e para efectuar o controlo utilizou-se um sistema de eventos discretos com dois estados, que se encontra representado na Fig. 4.13. Um dos estados inicia o processo para efectuar a estimativa das coordenadas X e Y do *FOE*. O outro estado monitoriza a posição desejada da câmara, que foi calculada na malha de realimentação do controlador indicado na Fig. 4.14, utilizando as estimativas de X e Y obtidas no primeiro estado.

O sinal obtido à saída do controlador indica o erro angular relativamente à posição da câmara. No diagrama 4.14 existem dois sinais de erro X e Y o que poderia levar o leitor a pensar num sistema de duas entradas duas saídas. No entanto ambos os sinais são desacoplados, i.e., tudo se passa como se estivesse na presença de dois sistemas com uma entrada uma saída. O sinal de erro do X controla o ângulo de “Pan” da câmara, enquanto o sinal de erro do Y o “tilt”. Definindo as funções de erro como:

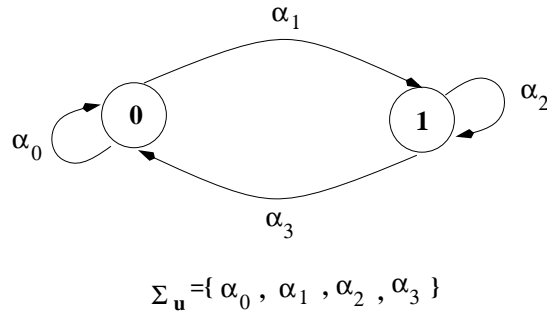
$$\begin{aligned} erro_X[n] &= X[n] \ominus X_{desejado} \\ erro_Y[n] &= Y[n] \ominus Y_{desejado} \end{aligned} \quad (4.15)$$

o controlador para cada um dos sistemas pode-se ser descrito através da seguinte expressão:

$$u[n] = k_p erro[n] + k_i \sum_{k=0}^m erro[n \ominus k] + k_d (erro[n] \ominus erro[n \ominus 1]) \quad (4.16)$$

onde os k_p , k_i e k_d são ajustados experimentalmente e a função erro será substituída pela respectiva função em cada caso. As acções do controlador são executadas repartidamente

Modelo para o controlo do alinhamento



Descrição dos Estados

- 0**: Câmara em repouso e sistema móvel em movimento
- 1**: Controlo da câmara - câmara e sistema móvel em movimento

Descrição dos eventos

- α_0 : Estima as coordenadas do *FOE*
- α_1 : Estimação do *FOE* concluída
- α_2 : O motor que controla a posição da câmara ainda não chegou à posição final indicada pelo controlador
- α_3 : O motor que controla a posição da câmara chegou à posição pretendida e encontra-se parado

Figura 4.13: Modelo do sistema de eventos discreto para o controlo do alinhamento da câmara.

por cada um dos estados discretos, i.e., no estado **0** adquirem-se as imagens para estimar o *FOE*, enquanto no estado **1** posiciona-se a câmara na posição calculada pelo controlador. Salienta-se que a posição das câmaras é controlada por motores de passo.

Na maioria dos casos, o alinhamento da câmara foi bem sucedido, devido à cena, altamente texturada e ao movimento entre imagens ser bastante pequeno, o que permite uma maior estabilidade na detecção do fluxo óptico. A Fig. 4.15 apresenta um exemplo da sequência efectuada pela câmara.

A Fig. 4.16 apresenta os resultados para o alinhamento duma câmara agora num ambiente real. Neste caso a câmara foi alinhada com o vector de translação do robot e manteve-se bastante estável como pode ser visto. No entanto para que tal acontecesse,

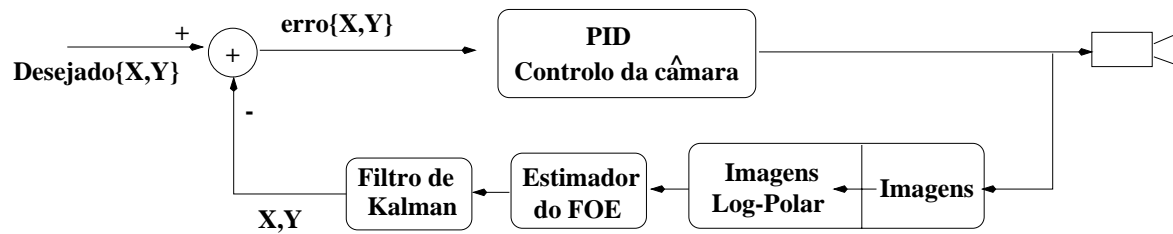


Figura 4.14: Controlador utilizado para o controlo do alinhamento da câmara. Note-se que algumas acções do controlador são efectuadas no estado $\mathbf{0}$, enquanto outras no estado $\mathbf{1}$.

foi necessário colocar desenhos texturados no chão, de modo a existir textura para uma melhor estimação do fluxo óptico e consequentemente uma melhor estimativa da posição do *FOE*.

O controlo subjacente a este processo passa pela tentativa de estabilizar a posição do *FOE* no centro da imagem. Definindo a função erro como a diferença das coordenadas relativamente ao centro da imagem, e utilizando um *PID*, pode-se assim controlar os movimentos de “pan” e “tilt” da câmara. O sistema experimental encontra-se representado na Fig 5.13, onde os espelhos são usados para a navegação e os Pan & Tilt para as experiências de alinhamento da câmara usando o *FOE*.

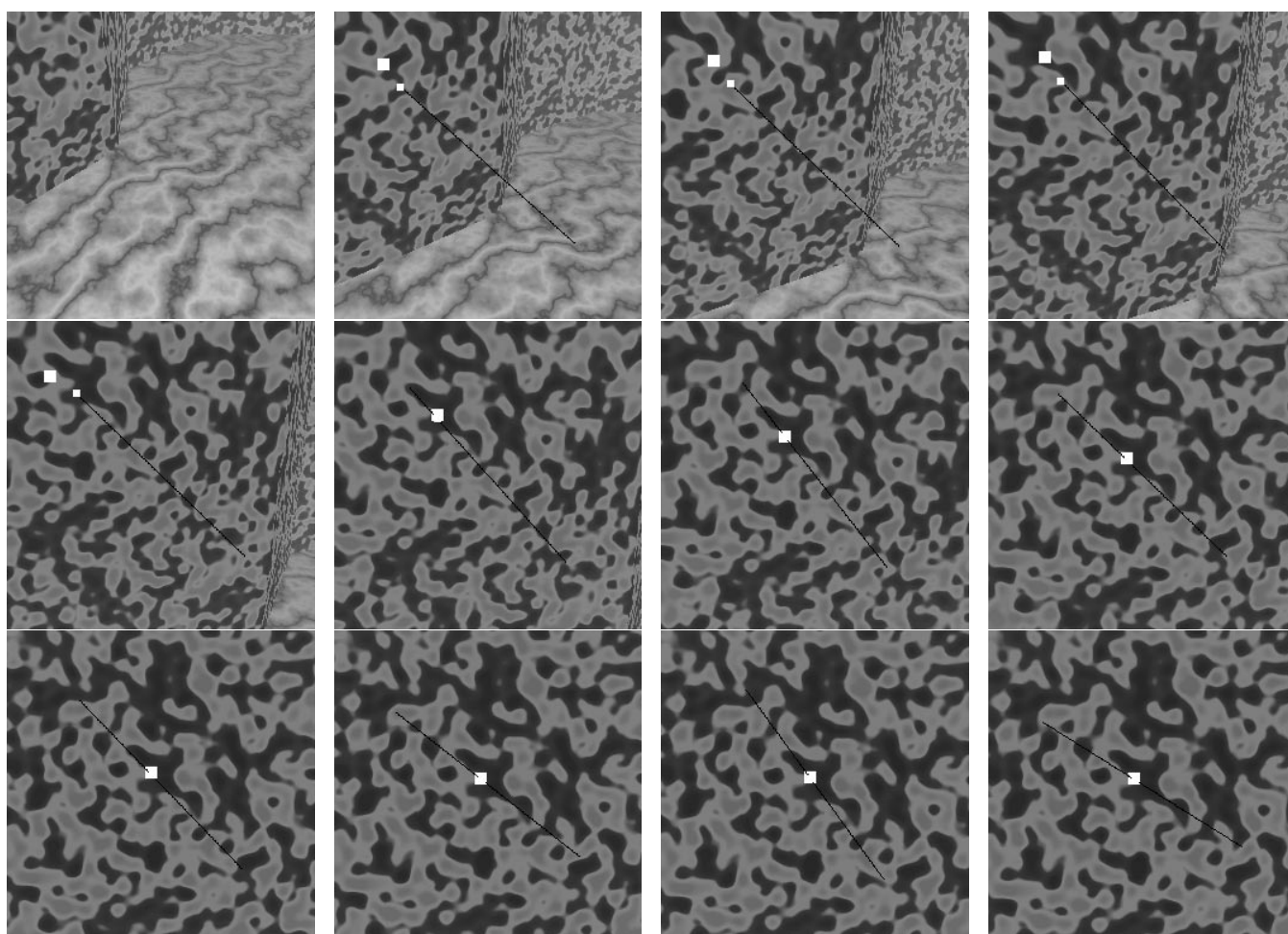


Figura 4.15: Sequência de alinhamento de uma câmara com o vector de movimento translacional de um robot, na iteração 0, 11, 17, 21, 25, 29, 33, 37, 41, 45, 49 e 53. O ponto a branco indica a posição do *FOE* detectado na imagem log-polar que foi depois reconvertido para a imagem cartesiana.



Figura 4.16: Sequência de alinhamento de uma câmara com o vector de movimento translacional de um robot. O ponto a preto indica a posição do *FOE* detectado na imagem log-polar que foi depois reconvertido para a imagem cartesiana.

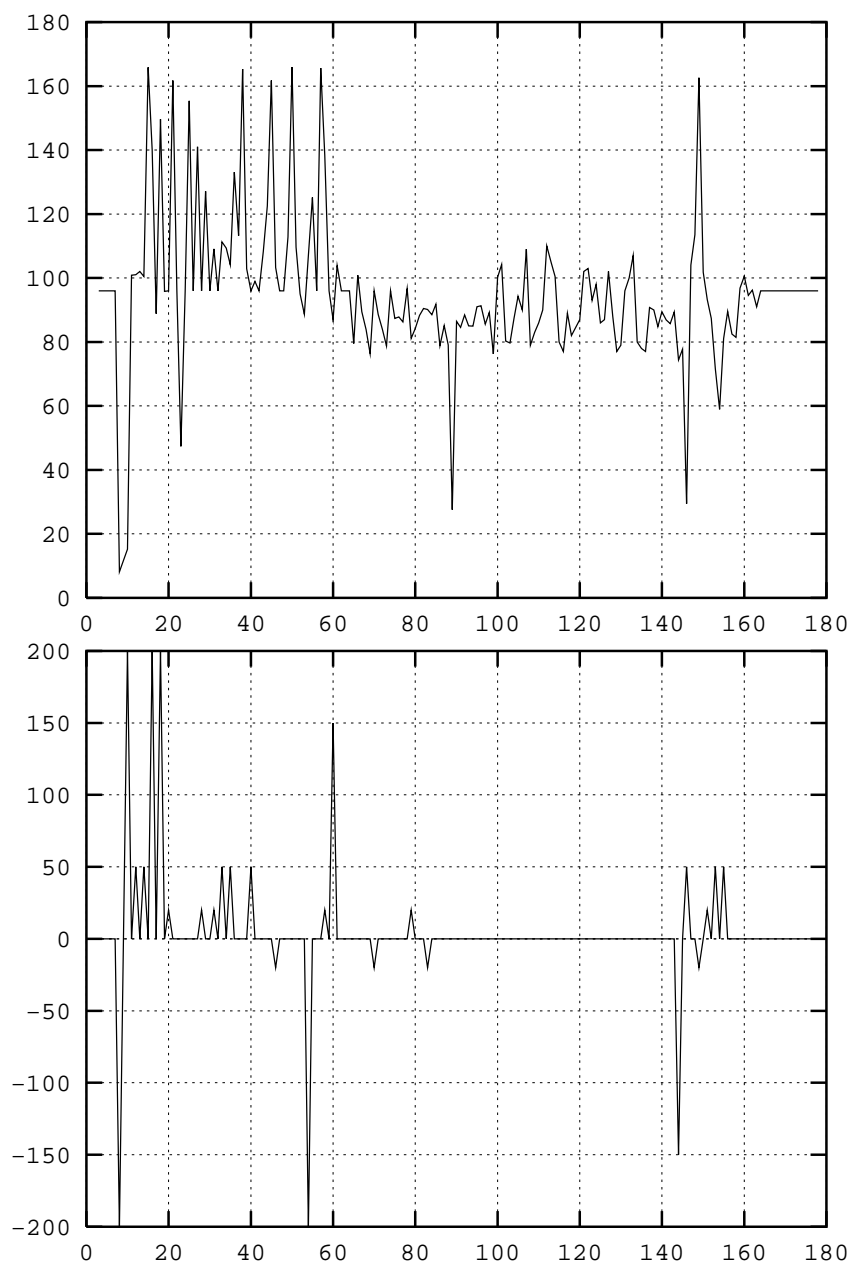


Figura 4.17: Em cima encontra-se representada a coordenada x do *FOE*, enquanto em baixo temos os comandos em número de passos enviados para o motor de “Pan” do sistema. Estes gráficos foram obtidos para o exemplo descrito na Fig. 4.16.

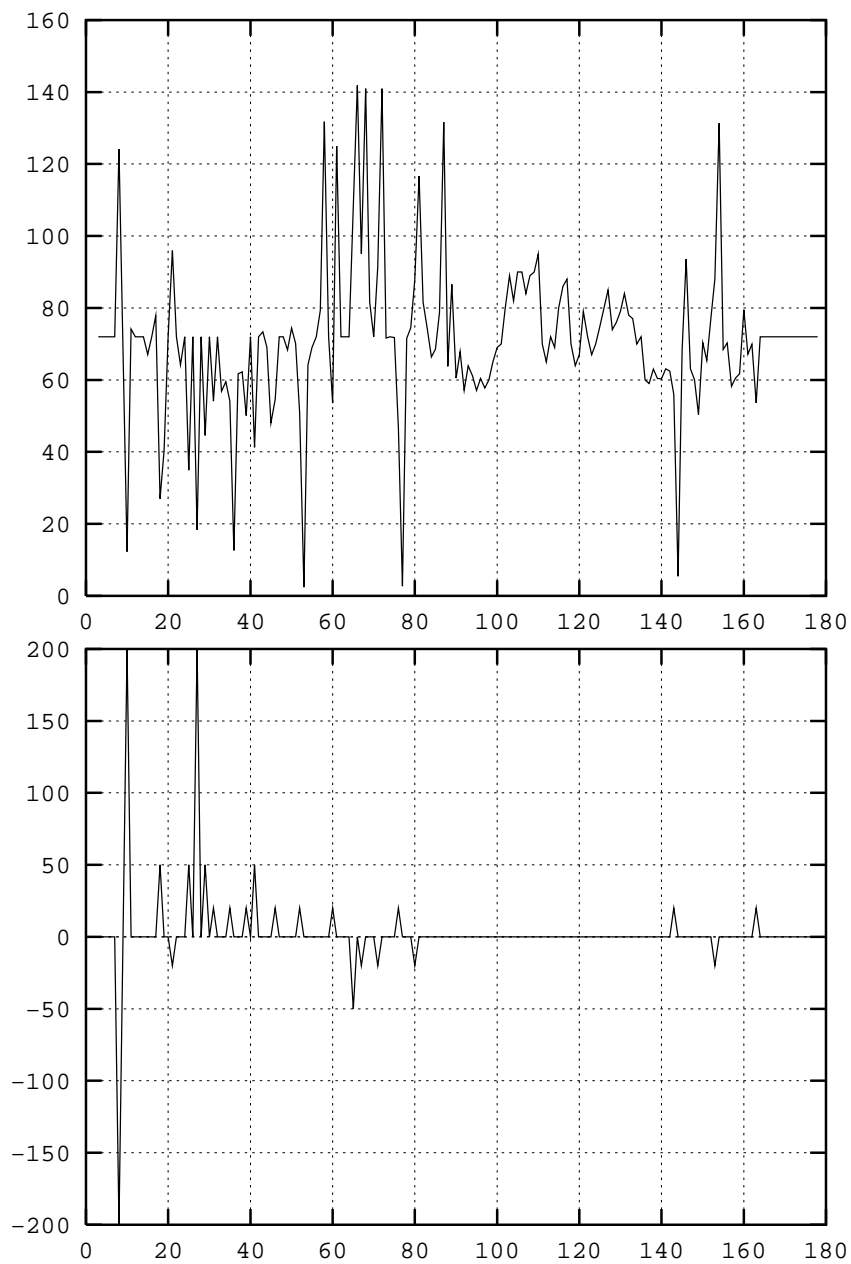


Figura 4.18: Em cima encontra-se representada a coordenada y do FOE , enquanto em baixo temos os comandos em número de passos enviados para o motor de “Tilt” do sistema. Estes gráficos foram obtidos para o exemplo descrito na Fig. 4.16.

Capítulo 5

Navegação utilizando visão artificial

5.1 Introdução

Neste capítulo explora-se a visão por computador para conduzir um robot móvel através dos corredores de ligação de salas de um ambiente estruturado, tal como encontramos em interiores de edifícios. O protótipo tem como objectivo evitar qualquer embate com as superfícies das paredes do corredor. Para levar a cabo tal tarefa é proposta uma solução com base no equilíbrio de fluxo óptico de câmaras divergentes.

Numa primeira abordagem para solucionar o problema foi utilizado o fluxo óptico medido em imagens obtidas por duas câmaras para controlar a plataforma móvel. Cada uma das câmaras adquire imagens de uma das paredes. Com base na diferença dos fluxos calculados em cada uma das imagens, poder-se-á detectar se o robot se encontra mais afastado ou mais perto duma das paredes. Desta forma, prova-se que é possível controlar os movimentos de rotação e translação do robot móvel.

Uma extensão a esta abordagem é proposta, baseando-se na utilização das duas câmaras, mas agora associando a cada câmara dois espelhos que permitem simular quatro câmaras virtuais (duas câmaras virtuais por cada câmara real). Cada um destes espelhos tem por função fornecer uma imagem da superfície da mesma parede do corredor. De seguida, através de sinais de controlo sintetizados de cada uma das duas imagens da mesma parede, pode-se estimar de uma forma muito simples um sinal proporcional à orientação

que a superfície da parede faz com o robot móvel. A única restrição à solução, advém do facto da alteração da orientação das paredes do corredor necessitar de manter uma certa continuidade. Esta imposição tem a ver com o facto das imagens obtidas pelos dois espelhos terem um deslocamento espacial horizontal. Por este facto, os locais onde se dão as transições das paredes com diferentes orientações são os pontos mais delicados, uma vez que um espelho fornece imagens de uma zona com uma determinada orientação, enquanto as imagens obtidas pelo outro apresentam outra orientação distinta. Desta forma o sistema desenvolvido é incapaz de navegar em locais onde a orientação das paredes apresentem descontinuidades muito altas.

Na próxima secção vamos analisar o problema do posicionamento das câmaras em relação ao centro dum referencial de locomoção. Para tal é introduzido o conceito de esfera de imagem virtual, cujo centro coincide com o referencial de locomoção. Pretende-se estudar a forma de posicionar câmaras convencionais na superfície dessa esfera, de modo a obter amostras do sinal captado nessa esfera de imagens. Inicialmente o estudo é feito a nível de um equador da esfera, utilizando câmaras divergentes e de seguida estuda-se o efeito da mudança do ângulo de vergência das mesmas câmaras.

Finalmente são descritas as duas abordagens indicadas anteriormente e de seguida são apresentados os resultados finais.

5.2 Motivação

Como é do conhecimento geral, a utilização do fluxo óptico como medida para um controlo de uma plataforma móvel apresenta alguns problemas, na medida em que por exemplo, o fluxo óptico induzido pelo movimento do observador tem de ser separado do fluxo gerado por um objecto que se movimenta livremente no meio ambiente. Trata-se aqui de um problema de segmentação (separação de fluxos nas imagens). Uma forma eficaz, de resolver o problema, pode ser obtida através de correlação, no entanto este processo envolve um elevado custo computacional. Na solução apresentada pressupõe-se que o universo em que o observador se movimenta é estático, sendo o fluxo óptico gerado através do movimento do observador móvel, utilizado como medida para recuperar alguma informação do meio

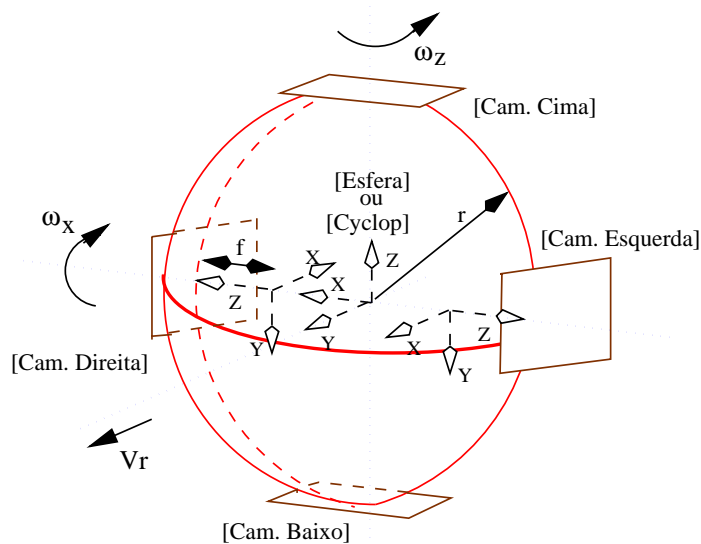


Figura 5.1: Modelo para um sistema de navegação num mundo tridimensional.

ambiente.

5.2.1 Sensores de imagens esféricos

Para fundamentar as restantes secções deste capítulo, suponha que se pretende controlar a atitude dum veículo no espaço tridimensional relativamente ao meio ambiente, à medida que este se movimenta. Considerando o sistema de referenciais representado na Fig. 5.1 e modelando o sensor visual por uma superfície esférica com um raio r , apenas é necessário controlar as três velocidades indicadas na figura (não considerando o grau de liberdade de “swing” do veículo) para atingir qualquer ponto do espaço. Bom, mas na realidade não temos um sensor esférico, contudo podemos simular um, caso se utilizem câmaras convencionais cujas posições obedeçam à geometria do hipotético sensor esférico. Desta forma, a pergunta que se põe à partida, é quantas câmaras se devem utilizar e onde se devem colocar na superfície esférica para adquirir a informação necessária ao controlo do veículo. Segundo [Nelson, R.C., Aloimonos, J. 87] e [F. Bergholm 90], o fluxo medido ao longo da superfície da esfera que apresenta maior relevância para o efeito, é o fluxo

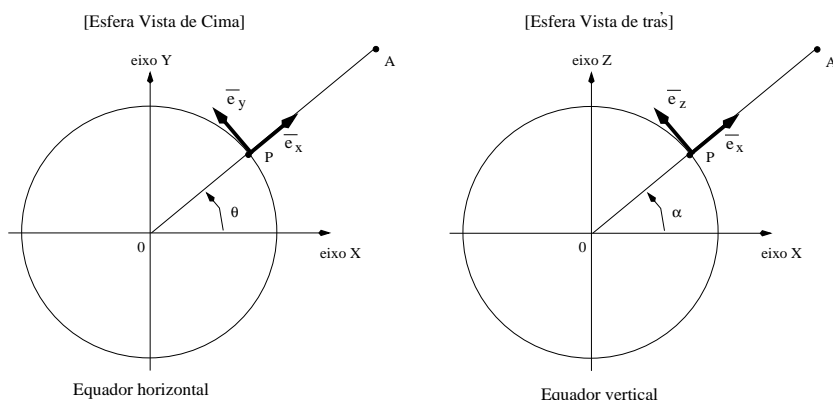


Figura 5.2: Decomposição do vector velocidade $\dot{\vec{p}}$ nos referenciais descritos pelos versores $\vec{e}_x, \vec{e}_y, \vec{e}_z$. À esquerda, corte da superfície esférica segundo um plano contendo os eixos x e y . À direita, corte da superfície esférica segundo um plano contendo os eixos x e z

medido segundo o equador e perpendicular à direcção de movimento do veículo. Vamos então de seguida efectuar a análise do fluxo projectado na esfera.

Considere-se uma esfera de raio unitário em movimento com velocidade expressa por $(\vec{v}, \vec{\omega})$ com $\vec{v}, \vec{\omega} \in \mathfrak{R}^3$. Seja d a distância do ponto central de projecção da esfera a um ponto tridimensional A com vector \vec{x} . Então $d = \sqrt{x^2 + y^2 + z^2}$ e a seguinte relação é válida:

$$\dot{\vec{x}} = \vec{\omega} \times \vec{x} + (\vec{v}/d) \cdot \|\vec{x}\| \quad (5.1)$$

Verifica-se que qualquer ponto P ao longo da linha de projecção do ponto A pode ser descrito através de $\vec{p} = \vec{x}\mu, \mu > 0$ e $\mu \in \mathfrak{R}$, em particular se P pertencer à superfície da esfera tem módulo unitário o que implica $\mu = \|\vec{x}\|^{-1}$. A velocidade do ponto A projectada em P é assim dada por:

$$\dot{\vec{p}} = \vec{\omega} \times \vec{p} + (\vec{v}/d) \quad (5.2)$$

Este vector de velocidade $\dot{\vec{p}}$ pode ser decomposto em três componentes descritas em qualquer sistema de coordenadas ortogonal adequado para o efeito. Como neste momento interessa estudar o comportamento do fluxo ao longo do equador horizontal da esfera (exemplo esquerdo da Fig. 5.2), vamos descrever o vector velocidade segundo um referencial posicionado nesse equador da esfera, onde o versor \vec{e}_x tem o mesmo sentido da direcção radial do ponto A considerado (veja-se Fig. 5.1 e 5.2). Desta forma o ponto

P na superfície da esfera e correspondente à projecção de A na sua superfície, pode ser definido pelas coordenadas $(\cos(\theta), \sin(\theta), 0)$. Os versores do referencial são dados por $\vec{e}_y = [\sin(\theta), \cos(\theta), 0]^T$, $\vec{e}_z = [0, 0, 1]^T$ e o versor \vec{e}_x não tem interesse uma vez que a velocidade segundo esta direcção não pode ser estimada na superfície da esfera. Assim a partir de (5.2) temos:

$$\begin{cases} \vec{e}_y \cdot \dot{\vec{p}} = \omega_z + \cos(\theta)v_y/d \Leftrightarrow \sin(\theta)v_x/d & \rightarrow \text{fluxo paralelo ao equador horizontal} \\ \vec{e}_z \cdot \dot{\vec{p}} = \omega_x \sin(\theta) \Leftrightarrow \omega_y \cos(\theta) + v_z/d & \rightarrow \text{fluxo normal ao equador horizontal} \end{cases} \quad (5.3)$$

No nosso caso as velocidades da esfera são dadas por $\vec{v} = [0, v_y, 0]^T$ ou $\vec{v} = [0, v_r, 0]^T$ e $\vec{\omega} = [\omega_x, 0, \omega_z]^T$ pelo que as equações anteriores tomam a seguinte forma:

$$\begin{cases} \vec{e}_y \cdot \dot{\vec{p}} = \omega_z + \cos(\theta)v_y/d & \rightarrow \text{fluxo paralelo ao equador horizontal} \\ \vec{e}_z \cdot \dot{\vec{p}} = \omega_x \sin(\theta) & \rightarrow \text{fluxo normal ao equador horizontal} \end{cases} \quad (5.4)$$

Pelas equações (5.4) concluímos que o fluxo normal ao equador permite estimar a velocidade angular ω_x uma vez que o fluxo não depende da profundidade d . Outra conclusão importante a tirar é que o fluxo paralelo ao equador tem um comportamento sinusoidal ao longo desse mesmo equador (note-se que é modulado pela profundidade do respectivo ponto). Também é importante salientar que o fluxo paralelo ao equador após a remoção do factor ω_z , permite estimar sinais inversamente proporcionais à profundidade em cada ponto visualizado na superfície da esfera.

O estudo realizado para o equador horizontal pode ser feito para a mesma circunferência contida no plano $Y = 0$ (exemplo direito da Fig. 5.2, equador vertical) e após se efectuar a decomposição da velocidade $\dot{\vec{p}}$ segundo as direcções dos versores \vec{e}_y e \vec{e}_z , a partir de (5.2) obtemos:

$$\begin{cases} \vec{e}_y \cdot \dot{\vec{p}} = \omega_z \cdot \cos(\alpha) \Leftrightarrow \omega_x \cdot \sin(\alpha) + v_y/d & \rightarrow \text{fluxo normal ao equador vertical} \\ \vec{e}_z \cdot \dot{\vec{p}} = 0 & \rightarrow \text{fluxo paralelo ao equador vertical} \end{cases} \quad (5.5)$$

Após toda esta análise, a pergunta que se segue é, como se pode controlar o sistema, usando as velocidades medidas na superfície esférica? Bom, já se referiu anteriormente da possibilidade da velocidade medida na esfera ser inversamente proporcional à profundi-

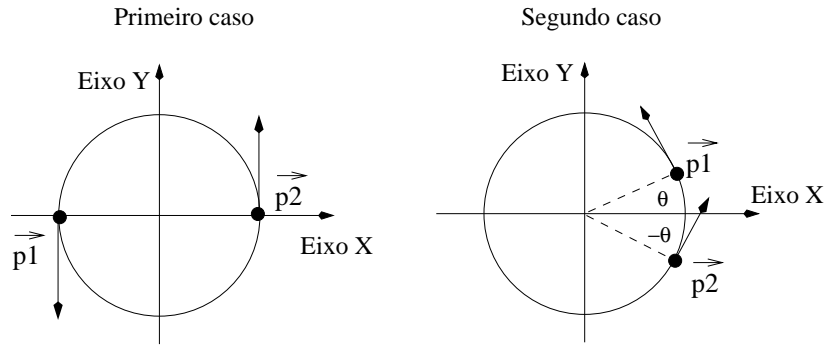


Figura 5.3: Representação dos pontos relevantes para efectuar o controlo da velocidade ω_z .

de, pelo que não será de estranhar uma possível comparação entre as velocidades medidas em diferentes pontos da esfera, desde que se remova o efeito da velocidade de rotação. Por conseguinte, é perfeitamente natural que se comparem dois pontos laterais divergentes, de modo a nos permitir controlar as velocidades ω_z e ω_x .

No entanto, por simplicidade, para já considere-se apenas o controlo da velocidade ω_z . Analisando a Fig. 5.3, para o primeiro caso, onde o fluxo $\vec{e}_y \cdot \dot{\vec{p}}$ paralelo ao equador é utilizado (equação (5.4)), temos:

- $\vec{e}_y \cdot \dot{\vec{p}}_2 + \vec{e}_y \cdot \dot{\vec{p}}_1 = 2\omega_z + v_y \left(\frac{1}{d_2} \Leftrightarrow \frac{1}{d_1} \right)$ que depende da velocidade ω_z
- $\vec{e}_y \cdot \dot{\vec{p}}_2 \Leftrightarrow \vec{e}_y \cdot \dot{\vec{p}}_1 = v_y \left(\frac{1}{d_2} + \frac{1}{d_1} \right)$ onde é impossível estimar a diferença de profundidade entre os dois pontos

Estes resultados à partida indiciam uma dependência da velocidade ω_z , que por sua vez é prejudicial ao controlo do sistema.

Verificando agora o segundo caso da mesma figura, e utilizando os dois pontos de vista da mesma posição lateral, obtemos:

- $\vec{e}_y \cdot \dot{\vec{p}}_2 + \vec{e}_y \cdot \dot{\vec{p}}_1 = 2\omega_z + \cos(\theta) v_y \left(\frac{1}{d_2} + \frac{1}{d_1} \right)$ - que depende da velocidade ω_z
- $\vec{e}_y \cdot \dot{\vec{p}}_2 \Leftrightarrow \vec{e}_y \cdot \dot{\vec{p}}_1 = \cos(\theta) v_y \left(\frac{1}{d_2} \Leftrightarrow \frac{1}{d_1} \right)$, é possível comparar as profundidades relativas entre os dois pontos, mesmo na presença das velocidades de rotação ω_z e ω_x

Destas duas análises, a conclusão a retirar é a seguinte: para controlar a velocidade ω_z é necessário utilizar quatro câmaras, duas para cada ponto de vista lateral, caso se pretenda remover a influência “maléfica” da velocidade ω_z . Note-se que o controlo pode ser efectuado de forma independente, i.e., obteve-se uma medida proporcional à orientação da linha definida pelos dois pontos visualizados, com o referencial de locomoção.

Quanto ao controlo da velocidade ω_x apenas são necessárias duas câmaras posicionadas da forma representada na Fig. 5.1 através de [cam. Cima] e [cam. Baixo], porque o valor de ω_x pode ser estimado através do fluxo obtido nas câmaras laterais, tal como foi explicado anteriormente. Convém, no entanto verificar que de facto através do uso do fluxo $\vec{e}_y \cdot \vec{p}$ normal ao equador vertical dado pela eq. (5.5), a diferença entre o fluxo medido nos pontos de cima e de baixo da esfera permite comparar as profundidades relativas desses mesmos pontos. Após efectuar esse cálculo temos:

$$\Leftrightarrow 2\omega_x + v_y \left(\frac{1}{d_2} \Leftrightarrow \frac{1}{d_1} \right) \quad (5.6)$$

onde ω_x pode então ser removido através da estimação obtida pelo fluxo normal ao equador horizontal $\vec{e}_z \cdot \vec{p}$ dado pela equação 5.4, onde o termo restante verifica de facto tal critério. Em resumo, no total são necessárias seis câmaras, quatro para controlar a velocidade ω_z e duas para a velocidade ω_x .

Note-se no entanto, que se optar-se por medir as velocidades na superfície da esfera só no momento em que o sistema se movimenta com velocidade linear, então apenas é necessário utilizar quatro câmaras no total para efectuar o controlo do sistema. Estas câmaras devem estar posicionadas como se encontra representado na Fig. 5.1. Com as câmaras direita e esquerda o sistema pode controlar a velocidade ω_z e com as outras duas a velocidade ω_x . Contudo apenas com estas quatro câmaras é impossível controlar o sistema, enquanto se efectua movimentos de rotação. Uma das formas de resolver este problema é através do uso de mais duas câmaras horizontais, tal como foi referido.

Contudo, existe nesta análise simplificações que convém não levar em conta, uma vez que na realidade não temos a possibilidade de garantir que o centro do sensor esférico se encontra coincidente com o referencial de locomoção. Para isso vamos numa primeira fase considerar um pequeno deslocamento. De seguida analisa-se mais em pormenor esta situação.

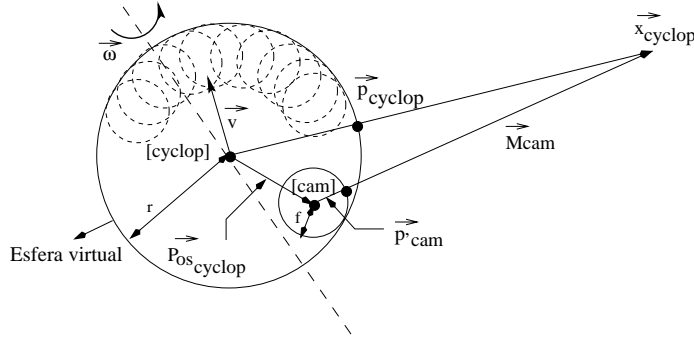


Figura 5.4: Pequenas superfícies da esfera virtual de raio r obtidas através de sensores esféricos reduzidos.

Simulação de uma esfera virtual através de pequenos sensores esféricos

Dada a impossibilidade referida, podemos tentar construir o sensor esférico através do uso de pequenos sensores esféricos, dado que estes são mais fáceis de posicionar. Para simular o sensor esférico virtual a partir de pequenos sensores esféricos é necessário relacionar as velocidades medidas em cada um dos sensores.

Considere-se então um pequeno sensor esférico com raio f representado na Fig. 5.4 e centrado num referencial $[cam]$ que permite simular parte da superfície esférica de raio r que neste momento é considerada como um sensor esférico virtual (esfera no referencial $[cyclop]$). Comparando as relações entre as velocidades medidas na esfera pequena (de raio f) e as medidas na esfera virtual (de raio r) para o mesmo ponto tridimensional, temos:

$$\dot{\vec{p}}_{cyclop} = \frac{\dot{\vec{x}}_{cyclop}}{\|\vec{x}_{cyclop}\|} r \quad (5.7)$$

sendo $\dot{\vec{x}}_{cyclop} = \vec{\omega} \times \vec{x}_{cyclop} + \vec{v}$. Nesta situação verificam-se as seguintes relações:

$$\begin{cases} \vec{x}_{cyclop} = \vec{P}os_{cyclop} + {}^{cyclop}R_{cam} \vec{M}_{cam} \\ \|\vec{M}_{cam}\| = \|\vec{x}_{cyclop} - \vec{P}os_{cyclop}\| \\ \dot{\vec{x}}_{cyclop} = \vec{\omega} \times (\vec{P}os_{cyclop} + {}^{cyclop}R_{cam} \vec{M}_{cam}) + \vec{v} \end{cases} \quad (5.8)$$

onde ${}^{cyclop}R_{cam}$ é uma matriz de transformação que representa a orientação relativa dos referenciais associados a cada uma das esferas. A velocidade do mesmo ponto medida na

esfera pequena é:

$$\dot{p}'_{cam} = \frac{\dot{\vec{M}}_{cam}}{\|\dot{\vec{M}}_{cam}\|} f \quad (5.9)$$

Dado que a relação entre $\dot{\vec{M}}_{cam}$ e $\dot{\vec{x}}_{cyclop}$ é dada por:

$${}^{cyclop}R_{cam}\dot{\vec{M}}_{cam} = \dot{\vec{x}}_{cyclop} \quad (5.10)$$

expressando a velocidade \dot{p}'_{cam} em função da velocidade $\dot{\vec{x}}_{cyclop}$ obtém-se:

$$\dot{p}'_{cam} = [{}^{cyclop}R_{cam}]^{-1} \frac{\dot{\vec{x}}_{cyclop}}{\|\dot{\vec{M}}_{cam}\|} f \quad (5.11)$$

no entanto, descrevendo esta velocidade no referencial $[cyclop]$, obtém-se finalmente a relação pretendida:

$$\dot{p}'_{cyclop} = {}^{cyclop}R_{cam}\dot{p}'_{cam} = \dot{p}'_{cyclop} \frac{\|\vec{x}_{cyclop}\|}{\|\vec{x}_{cyclop} \Leftrightarrow \vec{P}os_{cyclop}\|} \frac{f}{r} \quad (5.12)$$

Comparando neste momento as equações (5.7) e (5.12) as principais diferenças encontram-se no factor $\frac{f}{r}$ que se pode considerar à partida conhecido, e o factor desconhecido $G = \frac{\|\vec{x}_{cyclop}\|}{\|\vec{x}_{cyclop} - \vec{P}os_{cyclop}\|}$ que representa a relação entre a profundidade do ponto tridimensional medida na esfera imaginária e a profundidade medida na esfera pequena. Contudo repare-se que estas velocidades são velocidades tridimensionais e portanto nas respectivas esferas apenas são medidas as velocidades no plano tangente à esfera, o que corresponde a retirar a componente radial, respectivamente a componente na direcção \vec{p}'_{cyclop} e \vec{p}'_{cam} . Desta forma as duas velocidade estimadas em cada um dos sensores nunca serão iguais. Este facto tem a ver com a decomposição da velocidade ser efectuada segundo referenciais com diferentes orientações, onde não é possível estimar as componentes radiais das velocidades. Existe ainda o problema da projecção na esfera pequena variar com a variação da profundidade do mesmo ponto. Como é óbvio tal situação não se passa no caso da esfera virtual.

Contudo, estes problemas são facilmente solucionados se os vectores \vec{p}'_{cam} e \vec{p}'_{cyclop} forem colineares. Nesta situação $\vec{P}os_{cyclop} = \frac{\vec{x}_{cyclop}}{\|\vec{x}_{cyclop}\|}(r \Leftrightarrow f)$ e $G = \frac{\|\vec{x}_{cyclop}\|}{\|\vec{x}_{cyclop}\| - (r-f)} \approx 1$ para $\|\vec{x}_{cyclop}\| \gg r$.

A título de exemplo, se $r = 15cm$, $f = 25mm$ e $\|\vec{x}_{cyclop}\| \gg 1,5m$ este factor será $G = \frac{12}{11} \approx 1,0(90)$ e na presença de velocidades baixas pode ser desprezado.

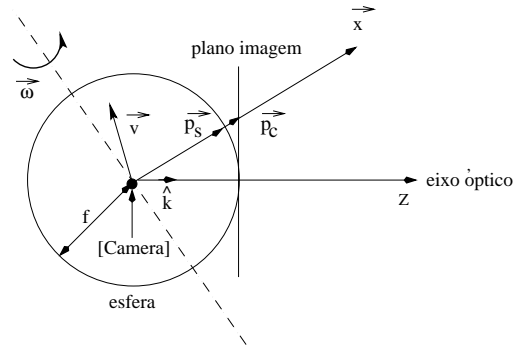


Figura 5.5: Representação das diferenças entre o pequeno sensor esférico e uma câmara convencional.

Simulação do sensor esférico através de câmaras convencionais

Como na realidade os pequenos sensores esféricos vão ser simulados através de câmaras convencionais é importante estudar as diferenças entre ambos.

Considerando a representação da Fig. 5.5, a velocidade $\dot{\vec{x}}$ “sentida” no ponto \vec{p}_s da esfera é $\dot{\vec{p}}_s = \frac{\dot{\vec{x}}}{\|\vec{x}\|} f$, mas projectada na esfera apenas se pode estimar a velocidade $\dot{\vec{p}}_p = \dot{\vec{p}}_s \Leftrightarrow (\dot{\vec{p}}_s \cdot \hat{x}) \hat{x} = \frac{\dot{\vec{x}} \cdot \hat{x}}{\|\vec{x}\|} f \hat{x} \Leftrightarrow \frac{\dot{\vec{x}} \cdot \hat{x}}{\|\vec{x}\|} f \hat{x}$ onde $(\frac{\dot{\vec{x}} \cdot \hat{x}}{\|\vec{x}\|} f \hat{x})$ é a componente radial. Para o caso da câmara, diferenciando a lei de projecção geométrica $\vec{p}_c = \frac{f}{\vec{x} \cdot \hat{k}} \vec{x}$ em ordem a t , pode-se expressar $\dot{\vec{p}}_c = \frac{f}{\vec{x} \cdot \hat{k}} \dot{\vec{x}} \Leftrightarrow \frac{f}{(\vec{x} \cdot \hat{k})^2} (\dot{\vec{x}} \cdot \hat{k}) \vec{x}$, que infelizmente não coincide com a velocidade $\dot{\vec{p}}_p$ estimada na superfície da esfera. Contudo se o vector \vec{x} e o versor \hat{k} forem colineares as duas velocidades são iguais. Desta forma, apenas se pode utilizar a zona da imagem onde o sensor é tangente à superfície esférica, porque só nesse situação ambas as velocidades podem ser aproximadas.

5.2.2 Caso especial da utilização de um sistema de visão activa

Apesar das análises efectuadas anteriormente, não foi considerada uma situação, que se prende com o facto das câmaras não se posicionarem de uma forma tangente ao sensor esférico virtual. Um dos exemplos que não obedece a essa situação, é um sistema de visão activa. Portanto é de alguma forma relevante que se efectue um estudo mais detalhado, para esta possibilidade, dado o facto de no nosso laboratório existirem equipamentos que obedecem

a essas especificações. Note-se que para esta situação, o centro de projecção das câmaras não se situa no centro da esfera virtual e as câmaras não são tangentes à superfície da mesma esfera.

Portanto convém agora utilizar o modelo das câmaras convencionais. Para estimar a informação medida nas imagens, é necessário determinar o fluxo induzido nas imagens devido ao movimento do veículo. Para isso, admitindo que todas as câmaras têm a mesma distância focal f , a relação de rotação entre o referencial da câmara direita e esquerda relativamente ao referencial $[Cyclop]$ é respectivamente dada por R_{ri} , R_{li} :

$$R_{ri} = \begin{bmatrix} 0 & 0 & 1 \\ \Leftrightarrow 1 & 0 & 0 \\ 0 & \Leftrightarrow 1 & 0 \end{bmatrix} \quad R_{li} = \begin{bmatrix} 0 & 0 & \Leftrightarrow 1 \\ 1 & 0 & 0 \\ 0 & \Leftrightarrow 1 & 0 \end{bmatrix} \quad (5.13)$$

A translação dos referenciais pode ser expressa para a câmara direita e esquerda respectivamente pelos vectores $\mathbf{O}\vec{\mathbf{F}}_r = [r \Leftrightarrow f, 0, 0]^T$ e $\mathbf{O}\vec{\mathbf{F}}_l = [\Leftrightarrow r + f, 0, 0]^T$. As velocidades do veículo são dadas pelos vectores $\vec{v} = [0, v_r, 0]^T$ e $\vec{\omega} = [\omega_x, 0, \omega_z]^T$, pelo que as velocidades associadas a um ponto tridimensional $\vec{\mathbf{p}}_i = [X_i, Y_i, Z_i]^T$ (onde o índice i indica câmara direita ou esquerda quando substituído respectivamente por $i = r$, $i = l$), são neste caso dadas por (veja-se apêndice C):

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix}_r = R_{ri}^{-1} [\Leftrightarrow \vec{v} \Leftrightarrow \vec{\omega} \times (R_{ri}\vec{\mathbf{p}}_r + \mathbf{O}\vec{\mathbf{F}}_r)] \quad \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix}_l = R_{li}^{-1} [\Leftrightarrow \vec{v} \Leftrightarrow \vec{\omega} \times (R_{li}\vec{\mathbf{p}}_l + \mathbf{O}\vec{\mathbf{F}}_l)] \quad (5.14)$$

Substituindo este resultado na equação 2.27 obter-se-ia o fluxo induzido em ambas as imagens, i.e., $\dot{u}_l(u, v)$, $\dot{v}_l(u, v)$, $\dot{u}_r(u, v)$ e $\dot{v}_r(u, v)$. Como se pretende comparar as profundidades dos pontos visualizados, convém que esses pontos sejam pontos correspondentes, logo pertencentes a uma mesma recta perpendicular a ambos os planos imagem. Este facto significa que os fluxos comparados devem ser $\dot{u}_r(u, v)$, $\dot{v}_r(u, v)$, $\dot{u}_l(\Leftrightarrow u, v)$ e $\dot{v}_l(\Leftrightarrow u, v)$ (veja-se Fig. 5.6). Somando os respectivos fluxos \dot{u} obtém-se:

$$\dot{u}_r(u, v) + \dot{u}_l(\Leftrightarrow u, v) = 2\omega_z \left(f_x + \frac{u^2}{f_x} \right) + \frac{f_x v_r (Z_l \Leftrightarrow Z_r)}{Z_r Z_l} + \frac{f_x \omega_z (Z_l + Z_r) (r \Leftrightarrow f)}{Z_r Z_l} \quad (5.15)$$

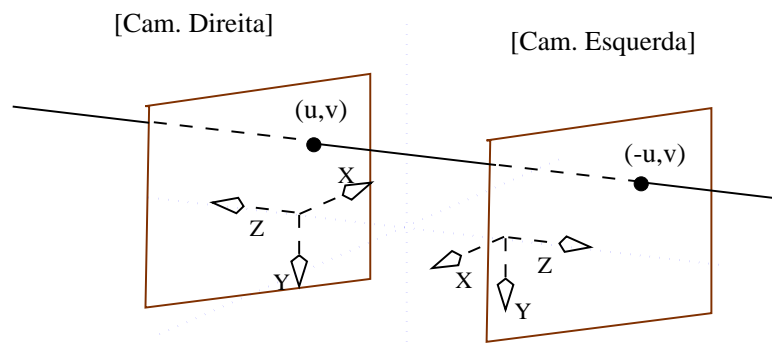


Figura 5.6: Os fluxos comparados $\dot{u}_r(u, v)$, $\dot{v}_r(u, v)$, $\dot{u}_l(\Leftrightarrow u, v)$ e $\dot{v}_l(\Leftrightarrow u, v)$.

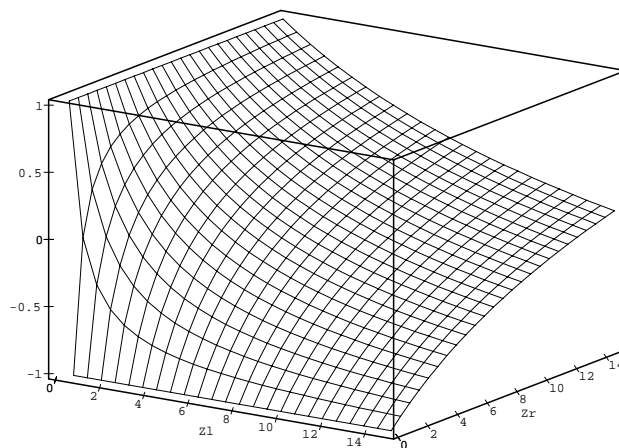


Figura 5.7: Medida da distância efectuada pelo sistema, neste caso $\Leftrightarrow \frac{z_l - z_r}{z_l + z_r}$. A vantagem desta medida deve-se ao facto de ser independente das velocidades do sistema, contrariamente à apresentada na Fig. 5.8.

Subtraindo os fluxos \dot{u} obtém-se:

$$\dot{u}_r(u, v) \Leftrightarrow \dot{u}_l(\Leftrightarrow u, v) = 2 \frac{\omega_x v f_x}{f_y} + \frac{f_x v_r (Z_l + Z_r)}{Z_r Z_l} + \frac{f_x \omega_z (Z_l \Leftrightarrow Z_r) (r \Leftrightarrow f)}{Z_r Z_l} \quad (5.16)$$

Somando os respectivos fluxos \dot{v} obtém-se:

$$\dot{v}_r(u, v) + \dot{v}_l(\Leftrightarrow u, v) = \Leftrightarrow 2 \frac{\omega_x u f_y}{f_x} \quad (5.17)$$

Subtraindo os fluxos \dot{v} obtém-se:

$$\dot{v}_r(u, v) \Leftrightarrow \dot{v}_l(\Leftrightarrow u, v) = 2 \frac{\omega_z u v}{f_x} \quad (5.18)$$

Dados os resultados apresentados nas equações 5.15, 5.16, 5.17 e 5.18, as seguintes conclusões podem ser tomadas:

- as câmaras devem ter parâmetros f_x , f_y idênticos, caso contrário os resultados obtidos não permitem tirar as conclusões que se seguem, uma vez que as equações 5.15 a 5.18 irão possuir termos que dependem da profundidade em cada pixel.
- as equações 5.17 e 5.18 permitem estimar as velocidades de rotação ω_x e ω_z , já que os termos presentes nas equações não dependem da profundidade, no entanto para efectuar a estimação é necessário calibrar as câmaras.
- o primeiro termo presente nas equações 5.15, 5.16 representa o efeito das rotações, e é um factor prejudicial, no entanto pode ser anulado se a estimação de ω_x e ω_z for correcta.
- o terceiro termo presente nas equações 5.15, 5.16 é devido às câmaras não se encontrarem posicionadas no eixo de rotação, e o seu efeito é tanto maior quanto maior a distância ao centro ($r \Leftrightarrow f$). No caso de esta distância ser nula, o terceiro termo desaparece. Os termos dependentes da rotação representam uma quantidade indesejada em ambas as equações.
- no caso de se optar por um controlo em que se estimam estas quantidades apenas na presença da translação (quando se efectuam rotações com o veículo a medida de

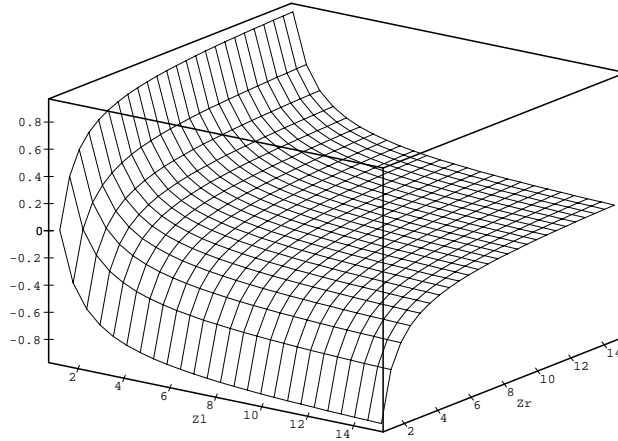


Figura 5.8: Medida da distância efectuada pelo sistema, neste caso $\propto \frac{Z_l - Z_r}{Z_l \cdot Z_r}$. Esta medida é preferível à representada na Fig. 5.7, devido a zona mais planar em locais onde Z_l e Z_r são aproximadamente iguais, e nos outros possui um crescimento exponencial. No entanto o pequeno senão, é o facto da medida ser proporcional a $f_x v_r$.

sinais de erro para o controlo são desprezados), então dividindo (5.15) por (5.16) obtém-se uma estimativa da profundidade independente dos parâmetros das câmaras e das velocidades, i.e., estaremos a medir $\frac{Z_l - Z_r}{Z_l + Z_r}$ (este resultado é importante, pois pode ser medido sem ser efectuada nenhuma calibração dos parâmetros intrínsecos das câmaras, a menos do seu posicionamento).

Por outro lado, se se utilizar apenas o segundo termo da equação (5.15), estaremos a medir uma quantidade proporcional a $\frac{Z_l - Z_r}{Z_l \cdot Z_r}$, no entanto a constante de proporcionalidade depende de $f_x v_r$. Esta medida é preferível se a velocidade v_r de translação for constante, uma vez que a medida pode variar com v_r . O facto desta preferência pode ser constatado através das Fig. 5.8 e 5.7. Na eventualidade de se optar por um controlo deste género, note-se que o facto das câmaras não se encontrarem no centro de rotação é irrelevante, pois o termo associado será nulo.

Importa referir que as medidas efectuadas apenas contemplam o resultado para um pixel em cada uma das imagens, pelo que efectuando uma média para todos os pontos das ima-

gens para o caso da medida $\frac{Z_l - Z_r}{Z_l + Z_r}$, obtém-se uma estimativa proporcional à profundidade de todos os pontos visualizados

$$\frac{\sum_{i=0}^{n-1} \frac{Z_{li} - Z_{ri}}{Z_{li} + Z_{ri}}}{n} \quad (5.19)$$

Em termos práticos, apenas os locais onde se encontram as arestas dos objectos apresentam fluxo óptico normal. Este facto introduz uma restrição muito forte caso se pretenda utilizar o algoritmo de fluxo óptico normal, uma vez que os “pixels” correspondentes indicados na Fig. 5.6 podem não apresentar fluxo devido à ausência de arestas nesses mesmos pontos. Nesta situação será impossível de medir correctamente as quantidades indicadas, uma vez que é necessário utilizar o fluxo desses dois pontos. Uma forma de ultrapassar este facto pode passar pela utilização do fluxo afim, já que desta forma para todos os “pixels” da imagem é possível calcular o fluxo óptico. No entanto a introdução deste método de cálculo obriga a restringir o meio ambiente a superfícies planares, não sendo no entanto crítico, já que as paredes do corredor podem ser aproximadas por planos.

De seguida vai-se generalizar o estudo do posicionamento das câmaras. A ideia é tentar que as restrições impostas até agora ao posicionamento das câmaras não sejam tão restritivas como as actuais.

Simetria entre câmaras Convergentes e Divergentes

Após a análise efectuada anteriormente, neste momento é importante generalizar outros posicionamentos para as câmaras. Pretende-se assim estudar qual o efeito da variação do ângulo de vergência das câmaras, nas equações 5.15, 5.16, 5.17 e 5.18. Para isso suponhamos que as câmaras direita e esquerda podem ser posicionadas segundo um ângulo de vergência simétrico (rotação do eixo y do referencial das câmaras), dando origem a um possível posicionamento simétrico convergente ou simétrico divergente, conforme indicado na Fig. 5.9. Da análise efectuada na secção anterior, apenas existe a necessidade de alterar a equação 5.14, de forma a incluir uma matriz de transformação que representa a rotação no eixo y , para cada um dos referenciais, ou seja, $M_r = R_{ri} R_y(\Leftrightarrow v_{rg})$ e $M_l = R_{li} R_y(v_{rg})$, onde v_{rg} é um ângulo de vergência, assumido como sendo simétrico para ambas as câmaras.

Verificou-se anteriormente para $v_{rg} = 0$, que é um dos casos mais simples, que as equações 5.15, 5.16, 5.17 e 5.18 dependem claramente das velocidades de rotação

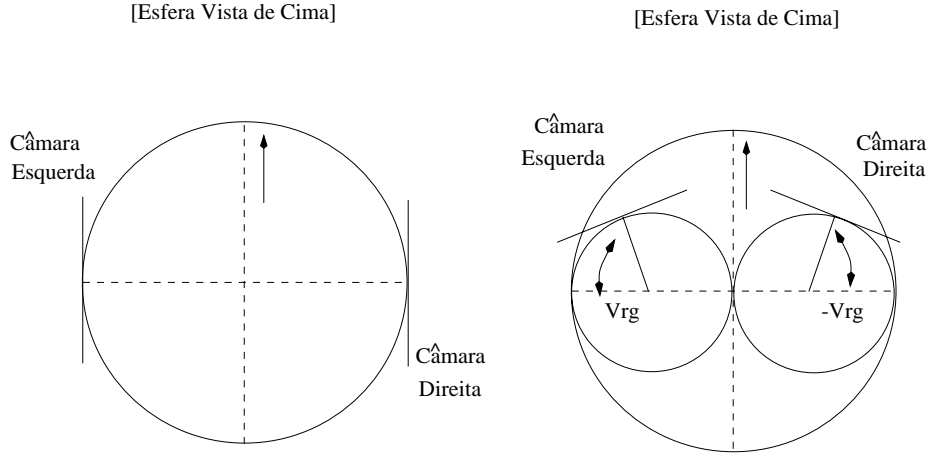


Figura 5.9: Estudo do posicionamento das câmaras para o caso convergente/divergente. À esquerda apresenta-se o caso de $v_{rg} = 0$ estudado na secção 5.2, enquanto na Fig. da direita temos o caso de um posicionamento simétrico divergente ($v_{rg} \in [0, \frac{\pi}{2}[$) ou convergente ($v_{rg} \in [\frac{\pi}{2}, \pi[$).

mesmo que as câmaras se encontrem posicionadas sobre o eixo de rotação, pelo que não é de esperar que estas se tornem mais simples para o caso de $v_{rg} \neq 0$. No entanto analisando as equações apenas na presença da velocidade de translação e somando os respectivos fluxos \dot{u} obtém-se:

$$\dot{u}_r(u, v) + \dot{u}_l(\Leftrightarrow u, v) = \frac{Vr [\cos(v_{rg})fx + \sin(v_{rg})u] (Zl \Leftrightarrow Zr)}{Zl Zr} \quad (5.20)$$

Subtraindo os fluxos \dot{u} obtém-se:

$$\dot{u}_r(u, v) \Leftrightarrow \dot{u}_l(\Leftrightarrow u, v) = \frac{Vr [\cos(v_{rg})fx + \sin(v_{rg})u] (Zl + Zr)}{Zl Zr} \quad (5.21)$$

Somando os respectivos fluxos \dot{v} obtém-se:

$$\dot{v}_r(u, v) + \dot{v}_l(\Leftrightarrow u, v) = \frac{\sin(v_{rg}) Vr v (Zl + Zr)}{Zl Zr} \quad (5.22)$$

Subtraindo os fluxos \dot{v} obtém-se:

$$\dot{v}_r(u, v) \Leftrightarrow \dot{v}_l(\Leftrightarrow u, v) = \Leftrightarrow \frac{\sin(v_{rg}) Vr v (\Leftrightarrow Zl + Zr)}{Zl Zr} \quad (5.23)$$

Analisando as equações 5.20, 5.21, 5.22 e 5.23, podemos verificar a seguinte propriedade:

$$\frac{\dot{u}_r(u, v) + \dot{u}_l(\Leftrightarrow u, v)}{\dot{u}_r(u, v) \Leftrightarrow \dot{u}_l(\Leftrightarrow u, v)} = \frac{\dot{v}_r(u, v) + \dot{v}_l(\Leftrightarrow u, v)}{\dot{v}_r(u, v) \Leftrightarrow \dot{v}_l(\Leftrightarrow u, v)} = \frac{Zl \Leftrightarrow Zr}{Zl + Zr} \quad (5.24)$$

cujo resultado não depende do ângulo de vergência (esta equação é válida para todos os pontos à excepção de alguns pontos u da imagem e posições de vergência que dão origem a uma divisão por zero, por exemplo, para o caso de \dot{v} para $v_{rg} = 0$ ou $v_{rg} = \pi$, e para o caso de \dot{u} para $u = 0$ quando $v_{rg} = \frac{\pi}{2}$ ou $v_{rg} = \frac{3\pi}{2}$). Este resultado é demasiado abrangente e importante e dele podem ser tiradas as seguintes conclusões, se forem excluídos os casos singulares:

- na presença apenas da velocidade de translação, com base no fluxo medido em posições correspondentes de ambas as imagens, pode-se estimar sempre o valor dado pela eq. 5.24.
- a profundidade é medida segundo o eixo óptico das câmaras e o posicionamento deste depende do ângulo de vergência.
- o valor medido é independente da posição de vergência das câmaras, logo estas podem ser convergentes ou divergentes.

Um outro processo onde pode ser utilizado este mesmo princípio de posicionamento das câmaras é o de seguimento de objectos no espaço tridimensional.

A título final, pode-se utilizar um sistema de visão activa normal para efectuar acções de navegação, apesar de este não verificar na totalidade as restrições do sensor esférico virtual.

5.3 Navegação com duas câmaras

Com base no descrito anteriormente, o primeiro processo de navegação que foi testado usa duas câmaras posicionadas simetricamente relativamente ao eixo de rotação do robot como indicado na Fig. 5.10. Tomando como base os seguintes factos:

- o movimento de rotação do robot deve ser controlado para evitar o choque com as paredes do corredor
- a velocidade de translação do robot pode ser controlada se for desejável, no entanto não é um caso crítico

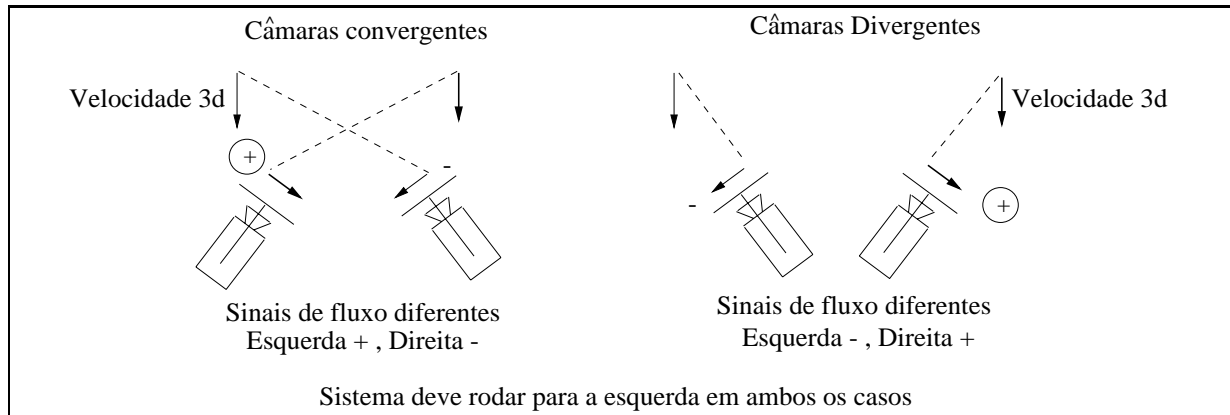


Figura 5.10: A figura mostra um exemplo, onde a plataforma móvel deve rodar para a esquerda, pois o fluxo do lado direito é superior em módulo ao do lado esquerdo. Utilizando a estratégia de posicionamento e de controle, é indiferente se as câmaras estão numa posição convergente ou divergente.

definiu-se o sinal de “feedback” que controla a velocidade de translação como

$$F feed = \sum \sqrt{\dot{u}^2 + \dot{v}^2} \quad (5.25)$$

Este sinal é o módulo do fluxo óptico, que fornece informação sobre os objectos texturados que estejam a ser visualizados. Assumindo uma textura homogênea e no caso do movimento ser elevado, provavelmente existem muitos objectos em linha de vista, pelo que será mais cauteloso abrandar a velocidade de translação.

Para efectuar o controlo do movimento de rotação do robot, efectua-se o cálculo indicado pela eq. 5.24 para cada pixel. De seguida somando o valor desse cálculo para vários “pixels” obtém-se:

$$H feed = \frac{\sum_{u=-k_u}^{k_u} \sum_{v=-k_v}^{k_v} \frac{\dot{u}_r(u,v) + \dot{u}_l(-u,v)}{\dot{u}_r(u,v) - \dot{u}_l(-u,v)}}{(2 * k_u + 1) * (2 * k_v + 1)} = \frac{\sum_{u=-k_u}^{k_u} \sum_{v=-k_v}^{k_v} \frac{Z_l - Z_r}{Z_l + Z_r}}{(2 * k_u + 1) * (2 * k_v + 1)} \quad (5.26)$$

Para as experiências realizadas para testes neste capítulo, o algoritmo de cálculo do fluxo óptico utilizado é o método afim. Com base no sinal obtido através da eq. 5.26 um dos algoritmos de controlo da velocidade angular que se pode elaborar para a plataforma móvel encontra-se indicado na Fig. 5.11.

A Fig. 5.12 apresenta o sistema de eventos discretos de modo a permitir controlar a velocidade de rotação do robot móvel. Este sistema discreto é necessário, pois a estimação

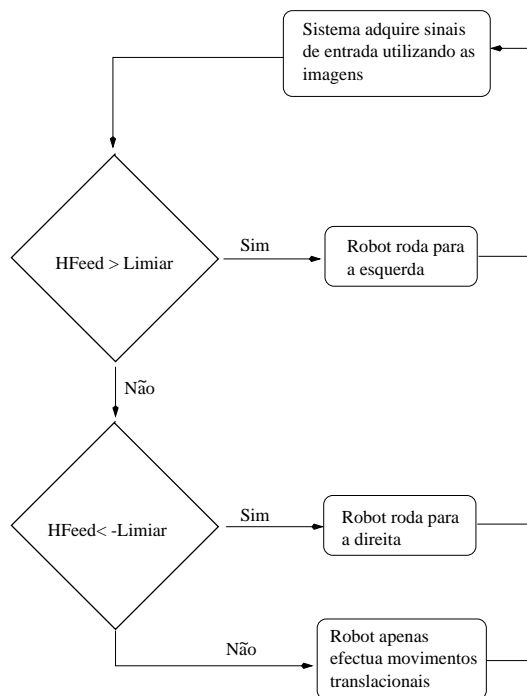


Figura 5.11: Algoritmo de controlo para determinar quando é que a plataforma móvel deve efectuar movimentos de rotação. A velocidade de rotação é estipulada através duma constante proporcional, i.e., $\omega_z = \omega_0 + k_p (HFeed \Leftrightarrow Limiar)$.

dos sinais de erro do sistema apenas são efectuadas quando o robot móvel apresenta unicamente velocidade de translação. O estado **0** serve para garantir esta condição, enquanto no estado **1** se executa o plano de controlo. O processamento das imagens é efectuado a $25Hz$ para ambas as câmaras e o controlo da plataforma móvel a $2Hz$ durante o estado **0**

Definindo a função de erro como:

$$e_{trans.} = (Feed_{direita} + Feed_{esquerda}) \Leftrightarrow F_{desejado} \quad (5.27)$$

pode-se usar um controlo baseado num controlador expresso por:

$$u[n] = k_p e[n] + k_i \sum_{k=0}^m e[n \Leftrightarrow k] + k_d (e[n] \Leftrightarrow e[n \Leftrightarrow 1]) \quad (5.28)$$

onde os parâmetros k_p , k_i e k_d são ajustados experimentalmente.

A saída do controlador é utilizada para comandar a velocidade de translação, a menos de uma constante obtida através de uma calibração experimental. Esta calibração foi efectuada para permitir controlar o veículo dentro de certos limites de velocidade. Note-se que o sinal $u[n]$ comanda a saída em velocidade do sistema relativamente a um valor pré-estabelecido, i.e., se $u[n] = 0$ a saída não será nula.

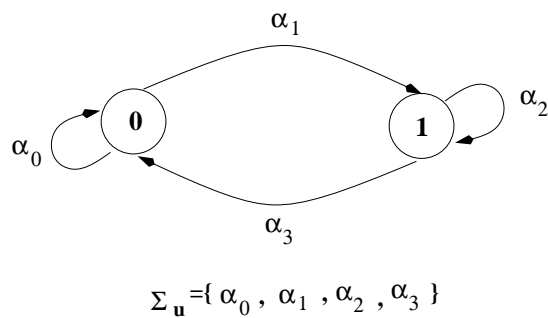
Dos teste efectuados com esta configuração nunca se obtiveram resultados práticos satisfatórios, devido à velocidade de rotação, pelo que se passou a configuração descrita na próxima secção.

5.4 Navegação com duas câmaras e espelhos

5.4.1 Espelhos como simuladores de várias câmaras

A técnica de navegação baseada em espelhos, usa a configuração indicada na Fig. 5.14. Associado a cada câmara existem dois espelhos, que simulam dois pontos de vista aproximadamente do mesmo local. Considerando que as paredes do corredor são planares (i.e. pequenas dimensões da parede podem ser aproximadas por um plano), com base nos sinais de “feedback” detectados em cada um dos espelhos, obtém-se uma medida que está relacionada com o ângulo de posicionamento do plano em relação ao sistema de medição

Modelo para o controlo da velocidade angular



Descrição dos Estados

- 0**: Plataforma móvel apenas se movimenta com velocidade translacional
- 1**: Controlo da velocidade angular do robot móvel

Descrição dos eventos

- α_0 : Estima as entidades angulares em causa enquanto o robot apenas efectua movimentos de translação. Esta estimativa é confirmada através da aquisição de várias amostras do sinal, uma vez que o processamento das imagens funciona a $25Hz$ enquanto o controlo da plataforma móvel é executado à cadência de $2Hz$. O sinal *HFeed* é obtido através da média das várias amostras efectuadas.
- α_1 : Inicia o controlador da velocidade angular do sistema
- α_2 : Aguarda um tempo fixo pré-estabelecido, T_s
- α_3 : Envia à plataforma móvel um comando para esta executar apenas movimentos translacionais

Figura 5.12: Modelo do sistema de eventos discreto para o controlo da posição angular do robot móvel.

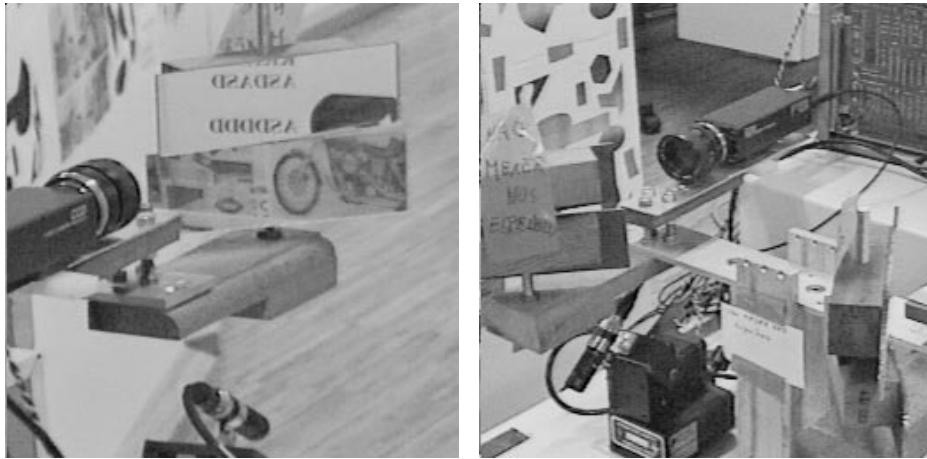


Figura 5.13: Sistema de navegação experimental com quatro espelhos. Utiliza uma placa de aquisição com dois processadores digitais de sinal.

(conforme pode ser visto na Fig. 5.15). Esta relação depende do modelo geométrico associado aos espelhos em conjunção com a câmara, e aos ângulos dos espelhos relativamente à câmara.

Desta forma é possível decidir se o robot móvel está ou não em rota de colisão com a parede.

Utilizando a configuração indicada na Fig. 5.14, pode-se verificar que o sistema é simétrico, pelo que apenas é necessário analisar uma das câmaras. Considere-se para isso a Fig. 5.15. Indicando os sinais medidos em cada imagem dos espelhos como:

1. $MMRU$ - fluxo óptico medido no espelho de proa na imagem da direita
2. $MMRD$ - fluxo óptico medido no espelho de popa na imagem da direita
3. $MMLU$ - fluxo óptico medido no espelho de proa na imagem da esquerda
4. $MMLD$ - fluxo óptico medido no espelho de popa na imagem da esquerda

uma estratégia de controlo para este caso, que pode ser utilizada é a seguinte (apenas considerando as estimativas da zona direita):

1. Se $\| MMRU \Leftrightarrow MMRD \| < \text{limiar}$ ENTÃO efectua movimentos de translação ou de rotação para a esquerda

Configuração do sistema de navegação com espelhos

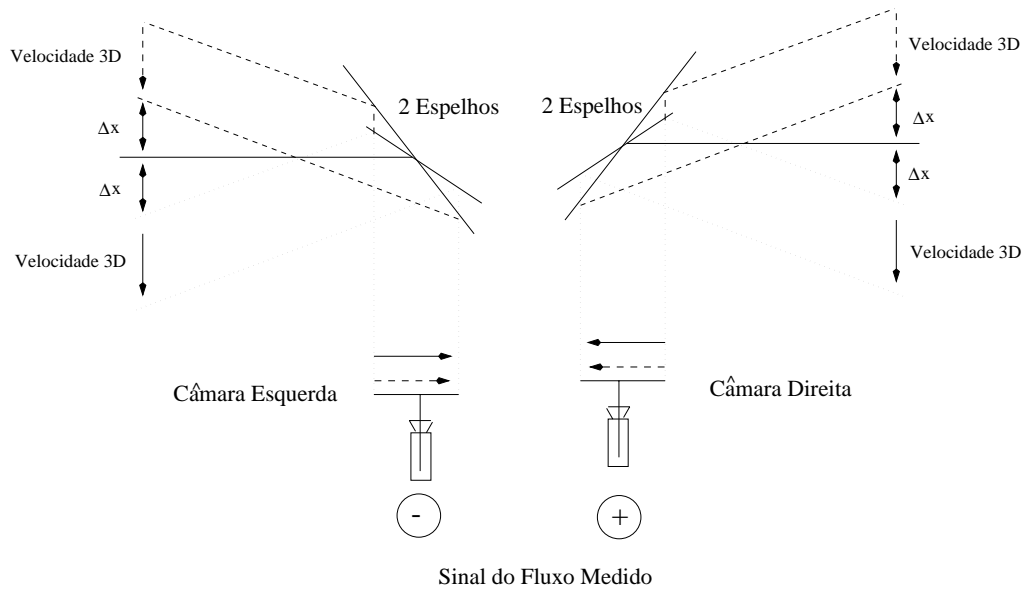


Figura 5.14: Nesta Fig. para simplicidade de desenho não se representou a câmara real através do modelo de perspectiva geométrica, apesar de este ser o modelo correcto.

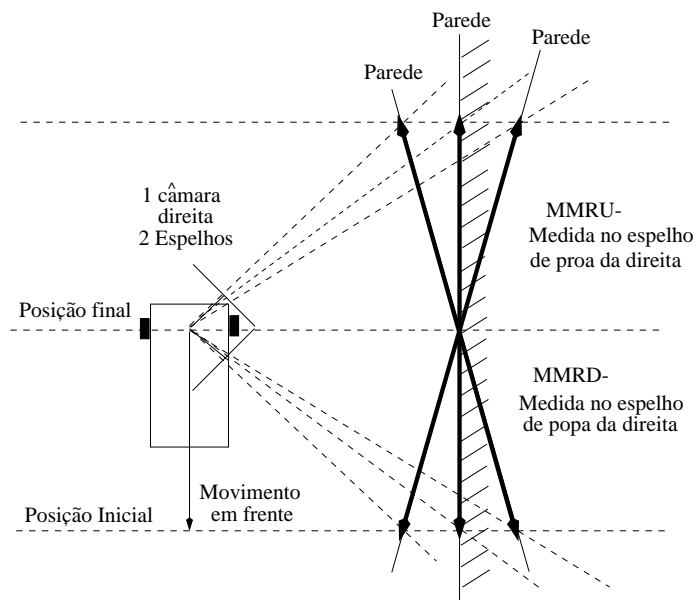


Figura 5.15: Identificação dos sinais medidos nos espelhos associados à câmara direita do sistema

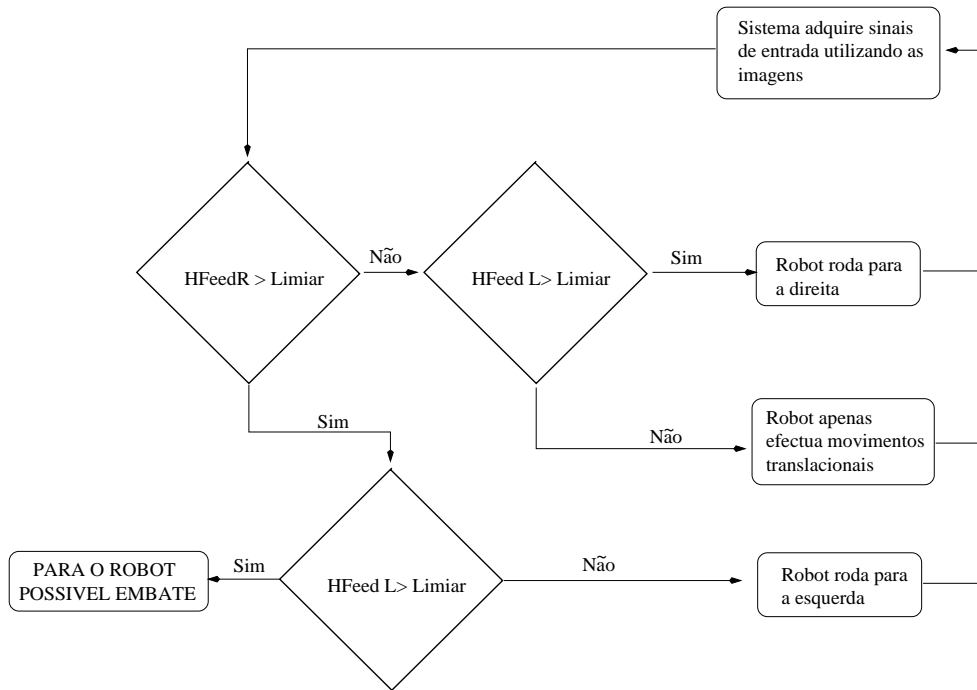


Figura 5.16: Algoritmo de controlo para determinar quando é que a plataforma móvel deve efectuar movimentos de rotação para o caso dos espelhos.

2. Se $MMRU \Leftrightarrow MMRD > \text{limiar}$ ENTÃO efectua movimentos de rotação para a esquerda
3. Se $MMRU \Leftrightarrow MMRD < \Leftrightarrow \text{limiar}$ ENTÃO efectua movimentos de rotação: para a esquerda ou direita (depende das medidas efectuadas nas imagens da câmara esquerda do sistema)

Definindo $HfeedR = MMRU \Leftrightarrow MMRD$ e $HfeedL = MMLU \Leftrightarrow MMLD$ o algoritmo global de funcionamento incorporando as estimativas obtidas das quatro imagens pode ser o descrito através do fluxograma representado na Fig. 5.16. Salienta-se para o facto de neste caso a velocidade de rotação ser constante bem como a velocidade translacional. O algoritmo de controlo apenas decide qual das acções a tomar.

O sistema de eventos discretos é o indicado na Fig. 5.12. No estado **1** é utilizada a estratégia de controlo indicada anteriormente, enquanto no estado **0** são adquiridos os sinais de medida necessários a este método de controlo, tal como descrito na Fig. 5.12.

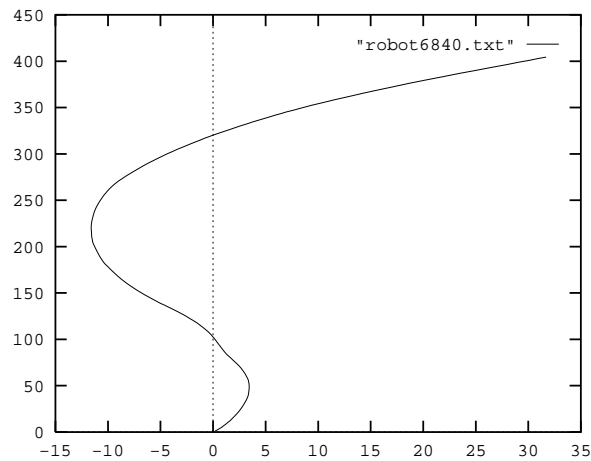


Figura 5.17: Evolução da posição Y versus X do robot.

O fluxo óptico é medido de igual modo ao processo sem espelhos, i.e. mede-se a componente de fluxo segundo a horizontal. No entanto, a diferença neste caso encontra-se no processo utilizado, i.e. essa componente é medida através dum processo de correlação unidimensional que detecta o deslocamento horizontal entre imagens consecutivas.

5.5 Resultados Simulados e Experimentais

Nesta secção apresentam-se os resultados simulados/práticos dos dois processos. O primeiro processo foi implementado tanto em simulação como numa situação real, no entanto os resultados com o equipamento existente nunca chegaram a ser suficientemente satisfatórios, pelo que não são aqui apresentados. Quanto ao caso da navegação com quatro espelhos, não foi efectuada simulação, pelo que apenas se apresentam os resultados obtidos numa situação real, que demonstram com sucesso o sistema de navegação.

5.5.1 Resultados simulados

Para validar a primeira estratégia, foi desenvolvido um robot móvel simulado, com parâmetros idênticos ao robot real, com a capacidade de gerar imagens do local onde se encontra posicionado através dum programa de “ray tracing” [Povray Team]. O meio ambiente é

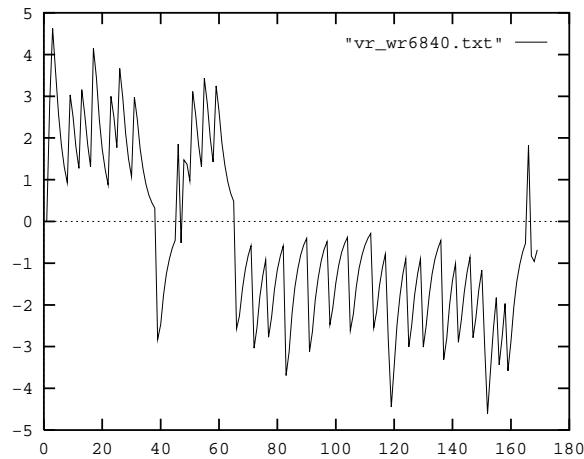


Figura 5.18: Velocidade angular do Robot para o caso da Fig. 5.17.

programável, e independente do processo, e neste caso específico foi programado um corredor virtual. O robot móvel apenas pode ser controlado em velocidade, e possui um modelo dum sistema de 1ª ordem com uma constante de tempo programável. A cena visualizada pode ser alterada pelo utilizador, para isso é necessário alterar o ficheiro que descreve a cena, sem maiores alterações. O simulador foi desenvolvido em sistema Unix, e segue uma filosofia de cliente/servidor. Permite vários utilizadores ligados simultaneamente até um máximo de 4 (este valor apenas é limitado pelo poder de cálculo da máquina, uma vez que o programa de “ray tracing” é computacionalmente pesado). Para o cliente utilizar o simulador basta usar uma ligação por “sockets” num determinado porto [FreeBSD Team]. O interface é semelhante ao duma “shell” através do uso de comandos pré-definidos.

Nos seguintes exemplos, o robot navega num corredor, onde as paredes interceptam-se em triângulo, num ponto mais à frente do robot (nota: o referencial do robot simulado é diferente do referencial da Fig. B.1, i.e., está rodado de 90° no sentido do ponteiro do relógio, e a orientação do robot é medida relativamente ao eixo dos Y). O tempo de amostragem é neste caso de 0.2 segundos, o “setup” inicial é $Vrg_{right} = 10.0^\circ$, $Vrg_{left} = \Leftrightarrow 10.0^\circ$ e a posição inicial do robot de $(X, Y, \theta) = (0.0, 0.0, \Leftrightarrow 10.0^\circ)$, o que levaria o robot a colidir com a parede, se este seguiu-se em frente. A Fig. 5.17 mostra a evolução da posição do robot para o caso do processo de navegação sem espelhos. Na cena de teste, o ângulo máximo admitido era neste caso de 45° , caso contrário o sistema era incapaz

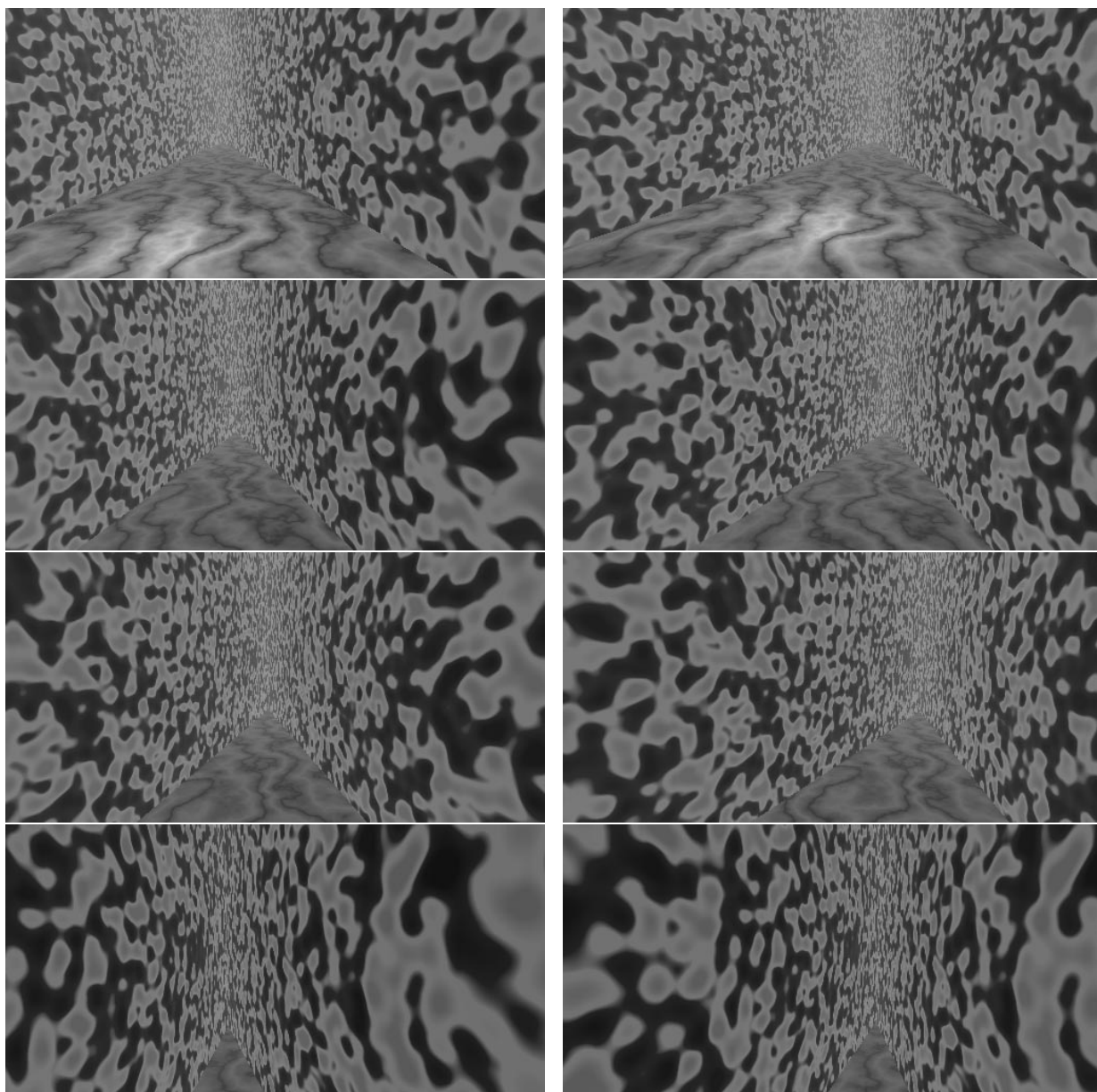


Figura 5.19: Imagens de ambas as câmaras nos seguintes tempos de amostragem: 4, 87, 102 e 180, para o caso de navegação sem espelhos.

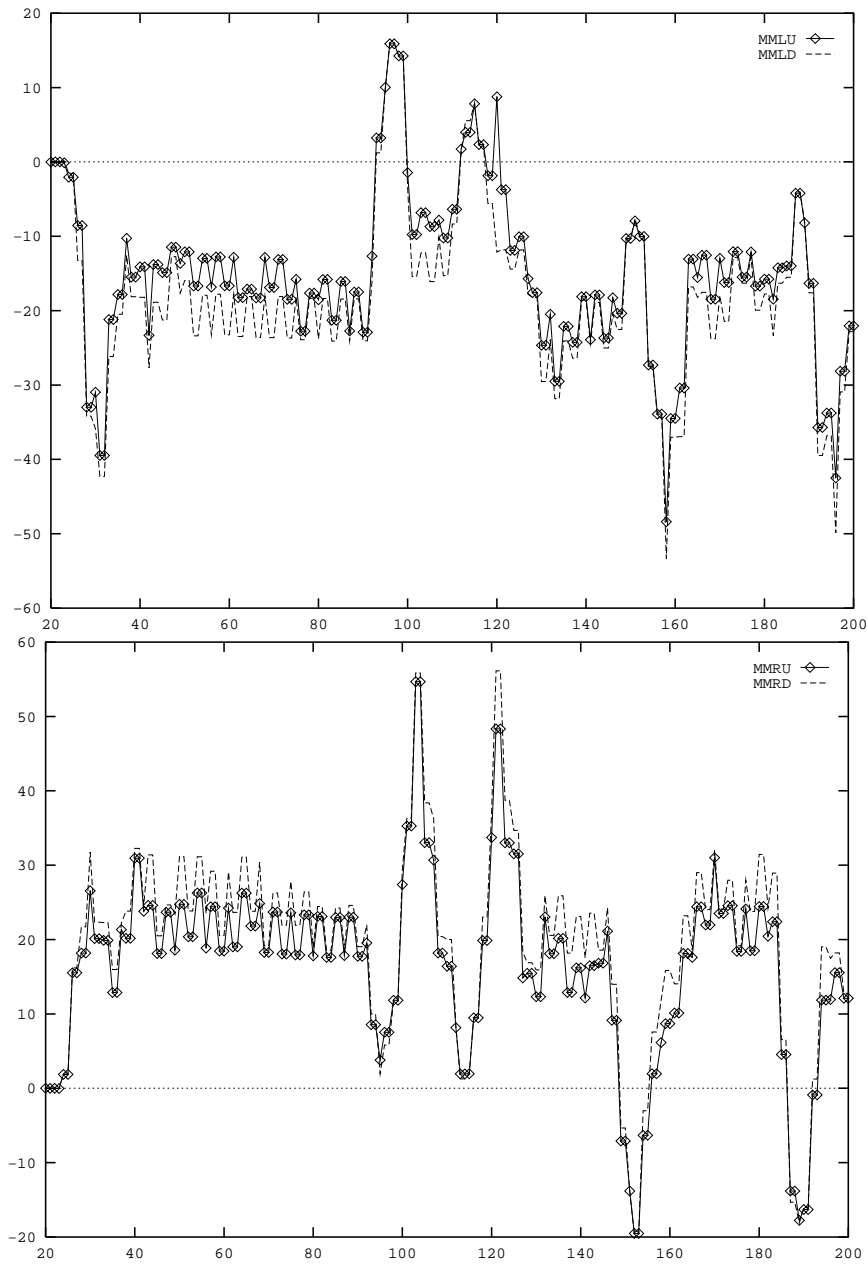


Figura 5.21: Medidas dos sinais MMLU/MMLD e MMRU/MMRD respectivamente (tempo de aquisição 0.08s ou seja 12.5Hz), para o caso da navegação com espelhos.

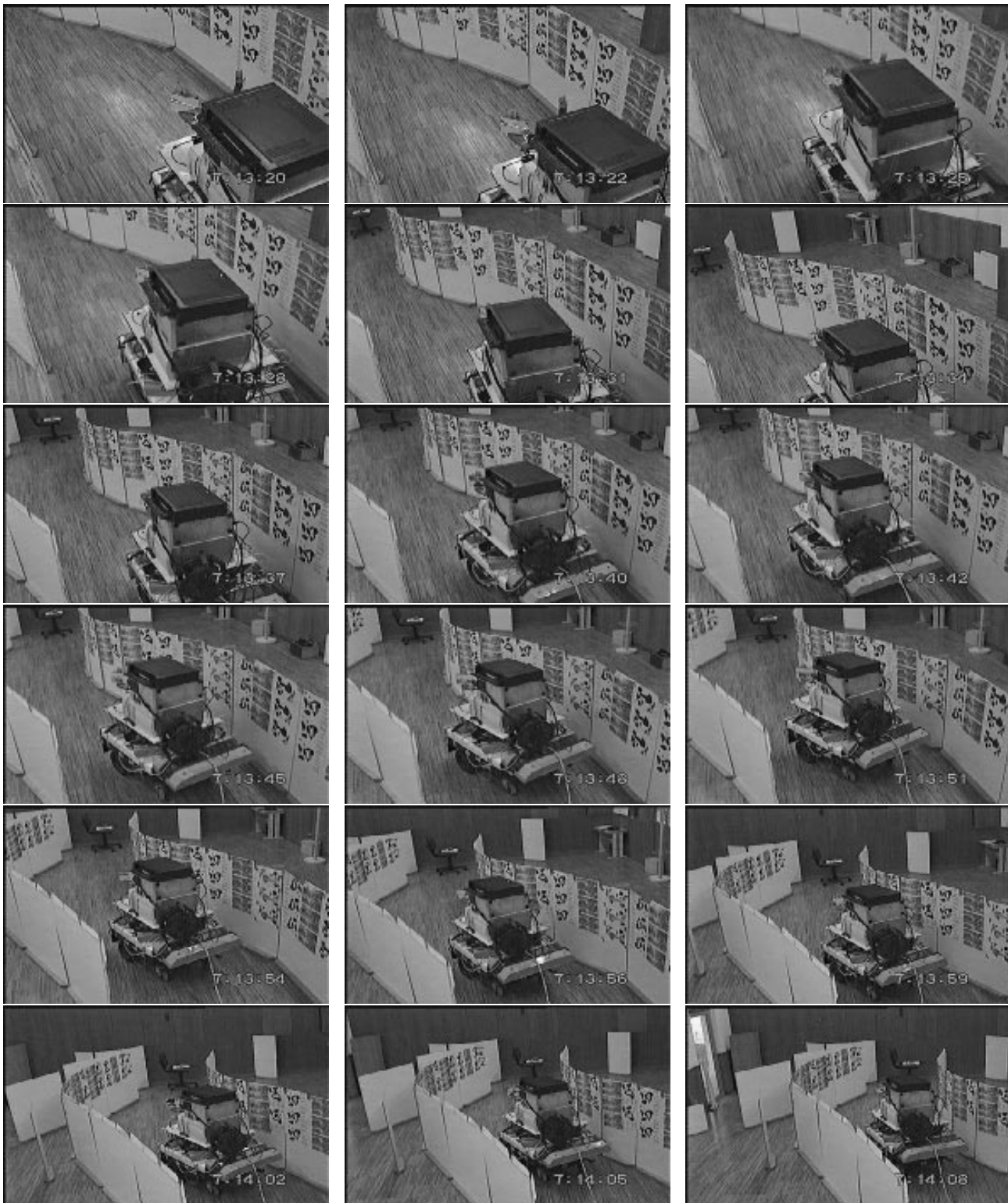


Figura 5.22: Navegação com espelhos (1ª parte - imagens exteriores)



Figura 5.23: Navegação com espelhos (2ª parte - imagens exteriores)

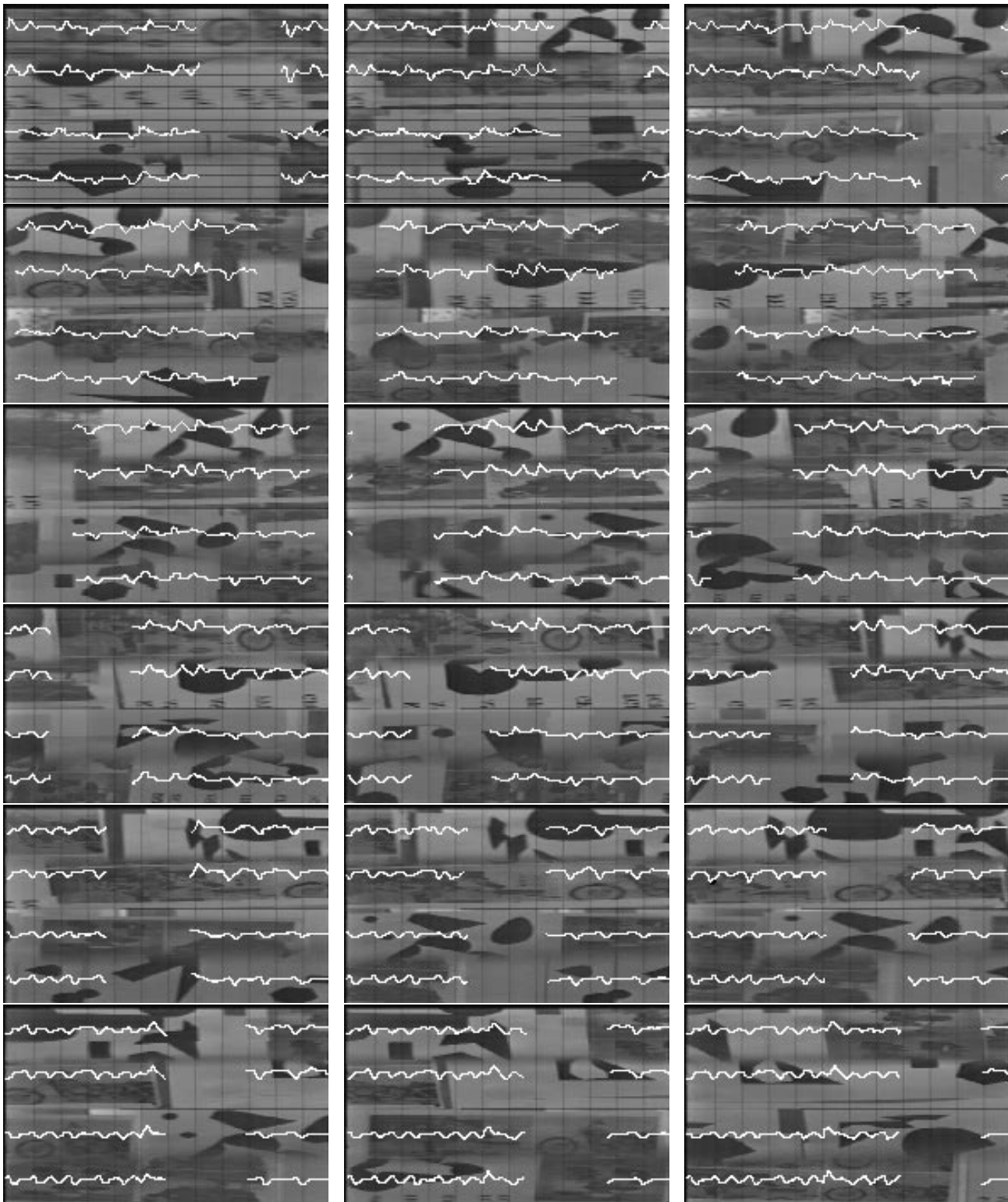


Figura 5.24: Navegação com espelhos (1ª parte - imagens interiores)

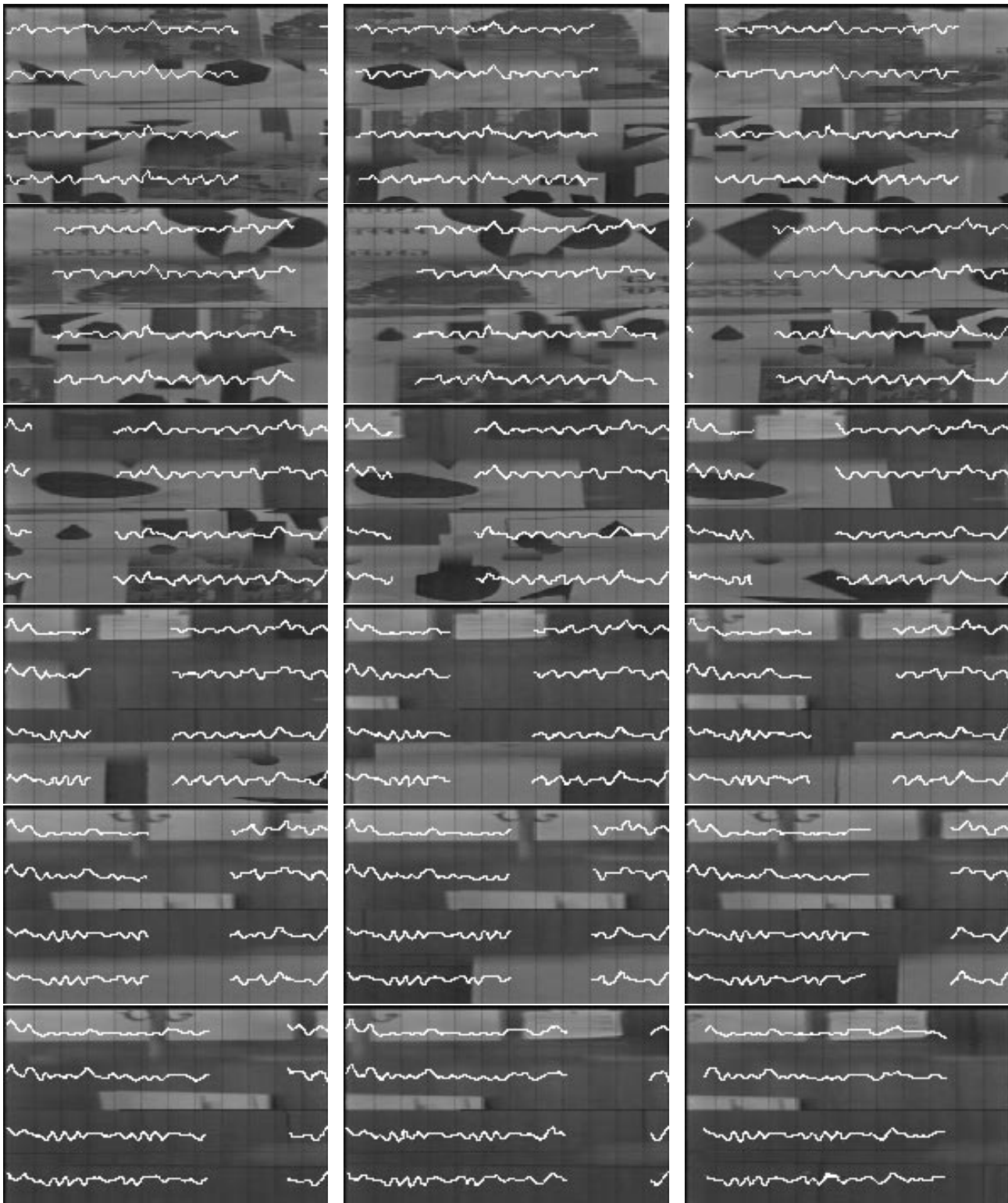


Figura 5.25: Navegação com espelhos (2ª parte - imagens interiores)

Capítulo 6

Conclusões e trabalho Futuro

6.1 Conclusões

Nesta tese pretendeu-se implementar alguns processos visuais, utilizando técnicas de visão. Concretamente o trabalho foi sendo definido à medida que o tempo foi decorrendo, tendo no entanto como objectivo principal o desenvolvimento de um sistema que fosse capaz de navegar num corredor. As soluções deveriam à partida utilizar fluxo óptico normal sobre imagens log-polar. Como é evidente, sabe-se a priori, que o fluxo óptico normal é ruidoso, e quando utilizado sobre imagens log-polar os sinais daí provenientes tornam-se ainda mais difíceis de tratar. Estes sinais são dependentes dos parâmetros utilizados na transformação log-polar, uma vez que esta transformação funciona como um filtro que deve ser ajustado para uma determinada gama. Devido ao ruído associado ao fluxo óptico normal, o sistema nunca chegou a funcionar correctamente através desta técnica.

O passo seguinte passou pela introdução do fluxo afim sobre as imagens log-polar. O processo de navegação passou a ser mais estável, devido ao fluxo afim minimizar globalmente a restrição do brilho, captando assim a tendência do movimento. O facto do modelo associado ao fluxo afim, ser definido para superfícies planares, não é uma restrição severa, uma vez que a aproximação é válida se as câmaras estiverem a visualizar uma área relativamente grande da parede. Este processo funciona no entanto com algumas restrições, devido ao problema da textura necessitar de ser aproximadamente igual em ambas as

paredes.

Finalmente, o sistema, que actualmente se encontra desenvolvido, e com resultados satisfatórios, utiliza uma técnica baseada em dois espelhos de modo a podermos comparar não o fluxo entre as duas paredes, mas duas quantidades do fluxo medido sobre a mesma parede. Desta forma desacoplamos o problema da textura, e o processo torna-se assim estável. É verdade, que ficam ainda alguns problemas por resolver, nomeadamente o problema da velocidade, isto porque à data o sistema tem dificuldades em navegar a uma velocidade superior à que consegue actualmente (230mm/s). Outros problemas que se podem apontar a esta técnica, têm a ver com a questão da continuidade das inclinações da parede (não funciona com ângulos rectos), e com o ângulo inicial com que o sistema inicia o processo, que não pode ser superior a uma inclinação de 35° .

Quanto à detecção do *FOE* e realinhamento da câmara de acordo com o vector de velocidade de translação do robot, já foram previamente apresentadas as respectivas conclusões, no entanto apenas seria importante de referir, que apesar de funcionalmente o processo apresentar um bom desempenho, a sua utilização num sistema prático tem pouca ou nenhuma utilidade. Isto porque à partida, ao se iniciar o movimento do robot, sabe-se quais são os parâmetros do movimento, pelo que o *FOE* não vem apresentar nenhuma informação valiosa ao processo. Uma das aplicações, em que talvez possa vir a ser útil, seria a de estabilizar a imagem relativamente a pequenos “micro-movimentos” da câmara, devido aos movimentos mais bruscos do robot, no entanto, para efectuar esta tarefa, era necessária uma elevada precisão no cálculo do *FOE*, o que está longe de ser o caso (o processo actual apresenta em média erros de 10%). No entanto não deixa de ser uma aplicação interessante, como mostram as mais recentes câmaras de filmagens aéreas (a estabilização não é como é óbvio feita por imagens, mas por inclinómetros).

Em conclusão:

- a resolução das imagens log-polares é inferior à resolução das imagens cartesianas, permitindo assim ganhar tempo para efectuar mais cálculos, no entanto à que ter em conta, o processo de transformação, pois em termos práticos, a transformação requer tabelas, o que do ponto de vista computacional significa um maior número de acessos à memória,

- o processo visual de navegação com fluxo óptico normal revelou-se instável, devido aos erros associados ao fluxo normal,
- a navegação com os espelhos é estável, no entanto, como tudo, pode ainda ser bastante melhorada,
- a detecção do *FOE* ainda não é suficientemente precisa para poder ser utilizada na resolução de problemas mais práticos e de maior utilidade,
- quanto às técnicas de detecção do fluxo óptico, pessoalmente, acho que enquanto não for resolvido o problema da estabilidade, isto é, ter a certeza que o pixel (u, v) se movimentou mesmo de uma velocidade (v_u, v_v) , tudo pode acontecer, e pessoalmente sou mais a favor de fluxo baseado em técnicas de correlação, do que técnicas baseadas na restrição do brilho.

6.2 Trabalho futuro

Relativamente a trabalho a desenvolver futuramente, no caso da navegação autónoma, gostaria primeiro de salientar, que normalmente um ser humano quando se desloca num meio ambiente, utiliza como referências pontos de interesse - “landmarks” -, sons, etc. Como o que nos interessa, é a utilização da visão, para efectuar navegação, se decidirmos enveredar por este caminho, o problema, neste caso, consiste em saber como são seleccionadas essas “landmarks”. É evidente, que na escolha das “landmarks” reside a chave do problema, uma vez que seleccionada uma “landmark”, pode-se impor como objectivo, chegar o mais perto possível, ou manter uma distância constante, etc. A vantagem deste processo, advém do facto de a nível do processo visual, poder ser resolvido com correladores, o que é um processo bem mais estável do que o fluxo óptico, se bem que mais dispendioso computacionalmente, e aqui sim, as imagens log-polar podem desempenhar um papel preponderante, conforme já foi provado por trabalhos desenvolvidos por [G. Sandini e C. Capurro 95]. Dos vários problemas que esta filosofia possa apresentar, um está relacionado com a estimação da distância da base móvel à “landmark”. As pessoas que já trabalharam em “tracking”, das quais eu faço parte, sabem perfeitamente, que

é necessário uma elevada precisão a nível do posicionamento dos motores de vergência, e de pan do sistema que tenta efectuar o tracking. Um outro factor crítico, encontra-se a nível da sincronização de processos, tais como, ler a posição do "encoder" o mais próximo possível do momento de aquisição da imagem, uma boa calibração, etc. O engraçado, é que relativamente à estimação da distância à "landmark", esta poder ser dada apenas por uma câmara onde o zoom seja controlado de modo a manter constantes as dimensões do padrão na imagem. Desta forma a distância à "landmark" seria dada em função da posição do motor de zoom da câmara, eliminando assim os problemas inerentes à utilização da triangulação a quando da utilização das duas câmaras. Existem como é evidente, problemas a nível do tracking da "landmark", que têm a ver com o ponto de vista da câmara relativamente à "landmark". Quanto ao processo de escolha da "landmark", é algo que a nível computacional é "transcendental", pois requer inteligência por parte da máquina (lá chegaremos, mas talvez infelizmente, não na minha geração). No entanto, uma das possibilidades para a selecção das "landmarks", pode passar pela simulação de dois ouvidos, de modo a implementar processos de sacada activados sonoramente. Este pode ser de facto um caminho a seguir, integrar processos auditivos em conjunção com processos visuais, uma vez que no caso dos seres humanos, podemos verificar que eles se encontram estritamente relacionados.

Apêndice A

Definições e convenções

Neste trabalho são usadas as seguintes notações:

(u, v) - referência a coordenadas cartesianas

(\dot{u}, \dot{v}) - referência ao fluxo óptico no plano cartesiano

(ρ, θ) - referência a coordenadas polares

$(\dot{\rho}, \dot{\theta})$ - referência ao fluxo óptico no plano polar (ou plano retinal)

(η, θ) - referência a coordenadas log-polares (num domínio contínuo)

$(\dot{\eta}, \dot{\theta})$ - referência ao fluxo óptico no plano log-polar (por vezes designado plano cortical)

(ζ, γ) - referência a coordenadas log-polares (num domínio discreto)

$(\dot{\zeta}, \dot{\gamma})$ - referência ao fluxo óptico no plano log-polar (por vezes designado plano cortical)

f - distância focal da câmara

s_u - factor de escala na direcção u do plano cartesiano

s_v - factor de escala na direcção v do plano cartesiano

$fx = f \cdot s_u$ - distância focal na direcção do eixo u do plano cartesiano

$fy = f \cdot s_v$ - distância focal na direcção do eixo v do plano cartesiano

$[X, Y, Z]$ - coordenadas tridimensionais no referencial câmara

i - número imaginário ($i = \sqrt{-1}$)

Apêndice B

O Sistema do Robot Móvel

O robot móvel utilizado no trabalho é designado por ROBUTER [Robuter 92]. Os movimentos executados pela plataforma móvel advêm de dois motores DC associados a cada uma das rodas motoras. Os movimentos permitidos por esta configuração encontram-se representados na Fig. B.1. O controlo dos motores é efectuado através dum sistema de multiprocessamento baseado no processador 68020, instalado na plataforma móvel. A única responsabilidade do PC que controla o robot, é a de enviar os parâmetros do movimento que se pretende executado. Os movimentos representados na Fig. B.1 podem ser divididos em três grupos: translacional (sem velocidade angular), rotação em torno do centro do eixo central representado na fig. pelo ponto *RC* (sem velocidade linear), e movimentos compostos. Para definir um dos três movimentos descritos, além de ser necessário indicar a velocidade angular e linear, é também necessário indicar a duração do movimento.

<i>Tipo de Movimento</i>	<i>Velocidade Linear (mm/s)</i>	<i>Velocidade Angular (mrad/s)</i>	<i>Trajectória Raio</i>	<i>Tempo de duração (unidades de 40ms)</i>
translacional	$\pm 1000, \dots, \pm 50, 0$	0	0	> 0
rotacional	0	$\pm 1000, \dots, \pm 100, 0$	0	> 0
composto	$\pm 1000, \dots, \pm 50, 0$	$\pm 1000, \dots, \pm 100, 0$	$> 400mm$	> 0

Tabela B.1: Parâmetros para controlar a velocidade da plataforma móvel

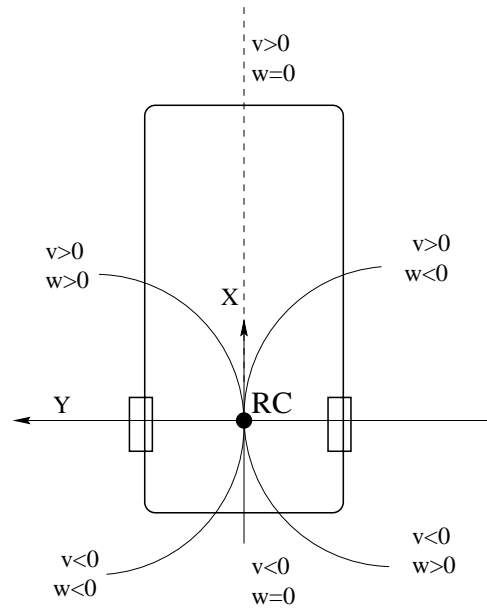


Figura B.1: Movimentos possíveis de serem executados pela plataforma móvel. w é a velocidade angular, v é a velocidade linear, r é o raio da trajetória e θ o ângulo de orientação.

Os valores válidos para as velocidades e duração do movimento dependem do tipo de movimento a executar, e são representados na tabela B.1. Se o movimento é do tipo composto, o raio e o arco da trajetória podem ser obtidos através das seguintes relações (T designa o tempo disponível para executar o movimento):

$$r[mm] = \frac{v[mm/s]}{w[mrad/s]} 10^3 \quad (B.1)$$

$$\theta[mrad] = \frac{w[mrad/s]}{T[s]} \quad (B.2)$$

Os movimentos descritos até esta altura, são baseados num controlo em velocidade, e podem ser substituídos ou parados a qualquer momento por outro movimento com outros parâmetros. A transição entre os movimentos é controlada pelo sistema de multiprocessamento, assegurando um movimento global suave do robot.

As seguintes linhas são exemplos de comandos que podem ser enviados à plataforma móvel, permitindo executar um controlo em velocidade:



Figura B.2: O sistema de navegação autónomo com o sistema de visão activa, para desenvolvimento de tarefas baseadas em informação visual.

1. usando um movimento composto: "MOTV LA V=100 W=100 T=50";
2. usando um movimento de rotação pura: "MOTV LA V=0 W=100 T=50";
3. usando um movimento linear: "MOTV LA V=100 T=50".

Outro tipo de movimentos possíveis, são os baseados no controlo da posição da plataforma móvel. Neste caso, os parâmetros especificados, definem a distância que cada uma das rodas motrizes devem percorrer, e o tempo em que elas devem executar essa tarefa (em unidades de 40ms)

O próximo comando exemplifica uma das formas deste controlo posicional, executando neste caso um movimento de rotação em torno do ponto *RC*:

1. "MOVE P RC=-200,200 P=100".

Na Fig B.2 encontra-se representado o sistema físico.

Apêndice C

Cinemática dos sistemas autónomos utilizados

Nesta secção, é desenvolvida a cinemática de posição/velocidade directa/inversa, para determinar a posição e orientação dum sistema de visão activa montado num robot móvel. O objectivo principal, é a obtenção das expressões do fluxo induzido a nível das imagens de uma câmara sob a influência dos movimentos dos graus de liberdade disponíveis.

C.1 Introdução

Em termos de análise da cinemática, pode-se considerar o robot como sendo constituído por corpos rígidos ligados entre si (a esta ligação também se chama de junta ou junção), que possuem capacidade de se movimentarem. Para descrever as relações de translação e rotação entre ligações adjacentes, Denavit, e Hartenberg (DH), propuseram um método que automaticamente estabelece a matriz de relação entre dois referenciais duma cadeia de referenciais [K.S.Fu 87]. Para um maior esclarecimento do que se segue, suponha-se que se possui $n + 1$ referenciais, numerados de 0 a n , e represente-se um referencial genérico como o i -ésimo referencial.

C.1.1 Transformações entre referenciais

Com este método, cada sistema de coordenadas pode ser estabelecido e determinado através de três regras básicas:

1. O eixo Z_{i-1} pertence ao eixo de movimento da i -ésima junção.
2. O eixo X_i é normal ao eixo Z_{i-1} , e o seu sentido aponta do eixo Z_{i-1} , para fora.
3. O eixo Y_i completa o referencial através da regra da mão direita.

A representação de DH dos referenciais associados aos corpos rígidos, dependem de quatro parâmetros associados a cada junta. Estes quatro parâmetros descrevem completamente cada junta, seja ela prismática (translaciona na direcção do eixo Z) ou de revolução (apenas roda em torno do eixo Z), e são definidos da seguinte forma:

1. θ_i é o ângulo da junção do eixo X_{i-1} para o eixo X_i , medido a partir do eixo Z_{i-1} (utilizando a regra da mão direita).
2. d_i é a distância medida da origem do $(i \Leftrightarrow 1)$ -ésimo sistema de coordenadas ao longo do eixo Z_{i-1} , até à intercepção dos eixos Z_{i-1} e X_i .
3. a_i é a distância desde o ponto de intercepção dos eixos Z_{i-1} e X_i até à origem do i -ésimo referencial medida ao longo do eixo X_i (ou a distância mais curta entre os eixos Z_{i-1} e Z_i).
4. α_i é o ângulo medido do eixo Z_{i-1} , para o eixo Z_i em torno do eixo X_i (usando a regra da mão direita).

A partir do momento, em que cada referencial se encontra estabelecido, obedecendo às regras do método DH, pode-se usar a matriz homogénea de transformação entre dois referenciais consecutivos, por exemplo, o i -ésimo e o $(i-1)$ -ésimo referencial. A matriz que relaciona os dois referenciais é dada por:

$${}^{i-1}A_i = \begin{bmatrix} \cos \theta_i & \Leftrightarrow \cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & \Leftrightarrow \sin \alpha_i \cos \theta_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{C.1})$$

onde α_i , a_i , d_i são constantes, enquanto θ_i é uma variável associada às junções de revolução.

No caso da junção ser prismática, a variável da junção é d_i , enquanto α_i , a_i e θ_i são constantes.

A transformação inversa pode ser calculada através de:

$${}^iA_{i-1} = \begin{bmatrix} \cos \theta_i & \sin \theta_i & 0 & \Leftrightarrow a_i \\ \Leftrightarrow \cos \alpha_i \sin \theta_i & \cos \alpha_i \cos \theta_i & \sin \alpha_i & \Leftrightarrow d_i \sin \alpha_i \\ \sin \alpha_i \sin \theta_i & \Leftrightarrow \sin \alpha_i \cos \theta_i & \cos \alpha_i & \Leftrightarrow d_i \cos \alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{C.2})$$

A relação entre o referencial final, e o primeiro referencial, pode assim ser dada através da relação da matriz de cinemática directa:

$${}^0A_n = {}^0A_1 \cdot {}^1A_2 \cdot \dots \cdot {}^{n-1}A_n \quad (\text{C.3})$$

e para a transformação inversa:

$${}^nA_0 = {}^nA_{n-1} \cdot \dots \cdot {}^2A_1 \cdot {}^1A_0 \quad (\text{C.4})$$

C.1.2 Relações de velocidade

Nesta secção é apresentada a forma de relacionar as velocidades associadas a cada referencial, de modo a poder calcular a velocidade num determinado ponto dum referencial.

Aproximação geométrica

Na representação de DH, a matriz que relaciona os referenciais 0-ésimo e i -ésimo, pode ser escrita da seguinte forma:

$${}^0A_i = \begin{bmatrix} {}^0R_i & {}^0d_i \\ 0 & 1 \end{bmatrix} \quad (\text{C.5})$$

onde 0d_i representa a relação de deslocamento do referencial i -ésimo, e 0R_i representa a sua orientação.

A aproximação geométrica, é baseada no produto vectorial (veja-se a secção C.4), i.e., se um corpo associado com um referencial, se movimenta com uma velocidade angular $\vec{\omega}_i$ e uma velocidade linear \vec{v}_i , um ponto $\vec{P}_i = [X, Y, Z]^T$, no i -ésimo referencial terá uma velocidade

$$\dot{\vec{P}}_i = \vec{\omega}_i \times \vec{P}_i + \vec{v}_i \quad (\text{C.6})$$

esta velocidade pode ser expressa no 0-ésimo referencial através da matriz de orientação 0R_i :

$$\dot{\vec{P}}_0 = {}^0R_i \cdot \dot{\vec{P}}_i = {}^0R_i \cdot (\vec{\omega}_i \times \vec{P}_i + \vec{v}_i) \quad (\text{C.7})$$

contudo, notando que no caso em que R é uma matriz ortogonal, verifica-se a seguinte relação,

$$R(\vec{a} \times \vec{b}) = (R\vec{a}) \times (R\vec{b}), \quad (\text{C.8})$$

e porque, na representação de DH, a rotação é efectuada em torno do eixo Z_i , e a translação também é segundo a direcção do eixo Z_i , a velocidade $\dot{\vec{P}}_i$ pode ser expressa no 0-ésimo referencial como:

$$\dot{\vec{P}}_0 = \|\vec{\omega}_i\| \cdot ({}^0R_i \cdot [0, 0, 1]^T) \times ({}^0R_i \cdot P_i) + \|\vec{v}_i\| {}^0R_i \cdot [0, 0, 1]^T \quad (\text{C.9})$$

mas ${}^0R_i [0, 0, 1]^T$ representa o eixo Z_i no 0-ésimo referencial, e ${}^0R_i \cdot P_i$ pode ser expresso como ${}^0A_i \cdot [P_i^T, 1]^T \Leftrightarrow {}^0d_i$.

No caso de um ponto genérico pertencente ao n -ésimo referencial, a influência das velocidades associadas ao i -ésimo referencial na velocidade do ponto genérico é dada por (expressa no 0-ésimo referencial):

$$\dot{\vec{P}}_0 = \|\vec{\omega}_i\| \cdot (Z_i \times ({}^0A_n \cdot [P_n^T, 1]^T \Leftrightarrow {}^0d_i)) + \|\vec{v}_i\| Z_i \quad (\text{C.10})$$

esta relação torna-se mais simples se o ponto P for $[0, 0, 0]^T$,

$$\dot{\vec{P}}_0 = \|\vec{\omega}_i\| \cdot (Z_i \times ({}^0d_n \Leftrightarrow {}^0d_i)) + \|\vec{v}_i\| Z_i \quad (\text{C.11})$$

Podem também ser necessário saber a velocidade angular resultante, devido às várias velocidades angulares associadas aos referenciais. As velocidades angulares podem ser

somadas se forem expressas relativamente ao mesmo referencial. Descrevendo todas as velocidades no 0-ésimo referencial e definindo a sua soma como ω_0^{all} , então teremos:

$$\vec{\omega}_0^{all} = \vec{\omega}_0 + {}^0 R_1 \cdot \vec{\omega}_1 + \dots + {}^0 R_n \cdot \vec{\omega}_n \quad (C.12)$$

mas como estamos a usar a representação DH, esta relação pode ser descrita por

$$\vec{\omega}_0^{all} = \vec{\omega}_0 + Z_1 \parallel \vec{\omega}_1 \parallel + \dots + Z_n \parallel \vec{\omega}_n \parallel \quad (C.13)$$

tal como para a velocidade linear \vec{v}_i .

Suponha-se agora, que se pretende saber a velocidade linear \vec{v} e a velocidade angular $\vec{\omega}$ de qualquer ponto do n -ésimo referencial, relativamente ao 0-ésimo referencial em função das velocidades angulares de cada referencial. Basta para isso somar as velocidades lineares e angulares, descritas todas num mesmo referencial, por exemplo, o 0-ésimo referencial (${}^0\vec{\omega}_i, {}^0\vec{v}_i$). Se a i -ésima junta é de revolução, então a sua influência é dada por

$$J_i = \begin{bmatrix} \begin{bmatrix} {}^0\vec{v}_i \\ {}^0\vec{\omega}_i \end{bmatrix} = J_i \cdot \omega_i \\ Z_i \times ({}^0 A_n \cdot [P_n^T, 1]^T \Leftrightarrow {}^0 d_i) \\ Z_i \end{bmatrix} \quad (C.14)$$

se a junta for prismática então:

$$J_i = \begin{bmatrix} \begin{bmatrix} {}^0\vec{v}_i \\ {}^0\vec{\omega}_i \end{bmatrix} = J_i \cdot v_i \\ Z_i \\ 0 \end{bmatrix} \quad (C.15)$$

Adicionando todas as influências e denotando q_i como ω_i ou v_i se a junta for de revolução ou prismática respectivamente, então

$$\begin{bmatrix} \vec{v} \\ \vec{\omega} \end{bmatrix} = J \cdot \begin{bmatrix} q_0 \\ \vdots \\ q_i \end{bmatrix} \quad (C.16)$$

onde J é a matriz jacobiana, dada por:

$$J = \begin{bmatrix} J_0 & \dots & J_n \end{bmatrix} \quad (C.17)$$

Aproximação analítica

As relações de velocidade, podem ser obtidas analiticamente, através da definição da derivada, i.e., pode-se estimar $\dot{X}, \dot{Y}, \dot{Z}$ de qualquer ponto, referenciado a um referencial, se se souber o incremento infinitesimal na posição desse ponto. Em termos matemáticos, definindo as funções:

$$\begin{aligned}\vec{P}_n(t=0) &= [X, Y, Z, 1]; \\ \vec{P}_0(t=0) &= {}^0 A_n(\theta_0, \dots, \theta_i, d_0, \dots, d_j) \cdot \vec{P}_n(t=0) \\ \vec{P}_0(t=\Delta t) &= {}^0 A_n(\theta_0 + \Delta t\omega_0, \dots, \theta_i + \Delta t\omega_i, d_0 + \Delta tv_0, \dots, d_j + \Delta tv_j) \cdot \vec{P}_n(t=0)\end{aligned}\tag{C.18}$$

apenas é necessário usar a definição da derivada para calcular a velocidade:

$$\dot{\vec{P}}_0(t=0) = \lim_{\Delta t \rightarrow 0} \frac{\vec{P}_0(t=\Delta t) \ominus \vec{P}_0(t=0)}{\Delta t}\tag{C.19}$$

A velocidade angular poderia também ser calculada, e o resultado final é idêntico ao expresso em C.13.

C.2 Cinemática (directa e inversa)

Nesta secção, apresenta-se dois exemplos, de acordo com os sistemas usados no desenvolvimento dos trabalhos apresentados nesta tese. Relembrando as regras explicadas na secção C.1.1, a representação de DH, enquanto sistemática, permite ainda assim algum grau de liberdade na escolha de alguns parâmetros.

C.2.1 Caso do sistema de visão activa

A primeira configuração usa um sistema de visão activa, num robot móvel. Os referenciais encontram-se representados na Fig. C.1. O Robot móvel pode efectuar movimentos de translação e rotação com as respectivas velocidades v_r e ω_r . O sistema de visão activa possui quatro graus de liberdade, i.e., Pan, Tilt e vergência com velocidades angulares respectivas ω_n , ω_t e ω_{cr} e ω_{cl} .

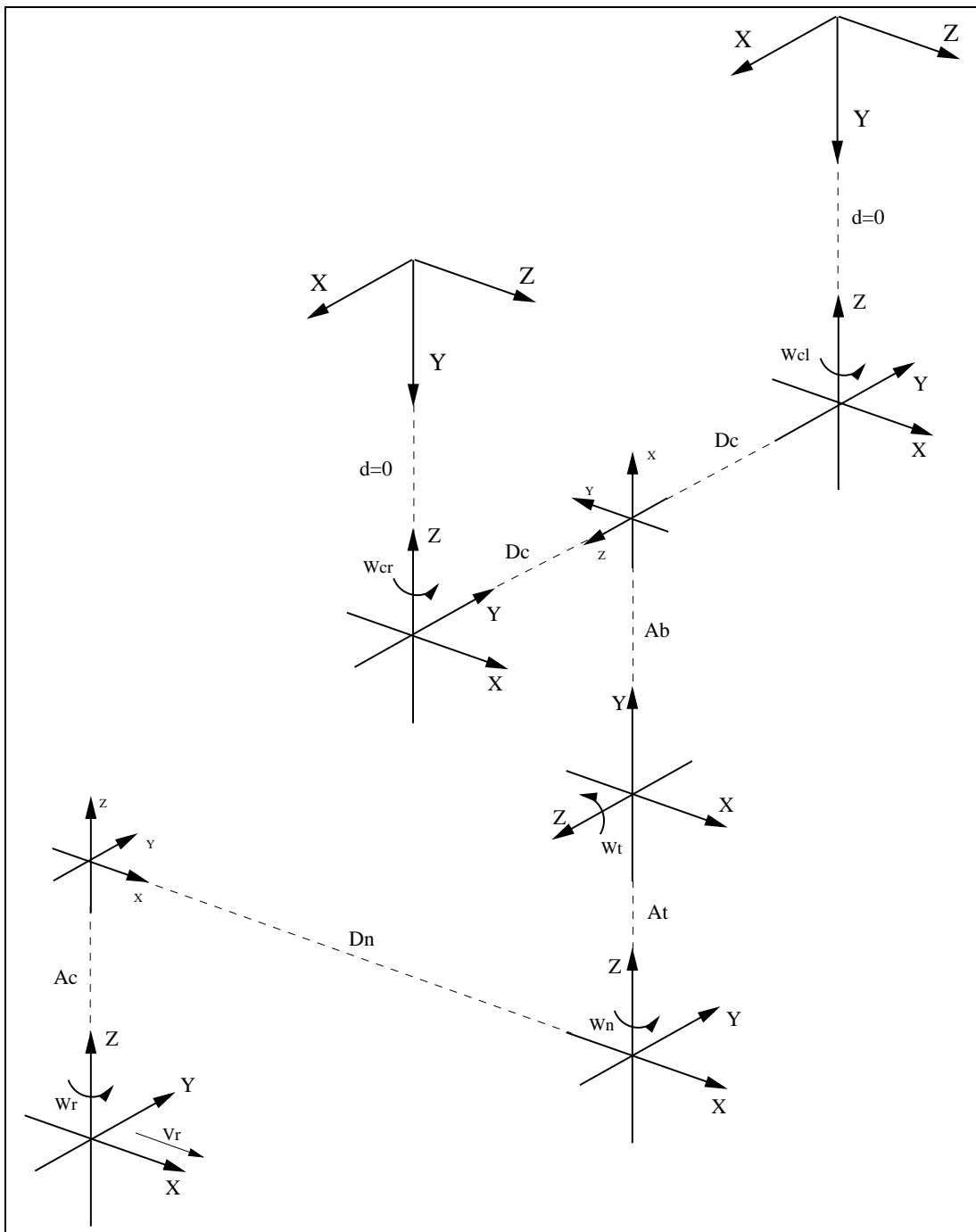


Figura C.1: Referenciais do sistema de visão activa e do robot móvel, de acordo com a representação de DH.

Os parâmetros do método de Denavit e Hartenberg para a câmara da esquerda são assim dados por (desde os referenciais de baixo para os de cima):

$$\begin{bmatrix} 0 & 0 & Ac & 0 & 0 & \omega r & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & Vr \\ 0 & 0 & 0 & Dn & 0 & 0 & 0 \\ 0 & 90^\circ & At & 0 & \theta n & \omega_n & 0 \\ 90^\circ & 0 & 0 & Ab & \theta t & \omega t & 0 \\ \Leftrightarrow 90^\circ & \Leftrightarrow 90^\circ & \Leftrightarrow Dc & 0 & 0 & 0 & 0 \\ \Leftrightarrow 90^\circ & \Leftrightarrow 90^\circ & 0 & 0 & \theta cl & \omega cl & 0 \end{bmatrix} \quad (\text{C.20})$$

onde as colunas representam respectivamente:

1. θ_i
2. α_i
3. a_i
4. d_i
5. variável da posição angular (relacionada com o $(i \Leftrightarrow 1)$ -ésimo ref.)
6. variável da velocidade angular (relacionada com o $(i \Leftrightarrow 1)$ -ésimo ref.)
7. variável da velocidade translacional (relacionada com o $(i \Leftrightarrow 1)$ -ésimo ref.)

e para a câmara da direita:

$$\begin{bmatrix} 0 & 0 & Ac & 0 & 0 & \omega r & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & Vr \\ 0 & 0 & 0 & Dn & 0 & 0 & 0 \\ 0 & 90^\circ & At & 0 & \theta n & \omega_n & 0 \\ 90^\circ & 0 & 0 & Ab & \theta t & \omega t & 0 \\ \Leftrightarrow 90^\circ & \Leftrightarrow 90^\circ & Dc & 0 & 0 & 0 & 0 \\ \Leftrightarrow 90^\circ & \Leftrightarrow 90^\circ & 0 & 0 & \theta cr & \omega cr & 0 \end{bmatrix} \quad (\text{C.21})$$

As matrizes para os referenciais da câmara da direita são dados neste caso por:

$${}^0A_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & Ac \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{C.22})$$

$${}^1A_2 = \begin{bmatrix} 1 & 0 & 0 & Dn \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{C.23})$$

$${}^2A_3 = \begin{bmatrix} \cos(\theta n) & 0 & \sin(\theta n) & 0 \\ \sin(\theta n) & 0 & \Leftrightarrow \cos(\theta n) & 0 \\ 0 & 1 & 0 & At \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{C.24})$$

$${}^3A_4 = \begin{bmatrix} \Leftrightarrow \sin(\theta t) & \Leftrightarrow \cos(\theta t) & 0 & \Leftrightarrow Ab \sin(\theta t) \\ \cos(Tt) & \Leftrightarrow \sin(Tt) & 0 & Ab \cos(\theta t) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{C.25})$$

$${}^4A_5 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ \Leftrightarrow 1 & 0 & 0 & 0 \\ 0 & \Leftrightarrow 1 & 0 & Dc \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{C.26})$$

$${}^5A_6 = \begin{bmatrix} \sin(\theta cr) & 0 & \cos(\theta cr) & 0 \\ \Leftrightarrow \cos(\theta cr) & 0 & \sin(\theta cr) & 0 \\ 0 & \Leftrightarrow 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{C.27})$$

Cinemática directa

Usando a propriedade C.3 pode-se encontrar a seguinte relação (com $C = \cos$ e $S = \sin$):

$$\begin{bmatrix} C(\theta_n)C(\theta_t)S(\theta_{cr}) + S(\theta_n)C(\theta_{cr}) & C(\theta_n)S(\theta_t) & C(\theta_n)C(\theta_t)C(\theta_{cr}) - S(\theta_n)S(\theta_{cr}) & S(\theta_n)Dc - C(\theta_n)Ab S(\theta_t) + Dn \\ S(\theta_n)C(\theta_t)S(\theta_{cr}) - C(\theta_n)C(\theta_{cr}) & S(\theta_n)S(\theta_t) & S(\theta_n)C(\theta_t)C(\theta_{cr}) + C(\theta_n)S(\theta_{cr}) & -C(\theta_n)Dc - S(\theta_n)Ab S(\theta_t) \\ S(\theta_t)S(\theta_{cr}) & -C(\theta_t) & S(\theta_t)C(\theta_{cr}) & Ab C(\theta_t) + At + Ac \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{C.28})$$

Cinemática inversa

Usando a propriedade C.4 conclui-se que:

$${}^6A_0 = [{}^0A_6]^{-1} \quad (\text{C.29})$$

Relações de velocidade

Para este exemplo, não se apresenta o resultado, devido a matriz jacobiana possuir demasiados elementos. No entanto, tal como explicado na secção C.3, tanto o método geométrico como o método analítico dão o mesmo resultado. Os resultados podem ser verificados através do programa listado na página de Web <http://www.isr.uc.pt/~sousa/ftp.html>.

C.2.2 Caso dos Pan & Tilt

A segunda configuração, usa duas unidades de Pan&Tilt, com um robot móvel. Os referenciais encontram-se representados na Fig. C.1. O Robot móvel pode efectuar movimentos de translação e rotação com as respectivas velocidades v_r e ω_r . Cada unidade de Pan&Tilt possui dois graus de liberdade, i.e., Pan, Tilt com velocidades angulares respectivas ω_{pr} , ω_{tr} ou ω_{pl} , ω_{tl} .

Os parâmetros do método de Denavit e Hartenberg para a câmara da esquerda são

assim dados por (desde os referenciais de baixo para os de cima):

$$\begin{bmatrix} 0 & 0 & Ac & 0 & 0 & \omega r & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & Vr \\ 0 & 0 & 0 & Dn & 0 & 0 & 0 \\ 0 & 90 & Ap & 0 & 0 & 0 & 0 \\ 0 & \Leftrightarrow 90 & \Leftrightarrow Dc & 0 & 0 & 0 & 0 \\ 0 & 90 & At & 0 & \theta pl & \omega pl & 0 \\ 90 & 90 & 0 & Ai & \theta tl & \omega tl & 0 \\ 90 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (C.30)$$

e para a câmara direita:

$$\begin{bmatrix} 0 & 0 & Ac & 0 & 0 & \omega r & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & Vr \\ 0 & 0 & 0 & Dn & 0 & 0 & 0 \\ 0 & 90 & Ap & 0 & 0 & 0 & 0 \\ 0 & \Leftrightarrow 90 & Dc & 0 & 0 & 0 & 0 \\ 0 & 90 & At & 0 & \theta pr & \omega pr & 0 \\ 90 & 90 & 0 & Ai & \theta tr & \omega tr & 0 \\ 90 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (C.31)$$

As matrizes para os referenciais da câmara da esquerda são dados neste caso por:

$${}^0A_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & Ac \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (C.32)$$

$${}^1A_2 = \begin{bmatrix} 1 & 0 & 0 & Dn \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (C.33)$$

$${}^2A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & \Leftrightarrow 1 & 0 \\ 0 & 1 & 0 & Ap \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (C.34)$$

$${}^3A_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & \Leftrightarrow 1 & 0 & \Leftrightarrow Dc \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (C.35)$$

$${}^4A_5 = \begin{bmatrix} \cos(\theta pl) & 0 & \sin(\theta pl) & 0 \\ \sin(\theta pl) & 0 & \Leftrightarrow \cos(\theta pl) & 0 \\ 0 & 1 & 0 & At \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (C.36)$$

$${}^5A_6 = \begin{bmatrix} \Leftrightarrow \sin(\theta tl) & 0 & \cos(\theta tl) & \Leftrightarrow Ai \sin(\theta tl) \\ \cos(\theta tl) & 0 & \sin(\theta tl) & Ai \cos(\theta tl) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (C.37)$$

$${}^6A_7 = \begin{bmatrix} 0 & \Leftrightarrow 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (C.38)$$

Cinemática directa

Usando a propriedade C.3 pode-se encontrar a seguinte relação:

$${}^0A_6 = \begin{bmatrix} \sin(\theta pl) & \cos(\theta pl) \sin(\theta tl) & \cos(\theta pl) \cos(\theta tl) & \Leftrightarrow \cos(\theta pl) Ai \sin(\theta tl) + Dn \\ \Leftrightarrow \cos(\theta pl) & \sin(\theta pl) \sin(\theta tl) & \sin(\theta pl) \cos(\theta tl) & \Leftrightarrow \sin(\theta pl) Ai \sin(\theta tl) + Dc \\ 0 & \Leftrightarrow \cos(\theta tl) & \sin(\theta tl) & Ai \cos(\theta tl) + At + Ap + Ac \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (C.39)$$

Cinemática inversa

Usando a propriedade C.4 conclui-se que (com $C = \cos$ e $S = \sin$):

$${}^6A_0 = \begin{bmatrix} S(\theta_{pl}) & -C(\theta_{pl}) & 0 & -S(\theta_{pl})D_n + C(\theta_{pl})D_c \\ C(\theta_{pl})S(\theta_{tl}) & S(\theta_{pl})S(\theta_{tl}) & -C(\theta_{tl}) & -S(\theta_{tl})C(\theta_{pl})D_n - S(\theta_{tl})S(\theta_{pl})D_c + C(\theta_{tl})A_t + C(\theta_{tl})A_p + C(\theta_{tl})A_c + A_i \\ C(\theta_{pl})C(\theta_{tl}) & S(\theta_{pl})C(\theta_{tl}) & S(\theta_{tl}) & -C(\theta_{tl})C(\theta_{pl})D_n - C(\theta_{tl})S(\theta_{pl})D_c - S(\theta_{tl})A_t - S(\theta_{tl})A_p - S(\theta_{tl})A_c \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (C.40)$$

Relações de velocidade

Usando o programa de Maple referido, a relação directa da velocidade para o pan&tilt da direita, é dada por (para um ponto $P = [0, 0, 0]^T$ no i -ésimo referencial):

$$\begin{bmatrix} \sin(Tpr)Ai \sin(Ttr) + Dc & 0 & \sin(Tpr)Ai \sin(Ttr) & \Leftrightarrow 1.0 \cos(Tpr)Ai \cos(Ttr) \\ \Leftrightarrow 1.0 \cos(Tpr)Ai \sin(Ttr) + Dn & 0 & \Leftrightarrow 1.0 \cos(Tpr)Ai \sin(Ttr) & \Leftrightarrow 1.0 \sin(Tpr)Ai \cos(Ttr) \\ 0 & 1 & 0 & \Leftrightarrow 1.0 Ai \sin(Ttr) \\ 0 & 0 & 0 & \sin(Tpr) \\ 0 & 0 & 0 & \Leftrightarrow \cos(Tpr) \\ 1 & 0 & 1 & 0 \end{bmatrix} \quad (C.41)$$

as variáveis de controlo das junções são dadas pelo vector

$$[\omega_r, Vr, \omega_{pr}, \omega_{tr}]^T. \quad (C.42)$$

Multiplicando a matriz pelo vector obtemos o resultado no seguinte formato vectorial

$$[v_x, v_y, v_z, \omega_x, \omega_y, \omega_z]^T. \quad (C.43)$$

A relação inversa da velocidade pode ser calculada através do programa em maple.

C.3 Matrizes “skew symmetric”

Uma matriz S diz-se que é uma matriz “*skew symmetric*” se e se só

$$S^T + S = 0 \quad (C.44)$$

ou por outras palavras $s_{ij} + s_{ji} = 0$ e $s_{ii} = 0$. Para o caso de matrizes de 3×3 a matriz skew simétrica tem a forma

$$S = \begin{bmatrix} 0 & \Leftrightarrow s_1 & s_2 \\ s_1 & 0 & \Leftrightarrow s_3 \\ \Leftrightarrow s_2 & s_3 & 0 \end{bmatrix} \quad (\text{C.45})$$

Se $\vec{a} = [a_x, a_y, a_z]^T$ é um vector, a matriz skew simétrica pode ser definida como $S(\vec{a})$:

$$S(\vec{a}) = \begin{bmatrix} 0 & \Leftrightarrow a_z & a_y \\ a_z & 0 & \Leftrightarrow a_x \\ \Leftrightarrow a_y & a_x & 0 \end{bmatrix} \quad (\text{C.46})$$

Uma das propriedades importante da matriz $S(\vec{a})$ é a linearidade. Pelo que para quaisquer vectores \vec{a} e \vec{b} pertencentes a \mathfrak{R}^3 e escalares α e β têm-se:

$$S(\alpha\vec{a} + \beta\vec{b}) = \alpha S(\vec{a}) + \beta S(\vec{b}) \quad (\text{C.47})$$

Outra propriedade importante da matriz $S(\vec{a})$ é que para qualquer vector $\vec{p} = [p_x, p_y, p_z]^T$:

$$S(\vec{a})\vec{p} = \vec{a} \times \vec{p} \quad (\text{C.48})$$

propriedade esta que pode ser verificada por cálculos directos.

Outro facto importante, é que se $R_{3 \times 3}$ é uma matriz ortogonal, então usando a relação anterior teremos:

$$RS(\vec{a})R^T = S(R\vec{a}) \quad (\text{C.49})$$

Considere-se que uma matriz de rotação R é função de uma única variável θ . Como R é ortogonal para qualquer valor de θ então:

$$R(\theta).R(\theta)^T = I \quad (\text{C.50})$$

derivando ambos os membros da equação C.50 em ordem a θ e usando a regra do produto temos

$$\frac{dR(\theta)}{d\theta}.R(\theta)^T + R(\theta)\frac{dR^T(\theta)}{d\theta} = 0 \quad (\text{C.51})$$

definindo a matriz S como

$$S = \frac{dR(\theta)}{d\theta} \cdot R(\theta)^T \quad (\text{C.52})$$

pode-se verificar que a equação C.51 é idêntica à equação C.44, i.e., a matriz S é uma matriz skew simétrica. Multiplicando à direita ambos os termos da equação C.52 por $R(\theta)$ e usando a eq. C.50 temos

$$\frac{dR(\theta)}{d\theta} = S \cdot R(\theta). \quad (\text{C.53})$$

Pode se verificar facilmente através de cálculos que,

$$\frac{dRx(\theta_x)}{d\theta_x} = S(\hat{x}) \cdot Rx(\theta_x) \quad \frac{dRy(\theta_y)}{d\theta_y} = S(\hat{y}) \cdot Ry(\theta_y) \quad \frac{dRz(\theta_z)}{d\theta_z} = S(\hat{z}) \cdot Rz(\theta_z) \quad (\text{C.54})$$

Suponha-se agora, que temos uma matriz de rotação, relacionando dois referenciais, e que se vai efectuar uma rotação que pode ser decomposta em três rotações em torno de cada um dos eixos do referencial, i.e.:

$${}^0R_i(\delta t) = {}^0R_i(0) \cdot R_{\theta_x} \cdot R_{\theta_y} \cdot R_{\theta_z} \quad (\text{C.55})$$

onde θ_x , θ_y , θ_z são ângulos infinitesimais, e a matriz está definida num sistema directo, com os ângulos obedecendo à regra da mão direita, derivando em ordem a t obtemos,

$$\frac{d{}^0R_i(\delta t)}{dt} = {}^0R_i(0) \cdot \frac{\partial R_{\theta_x}}{\partial \theta_x} \cdot R_{\theta_y} \cdot R_{\theta_z} \cdot \frac{\partial \theta_x}{\partial t} + {}^0R_i(0) \cdot R_{\theta_x} \cdot \frac{\partial R_{\theta_y}}{\partial \theta_y} \cdot R_{\theta_z} \cdot \frac{\partial \theta_y}{\partial t} + {}^0R_i(0) \cdot R_{\theta_x} \cdot R_{\theta_y} \cdot \frac{\partial R_{\theta_z}}{\partial \theta_z} \cdot \frac{\partial \theta_z}{\partial t} \quad (\text{C.56})$$

mas ${}^0R_i(\delta t)$ é uma matriz ortogonal, pelo que se verifica a eq. C.50, i.e.,

$${}^0R_i(0) \cdot R_{\theta_x} \cdot R_{\theta_y} \cdot R_{\theta_z} \cdot R_{\theta_z}^T \cdot R_{\theta_y}^T \cdot R_{\theta_x}^T \cdot {}^0R_i(0)^T = I \quad (\text{C.57})$$

derivando esta eq. em ordem a ω_x , obtemos uma matriz skew dada por

$$S = {}^0R_i(0) \cdot \frac{\partial R_{\theta_x}}{\partial \theta_x} \cdot R_{\theta_x}^T \cdot {}^0R_i(0)^T = {}^0R_i(0) \cdot S(\hat{x}) \cdot {}^0R_i(0)^T = S({}^0R_i(0) \cdot \hat{x}) \quad (\text{C.58})$$

e pode facilmente verificar-se que

$$S({}^0R_i(0) \cdot \hat{x}) \cdot R_{\theta_x} \cdot R_{\theta_y} \cdot R_{\theta_z} = {}^0R_i(0) \cdot \frac{\partial R_{\theta_x}}{\partial \theta_x} \cdot R_{\theta_y} \cdot R_{\theta_z} \quad (\text{C.59})$$

derivando a eq. C.57 em ordem a ω_y , obtemos uma matriz skew dada por

$$S = {}^0 R_i(0) \cdot R_{\theta_x} \cdot \frac{\partial R_{\theta_y}}{\partial \theta_y} \cdot R_{\theta_y}^T \cdot R_{\theta_x}^T \cdot {}^0 R_i(0)^T = {}^0 R_i(0) \cdot R_{\theta_x} \cdot S(\hat{y}) \cdot R_{\theta_x}^T \cdot {}^0 R_i(0)^T = S \left({}^0 R_i(0) \cdot R_{\theta_x} \cdot \hat{y} \right) \quad (\text{C.60})$$

e pode-se verificar facilmente que

$$S \left({}^0 R_i(0) \cdot R_{\theta_x} \cdot \hat{y} \right) \cdot R_{\theta_x} \cdot R_{\theta_y} \cdot R_{\theta_z} = {}^0 R_i(0) \cdot R_{\theta_x} \cdot \frac{\partial R_{\theta_y}}{\partial \theta_y} \cdot R_{\theta_z} \quad (\text{C.61})$$

e finalmente derivando a eq. C.57 em ordem a ω_z , obtemos uma matriz skew dada por

$$S = {}^0 R_i(0) \cdot R_{\theta_x} \cdot R_{\theta_y} \cdot \frac{\partial R_{\theta_z}}{\partial \theta_z} \cdot R_{\theta_z}^T \cdot R_{\theta_y}^T \cdot R_{\theta_x}^T \cdot {}^0 R_i(0)^T = S \left({}^0 R_i(0) \cdot R_{\theta_x} \cdot R_{\theta_y} \cdot \hat{z} \right) \quad (\text{C.62})$$

e pode-se verificar facilmente que

$$S \left({}^0 R_i(0) \cdot R_{\theta_x} \cdot R_{\theta_y} \cdot \hat{z} \right) \cdot R_{\theta_x} \cdot R_{\theta_y} \cdot R_{\theta_z} = {}^0 R_i(0) \cdot R_{\theta_x} \cdot R_{\theta_y} \cdot \frac{\partial R_{\theta_z}}{\partial \theta_z} \quad (\text{C.63})$$

agora, usando as eq. C.59, C.61 e C.63 na eq. C.56 obtemos,

$$\frac{d {}^0 R_i(\delta t)}{dt} = S \left({}^0 R_i(0) \cdot \left[\dot{\theta}_x \cdot \hat{x} + \dot{\theta}_y \cdot R_{\theta_x} \cdot \hat{y} + \dot{\theta}_z \cdot R_{\theta_x} \cdot R_{\theta_y} \cdot \hat{z} \right] \right) \cdot {}^0 R_i(0) \cdot R_{\theta_x} \cdot R_{\theta_y} \cdot R_{\theta_z} \quad (\text{C.64})$$

onde o factor entre [...] é dado por

$$\left(\dot{\theta}_x + \dot{\theta}_z \cdot \sin(\theta_y) \right) \hat{x} + \left(\dot{\theta}_y \cos(\theta_x) \Leftrightarrow \dot{\theta}_z \cdot \sin(\theta_x) \cos(\theta_y) \right) \hat{y} + \left(\dot{\theta}_y \sin(\theta_x) + \dot{\theta}_z \cdot \cos(\theta_x) \cos(\theta_y) \right) \hat{z} \quad (\text{C.65})$$

mas à medida que $\delta t \rightarrow 0$ também $\theta_x = \theta_y = \theta_z \rightarrow 0$, $R_{\theta_x} = R_{\theta_y} = R_{\theta_z} \rightarrow I$ e finalmente obtém-se

$$\frac{d {}^0 R_i(0)}{dt} = S(\vec{\omega}) \cdot {}^0 R_i(0) \quad (\text{C.66})$$

onde $\vec{\omega} = {}^0 R_i(0) \cdot \left[\dot{\theta}_x, \dot{\theta}_y, \dot{\theta}_z \right]^T = {}^0 R_i(0) \cdot \left[\omega_x, \omega_y, \omega_z \right]^T$.

C.4 A influência da velocidade angular

Supondo que a eq. C.5 é variável no tempo, escrevendo a equação de posição,

$$\vec{p}_0(t) = {}^0 R_i(t) \cdot \vec{p}_i + {}^0 d_i(t) \quad (\text{C.67})$$

derivando a eq. anterior e usando a regra do produto obtemos (note-se que \vec{p}_i é constante):

$$\dot{\vec{p}}_0(t) = {}^0\dot{R}_i(t) \cdot \vec{p}_i + {}^0\dot{d}_i(t) = S(\vec{\omega}) \cdot {}^0R_i(t) \cdot \vec{p}_i + {}^0\dot{d}_i(t) = \vec{\omega} \times ({}^0R_i(t) \cdot \vec{p}_i) + \vec{v} = \vec{\omega} \times \vec{r} + \vec{v} \quad (\text{C.68})$$

onde $\vec{r} = {}^0R_i(t) \cdot \vec{p}_i$ é a posição do vector expressa no 0-ésimo referencial e $\vec{v} = {}^0\dot{d}_i(t)$ é a velocidade a que o i -ésimo referencial se movimenta, e $\vec{\omega} = {}^0R_i \cdot [\omega_x, \omega_y, \omega_z]^T$.

Também é possível obter a função aceleração, notando que o seguinte se verifica,

$$\frac{d(\vec{a} \times \vec{b})}{dt} = \frac{d\vec{a}}{dt} \times \vec{b} + \vec{a} \times \frac{d\vec{b}}{dt} \quad (\text{C.69})$$

derivando a expressão C.68 obtemos,

$$\ddot{\vec{p}}_0(t) = \dot{\omega} \times \vec{r} + \omega \times (\vec{\omega} \times \vec{r}) + \dot{\vec{v}} \quad (\text{C.70})$$

$\dot{\vec{v}}$ é a aceleração linear. O termo $\omega \times (\vec{\omega} \times \vec{r})$ é chamado de aceleração centrípeta do ponto. A direcção do vector é sempre direccionada na direcção do eixo de rotação e é perpendicular a esse eixo. O termo $\dot{\omega} \times \vec{r}$ é chamado da aceleração transversal.

Contudo, num caso mais geral, quando um ponto \vec{p}_i possui velocidade, a relação anterior modifica-se para:

$$\dot{\vec{p}}_0(t) = \vec{\omega} \times \vec{r} + \vec{v} + {}^0R_i(t) \cdot \dot{\vec{p}}_i \quad (\text{C.71})$$

e

$$\ddot{\vec{p}}_0(t) = \dot{\omega} \times \vec{r} + \omega \times (\vec{\omega} \times \vec{r}) + 2\omega \times ({}^0R_i(t) \cdot \dot{\vec{p}}_i) + {}^0R_i(t) \cdot \ddot{\vec{p}}_i + \dot{\vec{v}} \quad (\text{C.72})$$

onde o termo ${}^0R_i(t) \cdot \ddot{\vec{p}}_i$ representa a aceleração linear (a aceleração do ponto). O termo $2\omega \times ({}^0R_i(t) \cdot \dot{\vec{p}}_i)$ é conhecido como a aceleração de Coriolis.

Apêndice D

Pormenores da realização prática

Para realizar a implementação em tempo real, usou-se neste caso uma placa de processamento de imagem (TDMB412), com dois processadores TMS320C40s (ou C40). Apenas um dos processadores C40, pode adquirir imagens, e será designado como o C40 master (TDM435), e o outro C40 será referenciado como o C40 slave (TDM407), tal como pode ser visto na Fig. D.3.

Os processadores TMS320C40 são ideais para o processamento de sinal, em aplicações que requerem processamento massivo, e bastante repetitivo. As aplicações típicas para múltiplos C40 incluem:

1. Processamento de imagem & Visão por computador
2. Processamento de sinais vindos de Radars e sonares
3. Sistemas de telecomunicações & processamento de voz
4. Ruído & Análise de vibração e controlo
5. Controlo em tempo real de alta velocidade & Instrumentação
6. Tratamento de imagens médicas

O processador TMS320C40 possui 6 portos de alta velocidade para efectuar comunicações entre processadores. Estes portos são bidireccionais e possuem uma largura de banda de

20MBytes/segundo. Seis canais de DMA permitem I/O paralelamente às operações do CPU. Um C40 a 50MHz é capaz de efectuar 275 milhões de operações/seg e 50MFLOPS. Os modos de endereçamento incluem o linear, circular e bit-reversed. O processador utiliza também uma unidade de 40/32 bits de floating/fixed point.

A placa de processamento TDMB412 é uma motherboard de módulos TIM-40 para o bus do PC. Esta motherboard possui 4 slots, onde podem ser colocados vários módulos de I/O TIM-40. O interface da motherboard com o PC hospedeiro é garantido através de uma FIFO de 16-bits de barramento. A placa inclui também hardware específico, para permitir efectuar debug do sistema. A placa permite ainda a interligação entre várias motherboards, através de um bus de alta velocidade, podendo assim construir-se várias topologias de rede, baseadas em múltiplos C40.

Os módulos de processamento TIM-40 podem ter um ou mais DSPs C40. Alguns destes módulos são específicos, e permitem realizar funções tais como, input/output digital/analógico, gráficos, e framegrabbing.

A placa TDM435 é um módulo 3 TIM-40 que incorpora um sistema de aquisição de imagem monocromático com uma resolução de 8 bits por pixel, um processador C40, e um sistema de display RGB de 1024 x 1024 x 24-bit com um plano de overlay de 4-bits. O sistema de aquisição permite seleccionar uma de quatro entradas que é guardada num de dois blocos de memória VRAM (memória do sistema de vídeo) de 1K x 1K, podendo ser acedido ao mesmo tempo pelo sistema de aquisição, pelo C40 e pelo sistema de display. São permitidos vários modos de acesso à memória VRAM, acessos de 4 bytes, 1 byte, com máscara, escrita/leitura de blocos, etc, o que dá ao sistema uma elevada flexibilidade. O conector de expansão da memória global, é usado para aceder à memória incorporada na motherboard principal, que é mapeada na zona de memória global do C40, permitindo assim salvar várias imagens consecutivamente. O sistema permite adquirir sinais PAL ou NTSC, e está equipado com vários filtros antialiasing, que podem ser seleccionados por software. O sistema de aquisição permite também programar o endereço inicial, período entre linhas, tamanho da frame e tamanho de uma linha. O sistema de display, permite seleccionar qualquer parte dos 2 blocos de 1K x 1K "pixels" para efectuar o display. O diagrama de blocos encontra-se representado na Fig D.1.

A placa TDM407 é um módulo TIM-40 que incorpora um DSP C40. Este módulo pode

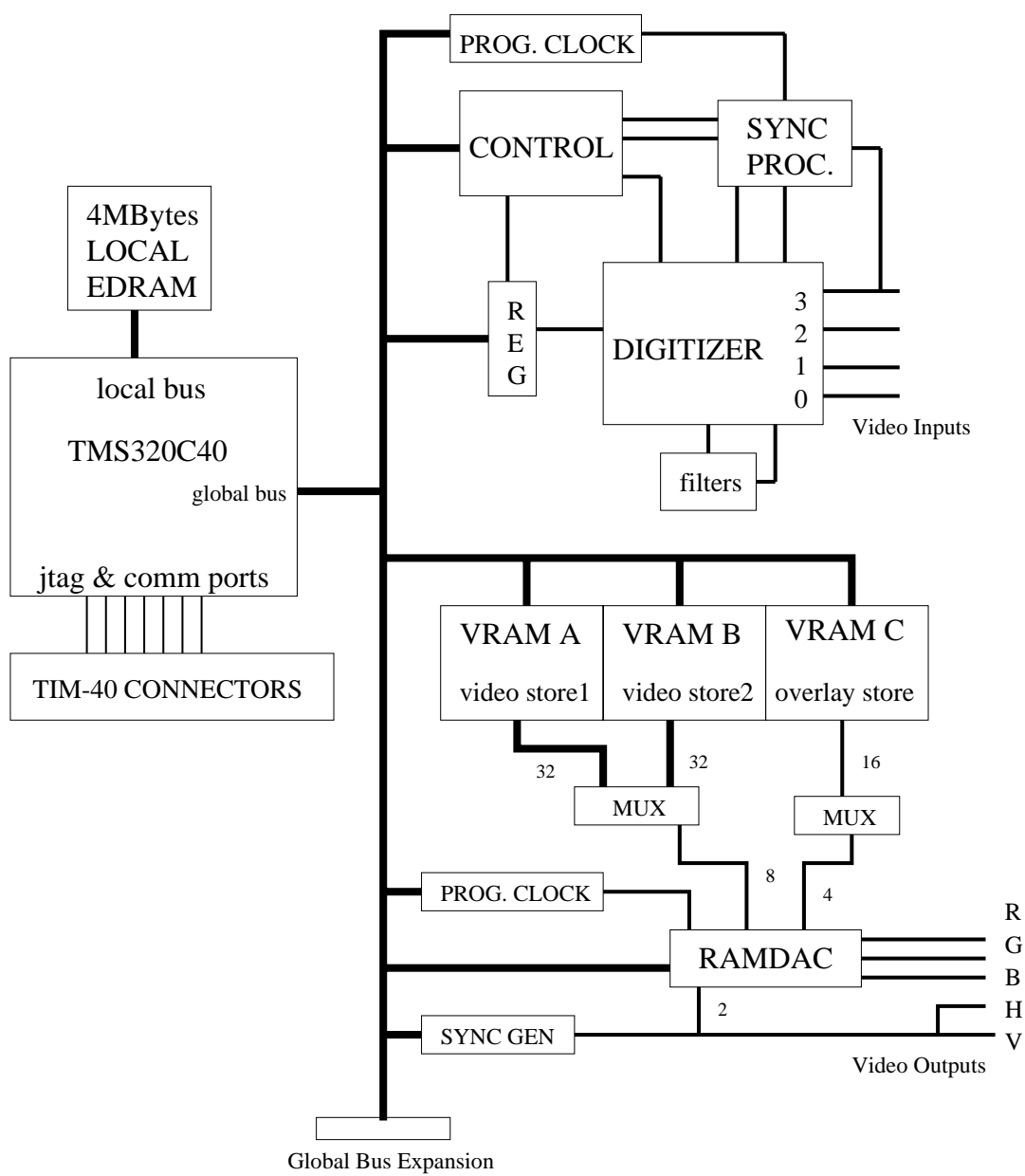


Figura D.1: Diagrama de blocos do TDM435.

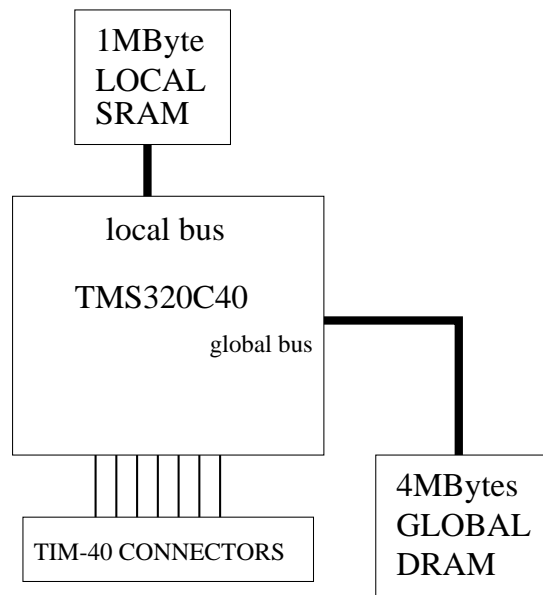


Figura D.2: Diagrama de blocos do TDM407.

incorporar memória DRAM ou SRAM de modo a obter um óptimo, mistura de velocidade e espaço. A combinação de memória pode ser do tipo DRAM/SRAM, DRAM/DRAM, SRAM/SRAM ou SRAM/DRAM mapeadas no global bus e no local bus. Num módulo que incorpore os dois tipos de memória, o código, stack e heap podem ser alocados na memória de 0-ws SRAM até 1MB, enquanto outras alocações de maior escala podem ser feitas na memória global até 4MB. O conector de expansão da memória global, é usado para aceder à memória incorporada na motherboard principal, que é mapeada na zona de memória global do C40. O diagrama de blocos encontra-se representado na Fig. D.2.

O computador hospedeiro, neste caso é um PC, que comunica com o C40 master, através de um canal FIFO, e ambos os C40 podem transferir dados entre si utilizando um canal de DMA existente e por um porto para enviar comandos. O tempo perdido na transferência por DMA é quase nulo, porque o canal tem um alta largura de banda, e ambos os C40 podem enviar dados enquanto estão a efectuar processamento. O Frame grabber pode adquirir imagens a 25 Hz. No entanto como no nosso caso existem duas câmaras, e o formato do vídeo de entrada é PAL interlaçado, para diminuir o tempo de aquisição, apenas se capta um dos campos da imagem para cada câmara, obtendo-se assim

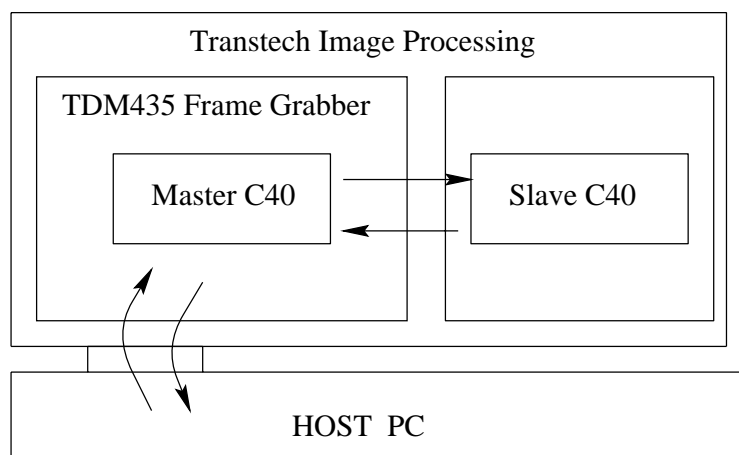


Figura D.3: A placa de processamento de imagem é utilizada em conjunção com o sistema de visão activa, para desenvolver métodos de navegação visuais.

um frame rate de 50 Hz para ambas as câmaras.

Para este processo funcionar perfeitamente, as duas câmaras devem-se encontrar sincronizadas. O tempo de aquisição pode ser paralelizado, sobre o tempo de cálculo do fluxo óptico, uma vez que podem ser definidos vários buffers onde a imagem pode ser guardada, e o C40 master permite efectuar a aquisição através de interrupção, que é activada ao fim de cada campo da imagem. Desta forma é possível garantir que para a mesma câmara, a imagem é sempre adquirida do mesmo campo, pois caso contrário, existiria um shift vertical entre as imagens, o que em termos de fluxo óptico é prejudicial, pois com as câmaras paradas, seria detectado fluxo. O problema desta solução, é que a imagem adquirida apenas é tratada no próximo ciclo, pois para cada câmara existem 2 buffers, um com a última imagem adquirida, e outro para onde a interrupção “agulhou” a aquisição actual. De qualquer forma, tal não é crítico, pois o atraso relativamente à imagem mais actualizada é de $\frac{2}{50}$ s. Para garantir coerência temporal para o filtro de kalman, a interrupção guarda o tempo actual, quando inicia uma nova aquisição. Assim que um novo ciclo de processamento é executado, o C40 master efectua a transformação log-polar e calcula as derivadas temporais, fazendo de seguida o download das matrizes ao C40 slave e recebe os resultados no próximo ciclo para a câmara em questão. O C40 slave calcula então o fluxo óptico de acordo com o tipo de fluxo utilizado, gera os sinais de “feedback”

e efectua o upload para o C40 master no próximo ciclo para a câmara em questão. O C40 master mantém um filtro de kalman para ambas as imagens, efectuando integração temporal, e efectua o upload para o host a uma cadência de 12.5Hz. Por sua vez o host através dum porto série envia os dados a um sistema unix, onde é efectuado o algoritmo de controlo e onde são enviados os comandos ao robot móvel. Devido à implementação efectuada, existe um delay de 4 ciclos ($\frac{4}{50}$) entre a imagem actualmente adquirida e o seu resultado, dois dos delays é devido à aquisição e dois devido às comunicações entre master e slave.

A transformação log-polar é efectuada através de uma tabela, onde são guardadas as posições do plano cartesiano onde devem ser feitas as amostras. Os “pixels” na redondeza do pixel amostrado, são usados para efectuar uma convulsão gaussiana, baseada apenas em shifts, a máscara usada é:

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (\text{D.1})$$

Uma das grandes vantagens do C40 é permitir instruções paralelas (leitura, leitura/escrita, escrita, adicionar/leitura, ...), desta forma em alguns dos processos o tempo de processamento pode ser reduzido a metade ou mais conforme o caso. Outra das vantagens, é a de permitir aceder a 4 “pixels”, apenas com um acesso a memória de 32 bits, desta forma, por exemplo, a implementação da máscara apenas obriga a 3 acessos por cada ponto log-polar amostrado (sem utilizar as instruções paralelas). A transformação log-polar é um dos casos, em que se pode aproveitar ao máximo estas capacidades, a tipo de exemplo, a transformação log-polar codificada em assembler gasta cerca $\frac{1}{4}$ do tempo da versão codificada em c (isto porque o compilador de c não codifica instruções paralelas).

Para efectuar o cálculo do fluxo óptico normal, foi usada também uma tabela. Em cada entrada da tabela, é guardado o resultado da equação:

$$\frac{E_1}{E_1^2 + E_2^2} \quad (\text{D.2})$$

para $E_1, E_2 = 0, \dots, 256$. Desta forma para cada ponto usado é necessário efectuar menos uma multiplicação e menos uma divisão. Nesta implementação, apenas se usa 2 níveis na pirâmide. A imagem de maior resolução tem um tamanho de 110x150 “pixels”. O tempo

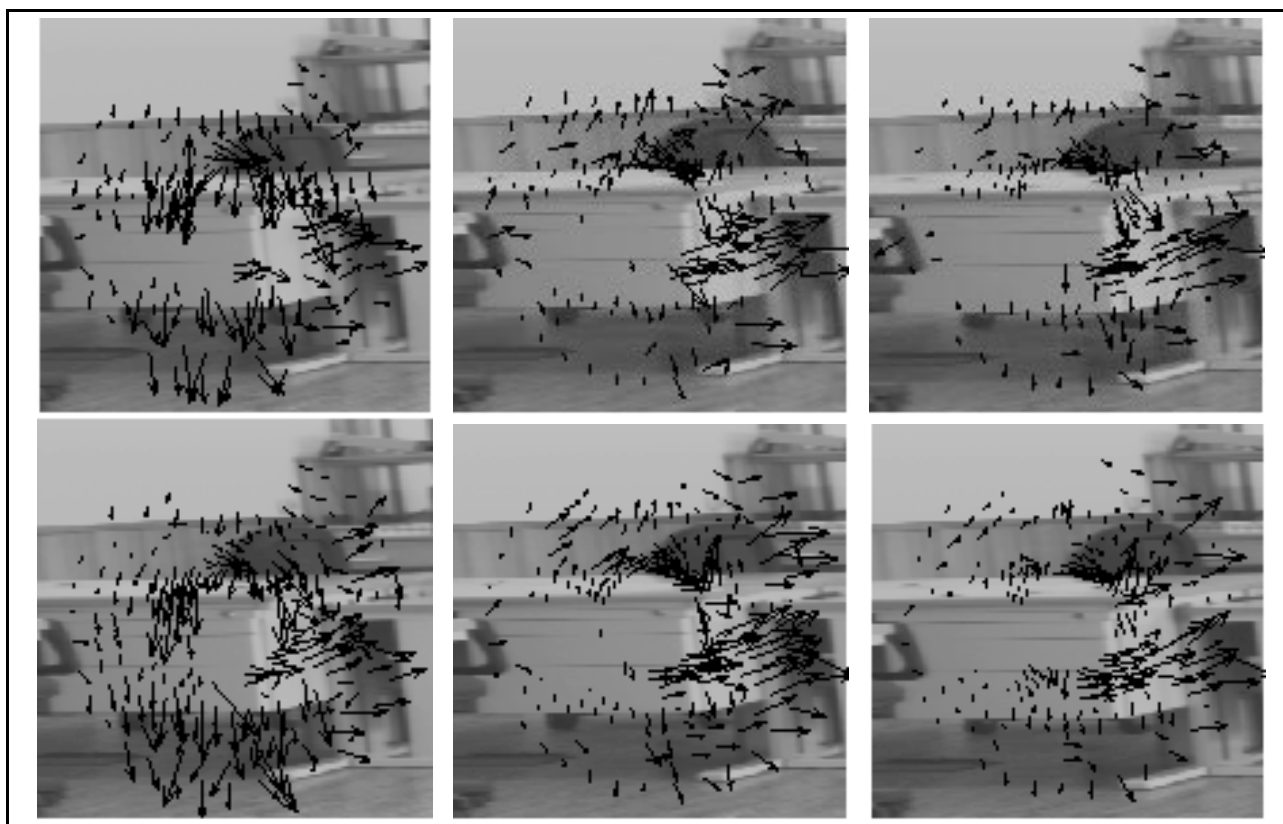


Figura D.4: Cena de teste: Esquerda - movimento de rotação e de translação, e nos outros apenas é efectuado movimento de translação. Em cima: um nível de pirâmide, Em baixo: 2 níveis na pirâmide. O algoritmo é o fluxo óptico normal.

de processamento anda à volta dos 10Hz, uma vez que o filtro α - β torna mais pesado o processamento. A Fig. D.4 mostra os resultados obtidos por este algoritmo.

Quanto ao cálculo do fluxo afim, não merece grandes reparos, a menos que se consegue neste algoritmo obter 25Hz para a mesma dimensão da imagem (sem utilizar pirâmides).

Para o caso do processo de navegação com espelhos, a implementação aproveita ao máximo todas as potencialidades descritas anteriormente, conseguindo-se obter uma taxa de processamento a 50Hz e resultados para ambas as câmaras a 25Hz. A tipo de exemplo, para o histograma vertical contribuem 100 linhas horizontais. O histograma final possui uma resolução horizontal de 384 linhas, isto porque em cada acesso se obtêm 4 “pixels” numa palavra de 32bits, mas apenas se utilizam 2 para permitir efectuar a soma desses

dois “pixels” simultaneamente através duma máscara que zera os outros dois. O C40 master neste caso adquire a imagem duma câmara e efectua os dois histogramas para as imagens dos dois espelhos dessa câmara. De seguida envia os dois histogramas ao C40 slave que efectua os dois processos de correlação para estimar o deslocamento horizontal nas imagens. Ambos os processadores respeitam os 50Hz, no entanto o resultado da aquisição actual, só se encontra disponível com 2 ciclos de atraso.

Em conclusão: a performance dos algoritmos depende claramente da experiência de programação do utilizador. Existe um certo tempo de aprendizagem, até o utilizador tirar partido de todas as potencialidades dos dois C40, uma vez que a programação em assembler deve obedecer à forma como o pipeline do processador funciona de modo a introduzir o mínimo de wait states entre as instruções. Outro tipo de implementação que deve ser evitada, é a utilização de cálculos em floating point, pois as instruções nativas do processador apenas possuem precisão simples, caso seja necessário maior precisão, esta só é possível através das rotinas de emulação do compilador de c, perdendo-se assim o partido das instruções paralelas. Um dos casos que pode exemplificar este problema, prende-se com a estimação do fluxo afim através do método dos mínimos quadrados, que teve de utilizar a emulação, não sendo no entanto neste caso um problema crítico.

Apêndice E

Filtros α - β e α - β - γ

E.1 Introdução

Os métodos de filtragem e predição podem ser usados para a estimação presente ou futura de quantidades cinemáticas, tais como a posição, velocidade e aceleração. Estas características tornam os filtros bastante aconselháveis na utilização de trabalhos que se desenvolvam em torno de tarefas de tracking, ou outras, tais como seguir a trajectória de um corpo no espaço (ex. estimar a velocidade e aceleração do fluxo óptico).

Existem 2 modos de abordar o problema. O primeiro é a utilização de coeficientes fixos para o filtro (filtros α - β e α - β - γ), e o segundo, coeficientes que variam com o instante de amostragem (filtros de kalman), que são determinados à priori através de um modelo de ruído e da dinâmica do sistema. O primeiro processo possui vantagens computacionais, no entanto os filtros de kalman, possuem características de alta precisão quando se trata de efectuar seguimento de sinais ruidosos. O filtro usado na estimação do fluxo óptico é do primeiro género, e pode ser utilizado em sistemas lineares invariantes .

Neste tipo de filtros, as variações são "acomodadas" ao longo do tempo através da sua modelação como ruído. Este facto proporciona que os sistemas lineares dinâmicos com coeficientes invariáveis no tempo possuam equações de estado e de medida, bastante mais simples do que aquelas que são necessárias para o caso variante no tempo. Os ganhos associados a ambos os filtros podem ser calculados "off-line", e em ambos os casos, a

implementação pode ser recursiva, onde os dados adquiridos no passado são incluídos nas presentes estimativas. Desta forma, todos os dados são utilizados, mas são "esquecidos" de forma exponencial. A estimação do valor da variável no instante k é $\hat{x}(k|k)$ e será indicado como x_s . Com estes filtros, pode-se prever os valores para o próximo instante. A predição um passo à frente é denominada por $x_p = x(k+1|k)$, e significa a estimativa no instante $k+1$ dadas as amostras até ao instante k .

Filtros com coeficientes fixos tem a vantagem de uma fácil implementação. Um dos exemplos mais aplicados deste tipo de filtros é o filtro α - β .

E.2 Filtro α - β

Este filtro é utilizado especialmente, quando apenas existem amostras de posição do objecto e assume um modelo cuja velocidade é constante. Começamos então por assumir um modelo de velocidade constante: considera-se que o objecto evolui a partir de uma velocidade inicial através de um processo de acelerações, que são constantes durante uma amostra, mas independentes entre as várias amostras. A equação do filtro α - β que prediz o seu estado corrente tem a seguinte forma:

$$x_s(k) = \hat{x}(k|k) = x_p(k) + \alpha[x_0(k) \Leftrightarrow x_p(k)] \quad (\text{E.1})$$

$$v_s(k) = \hat{\dot{x}}(k|k) = v_s(k \Leftrightarrow 1) + \frac{\beta}{qT}[x_0(k) \Leftrightarrow x_p(k)] \quad (\text{E.2})$$

$$x_p(k+1) = \hat{x}(k+1|k) = x_s(k) + Tv_s(k) \quad (\text{E.3})$$

A variável T simboliza o intervalo de tempo, $x_0(k)$ é a medida efectuada no instante k e o α e β são os parâmetros de ganho fixos. A quantidade q é normalmente definida como a unidade, mas em casos onde existam falhas de medidas pode tomar o número do valor de iterações desde a última medida efectuada. O processo de inicialização pode ser definido por:

$$x_s(1) = x_p(2) = x_0(1)$$

$$v_s(1) = 0$$

$$v_s(2) = \frac{[x_0(2) \Leftrightarrow x_0(1)]}{T}$$

A equação (E.1) é usada directamente quando a observação é efectuada no instante k . Os valores óptimos de α e β são deduzidos em [S.S. Blackman 86] e dependem apenas do quociente entre o desvio padrão do ruído do processo, e o desvio padrão do ruído da medida. A este quociente chama-se de índice λ , e os ganhos α e β são calculados na dependência de λ através das seguintes expressões:

$$\alpha = \frac{\lambda^2 + 8\lambda \Leftrightarrow (\lambda + 4)\sqrt{\lambda^2 + 8\lambda}}{8} \quad (\text{E.4})$$

$$\beta = \frac{\lambda^2 + 4\lambda \Leftrightarrow \lambda\sqrt{\lambda^2 + 8\lambda}}{4} \quad (\text{E.5})$$

E.2.1 Filtro α - β - γ

O filtro α - β - γ é idêntico ao filtro α - β , só que é baseado num modelo de aceleração constante. Por esta razão o filtro efectua uma predição quadrática, em vez de linear. No geral este filtro tende a ser mais sensível ao ruído, mas apresenta um desempenho superior na predição de velocidades que variam lentamente. As suas equações são as seguintes:

$$x_s(k) = \hat{x}(k|k) = x_p(k) + \alpha[x_0(k) \Leftrightarrow x_p(k)] \quad (\text{E.6})$$

$$v_s(k) = \hat{\dot{x}}(k|k) = v_s(k \Leftrightarrow 1) + T a_s(k \Leftrightarrow 1) + \frac{\beta}{qT}[x_0(k) \Leftrightarrow x_p(k)] \quad (\text{E.7})$$

$$a_s(k) = \hat{\ddot{x}}(k|k) = a_s(k \Leftrightarrow 1) + \frac{\gamma}{(qT)^2}[x_0(k) \Leftrightarrow x_p(k)] \quad (\text{E.8})$$

$$x_p(k+1) = \hat{x}(k+1|k) = x_s(k) + T v_s(k) + \frac{T^2}{2} a_s(k) \quad (\text{E.9})$$

Uma das inicializações possíveis para o filtro é:

$$\begin{aligned} x_s(1) &= x_p(2) = x_0(1) \\ v_s(1) &= a_s(1) = a_s(2) = 0 \\ v_s(2) &= \frac{[x_0(2) \Leftrightarrow x_0(1)]}{T} \\ a_3(3) &= \frac{[x_0(3) \Leftrightarrow 2x_0(2) + x_0(1)]}{T^2} \end{aligned}$$

Os valores ótimos para α e β são definidos como no caso anterior e dependem do índice λ . O valor ótimo para γ é dado por:

$$\gamma = \frac{\beta^2}{\alpha} \tag{E.10}$$

Bibliografia

- [Alan Oppenheim 79] Alan V. Oppenheim, Alan S. Willsky, *Signals and Systems*, Prentice Hall International, Inc.
- [Alberto Elfes 95] Alberto Elfes, *Robot Navigation: Integrating Perception, Environmental Constraints and Task Execution Within a Probabilistic Framework*, International Workshop, Reasoning with Uncertainty in Robotics Conf. - RUR, 95, Amsterdam, The Netherlands, December 1995, Springer ISBN 3-540-61376-5.
- [A.S. Roger e and E.L. Schwartz 90] Roger, A. S. and Schwartz, E.L. *Design considerations for a Space-Variant Sensor with Complex Logarithmic Geometry*, Proc. 10th International Conference on Pattern Recognition, Atlantic City, pp. 278-285, 1990.
- [Barnard e Thompson 80] S.T. Barnard and W.B. Thompson, *Disparity analysis of images*, IEEE Trans. Pattern Analysis and Machine Intelligence, PAMI-2:333-340, 1980.
- [B. Lucas e T. Kanade 81] Lucas, B. and Kanade, T., *An iterative image registration technique with an application to stereo vision*, Proc. DARPA Image Understanding Workshop pp 121-130, 1981.
- [Burt, Xen e Xu 83] P.J. Burt, C. Yen, and X. Xu, *Multi-resolution flow-through motion analysis*, In Proc. IEEE CVPR, pp. 246-252, 1983.
- [Carl Weiman 94] Carl F. R. Weiman, *Log-Polar binocular vision system*, NASA phase II, SBIR Final report, December 1994.
- [C. Brown 95] Christopher Brown, *Tutorial on Filtering, Restoration, and State Estimation*, University of Rochester, Technical Report 534, June, 1995.

- [C. Fermuller e Y. Aloimonos 91] C. Fermuller and Y. Aloimonos, *Estimating 3-D motion from image gradients*, Technical Report CARTR-618 Computer Vision Laboratory, Center for Automation Research, University of Maryland, 1991 .
- [C. Fermuller e Y. Aloimonos 92] C. Fermuller and Y. Aloimonos, *The Role of Fixation in Visual Motion Analysis*, Computer Vision Laboratory, Center for Automation Research, University of Maryland, 1992.
- [Davison e Murray 95] David W. Murray, Ian D. Reid and Andrew J. Davison, *Steering and Navigation Behaviours using Fixation*, Parks Road, Oxford University.
- [Davison e Murray 97] Andrew J Davison and David W Murray, *Mobile Robot Localisation Using Active Vision*, Parks Road, Oxford University, 1997.
- [D. Coombs e H. Herman 95] D. Coombs, M. Herman, T. Hong and M. Nashman, *Real-time Obstacle Avoidance Using Central Flow Divergence and Peripheral Flow* , National Institute of Standards and Technology, Nist Internal Report, June, 1995.
- [D. Hegger 87] D. Heeger, *A model for extraction of optical flow*, Proc. International Conf. Computer Vision, London, pg 181-190, 1987.
- [D. Marr e Poggio 79] D.Marr, and T.Poggio, *A computational theory of human stereo vision*, Proc. Royal Soc., London, B-204:301-308,1979.
- [Erwin Kreyszig 79] Erwin Kreyszig, *Advanced Engineering Mathematics*, Third Edition, ISBN 0-471-50728-8, 1979.
- [E.L. Schwartz 77] E.L. Schwartz. *Spatial Mapping in the Primate Sensory Projection: Analytic Structure and Relevance to Perception*, Biological Cybernetics, 25:181-194,1977.
- [FreeBSD Team] FreeBSD Team, *The FreeBSD Documentation Project*, <http://www.FreeBSD.org>.
- [F. Bergholm 90] F. Bergholm, *Decomposition Theory and Transformations of Visual Directions*, ICCV 90- Int. Conf. on Computer Vision, Dec 1990, pp. 85-90.

- [F. Bergholm 93] Frederik Bergholm, *On Velocity Estimation and Mean Square Error*, Proc. 8th Scandinavian Conference on Image Analysis, Trans0, Norway, pp 1093-1100, May 1993.
- [G. Sandini e C. Capurro 95] G. Sandini, C. Capurro, F. Panerai, *Space variante vision for an active camera mount*, In Proc. SPIE AeroSense95, Orlando, Florida, April 1995.
- [H. Wechsler 90] Harry Wechsler, *Computacional Vision*, Academic Press, Inc, 1990.
- [Horn e Schunck 81] Horn, B.K.P., and Schunck, B.G., *Determining optical flow*, Artificial Intelligence 17: 185-204, 1981.
- [H.H. Nagel 83] Nagel, H.H., *Displacement vectors derived from second-order intensity variations in image sequences*, Comput. Graph. Image. Process, 21: pp 85-117, 1983.
- [H.H. Nagel 87] Nagel, H.H., *On the estimation of optical flow: Relations between different approaches and some new results*, Artificial Intelligence, 33 pp 299-324, 1987.
- [I. Sousa e J. Dias 96] Inácio Sousa, Jorge Dias, Helder Araújo, *Normal Optical Flow Based on Log-Polar Images*, 4th International Symposium on Intelligent Robotics Systems, Lisbon, 22-26 July, 1996.
- [J. Barron e D. Fleet 94] J.L. Barron, D.J. Fleet, S.S. Beauchemin, *Systems and Experiment Performance of Optical Flow Techniques*, International Journal of Computer Vision, Kluwer Academic Publishers, pp 43-77, December, 1994.
- [J. Batista e J. Dias 93] J. Batista, J. Dias, H. Araújo, A. de Almeida, *Monoplanar Camera Calibration*, British Machine Vision Conference, 21-23 September, Surrey, UK, 1993, Vol2, pp. 476-488.
- [J. Dias e C. Paredes 95] J. Dias, C. Paredes, I. Fonseca, H. Araújo, J. Batista, A. de Almeida, *Simulating pursuit with Machines - Experiments with Robots and Artificial Vision*, ICRA'94, Int. Conference on Robotics and Automation, Nagoya, Japan, May 21-27, 1995.

- [J. Dias e C. Paredes 96] Jorge Dias, Carlos Paredes, Inácio Fonseca, Jorge Batista, Helder Ara'ujo, *Simulating Pursuit with Machines, Using Active Vision and Mobile Robots*, IEEE Transactions on Robotics and Automation, IEEE Transactions on Robotics and Automation. February 1998, Volume 14, Number 1, pp 1-18.
- [J. Dias e J. Batista 93] J. Dias, J. Batista, C. Simplicio, H. Ara'ujo, A. de Almeida, *Implementation of an Active Vision System*, International Workshop on Mechatronical Computer Systems for Perception and Action, Halmstad University, Sweden, pp 45-53, June 1-3, 1993.
- [J. Little, H. Bulthof e T. Poggio 88] J. Little, H. Bulthof, and T. Poggio. *Parallel Optical Flow using local voting*, In IEEE 2nd International Conference in Computer Vision, 1988.
- [J. Santos-Victor e G. Sandini 95] J. Santos-Victor, G. Sandini, F. Curotto, S. Garibaldi, *Divergent Stereo in Autonomous Navigation: From Bees to Robots* VisLab-TR 01/95, International Journal of Computer Vision, 14, pp 159-177, March 1995.
- [K.S. Fu 87] K.S. Fu, R.C. Gonzalez and C.S.G. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*, McGraw-Hill Book Company, 1987.
- [L.F. Cheong e C. Fermuller 96] L.F. Cheong, C. Fermuller, Y. Aloimonos, *Spatiotemporal representations for visual navigation*, Fourth European Conference on Computer Vision 15 - 18 April 1996 at the University of Cambridge, UK.
- [Luca Bogoni e Ruzena Bajcsy 94] , Luca Bogoni and Ruzena Bajcsy, *Functionality Investigation using a Discrete Event System Approach* Journal of Robotics and Autonomous Systems, Special issues on Discrete Event Systems Applications, Vol. 13, No. 3, pages 173-196, November 1994.
- [M. Blackburn e H. Nguyen 94] Michael Blackburn and Hoa Nguyen, *Autonomous Visual Control of a Mobile Robot*, NRad Robotics, Naval Command, Control and Ocean Surveillance Center, San Diego, CA 92152-7383, presented at the 1994 ARPA Image Understanding Workshop, November 13-16, 1994, Monterey, CA.

- [M. Irani e B. Rousso 94] Michal Irani, Benny Rousso and Shmuel Peleg, *Recovery of Ego-Motion Using Image Stabilization*, IEEE Transactions, 1994, pp. 454-460.
- [M. Tistarelli e G. Sandini 93] M. Tistarelli and G. Sandini, *On the advantages of polar and log-polar mapping for direct estimation of time-to-impact from optical flow*, IEEE Transactions and PAMI, 14(4):401-410, 1993.
- [M. Tistarelli e E. Grosso 90] M. Tistarelli, E. Grosso and G. Sandini, *Dynamic Stereo in Visual Navigation*, University of Genoa Technical report, LIRA-TR 4/90 - October 1990.
- [M.J. Hawken e A.J. Parker 94] M.J. Hawken and A.J. Parker. *Spatial Receptive Field Organization in Monkey V1 and its Relationship to the Cone Mosaic*, In Models of Neural Functions. Landy and Monshon, 1994.
- [N. Ancona e T. Poggio 93] Nicola Ancona and Tomaso Poggio, *Optical flow from 1D Correlation: application to a simple Time-To-Crash Detector*, Massachusetts Institute, AI Memo, N° 1375, C.B.C.L. paper N° 74, 1993.
- [Nelson, R.C., Aloimonos, J. 87] Nelson, R.C., Aloimonos, J., *Finding Motion Parameters From Spherical Flow Fields (Or The Advantages Of Having Eyes In The Back Of Your Head)*, WCV(87), pp. 145-150.
- [P. Fornland 95] Pär Fornland, *Direct Obstacle Detection and Motion from Spatio-Temporal Derivatives*, Active Perception Laboratory (CVAP), KTH, Stockholm, Prague, Czech Republic, September 1995.
- [Peixoto 95] Paulo José Monteiro Peixoto, *Estudo de Transparência e coerência de movimento em sequências de imagens*, Tese de Mestrado, Dept. de Engenharia Eletrotécnica, Universidade de Coimbra, 1995.
- [P. Nordlund e T. Uhlin 95] P. Nordlund and T. Uhlin, *Closing the Loop: Detection and Pursuit of a Moving Object by a Moving Observer*, CVAP-Computer Vision and Active Perception Laboratory, NADA- Royal Institute of Technology, June, 1995.

- [Povray Team] PovRay - Persistence Of Vision Development Team,
<http://www.povray.org>.
- [Ricardo e Michel 91] Swain Oropeza Ricardo and Devy Michel, *Visually-guided Navigation of a Mobile Robot in a Structured Environment* LAAS-CNRS, Toulouse, France.
- [R. Guissin e S. Ullman 91] Rami Guissin, Shimon Ullman, *Direct Computation of the Focus of Expansion From Velocity Field Measurements*, IEEE Transactions, 1991, pp. 146-155.
- [Roth, Gerhard e Levine 92] Roth, Gerhard, and Levine, Martin, *Minimal Subset Random Sampling for Pose Determination and Refinement*, AMV Strategies, pp. 1-21.
- [Robuter 92] ROBOSOFT Technopole d'Izarbel F-64210 BIDART tel (33) 05 59 41 53 60 fax (33) 05 59 41 53 79 <http://www.robosoft.fr>.
- [S.S. Blackman 86] Samuel S. Blackman, *Multiple-Target Tracking with Radar Application*, Artech House, Inc, ISBN 0-89006-176-3.
- [S. Uras e F. Girosi 88] Uras, S., Girosi, F., Verri, A., and Torre, V., *A computational approach to motion perception*, Biol. Cybern 60: pp 79-97, 1988.
- [T. Cord e D. Pallmer 94] Thomas Cord and Dirk Pallmer, *Distance Measuring for Mobile Robots Using Axial Motion Stereo*, Forschungszentrum Informatik, in Proceedings of the 3th International Workshop on Robotics in Alpe-Adria Region, Bled, July 1994.
- [V. Sundaeswaran e P. Bouthemy 94] , Venkataraman Sundaeswaran, Patrick Bouthemy and François Chaumette, *Visual servoing using dynamic image parameters* INRIA-Institut National de Recherche en Informatique et en Automatique Rapport de recherche n° 2336 -Août 1994.
- [Yaakov Xiao 93] Yaakov Bar-Shalom, Xiao-Rong Li, *Estimation and Tracking: Principles, Techniques, and Software* Artech House, inc, 1993.

- [Y. Aloimonos e Z. Duric 94] Yiannis Aloimonos and Zoran Duric, *Estimating The Heading Direction Using Normal Flow*, International Journal of Computer Vision, Kluwer Publishers, 1994, 13:1, pp. 33-56.