



Universidade do Porto
Faculdade de Engenharia

FEUP

Armindo Fernandes dos Santos Martins

Sistema de Navegação Baseado na Leitura Óptica do Terreno

Sistema de Navegação Baseado na Leitura

FACULDADE DE ENGENHARIA E ARQUITECTURA UNIVERSIDADE DO PORTO

DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES

ERRATA

Página	Parágrafo	Linha	Onde está	Deve estar
40	8	2	... bem com algumas...	... bem como algumas...
52	2	2	...sujeita vibrações...	...sujeita a vibrações...
77	5	1	Caso particulares...	Casos particulares...
81	3	1	...ver o mesmo o mesmo...	...ver o mesmo...
107	1	1	...e Figura 4.3...	...e Figura 4.13...
157	5	1	...para = 0° no sub-capítulo 5.4...	...para $\theta = 0^\circ$ no sub-capítulo 6.2...

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO
DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES

SISTEMA DE NAVEGAÇÃO BASEADO NA LEITURA ÓPTICA DO TERRENO

ARMINDO FERNANDES DOS SANTOS MARTINS

LICENCIADO EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES
PELA FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

DISSERTAÇÃO SUBMETIDA PARA SATISFAÇÃO PARCIAL DOS REQUISITOS DO GRAU DE
MESTRE EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES
(ÁREA DE ESPECIALIZAÇÃO DE INFORMÁTICA INDUSTRIAL)

DISSERTAÇÃO REALIZADA SOB A SUPERVISÃO DE
PROFESSOR DOUTOR FRANCISCO JOSÉ DE OLIVEIRA RESTIVO
DO DEPARTAMENTO DE ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES
DA FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

PORTO, JUNHO DE 2003

*Dedico esta dissertação
à minha mulher Carla.*

Resumo

Os sistemas de posicionamento tendo em vista a condução de um veículo de forma autónoma têm sido alvo de intensa investigação pelas diversas áreas de aplicações, ao nível industrial, militar, segurança pública, exploração de novos locais, aviação, marinha, domésticas, entretenimento, entre outras.

A presente dissertação visa desenvolver e validar um sistema que, a partir de imagens sucessivas do terreno obtidas através de leitura óptica, passa por determinar a posição do veículo, atribuindo-se particular ênfase ao veículo AGV (*Automatic Guided Vehicle*).

A primeira parte da dissertação consiste num trabalho de pesquisa, no que diz respeito aos sistemas de navegação existentes para os AGV's, e também aos processos de comparação entre imagens, em que se baseará o método desenvolvido.

A metodologia seguida nesta dissertação adopta, o método de *matching* baseado na transformada de Fourier – Mellin. Este método permite obter a translação, a rotação e o factor de escala entre duas imagens.

Na segunda parte desta dissertação apresenta-se o trabalho prático desenvolvido, assim como a sua simulação e validação, e quadros resumo dos principais parâmetros de projecto.

Mostra-se também como a aplicação desenvolvida poderá ser facilmente integrada numa arquitectura modular simples de um sistema de navegação, constituindo-se assim em mais uma opção a outros sistemas.

Nas conclusões mostra-se como os objectivos foram atingidos e referem-se algumas linhas de trabalho futuras.

Abstract

The systems of positioning, considering the steering of a vehicle autonomously have been object of intense investigation by different areas of applications at industrial, military, public security, exploration of new locations, aviation, marine, domestic, entertainment level, to name a few.

This dissertation aims to develop and validate a system that, from successive images of the field obtained through optic reading, determines the position of the vehicle, giving particular relevance to the AGV vehicle (*Automatic Guided Vehicle*).

The first part of the dissertation consists of a research work, in what concerns the existing navigation systems for the AGV, as well as processes of comparison of images, that will be the bases of the developed method.

The methodology followed in this dissertation uses the *matching* method based in the Fourier – Mellin transform. This method allows obtaining the translation, the rotation and the scaling of two images.

The second part of this dissertation presents the practical work developed, as well as its simulation and validation, and tables summarising the main parameters of the project.

It also shows how the developed application can easily be integrated in a simple modular architecture of a navigation system, thus becoming a further alternative to other system.

The conclusion shows how the objectives were attained and mentions some future lines of work.

Résumé

Les systèmes du positionnement concernant la conduction d'un véhicule de forme autonome ont été objet d'enquête intensive par les divers domaines d'applications, au niveau industriel, militaire, de la sécurité publique, de l'exploitation de nouveaux endroits, de l'aviation, marine, domestique, de la récréation, d'entre autres.

Le but de la présente dissertation est le développement et la validation d'un système qui, à partir d'images successives du terrain obtenues par lecture optique, détermine la position du véhicule, en donnant particulière emphase au véhicule AGV (*Automatic Guided Vehicle*)

La première partie de la dissertation consiste à un travaille de recherche, en ce qui concerne les systèmes de navigation existants pour les AGV, aussi bien que les procès de comparaison entre images, lesquels seront la base de la méthode développée.

La méthodologie appliquée à cette dissertation adopte la méthode du *matching* basée sur la transformée de Fourier – Mellin. Cette méthode permet obtenir la translation, la rotation et le facteur d'échelle entre deux images.

La deuxième partie de cette dissertation présente le travail pratique développé, ainsi bien que la simulation et validation, et des tables qui résumant les paramètres du project.

On montre aussi comme l'application développée pourra facilement être intégrée dans une architecture modulaire simple d'un système de navigation, devenant ainsi une alternative de plus aux autres systèmes.

Dans la conclusion ont montre comme les buts ont été réussis e on réfère quelques lignes de travaille vers l'avenir.

Agradecimentos

Gostaria de agradecer em primeiro lugar ao Prof. Dr. Francisco José de Oliveira Restivo, na qualidade de orientador desta dissertação, pela forma paciente com que orientou este trabalho, quer na orientação científica, quer em diversos aspectos da sua execução e pela disponibilidade sempre manifestada, inclusive ao fim de semana.

Agradeço à Prof. Graça Ruano e à Eng.^a Margarida Madeira, pela simpatia e por facultar o acesso ao equipamento e instalações, da Faculdade de Ciências e Tecnologia da Universidade do Algarve, permitindo-me assim trabalhar no período de férias.

A todos os meus colegas de trabalho que me *aturaram* nos dias difíceis.

Agradeço também aos meus amigos pelo apoio que me deram e pelos encontros não realizados.

À minha amiga Ana pelo contributo para a dissertação, com uma pronúncia Algarvia.

Aos meus pais, sogros e irmão pela tolerância à minha ausência, pelas respostas negativas aos convites efectuados.

Por último à minha mulher pelas noitadas a ler e a corrigir parte da dissertação, por toda a paciência com que me apoiou e deu força, principalmente nos momentos de desânimo e de maior dificuldade, e por todo o seu amor e compreensão.

Conteúdo

1	Introdução.....	25
1.1	Motivação.....	25
1.2	Organização da dissertação.....	29
2	Estado da Arte.....	31
2.1	Navegação de AGV's.....	31
2.2	<i>Matching</i> de imagens.....	35
2.3	Aplicação de <i>matching</i> de imagens.....	40
3	Metodologia seguida.....	51
3.1	Caracterização do Problema.....	51
3.2	Transformada de Fourier.....	53
3.2.1	<i>Transformada de Fourier Discreta – DFT</i>	53
3.2.2	<i>Algumas propriedades importantes da transformada de Fourier discreta</i>	55
3.3	Transformada de Fourier – Mellin.....	75
4	Descrição do sistema desenvolvido.....	87
4.1	Ferramenta de desenvolvimento.....	87
4.2	Módulos do Programa.....	90
4.2.1	<i>Módulo Principal</i>	90
4.2.2	<i>Determinação da Rotação</i>	92
4.2.3	<i>Determinação da translação</i>	102
4.3	Limitações.....	106
5	Descrição do sistema de simulação.....	109
5.1	Geração das trajectórias.....	110
5.1.1	<i>Geração da sequência de imagens da trajectória, Seqmov.m</i>	110
5.1.2	<i>Geração das imagens de um trajecto numa superfície</i>	112
5.2	Determinação da trajectória a partir das imagens da superfície.....	121

6	Validação e verificação de resultados	129
6.1	Parametrização do sistema de navegação.....	129
6.1.1	<i>Tempo de processamento</i>	132
6.1.2	<i>Espaçamento máximo entre duas imagens.....</i>	133
6.1.3	<i>Gama de rotações entre duas imagens</i>	146
6.1.4	<i>Quadros resumo</i>	157
6.2	Distância máxima entre referências.....	158
6.3	Simulação	161
7	Arquitectura e aplicabilidade do sistema de navegação	167
7.1	Aplicabilidade do sistema de navegação desenvolvido.....	169
8	Conclusões e perspectivas de trabalhos futuros.....	171
	Referências Bibliográficas	173

Lista de Figuras

Figura 1.1	Plataforma de fabrico [1].	26
Figura 1.2	Número de imagens e dimensão destas entre o ponto A e o ponto B.	28
Figura 1.3	Terreno uniforme com imagens deslocadas mas iguais ($I_1 = I_2$).	28
Figura 2.1	Tipos de motricidade existente nos AGV's [3].	32
Figura 2.2	Sistema de AGV filoguiado [3].	33
Figura 2.3	Sistema de AGV por laser [2].	34
Figura 2.4	Imagens de ressonância magnética à cabeça de um paciente; a) Fatia (<i>slice</i>) T1 e b) Fatia (<i>slice</i>) T2 [6].	41
Figura 2.5	Resultado da aplicação do algoritmo de <i>matching</i> baseado na transformada de Fourier – Mellin às imagens da Figura 2.4. O pico corresponde a um ângulo de 6° e a uma escala de 100% [6].	42
Figura 2.6	Saída do filtro de comparação só de fase linear aplicado às duas imagens da Figura 2.4, mas com T2 rodado de -6° . O pico indica o deslocamento entre as duas imagens (5, -4) [6].	42
Figura 2.7	Zona do cérebro demarcada nas duas imagens da Figura 2.4 [6].	43
Figura 2.8	Imagens de satélite do mesmo local em tempos diferentes [6].	44
Figura 2.9	Resultado da aplicação do algoritmo de <i>matching</i> baseado na transformada de Fourier – Mellin às duas imagens da Figura 2.8 [6].	45
Figura 2.10	Saída do filtro de comparação só de fase aplicado às duas imagens da Figura 2.8, mas com a imagem b) rodado de -7° . O pico indica o deslocamento entre as duas imagens (-12, 1) [6].	45
Figura 2.11	Impressão digital [6].	47
Figura 2.12	Conjunto de impressões digitais incluindo a da Figura 2.11 [6].	47

Figura 2.13	Resultado da aplicação do algoritmo de <i>matching</i> baseado na transformada de Fourier – Mellin entre a imagem da Figura 2.11 e cada uma das imagens da Figura 2.12 da a) até à d) respectivamente [6].....	48
Figura 2.14	Imagens de objectos; a) faca e chave, b) <i>template</i> da chave [6].	49
Figura 2.15	Resultado da aplicação do algoritmo de <i>matching</i> baseado na transformada de Fourier – Mellin às imagens da Figura 2.14 [6].....	49
Figura 3.1	Imagens consecutivas relacionadas por uma translação δ e uma rotação θ ..	52
Figura 3.2	DFT com $F(u) = F(e^{j\omega}) \Big _{\omega=\omega_u=\frac{2\pi u}{N}}$ [103].....	54
Figura 3.3	DFT bidimensional a partir da DFT unidimensional [104].	56
Figura 3.4	Comparação do espectro unidimensional da amplitude e da fase para a sequência original e com translação x_0 [103].	58
Figura 3.5	Ilustração da propriedade da periodicidade da transformada de Fourier [104].	60
Figura 3.6	Propriedade Rotação da Transformada Fourier: a) Imagem original; b) Espectro da imagem original; c) Imagem rodada; d) Espectro resultante [104].	62
Figura 3.7	Ilustração gráfica da convolução de funções unidimensionais contínuas [104].	65
Figura 3.8	Comparação entre a convolução de funções unidimensionais contínuas e as discretas [104].	68
Figura 3.9	Ilustração das operações necessárias para fazer a convolução bidimensional das funções $f(x, y)$ e $g(x, y)$ contínuas [104].	70
Figura 3.10	Localização de uma parte da imagem numa outra [107].	72
Figura 3.11	Ilustração gráfica da correlação [104].	73
Figura 3.12	Exemplos de sinais com linearidade da fase diferentes; a) fase zero, b) fase linear e c) fase não linear [109].	78
Figura 3.13	Exemplo da linearidade da fase para diferentes deslocamentos; a) fase linear, b) fase não linear [109].	79
Figura 3.14	a) Espectro da amplitude e da fase do sinal $d(n)$ para $(k=0)$; b)	

	deslocamento no sentido positivo ($k>0$); c) deslocamento no sentido negativo ($k<0$) [103].....	79
Figura 3.15	Visualização da fase no domínio das frequências provocado pelo deslocamento do sinal no domínio dos tempos [109].....	80
Figura 3.16	Informação presente na fase [109].....	81
Figura 3.17	Jean Baptiste Joseph Fourier e a sua Transformada de Fourier; a) imagem original; b) espectro da amplitude; c) espectro da fase; d) reconstrução só a partir da amplitude; e) Reconstrução só a partir da fase [105].	82
Figura 4.1	Fluxograma do módulo principal <i>Navega.m.</i>	91
Figura 4.2	Fluxograma do modulo de determinação da rotação, <i>Rota_escala.m.</i>	92
Figura 4.3	Janela de Hanning aplicada às imagens <i>imageref</i> e <i>imagein</i>	93
Figura 4.4	Imagem de referência com janela de Hanning aplicada.	94
Figura 4.5	Transformada de Fourier e deslocação para o centro do espectro.....	95
Figura 4.6	Filtro passa alto para extracção da amplitude.....	96
Figura 4.7	Espectro da amplitude da imagem original.	96
Figura 4.8	Espectro da amplitude da imagem original em coordenadas logarítmicas – polares.....	98
Figura 4.9	Espectro da amplitude da imagem original em coordenadas logarítmicas – polares, a) segundo o eixo radial, b) segundo o eixo radial, mas com janela de Hanning aplicada a este.....	99
Figura 4.10	Transformada de Fourier do espectro da amplitude <i>ext_imref_lp</i>	99
Figura 4.11	Transformada de Fourier do espectro da amplitude <i>ext_imin_lp</i>	100
Figura 4.12	Resultado da correlação de fase.....	101
Figura 4.13	Fluxograma do módulo <i>Translacao.m</i>	103
Figura 4.14	Correlação cruzada entre as transformadas de Fourier da imagem de referência <i>imageref</i> e da imagem de entrada <i>imagein</i> sem o ângulo de rotação <i>theta</i>	104
Figura 4.15	Correlação cruzada entre as transformadas de Fourier da imagem de referência <i>imageref</i> e da imagem de entrada <i>imagein</i> sem o ângulo de	

	rotação $theta+180^\circ$	105
Figura 5.1	Fluxograma do módulo de obtenção da sequência de imagens da trajectória, <i>Seqmov.m</i>	110
Figura 5.2	Fotografia base usada para a criação da sequência de imagens.....	111
Figura 5.3	Fluxograma do módulo de fotografia da superfície nos pontos pretendidos, <i>Photo.m</i>	113
Figura 5.4	Trajecto em curva com as imagens <i>I</i> afectadas por rotação e translação do veículo <i>V</i>	114
Figura 5.5	Trajecto em curva com as imagens <i>I</i> afectadas só por translação do veículo <i>V</i>	114
Figura 5.6	Fluxograma do módulo <i>Cria_chao.m</i> , de criação da superfície com a dimensão $(mv+1)$ por $(mh+1)$	115
Figura 5.7	Superfície (<i>chao</i>) criada a partir de uma imagem base (<i>image_b</i>).....	116
Figura 5.8	Fluxograma do módulo de posicionamento do ponto (X, Y) , do início da fotografia, na superfície <i>Impos.m</i>	117
Figura 5.9	Superfície <i>chao</i> , exemplo de localização da área a fotografar.....	118
Figura 5.10	Fluxograma do módulo usado para retirar a fotografia pretendida, para pontos não inteiros, <i>ImFrac.m</i>	120
Figura 5.11	Fluxograma do módulo de obtenção da trajectória a partir de uma sequência de imagens, <i>Simula.m</i>	122
Figura 5.12	Duas primeiras imagens (256x256) de um trajecto sobre a superfície <i>chao</i> ; a) imagem de referência <i>imageref</i> com o ponto inicial em 1100, 800; b) imagem a comparar <i>imagein</i> deslocada de 40 pixels, em relação a <i>imageref</i> , em ambos os eixos.	123
Figura 5.13	Diferentes métodos de rodar a imagem de um ângulo $theta$	124
Figura 5.14	Imagem <i>imagein</i> rodada de um ângulo $theta$ igual a 20°	125
Figura 6.1	Imagem base para gerar uma superfície com característica aleatórias.....	130
Figura 6.2	Imagem base para gerar uma superfície com característica <i>paisagística</i>	130
Figura 6.3	Imagem base para gerar uma superfície com característica uniforme.	131

Figura 6.4	Tempo de processamento em função da dimensão das imagens: a) Superfície aleatória; b) Superfície <i>paisagística</i> ; c) Superfície uniforme.....	132
Figura 6.5	Erro de X e Y , função da dimensão da translação para $\theta=0$ e superfície aleatória.....	134
Figura 6.6	Erro de X e Y , função da dimensão da translação para $\theta=0$ e superfície <i>paisagística</i>	134
Figura 6.7	Erro de X e Y , função da dimensão da translação para $\theta=0$ e superfície uniforme.....	135
Figura 6.8	Visualização pormenorizada da Figura 6.5 (superf. aleatória).....	136
Figura 6.9	Visualização pormenorizada da Figura 6.6 (superf. <i>paisagística</i>).....	136
Figura 6.10	Visualização pormenorizada da Figura 6.7 (superf. uniforme).....	137
Figura 6.11	Erro de X e Y , função da dimensão da translação para $\theta=10$ e superfície aleatória.....	138
Figura 6.12	Erro de X e Y , função da dimensão da translação para $\theta=10$ e superfície <i>paisagística</i>	138
Figura 6.13	Erro de X e Y , função da dimensão da translação para $\theta=10$ e superfície uniforme.....	139
Figura 6.14	Visualização pormenorizada da Figura 6.11 (superf. aleatória).....	140
Figura 6.15	Visualização pormenorizada da Figura 6.12 (superf. <i>paisagística</i>).....	140
Figura 6.16	Visualização pormenorizada da Figura 6.13 (superf. uniforme).....	141
Figura 6.17	Erro de X e Y , função da dimensão da translação para $\theta=40$ e superfície aleatória.....	142
Figura 6.18	Erro de X e Y , função da dimensão da translação para $\theta=40$ e superfície <i>paisagística</i>	142
Figura 6.19	Erro de X e Y , função da dimensão da translação para $\theta=40$ e superfície uniforme.....	143
Figura 6.20	Visualização pormenorizada da Figura 6.17 (superf. aleatória).....	144
Figura 6.21	Visualização pormenorizada da Figura 6.18 (superf. <i>paisagística</i>).....	144
Figura 6.22	Visualização pormenorizada da Figura 6.19 (superf. uniforme).....	145

Figura 6.23	Erro de θ , função do ângulo de rotação entre as imagens de uma superfície aleatória para $X, Y = 0$	147
Figura 6.24	Erro de θ , função do ângulo de rotação entre as imagens de uma superfície paisagística para $X, Y = 0$	147
Figura 6.25	Erro de θ , função do ângulo de rotação entre as imagens de uma superfície uniforme para $X, Y = 0$	148
Figura 6.26	Visualização pormenorizada da Figura 6.23 (superf. aleatória).....	149
Figura 6.27	Erro de θ , função do ângulo de rotação entre as imagens consecutivas de uma superfície aleatória para $X, Y = 5, 10$ e 20	150
Figura 6.28	Erro de θ , função do ângulo de rotação entre as imagens consecutivas de uma superfície paisagística para $X, Y = 5, 10$ e 20	150
Figura 6.29	Erro de θ , função do ângulo de rotação entre as imagens consecutivas de uma superfície uniforme para $X, Y = 5, 10$ e 20	151
Figura 6.30	Visualização pormenorizada da Figura 6.27 (superf. aleatória).....	152
Figura 6.31	Visualização pormenorizada da Figura 6.29 b) (superf. uniforme).	152
Figura 6.32	Erro de θ , função do ângulo de rotação entre as imagens consecutivas de uma superfície aleatória para $X, Y = 10, 20$ e 40	153
Figura 6.33	Erro de θ , função do ângulo de rotação entre as imagens consecutivas de uma superfície paisagística para $X, Y = 10, 20$ e 40	153
Figura 6.34	Erro de θ , função do ângulo de rotação entre as imagens consecutivas de uma superfície uniforme para $X, Y = 10, 20$ e 40	154
Figura 6.35	Visualização pormenorizada da Figura 6.32 (superf. aleatória).....	155
Figura 6.36	Visualização pormenorizada da Figura 6.33 (superf. paisagística).....	155
Figura 6.37	Visualização pormenorizada da Figura 6.34 (superf. uniforme).....	156
Figura 6.38	Quadro resumo de alguns resultados da simulação efectuada para $\theta = 0^\circ$	157
Figura 6.39	Quadro resumo de alguns resultados da simulação efectuada para $\theta = 10^\circ$	158
Figura 6.40	Quadro resumo de alguns resultados da simulação efectuada para $\theta = 40^\circ$	158

Figura 6.41	Quadro resumo de alguns resultados da simulação efectuada.....	158
Figura 6.42	Distância máxima a percorrer em cm, para um determinado erro máximo, dependendo da dimensão das imagens.	160
Figura 6.43	Deslocamento a partir de imagens desfasadas entre si de 8, 25 e 55 pixels, respectivamente, para as imagens de dimensão 64x64, 128x128 e 256x256 sem rotação.	161
Figura 6.44	Erro registado em cada ponto calculado.	162
Figura 6.45	Erro acumulado durante o trajecto.....	162
Figura 6.46	Espaço percorrido por segundo.	163
Figura 6.47	Deslocamento a partir de imagens desfasadas entre si de 16, 40 e 70 pixels, respectivamente, para as imagens de dimensão 64x64, 128x128 e 256x256 sem rotação.	164
Figura 6.48	Imagens deslocadas sobre uma linha.	164
Figura 6.49	Erro registado em cada ponto calculado para a 2. ^a simulação.....	165
Figura 6.50	Erro acumulado durante o trajecto para a 2. ^a simulação.....	165
Figura 6.51	Espaço percorrido por segundo para a 2. ^a simulação.....	166
Figura 7.1	Sistema de condução de um veículo baseado numa arquitectura modular...	168

Abreviaturas

- AGV** – Automatic Guided Vehicle
- DFT** – Discrete Fourier Transform
- FFT** – Fast Fourier Transform
- IFFT** – Inverse Fast Fourier Transform
- CIM** – Computer Integrated Manufactured
- CCP** – Centro de CIM do Porto
- IDIT** – Instituto de Desenvolvimento e Inovação Tecnológica
- GPS** – Global Position System
- SNR** – Signal to Noise Ratio
- SLI** – Sistema Linear e Invariante
- GPIB** – General Purpose Interface Bus
- VISA** – Virtual Instrument Software Architecture

Capítulo 1

Introdução

O autor da dissertação “*Sistema de Navegação baseada na leitura óptica do terreno*” pretende de certa forma contribuir para o estudo de um assunto que tem sido alvo de intensa investigação e que diz respeito à condução de um veículo de uma forma autónoma.

Este é um assunto sobejamente importante, porque abrange um leque muito grande de aplicações, podendo estas serem ao nível militar, ao nível da segurança pública, exploração de novos locais, aviação, marinha, industrial, domésticos, entretenimento, entre outros.

Mediante a diversidade de aplicações, também existe um sem número de soluções, cada uma com as suas vantagens e desvantagens.

Uma das soluções cada vez mais desenvolvida é baseada na leitura óptica do meio envolvente, isto porque a velocidade de processamento aumenta de dia para dia, existe uma maior facilidade de criação de processadores dedicados, melhorando e facilitando muito o processamento de imagem.

Esta dissertação trata precisamente de, a partir de imagens adquiridas do terreno, determinar o posicionamento do veículo.

1.1 Motivação

A motivação para este trabalho surgiu realmente da introdução no mercado do rato óptico da Microsoft, cujo princípio de funcionamento se baseará na determinação do deslocamento numa superfície a partir da aquisição e processamento de imagens de 16x16

pixels dessa mesma superfície.

O aparecimento do rato óptico da Microsoft ocorreu aproximadamente na mesma altura que o CCP – Centro CIM do Porto, em virtude da sua fusão por incorporação no IDIT – Instituto de Desenvolvimento e Inovação Tecnológica, transferiu a sua plataforma de fabrico para Santa Maria da Feira.

Esta plataforma, representada pela Figura 1.1, consiste essencialmente numa célula flexível de fabrico e numa célula de inspecção e montagem, sendo o fluxo de materiais assegurado por um sistema de armazenamento e transporte que inclui um AGV – *Automatic Guided Vehicle*, fornecido pela Efacec Automação e Robótica, SA em 1995.

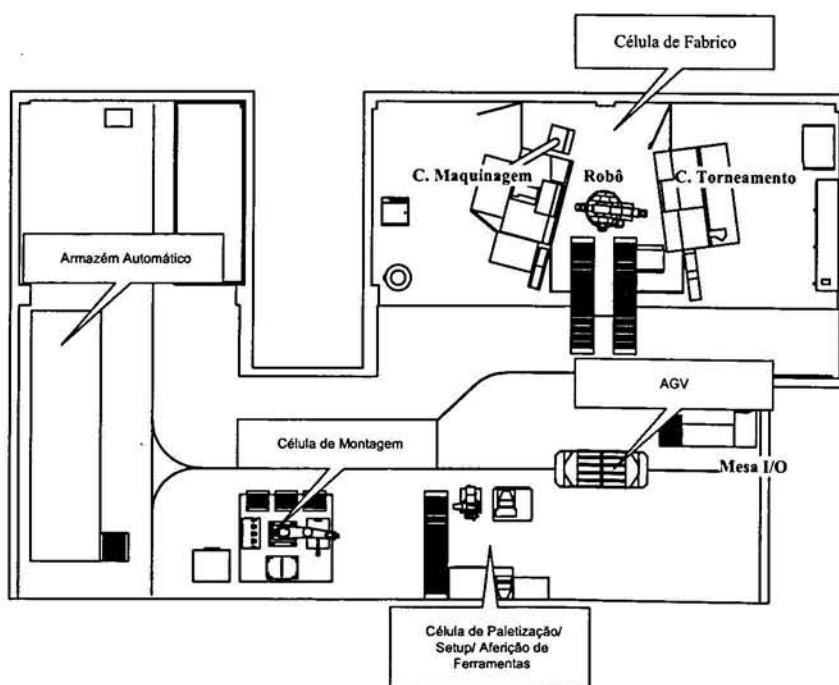


Figura 1.1 Plataforma de fabrico [1].

Este AGV usa o sistema filoguiado com recurso a um fio condutor inserido no solo, que funciona como antena emissora. Por outro lado o AGV transporta um receptor que determina um sinal de erro medindo o afastamento relativamente ao condutor.

Esta tecnologia é pouco flexível, na medida em que uma simples alteração da geometria dos trajectos implica, para além da reinstalação dos fios condutores do sistema de

navegação, alterações ao nível do software de razoável complexidade.

A própria Efacec já adoptou outra solução, fornecida pela empresa sueca NDC (www.ndc.se), baseada num sistema de navegação por triangulação que recorre a um emissor laser rotativo embarcado no AGV e a dispositivos reflectores instalados estrategicamente em vários locais da instalação fabril.

Existem contudo muitas outras soluções, baseadas em diversos princípios de funcionamento, e o CCP/IDIT e a Efacec por mais do que uma vez consideraram a hipótese de conceber e desenvolver um sistema de sua propriedade.

Desenvolvendo o autor desta dissertação a sua actividade profissional na Efacec, embora noutra empresa do grupo, compreende-se a sua motivação para responder a este desafio que foi colocado pelo seu orientador: averiguar se o princípio de funcionamento do rato óptico da Microsoft poderá ser utilizado para a navegação de AGV's em ambiente industrial real.

Este desafio não é trivial. Por um lado, enquanto que o controlo do rato óptico é realizado em malha fechada, pelo *feedback* visual, no caso dos AGV's a navegação será cega, pelo menos entre pontos de referência. Por outro lado, enquanto que no rato óptico é possível mudar facilmente a superfície do terreno em caso de dificuldade, numa instalação fabril isso será normalmente mais difícil.

O problema a que esta dissertação procura dar resposta é um problema de engenharia, que consiste em definir metodologia de projecto do sistema de navegação e as regras subjacentes ao dimensionamento de todo o sistema, isto é, conhecendo a distância a percorrer em navegação cega e a velocidade do veículo, determinar a frequência de aquisição de imagens e a sua dimensão e resolução, bem como o conjunto de cálculos a realizar entre duas imagens. Numa situação de tempo real, se o veículo se deslocar a uma velocidade v entre dois pontos distantes de d e se um erro máximo de posicionamento ε obrigar ao processamento de n imagens intermédias de $x.y$ pixel e p pixel/unidade de comprimento, então a determinação da posição relativa de duas imagens consecutivas não pode durar mais do que $\frac{d}{n.v}$.

Obviamente que este problema não tem uma solução única. A posição relativa de pontos A e B pode ser determinada com base em duas imagens, desde que haja uma área comum às duas de dimensão suficiente. À medida que a distância aumenta, e para um determinado erro máximo de posicionamento, as imagens crescem em número de pixels para dimensões inportáveis, tornando-se necessárias imagens intermédias, tantas mais quanto menor for a dimensão de cada uma, Figura 1.2, e sendo certo que os erros acumulados irão exigir no caso de muitas imagens que em cada uma haja mais pixels por unidade de comprimento do terreno.

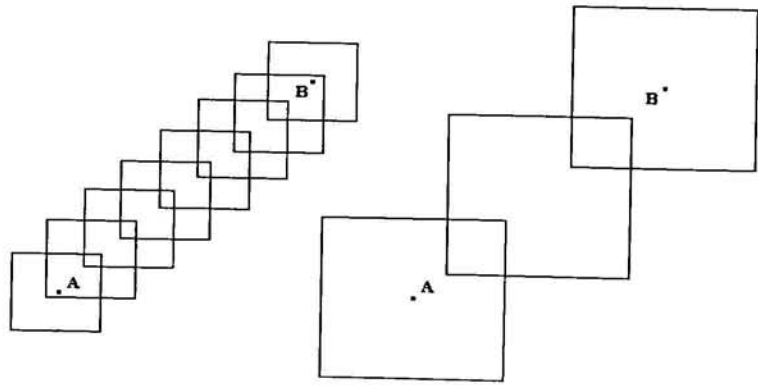


Figura 1.2 Número de imagens e dimensão destas entre o ponto A e o ponto B.

Outro problema tem a ver com o aspecto do terreno. O método não resulta obviamente num terreno com aspecto uniforme, Figura 1.3, e se houver uma estrutura periódica pode haver problemas de *aliasing*.

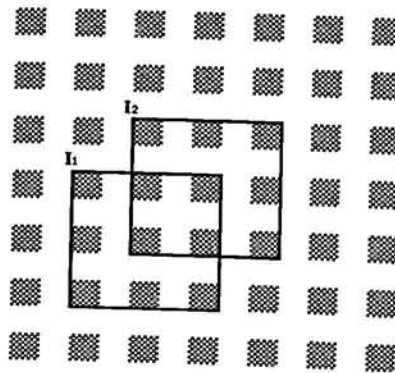


Figura 1.3 Terreno uniforme com imagens deslocadas mas iguais ($I_1 = I_2$).

1.2 Organização da dissertação

A dissertação inicia-se com uma revisão dos métodos de navegação de AGV's que têm sido utilizados industrialmente e dos métodos de determinação de fragmentos correspondentes em duas imagens, que se pretende aplicar, bem como das suas aplicações noutros domínios.

No terceiro capítulo, caracteriza-se o problema e apresenta-se com algum detalhe a transformada de Fourier – Mellin que vai ser utilizada, justificando-se a sua adopção.

No quarto capítulo, é apresentada e defendida a opção de utilização do Matlab como ferramenta de desenvolvimento, assim como é descrito todo o sistema desenvolvido. Analisa-se também algumas limitações da solução.

O quinto capítulo descreve com algum detalhe o sistema de simulação desenvolvido em Matlab, usado para fazer a validação e verificação de resultados.

No capítulo seis faz-se um estudo exaustivo do funcionamento do sistema, identifica-se claramente quais as suas limitações e regista-se informação suficiente, para auxiliar em aplicações futuras.

O capítulo sete apresenta uma proposta de arquitectura modular e algumas aplicações possíveis para o sistema de navegação desenvolvido.

Por fim serão apresentadas as conclusões da dissertação e possíveis trabalhos futuros.

Capítulo 2

Estado da Arte

Uma das aplicações mais comuns dos sistemas de navegação encontra-se na indústria, nos veículos de transporte de material (AGV) entre pontos específicos da nave fabril. Como tal torna-se oportuno fazer uma revisão e dar uma breve explicação dos diferentes sistemas de navegação existentes nos AGV's, mostrando as vantagens e desvantagens de cada um.

Como no processamento entre imagens é muitas vezes usada a determinação de semelhanças (*matching*) entre estas, e é a técnica utilizada nesta dissertação, neste capítulo também se referencia alguns dos métodos existentes para esta técnica.

Por fim podem-se ver algumas aplicações dos métodos referenciados.

2.1 Navegação de AGV's

Os AGV (*Automatic Guided Vehicles*) existem desde 1950. Há uma vasta gama destes veículos, adaptados ao tipo de indústria onde se inserem, porque em termos de carga podem ir desde pequenas cargas, até valores superiores a 100 toneladas. Podem ser especificados para piso macio, interior da indústria, ou para pisos bastante rugosos, no exterior.

Em termos de tracção e mudança de direcção existem essencialmente três tipos, representados na Figura 2.1: a) mostra o AGV com duas rodas de apoio laterais e uma roda directriz e motriz; b) é um sistema composto por duas rodas de apoio, uma à frente e outra atrás e duas motrizes mudando de direcção conforme a tracção aplicada em cada

roda e por ultimo, c) tem as duas rodas de apoio nas laterais e as de tracção, direccionalizantes, uma à frente e outra atrás, funcionando em contra brecaagem; este último foi o sistema fornecido pela Efacec ao CCP/IDIT.

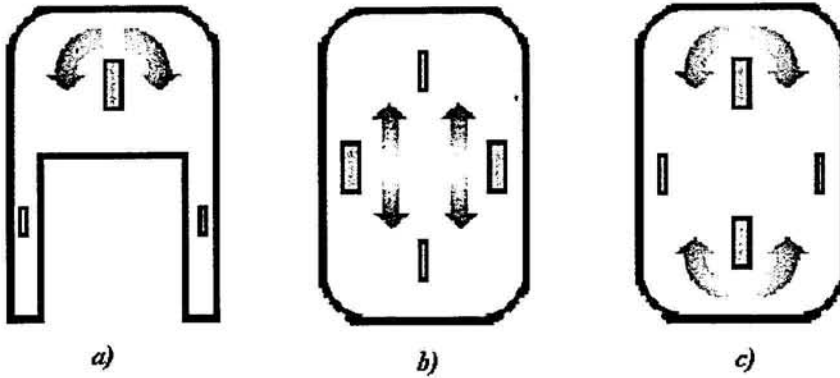


Figura 2.1 Tipos de motricidade existente nos AGV's [3].

Cada vez mais os AGV's são controlados por computador, permitindo otimizar a sua utilização, processando ordens de transporte, quer entre pontos da linha de fabrico, quer no carregamento dos camiões a partir directamente da linha assim como, no armazenamento. Dependendo do tipo de carga pode manobrá-las até uma altura de 10 ou mais metros.

Os métodos de navegação dos AGV's são vários e podem ser subdivididos em dois grupos distintos, o dos sistemas fechados e o dos sistemas abertos, ou seja, os que implicam um movimento sempre pelos mesmos trajectos e os que têm trajectos livres, apenas limitados a uma determinada área. Estes últimos necessitam de armazenar em memória um mapa sobre a área onde se encontram e sobre o trajecto a efectuar.

Os sistemas fechados requerem ou um trajecto marcado no solo, por exemplo uma pista de material reflector em que o AGV detecta a reflexão da luz, ou um fio condutor inserido no solo que é designado por sistema filoguiado. Este é o sistema mais utilizado, e tem como base de funcionamento um campo electromagnético criado por uma corrente a circular no condutor enterrado no solo. Por outro lado, o veículo tem uma antena composta por duas bobinas, como é mostrado na Figura 2.2, que circulam uma de cada lado do con-

dutor enterrado.

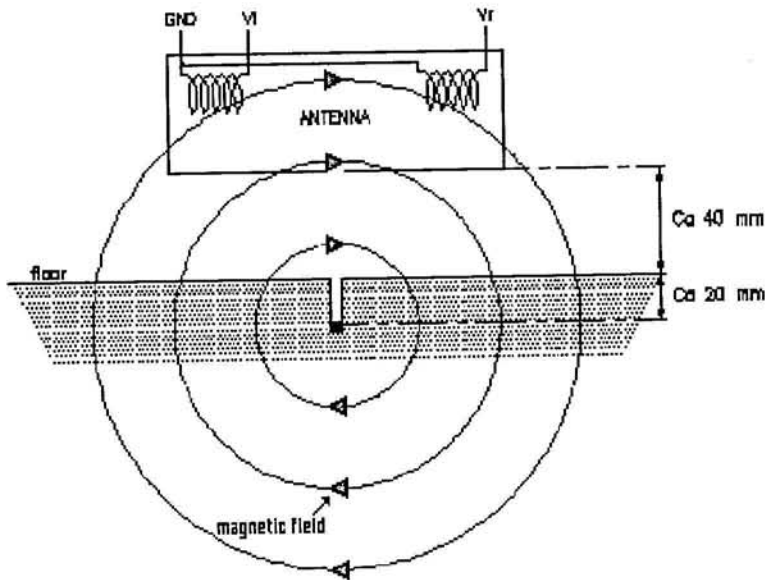


Figura 2.2 Sistema de AGV filoguiado [3].

O campo electromagnético provocado pela corrente a passar no condutor gera tensões nas bobinas que quando se encontram à mesma distância do condutor, têm o mesmo nível de tensão, logo a diferença é igual a zero. Se houver um deslocamento para um dos lados então a diferença das tensões já é diferente de zero e o AGV recebe informação para se deslocar num determinado sentido, por forma a que a diferença se anule.

O sistema filoguiado tem a vantagem de ter uma grande precisão. Usando componentes de alta qualidade, o corte necessário a fazer no chão para a introdução do condutor é mínimo. Como grande desvantagem, sempre que for preciso mudar a trajectória tem que se voltar a instalar o condutor e se for uma zona muito movimentada pode ser difícil o AGV chegar rapidamente ao destino, porque ele não se pode desviar.

Existe uma alternativa ao filoguiado, mas que segue mais ou menos o mesmo princípio, que é substituir o condutor por pequenos ímãs espaçados de cinco a dez metros entre si e colocar no lugar da antena, transportada pelo AGV, um sensor rotativo que determina a posição do íman. Neste caso será necessário colocar um odómetro nas rodas para a navegação entre ímãs, e controlar a correcção de trajectória indicada pelo sensor na vi-

zinhança de cada íman.

A principal vantagem em relação ao filoguiado é que a instalação dos ímans fica muito mais barata e permite a instalação em solos por exemplo de madeira.

Em contrapartida o custo é superior no sistema de navegação instalado no AGV, tem uma precisão inferior e os ímans no chão atraem as partículas metálicas que possam existir na proximidade.

Cada vez mais os sistemas a implementar nos AGV's são sistemas de espaço aberto, porque permitem uma maior liberdade de trajectos, possibilitando uma fácil alteração da disposição do equipamento (*layout*) da empresa, bastando para isso definir um novo percurso a seguir, e permitem também ter vários AGV's em simultâneo na mesma área, sem problemas.

Estes sistemas necessitam de ter um mapa da área, podendo ser fornecidas ao AGV partes do mapa e vários pontos de referência ao longo do percurso.

O sistema mais usado é o que se baseia em laser e é constituído por um laser rotativo, normalmente com uma frequência de 8Hz, que faz incidir o seu raio em reflectores colocados em pontos estratégicos do percurso, Figura 2.3, permitindo por triangulação determinar a sua posição.

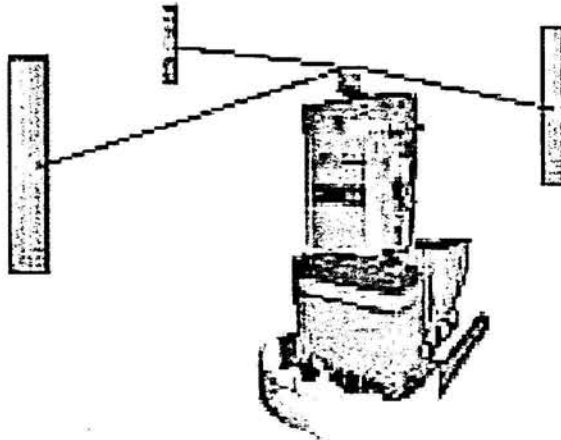


Figura 2.3 Sistema de AGV por laser [2].

O número mínimo de pontos de reflexão em cada posição é três, recomenda-se cinco pontos a uma distância no máximo de trinta metros a partir do AGV.

As desvantagens deste sistema têm a ver com o custo, pois é mais dispendioso, entre 15 a 30%, do que o filoguiado, tem menos precisão e os alvos têm que estar colocados com boa visibilidade para o AGV.

Existem outros sistemas, menos usados, como por exemplo os baseados no GPS (*Global Position System*) que, têm a vantagem de funcionar praticamente por todo o planeta, mas tem um grande problema no que respeita à precisão e ao facto de não poder haver obstáculos entre a antena e o satélite, pelo que só funcionam no exterior das instalações.

Como se pode apurar da revisão efectuada, ainda há muito a desenvolver em relação à navegação dos AGV's.

2.2 *Matching* de imagens

O *matching* de imagens tem uma aplicação muito grande no processamento de imagem. Encontram-se aplicações na medicina, no reconhecimento de padrões, na análise de imagens satélite, etc..

O objectivo é determinar fragmentos correspondentes entre as imagens, independentemente das câmaras de aquisição, das condições de iluminação, podendo ser adquiridas de diferentes secções do objecto, observadas em diferentes alturas temporais ou obtidas de diferentes ângulos.

A determinação dos fragmentos na maior parte das situações é usada para classificar e fazer o registo das imagens.

Todos os algoritmos desenvolvidos e classificados [4]-[7] tem geralmente adoptado uma estratégia baseada em quatro passos principais: selecção de características chave; *matching* dessas características; avaliação das semelhanças e transformação da imagem alvo.

A classificação dos algoritmos existentes em geral é muito complexa, porque envolve muitos critérios de classificação os quais se descrevem a seguir.

Um dos critérios é a dimensão das imagens, uma vez que estas podem ser bidimensionais ou tridimensionais.

Outro critério tem a ver com as propriedades extrínsecas, ou intrínsecas subjacentes às imagens, em que nas propriedades estão implícitos os atributos e o inter-relacionamento entre as imagens.

A ligação existente entre as imagens, por exemplo por interpolação, por aproximação, será mais um critério a ter em conta.

Tem que se ter em conta também o domínio onde estão inseridas as imagens e se se pretende dar importância ao global ou a uma parte da imagem, assim como à elasticidade dos objectos que compõe a imagem, se são objectos rígidos, se são adquiridos por projecção, ou se estão sujeitos a deformações.

A determinação dos parâmetros surge como outro critério, porque podem ser obtidos directamente, se à partida forem conhecidos, ou então podem ser obtidos por busca orientada.

Um critério muito importante é o tipo de interacção que vai existir com o sistema, em que pode haver uma interactividade do utilizador na comparação das imagens, ou essa interacção pode ser semi ou completamente automática.

Os critérios referidos são os mais importantes a tomar em consideração na escolha do algoritmo.

Os algoritmos baseiam-se em métodos de *matching* existentes, tendo os principais como base a correlação cruzada, a transformada de Fourier (Jean Baptiste Joseph Fourier, matemático Francês – 1768 a 1830), o ajuste da superfície tridimensional, o ajuste de pontos e também podem ser baseados na determinação dos momentos.

A correlação cruzada é o método mais popular e é muitas vezes designado por *template matching* [8]-[14].

Na correlação cruzada é seleccionado um objecto de uma imagem de referência, sendo considerado como *template*, e que será utilizado para calcular a correlação cruzada entre este e a imagem a comparar.

A correlação é maximizada, movendo o *template* na imagem a comparar. A posição do máximo determina assim a transformação geométrica existente entre a imagem a comparar e a de referência.

Este processo é feito de uma forma exaustiva e ocupa muito tempo do processador ou seja, segundo Praft [10], para algumas imagens a correlação pode tornar-se muito demorada. Para além disso, imagens com ruído e imagens distorcidas podem mascarar o pico máximo da correlação, sendo bastante difícil localiza-lo. Por outro lado é muito simples e fácil de programar, sendo aplicado muitas vezes no registo de imagens [8], [10], [15] e [16].

O método de *matching* baseado na transformada de Fourier tira partido da grande vantagem de passar para o domínio das frequências, aproveitando toda a experiência existente sobre esta mudança de domínio. Como exemplo, a correlação de duas funções no domínio espacial torna-se na multiplicação complexa do conjugado do espectro no domínio das frequências.

A implementação da correlação pelo método da correlação cruzada, já descrita, torna-se muito eficiente quando efectuada no domínio da frequência recorrendo à transformada rápida de Fourier (*Fast Fourier Transform – FFT*).

Como o espectro de uma imagem tem características especiais que não aparecem no domínio espacial, o método que recorre à transformada de Fourier pode apresentar maior eficiência nos resultados. Por exemplo, o filtro de comparação só da fase (*phase – only matched filtering*) de duas imagens obtidas da mesma cena pode produzir uma relação sinal ruído (*Signal to Noise Ratio – SNR*) muito alta e também um pico bem definido no máximo da correlação, e assim o *matching* do espectro da fase não tem os mesmos problemas da correlação cruzada em detectar o máximo [17] e [18].

Este método tem um problema, pois quando uma das imagens é afectada por uma rotação torna-se insuficiente para fazer o *matching*.

Existe uma técnica que permite fazer o *matching* de uma imagem bidimensional com uma translação, rotação e escala aplicada em relação à imagem de referência. Essa

técnica é designada por algoritmo de *matching* baseado na transformada de Fourier-Mellin (Robert Hjalmar Mellin, matemático Finlandês – 1854 a 1933).

Esta técnica é composta por dois processos sequenciais. Primeiro calcula-se para cada imagem um descritor invariante e a seguir faz-se o *matching* desses descritores.

O descritor é determinado reamostrando o espectro da amplitude (Fourier) das imagens, que está em coordenadas cartesianas, em coordenadas logarítmicas – polares e apresentando-o estas numa grelha rectangular [19]-[21], designada transformada de Fourier – Mellin [22]-[24].

O descritor é invariante para a translação, enquanto que a rotação e a escala estão representadas na grelha rectangular, em que a abcissa representa o eixo angular e a ordenada o eixo radial [25] - [40].

O descritor da imagem é designado por descritor invariante de Fourier – Mellin.

A seguir, aplica-se um filtro de comparação só de fase linear (*symmetric phase-only matched filtering*) [41][42] aos descritores de Fourier – Mellin das duas imagens.

Quando as imagens são idênticas obtemos um pico máximo muito fino na correlação obtendo assim a translação.

No capítulo 3, este método será apresentado de forma pormenorizada.

Um outro método de *matching* é o designado por ajuste da superfície tridimensional (3-D *surface fitting*), usado muitas vezes no registo e combinação de imagens 3-D médicas.

Este método pode ser descrito em cinco passos [43]. Primeiro calibra-se o sistema de aquisição de imagem de forma a que a dimensão do *voxel* seja cuidadosamente reconhecida. A seguir retiram-se as formas comuns das imagens, normalmente de uma forma interactiva. No passo seguinte estima-se o registo dos parâmetros minimizando a distância média entre as formas retiradas. No quarto passo move-se a imagem de forma a ajustar as coordenadas e por fim é feita a validação. Esta validação normalmente é feita por inspecções cuidadas para verificar a ausência de marcas externas.

A principal vantagem deste método é a simplicidade do algoritmo, mas por outro lado existem várias desvantagens, como por exemplo a ineficácia do processamento e, principalmente, a dificuldade de extrair com precisão superfícies 3-D, a falta de imunidade ao ruído, e o facto de frequentemente o utilizador ter que intervir no processo.

O método de ajuste de pontos (*point fitting*), ao contrário de alguns dos métodos já referidos, não faz nenhuma busca exaustiva de parâmetros no espaço. Na maioria das situações esses pontos são encontrados manualmente de uma imagem para a outra.

Todos os parâmetros/resultados, pretendidos serão obtidos resolvendo um certo número de equações em torno dos pontos marcados [44] - [62].

Quando esses pontos são eficazmente marcados nas duas imagens, este método apresenta-se como sendo o mais rápido e o mais robusto, mesmo na presença de ruído. Mas se esses pontos não forem marcados devidamente, o que acontece frequentemente com a marcação automática, então os parâmetros estimados são fortemente alterados, obtendo-se resultados errados.

O ultimo método que iremos rever tem como princípio a determinação do momento, é muitas vezes usado para determinar a posição, orientação e a alteração de dimensão de um objecto numa imagem.

Os diferentes tipos de momentos podem ser calculados através da atribuição de diferentes expoentes na integração, sendo este o número de ordem do momento. Podem também ser obtidos recorrendo a uma mudança de domínio, como por exemplo, do domínio espacial para o domínio da frequência.

Uma imagem, em princípio, pode ser reconstruída a partir de um conjunto de momentos de um determinado tipo, mas quanto mais elevado for o número de ordem dos momentos, maior é a sensibilidade ao ruído e à distorção que possa existir, por isso a maior parte das vezes este método, para momentos de ordens elevadas, é aplicado a imagens binárias.

A reconstrução de sistemas a partir dos momentos torna-se falível e com uma aplicação prática pouco acentuada nesta área, excepto em alguns estudos efectuados em com-

pressão de imagem.

No entanto os momentos de ordem mais baixa retêm algumas propriedades da imagem que permitem caracterizar, avaliar e manipular a informação das imagens.

Abu-Mostafa e Praltis [68] descobriram o *matching* baseado no momento ou na invariabilidade do momento, cujo poder discriminativo é muito restrito, principalmente porque a imagem inteira encontra-se num número muito restrito de momentos.

Para um maior aprofundamento no *matching* baseado no momento, sugere-se a bibliografia indicada [63] - [83].

Concluimos assim a revisão dos principais métodos de *matching* existentes.

A selecção do método a usar depende do tipo e das características das imagens a comparar.

Estas características podem ser a textura, a cor, a forma, a posição, a amplitude, a fase, o histograma, etc.. Qualquer uma ou mais características podem ser usadas. O critério principal para a escolha da característica é a inalteração desta na presença de uma transformação linear (translação, rotação, escala, inclinação).

2.3 Aplicação de *matching* de imagens

O *matching* de imagens tem um sem número de aplicações num sem número de áreas.

Neste sub-capítulo vai ser feita uma breve referência a algumas dessas aplicações, bem com algumas das aplicações do algoritmo de *matching* baseado na transformada de Fourier – Mellin já referida, e que se encontra pormenorizada no capítulo 3.

O *matching* de imagem é usado nomeadamente nas seguintes áreas: a nível militar, em radares e reconhecimento de alvos por satélite; na pesquisa e classificação de imagens, por exemplo numa biblioteca de imagens, no meio comercial (catálogos de moda, jornalismo), na cultura (museus, galerias de arte), na educação (leitura de slides, gráficos), na *web* (filtrar as imagens próprias só para adultos), na segurança (detectando grandes movimentações por exemplo num estádio de futebol), na arquitectura e desenhos de engenharia

(bases de dados de CAD), entre muitas outras.

As aplicações referidas podem usar um ou mais do que um dos métodos de *matching* descritos anteriormente.

Em algumas aplicações referidas a seguir, como o registo de imagens médicas, a observação de imagens a partir do espaço, o reconhecimento de impressões digitais e por fim a identificação de vários objectos, tem sido aplicado com bons resultados o algoritmo de *matching* baseado na transformada de Fourier – Mellin.

Registo de imagens médicas

Nos diagnósticos clínicos, muitas vezes é necessário comparar imagens de um paciente retiradas em diferentes alturas temporais. Esta comparação só pode ser feita quando as imagens estão correctamente registadas [84]-[87].

Podemos ver um exemplo de duas imagens de ressonância magnética à cabeça de um paciente, Figura 2.4, em que em a) temos uma fatia (*slice*) designada por T1 e em b) temos outra designada por T2. O objectivo é registar as duas imagens realçando uma parte da imagem com interesse.

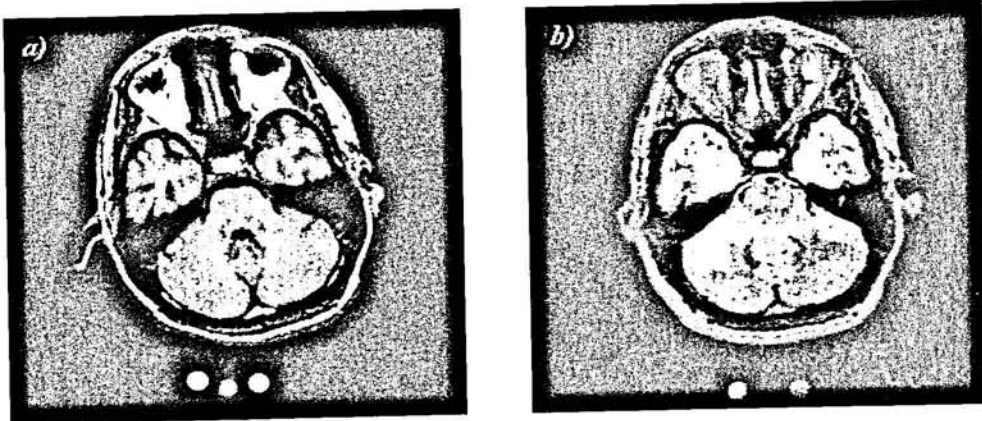


Figura 2.4 Imagens de ressonância magnética à cabeça de um paciente; a) Fatia (*slice*) T1 e b) Fatia (*slice*) T2 [6].

Considera-se a imagem T1 como sendo a imagem de referência.

Aplicando-se o algoritmo de *matching* baseado na transformada de Fourier – Me-

lin à imagem T2 e T1, obtemos como resultado a Figura 2.5. O pico máximo corresponde a um ângulo de rotação de 6° e a um factor de escala de 100% [6].

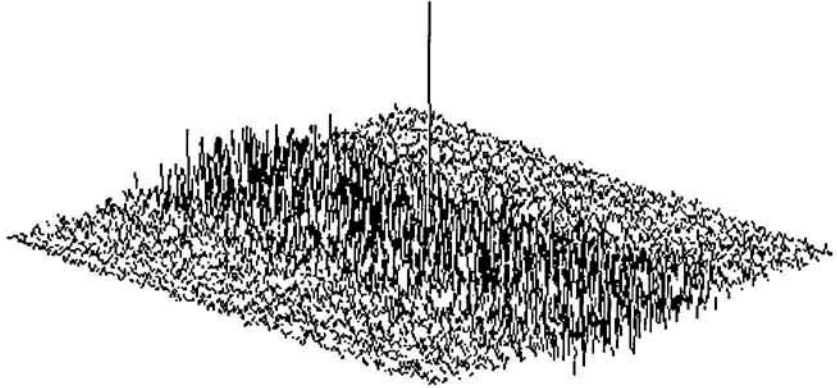


Figura 2.5 Resultado da aplicação do algoritmo de *matching* baseado na transformada de Fourier – Mellin às imagens da Figura 2.4. O pico corresponde a um ângulo de 6° e a uma escala de 100% [6].

Seguidamente roda-se a imagem T2 de um ângulo de -6° e aplica-se às duas imagens o filtro de comparação só de fase linear, obtendo assim a Figura 2.6, com o pico máximo a definir a translação existente entre as duas imagens (5, -4).

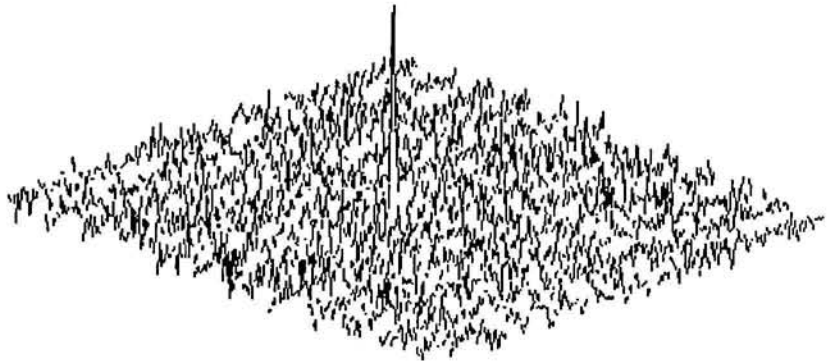


Figura 2.6 Saída do filtro de comparação só de fase linear aplicado às duas imagens da Figura 2.4, mas com T2 rodado de -6° . O pico indica o deslocamento entre as duas imagens (5, -4) [6].

Marca-se a região do cérebro da Figura 2.4 b) como região de interesse, indicada pela linha mais acentuada na Figura 2.7 b).

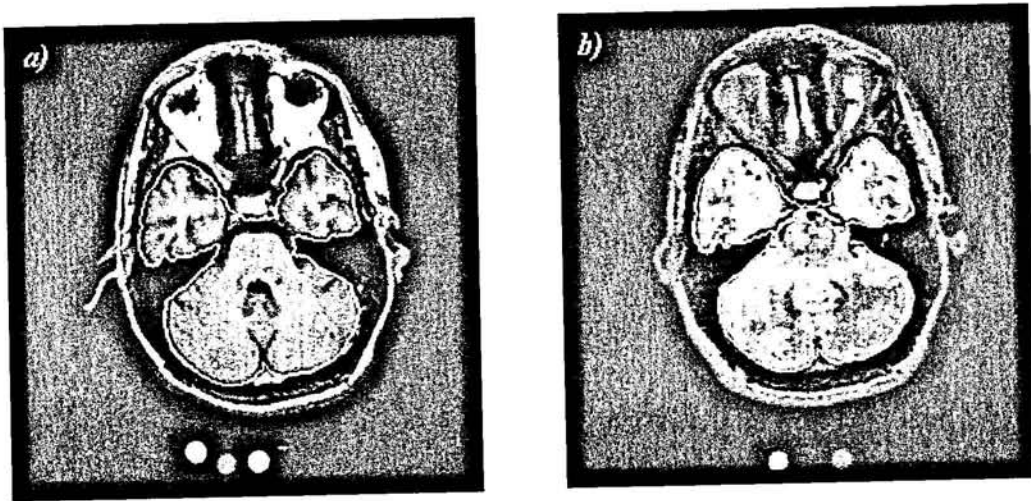


Figura 2.7 Zona do cérebro demarcada nas duas imagens da Figura 2.4 [6].

A mesma região anatómica da Figura 2.4 a) pode ser correctamente localizada. O resultado está indicado na Figura 2.7 a) pela linha mais acentuada.

Observação de imagens a partir do espaço

A observação a partir do espaço permite analisar por exemplo a desflorestação numa determinada zona, o trânsito, a alteração do meio envolvente de um ponto específico, entre muitos outros.

O processamento automático deste tipo de observação envolve a avaliação, o inventário e a identificação das estruturas envolventes.

A observação cronológica, a comparação com diferentes sensores e a síntese das cores que compõe as imagens requerem o registo das diferentes imagens de um determinado local [88]-[90].

Na Figura 2.8 temos duas imagens de satélite do mesmo local adquiridas em tempos diferentes.

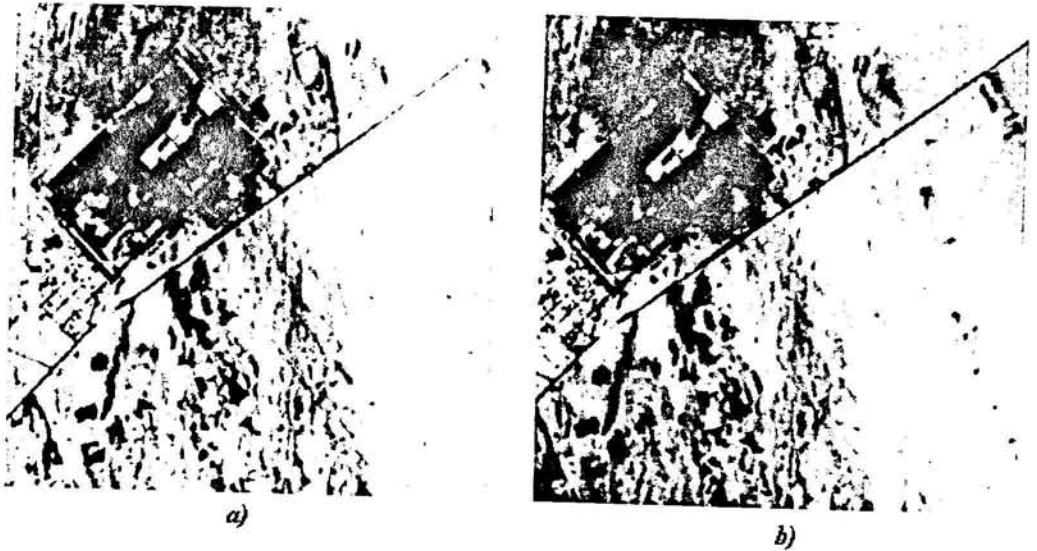


Figura 2.8 Imagens de satélite do mesmo local em tempos diferentes [6].

Como a altitude da plataforma de aquisição de imagem se encontra a uma distância muito grande implica que as variações topográficas sejam muito pequenas, as fotografias têm projecções aproximadamente paralelas e a distorção entre as imagens pode ser ignorada.

Se pretendermos determinar a diferença geométrica que existe entre as duas imagens, podemos aplicar o algoritmo de *matching* baseado na transformada de Fourier – Mellin e obtemos o resultado apresentado na Figura 2.9. Daqui obtemos o ângulo de rotação e o factor de escala, entre as imagens, de 7° e de 100% respectivamente.

Depois de rodar a Figura 2.8 b) de um ângulo de -7° , obtemos a translação, aplicando um filtro de comparação só de fase. A Figura 2.10 apresenta o pico que nos indica a translação entre as imagens $(-12, 1)$.

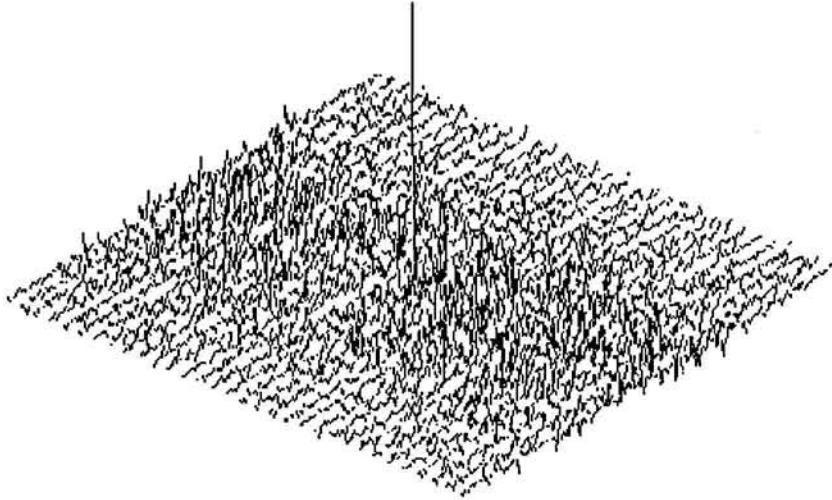


Figura 2.9 Resultado da aplicação do algoritmo de *matching* baseado na transformada de Fourier – Mellin às duas imagens da Figura 2.8 [6].

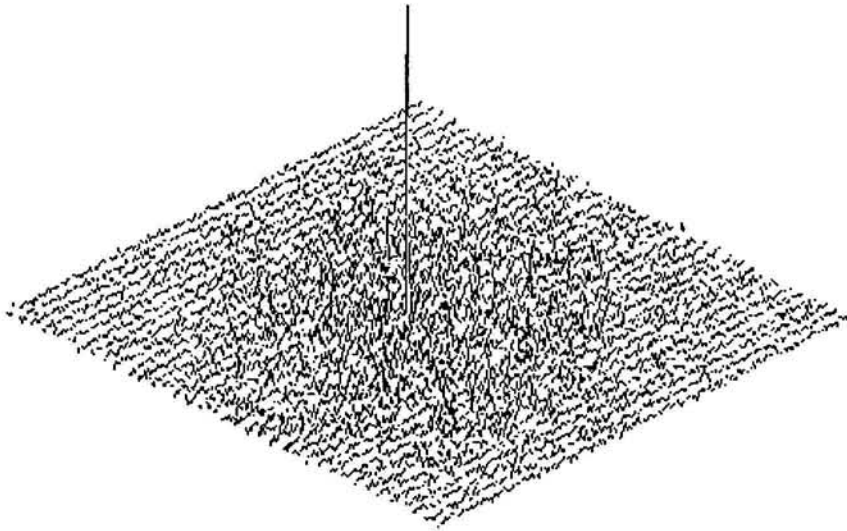


Figura 2.10 Saída do filtro de comparação só de fase aplicado às duas imagens da Figura 2.8, mas com a imagem b) rodado de -7° . O pico indica o deslocamento entre as duas imagens $(-12, 1)$ [6].

Obtemos assim a diferença geométrica entre as duas imagens da Figura 2.8, que é

respectivamente 7° , -12 e 1 para rotação, translação no eixo do xx e translação no eixo do yy .

Reconhecimento de impressões digitais

As impressões digitais são consideradas como um dos meios mais seguros de identificação dos indivíduos.

O reconhecimento das impressões digitais baseia-se na identificação e comparação de pequenos pormenores existentes nestas.

Estes pormenores são compostos por pequenas irregularidades representadas por sulcos marcados nos dedos, sendo estes únicos por indivíduo.

Num sistema de reconhecimento de impressões digitais primeiro é feita uma segmentação e uma binarização da imagem. Posteriormente, todos os pormenores são identificados e é registada a localização de cada um num gráfico bidimensional.

Sempre que for necessário fazer o *matching* de uma impressão digital, recorre-se ao gráfico com todos os pormenores dessa impressão e procura-se as semelhanças entre esse e todos os outros existentes.

A dificuldade neste sistema de reconhecimento está directamente relacionada com a correcta ou incorrecta identificação de todos os pormenores [91] - [95].

Existe uma vasta gama de sistemas de identificação de impressões digitais baseadas em correlação óptica, que no entanto só funcionam se as duas impressões tiverem a mesma orientação.

Mais uma vez pode ser aplicado aqui o algoritmo da transformada de Fourier - Mellin, porque consegue reconhecer as impressões digitais mesmo que tenham orientações diferentes.

As figuras a seguir apresentadas ilustram um exemplo, seguindo os passos já referenciados nas aplicações anteriores, e tem como objectivo determinar uma impressão digital, Figura 2.11, de entre quatro existentes, Figura 2.12, ou seja é aplicado o *matching*, através do algoritmo da transformada de Fourier - Mellin, entre a imagem de referência e

cada uma das outras. Podemos concluir analisando a Figura 2.13, que só a impressão digital da Figura 2.12 a) pertence ao mesmo dedo do que a impressão da Figura 2.11.



Figura 2.11 Impressão digital [6].



Figura 2.12 Conjunto de impressões digitais incluindo a da Figura 2.11 [6].

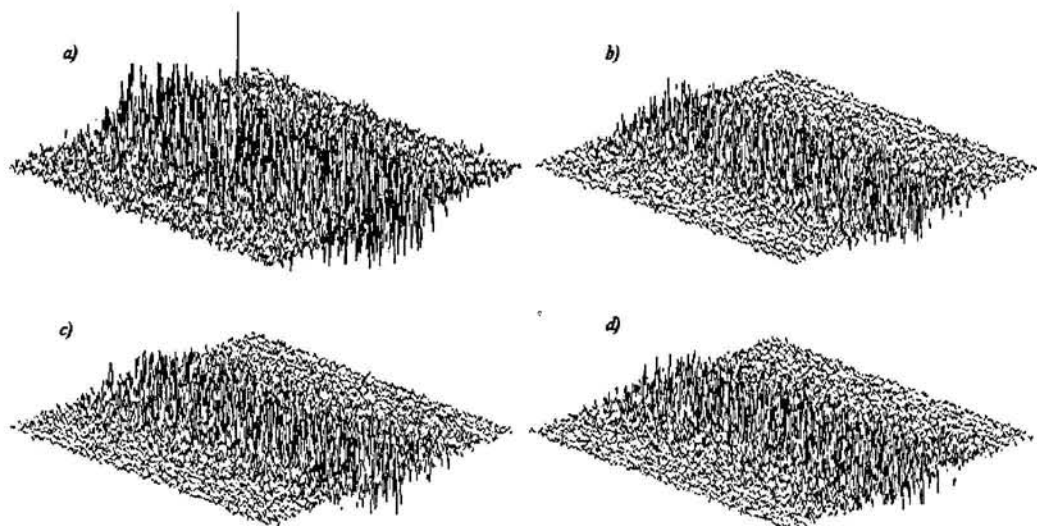


Figura 2.13 Resultado da aplicação do algoritmo de *matching* baseado na transformada de Fourier – Mellin entre a imagem da Figura 2.11 e cada uma das imagens da Figura 2.12 da a) até à d) respectivamente [6].

Identificação de objectos

A visão robótica muitas vezes requer a identificação e análise de alguns objectos simples que se encontram sobrepostos, por forma a individualizá-los.

Vários sistemas identificam esses objectos, fazendo o *matching* dos seus contornos, mas a correcta segmentação desses objectos parte de algumas suposições e nem sempre são muito viáveis na prática. Na verdade muitas demonstrações de sistemas de visão usam imagens ideais, obtidas com um controlo apertado ou então usando modelos criados por computador [96]-[98].

Este problema pode ser resolvido recorrendo, uma vez mais, ao algoritmo da transformada de Fourier – Mellin.

Suponhamos que temos a Figura 2.14 a) composta por uma faca e uma chave e pretende-se identificar a chave, Figura 2.14 b).

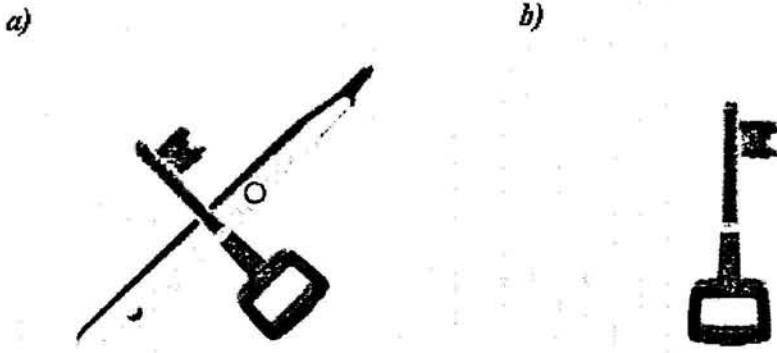


Figura 2.14 Imagens de objectos; a) faca e chave, b) *template* da chave [6].

Fazendo o *matching*, baseado no algoritmo da transformada de Fourier – Mellin, entre as duas figuras, obtemos a Figura 2.15. O pico mais pronunciado indica-nos a presença da chave e qual a sua localização.

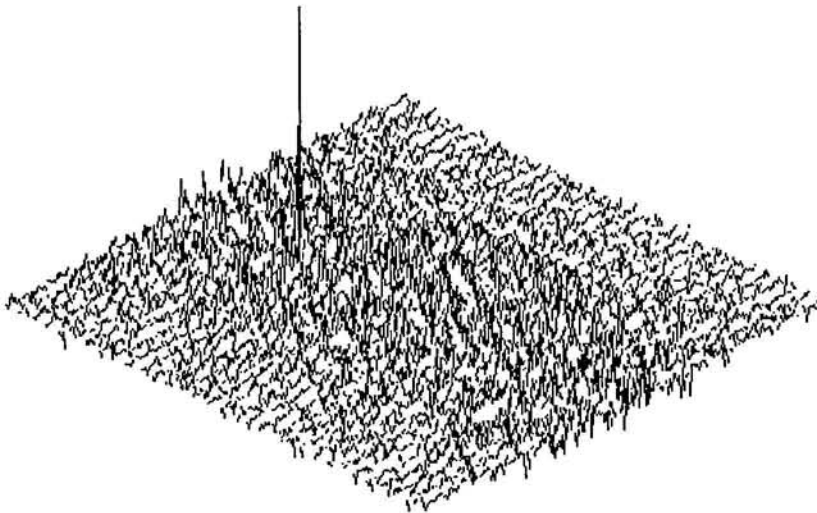


Figura 2.15 Resultado da aplicação do algoritmo de *matching* baseado na transformada de Fourier – Mellin às imagens da Figura 2.14 [6].

A grande vantagem deste sistema é a grande imunidade ao ruído e os baixos requisitos de processamento.

São de referir como desvantagem o facto do processo de *matching* se basear na imagem total, tornando-se difícil localizar o objecto se houver mais do que um igual na mesma imagem, ou então a situação em que o fundo da imagem é diferente em relação à imagem de referência.

Concluimos assim a análise de algumas aplicações que utilizam o *matching* de imagens, principalmente as que recorrem ao *matching* de imagens baseada no algoritmo da transformada de Fourier – Mellin.

Capítulo 3

Metodologia seguida

A metodologia seguida nesta dissertação tem como base um dos métodos referidos no capítulo anterior para a detecção de *matching* entre duas imagens.

Os métodos utilizadas na detecção de *matching* dividem-se em duas grandes classes: os métodos no domínio original e os métodos num domínio transformado, nomeadamente no domínio das frequências.

Neste trabalho optou-se por usar um método num domínio transformado, domínio das frequências, assim o método seleccionado baseia-se na transformada de Fourier, mais concretamente na transformada de Fourier – Mellin [99] - [101], e no método da correlação cruzada, no domínio da frequência.

Posto isto, a tarefa seguinte é caracterizar o problema ao qual esta dissertação pretende dar resposta.

Torna-se oportuno para este trabalho fazer uma breve revisão às propriedades mais relevantes da transformada de Fourier, assim como é apresentado pormenorizadamente o algoritmo de *matching* baseado na transformada de Fourier – Mellin.

3.1 Caracterização do Problema

O método de navegação utilizado, como foi referido no capítulo anterior, baseia-se na determinação da posição relativa de duas imagens consecutivas do terreno sobre o qual o veículo se move.

O problema subjacente resulta do facto de as imagens consecutivas não poderem ser muito próximas, pois daí poderão resultar erros acumulados que inviabilizam o processo, nem muito afastadas, pois a sua posição relativa vai ser determinada a partir da correlação cruzada entre as imagens, para além do mais a velocidade do veículo e a precisão no movimento, fica dependente da dimensão das imagens e da relação entre pixel e centímetro.

Temos ainda outras situações tais como: o terreno poder ter alterações de iluminação ao longo do trajecto; a câmara de aquisição de imagens estar sujeita vibrações; surgirem novos objectos entre imagens consecutivas, tornando difícil a determinação da posição relativa.

Um problema adicional resulta do facto de duas imagens consecutivas serem realmente duas imagens distintas, sem qualquer correspondência ao nível do pixel, permitindo a observação de deslocamentos correspondentes à fracção de pixel, ou seja, não existe nenhum pixel comum às duas imagens.

Duas imagens consecutivas, I_1 e I_2 podem estar relacionadas por uma translação δ e por uma rotação θ , Figura 3.1.

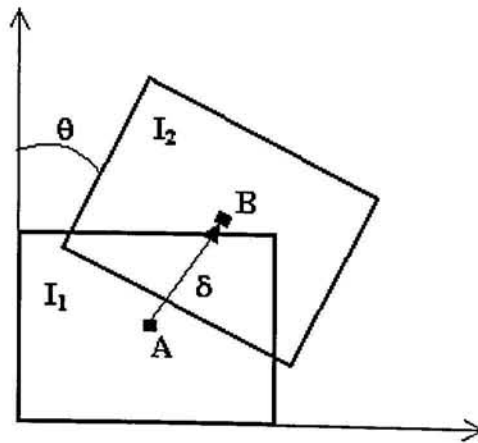


Figura 3.1 Imagens consecutivas relacionadas por uma translação δ e uma rotação θ .

A determinação da posição relativa entre duas imagens consecutivas passa pela determinação da translação δ e da rotação θ obtendo assim, o movimento do veículo e a sua orientação.

3.2 Transformada de Fourier

Uma vez que o método usado neste trabalho baseia-se na transformada de Fourier, considera-se necessário fazer uma breve revisão às propriedades mais relevantes desta, tendo em conta principalmente a transformada discreta de Fourier bidimensional.

Como esta goza da propriedade da separabilidade, propriedade demonstrada mais adiante, que permite decompor a transformada discreta bidimensional em transformadas unidimensionais, algumas das propriedades vão ser verificadas para transformadas unidimensionais, simplificando bastante a sua demonstração.

A transformada discreta de Fourier bidimensional é caracterizada por ser uma ferramenta matemática com grande aplicação no processamento digital de imagens, pois simplifica bastante o processamento fazendo uma mudança do domínio do tempo ou espaço (x, y) para o domínio da frequência (u, v) . Neste domínio poderemos facilmente aplicar filtros e todo o conhecimento existente, por forma a processar a imagem atingindo o objectivo pretendido, por exemplo, realce, restauração, descodificação da imagem, posição relativa, etc..

3.2.1 Transformada de Fourier Discreta – DFT

A transformada de Fourier discreta (DFT) [102] e [103], de um sinal discreto unidimensional $f(x)$ de comprimento N é uma amostragem da sua transformada de Fourier $F(\omega)$ em N pontos igualmente espaçados no intervalo $[0, 2\pi]$, Figura 3.2,

$$\omega_u = \frac{2\pi u}{N}, u = 0 \dots N-1,$$

$$F(e^{j\omega}) = \sum_{x=0}^{N-1} f(x)e^{-j\omega x},$$

$$F(u) = F(e^{j\omega_u}) = \sum_{x=0}^{N-1} f(x) e^{-\frac{j2\pi xu}{N}}, \quad u = 0 \dots N-1. \quad (1)$$

A transformada inversa é

$$f(x) = \frac{1}{N} \sum_{u=0}^{N-1} F(u) e^{\frac{j2\pi xu}{N}}, \quad x = 0 \dots N-1. \quad (2)$$

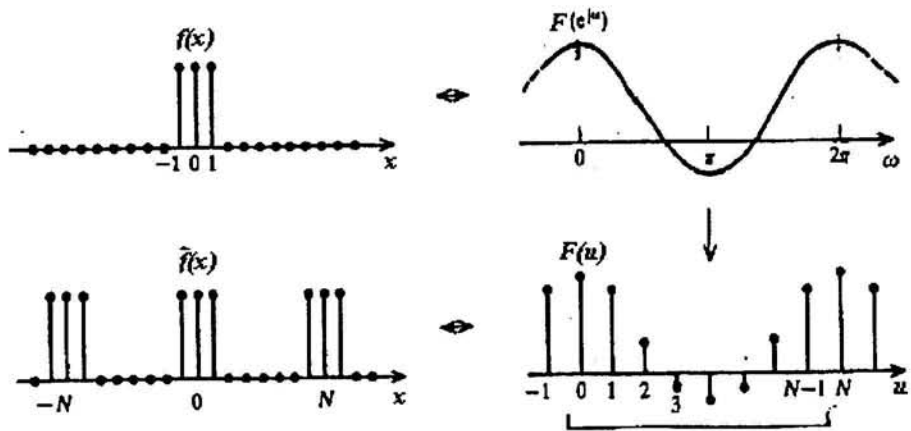


Figura 3.2 DFT com $F(u) = F(e^{j\omega}) \Big|_{\omega=\omega_u=\frac{2\pi u}{N}}$ [103].

Podemos verificar que (1) e (2) são uma transformada e a sua transformada inversa, substituindo em (2) $F(u)$ por (1) fica

$$f(x) = \frac{1}{N} \sum_{u=0}^{N-1} \sum_{r=0}^{N-1} f(r) e^{-\frac{j2\pi ru}{N}} e^{\frac{j2\pi xu}{N}}, \quad x = 0 \dots N-1,$$

$$f(x) = \frac{1}{N} \sum_{r=0}^{N-1} f(r) \sum_{u=0}^{N-1} e^{\frac{j2\pi(x-r)u}{N}}, \quad x = 0 \dots N-1$$

e como o somatório em u vale

$$\sum_{u=0}^{N-1} e^{\frac{j2\pi(x-r)u}{N}} \Bigg| \begin{array}{l} 0, \text{ se } r \neq x \\ N, \text{ se } r = x \end{array}$$

resulta a identidade

$$f(x) = \frac{1}{N} f(r)N \Big|_{r=x} = f(x).$$

As expressões (1) e (2) podem ser escritas de uma forma mais compacta, por forma a simplificar a sua utilização, introduzindo a notação

$$W_N = e^{-j\frac{2\pi}{N}} = \cos \frac{2\pi}{N} - j \sin \frac{2\pi}{N},$$

e ficam respectivamente

$$F(u) = \sum_{x=0}^{N-1} f(x)W_N^{xu}, \quad u = 0 \dots N-1, \quad (3)$$

$$f(x) = \frac{1}{N} \sum_{u=0}^{N-1} F(u)W_N^{-xu}, \quad x = 0 \dots N-1. \quad (4)$$

A transformada de Fourier discreta bidimensional, [104] e [105], é por sua vez

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)W_M^{ux}W_N^{vy}, \quad u = 0 \dots M-1, v = 0 \dots N-1, \quad (5)$$

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v)W_M^{-ux}W_N^{-vy}, \quad x = 0 \dots M-1, y = 0 \dots N-1. \quad (6)$$

Quando $M=N$ isto é, temos número igual de amostras em cada dimensão,

$$F(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y)W_N^{(ux+vy)}, \quad u, v = 0 \dots N-1, \quad (7)$$

$$f(x, y) = \frac{1}{N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u, v)W_N^{-(ux+vy)}, \quad x, y = 0 \dots N-1. \quad (8)$$

3.2.2 Algumas propriedades importantes da transformada de Fourier discreta

Como já foi referido, muitas propriedades da transformada de Fourier unidimensional podem ser estendidas às transformadas bidimensionais [106], uma vez que a transformada bidimensional pode ser separada em duas transformadas unidimensionais.

Separabilidade

As equações (5) e (6) da transformada de Fourier discreta bidimensional podem ser escritas separando as operações das linhas em relação às colunas,

$$F(u, v) = \sum_{x=0}^{M-1} W_M^{ux} \sum_{y=0}^{N-1} f(x, y) W_N^{vy}, \quad u = 0 \dots M-1, v = 0 \dots N-1, \tag{9}$$

$$f(x, y) = \frac{1}{M} \sum_{u=0}^{M-1} W_M^{-ux} \frac{1}{N} \sum_{v=0}^{N-1} F(u, v) W_N^{-vy}, \quad x = 0 \dots M-1, y = 0 \dots N-1. \tag{10}$$

Desta forma podemos obter a transformada de Fourier bidimensional a partir da aplicação sucessiva da transformada de Fourier unidimensional. Esta situação torna-se evidente se escrevermos a equação (9) da seguinte forma

$$F(u, v) = \sum_{x=0}^{M-1} F(x, v) W_M^{ux}, \tag{11}$$

em que

$$F(x, v) = \sum_{y=0}^{N-1} f(x, y) W_N^{vy}. \tag{12}$$

Para cada valor de x a equação (12) é uma transformada unidimensional com índice $v = 0, 1, \dots, N-1$.

Resulta assim que a transformada $F(x)$ pode ser calculada através da transformada de Fourier unidimensional de $f(x)$ para cada linha, e que o resultado pretendido, $F(u, v)$, pode ser obtido calculando a transformada de Fourier unidimensional, de $F(x, v)$ para cada coluna, como é indicado na equação (11) e esquematizado na Figura 3.3.

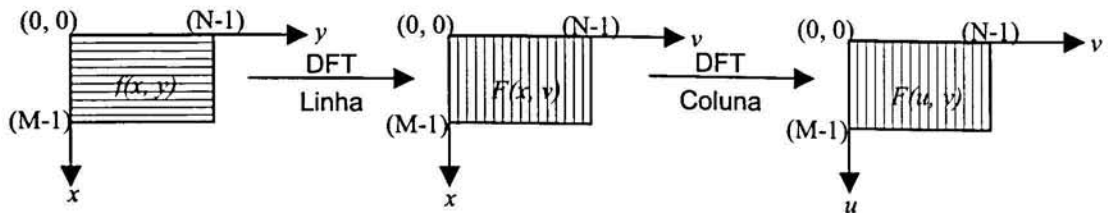


Figura 3.3 DFT bidimensional a partir da DFT unidimensional [104].

O mesmo resultado pode ser obtido se fizermos em primeiro lugar a transformada de $f(x, y)$ para cada coluna e depois a transformada do resultado para cada linha.

Para a transformada inversa (10) seria aplicado o mesmo raciocínio.

Translação

A propriedade translação da transformada de Fourier é

$$f(x - x_0, y - y_0) \Leftrightarrow F(u, v) W_M^{ux_0} W_N^{vy_0} \quad (13)$$

e

$$f(x, y) W_M^{-ux_0} W_N^{-vy_0} \Leftrightarrow F(u - u_0, v - v_0). \quad (14)$$

Podemos verificar a propriedade para a equação (13) fazendo

$$\begin{aligned} F(u, v) W_M^{ux_0} W_N^{vy_0} &= \left(\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) W_M^{ux} W_N^{vy} \right) W_M^{ux_0} W_N^{vy_0} \\ &= \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) W_M^{u(x+x_0)} W_N^{v(y+y_0)} \\ &= \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x - x_0, y - y_0) W_M^{ux} W_N^{vy}. \end{aligned}$$

e a equação (14) seria de modo análogo.

Um aspecto importante que resulta da equação (13) é que a translação de $f(x, y)$ não afecta a amplitude da sua transformada de Fourier, mas apenas a sua fase

$$\left| F(u, v) W_M^{ux_0} W_N^{vy_0} \right| = |F(u, v)|$$

e

$$\text{Arg} \left[F(u, v) W_M^{ux_0} W_N^{vy_0} \right] = \text{Arg} [F(u, v)] - 2\pi \left(\frac{ux_0}{M} + \frac{vy_0}{N} \right).$$

Esta situação pode ser verificada na Figura 3.4 para um sinal unidimensional $f(x)$.

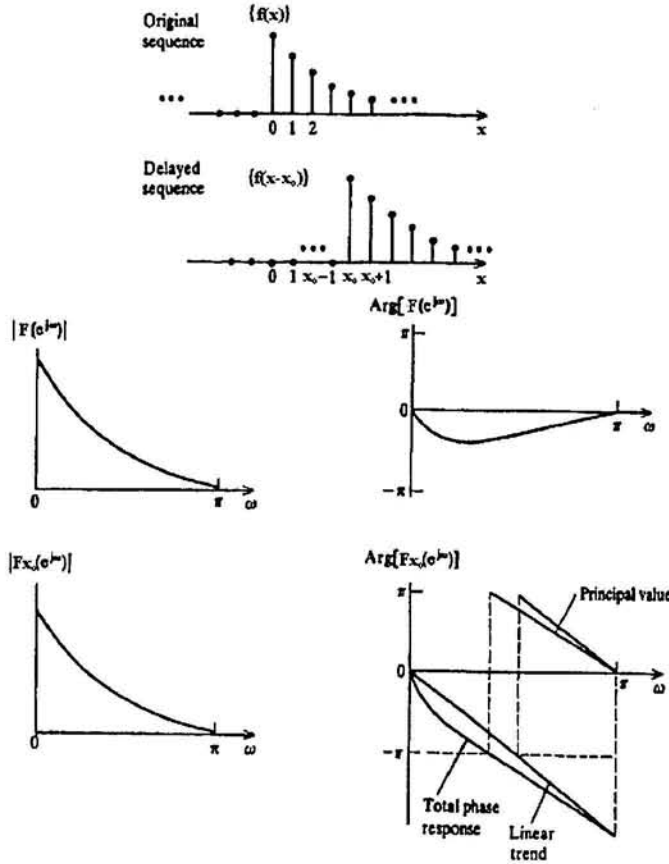


Figura 3.4 Comparação do espectro unidimensional da amplitude e da fase para a sequência original e com translação x_0 [103].

O sinal submetido a uma translação de x_0 tem a sua fase alterada por termo $-\frac{2\pi}{M}ux_0$, que varia linearmente com a frequência u .

Periodicidade e simetria do conjugado

A transformada de Fourier discreta e a sua inversa são periódicas de período M para as linhas e de período N para as colunas, ou seja

$$F(u, v) = F(u + M, v) = F(u, v + N) = F(u + M, v + N) \tag{15}$$

e

$$f(x, y) = f(x + M, y) = f(x, y + N) = f(x + M, y + N). \tag{16}$$

A verificação pode ser feita se substituirmos em (5) u por $u+M$ e v por $v+N$.

Temos

$$F(u+M, v+N) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) W_M^{x(u+M)} W_N^{y(v+N)}$$

$$F(u+M, v+N) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) W_M^{ux} W_N^{vy} W_M^{xM} W_N^{yN}$$

e como

$$W_M^{xM} W_N^{yN} = 1, \quad x = 0 \dots M-1, \quad y = 0 \dots N-1$$

vem

$$F(u+M, v+N) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) W_M^{ux} W_N^{vy} = F(u, v).$$

A verificação no domínio do espaço $f(x, y)$ seria idêntica.

A transformada de Fourier do conjugado da imagem é $F^*(-u, -v)$.

Podemos ver que

$$f(x) \Leftrightarrow F(u),$$

$$f^*(x) \Leftrightarrow F^*(-u),$$

$$F(u) = \sum_{x=0}^{N-1} f(x) W_N^{xu} \text{ e como}$$

$$(W_N^\alpha)^* = W_N^{-\alpha}, \text{ vem}$$

$$\sum_{x=0}^{N-1} f^*(x) W_N^{xu} = \left(\sum_{x=0}^{N-1} f(x) W_N^{-xu} \right)^* = F^*(-u).$$

Assim se $f(x, y)$ for puramente real temos

$$f(x, y) = f^*(x, y) \text{ e}$$

$$F(u, v) = F^*(-u, -v) \quad (17)$$

ou ainda

$$|F(u, v)| = |F(-u, -v)|, \quad (18)$$

$$\text{Arg}[F(u, v)] = -\text{Arg}[F(-u, -v)] \quad (19)$$

em que, $F^*(u, v)$ é o complexo conjugado de $F(u, v)$.

Verifiquemos a implicação da equação (15) e da equação (18) na amplitude da transformada de Fourier.

Por forma a tornar a verificação mais simples, vamos analisar na situação unidimensional, onde

$$F(u) = F(u + N) \text{ e } |F(u)| = |F(-u)|.$$

A propriedade da periodicidade indica que $F(u)$ tem o período de comprimento N e a propriedade da simetria mostra que a amplitude da transformada está centrada na origem, ver Figura 3.5 a).

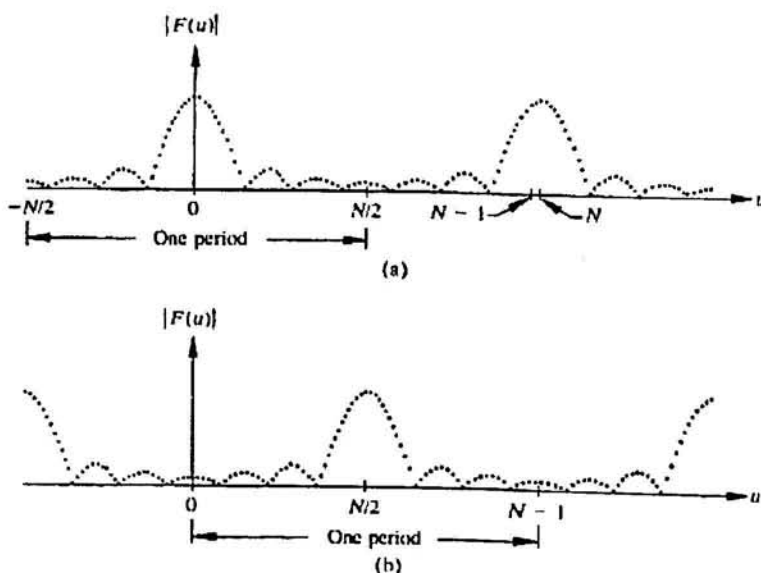


Figura 3.5 Ilustração da propriedade da periodicidade da transformada de Fourier [104].

O facto da transformada discreta de Fourier ser calculada para todos os valores de u no intervalo $[0, N-1]$ e produzir um sinal como o representado na Figura 3.5 a), mostra que os valores da amplitude da transformada no intervalo $[(N/2)+1, N-1]$ é o reflexo dos valores na metade do período do lado esquerdo da origem.

Sendo assim para representar um período inteiro basta deslocar a origem da transformada para $u = N/2$, Figura 3.5 b), multiplicando $f(x)$ por $(-1)^x$.

Rotação

A propriedade da rotação diz-nos que se sujeitarmos $f(x, y)$ a uma rotação θ_0 , a transformada de Fourier $F(u, v)$ também é afectada dessa rotação θ_0 , e que o contrário também se verifica.

Se fizermos uma mudança das coordenadas, por reamostragem, de cartesianas para polares, fica

$$f(r, \theta) \Leftrightarrow F(\omega, \phi) \quad 0 \leq r \leq M', \quad 0 \leq \theta \leq N' \text{ e } 0 \leq \omega \leq M', \quad 0 \leq \phi \leq N'$$

com

$$x = r \cos \theta, \quad y = r \sin \theta \quad -\infty < x, y < \infty$$

e

$$u = \omega \cos \phi, \quad v = \omega \sin \phi, \quad -\infty < u, v < \infty$$

então temos

$$f(r, \theta + \theta_0) \Leftrightarrow F(\omega, \phi + \theta_0). \quad (20)$$

Podemos verificar a equação (20) se substituirmos na equação (5) as variáveis x, y, u e v pelas suas coordenadas polares juntamente com a rotação pretendida (θ_0) e fica

$$\begin{aligned} \mathfrak{F}[f(r, \theta + \theta_0)] &= \sum_{r=0}^{M'-1} \sum_{\theta=0}^{N'-1} f(r, \theta + \theta_0) e^{-j2\pi \left[\frac{\omega \cos \phi \cos(\theta + \theta_0)}{M} + \frac{\omega \sin \phi \sin(\theta + \theta_0)}{N} \right]} \\ &= \sum_{r=0}^{M'-1} \sum_{\theta=0}^{N'-1} f(r, \theta + \theta_0) e^{-j2\pi \frac{\omega r N + \omega r M}{MN} \left[\frac{\cos \phi \cos(\theta + \theta_0) + \sin \phi \sin(\theta + \theta_0)}{p} \right]}, \quad (21) \end{aligned}$$

trabalhando P

$$\begin{aligned}
 P &= \cos \phi \cos(\theta + \theta_0) + \operatorname{sen} \phi \operatorname{sen}(\theta + \theta_0) \\
 &= \cos \phi (\cos \theta \cos \theta_0 - \operatorname{sen} \theta \operatorname{sen} \theta_0) + \operatorname{sen} \phi (\operatorname{sen} \theta \cos \theta_0 + \operatorname{sen} \theta_0 \cos \theta) \\
 &= \cos \phi \cos \theta_0 \cos \theta - \cos \phi \operatorname{sen} \theta_0 \operatorname{sen} \theta + \operatorname{sen} \phi \cos \theta_0 \operatorname{sen} \theta + \operatorname{sen} \phi \operatorname{sen} \theta_0 \cos \theta \\
 &= \frac{1}{2} \cos \theta [\cos(\phi - \theta_0) + \cos(\phi + \theta_0)] - \frac{1}{2} \operatorname{sen} \theta [\operatorname{sen}(\theta_0 - \phi) + \operatorname{sen}(\theta_0 + \phi)] + \\
 &\quad + \frac{1}{2} \operatorname{sen} \theta [\operatorname{sen}(\phi - \theta_0) + \operatorname{sen}(\phi + \theta_0)] + \frac{1}{2} \cos \theta [\cos(\phi - \theta_0) - \cos(\phi + \theta_0)] \\
 &= \cos \theta \cos(\phi - \theta_0) + \operatorname{sen} \theta \operatorname{sen}(\phi - \theta_0)
 \end{aligned}$$

e substituindo na equação (21) P

$$\mathfrak{F}[f(r, \theta + \theta_0)] = \sum_{r=0}^{M'-1} \sum_{\theta=0}^{N'-1} f(r, \theta + \theta_0) e^{-j2\pi \frac{\omega r N + \omega' r M}{MN} [\cos \theta \cos(\phi - \theta_0) + \operatorname{sen} \theta \operatorname{sen}(\phi - \theta_0)]}$$

concluimos que

$$\mathfrak{F}[f(r, \theta + \theta_0)] = \sum_{r=0}^{M'-1} \sum_{\theta=0}^{N'-1} f(r, \theta + \theta_0) e^{-j2\pi \left[\frac{\omega \cos(\phi - \theta_0) r \cos \theta}{M} + \frac{\omega' \operatorname{sen}(\phi - \theta_0) r \operatorname{sen} \theta}{N} \right]}$$

A propriedade Rotação está ilustrada na Figura 3.6.

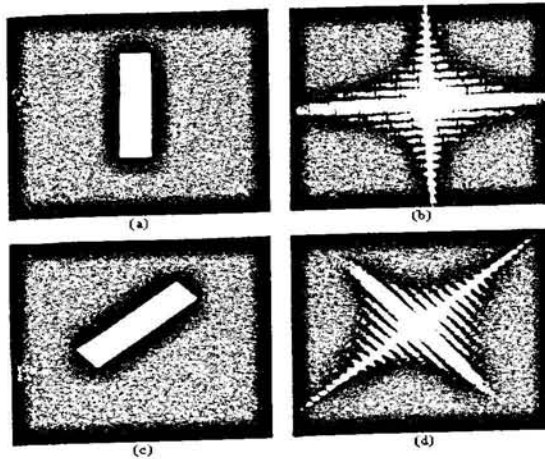


Figura 3.6 Propriedade Rotação da Transformada Fourier: a) Imagem original; b) Espectro da imagem original; c) Imagem rodada; d) Espectro resultante [104].

Convolução

A convolução e a correlação, que será abordada posteriormente, têm uma importância fundamental em processamento de imagem, e podem ser realizadas com base na transformada de Fourier.

De forma a facilitar a análise da convolução, começaremos por considerar a convolução de duas funções contínuas unidimensionais, passando depois para as funções discretas. Finalmente analisamos as situações bidimensionais, quer no caso contínuo quer no discreto.

A convolução de duas funções $f(x)$ e $g(x)$, representada por $f(x) * g(x)$, é a resposta de um SLI (Sistema Linear e Invariante) com reposta impulsional $f(x)$ à entrada $g(x)$ e é definida pelo integral

$$f(x) * g(x) = \int_{-\infty}^{+\infty} f(k)g(x - k)dk, \tag{22}$$

em que k é uma variável de integração.

A convolução $f(x) * g(x)$ tem como principal característica o facto de no domínio das frequências ser o produto das transformadas de Fourier, $F(u)G(u)$, ou seja

$$f(x) * g(x) \Leftrightarrow F(u)G(u) \tag{23}$$

e portanto pode ser obtida pela transformada inversa de Fourier do produto $F(u)G(u)$. Por outro lado a convolução no domínio das frequências, $F(u) * G(u)$, pode ser obtida pelo produto $f(x)g(x)$ no domínio dos xx , ficando

$$f(x)g(x) \Leftrightarrow F(u) * G(u). \tag{24}$$

As equações (23) e (24) são designadas normalmente pelo Teorema da Convolução.

A equação (23) pode ser demonstrada fazendo

$$\mathfrak{F}[f(x) * g(x)] = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(k)g(x - k)dk e^{-j2\pi ux} dx = \int_{-\infty}^{+\infty} f(k) \int_{-\infty}^{+\infty} g(x - k)e^{-j2\pi ux} dx dk,$$

como a propriedade da translação unidimensional da função contínua é

$$\int_{-\infty}^{+\infty} g(x-k)e^{-j2\pi ux} dx = G(u)e^{-j2\pi uk},$$

fica

$$\mathfrak{F}[f(x) * g(x)] = \int_{-\infty}^{+\infty} f(k)e^{-j2\pi uk} dk G(u) = F(u)G(u).$$

A visualização desta propriedade torna-se mais fácil se analisarmos o exemplo da Figura 3.7, onde temos em (a) e (b) respectivamente a função $f(x)$ e $g(x)$, representadas na variável de integração k . Antes da integração é necessário colocar a função $g(x)$ na forma pretendida, $g(x-k)$, como é representado na Figura 3.7 (c) e (d). Então para cada valor de x é multiplicado $f(k)$ pelo correspondente $g(x-k)$, zona sombreada na Figura 3.7 (e), e é feita a integração do produto de $-\infty$ a $+\infty$. A Figura 3.7 (e) é válida para $0 \leq x \leq 1$. Como o produto é 0 para k fora do intervalo $[0, x]$, então $f(x) * g(x) = \frac{x}{2}$, ficamos simplesmente com a área da região sombreada da Figura 3.7 (e). Para $1 \leq x \leq 2$ temos o que está representado na Figura 3.7 (f) obtendo $f(x) * g(x) = 1 - \frac{x}{2}$, porque o produto $f(k)g(x-k)$ é 0 para valores de x fora do intervalo $[0, 2]$. Finalmente

$$f(x) * g(x) = \begin{cases} \frac{x}{2} & , \text{ se } 0 \leq x \leq 1 \\ 1 - \frac{x}{2} & , \text{ se } 1 \leq x \leq 2 \\ 0 & , \text{ se não} \end{cases}$$

Na Figura 3.7 (g) está representado o resultado final.

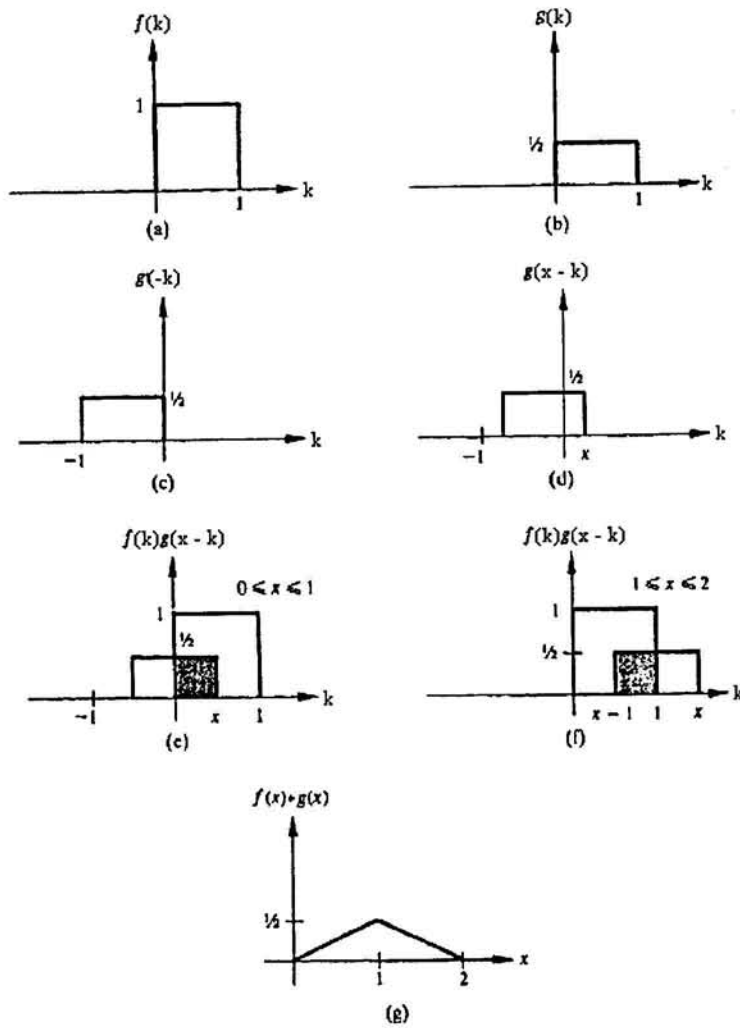


Figura 3.7 Ilustração gráfica da convolução de funções unidimensionais contínuas [104].

Da uma forma semelhante podia ser analisada a convolução no domínio da frequência.

Podemos referir algumas propriedades da convolução como:

Propriedade comutativa

$$f(x) * g(x) = g(x) * f(x), \tag{25}$$

fazendo-se

$$f(x) * g(x) = \int_{-\infty}^{+\infty} f(k)g(x-k)dk$$

e substituindo k por $(x-j)$, obtemos

$$\begin{aligned} f(x) * g(x) &= \int_{-\infty}^{+\infty} f(x-j)g[x-(x-j)]d(x-j) \\ &= \int_{-\infty}^{+\infty} g(j)f(x-j)dj \\ &= g(x) * f(x) \end{aligned}$$

como pretendíamos verificar;

Propriedade Distributiva

$$f(x) * [g(x) + h(x)] = f(x) * g(x) + f(x) * h(x). \quad (26)$$

A equação (26) pode ser demonstrada se fizermos

$$\begin{aligned} f(x) * [g(x) + h(x)] &= \int_{-\infty}^{+\infty} f(k)[g(x-k) + h(x-k)]dk \\ &= \int_{-\infty}^{+\infty} f(k)g(x-k)dk + \int_{-\infty}^{+\infty} f(k)h(x-k)dk \\ &= f(x) * g(x) + f(x) * h(x), \end{aligned}$$

verificando-se assim a propriedade distributiva;

Propriedade associativa

$$f(x) * [g(x) * h(x)] = [f(x) * g(x)] * h(x) \quad (27)$$

A melhor forma de ser verificada a propriedade associativa é recorrendo às transformadas de Fourier de cada uma das funções, ou seja

$$f(x) \Leftrightarrow F(u),$$

$$g(x) \Leftrightarrow G(u),$$

$$h(x) \Leftrightarrow H(u)$$

e

$$\begin{aligned}\mathfrak{F}\{f(x) * [g(x) * h(x)]\} &= F(u)[G(u)H(u)] \\ &= F(u)G(u)H(u),\end{aligned}$$

logo podemos concluir que

$$\mathfrak{F}^{-1}\{[F(u)G(u)]H(u)\} = [f(x) * g(x)] * h(x).$$

Referimo-nos até aqui foi a funções unidimensionais contínuas e em seguida vamos analisar a convolução para funções unidimensionais discretas.

Sejam as funções $f(x)$ e $g(x)$ amostradas respectivamente,

$$f(x) = \{f(0), f(1), f(2), \dots, f(A-1)\} \text{ e}$$

$$g(x) = \{g(0), g(1), g(2), \dots, g(B-1)\}.$$

Como a transformada de Fourier e a sua inversa são funções periódicas, (15) e (16), para que o teorema da convolução discreta seja consistente em termos de periodicidade temos que ter $f(x) = f(x+M)$ e $g(x) = g(x+M)$. Nesta situação o resultado da convolução discreta de $f(x)$ com $g(x)$ também é periódico de período M .

O valor de M tem que ser igual ou superior a $A+B-1$ para que não haja erro de sobreposição, não ocorra *aliasing* [104]. Como $f(x)$, $g(x)$ e $f(x) * g(x)$ devem ter a mesma dimensão, é necessário acrescentar zeros a $f(x)$ e a $g(x)$ por forma a que fiquem com a mesma dimensão de M , ou seja

$$f_e(x) = \begin{cases} f(x) & 0 \leq x \leq A-1 \\ 0 & A \leq x \leq M-1 \end{cases}$$

e

$$g_e(x) = \begin{cases} g(x) & 0 \leq x \leq B-1 \\ 0 & B \leq x \leq M-1. \end{cases}$$

A convolução discreta de $f_e(x)$ com $g_e(x)$ é representada pela expressão (28)

$$f_e(x) * g_e(x) = \frac{1}{M} \sum_{m=0}^{M-1} f_e(m) g_e(x-m), \quad x=0 \dots M-1. \quad (28)$$

A função convolução é assim uma sequência discreta, periódica de comprimento M .

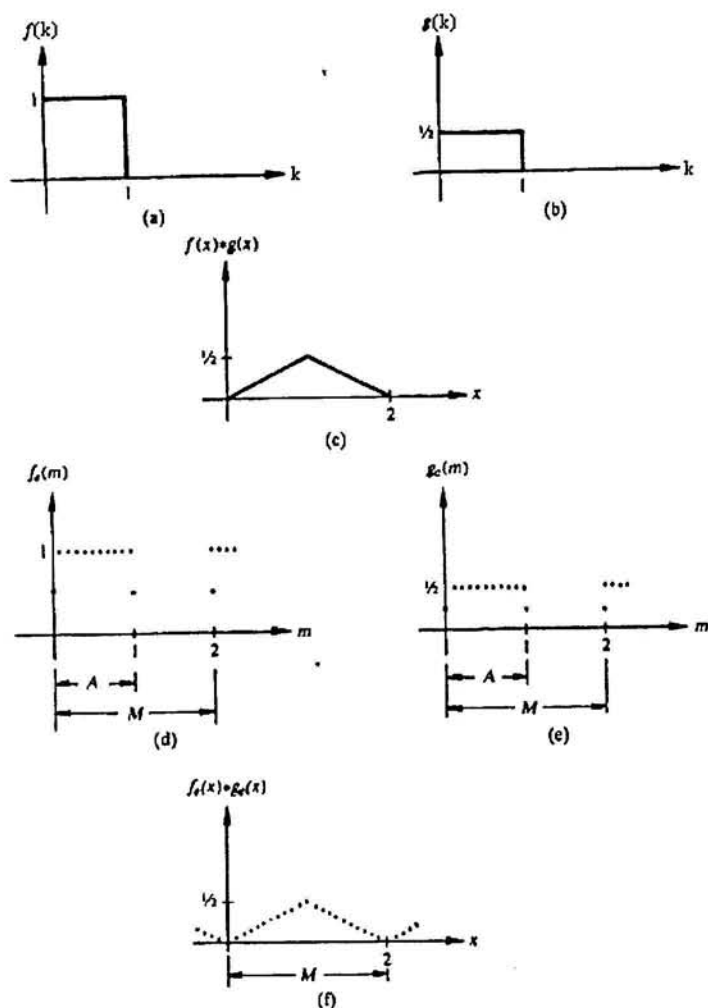


Figura 3.8 Comparação entre a convolução de funções unidimensionais contínuas e as discretas [104].

A visualização da convolução discreta é basicamente a mesma que a apresentada para a convolução contínua. A única diferença que temos é que na convolução discreta estamos a trabalhar com amostras, logo em vez da soma infinitesimal (integral) temos o somatório.

Desta forma as equações (23) e (24) passam a ser

$$f_e(x) * g_e(x) \Leftrightarrow F_e(u)G_e(u), \quad (29)$$

$$f_e(x)g_e(x) \Leftrightarrow F_e(u) * G_e(u). \quad (30)$$

Na Figura 3.8 podemos ver a comparação entre a convolução contínua, exemplo apresentado na Figura 3.7, e a convolução discreta.

A equação da convolução bidimensional de duas funções contínuas é análoga à equação (22) da convolução unidimensional. Sendo assim para duas funções $f(x, y)$ e $g(x, y)$ temos

$$f(x, y) * g(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(k, l)g(x-k, y-l)dkdl, \quad (31)$$

em que k e l são duas variáveis de integração.

O teorema da convolução de funções bidimensionais é então expresso pelas relações

$$f(x, y) * g(x, y) \Leftrightarrow F(u, v)G(u, v) \quad (32)$$

e

$$f(x, y)g(x, y) \Leftrightarrow F(u, v) * G(u, v). \quad (33)$$

A ilustração gráfica da equação (31) é mais difícil de representar do que a da equação (22). Pode-se ver essa representação na Figura 3.9, onde visualizamos as operações necessárias para fazer a convolução bidimensional das funções $f(x, y)$ e $g(x, y)$, à semelhança do exemplo apresentado na Figura 3.7.

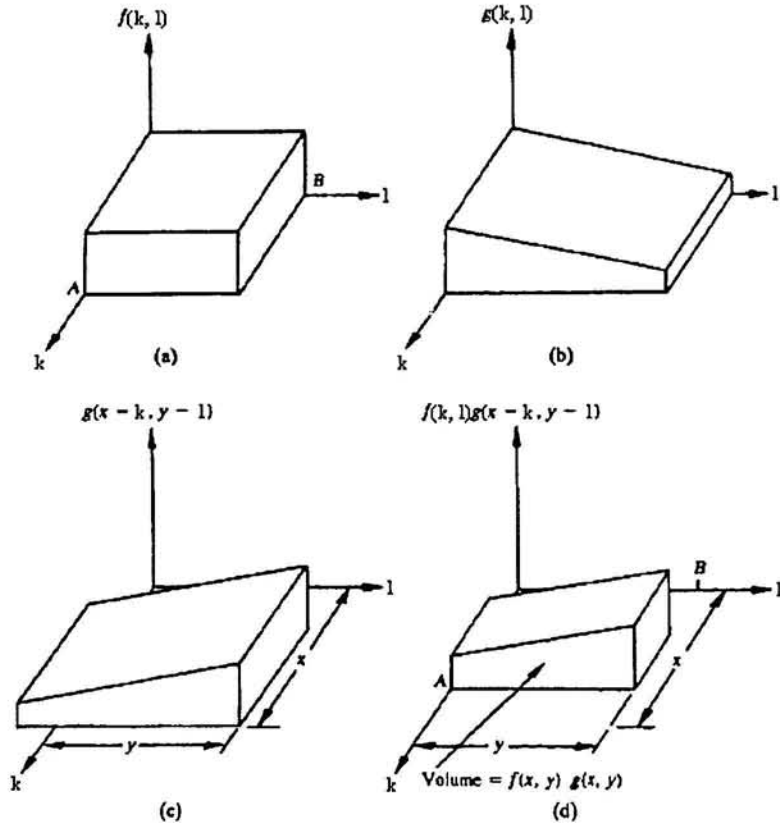


Figura 3.9 Ilustração das operações necessárias para fazer a convolução bidimensional das funções $f(x, y)$ e $g(x, y)$ contínuas [104].

A verificação matemática da convolução bidimensional, para funções ou sinais contínuos, é em tudo semelhante à já efectuada para uma dimensão.

Ao amostrarmos as funções $f(x, y)$ e $g(x, y)$ com uma dimensão $A \times B$ e $C \times D$ respectivamente, pode-se determinar a convolução discreta bidimensional. Como foi visto para a situação de uma dimensão, também aqui estas funções amostradas têm que ser periódicas com período M e N , na direcção x e y respectivamente. Por forma a evitar o erro de sobreposição, isto é, não ocorrer *aliasing* [106], M e N têm que ser tais que

$$M \geq A + C - 1$$

e

$$N \geq B + D - 1.$$

As funções $f(x, y)$ e $g(x, y)$, passam a ser

$$f_e(x, y) = \begin{cases} f(x, y) & 0 \leq x \leq A-1 \text{ e } 0 \leq y \leq B-1 \\ 0 & A \leq x \leq M-1 \text{ ou } B \leq y \leq N-1 \end{cases}$$

e

$$g_e(x, y) = \begin{cases} g(x, y) & 0 \leq x \leq C-1 \text{ e } 0 \leq y \leq D-1 \\ 0 & C \leq x \leq M-1 \text{ ou } D \leq y \leq N-1. \end{cases}$$

A convolução bidimensional de $f_e(x, y)$ e $g_e(x, y)$ fica definida por

$$f_e(x, y) * g_e(x, y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f_e(m, n) g_e(x-m, y-n), \quad x=0 \dots M-1 \text{ e } y=0 \dots N-1. \quad (34)$$

Sendo assim o teorema da convolução discreta bidimensional é representado pelas equações

$$f_e(x, y) * g_e(x, y) \Leftrightarrow F_e(u, v) G_e(u, v) \quad (35)$$

e

$$f_e(x, y) g_e(x, y) \Leftrightarrow F_e(u, v) * G_e(u, v). \quad (36)$$

Em suma, concluímos que a convolução de duas funções ou sinais, independentemente de serem unidimensionais ou multidimensionais, ou de serem discretos ou contínuos, no domínio do tempo ou espaço, é equivalente à multiplicação dos seus espectros no domínio das frequências. Concluímos também que o inverso é válido ou seja, a multiplicação de duas funções ou sinais no domínio do tempo ou espaço, é equivalente à convolução dos seus espectros no domínio das frequências.

Correlação

A correlação, no processamento de imagem, tem como principal aplicação encontrar a semelhança entre uma imagem conhecida e um conjunto de imagens desconhecidas, ou localizar uma parte da imagem pretendida numa outra, ver Figura 3.10.

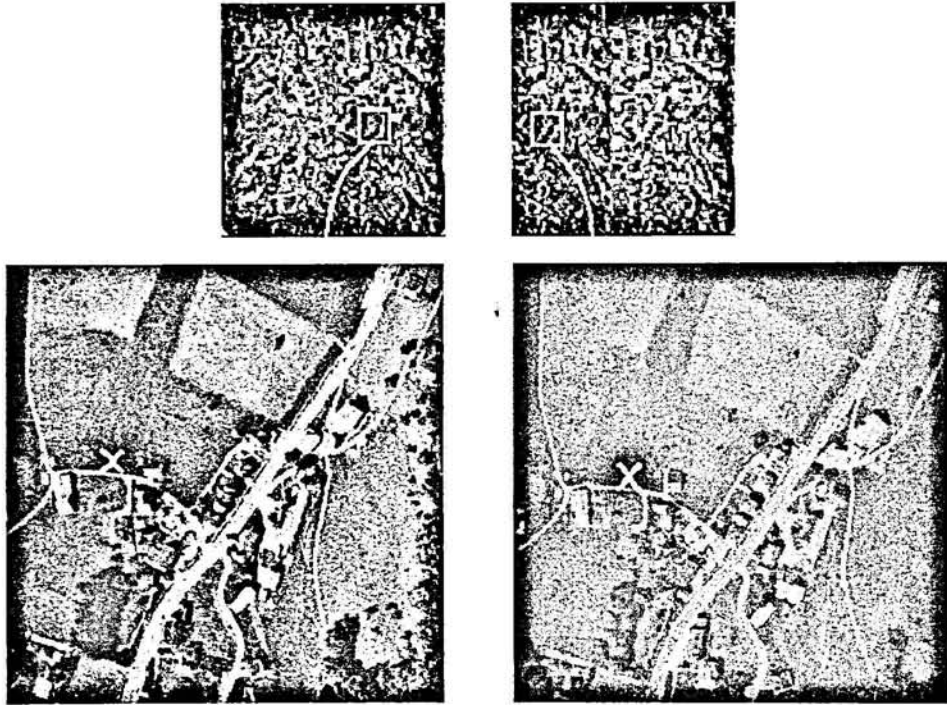


Figura 3.10 Localização de uma parte da imagem numa outra [107].

Existem normalmente duas designações usadas para a correlação. Uma é a autocorrelação, quando as duas funções a correlacionar são iguais, e no caso de serem as duas diferentes, é a correlação cruzada. Esta é a mais utilizada.

A correlação cruzada entre duas funções ou sinais, contínuos ou discretos, unidimensionais ou multidimensionais, dá-nos o grau de semelhança entre essas funções ou sinais.

Nas funções contínuas e unidimensionais $f(x)$ e $g(x)$ temos a correlação definida por

$$f(x) \circ g(x) = \int_{-\infty}^{+\infty} f^*(k)g(x+k)dk, \quad (37)$$

em que $f^*(k)$ é o complexo conjugado de $f(k)$. As equações (37) e (22) são similares.

Para fazer a correlação, basta deslocar $g(x)$ por $f(x)$ e integrar o produto de $-\infty$ a $+\infty$ para cada valor de x . O processo pode ser visualizado na Figura 3.11, à semelhança do

exemplo analisado na Figura 3.7.

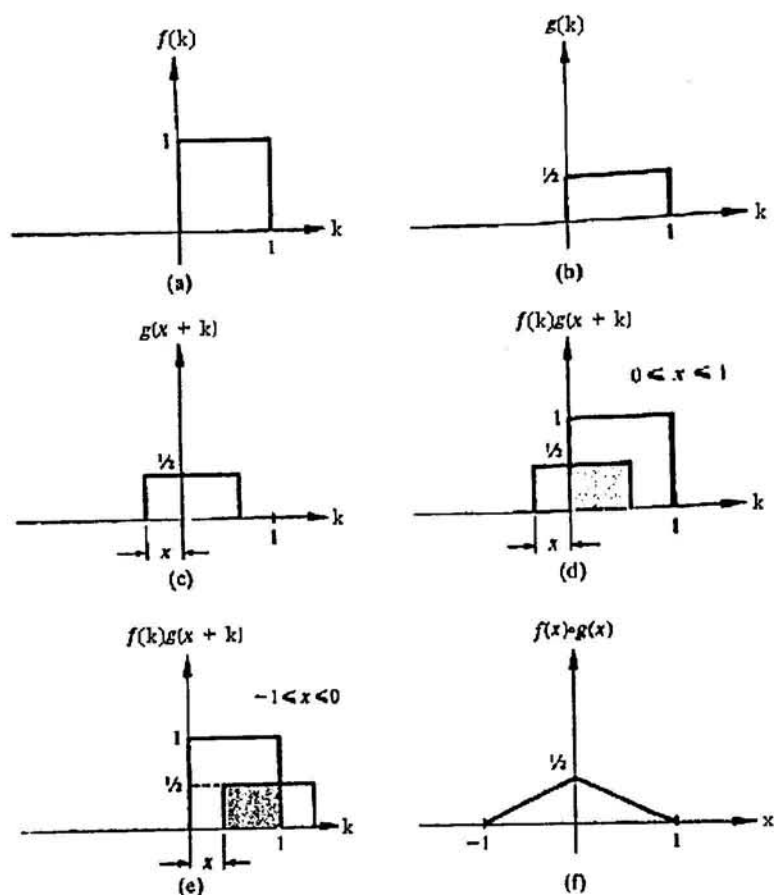


Figura 3.11 Ilustração gráfica da correlação [104].

O teorema da correlação neste caso é

$$f(x) \circ g(x) \Leftrightarrow F^*(u)G(u), \quad (38)$$

e

$$f^*(x)g(x) \Leftrightarrow F(u) \circ G(u). \quad (39)$$

A demonstração da equação (38) é em tudo semelhante à demonstração efectuada para a equação (23), tendo em conta que $f^*(x)$ é o complexo conjugado de $f(x)$ e $F^*(u)$ é o complexo conjugado da transformada de Fourier de $F(u)$.

Na situação de funções discretas unidimensionais $f_e(x)$ e $g_e(x)$, a correlação é equi-

valente à equação (37),

$$f_e(x) \circ g_e(x) = \frac{1}{M} \sum_{m=0}^{M-1} f_e^*(m) g_e(x+m), \quad x=0 \dots M-1. \quad (40)$$

As considerações que foram feitas na convolução em relação à periodicidade e à escolha do valor de M , também se aplicam à equação (40).

O teorema da correlação nesta situação é então

$$f_e(x) \circ g_e(x) \Leftrightarrow F_e^*(u) G_e(u), \quad (41)$$

$$f_e^*(x) g_e(x) \Leftrightarrow F_e(u) \circ G_e(u). \quad (42)$$

Nas equações (40) e (42), $f_e^*(x)$ é o complexo conjugado de $f_e(x)$ e na equação (41) temos que $F_e^*(u)$ é o complexo conjugado da transformada de Fourier de $f_e(x)$.

A correlação cruzada para funções ou sinais contínuos bidimensionais pode ser analisada da mesma forma que foi analisada a correlação cruzada contínua unidimensional. Sendo assim, se $f(x, y)$ e $g(x, y)$ forem funções de variáveis contínuas, então a correlação é definida por

$$f(x, y) \circ g(x, y) = \int_{-\infty-\infty}^{+\infty+\infty} \int f^*(k, l) g(x+k, y+l) dk dl. \quad (43)$$

Nesta situação o teorema da correlação cruzada de funções ou sinais contínuos bidimensionais é definido pelas equações

$$f(x, y) \circ g(x, y) \Leftrightarrow F^*(u, v) G(u, v) \text{ e} \quad (44)$$

$$f^*(x, y) g(x, y) \Leftrightarrow F(u, v) \circ G(u, v), \quad (45)$$

em que $f^*(x, y)$ é o complexo conjugado da função ou sinal $f(x, y)$ e $F^*(u, v)$ é o complexo conjugado da transformada de Fourier de $f(x, y)$.

Funções ou sinais discretos bidimensionais têm como equação de correlação cruzada

$$f_e(x, y) * g_e(x, y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f_e(m, n) g_e(x-m, y-n), \quad x=0 \dots M-1 \text{ e } y=0 \dots N-1 \quad (46)$$

O teorema da correlação fica assim

$$f_e(x, y) \circ g_e(x, y) \Leftrightarrow F_e^*(u, v) G_e(u, v) \quad (47)$$

e

$$f_e^*(x, y) g_e(x, y) \Leftrightarrow F_e(u, v) \circ G_e(u, v) \quad (48)$$

em que $f_e^*(x, y)$ é o complexo conjugado de $f_e(x, y)$ e $F_e^*(u, v)$ é o complexo conjugado da transformada de Fourier de $f_e(x, y)$.

Com estas funções ou sinais discretos, temos que ter em atenção as considerações que foram feitas para a convolução de funções ou sinais discretos bidimensionais, em relação à periodicidade e à escolha do valor de M e de N .

3.3 Transformada de Fourier – Mellin

Como já foi referido, o algoritmo seleccionado para a determinação da translação e rotação entre duas imagens consecutivas foi o método de *matching* baseado na transformada de Fourier – Mellin, seguido da correlação cruzada da fase.

Este método tem como principal vantagem a capacidade de fornecer, por comparação, a translação, a rotação e a escala, com grande precisão, confiabilidade e eficiência. A transformada de Fourier – Mellin tem sido aplicada principalmente no processamento de imagens médicas, no reconhecimento de impressões digitais e na identificação de objectos. Ultimamente o conceito da transformada de Fourier – Mellin tem sido utilizado para colocar a marca de água (Watermark) [108] na protecção dos direitos de autor de imagens digitais, por forma a evitar a cópia indevida.

A confiabilidade é muito importante, porque podemos ter alterações de iluminação entre a imagem de referência e a nova imagem adquirida. Essas alterações de iluminação podem ser de intensidade de luz, da sua direcção, de sombras, assim como podem aparecer novos objectos na imagem, e o algoritmo tem que garantir que estas alterações não afec-

tam o resultado. O algoritmo sendo confiável e robusto permite-nos também adquirir as imagens a partir de uma câmara de baixa qualidade, uma câmara USB por exemplo, permitindo assim um baixo custo no sistema de aquisição das imagens. Para além do que já foi referido o algoritmo tem que ser numericamente eficiente por forma a permitir o funcionamento em tempo real.

O algoritmo usado é um algoritmo que tem como componente principal a transformada de Fourier – Mellin, mas não é de descurar a importância da correlação cruzada da fase, pelo que vamos começar por fazer uma análise resumida das características principais da fase.

Considerando uma sequência complexa $f(x)$, podemos escrevê-la segundo as suas duas componentes

$$f(x) = f_R(x) + jf_I(x) \quad (49)$$

em que $f_R(x)$ é a componente real e $f_I(x)$ é a componente imaginária. Aplicando a identidade de Euler à definição da transformada de Fourier

$$F(e^{j\omega}) = \sum_{x=-\infty}^{+\infty} f(x)e^{-j\omega x}, \quad (50)$$

obtemos

$$F(e^{j\omega}) = \sum_{x=-\infty}^{+\infty} [f_R(x) + jf_I(x)][\cos(\omega x) - j\sin(\omega x)]. \quad (51)$$

A função (51) pode ser expressa na sua parte real e imaginária,

$$F(e^{j\omega}) = F_R(e^{j\omega}) + jF_I(e^{j\omega}) \quad (52)$$

em que,

$$F_R(e^{j\omega}) = \sum_{x=-\infty}^{+\infty} [f_R(x)\cos(\omega x) + f_I(x)\sin(\omega x)] \quad (53)$$

e

$$F_I(e^{j\omega}) = \sum_{x=-\infty}^{+\infty} [f_I(x) \cos(\omega x) - f_R(x) \sin(\omega x)]. \quad (54)$$

A equação (52) também pode ser expressa sob a forma de vector complexo,

$$F(e^{j\omega}) = |F(e^{j\omega})| e^{j \text{Arg}[F(e^{j\omega})]}. \quad (55)$$

Quando $F(e^{j\omega})$ é a função de transferência de um sistema, $|F(e^{j\omega})|$ é designado pela resposta em amplitude e $\text{Arg}[F(e^{j\omega})]$ é designado pela resposta em fase. Da mesma forma, para uma sequência de sinais, $F(e^{j\omega})$ é o espectro, $|F(e^{j\omega})|$ é o espectro da amplitude e $\text{Arg}[F(e^{j\omega})]$ é o espectro da fase.

O espectro da amplitude é

$$|F(e^{j\omega})| = (F(e^{j\omega}) F^*(e^{j\omega}))^{1/2} \quad (56)$$

em que $F^*(e^{j\omega})$ é o complexo conjugado de $F(e^{j\omega})$. Uma expressão equivalente a (56) é

$$|F(e^{j\omega})| = (F_R^2(e^{j\omega}) + F_I^2(e^{j\omega}))^{1/2}. \quad (57)$$

Da mesma forma o espectro da fase é obtido por

$$\phi(e^{j\omega}) = \text{Arg}[F(e^{j\omega})] = \arctan \left[\frac{F_I(e^{j\omega})}{F_R(e^{j\omega})} \right]. \quad (58)$$

Caso particulares do espectro da fase são a fase zero e a fase linear. Na Figura 3.12 estão representados exemplos de sinais que contém fases lineares e não lineares, respectivamente a), b) e c).

O sinal sendo simétrico, lado direito igual ao lado esquerdo, em torno da amostra número zero tem fase zero, ver Figura 3.12 a). Se o sinal simétrico estiver deslocado, por exemplo de 5, conforme mostra na Figura 3.12 b), tem fase linear. Qualquer sinal de fase linear pode ser convertido num sinal de fase zero se for centrado na amostra número zero. Na Figura 3.12 c) podemos ver um sinal com fase não linear, em que é um sinal não simétrico.

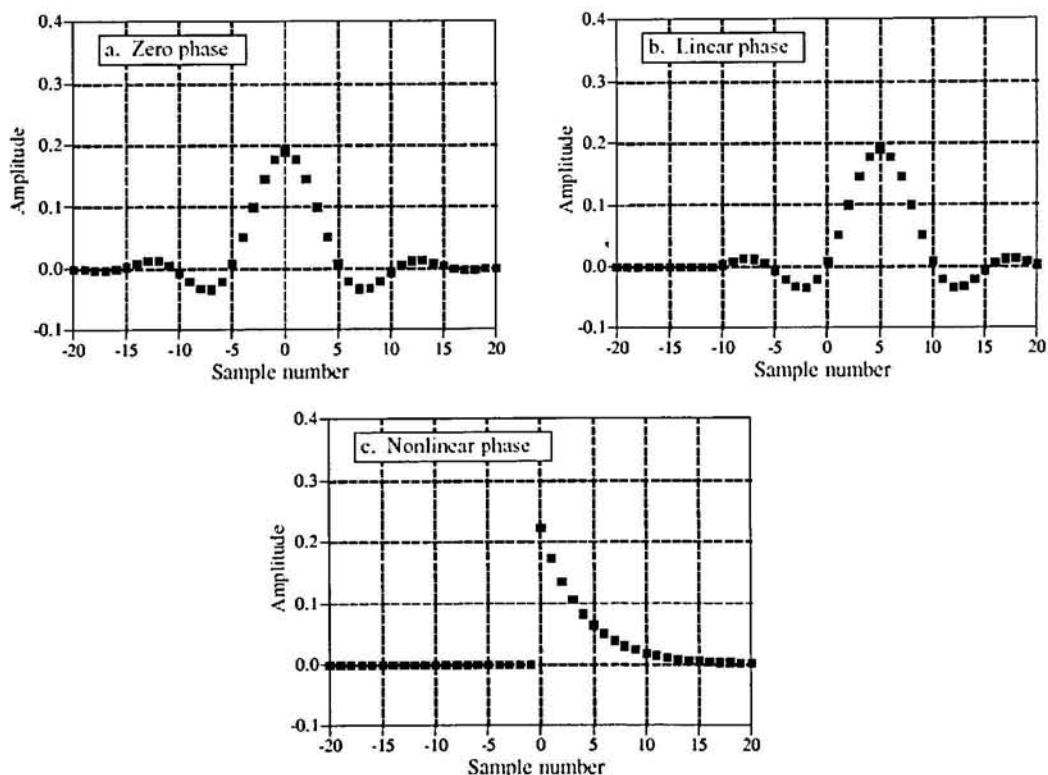


Figura 3.12 Exemplos de sinais com linearidade da fase diferentes; a) fase zero, b) fase linear e c) fase não linear [109].

A linearidade ou não da fase pode ser melhor entendida se analisarmos a Figura 3.13. Nesta figura temos em a) o espectro da fase de um sinal com fase linear centrado em várias posições e em b) o espectro de fase de um sinal de fase não linear também em várias posições. O deslocamento do sinal não altera as características de linearidade ou não linearidade da fase.

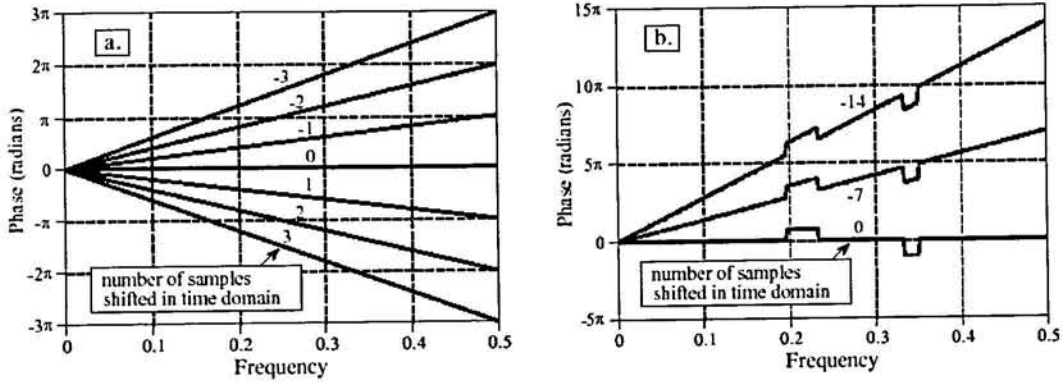


Figura 3.13 Exemplo da linearidade da fase para diferentes deslocamentos; a) fase linear, b) fase não linear [109].

A propriedade da translação da transformada de Fourier de sinais discretos (DFT) diz-nos que se um sinal no domínio dos tempos for deslocado, ocorre uma alteração na fase e não se altera a amplitude no domínio das frequências, ver Figura 3.14. Nesta figura temos que $d(n)$ é o sinal original, $|D_0(e^{j\omega})|$ é o espectro da amplitude, $Arg[D_0(e^{j\omega})]$ é o espectro da fase do sinal original (a) e depois de aplicada uma translação k (b) (c).

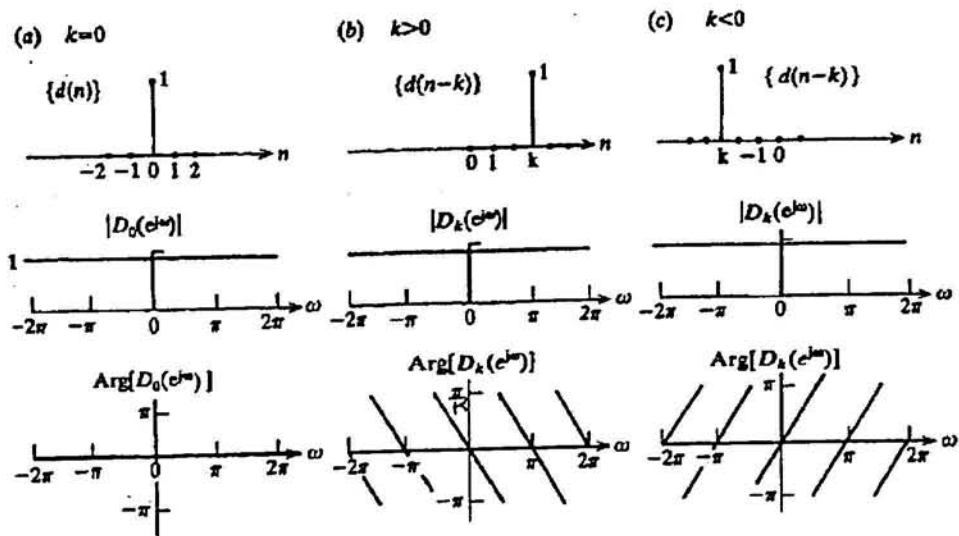


Figura 3.14 a) Espectro da amplitude e da fase do sinal $d(n)$ para $(k=0)$; b) deslocamento no sentido positivo $(k>0)$; c) deslocamento no sentido negativo $(k<0)$ [103].

Na Figura 3.15 temos outro exemplo da alteração do espectro da fase em relação ao deslocamento de um sinal discreto no domínio dos tempos. Como para a DFT o sinal tem que ser periódico e se este é simétrico em relação a θ , também o é obrigatoriamente em relação a $N/2$, o que na Figura 3.15 acontece para a amostra 256 ($N/2$). Podemos verificar que quando o deslocamento é feito para a esquerda (a) o declive é positivo (d) e quando é para a direita (c) o declive é negativo (f).

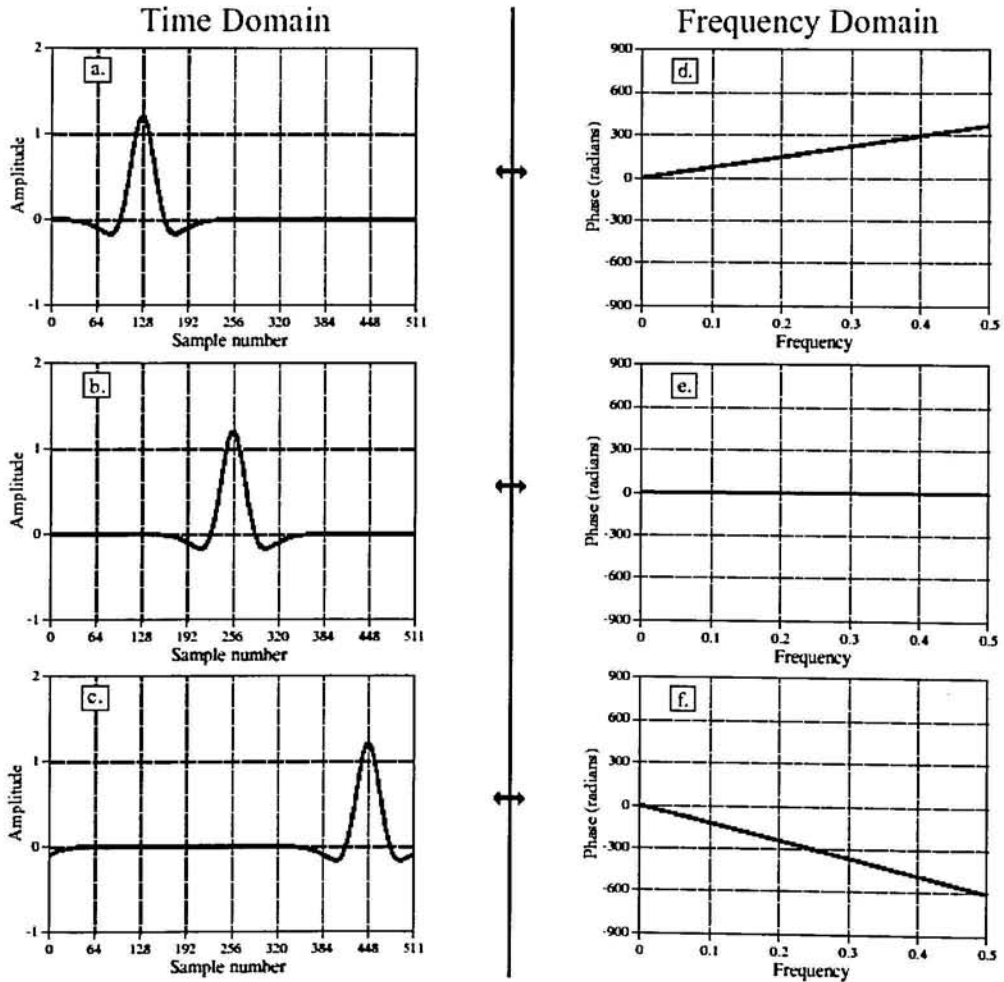


Figura 3.15 Visualização da fase no domínio das frequências provocado pelo deslocamento do sinal no domínio dos tempos [109].

O declive da fase é tanto maior quanto maior for o deslocamento do sinal, como pode ser visto pela Figura 3.13 a).

Nas duas figuras que se seguem podemos ver a informação que se pode retirar a partir da fase.

Na Figura 3.16 a) temos um sinal com a forma de um degrau em rampa. Se calcularmos a DFT deste sinal e substituirmos a fase por valores aleatórios entre $-\pi$ e π , ou seja se desprezarmos a fase, e calcularmos a transformada inversa obtemos o sinal representado em b), em que deixamos de reconhecer o degrau. Mas se depois de calculada a DFT em vez de desprezarmos a fase, desprezarmos a amplitude do espectro e procedermos do mesmo modo, obtemos o sinal representado em c), que exhibe dois impulsos bem definidos do começo e do fim do degrau.

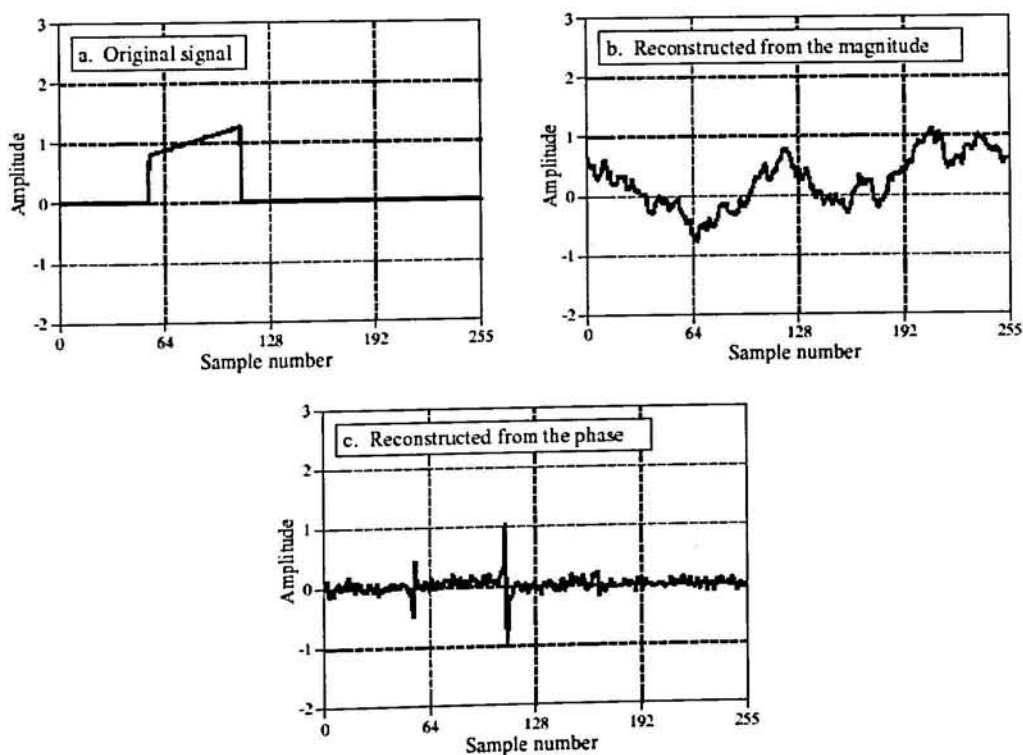


Figura 3.16 Informação presente na fase [109].

À semelhança do que foi visualizado para a figura anterior, pode-se ver o mesmo o mesmo efeito na Figura 3.17, em que se aplica a DFT a uma imagem bidimensional a).

A Figura 3.17 b) representa o espectro da amplitude e a c) o espectro da fase. Se

fizemos a fase igual a zero e calcularmos a transformada inversa obtemos d), em que não reconhecemos a imagem original. Por outro lado, se atribuirmos um valor constante à amplitude e procedermos do mesmo modo então obtemos e), verificando-se que podemos a partir desta imagem reconhecer o retrato da imagem original.

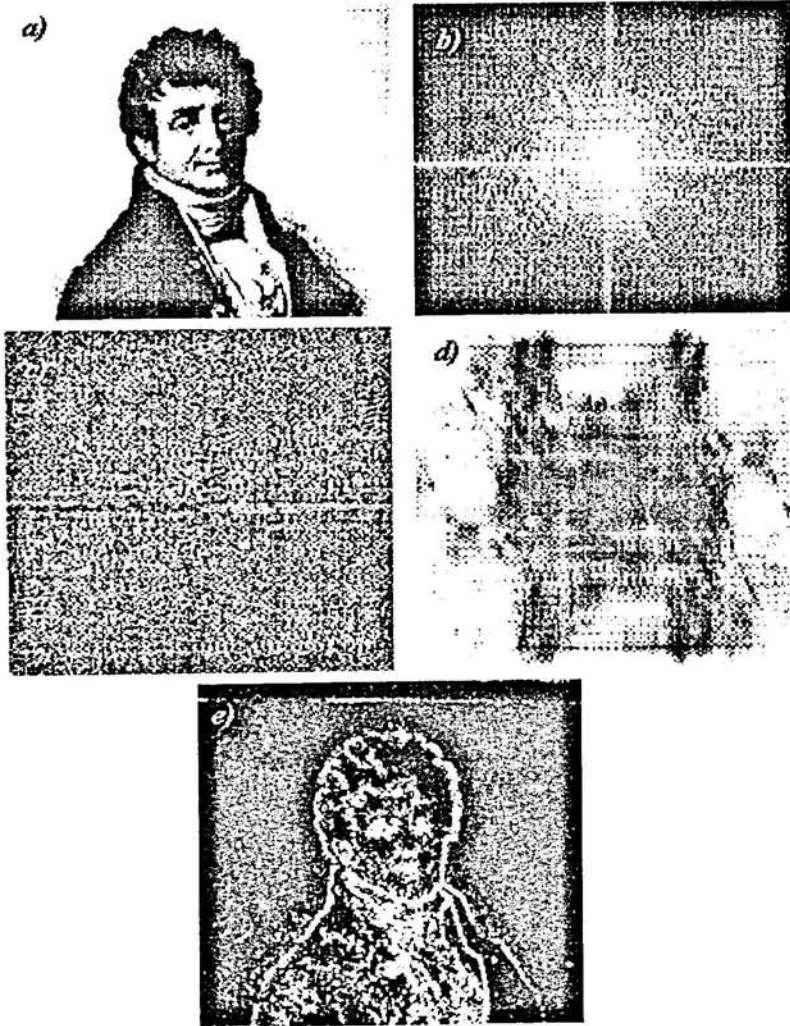


Figura 3.17 Jean Baptiste Joseph Fourier e a sua Transformada de Fourier; a) imagem original; b) espectro da amplitude; c) espectro da fase; d) reconstrução só a partir da amplitude; e) Reconstrução só a partir da fase [105].

Resumidamente, podemos dizer que muita da informação sobre o sinal original está presente na fase, e não apenas na amplitude do seu espectro. Em contraste com esta

situação, temos outros sinais, como por exemplo os sinais áudio, em que a amplitude do espectro é a que contém a informação mais útil.

A amplitude do espectro especifica quanto é a contribuição de cada componente sinusoidal para a imagem, enquanto que a fase especifica a posição relativa de cada componente.

No trabalho é usada a correlação cruzada da fase para retirar a translação, rotação e escala. Como já foi referido, para além da correlação, baseamo-nos no método designado por transformada de Fourier – Mellin. A essência deste método reside no facto de transformar a rotação e a escala numa translação, permitindo assim obtê-las facilmente, por correlação cruzada da fase.

O primeiro passo é calcular a transformada de Fourier das imagens. Seguidamente por reamostragem converte-se o espectro da amplitude, de coordenadas cartesianas para coordenadas polares – logarítmicas. Aplica-se então o método de correlação da fase para se obter a rotação e a escala. Retiram-se estas da imagem original e determina-se a translação, mais uma vez pelo método da correlação cruzada da fase ou seja, supondo que temos uma imagem $f_1(x, y)$ e uma outra $f_2(x, y)$ que estão relacionadas por

$$f_2(x, y) = f_1[a(x \cos \phi_0 + y \sin \phi_0) - x_0, a(-x \sin \phi_0 + y \cos \phi_0) - y_0],$$

$$-\infty < x, y < \infty, \quad (59)$$

em que a é o factor de escala, ϕ_0 o ângulo de rotação e (x_0, y_0) a translação aplicada à imagem $f_1(x, y)$.

As transformadas de Fourier, $F_1(u, v)$ e $F_2(u, v)$ de $f_1(x, y)$ e $f_2(x, y)$ estão relacionadas por,

$$F_2(u, v) = e^{-j\varphi(u, v)} \frac{1}{a^2} \left| F_1\left(\frac{1}{a}(u \cos \phi_0 + v \sin \phi_0), \frac{1}{a}(-u \sin \phi_0 + v \cos \phi_0)\right) \right|,$$

$$-\infty < u, v < \infty, \quad (60)$$

onde $\varphi(u, v)$ é a fase de $F_2(u, v)$. Esta fase depende da translação, da escala e da rotação, mas a amplitude de $F_2(u, v)$,

$$|F_2(u, v)| = \frac{1}{a^2} \left| F_1 \left(\frac{1}{a}(u \cos \phi_0 + v \text{sen} \phi_0), \frac{1}{a}(-u \text{sen} \phi_0 + v \cos \phi_0) \right) \right|,$$

$$-\infty < u, v < \infty, \quad (61)$$

é invariante para a translação.

Da equação (61) podemos ver que a rotação da imagem $f_1(x, y)$ provoca também uma rotação do mesmo ângulo ϕ_0 , na amplitude de $F_1(u, v)$, e a escala é afectada de $1/a^2$.

Tomando o espectro da amplitude $|F_1(u, v)|$ e $|F_2(u, v)|$, reamostrados de coordenadas cartesianas (x, y) para coordenadas polares (ϕ, r) , obtemos duas 'imagens' $g_1(\phi, r)$ e $g_2(\phi, r)$

$$g_1(\phi, r) = |F_1(r \cos \phi, r \text{sen} \phi)|, \quad 0 \leq \phi < 2\pi, 0 \leq r < \infty \quad (62)$$

e

$$g_2(\phi, r) = |F_2(r \cos \phi, r \text{sen} \phi)|, \quad 0 \leq \phi < 2\pi, 0 \leq r < \infty. \quad (63)$$

De notar que o espectro da amplitude é uma função periódica com um ângulo polar ϕ e como a 'imagem' original é real,

$$g_1(\phi \pm n\pi, r) = g_1(\phi, r), \quad \text{para } n = \dots, -2, -1, 0, 1, 2, \dots, \quad (64)$$

segundo a propriedade da periodicidade e simetria do conjugado.

Podemos assim considerar apenas $0 \leq \phi < \pi$. Se na equação (61) substituirmos as coordenadas cartesianas pelas polares passamos a ter

$$\begin{cases} \frac{1}{a}(u \cos \phi_0 + v \text{sen} \phi_0) \\ \frac{1}{a}(-u \text{sen} \phi_0 + v \cos \phi_0) \end{cases} = \begin{cases} \frac{1}{a}(r \cos \phi \cos \phi_0 + r \text{sen} \phi \text{sen} \phi_0) \\ \frac{1}{a}(-r \cos \phi \text{sen} \phi_0 + r \text{sen} \phi \cos \phi_0), \end{cases}$$

$$\begin{cases} \frac{r}{a} \cos(\phi - \phi_0) \\ \frac{r}{a} \text{sen}(\phi - \phi_0), \end{cases} \quad 0 \leq \phi < \pi, \quad (65)$$

podendo assim facilmente verificar que

$$g_2(\phi, r) = a^2 g_1\left(\phi - \phi_0, \frac{r}{a}\right). \quad (66)$$

A rotação passou assim a ser vista como se se tratasse de uma translação, ao longo do eixo angular. A escala também pode ser reduzida a um deslocamento linear se passarmos para uma escala logarítmica. Ou seja,

$$g_1(\phi, \lambda) = g_1(\phi, \log r), \quad -\infty < \lambda < \infty \quad (67)$$

e

$$\begin{aligned} g_2(\phi, \lambda) &= g_2(\phi, \log r) = a^2 g_1\left(\phi - \phi_0, \log \frac{r}{a}\right) \\ &= a^2 g_1(\phi - \phi_0, \lambda - \rho), \quad -\infty < \lambda < \infty, \end{aligned} \quad (68)$$

onde $\lambda = \log(r)$ e $\rho = \log(a)$.

Temos assim nesta representação polar – logarítmica a rotação e a escala reduzidas a simples translações.

A partir daqui aplica-se a correlação cruzada da fase às transformadas de Fourier de $g_1(\phi, \lambda)$ e $g_2(\phi, \lambda)$, e determinam-se as duas componentes, rotação e escala.

Sabendo os valores da rotação e da escala podemos retirar estas da imagem original $f_2(x, y)$ e determinar a translação, aplicando mais uma vez a correlação cruzada da fase às transformadas de Fourier da imagem $f_1(x, y)$ e da imagem $f_2(x, y)$, esta já sem rotação nem escala.

Obtemos assim as três componentes pretendidas, translação, rotação e escala.

Conclui-se assim a análise da metodologia seguida nesta dissertação.

No próximo capítulo será analisada a ferramenta usada para a implementação desta metodologia, assim como a sua implementação.

Capítulo 4

Descrição do sistema desenvolvido

O trabalho de projecto desenvolvido no âmbito desta dissertação teve por objectivo a implementação do algoritmo de determinação da posição relativa entre duas imagens, baseado na transformada de Fourier – Mellin e na correlação cruzada da fase, já descritos no capítulo anterior, e o desenvolvimento de um sistema de simulação para a validação dos resultados.

O desenvolvimento foi implementado recorrendo à ferramenta de programação MATLAB.

Neste capítulo será feita uma breve apresentação do Matlab e justificada a utilização do Matlab para desenvolvimento do sistema apresentado.

Ainda neste capítulo será descrito todo o sistema desenvolvido, ficando para o capítulo seguinte a descrição do sistema de simulação.

4.1 Ferramenta de desenvolvimento

De entre as ferramentas de programação existentes optou-se pelo MATLAB (versão 6.5), porque é uma ferramenta com uma linguagem simples, poderosa e fácil de utilizar, onde os problemas e as soluções são expressas em notações matemáticas que nos são familiares.

Esta linguagem é usada tipicamente em computação matemática, no desenvolvimento de algoritmos, na modelação, simulação e execução de protótipos, na exploração, visualização e análise de dados adquiridos externamente, na elaboração de gráficos cientí

ficos e de engenharia, assim como no desenvolvimento de aplicações incluindo construção de interfaces gráficos por forma a facilitar o acesso às aplicações desenvolvidas.

O MATLAB é uma ferramenta muito prática que nos permite resolver facilmente, através de programas pequenos, problemas de computação técnica complicados, como sendo operações com formulas que incluam processamento de vectores e de matrizes.

O nome MATLAB vem de *Matrix laboratory*. Inicialmente foi criado para facilitar o acesso a software com matrizes, tendo sido desenvolvido para os projectos LINPACK[110] e EISPACK[111].

Ao longo dos anos esta ferramenta tem evoluído com a contribuição de vários utilizadores, pela facilidade disponibilizada na criação e interligação de subrotinas, as quais devidamente agrupadas formam as designadas *toolboxes*. As subrotinas criadas podem facilmente ser utilizadas por outro utilizador como uma simples função do MATLAB. Todos estes motivos levam a que este programa seja usado frequentemente no meio universitário, como ferramenta de ensino na aprendizagem ou desenvolvimento aprofundado de cursos de matemática, de engenharia e ciência, assim como na indústria onde é seleccionada frequentemente para pesquisa de alta produtividade, desenvolvimento e análise de soluções tecnológicas mais rentáveis. Permite também trabalhar em duas plataformas de software distintas, como Windows ou Unix.

O MATLAB é constituído por quatro partes principais. O ambiente de trabalho é uma das partes importantes, porque disponibiliza através de interfaces gráficos um conjunto de ferramentas e facilidades que auxiliam no acesso a ficheiros e às funções do MATLAB. Este ambiente pode ser definido pelo utilizador, sendo possível colocar várias janelas visíveis ao mesmo tempo, como a janela de comandos, o histórico de comandos executados, o espaço de trabalho (*workspace*) e o directório actual de trabalho.

Outra parte importante é a biblioteca de funções matemáticas existente que contempla uma vasta gama de funções desde as mais simples como soma, seno, coseno, operações com números complexos às mais sofisticadas, matrizes inversas, matrizes de valores próprios (*eigenvalues*), funções de Bessel e transformadas de Fourier, entre outras.

A linguagem de programação do MATLAB é uma linguagem de alto nível com instruções de controlo de fluxo (*control flow statements*), estruturas de dados, entradas e saídas e é caracterizada por adoptar o paradigma da programação orientada a objectos. Esta linguagem permite fazer programas muito simples e céleres, só para fazer algum teste por exemplo, assim como permite fazer programas complexos para aplicações complexas.

Outra parte importante é a manipulação de gráficos, que inclui comandos de alto nível para visualizar, processar, animar e apresentar gráficos/imagens de duas e três dimensões, assim como comandos de baixo nível para alterar a aparência destes ou construir completamente gráficos/imagens, por forma a facilitar a interacção dos utilizadores com as aplicações.

Por último temos a capacidade de interface das aplicações MATLAB com outros programas (*API – Application Program Interface*), através de uma biblioteca, permite interligar com rotinas escritas externamente ao MATLAB. Permite partilhar dados com rotinas escritas noutras linguagens de programação, equipamentos periféricos que comuniquem directamente com o MATLAB, aplicações de *software* específicas (*ActiveX*) e trocas dinâmicas de dados, entre outras.

Como já foi referido anteriormente, existem conjuntos, devidamente agrupados, de soluções específicas que são designadas por *toolboxes*. As *toolboxes* são muito importantes, porque permitem que os utilizadores aprendam e apliquem tecnologias especializadas, e específicas para determinadas áreas. Estas áreas são por exemplo o processamento de sinal, sistemas de controlo (GPIB, VISA e Portas Série), redes neuronais, lógica *fuzzy*, sistemas de identificação, optimização, comunicações, processamento de imagem, entre muitas outras. As *toolboxes* são constituídas por conjuntos de funções MATLAB (ficheiros com extensão *.m*).

A *toolbox* de processamento de imagem, que foi a mais usada neste trabalho, permite processar imagens em praticamente qualquer formato, por exemplo *jpeg*, *jpg*, *bmp*, *gif*, entre outros. Esta *toolbox* suporta uma vasta gama de operações de processamento de imagem incluindo transformação espacial das imagens, processamento de vizinhança, aplicação de filtros lineares, transformadas, reconhecimento e análise de imagens, *deblur-*

ring, operação nas regiões de interesse, etc..

Existem ainda ferramentas de análise e detecção de falhas (*debug*) que permitem seguir o programa passo a passo por forma a serem detectados e localizados erros. Existem também geradores de relatórios, com os respectivos tempos gastos em cada função, de modo a permitir verificar onde é que o programa gasta a maior parte do tempo, fazendo com que se possa melhorar a sua performance.

Em suma, a ferramenta de programação MATLAB foi escolhida para escrever os programas deste trabalho pela sua simplicidade de manuseamento, pela facilidade de criação de pequenas rotinas reutilizáveis e por já haver bastantes funções desenvolvidas para esta ferramenta, que eram úteis.

4.2 Módulos do Programa

O programa desenvolvido para implementar o sistema proposto consiste num módulo principal designado por *Navega.m* que aceita duas imagens *imageref* e *imagein*, e determina a rotação e translação da segunda em relação à primeira.

Este programa chama dois módulos, um que determina a rotação e a escala entre duas imagens, designado por *Rota_escala.m*, e outro que determina o deslocamento entre duas imagens, designado por *Translacao.m*.

4.2.1 Módulo Principal

O módulo principal tem como parâmetros de entrada duas imagens designadas por *imageref* e *imagein*, em que *imageref* é a imagem de referência, e *imagein* é a imagem a comparar com *imageref*. A imagem *imagein* será semelhante à imagem *imageref* mas deslocada de X , Y e rodada de um ângulo θ . O deslocamento X , Y e o ângulo θ irão ser determinados pelo módulo. As duas imagens têm que ter a mesma dimensão.

Este módulo, designado por *Navega.m*, encontra-se esquematizado no fluxograma da Figura 4.1.

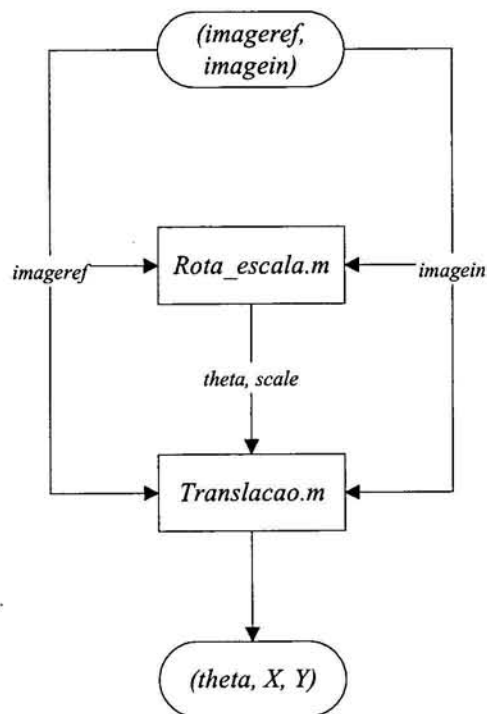


Figura 4.1 Fluxograma do módulo principal *Navega.m*.

O primeiro passo é determinar qual a rotação e a escala entre as duas imagens e para isso é executado a função *Rota_escala.m*.

O módulo *Navega.m* passa como parâmetros de entrada, à função *Rota_escala.m*, as duas imagens já referidas e recebe como resultado o ângulo *theta*, sendo este o ângulo de rotação entre as duas imagens, e o factor de escala *scale*.

Uma vez concluída a função *Rota_escala.m*, executa-se a função *Translacao.m*, cujos parâmetros de entrada são o resultado de *Rota_escala.m*, *theta* e *scale*, juntamente com as duas imagens originais *imageref* e *imagein*. A função *Translacao.m* retorna então o deslocamento *X* e *Y*, assim como o ângulo *theta* correcto. Ficamos assim com os três valores pretendidos, sendo estes respectivamente, o deslocamento no eixo dos *xx*, no eixo dos *yy* e a rotação entre as duas imagens.

Nos sub-capítulos seguintes irão ser analisadas em detalhe as duas funções, *Rota_escala.m* e *Translacao.m*.

4.2.2 Determinação da Rotação

O ângulo de rotação entre as duas imagens, *imageref* e *imagein*, é determinado no módulo *Rota_escala.m* o qual se encontra representado no fluxograma da Figura 4.2.

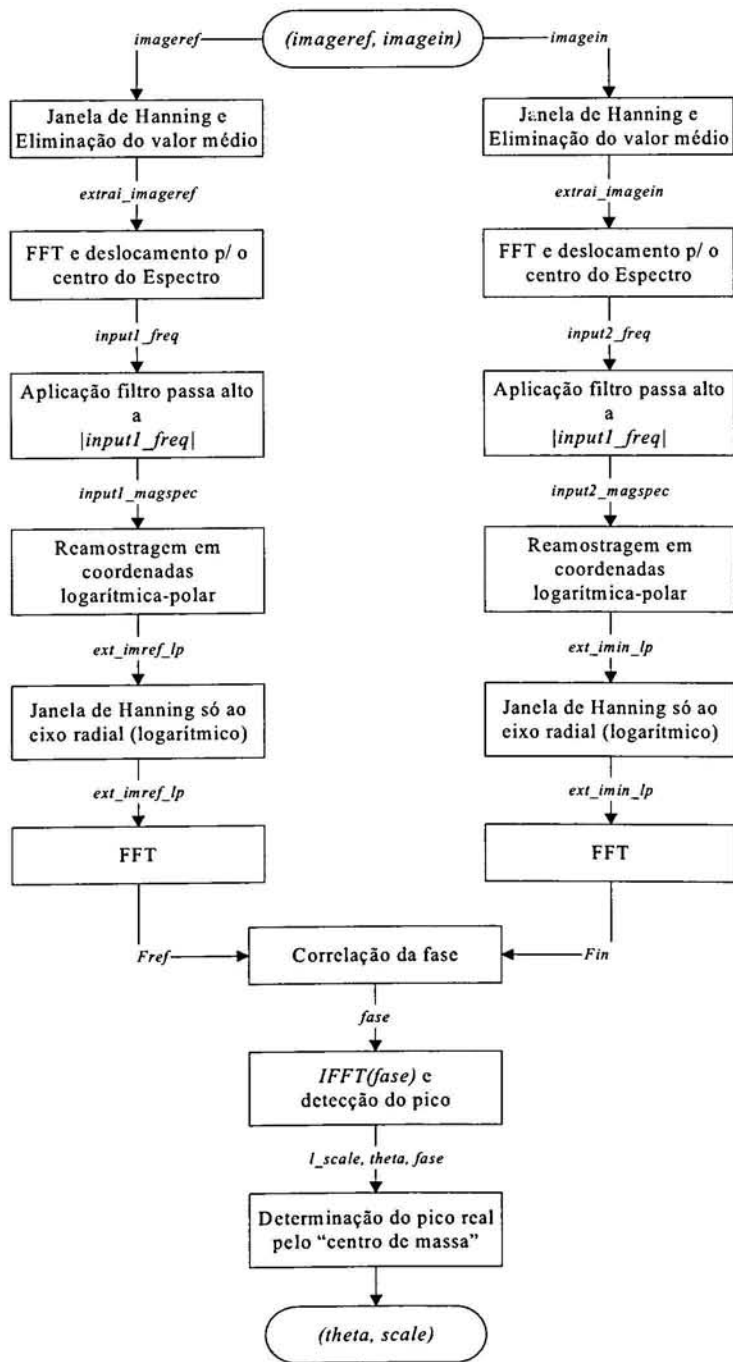


Figura 4.2 Fluxograma do modulo de determinação da rotação, *Rota_escala.m*.

O fluxograma mostra uma simetria de processamento das duas imagens até ao cálculo da correlação da fase. Como tal, basta analisar o processamento para uma das imagens.

Para a determinação do ângulo de rotação, como foi visto no capítulo três, é necessário transformar as imagens, reamostrar o espectro das amplitudes das imagens, de coordenadas cartesianas para coordenadas logarítmicas – polares, calcular as suas transformadas de Fourier, calcular a correlação das fases e calcular uma transformada inversa.

As imagens *imageref* e *imagein* são funções discretas puramente reais. Como a transformada discreta de Fourier e a sua inversa assumem uma função periódica e a imagem é truncada, corremos o risco de ter *aliasing*.

Uma forma de evitarmos o *aliasing* consiste em aplicar uma janela de Hanning às imagens, janela esta que pode ser visualizada na Figura 4.3.

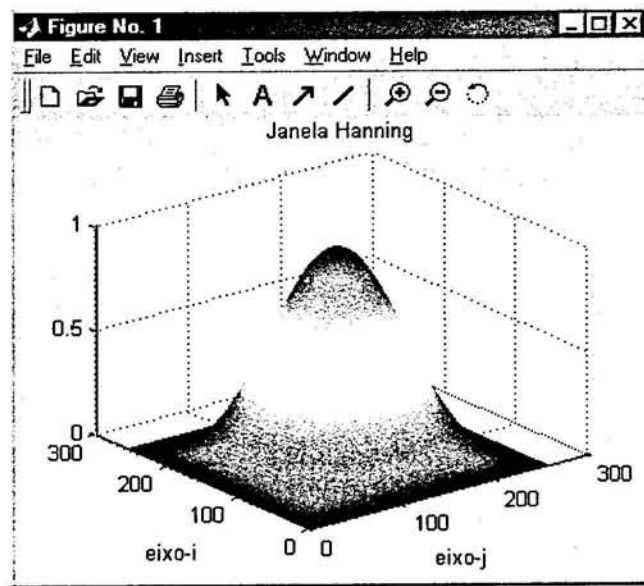


Figura 4.3 Janela de Hanning aplicada às imagens *imageref* e *imagein*.

Esta janela, para uma imagem de dimensão $N \times N$, tem como função a indicada na equação (69).

$$W_j = \frac{1}{2} \left[1 - \cos \frac{2\pi j}{N-1} \right], \quad j = 0, 1, \dots, N-1. \quad (69)$$

A função (69) tem o máximo no ponto $\left(\frac{N-1}{2}, \frac{N-1}{2}\right)$ e tem o valor zero nas extremidades, como pode ser verificado na Figura 4.3.

Ao aplicarmos a janela de Hanning à imagem de referência *imageref*, obtemos a Figura 4.4.

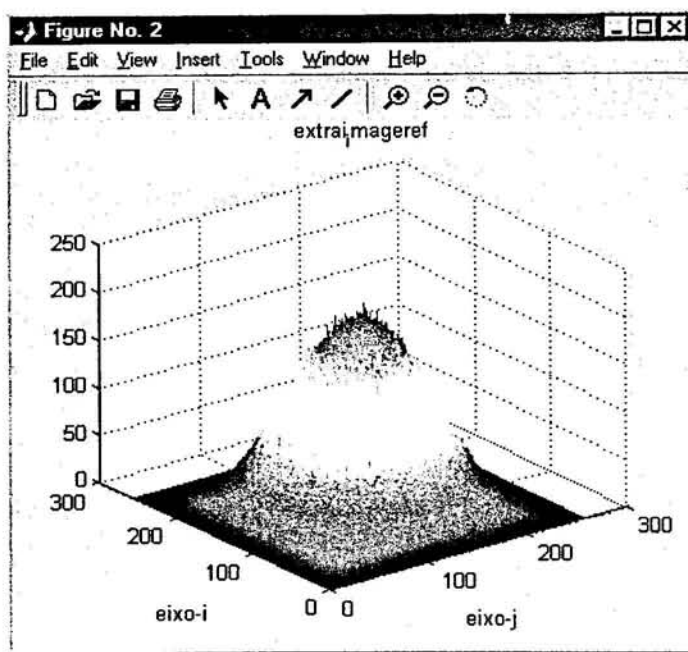


Figura 4.4 Imagem de referência com janela de Hanning aplicada.

Podemos verificar nesta figura que os valores das extremidades são zero, o que nos permite ter uma zona bem delimitada da imagem.

Ao resultado da aplicação da janela de Hanning à imagem de referência, a que se chamou *extrai_imageref*, é aplicada a transformada de Fourier seguida de um deslocamento para o centro do espectro. Assim obtemos como resultado o sinal *input1_freq*, representado na Figura 4.5.

Este procedimento pressupõe que previamente é feito um ajuste da imagem para ficar com valor médio nulo, por subtração do seu valor médio a cada pixel.

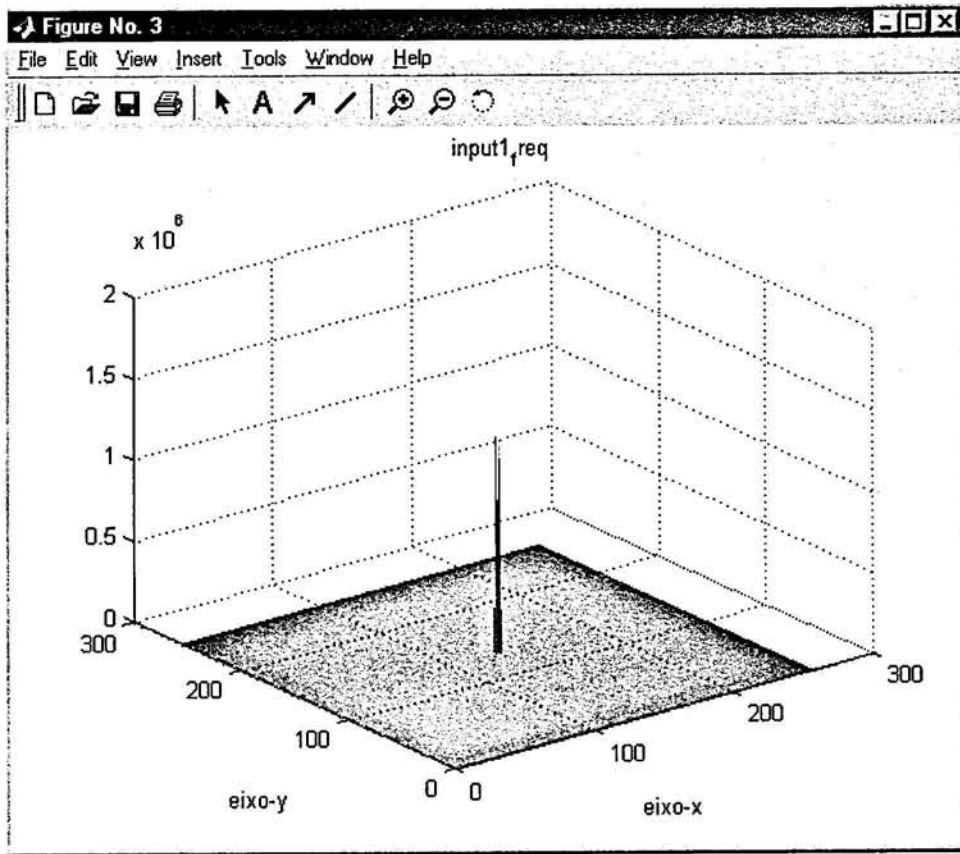


Figura 4.5 Transformada de Fourier e deslocação para o centro do espectro.

O deslocamento é feito para o centro do espectro, porque ao passarmos para coordenadas logarítmicas – polares – temos que ter em atenção à instabilidade numérica próxima da origem, uma vez que $\lim_{r \rightarrow 0} \rho = \lim_{r \rightarrow 0} \ln r = -\infty$ nas equações (67) e (68), em que r representa os valores no eixo logarítmico. Este eixo traduz o factor de escala.

Para obtermos a amplitude, tendo em consideração a instabilidade numérica referida, é aplicado um filtro passa alto de realce (*hipass emphasis filter*) segundo B. S. Reddy and B.N. Chatterji [100], ver Figura 4.6, com a função de transferência.

$$H(x, y) = (1 - \cos \pi x \cos \pi y)(2 - \cos \pi x \cos \pi y), \quad -0,5 \leq x, y \leq 0,5. \quad (70)$$

O filtro passa alto utilizado, tem o valor zero no ponto $\left(\frac{N-1}{2}, \frac{N-1}{2}\right)$ e o valor máximo dois nas extremidades, como pode ser visualizado na Figura 4.6.

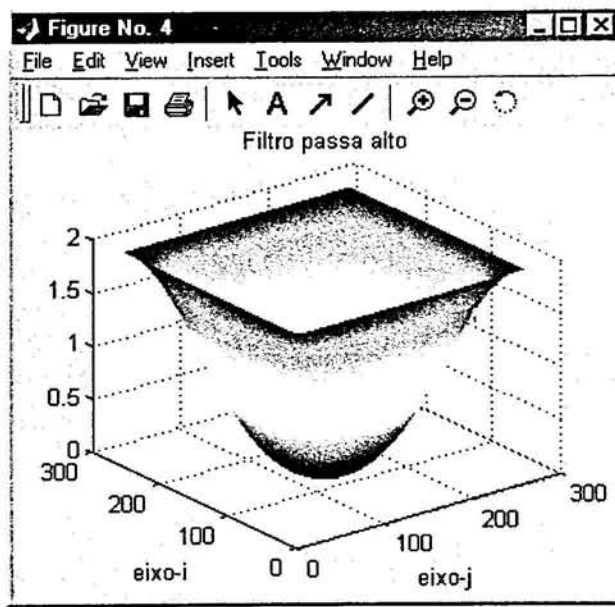


Figura 4.6 Filtro passa alto para extracção da amplitude.

Ao aplicarmos este filtro passa alto ao módulo do sinal *input1_freq*, obtemos o espectro da amplitude da imagem original. Esta contém a informação da rotação e da escala independentemente da translação, uma vez que a translação só afecta a fase. Ficamos assim com a função designada por *input1_magspec*, representada na Figura 4.7.

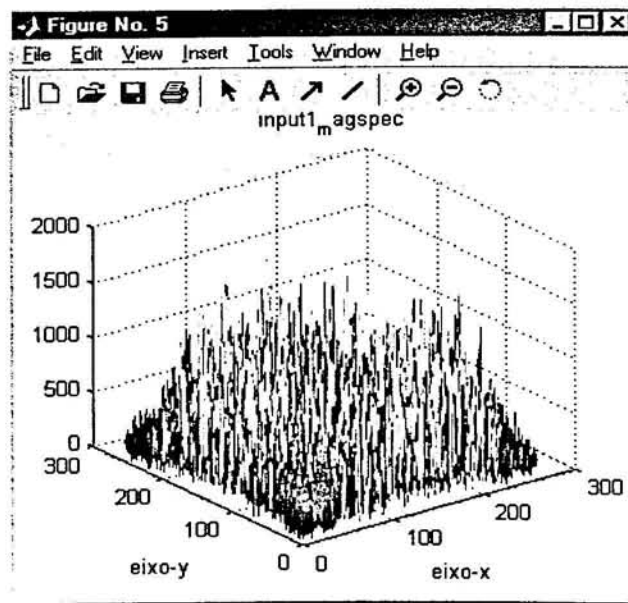


Figura 4.7 Espectro da amplitude da imagem original.

A função *input1_magspec* está em coordenadas cartesianas e o que pretendemos é em coordenadas logarítmicas – polares. Então o passo a seguir é converter para estas coordenadas.

A conversão para as coordenadas logarítmicas – polares é executada tendo como base a função *polartrans(im, nrad, ntheta, cx, cy, linlog, shape)*, desenvolvida em Matlab por Peter Kovesi [112].

A função *polartrans* recebe como parâmetros de entrada, a imagem *im* em coordenadas cartesianas, o número máximo de valores a converter para cada eixo, em que *nrad* é o máximo para o eixo radial e *ntheta* é o máximo para o eixo angular, o ponto de origem para a conversão *cx* e *cy*, normalmente o centro da imagem, o tipo de conversão para o eixo radial *linlog*, podendo ser linear (*'linear'*) ou logarítmico (*'log'*), e o parâmetro *shape*, que tem duas possibilidades, ou retorna as coordenadas polares de toda a imagem de entrada (*'full'*), ou retorna só os valores que estiverem dentro do círculo maior que pode ser definido na imagem (*'valid'*), o que apenas acontece na situação de termos os valores de origem *cx* e *cy* diferentes do centro da imagem.

A função referida retorna a imagem em coordenadas polares ou em coordenadas logarítmicas – polares com o eixo radial associado às linhas e o eixo angular associado às colunas. O tamanho da imagem é *nrad* x *ntheta*.

No trabalho a função foi usada da seguinte forma:

.....

```
[i_r, j_r] = size(extrai_imageref(:, :, 1));
```

```
cx_r = j_r/2+.5; % adicionar 0.5 porque o índice começa em 1
```

```
cy_r = i_r/2+.5;
```

.....

```
%Passa de cartesiana para log-polar,
```

```
ext_imref_lp=polartrans(input1_magspec(:, :, 1), 30, 360, cx_r, cy_r, 'log', 'full');
```

.....

em que os parâmetros de entrada são a imagem *input1_magspec*, o valor máximo de 30 para o eixo radial, o valor máximo de 360 para o eixo angular, o centro da imagem *cx_r* e *cy_r*, como ponto de origem, a string 'log' para transformar o eixo radial em logarítmico e a string 'full' para considerar a imagem toda *input1_magspec*.

O resultado obtido está representado na Figura 4.8 e foi designado por *ext_imref_lp*, tendo sido obtido por reamostragem, através de uma interpolação linear da imagem *input_magspec* executada pela função *polartrans.m*.

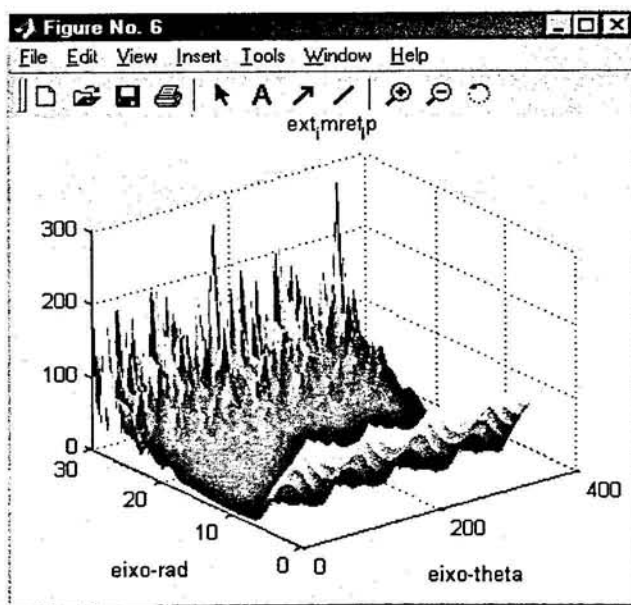


Figura 4.8 Espectro da amplitude da imagem original em coordenadas logarítmicas – polares.

A passagem de coordenadas cartesianas para logarítmicas polares é feita de uma só vez, mas poderia fazer-se primeiro uma reamostragem para coordenadas polares e depois uma para coordenadas logarítmicas – polares.

O espectro da amplitude normalmente tem uma gama de valores muito grande e pode provocar nas extremidades do eixo radial descontinuidades. Estas descontinuidades podem provocar *aliasing* ao calcular a transformada de Fourier do espectro da amplitude em coordenadas logarítmicas polares (*ext_imref_lp*).

Para evitarmos o *aliasing* aplicamos uma janela de Hanning ao eixo radial antes de

calcular a transformada de Fourier. Podemos verificar a diferença na Figura 4.9, em que temos em a) representado a Figura 4.8 segundo o eixo radial e em b) segundo o mesmo eixo, mas com a janela de Hanning aplicada.

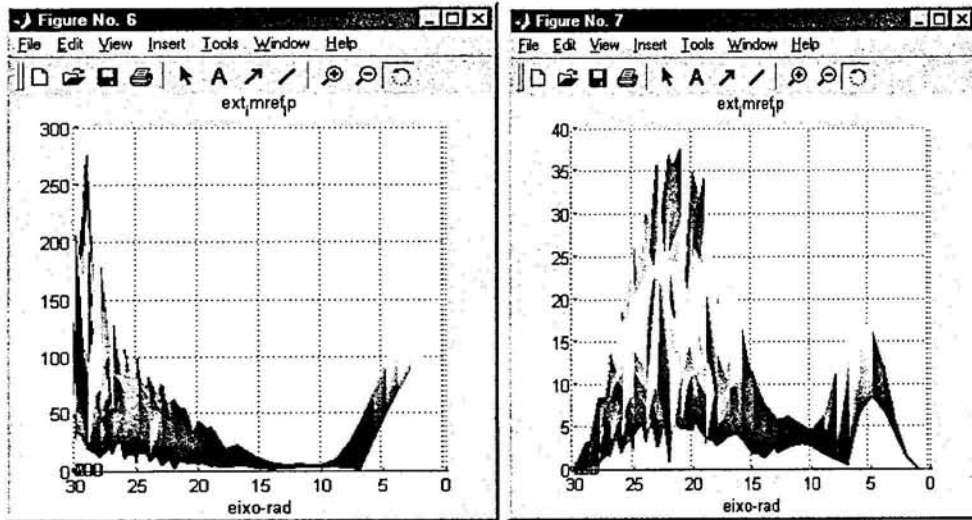


Figura 4.9 Espectro da amplitude da imagem original em coordenadas logarítmicas – polares, a) segundo o eixo radial, b) segundo o eixo radial, mas com janela de Hanning aplicada a este.

Depois da janela de Hanning aplicada pode ser calculada a transformada de Fourier do espectro da amplitude *ext_imref_lp*, e obtemos a função *Fref* que pode ser visualizada na Figura 4.10.

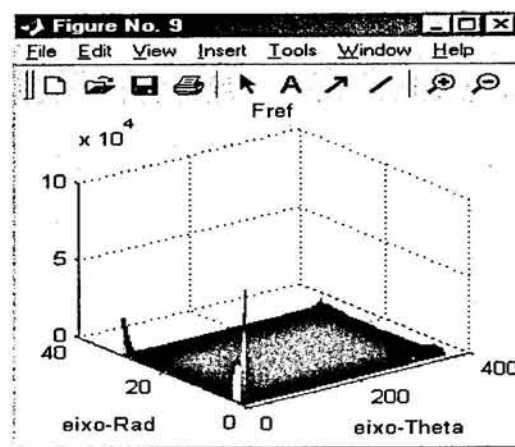


Figura 4.10 Transformada de Fourier do espectro da amplitude *ext_imref_lp*.

Todos os passos e cálculos efectuados para a imagem de referência *imageref* serão também efectuados para a imagem a comparar *imagein*, em que obtemos como resultado *Fin*. Este resultado encontra-se representado na Figura 4.11.

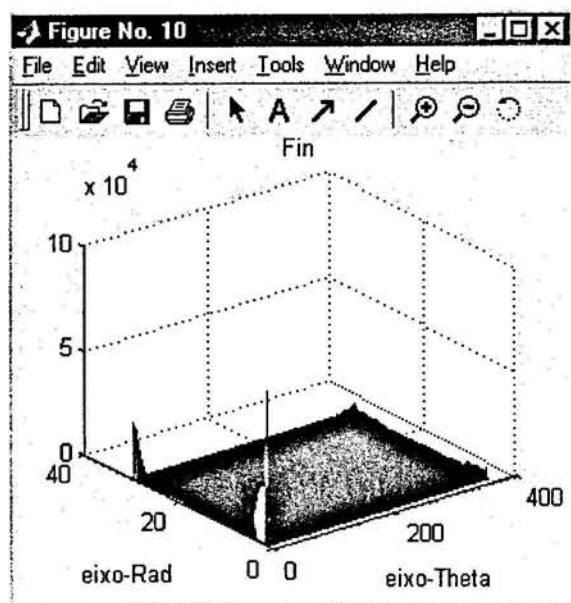


Figura 4.11 Transformada de Fourier do espectro da amplitude *ext_imin_lp*.

A partir das transformadas de Fourier *Fref* e *Fin*, faz-se a correlação cruzada da fase, ou seja aplica-se a equação,

$$fase = \mathfrak{F}^{-1} \left\{ \frac{Fin(Fref)^*}{|Fref|} \right\}, \quad (71)$$

em que \mathfrak{F}^{-1} é a transformada inversa de Fourier. O resultado obtido é a *fase*, de onde retiramos a rotação pretendida e a escala.

Este resultado pode ser visto graficamente na Figura 4.12.

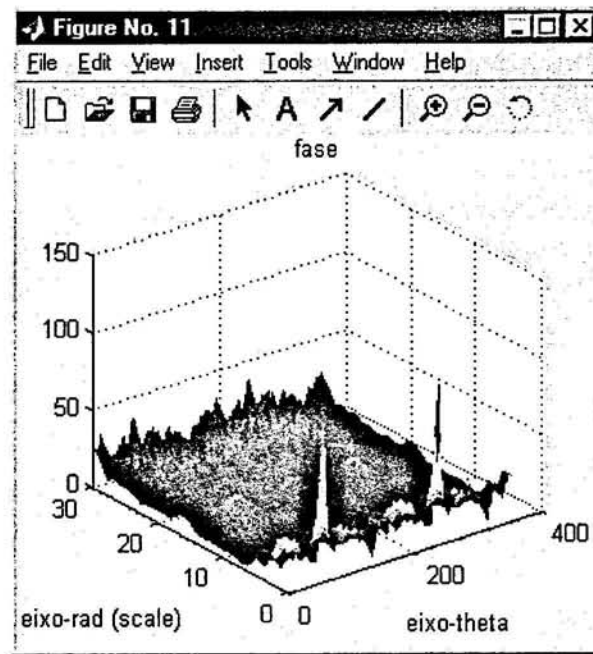


Figura 4.12 Resultado da correlação de fase.

Verificamos que podemos ter dois picos ou só um, dependendo das imagens *imageref* e *imagein*, em que na situação de dois picos um diz respeito ao ângulo θ e o outro diz respeito ao ângulo $\theta+180^\circ$, e na situação de um só pico este pode ser θ ou $\theta+180^\circ$.

O facto do ângulo poder ser θ ou $\theta+180^\circ$ tem a ver com a propriedade da periodicidade e da simetria do conjugado (funções puramente reais) da DFT, revista no ponto 3.2.2.

As coordenadas do pico máximo dão-nos o valor do ângulo de rotação θ , assim como o valor do factor de escala *scale*, ao qual não será dada grande importância na nossa aplicação.

O valor correcto do ângulo de rotação será determinado no módulo *Translacao.m*, o qual será abordado em 4.2.3.

A localização do pico máximo, porque temos um sinal discreto, pode estar entre duas amostras. De forma a resolver este problema foi usado o método de determinação do centro de massa.

$$theta = \frac{\sum m_i x_i}{\sum m_i} \quad e \quad scale = \frac{\sum m_i y_i}{\sum m_i}, \quad (77)$$

em que m_i são os valores da matriz *fase*, vizinhos do ponto máximo, e x_i, y_i são respectivamente as coordenadas *theta* e *scale* desses pontos.

Fica assim concluída a descrição do módulo *Rota_escala.m*. Os valores *theta* e *scale* determinados são respectivamente o ângulo de rotação pretendido e o factor de escala. O valor do factor de escala apesar de ser determinado não será utilizado nesta dissertação.

4.2.3 Determinação da translação

O módulo que vamos analisar de seguida é o módulo *Translacao.m*. Este módulo é executado depois de obtermos o resultado do módulo *Rota_escala.m*, que vai ser usado como parâmetro de entrada do módulo *Translacao.m*, juntamente com as duas imagens iniciais, *imageref* e *imagein*. Como resultado obtemos o valor do deslocamento e da rotação entre as duas imagens.

O módulo pode ser analisado facilmente pelo fluxograma apresentado na Figura 4.13.

O fluxograma permite-nos verificar que temos como parâmetros de entrada as imagens *imageref*, *imagein* e o ângulo *theta*.

Como foi visto em 4.2.2 o ângulo de rotação pode ser *theta* ou $theta+180^\circ$. É neste módulo que verificamos qual dos dois ângulos é o correcto. Para isso a imagem *imagein* é rodada de $-theta$ e $-(theta+180^\circ)$, dando origem a duas imagens *imagein'* e *imagein_180*.

O próximo passo é determinar a correlação da fase entre a imagem de referência, *imageref* e cada uma das duas imagens, *imagein'* e *imagein_180*, obtendo-se as fases *phase* e *phase_180* respectivamente.

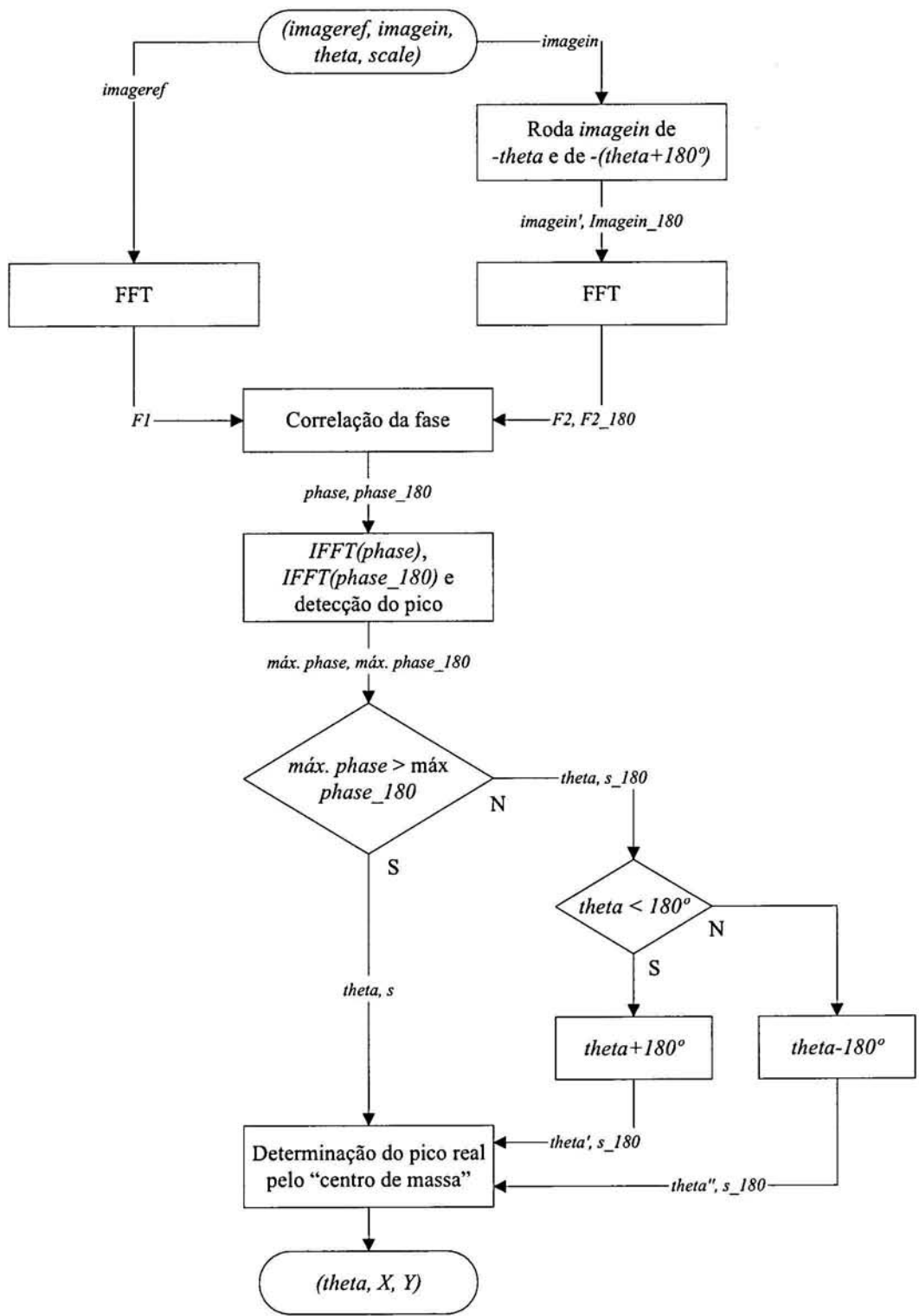


Figura 4.13 Fluxograma do módulo *Translacao.m*.

Para determinarmos as correlações de fase, começamos por calcular as transformadas de Fourier de cada uma das imagens *imageref*, *imagein* e *imagein_180*, respectivamente $F1$, $F2$ e $F2_{180}$,

$$phase = \mathfrak{F}^{-1} \left\{ \frac{F1(F2)^*}{|F2|} \right\} e \quad (73)$$

$$phase_{180} = \mathfrak{F}^{-1} \left\{ \frac{F1(F2_{180})^*}{|F2_{180}|} \right\}, \quad (74)$$

em que temos que \mathfrak{F}^{-1} é a transformada inversa de Fourier.

O resultado das duas correlações cruzadas, *phase* e *phase_180*, pode ser visto na Figura 4.14 e Figura 4.15, respectivamente, para o exemplo que temos vindo a utilizar para apresentação das figuras.

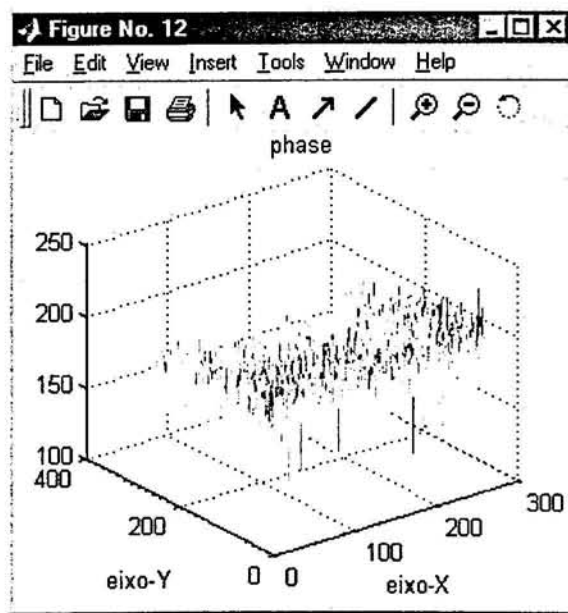


Figura 4.14 Correlação cruzada entre as transformadas de Fourier da imagem de referência *imageref* e da imagem de entrada *imagein* sem o ângulo de rotação *theta*.

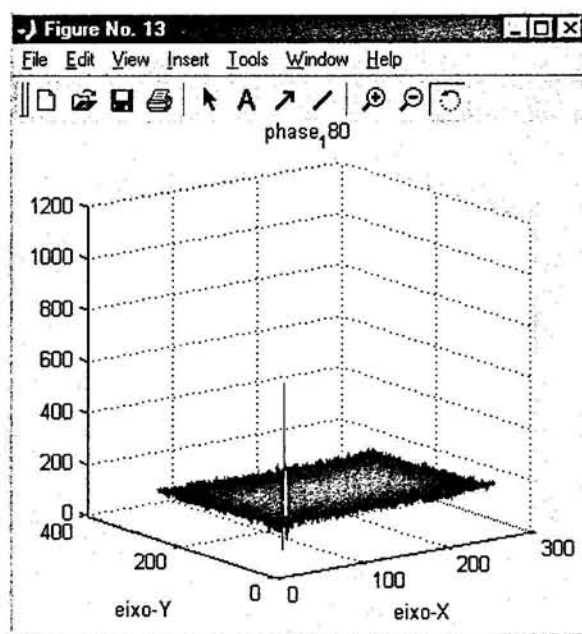


Figura 4.15 Correlação cruzada entre as transformadas de Fourier da imagem de referência *imageref* e da imagem de entrada *imagein* sem o ângulo de rotação $theta+180^\circ$.

Ao compararmos as duas figuras verificamos que, neste exemplo, o ângulo será $theta+180^\circ$, porque como podemos ver na Figura 4.14, não existe correlação entre a imagem de referência e a imagem *imagein* rodada de $-theta$. Por outro lado, na Figura 4.15, pode-se verificar que existe correlação entre a imagem de referência e a imagem *imagein* rodada de $-(theta+180^\circ)$.

A forma usada para detectar no Matlab qual dos dois ângulos é o correcto, consiste em detectar os máximos para cada figura e de seguida comparar os valores. O que apresentar o valor maior é que tem o valor correcto, pois como estamos perante uma situação em que pode haver ou não correlação, os máximos são muito distintos.

Na situação de o valor encontrado para o ângulo, ser $theta+180^\circ$, ainda temos que fazer mais uma verificação, pois $theta$ é inferior a 180° , então o ângulo correcto é $theta+180^\circ$, se não, é $theta-180^\circ$ e obtemos assim o valor correcto para $theta$. Isto quer dizer que temos o ângulo de rotação entre as duas imagens originais, *imageref* e *imagein*.

A determinação do deslocamento é feita determinando as coordenadas do ponto máximo da Figura 4.15.

À semelhança do que foi efectuado no módulo *rota_escala.m* para determinar quais as coordenadas do pico máximo verdadeiro, e uma vez que estamos perante sinais discretos onde o máximo pode estar entre dois valores consecutivos, foi usado o método de determinação do centro de massa representado,

$$X = \frac{\sum m_i x_i}{\sum m_i} \quad \text{e} \quad Y = \frac{\sum m_i y_i}{\sum m_i}, \quad (75)$$

em que m_i representa os valores da matriz *phase* ou *phase_180* vizinhos do ponto máximo e x_i, y_i são respectivamente as coordenadas x e y desses pontos.

Concluimos assim o módulo *translacao.m*. Os valores pretendidos e que serão desenvolvidos ao módulo principal *Navega.m* são o deslocamento X, Y e o ângulo de rotação *theta* entre as duas imagens de entrada *imageref* e *imagein*.

4.3 Limitações

O sistema desenvolvido tem algumas limitações, que passamos a analisar de seguida..

Uma das limitações mais críticas é o facto das imagens não poderem ter muitas simetrias, que podem provocar a determinação de parâmetros errados. A situação mais evidente é a apresentada no capítulo 1 (Figura 1.3) com as duas imagens I_1 e I_2 a serem exactamente iguais.

Outra das limitações traduz-se no facto das imagens terem que ter sempre uma parte comum porque senão não há possibilidade de determinar os parâmetros, uma vez que não há relação entre elas.

Tendo em conta que o factor de escala não está a ser considerado, as duas imagens têm que estar afectadas pelo mesmo factor de escala, senão será impossível determinar os valores pretendidos.

Por fim existe a limitação imposta pelo tempo de processamento, uma vez que este

sistema requer o cálculo de seis FFT's e três IFFT's, Figura 4.2 e Figura 4.3, mas tem a vantagem deste tempo ser constante e independente da rotação ou da translação ou seja, depende só da dimensão das imagens.

Capítulo 5

Descrição do sistema de simulação

A melhor forma de verificar o funcionamento do sistema desenvolvido, no âmbito desta dissertação, seria implementá-lo na prática. Como para se efectuar essa implementação era necessário despender muito tempo, e envolver muitos recursos, o que não foi possível e nem seria o objectivo principal desta dissertação, optou-se por criar um modelo de simulação tendo como base a ferramenta de programação Matlab.

O objectivo da simulação é validar e verificar de uma forma exaustiva se o sistema de navegação responde como pretendido ao problema proposto.

Da simulação é importante também retirar informação, expondo-a em gráficos, sobre as relações entre os parâmetros de projecto na implementação do sistema numa situação prática.

Todo o processo de simulação foi criado e desenvolvido em Matlab, principalmente por ser uma linguagem de programação simples, poderosa e muito fácil de trabalhar.

O modelo de simulação é constituído por dois módulos principais. Um diz respeito à criação de uma superfície e geração das imagens fotografadas dessa superfície, simulando a sequência de imagens que seriam obtidas a partir de uma câmara num veículo. O outro determina o percurso efectuado, executando o sistema desenvolvido e explicado no capítulo 4. A designação atribuída a estes dois módulos, foi respectivamente *Seqmov.m* e *Simula.m*.

Os dois módulos referidos vão ser descritos neste capítulo em pormenor, assim como os seus constituintes.

5.1 Geração das trajectórias

A geração das trajectórias tem como base o módulo designado por *Seqmov.m*. Este módulo simula a aquisição de imagens de trajectos, percorridos pelo veículo, numa determinada superfície.

A superfície é constituída por vários mosaicos, conforme a dimensão pretendida. O mosaico, por sua vez é uma fotografia, imagem, qualquer.

O trajecto então é obtido por fotografias consecutivas nessa superfície, simulando o trajecto do veículo.

5.1.1 Geração da sequência de imagens da trajectória, *Seqmov.m*

A obtenção da sequência de imagens que dizem respeito ao trajecto executado pelo veículo, faz-se recorrendo ao módulo principal *Seqmov.m* cujo fluxograma está representado na Figura 5.1.

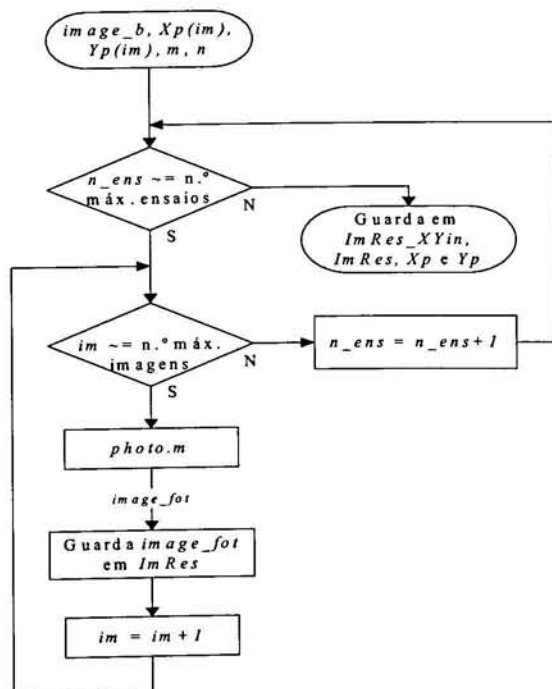


Figura 5.1 Fluxograma do módulo de obtenção da sequência de imagens da trajectória, *Seqmov.m*.

Como já foi referido, o objectivo do módulo *Seqmov.m* e dos módulos interligados com este, é simular a deslocação de um determinado veículo sobre uma superfície. Para isso temos que gerar uma superfície e retirar imagens espaçadas entre si, como se fosse o veículo a circular com uma câmara associada.

O módulo *Seqmov.m* recebe uma imagem *image_b*. Neste caso concreto usou-se uma fotografia (640x480) de um determinado piso industrial, Figura 5.2, a qual vai ser utilizada para criar a superfície de base para a obtenção das imagens pretendidas. A superfície de base é constituída por $mv+1$ mosaicos da *image_b* na vertical e $mh+1$ mosaicos na horizontal. O módulo responsável pela criação da superfície total, utilizado por *photo.m* é designado por *Cria_chao.m* e será explicado mais adiante.

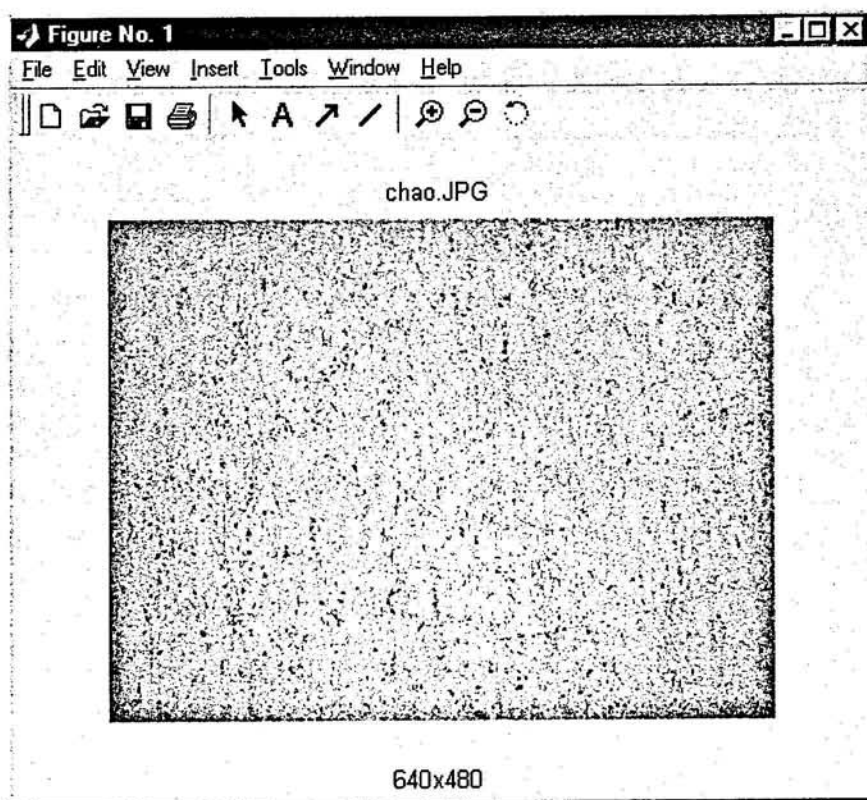


Figura 5.2 Fotografia base usada para a criação da sequência de imagens

Outros parâmetros de entrada do módulo *Seqmov.m* são as matrizes $Xp(im)$ e

$Yp(im)$, dos pontos iniciais de cada imagem do trajecto, e também a dimensão destas, que é dada pelos parâmetros m e n , definindo respectivamente a dimensão no eixo dos xx e no eixo dos yy . Ficamos assim com toda a informação necessária para conseguir uma sequência de imagens da superfície, correspondente a um trajecto. A quantidade de imagens é definida pela dimensão das matrizes $Xp(im)$, $Yp(im)$ e é indicada pela variável im .

Podem ser definidas várias matrizes (n_ens) de pontos para diferentes trajectos na superfície e são geradas assim várias sequências de imagens conforme o número de trajectos que pretendemos. Isto permite facilmente fazer comparações entre os resultados obtidos para os diferentes trajectos.

Quando todos os valores necessários já estão definidos, chama-se a função *photo.m*. Esta é a responsável por gerar as diferentes imagens da superfície a partir dos pontos $Xp(im)$, $Yp(im)$.

Cada imagem é guardada na variável *ImRes*, e por sua vez esta é guardada no ficheiro *ImRes_XYin* assim como os pontos de origem Xp , Yp de cada imagem.

A seguir será analisado o módulo *Photo.m*.

5.1.2 Geração das imagens de um trajecto numa superfície

O módulo *Photo.m* é o módulo que executa a função *Cria_chao.m*, para gerar a superfície. Este módulo é usado também para retirar uma imagem da superfície, isto é, fazer uma fotografia, no ponto Xp , Yp e com a dimensão pretendida. A imagem é obtida recorrendo a algumas funções Matlab, e a outras desenvolvidas no âmbito desta tese.

A interligação entre as diferentes funções, utilizadas no módulo *Photo.m*, pode ser vista no fluxograma representado na Figura 5.3.

Da análise do fluxograma podemos verificar que o módulo *Photo.m* recebe a imagem base *image_b*, o ponto Xp , Yp , a dimensão da fotografia m , n e pode receber um ângulo *theta*, permitindo aplicar uma rotação à imagem pretendida. Este módulo devolve uma imagem, com as características definidas, designada por *image_fot*.

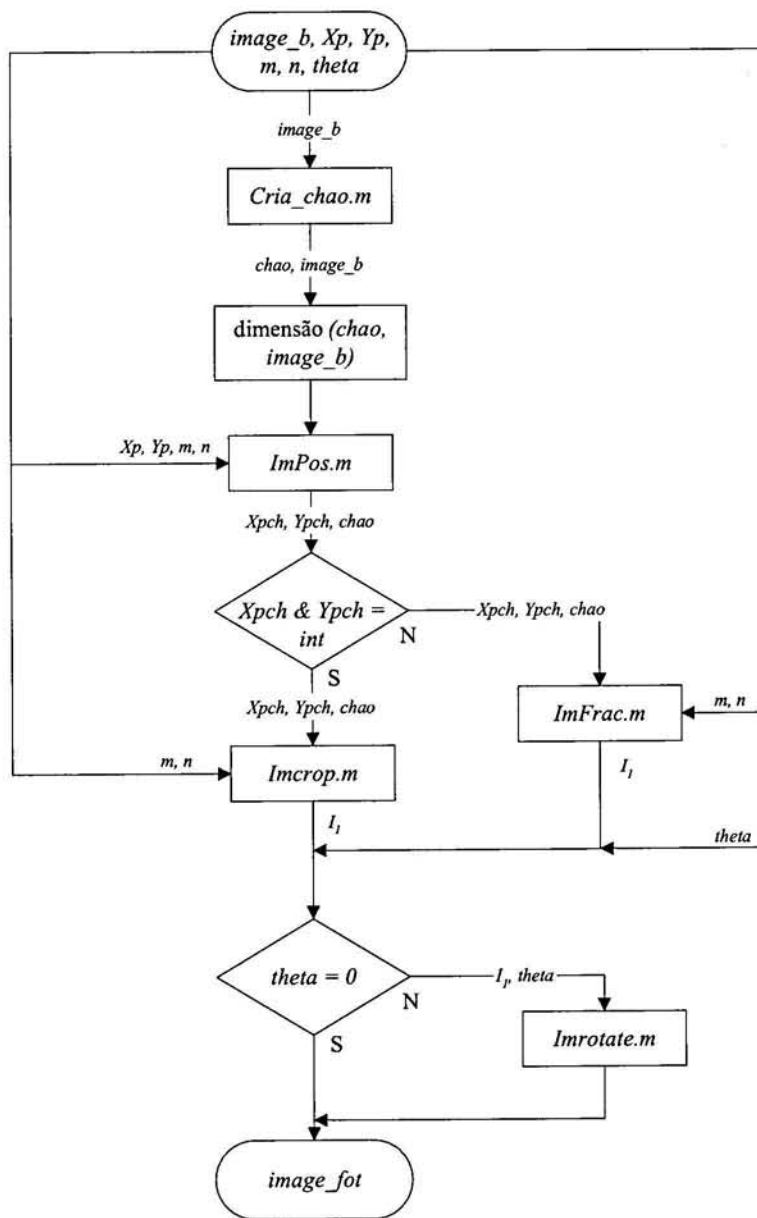


Figura 5.3 Fluxograma do módulo de fotografia da superfície nos pontos pretendidos, *Photo.m*.

O ângulo *theta*, visualizado no fluxograma, não foi usado, porque a simulação da rotação vai ser feita no módulo de simulação.

Se imaginarmos um veículo com uma câmara instalada, a mover-se no solo e a fotografá-lo com uma certa cadência, as imagens sucessivas estarão em princípio rodadas

uma relativamente à outra, Figura 5.4.

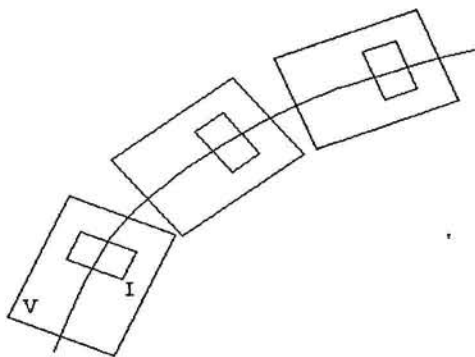


Figura 5.4 Trajecto em curva com as imagens *I* afectadas por rotação e translação do veículo *V*.

A simulação desta rotação depende do modo como se faz a condução do veículo nomeadamente de quais são as suas rodas directrizes. Este é um assunto fora do âmbito deste trabalho.

Estamos apenas interessados em saber se o método suporta a rotação. Para simplificar o problema optou-se por imaginar que o veículo só realiza movimento de translação, o que não limita o trajecto realizado, Figura 5.5.

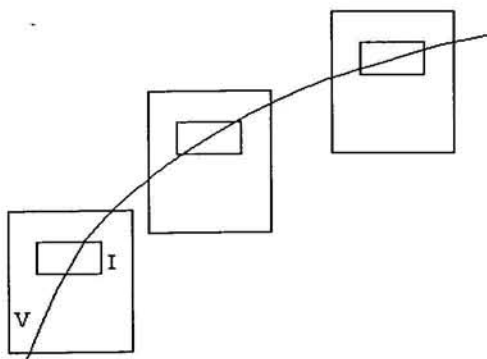


Figura 5.5 Trajecto em curva com as imagens *I* afectadas só por translação do veículo *V*.

No módulo de simulação introduziu-se um parâmetro que permite testar como o método se comporta relativamente à rotação.

A primeira função a ser chamada pelo módulo *Photo.m* é *Cria_chao.m*. Esta função foi desenvolvida, no âmbito desta tese, e tem como objectivo criar uma superfície a partir de uma imagem base *image_b*. Esta superfície tem a dimensão de $mv+1$ imagens na vertical por $mh+1$ imagens na horizontal.

A função *Cria_chao.m* pode ser vista no fluxograma da Figura 5.6.

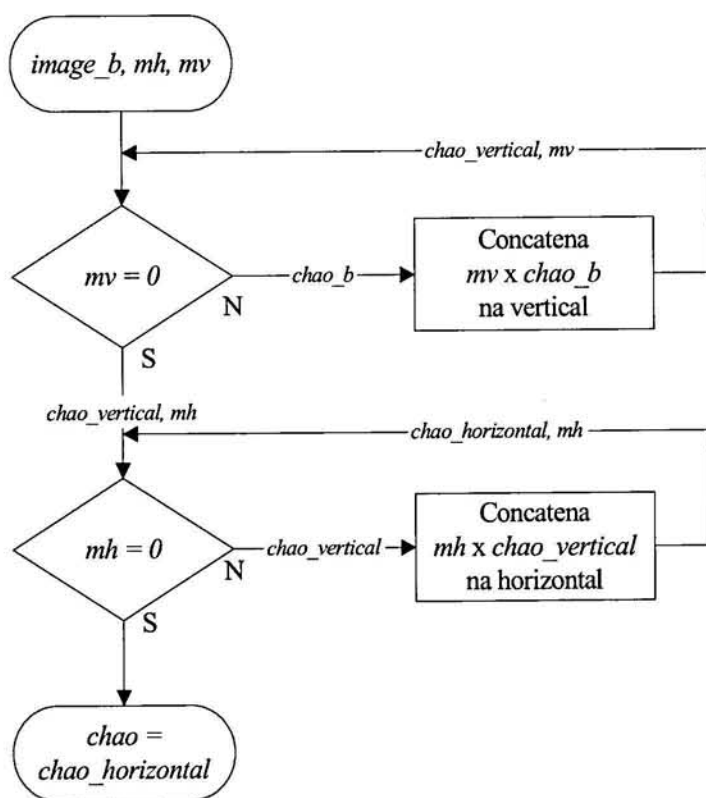


Figura 5.6 Fluxograma do módulo *Cria_chao.m*, de criação da superfície com a dimensão $(mv+1)$ por $(mh+1)$.

O número necessário de mosaicos para construir a superfície é de 4 ($mv=mh=1$) porque qualquer ponto da imagem pretendida pode sempre ser convertido num ponto dentro desses 4 mosaicos. A superfície encontra-se representada na Figura 5.7, e tem a dimensão de $2*640$ no eixo dos xx e $2*480$ no eixo dos yy .

Se a dimensão da fotografia pretendida fosse superior à dimensão da superfície o número de imagens na vertical e na horizontal teria que aumentar. Se pelo contrário a dimensão da fotografia for menor que a da superfície de base, que é a situação neste trabalho, pode-se garantir que para qualquer que seja o valor do ponto inicial, todas as fotografias serão obtidas a partir desta superfície.

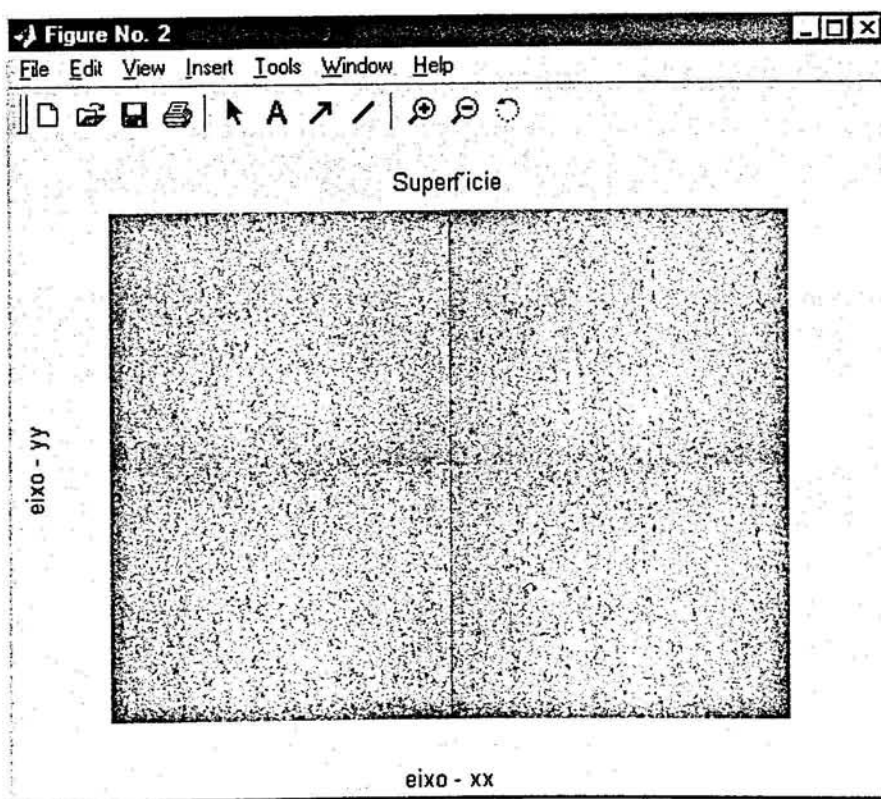


Figura 5.7 Superfície (*chao*) criada a partir de uma imagem base (*image_b*).

A função que determina a posição inicial da imagem pretendida na superfície de quatro mosaicos é a função designada por *Impos.m* e será executada a partir do módulo *Photo.m*. Esta função está representada pelo fluxograma da Figura 5.8.

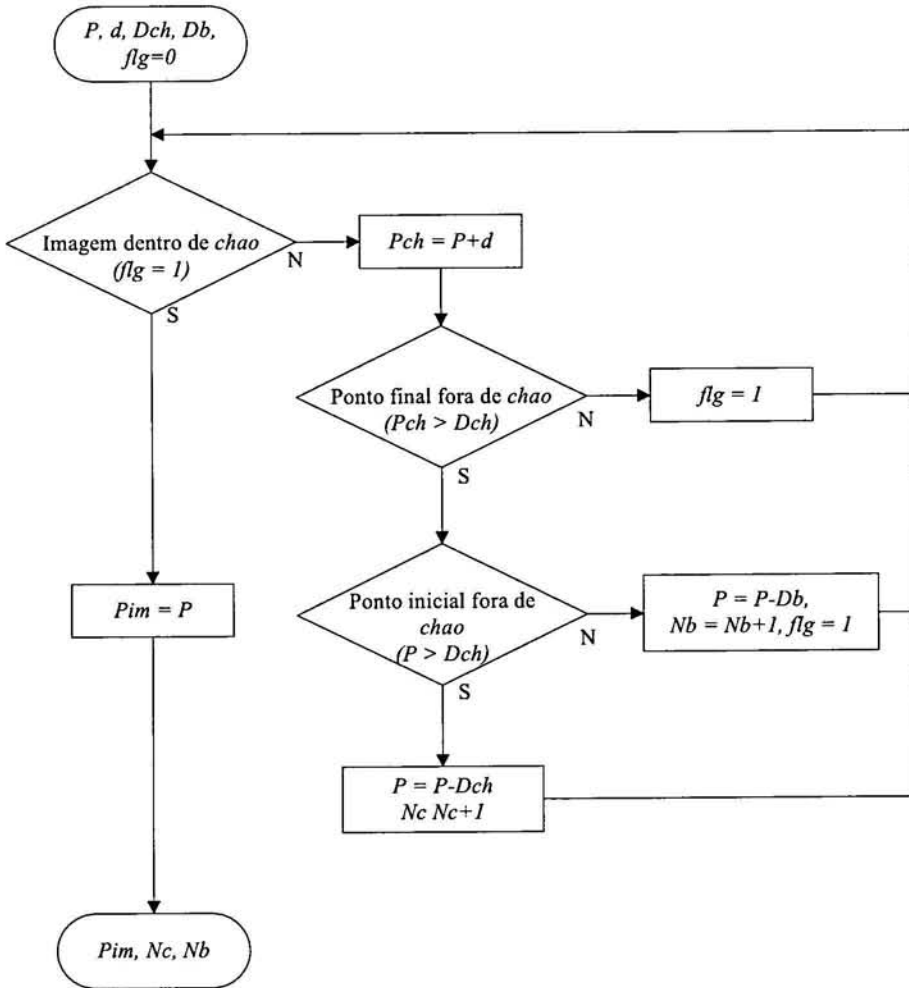


Figura 5.8 Fluxograma do módulo de posicionamento do ponto (X, Y) , do início da fotografia, na superfície *Impos.m*.

Impos.m recebe como parâmetros de entrada, representados na Figura 5.9, a dimensão da superfície *chao* (*Dch*), da imagem base *image_b* (*Db*), da fotografia pretendida (*d*) e do ponto inicial (*P*) da imagem a fotografar. Esta função será executada duas vezes. Uma para o eixo dos *xx* com o ponto inicial a corresponder a *Xp* e a dimensão da fotografia pretendida a corresponder a *m*, e outra para o eixo dos *yy* com o ponto inicial a corresponder a *yp* e a dimensão da fotografia pretendida a corresponder a *n*.

ponder a Yp e a dimensão da fotografia a corresponder a n .

A função *Impos.m*, primeiro verifica se a imagem pretendida se encontra dentro ou fora da superfície. Com esse intuito começa-se por verificar se o ponto inicial P mais a dimensão no eixo respectivo d , cujo resultado é o ponto final da imagem Pch , é maior ou menor que a dimensão Dch . Se for menor, é porque a imagem pretendida faz parte da superfície e então retorna o valor do ponto inicial P .

Na Figura 5.9, por outro lado, está representado a traço contínuo a situação de pretendemos uma imagem cujo ponto inicial P (1100x800) encontra-se dentro da superfície e o final $Pch=P+d$ fora desta, com $d=256$.

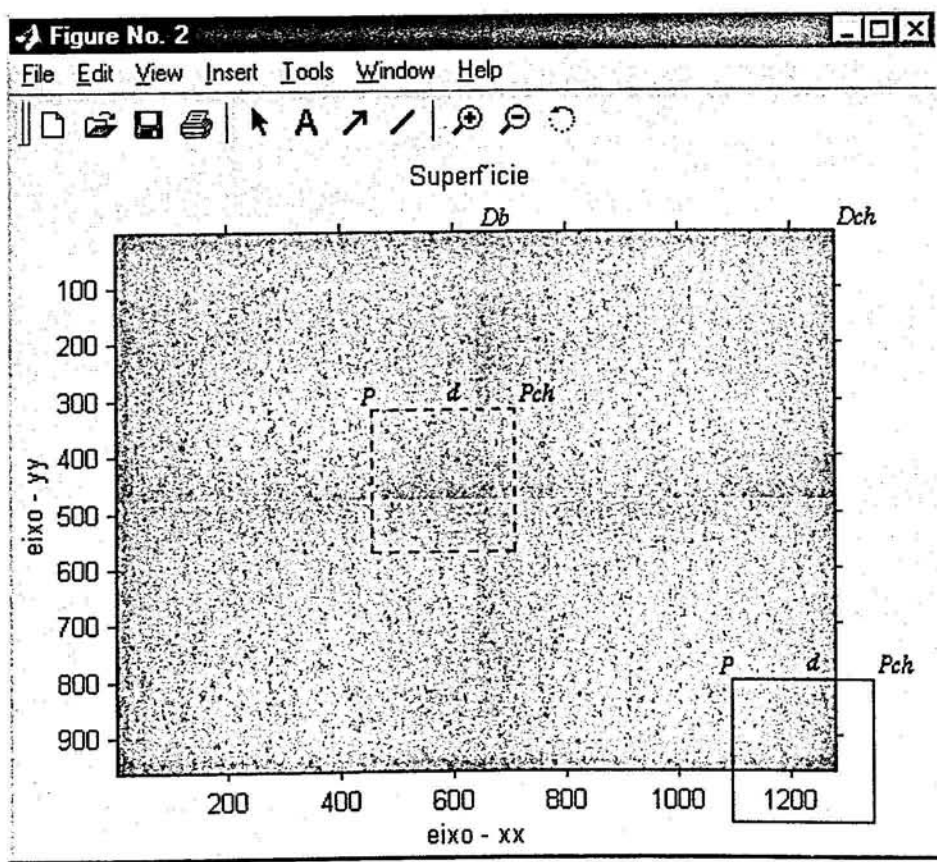


Figura 5.9 Superfície *chao*, exemplo de localização da área a fotografar.

No caso da situação descrita e representada na Figura 5.9, o ponto inicial P da imagem pretendida está dentro da superfície e o ponto final Pch fora desta, para a imagem

pertencer à superfície tem que se subtrair ao ponto P a dimensão Db da imagem $image_b$.

Ao fazer o mesmo para o outro eixo, passamos a ter a imagem representada a traçado na Figura 5.9.

Quando os dois pontos P e Pch pertencerem ao exterior da superfície, então o valor a subtrair ao ponto P será a dimensão da superfície Dch . Este valor será subtraído tantas vezes quantas as necessárias, até o ponto inicial P pertencer à superfície $chao$. A cada subtração é verificado se ambos os pontos pertencem ou não à superfície, até se obter uma das duas situações já referidas.

A partir do momento que temos os dois pontos P e Pch a pertencer a $chao$, a função $Impos.m$ devolve à função $Photo.m$ o ponto P determinado, assim como o número de vezes que teve que subtrair a dimensão quer da imagem $image_b$, quer da superfície $chao$. As variáveis usadas foram respectivamente Pim , Nb e Nc .

Nesta dissertação as duas variáveis Nb e Nc não foram usadas, porque como a primeira imagem obtida será a imagem de referência e todas as outras têm que ter partes em comum entre elas não é necessário a utilização destas duas variáveis. Poderia ser útil, se pretendêssemos relacionar o ponto inicial da imagem de referência com o ponto inicial da superfície.

Como já foi referido, a função $Impos.m$ será executada duas vezes pelo processo $Photo.m$ a fim de obter o ponto inicial para o eixo dos xx e para o eixo dos yy , respectivamente.

Após a obtenção do ponto inicial, $Xpch$ e $Ypch$ da imagem pretendida, verifica-se se são valores inteiros ou não. Se forem inteiros, então executa-se a função do Matlab $Imcrop.m$, senão executa-se a função desenvolvida pelo autor, designada por $Imfrac.m$.

A função $Imcrop.m$ retira da superfície $chao$, uma imagem com o ponto inicial $Xpch$ e $Ypch$, determinado por $Impos.m$, e com a dimensão da imagem pretendida m e n , devolvendo assim a imagem designada por I_1 ao módulo $Photo.m$.

A situação de termos valores do ponto inicial não inteiros é considerada porque, na realidade as imagens adquiridas por uma câmara, podem ser em qualquer posição da su-

perfície, ou seja temos valores de X e de Y reais, enquanto que na superfície que foi criada temos valores inteiros, pixels.

Por forma a resolver este problema foi então desenvolvida a função *ImFrac.m*, que faz o mesmo que a função *Imcrop.m*, mas recorre a uma reamostragem da imagem, a partir do ponto inicial não inteiro ($Xpch, Ypch$).

Esta reamostragem é feita através de uma interpolação bilinear.

O fluxograma da função *ImFrac.m* está esquematizado na Figura 5.10.

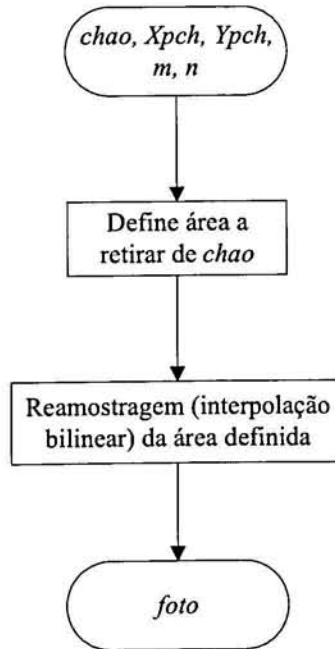


Figura 5.10 Fluxograma do módulo usado para retirar a fotografia pretendida, para pontos não inteiros, *ImFrac.m*.

Para concluir o módulo *Photo.m*, falta referir ainda a situação de se pretender aplicar uma rotação θ à imagem. Neste caso seria executada a função do Matlab *Imrotate.m* que rodaria a imagem de θ graus. Mas como já foi referido, na simulação não foi aplicada rotação da imagem a partir deste módulo, mas sim a partir do módulo *Simula.m*.

O módulo *Photo.m* devolve então ao *Seqmov.m* a imagem pretendida, e este guarda

essa imagem na variável *ImRes*, como se pode ver no fluxograma da Figura 5.1.

Depois de a guardar, verifica se ainda existem mais imagens para obter da superfície, e repete o processo até aqui descrito. Se não houver, verifica se as sequências de imagens estão todas adquiridas. No caso de ainda não estarem todas volta ao processo inicial já descrito.

Por fim serão guardadas, no ficheiro *ImRes_XYin*, todas as sequências de imagens obtidas referentes a um trajecto, assim como os pontos iniciais respectivos.

Fica assim concluído o processo de aquisição virtual de sequências de imagens, representando trajectos percorridos por um veículo, com uma câmara acoplada, sobre uma superfície.

A seguir resolve-se o problema inverso ou seja, a partir dessas imagens pretende-se obter o trajecto percorrido sobre a superfície, utilizando a metodologia desenvolvida.

5.2 Determinação da trajectória a partir das imagens da superfície

A simulação ficará completamente explicada, depois de ser apresentado o módulo que executa e permite tirar conclusões do sistema desenvolvido, ilustrado no capítulo 4.

O módulo é designado por *Simula.m*, e tem como principal objectivo determinar o trajecto percorrido pelo veículo a partir do conjunto de imagens adquiridas, permitindo também que seja feita uma comparação com o trajecto que o veículo percorreu na realidade, assim como calcular e gerar uma série de gráficos que permitirão a implementação na prática do sistema desenvolvido. *Simula.m* está esquematizado no fluxograma da Figura 5.11.

Os cálculos e gráficos gerados dizem respeito ao tempo de execução entre cada duas imagens, ao erro ponto a ponto, ao erro acumulado e à velocidade do veículo.

O erro ponto a ponto para o eixo x, y ou para o ângulo θ é determinado por

$$\varepsilon = \text{Valor ideal} - \text{Valor determinado}$$

e o erro acumulado é a soma de todos os erros ε ponto a ponto de um percurso.

No fluxograma da Figura 5.11 vemos que o módulo *Simula.m*, tem como parâmetros de entrada o ficheiro *ImRes_XYin* o qual contém as imagens dos percursos efectuados e os valores dos seus pontos iniciais, o ângulo *theta*, e também uma variável designada por *pix_cm* que faz a correspondência entre *cm* e *pixel*, ou seja 1 *cm* corresponde a *pix_cm* pixels.

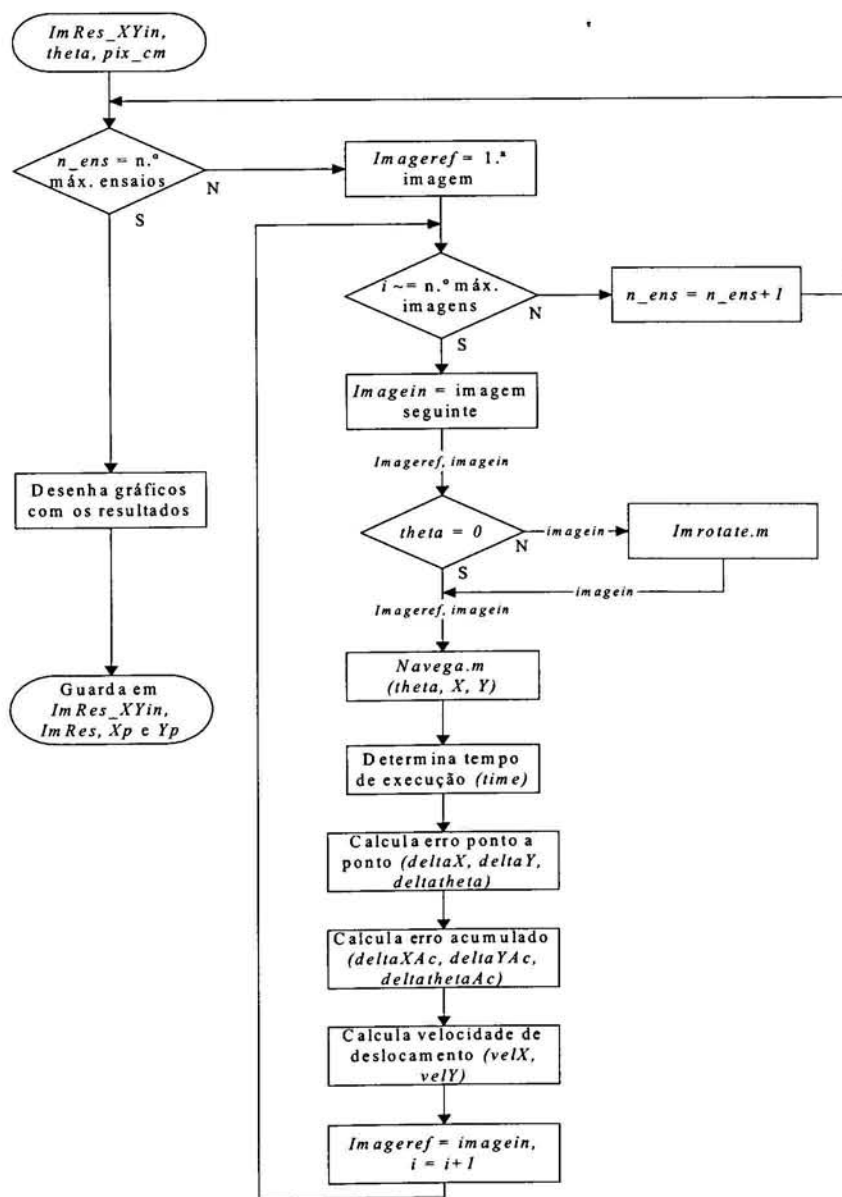


Figura 5.11 Fluxograma do módulo de obtenção da trajectória a partir de uma sequência de imagens, *Simula.m*.

A variável *pix_cm* destina-se a permitir determinar a velocidade em *cm/segundo* e o espaço percorrido, assim como o erro, em *cm*. Esta variável depende essencialmente da distância da câmara ao solo e da sua resolução.

O ângulo *theta* na realidade dá a orientação da câmara do veículo, como adiante será explicado.

O módulo *Simula.m* expõe em gráficos todos os resultados calculados, depois de terem sido determinados todos os trajectos existentes no ficheiro *ImRes_XYin*.

A determinação de um trajecto, a partir de um grupo de imagens, começa por considerar a primeira imagem desse grupo como imagem de referência *imageref* e a segunda como imagem a comparar *imagein*, Figura 5.12 a) e b) respectivamente.

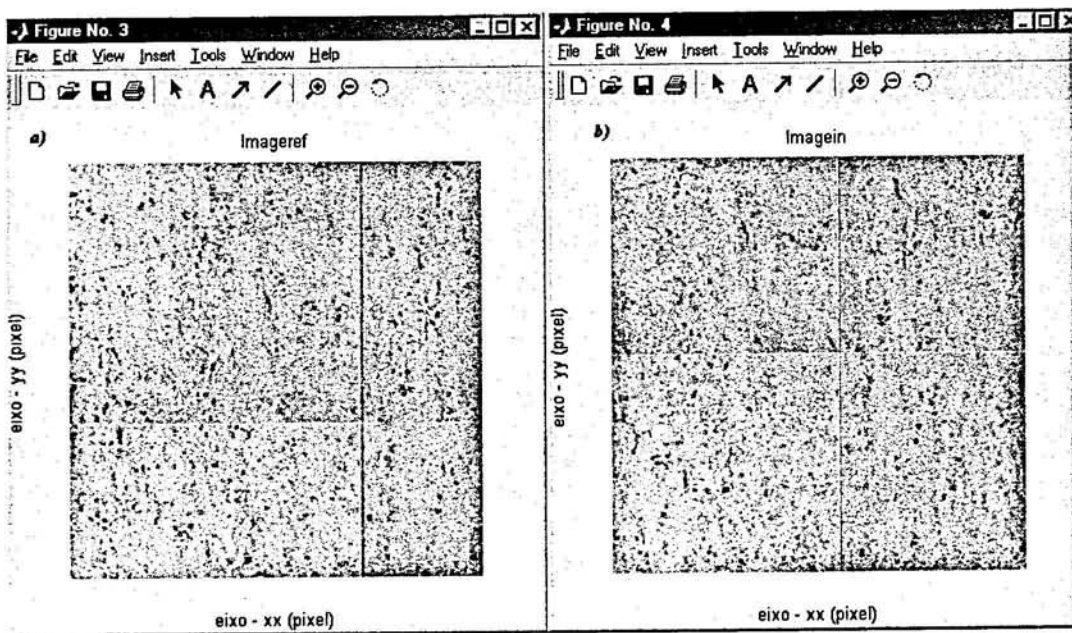


Figura 5.12 Duas primeiras imagens (256x256) de um trajecto sobre a superfície *chao*; a) imagem de referência *imageref* com o ponto inicial em 1100, 800; b) imagem a comparar *imagein* deslocada de 40 pixels, em relação a *imageref*, em ambos os eixos.

A sequência de imagens *ImRes_XYin* pressupõe, como se disse atrás, que o veículo só realiza movimento de translação Figura 5.5.

Mas como na realidade, numa curva, para além do deslocamento teremos uma rotação, a simulação também terá que a contemplar, por forma a verificar qual o comportamento do sistema na presença de rotação entre duas imagens.

A forma mais simples que o autor encontrou para simular a rotação, juntamente com o deslocamento, foi rodar a imagem *imagein* para cada par de imagens, ou seja, para cada par de imagens, *imageref* e *imagein*, afecta-se a imagem *imagein* de um ângulo *theta*.

Na Figura 5.13, a) e b) representam duas imagens sucessivas em que houve rotação do veículo entre elas e c), d) e e) representam duas imagens sucessivas em que só houve translação do veículo e em que procedeu-se à rotação por software.

Ao compararmos as duas imagens ilustradas b) e e), verificamos que são semelhantes, ou seja tanto uma como a outra têm o mesmo ângulo de rotação em relação à imagem base.

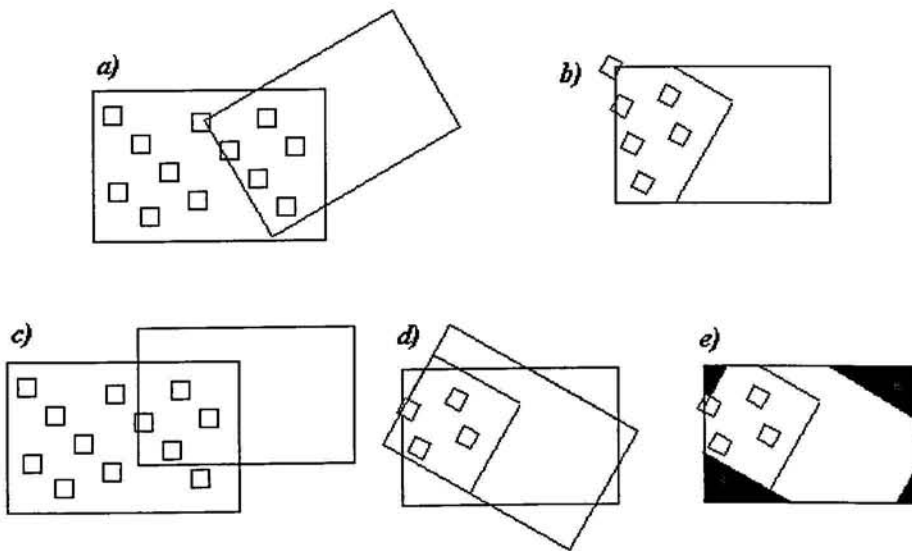


Figura 5.13 Diferentes métodos de rodar a imagem de um ângulo *theta*.

Se o ângulo *theta* é diferente de zero, então é aplicada uma rotação de *theta* à imagem *imagein*. Esta rotação é aplicada recorrendo à função do Matlab *Imrotate.m*. O ângulo *theta* pode ser inteiro ou fraccionário e pode ir de 1° a 360°.

A Figura 5.14 mostra a imagem *imagein* rodada de 20° .

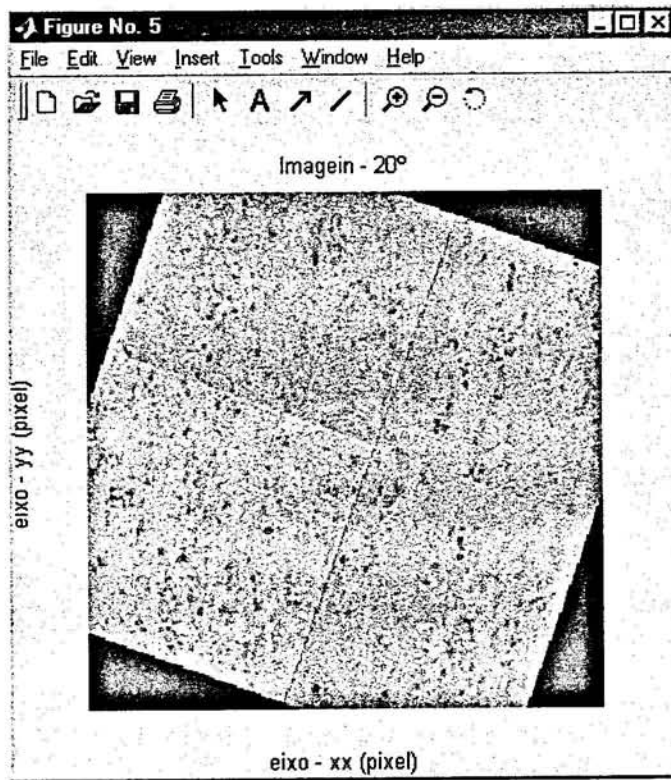


Figura 5.14 Imagem *imagein* rodada de um ângulo θ igual a 20° .

A partir das duas imagens *imageref* e *imagein*, com rotação θ ou não, executa-se a função *Navega.m*, descrita no capítulo 4, que determina qual a rotação θ e o deslocamento X, Y existentes entre estas imagens.

A função *Simula.m* também determina o tempo *time* que demorou a processar as duas imagens, usando a função Matlab designada por *cputime.m* que dá o tempo de *cpu*, tempo despendido pelo processador, para executar um determinado grupo de funções. Este tempo vai servir para determinar a velocidade máxima do veículo, assim como vai permitir comparar qual a dimensão das imagens mais adequada em cada situação, por forma a obter-se o melhor resultado possível.

Outros valores determinados pela função *Simula.m* são os erros. Estes podem ser determinados em cada ponto, ou seja, podem ser determinados a partir do deslocamento e rotação existente entre as duas imagens, ou acumulados ao longo do trajecto. O cálculo

destes erros vai permitir estimar que distância é que o veículo pode percorrer sem ter que ser novamente referenciado.

O erro em cada ponto é dado pela diferença entre os valores exactos e os valores calculados. Por exemplo, os erros determinados entre a *imageref* e a *imagein* representadas na Figura 5.12 a) e na Figura 5.14, respectivamente, foram os seguintes:

$$|\varepsilon_{theta}| = |20 - 20.02| = 0.02^\circ,$$

$$|\varepsilon_X| = |40 - 39.99| = 0.01 \text{ pixel},$$

$$|\varepsilon_Y| = |40 - 39.99| = 0.01 \text{ pixel},$$

em que 20° e 40 pixels são os valores exactos, e os valores determinados são 20,02° e 39,99 *pixel* respectivamente para o ângulo *theta* e para os deslocamentos *X* e *Y*.

O cálculo do erro acumulado é feito somando todos os erros calculados anteriormente para todos os pontos, ou seja, $|\varepsilon_{thetaAc}| = |\sum \varepsilon_{theta}|$, $|\varepsilon_{XAc}| = |\sum \varepsilon_X|$ e $|\varepsilon_{YAc}| = |\sum \varepsilon_Y|$.

O último cálculo realizado é o da velocidade máxima *vel*. Primeiro calcula-se o espaço percorrido, convertendo pixel para *cm*, ou seja

$$esp_percX = \frac{X}{pix_cm}, \quad esp_percY = \frac{Y}{pix_cm}$$

$$esp_percXY = \sqrt{(esp_percX)^2 + (esp_percY)^2},$$

em que *esp_percX* e *esp_percY* é o espaço percorrido, respectivamente segundo o eixo dos *xx* e segundo o eixo dos *yy*, *esp_percXY* é o espaço total percorrido em *cm*, *X* e *Y* é o espaço percorrido em pixels e *pix_cm* é o número de pixels por centímetro. Como sabemos o tempo, *time*, que demorou a processar cada duas imagens, ou seja a obter o ângulo *theta* e o deslocamento *X* e *Y*, podemos determinar a velocidade máxima de circulação do veículo

$$vel = \frac{esp_percXY}{time},$$

onde *esp_percXY* é o espaço total percorrido pelo veículo em *cm*, *time* é o tempo em se-

gundos de processamento entre imagens e *vel* a velocidade do veículo em centímetros por segundo.

Findo este ciclo, *imagein* passa a ser a nova imagem de referência *imageref* a terceira imagem passa a ser *imagein*.

São repetidos os passos até aqui descritos para todas as imagens que dizem respeito a um trajecto.

Depois de ter determinado o trajecto completo, a partir das imagens, então determinam-se as outras trajectórias existentes.

A seguir à determinação de todas as trajectórias e os resultados guardados numa estrutura designada por *Res*, apresentam-se em gráficos todos os resultados.

Capítulo 6

Validação e verificação de resultados

A validação e verificação de resultados, foi feita recorrendo ao sistema de simulação atrás descrito.

Pretende-se neste capítulo fazer um estudo exaustivo do funcionamento do sistema, identificar claramente quais as suas limitações e registar informação suficiente, para auxiliar em aplicações futuras.

No sub-capítulo 6.3, faz-se uma simulação da aplicação do sistema de navegação desenvolvido.

6.1 Parametrização do sistema de navegação

A simulação serviu para testar o sistema de navegação desenvolvido e visualizar em gráficos os resultados referentes aos parâmetros principais, isto é, aqueles que permitirão a um projectista, perante objectivas consultas, determinar se o problema tem solução e o seu desenvolvimento (dimensão das imagens, espaçamento entre as imagens e gama de rotações permitida).

A dimensão das imagens depende do tempo de processamento permitido entre duas imagens e da distância que se pretende percorrer. Para resolver esta situação, é necessário saber a relação pixel por centímetro, o número médio de pixels para cada deslocamento, assim como a rotação máxima entre cada duas imagens.

A simulação foi feita em três tipos de superfície diferentes, por forma a analisar o comportamento do sistema para superfícies com características distintas umas das outras.

A primeira superfície teve como imagem base uma fotografia de um piso industrial com características aleatórias, Figura 6.1.

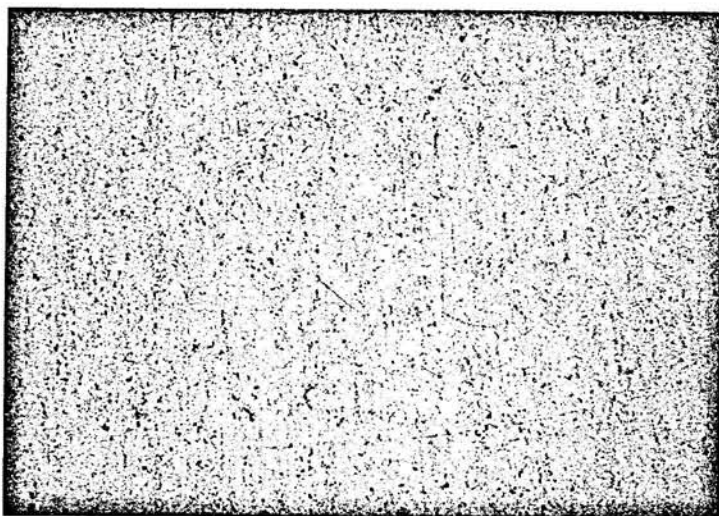


Figura 6.1 Imagem base para gerar uma superfície com característica aleatórias.

A segunda superfície tem como característica principal ser uma *paisagem*, por isso seleccionou-se como imagem base um arranjo de flores, representado na Figura 6.2. Esta imagem é retirada do Matlab e é designada por *Flowers.tif*.



Figura 6.2 Imagem base para gerar uma superfície com característica *paisagística*.

A última superfície usada para a simulação, tem características uniformes e para isso seleccionou-se uma fotografia, para imagem base, do tampo de uma cadeira, como é ilustrado na Figura 6.3.

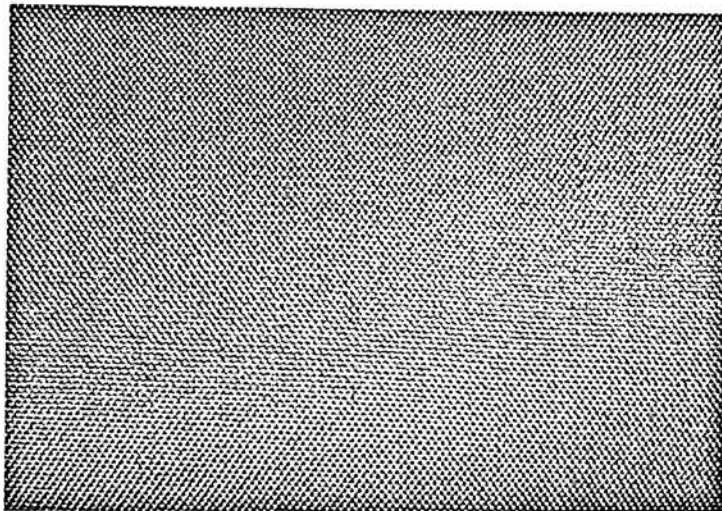


Figura 6.3 Imagem base para gerar uma superfície com característica uniforme.

Um dado necessário à avaliação do resultado e importante para a simulação, tem a ver com as características do computador onde foram feitas as simulações. Estas características são importantes, principalmente em relação ao tempo de processamento.

O computador usado é baseado num processador *x86 Family 6 Model 8 Stepping 6 GenuineIntel ~730MHz*, com 261.556kB de memória física, o sistema operativo é o *Microsoft Windows 2000 Professional* e versão *5.0.2195 Service Pack 3 Build 2195*, e por fim a ferramenta de programação é *Matlab*, versão 6.5, a funcionar em rede.

Como já foi referido foram gerados gráficos com os resultados das simulações. Os gráficos apresentados nesta dissertação são os que reflectem um comportamento médio em relação às várias simulações efectuadas para cada situação e foram subdivididos em três grupos: tempo de processamento, espaçamento máximo entre duas imagens e gamas de ângulos de rotação.

6.1.1 Tempo de processamento

Os gráficos apresentados na Figura 6.4 a), b) e c), dizem respeito ao tempo de processamento de duas imagens, dependendo da dimensão destas. As imagens foram obtidas a partir das superfícies aleatória, *paisagística* e uniforme, respectivamente.

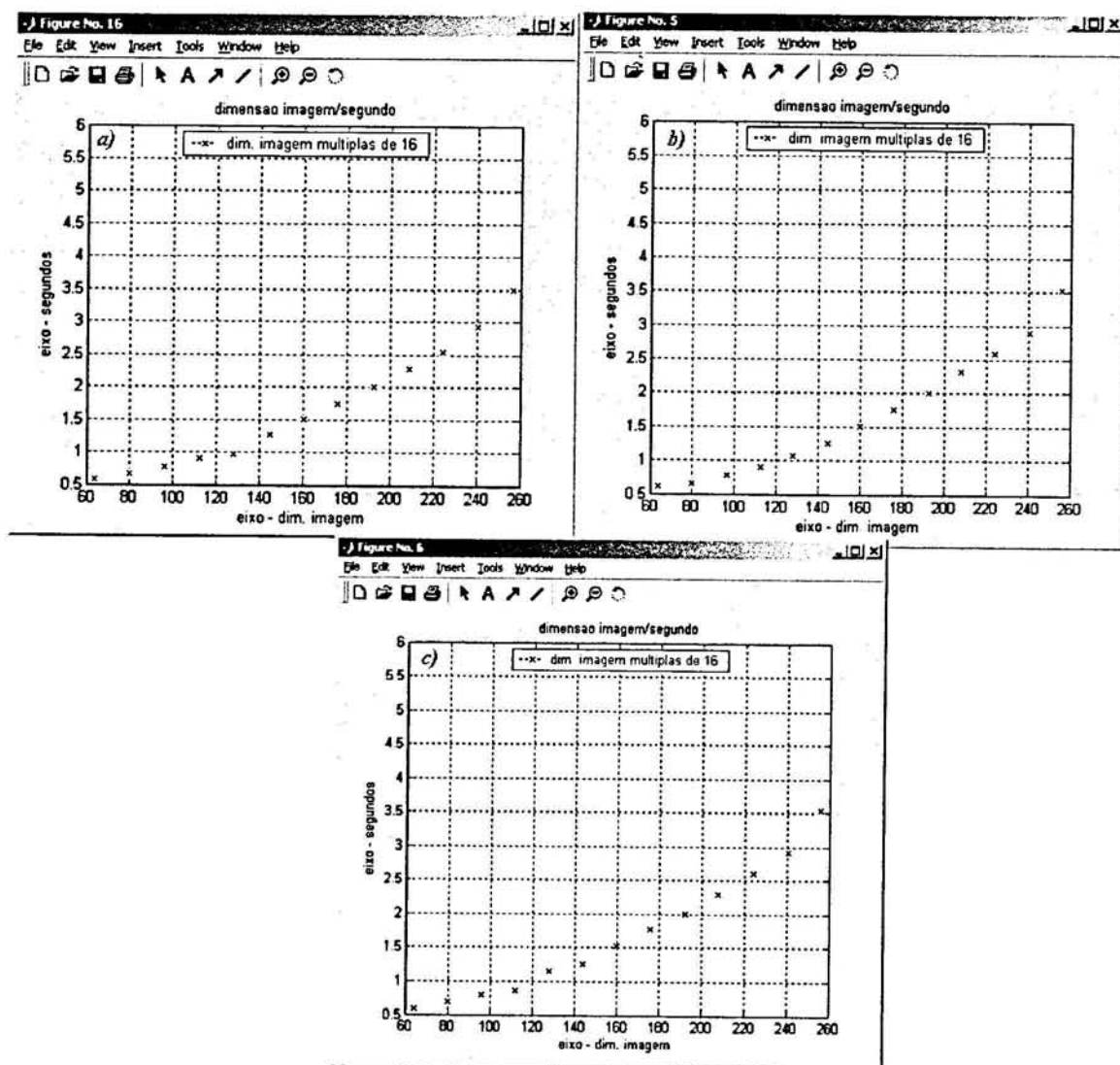


Figura 6.4 Tempo de processamento em função da dimensão das imagens: a) Superfície aleatória; b) Superfície *paisagística*; c) Superfície uniforme.

Os gráficos foram gerados para duas imagens de dimensões que começam em 64x64 e aumentam de 16 em 16 pixels, até à dimensão de 256x256, registando-se o tempo

de processamento para cada dimensão.

Este tempo de processamento é o tempo necessário para determinar a translação e a rotação entre cada par de imagens da mesma dimensão.

As dimensões das imagens seleccionadas são potências de dois, porque torna o cálculo das FFT's mais rápido, apesar do tempo de execução depender da dimensão da transformada. O tempo de processamento depende da dimensão das imagens e não das suas características nem do seu relacionamento. Podemos verificar pela análise da Figura 6.4 a), b) e c), que os gráficos para as diferentes superfícies são muito semelhantes e o tempo de processamento aumenta à medida que aumenta a dimensão das imagens.

6.1.2 Espaçamento máximo entre duas imagens

Pretende-se neste sub-capítulo determinar o espaçamento máximo entre duas imagens em diferentes situações, para as imagens de dimensão 64x64, 128x128 e 256x256.

A simulação consistiu em aumentar o espaçamento entre duas imagens, para cada superfície já apresentada, e registar o erro entre o espaçamento real e o calculado.

O espaçamento entre as imagens era aumentado com incrementos de 4, 6 e 10 pixels, respectivamente, para 64x64, 128x128 e 256x256.

Os incrementos foram aplicados em simultâneo aos dois eixos, eixo dos xx e eixo dos yy , uma vez que é a situação mais desfavorável.

Estas simulações foram feitas para ângulos de rotação de 0° , 10° e 40° entre as duas imagens.

Os gráficos serão apresentados por grupos, conforme o ângulo de rotação aplicado e serão também apresentados em pormenor algumas das zonas de maior interesse desses gráficos.

Espaçamento para $\theta=0^\circ$

Na Figura 6.5, Figura 6.6 e na Figura 6.7 estão representados os resultados da simulação, respectivamente, para as superfícies aleatória, paisagística e uniforme.

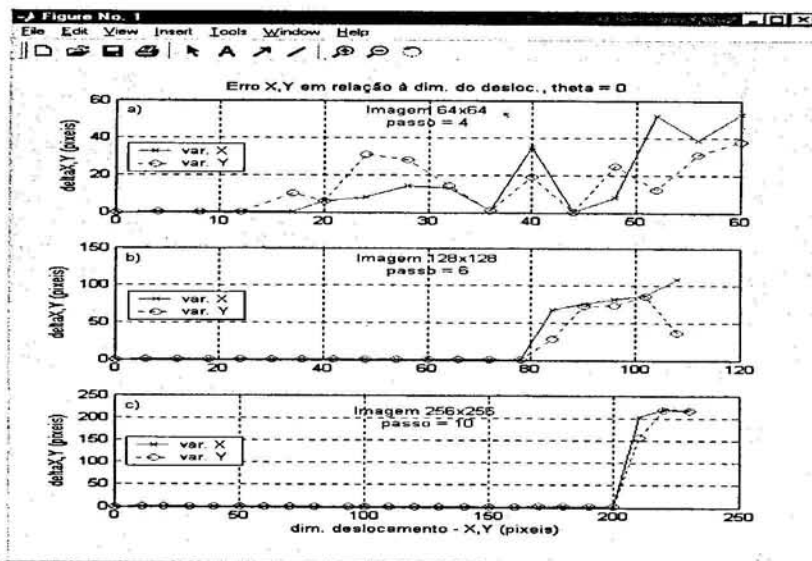


Figura 6.5 Erro de X e Y, função da dimensão da translação para $\theta=0$ e superfície aleatória.

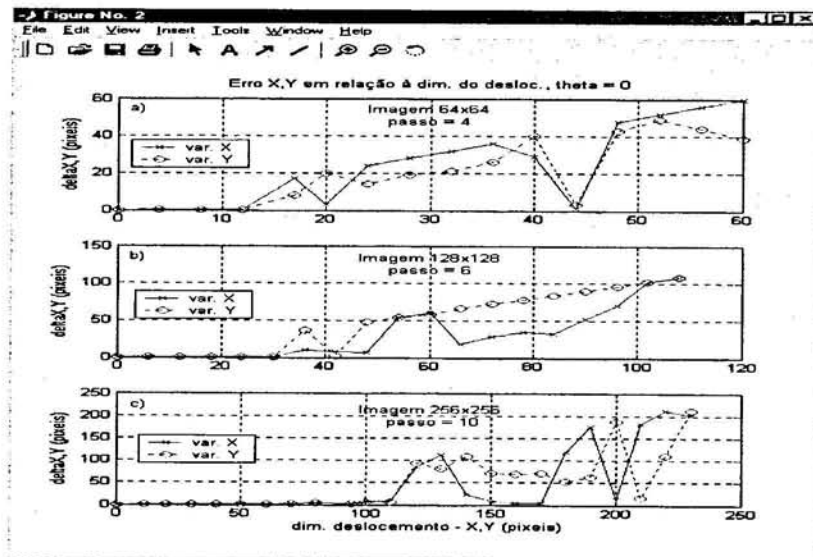


Figura 6.6 Erro de X e Y, função da dimensão da translação para $\theta=0$ e superfície paisagística.

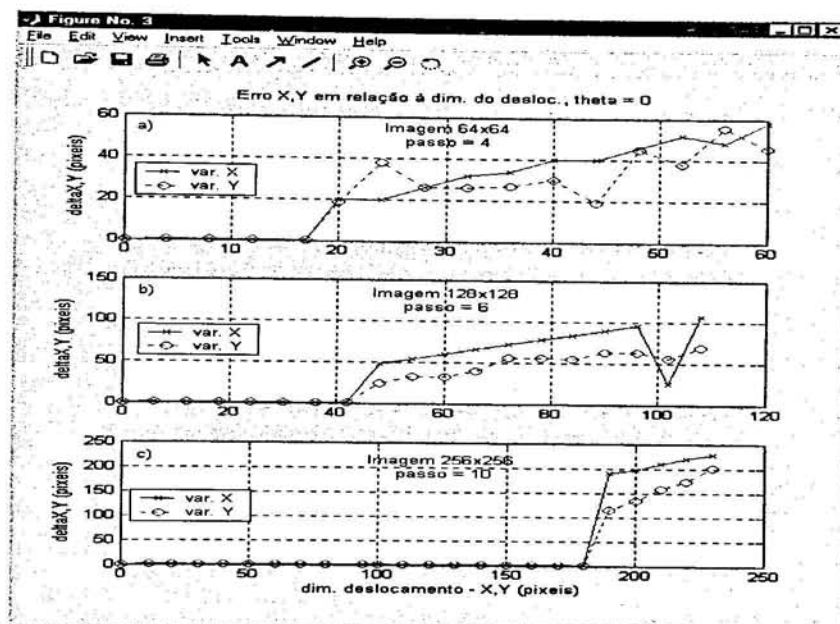


Figura 6.7 Erro de X e Y, função da dimensão da translação para $\theta=0$ e superfície uniforme.

Podemos verificar pela análise da Figura 6.5, Figura 6.6 e Figura 6.7, que a superfície com melhores resultados, que admite uma gama maior de deslocamento entre as imagens, é a superfície aleatória, Figura 6.5.

Por outro lado o gráfico da superfície uniforme, Figura 6.7, mostra que o comportamento é muito semelhante ao da superfície aleatória, Figura 6.5, mas corre-se o risco de haver *aliasing*, Figura 1.3.

Com os gráficos das figuras a seguir apresentados, pretende-se mostrar em pormenor os erros existentes na gama útil de espaçamentos, para cada um dos gráficos anteriores.

Gama útil do espaçamento, visualizado em pormenor ($\theta=0^\circ$)

Na Figura 6.8, Figura 6.9 e na Figura 6.10 pode-se ver mais em pormenor os erros na gama de translação verificando-se que o sistema não apresenta erros significantes.

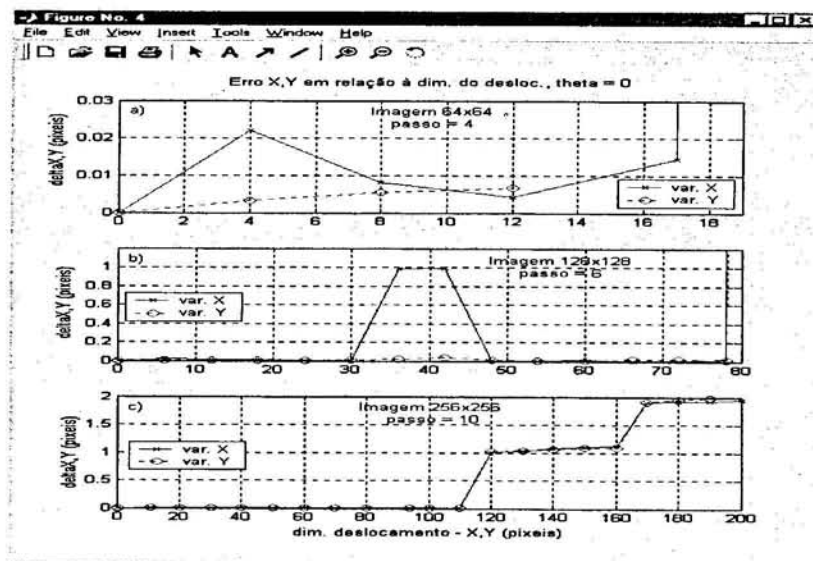


Figura 6.8 Visualização pormenorizada da Figura 6.5 (superf. aleatória).

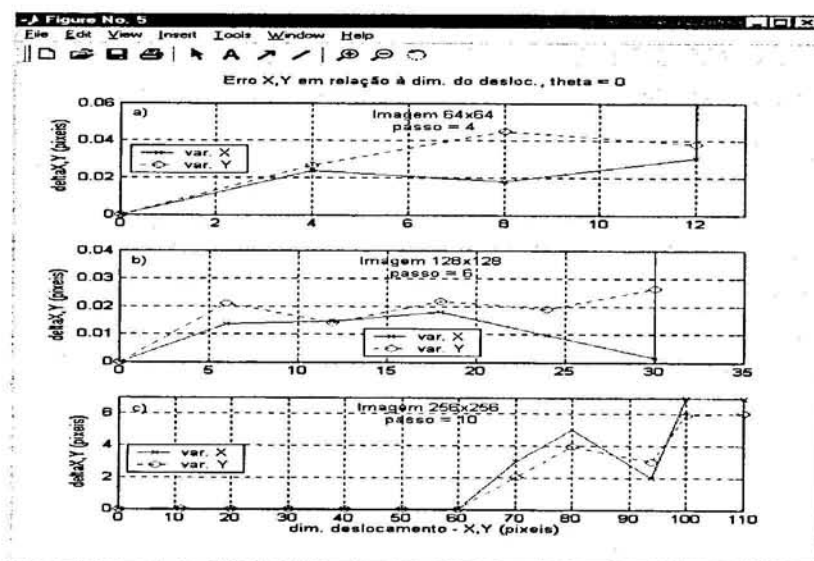


Figura 6.9 Visualização pormenorizada da Figura 6.6 (superf. paisagística).

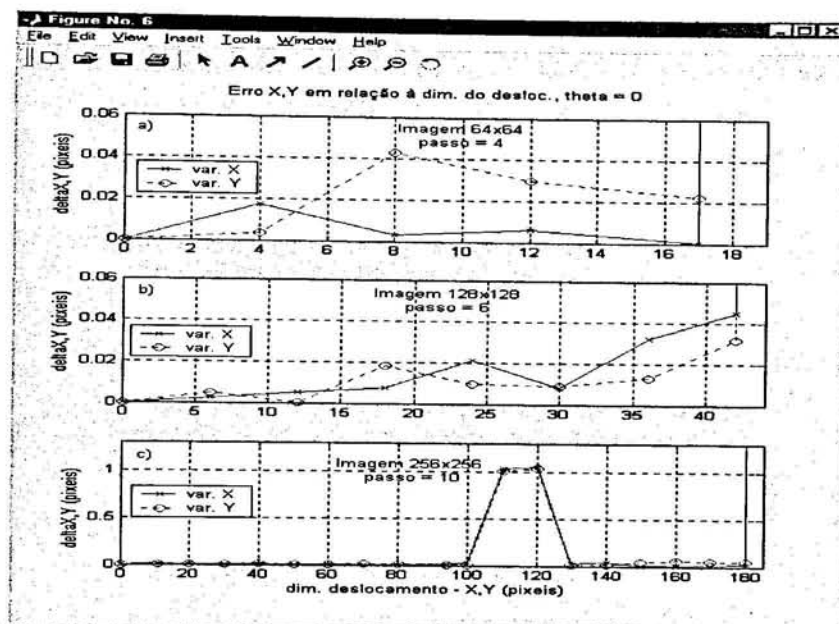


Figura 6.10 Visualização pormenorizada da Figura 6.7 (superf. uniforme).

Verifica-se pela Figura 6.8, Figura 6.9 e Figura 6.10 que a dimensão das imagens que proporciona um comportamento mais estável é 64x64 e por outro lado, 256x256 é a dimensão que tem um comportamento com desvios mais acentuados, mas aceitáveis.

A zona de erros em torno de um pixel ou superior deve ser evitada, porque poderá ser uma zona instável.

Da mesma forma que foi feita a simulação anterior, fez-se a simulação para ângulos de rotação diferentes de zero, tendo sido escolhidos os ângulos de 10° e 40°.

Espaçamento para $\theta=10^\circ$

Na Figura 6.11, Figura 6.12 e na Figura 6.13 estão representados os resultados da simulação, respectivamente, para as superfícies aleatória, *paisagística* e uniforme.

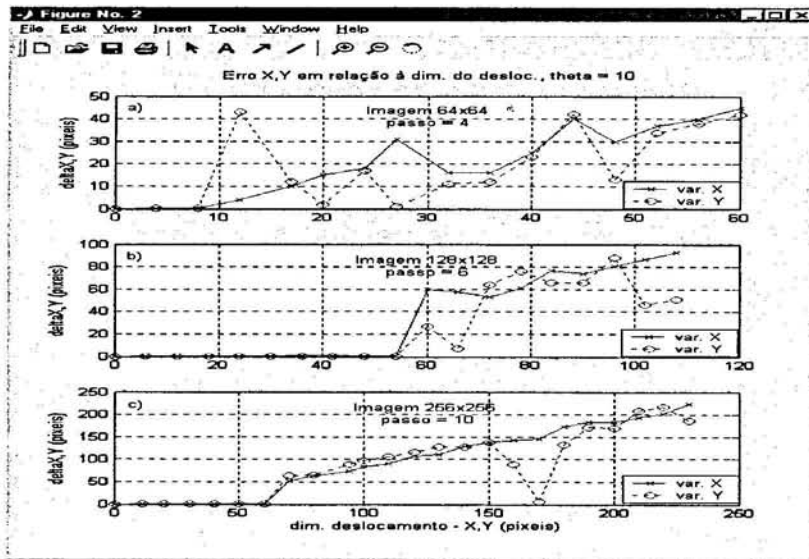


Figura 6.11 Erro de X e Y , função da dimensão da translação para $\theta=10$ e superfície aleatória.

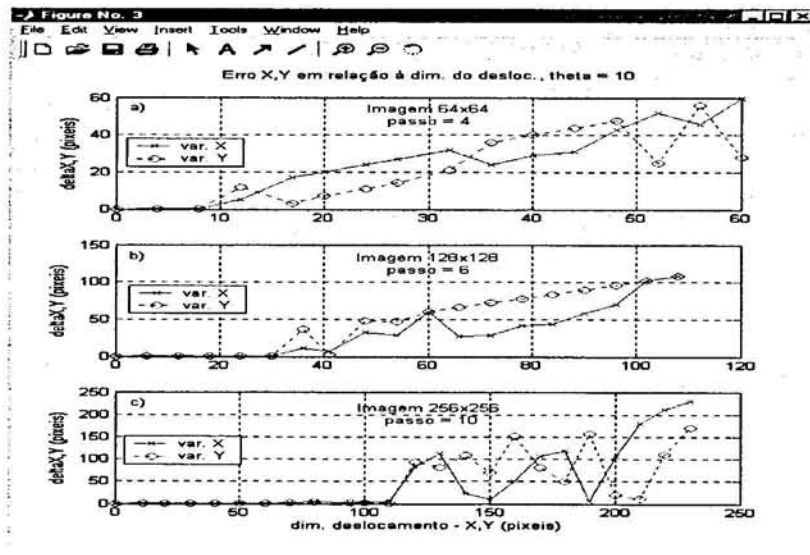


Figura 6.12 Erro de X e Y , função da dimensão da translação para $\theta=10$ e superfície *paisagística*.

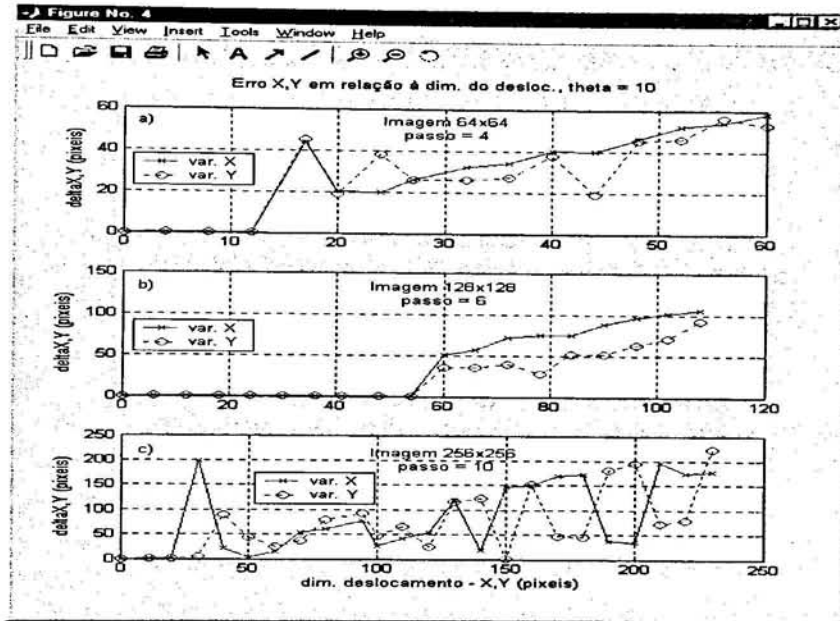


Figura 6.13 Erro de X e Y, função da dimensão da translação para $\theta=10$ e superfície uniforme.

Ao analisar as três figuras anteriores, para imagens de dimensão 64x64, pode-se ver que mantêm um comportamento estável para os três tipos de superfície.

Para as imagens de dimensão 256x256 não mostrou ter um comportamento tão estável, uma vez que na superfície uniforme admite um espaçamento bastante reduzido entre imagens.

As imagens de dimensão 128x128 mostraram ter um comportamento intermédio em relação às outras duas dimensões.

Gama útil do espaçamento, visualizado em pormenor ($\theta=10^\circ$)

Nas figuras seguintes está representado em pormenor a gama de espaçamentos, com erros admissíveis, para cada superfície.

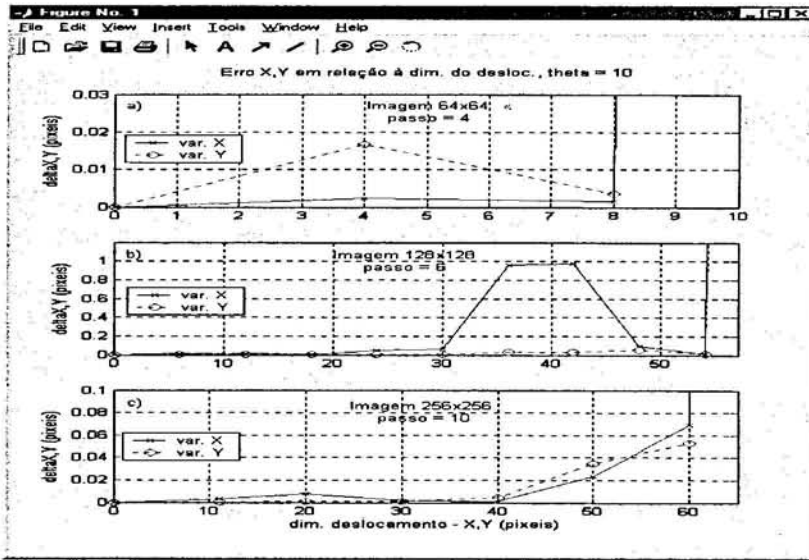


Figura 6.14 Visualização pormenorizada da Figura 6.11 (superf. aleatória).

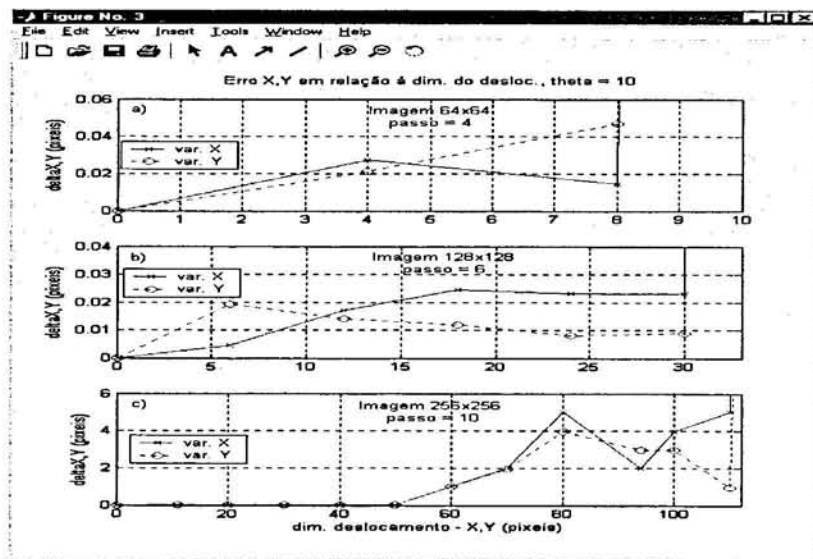


Figura 6.15 Visualização pormenorizada da Figura 6.12 (superf. paisagística).

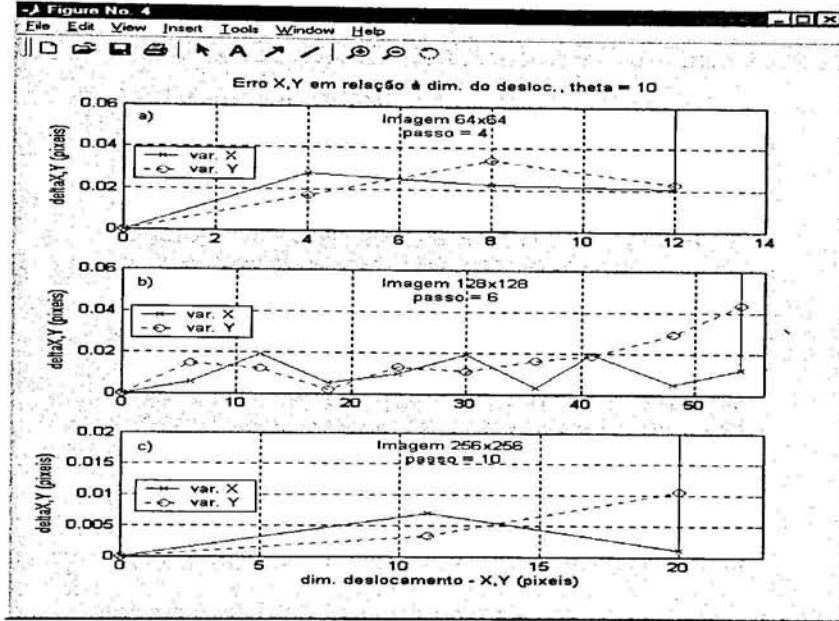


Figura 6.16 Visualização pormenorizada da Figura 6.13 (superf. uniforme).

Pode-se ver pela análise dos gráficos anteriores que na gama útil dos espaçamentos, os erros são bastante baixos à excepção das imagens de dimensão 256x256, para a superfície paisagística, estes mostram erros que podem ir até cinco pixels, para espaçamentos superiores a 50 e inferiores a 110 pixels, entre imagens.

Estas zonas de erros devem ser, tanto quanto possível, evitadas por forma a obter resultados válidos e controláveis.

Espaçamento para $\theta=40^\circ$

As três figuras seguintes apresentam o resultado das simulações anteriores, mas para um ângulo de rotação de 40° .

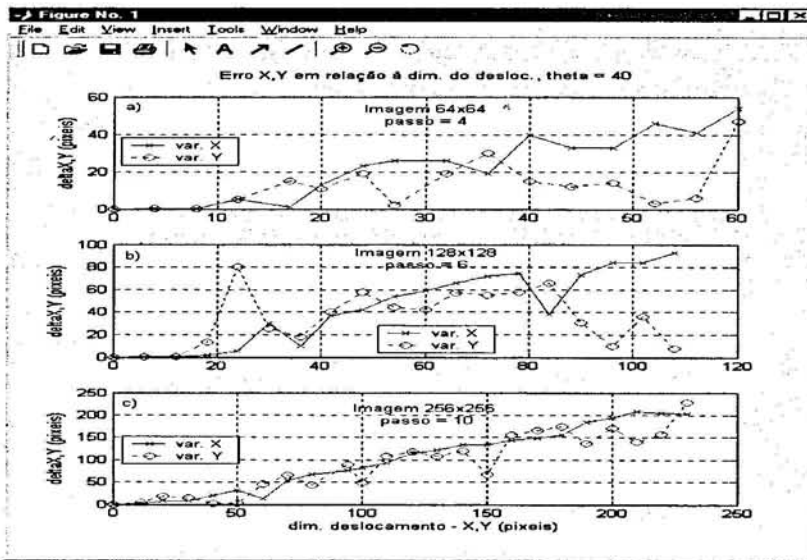


Figura 6.17 Erro de X e Y , função da dimensão da translação para $\theta=40^\circ$ e superfície aleatória.

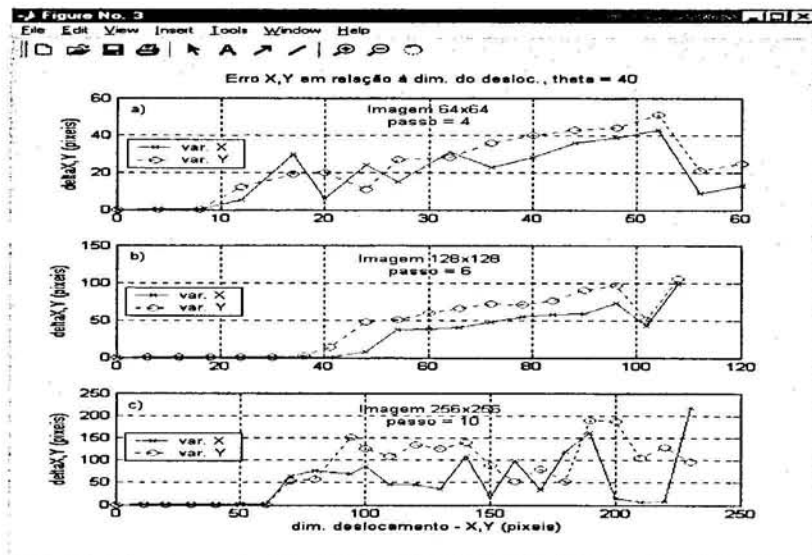


Figura 6.18 Erro de X e Y , função da dimensão da translação para $\theta=40^\circ$ e superfície paisagística.

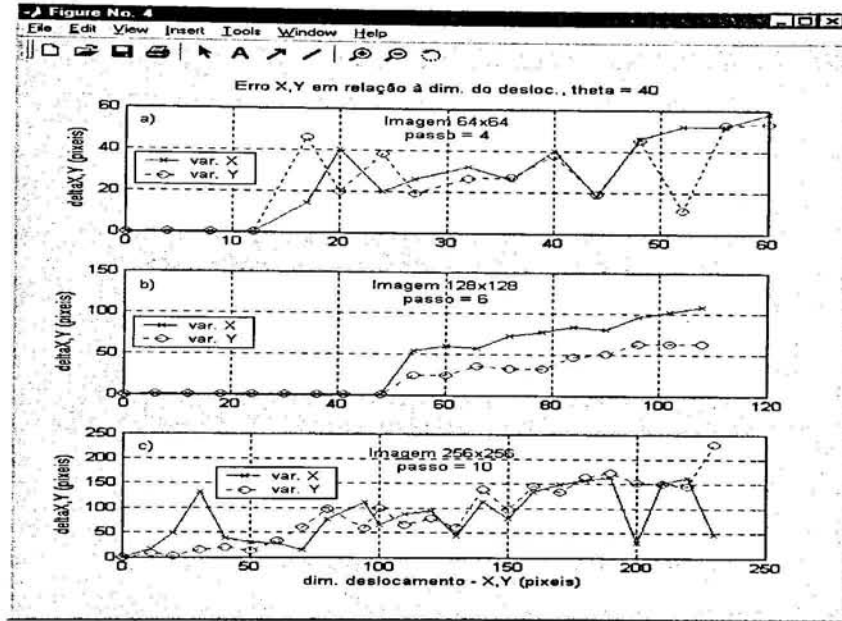


Figura 6.19 Erro de X e Y, função da dimensão da translação para $\theta=40$ e superfície uniforme.

Da análise das figuras anteriores conclui-se que as imagens de dimensão 64x64 foram as que tiveram um comportamento mais estável para os três tipos de superfície, ou seja, mantiveram a gama de deslocamentos admissíveis nas três superfícies.

As imagens de dimensão 256x256 foram as que variaram mais a sua gama de deslocamentos admissíveis para as três superfícies e no caso da superfície do tipo uniforme não é admitido nenhum deslocamento entre as imagens.

Quanto às imagens de dimensão 128x128 também houve alteração na sua gama de deslocamentos, mas não foram tão significativos como para as de dimensão 256x256, tendo registado a sua gama mais baixa na superfície do tipo aleatório.

Gama útil do espaçamento, visualizado em pormenor ($\theta=40^\circ$)

Nas figuras seguintes está representado em pormenor a gama de espaçamentos, com erros admissíveis, para cada superfície.

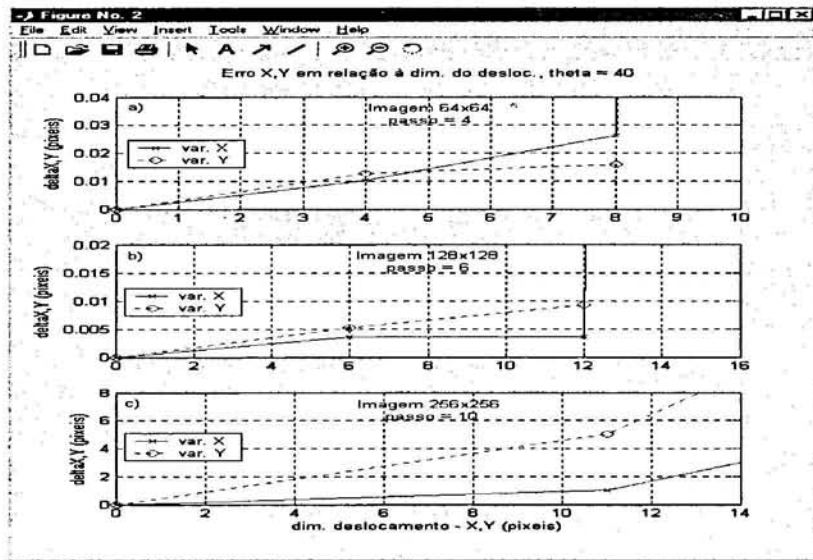


Figura 6.20 Visualização pormenorizada da Figura 6.17 (superf. aleatória).

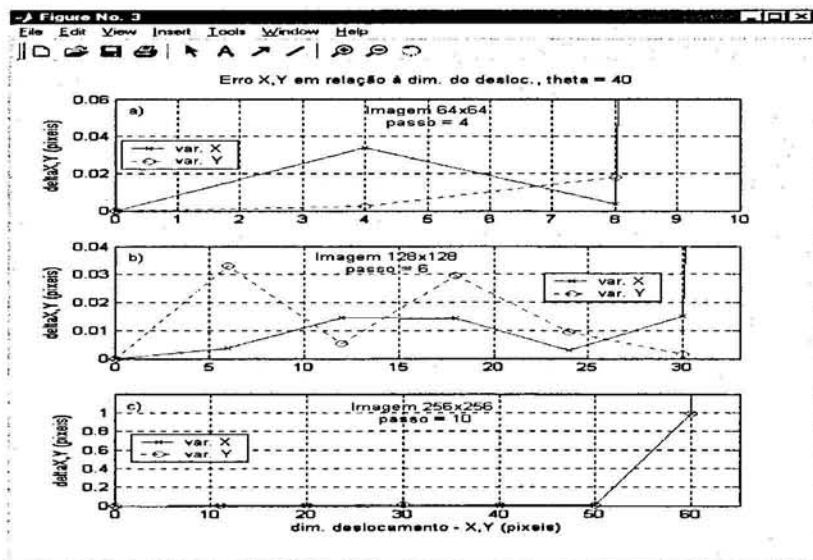


Figura 6.21 Visualização pormenorizada da Figura 6.18 (superf. paisagística).

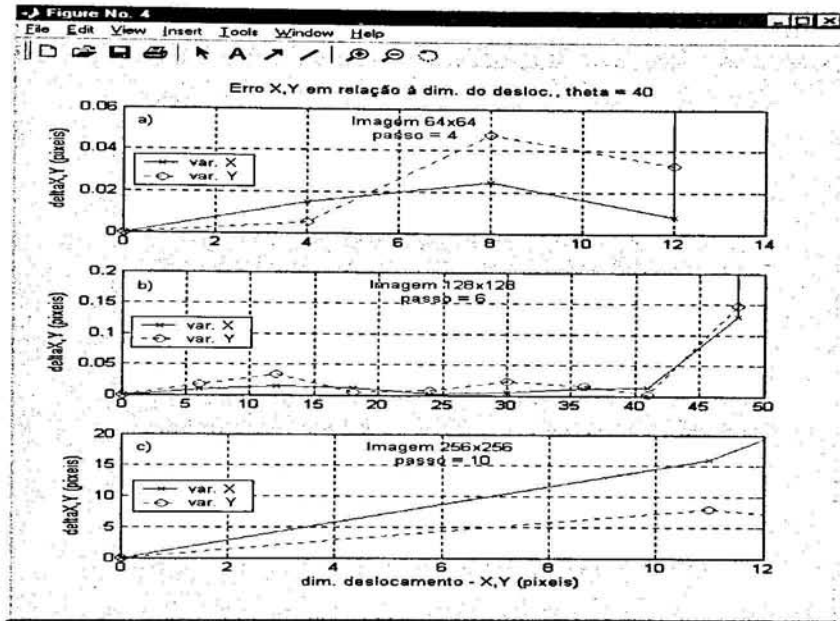


Figura 6.22 Visualização pormenorizada da Figura 6.19 (superf. uniforme).

Ao analisar a Figura 6.20, Figura 6.21 e a Figura 6.22, verifica-se que as imagens de dimensão 256x256 apresentam erros elevados para qualquer espaçamento, à excepção da situação de superfície *paisagística*, em que existe uma gama de espaçamentos com erros reduzidos.

Da mesma análise para as outras duas dimensões das imagens, verifica-se que os erros são pequenos, no entanto a gama de espaçamento para as imagens de dimensão de 128x128, para a superfície aleatória, é bastante inferior quando comparada com as outras duas superfícies.

Resumindo, em relação à translação, verifica-se que o sistema desenvolvido comporta-se melhor, de uma forma mais constante, para uma superfície com características *paisagísticas*, porque é uma imagem muito bem definida e com padrões distintos, proporcionando imagens consecutivas com as diferenças bem demarcadas. É com este tipo de superfície que o sistema tem um comportamento melhor na presença de rotação, para as imagens de dimensão 128x128 e 256x256.

A dimensão da imagem que se comporta melhor, de uma forma geral, é a de 64x64, porque manteve uma gama de deslocamento mais ou menos constante, quer para as

diferentes superfícies, quer com rotações aplicadas às imagens. Enquanto a de 256x256 foi a que apresentou piores resultados, praticamente só apresentando boas prestações quando as imagens não estão sujeitas, entre elas, a uma rotação. As imagens de 128x128 encontram-se a meio termo, ou seja, para rotações pequenas permite ter deslocações entre imagens relativamente grandes e vai reduzindo progressivamente, com o aumentar do ângulo de rotação.

A simulação subsequente é semelhante à que foi apresentada até aqui, só que em vez de alterar o espaçamento, altera-se a rotação e verifica-se o comportamento do sistema para as três superfícies já referidas.

6.1.3 Gama de rotações entre duas imagens

Neste sub-capítulo verifica-se qual ou quais as gamas de rotação, entre duas imagens, em que o sistema pode funcionar.

A verificação será feita para as três dimensões de imagens, 64x64, 128x128 e 256x256, e para as três superfícies aleatória, *paisagística* e uniforme.

O aumento da rotação, entre as duas imagens, na simulação foi feito com incrementos de 10° e verificou-se, registando, o erro entre o ângulo de rotação real e o calculado.

As simulações foram feitas para um deslocamento em ambos os eixos, de zero pixel, a seguir de 5, 10 e 20 pixels, respectivamente, para as imagens de dimensão 64x64, 128x128 e 256x256. Por fim aumentou-se o deslocamento para 10, 20 e 40 para as respectivas dimensões das imagens.

Sempre que necessário, mostra-se em pormenor as gamas de rotações com erros admissíveis, para cada situação.

Rotação para $X, Y = 0$

A Figura 6.23, Figura 6.24 e Figura 6.25 têm representados os resultados da simulação, respectivamente, para as superfícies aleatória, *paisagística* e uniforme.

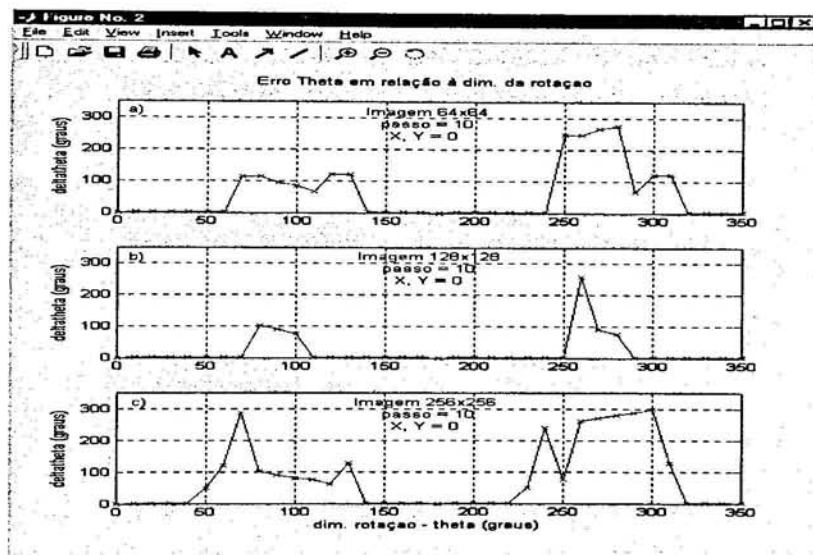


Figura 6.23 Erro de *theta*, função do ângulo de rotação entre as imagens de uma superfície aleatória para $X, Y = 0$.

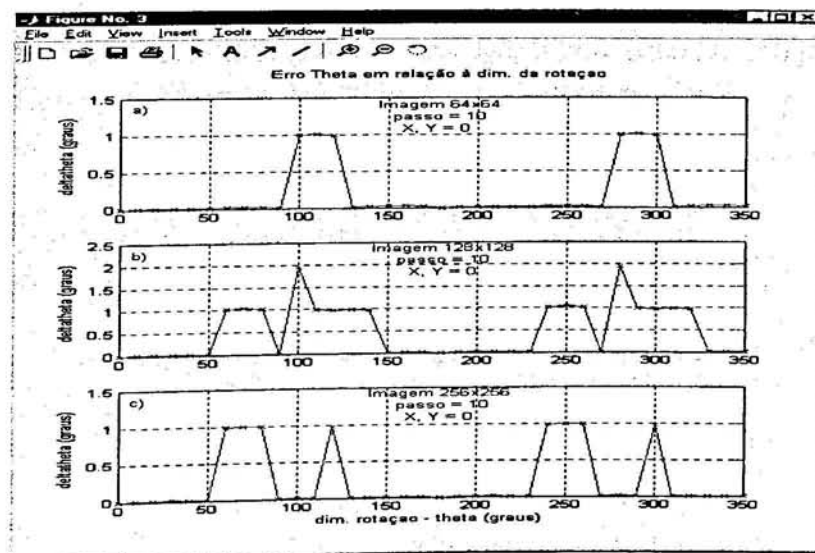


Figura 6.24 Erro de *theta*, função do ângulo de rotação entre as imagens de uma superfície *paisagística* para $X, Y = 0$.

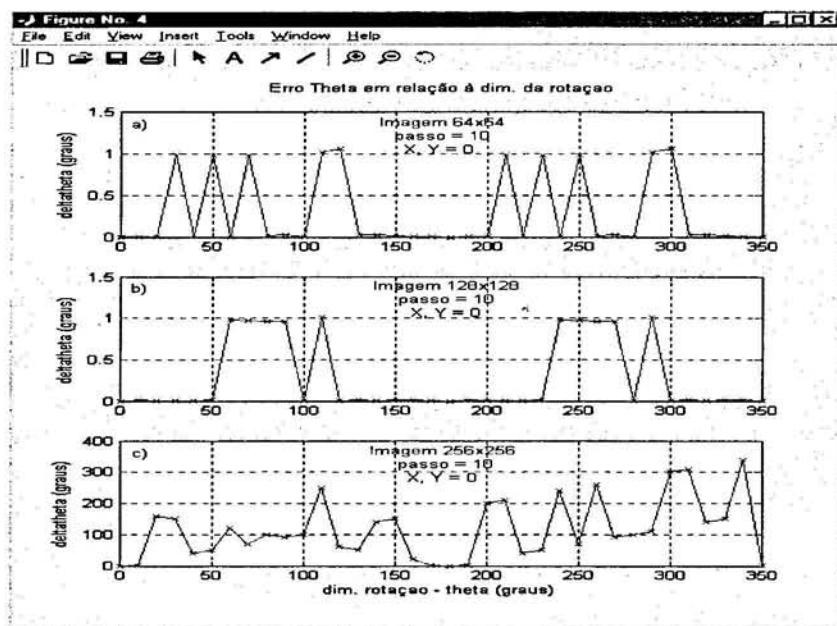


Figura 6.25 Erro de θ , função do ângulo de rotação entre as imagens de uma superfície uniforme para $X, Y = 0$.

Pode-se verificar pela Figura 6.23, que existem duas gamas de ângulos de rotação em que não é possível determinar correctamente a rotação entre as imagens. Essa gama situa-se em torno do ângulo 90° e do seu simétrico 270° . As gamas dos ângulos de rotação referidos são mais reduzidas para as imagens de dimensão 128×128 , por outro lado foram superiores para as imagens de dimensão 256×256 .

Da análise da Figura 6.24 verifica-se que o sistema desenvolvido determina a rotação para toda a gama dos 360° , apesar de apresentar variações de θ relativamente elevadas para os ângulos de rotação em torno dos 90° e dos 270° . O valor máximo do erro determinado, foi para os ângulos de 100° e de 280° sendo este valor de 2° , Figura 6.24 b).

Na Figura 6.25, superfície uniforme, a situação é idêntica à da superfície *paisagística*, em que para as imagens de dimensão de 64×64 e de 128×128 , respectivamente Figura 6.25 a) e b), o erro é aceitável para todos os ângulos de 0° a 360° , mas para a imagem de dimensão 256×256 , Figura 6.25 c), praticamente não se consegue determinar nenhum ângulo.

Gama útil da rotação, visualizada em pormenor ($X, Y = 0$)

Na Figura 6.26 estão representadas em pormenor as gamas de rotação com erros aceitáveis para a superfície aleatória.

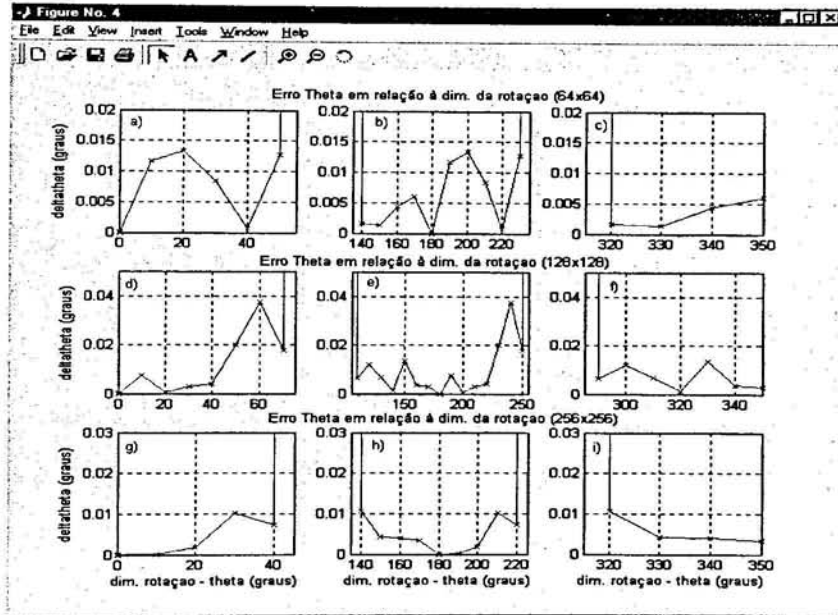


Figura 6.26 Visualização pormenorizada da Figura 6.23 (superf. aleatória).

Da análise da Figura 6.26, conclui-se que os erros, na gama dos ângulos de rotação aceitáveis, são bastante baixos, aumentando abruptamente para ângulos fora destas gamas.

Em relação às superfícies *paisagística* e *uniforme*, não são analisadas em pormenor, visto que os resultados obtidos na Figura 6.24 e na Figura 6.25 já estão pormenorizados, uma vez que o erro máximo obtido foi de 1°.

Rotação para $X, Y = 5, 10$ e 20

As figuras seguintes apresentam os resultados da simulação anterior, para as imagens deslocadas entre si, de 5, 10 e 20 para 64x64, 128x128 e 256x256.

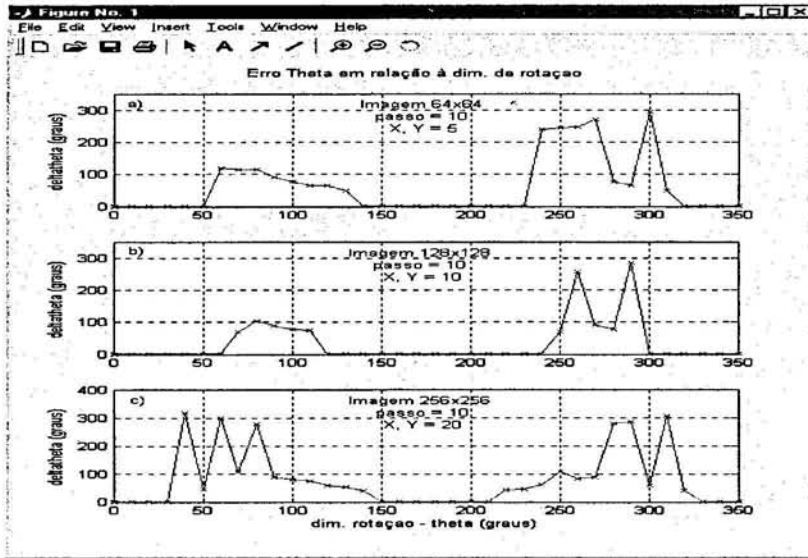


Figura 6.27 Erro de θ , função do ângulo de rotação entre as imagens consecutivas de uma superfície aleatória para $X, Y = 5, 10$ e 20 .

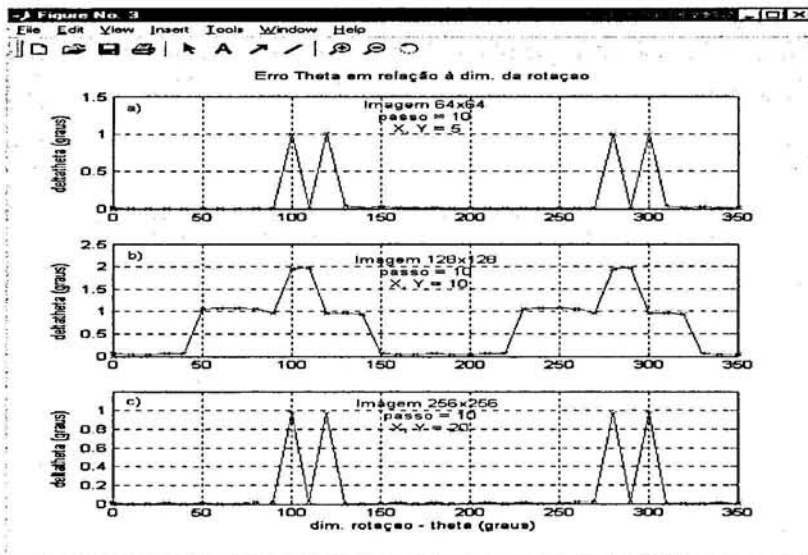


Figura 6.28 Erro de θ , função do ângulo de rotação entre as imagens consecutivas de uma superfície paisagística para $X, Y = 5, 10$ e 20 .

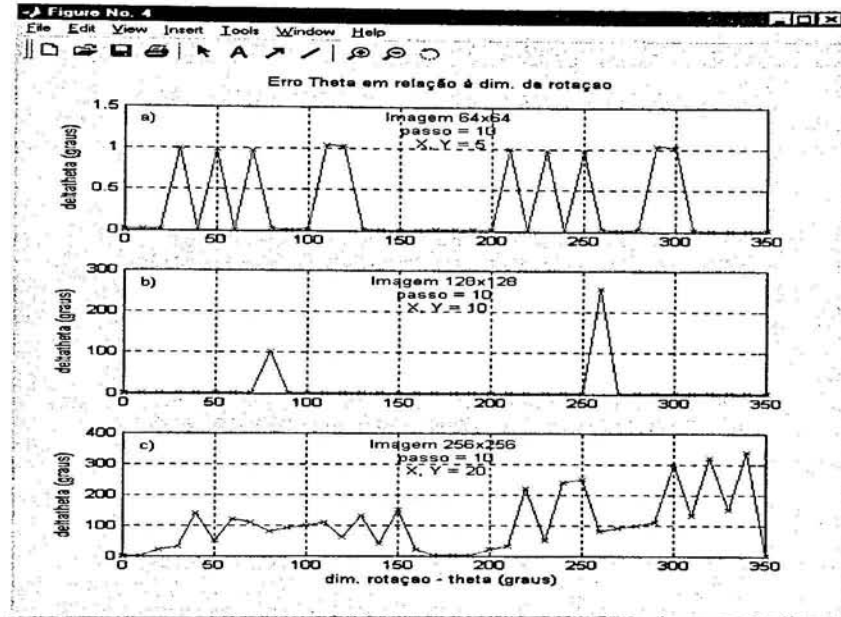


Figura 6.29 Erro de θ , função do ângulo de rotação entre as imagens consecutivas de uma superfície uniforme para $X, Y = 5, 10$ e 20 .

A Figura 6.27, superfície aleatória, mostra que existem duas gamas de ângulos de rotação que não podem ser usados, estas são em torno de 90° e 270° , a dimensão das imagens que apresentaram estas gamas mais pequenas foram as de 128×128 e as que apresentaram uma gama maior foram as de 256×256 .

Na Figura 6.28, superfície *paisagística*, vê-se que pode ser usada em todos os ângulos, no entanto tem que se ter cuidado em alguns desses ângulos de rotação, uma vez que apresentaram erros elevados, podendo ir até 2° .

A Figura 6.29, superfície uniforme, mostra que para as imagens de dimensão 64×64 pode ser usado também em todos os ângulos, apesar de apresentar alguns ângulos com erro na ordem de 1° . Para as imagens de 128×128 existem duas pequenas gamas de ângulos em que não pode ser usado, porque tem erros muito elevados. Para as imagens de dimensão 256×256 não permite nenhuma rotação entre as imagens.

Gama útil da rotação, visualizada em pormenor (X, Y = 5, 10 e 20)

A gama de ângulos de rotação com erros aceitáveis, para a superfície aleatória, pode ser vista mais em pormenor na Figura 6.30, assim como para a superfície uniforme e para as imagens de dimensão 128x128, pode ser vista na Figura 6.31.

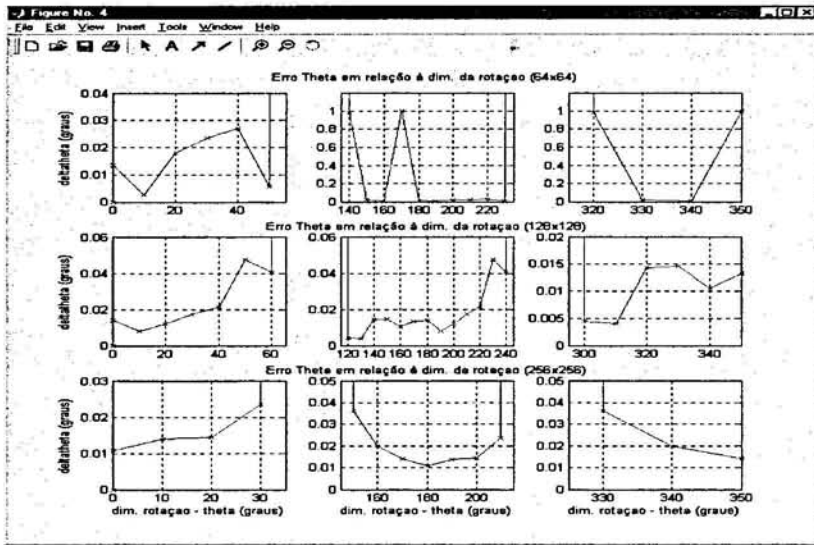


Figura 6.30 Visualização pormenorizada da Figura 6.27 (superf. aleatória).

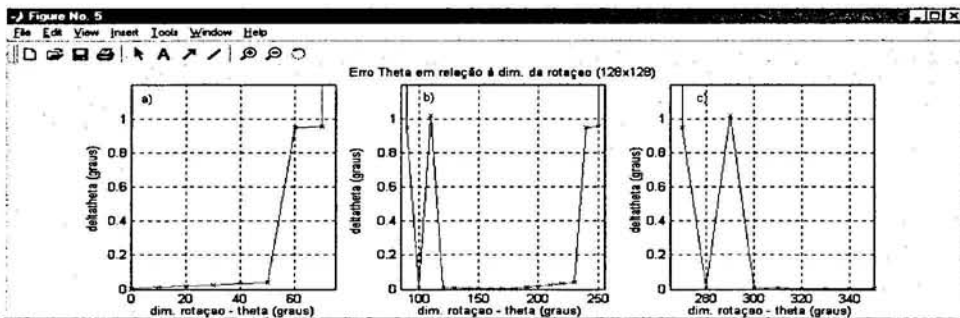


Figura 6.31 Visualização pormenorizada da Figura 6.29 b) (superf. uniforme).

O erro dos ângulos de rotação para a superfície aleatória, Figura 6.30, são pequenos, à exceção de alguns que têm erros até 1° para as imagens de dimensão 64x64.

Na Figura 6.31 só está representado para as imagens de dimensão 128x128, porque para 64x64 pode ser visto na Figura 6.29 e as de 256x256 não permitem ter nenhuma rotação.

Rotação para $X, Y = 10, 20$ e 40

As figuras seguintes apresentam os resultados das simulações anteriores, para as imagens deslocadas entre si, de 10, 20 e 40 para 64x64, 128x128 e 256x256.

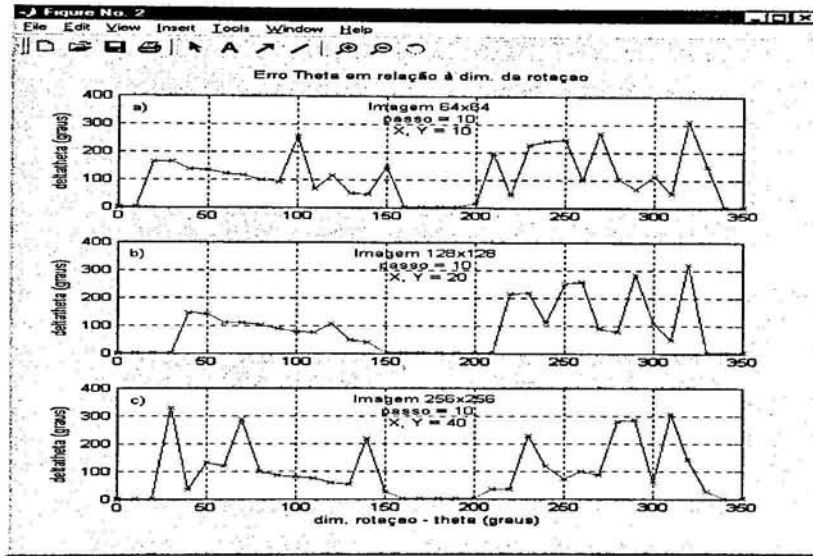


Figura 6.32 Erro de *theta*, função do ângulo de rotação entre as imagens consecutivas de uma superfície aleatória para $X, Y = 10, 20$ e 40 .

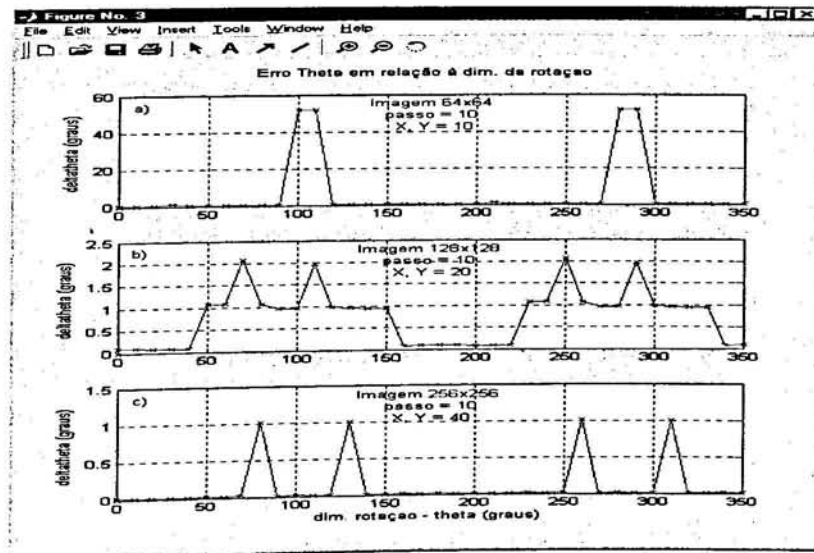


Figura 6.33 Erro de *theta*, função do ângulo de rotação entre as imagens consecutivas de uma superfície paisagística para $X, Y = 10, 20$ e 40 .

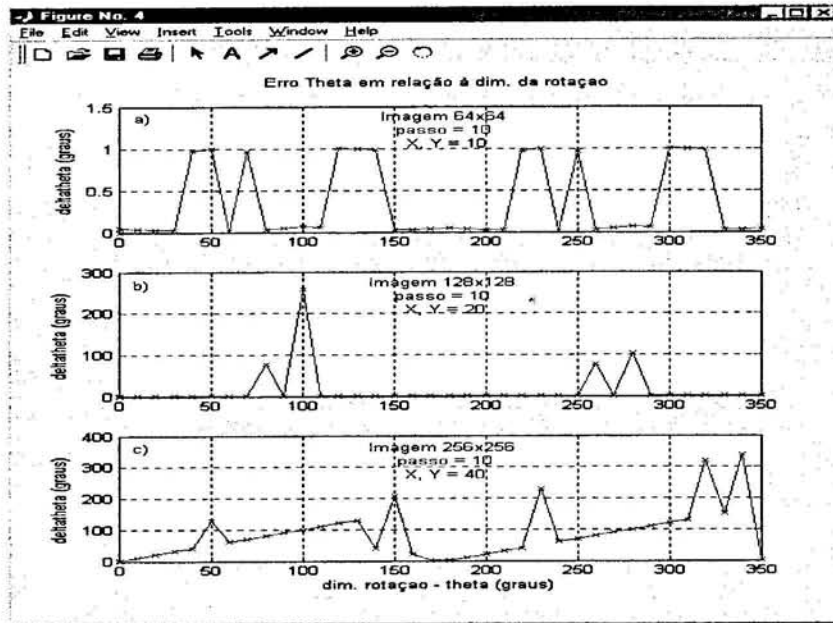


Figura 6.34 Erro de θ , função do ângulo de rotação entre as imagens consecutivas de uma superfície uniforme para $X, Y = 10, 20$ e 40 .

A Figura 6.32, superfície aleatória, mostra que as gamas de rotações com erros aceitáveis são muito reduzidas, ou seja, só funciona se os ângulos de rotação forem pequenos.

A superfície *paisagística*, Figura 6.33, em contrapartida, permite ter rotações entre imagens de 0° a 360° , mas para as imagens de dimensão 128×128 e 256×256 . Para as de 64×64 existem duas pequenas gamas de ângulos de rotação em que não é possível a determinação destes.

Por outro lado para a superfície uniforme, Figura 6.34, o sistema determinou todos os ângulos de rotação para as imagens de dimensão 64×64 , enquanto que para as de 256×256 nenhum ângulo foi correctamente determinado. Para as imagens de dimensão 128×128 houve uma pequena gama de ângulos em que não foi possível detectá-los.

Gama útil da rotação, visualizada em pormenor ($X, Y = 10, 20$ e 40)

A Figura 6.35 tem representado em pormenor as gamas de ângulos, com erros admissíveis, para a superfície aleatória, indicados na Figura 6.32.

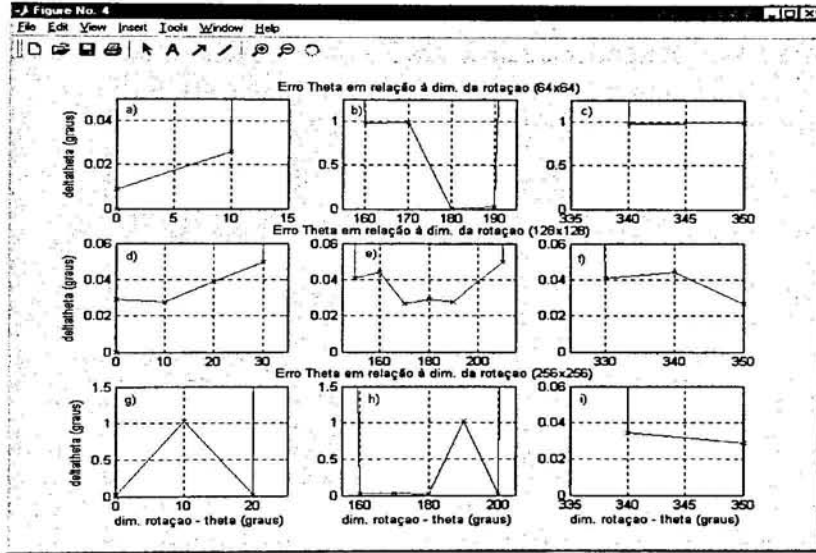


Figura 6.35 Visualização pormenorizada da Figura 6.32 (superf. aleatória).

As gamas de ângulos, com erros admissíveis, para a superfície *paisagística*, para as imagens de dimensão 64x64 estão representadas pormenorizadamente na Figura 6.36, para as outras dimensões podem ser vistas na Figura 6.33, uma vez que o erro máximo não ultrapassa os 2°.

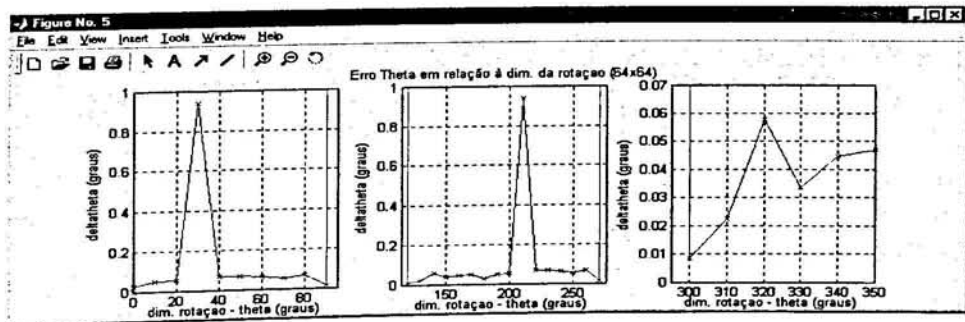


Figura 6.36 Visualização pormenorizada da Figura 6.33 (superf. paisagística).

No caso da superfície uniforme, Figura 6.34, pode-se ver a gama de ângulos, com erros admissíveis, em pormenor na Figura 6.37.

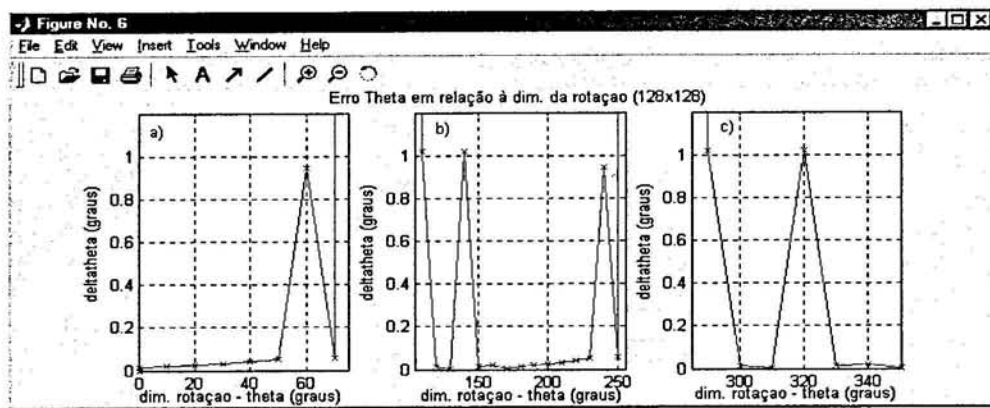


Figura 6.37 Visualização pormenorizada da Figura 6.34 (superf. uniforme).

Das três figuras, Figura 6.35, Figura 6.36 e Figura 6.37, tem que se ter em atenção alguns ângulos, uma vez que estes apresentaram erros que podem ir até 1° .

Em suma, a superfície aleatória é a que provoca maiores dificuldades ao sistema para a detecção da rotação. Nesta superfície os melhores resultados são conseguidos com imagens de dimensão 128x128, sendo a que apresentou uma gama superior de ângulos detectáveis pelo sistema, para esta superfície.

A superfície *paisagística* é a que permite melhores resultados, certamente porque entre duas imagens consecutivas, o que é diferente, é mesmo diferente, não existindo indefinição e isso reflecte-se nos resultados obtidos. Apesar do ângulo ser detectável de 0° a 360° em todas as situações, excepto nas imagens de dimensão 64x64, Figura 6.33 a), tem de se ter atenção ao facto de se ter encontrado erros de até 2° , para alguns ângulos das imagens de dimensão 128x128.

Na superfície uniforme, não é aconselhável utilizar imagens de 256x256, porque praticamente o sistema só detecta ângulos de rotação muito pequenos associados a deslocamentos pequenos. A dimensão que apresenta melhores resultados nesta superfície é a de 64x64, mas tem erros que podem ir até 1° .

6.1.4 Quadros resumo

Apresentam-se a seguir quadros resumo com os parâmetros principais para projecto.

Esses parâmetros são

Δd – dimensão do deslocamento entre duas imagens consecutivas em pixels

$\Delta \varepsilon$ – erro associado ao deslocamento Δd em pixels

Δt – tempo de processamento entre duas imagens em segundos.

Os quadros resumos que vão ser apresentados, são para ângulos de rotação $\theta = 0^\circ, 10^\circ$ e 40° , respectivamente.

Em Δd tomou-se para cada caso o valor próximo do maior, em que não acarreta erros $\Delta \varepsilon$ excessivos, e que sirva para os três tipos de superfícies aleatória, *paisagística* e uniforme.

Quer os valores de Δd quer os de $\Delta \varepsilon$ serão validados, para $\theta = 0^\circ$ no sub-capítulo 5.4.

Os resultados obtidos permitem-nos relacionar com relativa facilidade os parâmetros a utilizar numa aplicação prática.

Resultados obtidos para uma rotação $\theta=0^\circ$.

Dimensão da Imagem	Δd (px)	Δt (s)	$\Delta \varepsilon$ (px)
64x64	8	0.6	0.007
128x128	25	1.0	0.004
256x256	55	3.5	0.005

Figura 6.38 Quadro resumo de alguns resultados da simulação efectuada para $\theta = 0^\circ$.

Resultados obtidos para uma rotação $theta=10^\circ$.

Dimensão da Imagem	Δd (px)	Δt (s)	$\Delta \epsilon$ (px)
64x64	8	0.6	0.004
128x128	25	1.0	0.005
256x256	20	3.5	0.01

Figura 6.39 Quadro resumo de alguns resultados da simulação efectuada para $theta = 10^\circ$.

Resultados obtidos para uma rotação $theta=40^\circ$.

Dimensão da Imagem	Δd (px)	Δt (s)	$\Delta \epsilon$ (px)
64x64	8	0.6	0.025
128x128	12	1.0	0.01
256x256	X	3.5	X

Figura 6.40 Quadro resumo de alguns resultados da simulação efectuada para $theta = 40^\circ$.

6.2 Distância máxima entre referências

Começamos por recordar alguns dos resultados obtidos atrás, Figura 6.38, para uma rotação $theta=0^\circ$, que é o caso que vamos verificar.

Dimensão da Imagem	Δd (px)	Δt (s)	$\Delta \epsilon$ (px)
64x64	8	0.6	0.007
128x128	25	1.0	0.004
256x256	55	3.5	0.005

Figura 6.41 Quadro resumo de alguns resultados da simulação efectuada.

Normalmente numa situação de projecto, os dados do problema serão

D – distância em cm a percorrer,

$e_{m\acute{a}x}$ – erro máximo admissível em cm,

vel_{min} – velocidade mínima de circulação do veículo em cm/s

e pretende-se determinar a resolução da imagem, 64x64, 128x128 ou 256x256, e a relação imagem/objecto (posição da câmara), que se traduz em pix_cm (distância ao solo).

Os dados utilizados são os que se obtiveram atrás no quadro da Figura 6.41 e os dados recolhidos referem-se a uma relação de 16 pixels/cm que foi a utilizada neste estudo.

Outras relações de dimensão imagem/objecto traduzem-se em dois efeitos:

- alteração do tamanho do pixel, com influência directa e controlável sobre o erro máximo admitido (o mesmo erro em pixel corresponde a um erro diferente em cm);
- alteração da frequência espacial de amostragem, com problemas de *aliasing* associados (não tratado neste trabalho).

Calculando em primeiro lugar quantas imagens poderão ser utilizadas sem ultrapassar o erro máximo admitido $e_{máx}$ tem-se

$$\text{Número imagens} = \frac{e_{máx}}{e}$$

em que e é o erro em cm entre duas imagens, ou seja

$$e = \frac{\Delta\varepsilon}{pix_cm}$$

donde

$$\text{Número imagens} = \frac{e_{máx}}{\Delta\varepsilon} pix_cm.$$

A distância máxima que é possível percorrer depende do valor d entre duas imagens

$$d = \frac{\Delta d}{pix_cm}$$

donde

$$D = \text{número imagem} \cdot d = \frac{\Delta d}{\Delta \varepsilon} e_{\max}$$

isto é, D cresce linearmente com o erro máximo admitido, como se mostra na Figura 6.42, em que Δd e $\Delta \varepsilon$ foram utilizados os parâmetros do quadro da Figura 6.41.

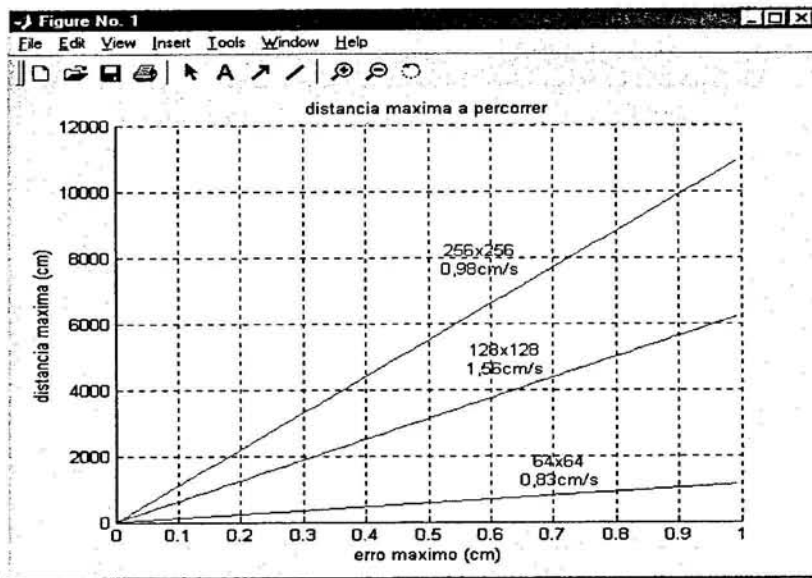


Figura 6.42 Distância máxima a percorrer em cm, para um determinado erro máximo, dependendo da dimensão das imagens.

A velocidade máxima do veículo é

$$V_{\max} = \frac{\Delta d}{\Delta t} = \frac{1}{\text{pix_cm}} \cdot \frac{\Delta d}{\Delta t}$$

que é função da relação dimensional imagem/objecto.

Na Figura 6.42 indicam-se as velocidades máximas para o caso de 16 pixels/cm. Outras relações conduzem a velocidades máximas diferentes, superiores quando o número de pixels/cm diminui, com a limitação decorrente da eventual introdução de *aliasing*.

Os tempos utilizados referem-se aos tempos Matlab, e podem ser muito melhorados com programação adequada.

6.3 Simulação

Na simulação usou-se a superfície aleatória, já apresentada, como base para definir a trajectória a calcular a partir das imagens adquiridas da superfície, tendo em consideração os parâmetros apresentados no quadro da Figura 6.41.

A distância D a percorrer foi de 100cm e simulou-se para as três dimensões 64x64, 128x128 e 256x256.

Com o conjunto de imagens adquiridas, usou-se o programa de simulação desenvolvido e determinaram-se os gráficos de deslocamento da Figura 6.43, assim como os seguintes.

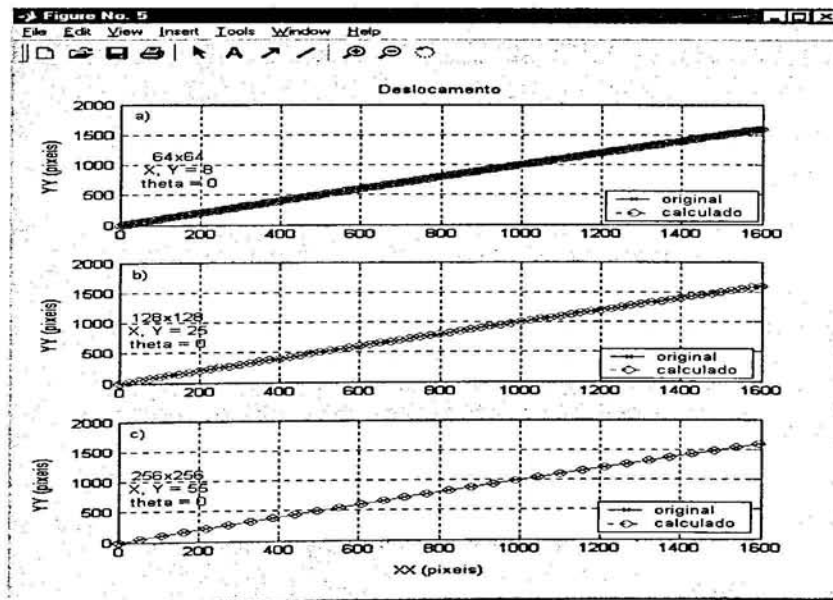


Figura 6.43 Deslocamento a partir de imagens desfasadas entre si de 8, 25 e 55 pixels, respectivamente, para as imagens de dimensão 64x64, 128x128 e 256x256 sem rotação.

Pode-se verificar, pela figura anterior, que o trajecto original coincide com o trajecto calculado.

A figura seguinte, Figura 6.44, mostra que o erro associado a cada ponto da trajectória calculado, se situa em torno dos valores de $\Delta\epsilon$ obtidos para o quadro da Figura 6.41.

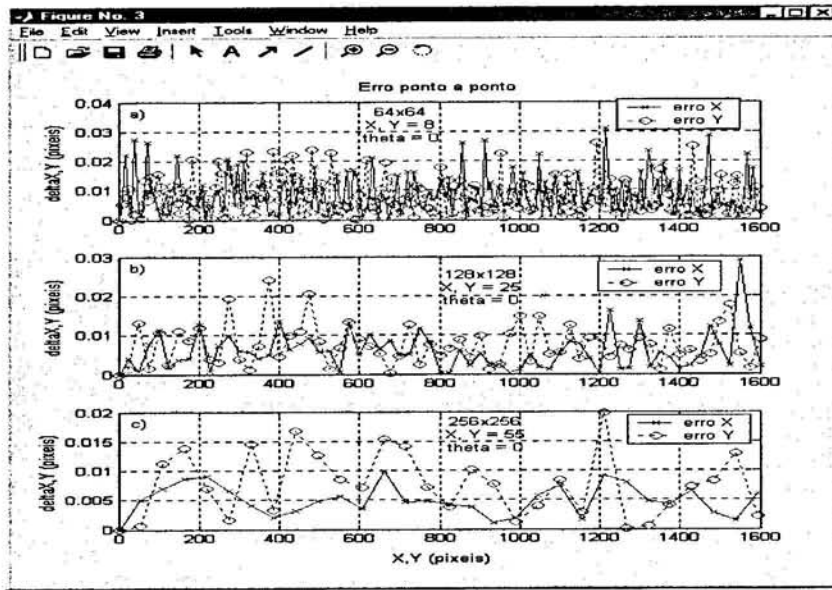


Figura 6.44 Erro registado em cada ponto calculado.

Na Figura 6.45 está registado o erro acumulado durante o trajecto, e pode-se verificar que o erro diminui à medida que aumenta a dimensão da imagem, como se tinha visto na Figura 6.42.

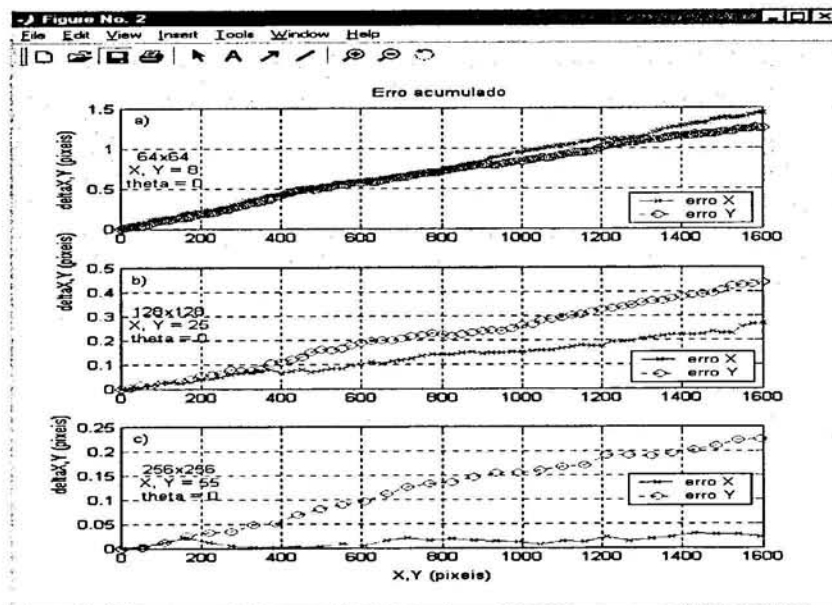


Figura 6.45 Erro acumulado durante o trajecto.

Pode-se ver pela figura seguinte que a velocidade máxima determinada, é sensivelmente a mesma que a indicada na Figura 6.42.

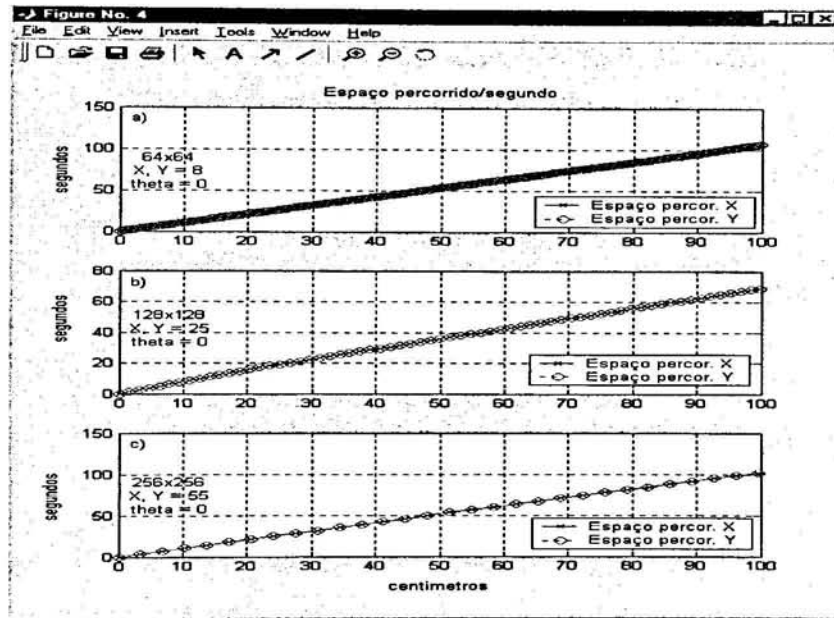


Figura 6.46 Espaço percorrido por segundo.

A partir da simulação anterior, pode-se concluir que os valores obtidos para o quadro da Figura 6.41 foram os correctos.

A simulação seguinte serviu para verificar o que poderá acontecer ao seleccionar valores superiores para Δd .

Os valores de Δd seleccionados foram 16, 40 e 70 pixels, respectivamente, para 64x64, 128x128 e 256x256.

A simulação foi feita para a mesma distância D , ou seja, 100cm e para as três dimensões das imagens, 64x64, 128x128 e 256x256.

Da figura que se segue pode-se ver que o deslocamento calculado, não coincide completamente com o original, esta situação deve-se ao facto de nem todos os pontos terem sido devidamente calculados, e verifica-se também que como para as imagens de dimensão 256x256 tem-se um deslocamento entre imagens elevado, a falha no cálculo de um ponto pode ser catastrófica.

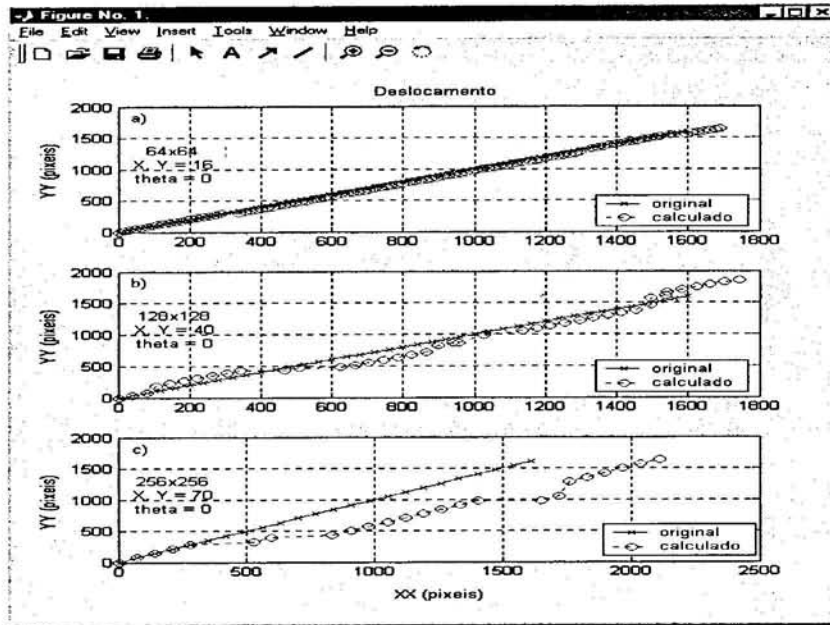


Figura 6.47 Deslocamento a partir de imagens desfasadas entre si de 16, 40 e 70 pixels, respectivamente, para as imagens de dimensão 64x64, 128x128 e 256x256 sem rotação.

Os pontos não foram devidamente calculados, ou seja, foram determinadas translações erradas, porque foram detectadas rotações, quando na realidade não existiam.

A detecção de uma rotação quando esta não existe, deve-se ao facto de, por exemplo, se existir uma linha na superfície e esta tiver uma determinada inclinação, o deslocar sobre essa linha pode dar a ideia que a imagem sofreu uma rotação Figura 6.48.

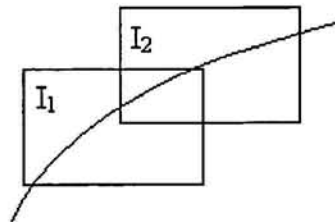


Figura 6.48 Imagens deslocadas sobre uma linha.

Na Figura 6.49 está representado o $\Delta\varepsilon$, para cada translação calculada.

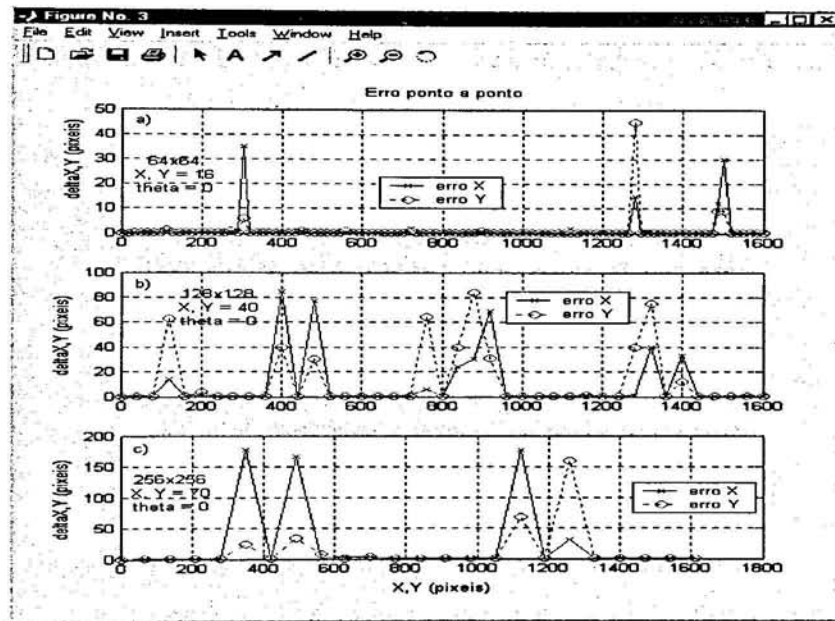


Figura 6.49 Erro registado em cada ponto calculado para a 2.^a simulação.

O erro acumulado pode ser visualizado na Figura 6.50.

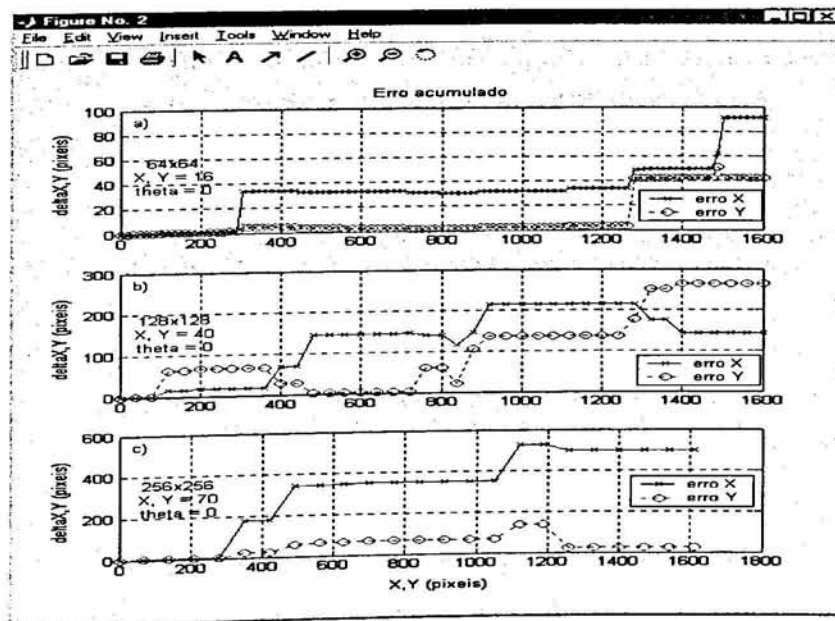


Figura 6.50 Erro acumulado durante o trajecto para a 2.^a simulação.

Na figura anterior para a dimensão de imagem 256x256 verifica-se que o erro acumulado é bastante elevado, apesar de só terem sido calculados quatro valores errados para a translação, situação já referida anteriormente.

A Figura 6.51 tem representada a velocidade máxima permitida. Ao comparar com a Figura 6.46, velocidade máxima para a simulação anterior, verifica-se que a velocidade aumentou, mas o espaço percorrido não foi o correcto.

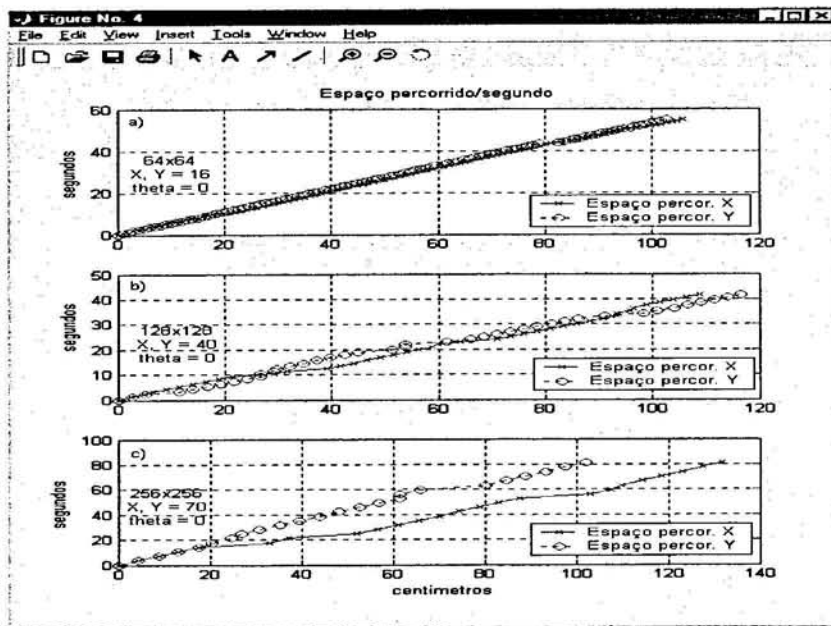


Figura 6.51 Espaço percorrido por segundo para a 2.^a simulação.

Conclui-se assim que se se aumentar o deslocamento Δd , com o objectivo de aumentar a velocidade, corre-se o risco do veículo não chegar devidamente ao destino. Nesta situação deve ter-se em atenção o tipo de superfície onde este se vai deslocar.

Capítulo 7

Arquitectura e aplicabilidade do sistema de navegação

Um sistema de condução de um veículo (por ex. AGV) tem normalmente uma arquitectura modular, composta por vários módulos, em que cada um desempenha uma função específica.

Estes módulos estão interligados entre si unicamente por parâmetros que permitem a troca de informação entre cada um, ou seja, os parâmetros de saída de um módulo serão os parâmetros de entrada do outro.

Assim, o sistema de navegação desenvolvido permite interligar-se facilmente numa arquitectura modular, porque necessita unicamente que lhe seja fornecido uma sequência de imagens para gerar como saída o posicionamento do veículo, determinado para cada par de imagens consecutivas.

Um sistema de condução de um veículo é constituído, em geral, por um conjunto de vários sistemas, em que podem ser designados por sistema de planeamento, gestão da navegação, posicionamento, controlo e por fim mecânico.

O sistema de planeamento define o trajecto que o veículo tem que efectuar, assim como as acções a executar.

O trajecto definido pelo sistema de planeamento, será passado ao sistema de gestão da navegação e este envia a informação do trajecto a percorrer (por ex. fio virtual), quer

para o sistema de controlo do veículo, quer para determinar o erro por comparação com a posição determinada pelo sistema de posicionamento.

O sistema de posicionamento, determina a posição do veículo em cada instante, por exemplo para os AGV's os métodos usados podem ser vistos no capítulo 2. Essa posição é comparada com a de referência, obtendo-se um erro que será fornecido ao sistema de controlo.

Por sua vez, o sistema de controlo verifica o erro, comparando com a trajectória fornecida pelo sistema de gestão da navegação, e dá instruções ao sistema mecânico.

A função do sistema mecânico, ligado à condução do veículo, é deslocar o veículo para as coordenadas pretendidas, a uma determinada velocidade.

O ciclo fica completo pela determinação da nova posição, recorrendo ao sistema de posicionamento.

Na Figura 7.1 podemos ver uma arquitectura de um sistema de condução de um veículo, como o apresentado.

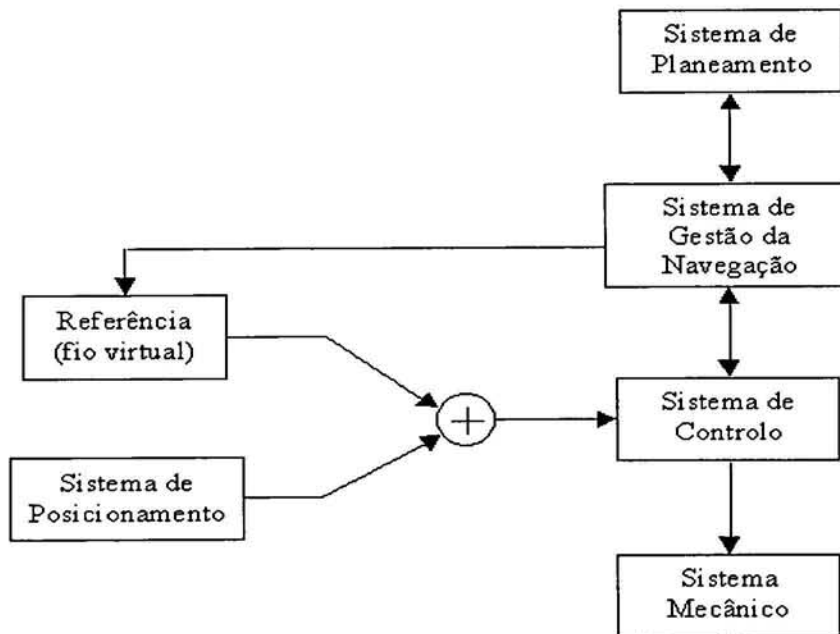


Figura 7.1 Sistema de condução de um veículo baseado numa arquitectura modular.

O sistema de navegação desenvolvido no âmbito desta dissertação enquadra-se perfeitamente no sistema de posicionamento indicado na Figura 7.1, uma vez que determina a posição do veículo a partir das imagens consecutivas do solo onde se desloca. Essa posição será comparada com a referência (por ex. fio virtual), gerando o erro necessário para o sistema de controlo.

Fica assim disponível mais um sistema a juntar aos apresentados no capítulo 2, permitindo permutar com relativa facilidade entre eles, seleccionando o que melhor cumpre os requisitos de projecto.

O sistema de navegação desenvolvido pode ser utilizado em várias aplicações, como iremos ver algumas delas no sub-capítulo seguinte.

7.1 Aplicabilidade do sistema de navegação desenvolvido

O sistema de navegação desenvolvido pode ser aplicado em vários equipamentos.

Um dos equipamentos mais adequados para a aplicação deste sistema é o AGV, uma vez que até foi este o principal motivador para esta tese.

A aplicação do sistema num AGV será relativamente fácil porque pode, por exemplo, substituir o sistema filoguiado, em que passamos a ter um fio virtual, que o AGV tem que seguir. Então o módulo de comparação terá simplesmente que comparar o trajecto calculado, a partir das imagens adquiridas, com o trajecto representado pelo fio virtual.

A aplicabilidade deste sistema dependerá principalmente da distância a percorrer, da velocidade, do erro e também do número de referências necessárias ao longo de cada trajecto.

Outras aplicações podem ser a nível dos robot's, comportando-se da mesma forma que nos AGV's, ou seja, permitindo a sua deslocação sem necessidade de haver nenhuma marca a identificar o trajecto.

Poderia também ser aplicável nos automóveis, por exemplo, nos sistemas de estacionamento automático, em desenvolvimento por algumas marcas, ou noutras aplicações

que não foram referidas, mas que não são de descurar, nomeadamente na área da aviação e na área marítima, entre outras.

Capítulo 8

Conclusões e perspectivas de trabalhos futuros

Os objectivos propostos no início do trabalho que conduziu a esta dissertação foram satisfatoriamente alcançados, ou seja, concluímos que o princípio de funcionamento do rato óptico pode ser utilizado para a navegação de AGV's em ambientes industriais reais.

A ferramenta de programação Matlab seleccionada para desenvolver e ensaiar todo o sistema, reduziu e simplificou bastante a programação necessária, assim como permitiu fazer uma verificação exhaustiva do sistema desenvolvido.

O método de *matching* baseado na transformada de Fourier – Mellin, mostrou ter um poder muito grande na determinação da rotação e da translação existentes entre duas imagens, como era pretendido para se poder determinar a trajectória a partir da sequência de imagens de uma superfície.

Da simulação efectuada concluiu-se que a superfície ideal para o sistema funcionar é uma superfície sem padrão nenhum na imagem. No entanto também se obtiveram bons resultados com superfícies de característica aleatória, superfícies encontradas frequentemente em naves fabris.

O trabalho de pesquisa efectuada na primeira parte desta dissertação, na área dos sistemas de navegação, e os resultados obtidos com o sistema desenvolvido, permitem-nos concluir que este sistema terá uma aplicação prática muito importante, quer na indústria

quer noutras aplicações, como por exemplo auxiliares de navegação para deficientes visuais.

São assim inúmeras as perspectivas futuras de utilização do sistema desenvolvido, permitindo uma vasta gama de aplicações em diferentes áreas.

A implementação prática do sistema deve passar pela melhoria de alguns aspectos essenciais:

- Melhorar a detecção da rotação, por exemplo através de filtros, por forma a delimitar melhor o pico máximo da correlação;
- Implementar a detecção da escala, permitindo o aumento ou a diminuição da resolução da imagem, logo podemos aumentar ou diminuir a velocidade, assim como a precisão do movimento;
- Melhorar o algoritmo por forma a diminuir o tempo de processamento entre imagens;
- Acrescentar funcionalidades, como por exemplo a previsão do movimento, pois se o veículo se desloca num determinado sentido, a uma determinada velocidade, é provável que não mude de sentido abruptamente, pelo que se isso aconteceu, então há uma grande probabilidade do valor determinado não ser o correcto.

Por outro lado o algoritmo adequa-se especialmente à utilização de processamento paralelo, o que pode contribuir para a sua utilização em situações mais exigentes.

Referências Bibliográficas

- [1] Manuel S. Silva, Carlos A. Sá e Francisco J. Restivo, *A oficina piloto do CCP – Centro de CIM do Porto*, TecnoMetal, n.º 180, pp. 21-25, Janeiro/Fevereiro, 1997.
- [2] AGV Guidance and control systems by Dan Carroll
<http://class.et.byu.edu/me586/Powerpoint/agv.ppt>.
- [3] AGV Electronics AB
<http://www.agve.se/frameset.htm>.
- [4] Brown L. G. (1992), *A survey of image registration techniques*, ACM Computing Surveys.
- [5] Vanden Elsen P. A., Pol D. E. & Viergever M. A. (1993), *Medical image matching – A review with classification*, IEEE Engineering in Medicine and Biology.
- [6] Chen Q. (1993), *Image Registration and its Applications in Medical Imaging*, PhD thesis, Vrije University Brussels, Faculty of Applied Sciences, Brussels.
- [7] Chapnick J. V., Noz M. E., Maguire G. Q. Jr., Kramer E. L., Sanger J. J., Birnbaum B. A. & Megibow A. J. (1993), *Techniques for multimodality image registration*, Proceedings of the 1993 IEEE 19th Annual Northeast Bioengineering Conference.
- [8] D. I. Barnea e H. F. Silverman, *A class of algorithms for fast image registration*, IEEE Trans. Computers, Vol. 21, No. 2, 1972, pp. 179-186.
- [9] G. J. Vanderburg e A. Rosenfeld, *Two – stage template matching*, IEEE Trans. Computers, Vol. 26, 1977, pp. 384-393.
- [10] M. Svedlow, C. P. McGlem, e P. E. Anuta, *Image registration: similarity measure and processing method comparisons*, IEEE Trans. Aerosp. Electron. Syst. Vol. 14, No. 1, 1978, pp. 141-149.

- [11] W. K. Pratt, *Digital Image Processing*, John Wiley & Sons, 1978, pp. 526-566.
- [12] A. Goshtasby, *Template matching in rotated images*, IEEE Trans. Patt. Anal. & Machin. Intellig. (PAMI), Vol.7, No.3, 1985, pp.338-344.
- [13] E. Peli, R. A. Angliere, e G. T. Timberlake, *Feature-based registration of retinal images*, IEEE Trans. Medic. Imaging, Vol.6, No.3, 1987, pp.272-278.
- [14] S. Mitra, B. s. Nutter, T. F. Krile, et al., *Automatic method for fundus image registration and analysis*, Applied Optics, Vol.27, No.6, 1988, pp.1107-1112.
- [15] N. Bownet e J. C. Liehn, *Image registration in electron microscopy: Application of a robust method*, J. Electron. Microsc. Tech., Vol.10, No.1, 1988, pp.27-33.
- [16] L. S. Hibbard e R. A. Hawkins, *Objective image alignment for three-dimensional reconstruction of digital autoradiograms*, J. Neurosci. Methods, Vol.3, No.3, 1983, pp.133-139.
- [17] J. L. Horner e P. D. Gianino, *Phase-only matched filtering*, Applied Opt., Vol.23, No.6, 1984, pp.812-816.
- [18] E. De Castro, G. Cristini, A. Martelli, et al., *Compensation of random eye motion in television ophthalmoscopy: preliminary results*, IEEE Trans. Med. Imag. Vol.8, No.3, 1989, pp.276-282.
- [19] R. J. Schalkoff, *Digital Image Processing and Computer Vision*, John Wiley & Sons, 1989, pp.279-286.
- [20] A. Apicella, J. S. Kippenhan, e J. H. Nagel, *Fast multi-modality image matching*, SPIE Vol.1092, Medical Imaging III: Image Processing, 1989, pp.252-263.
- [21] J. K. Brousil e D. R. Smith, *A threshold logic network for shape invariance*, IEEE Trans. Computers, EC-16, 1967, pp.818-828.
- [22] J. Altmann e H. J. P. Reitbock, *A fast correlation method for scale- and translation – invariant pattern recognition*, IEEE Trans. PAMI, Vol.6, No.1, 1984, pp.46-57.
- [23] Y. Sheng e H. H. Arsenault, *Experiments on pattern recognition using invariant*

- Fourier – Mellin descriptors*, J. Opt. Soc. Am. A, Vol.3, No.6, 1986, pp.771-776.
- [24] Z. Y. Sheng, C. Lejeune, e H. H. Arsenault, *Frequency-domain Fourier – Mellin descriptors for invariant pattern recognition*, Optical Engineering, Vol.27, No.5, 1988, pp.354-357.
- [25] J. Segman, *Fourier cross correlation and invariant transformations for an optimal recognition of functions deformed by affine groups*, J. Opt. Soc. Am. A, Vol.9, No.6, 1992, pp.895-902.
- [26] Y. N. Hsu, H. H. Arsenault, e G. April, *Rotation-invariant digital pattern recognition using circular harmonic expansion*, Applied Optics, Vol.21, No.22, 1982, pp.4012-4015.
- [27] D. Casasent e D. Psaltis, *Position, rotation and scale-invariant optical correlation*, Applied Optics, Vol.15, 1976, pp1793-1799.
- [28] D. Casasent e D. Psaltis, *New optical transforms for pattern recognition*, Proc. IEEE, Vol.65, 1977, pp.77-84.
- [29] D. Casasent e D. Psaltis, *Positional, rotational and scale invariant optical correlation method and apparatus*, U.S. Patent 4,084,255, 1978.
- [30] Y. N. Hsu e H. H. Arsenault, *Optical pattern recognition using circular harmonic expansion*, Applied Optics, Vol.21, No.22, 1982, pp.4016-4019.
- [31] E. L. Schwartz, *Spatial mapping in primate sensory projection: analytic structure and relevance to perception*, Biological Cybernetics, Vol.25, 1977, pp.181-194.
- [32] E. L. Schwartz, *Computational anatomy and functional architecture of striate cortex: A spatial mapping approach to perceptual coding*, Vision Res., Vol.20, 1980, pp.645-669.
- [33] E. L. Schwartz, *Cortical anatomy and spatial frequency analysis*, Perception, Vol.10, 1981 pp.455-468.
- [34] E. L. Schwartz, *Cortical mapping and perceptual invariance: A replay to Cavanagh*, Vision Research, Vol.23, 1983, pp.831-835.

- [35] P. Cavanagh, *Size and position invariance in the visual system*, Perception, Vol.7, 1978, pp.167-177.
- [36] P. Cavanagh, *Size invariance: Reply to Schwartz*, Perception, Vol.10, 1981, pp.491-499.
- [37] P. Cavanagh, *Functional size invariant is not provided by the cortical magnification factor*, Vision Res., Vol.22, 1982, pp.1409-1412.
- [38] C. Braccini, G. Gambardella, e A. Grattarola, *Shift-variant image processing for scale – invariant recognition*, Proceedings of SPIE, Vol.397, 1983, pp.318-325.
- [39] L. Massone, G. Sandini, e V. Tagliasco, *'Form-invariant' topological mapping strategy for 2D shape recognition*, CVGIP, Vol. 30, 1985, pp169-188.
- [40] A. Fuchs e H. Haken, *Pattern recognition and associative memory as dynamical processes in a synergetic system*, Biol. Cybern. Vol.60, 1988, pp.17-22.
- [41] J. L. Horner e P. D. Gianino, *Phase-only matched filtering*, Applied Optics, Vol.23, No.6, 1984, pp.812-816.
- [42] E. De Castro e C. Morandi, *Registration of translated and rotated images using finite Fourier transforms*, IEEE Trans. PAMI, Vol.9, No.5, 1987, pp.700-703.
- [43] D. L. G. Hill, D. J. Hawkes, N. A. Harrison, *A strategy for automated multimodal-ity image registration incorporating anatomical knowledge and imager characteristics*, in IPMI, H. H. Barrett e A. F. Gmitro eds., Springer-Verlag, 1993, pp.182-196.
- [44] M. Singh, W. Frei, T. Shibata, et al., *A digital technique for accurate change detection in nuclear medical images - with application to myocardial perfusion studies using Thallium- 201*, IEEE Trans. Nulc. Scien. Vol.26, No.1, 1979, pp.565-573.
- [45] G. Q. Maguire Jr, M. E. Noz, H. Rucinek, et al., *Graphics applied to medical image registration*, IEEE Computer Graphics & Applications, Vol.11, March, 1991, pp.20-28.

- [46] K. S. Arun, T. S. Huang, e S. D. Blostein, *Least squares fitting of two 3-D point sets*, IEEE Trans. PAMI, Vol.9, No.5, 1987, pp.698-700.
- [47] M. H. Deverell e W. F. Whimster, *A method of image registration for three-dimensional reconstruction of microscopic structures using an IBAS 2000 image analysis system*, Pathol. Res. Pract. Vol.185, No.5, 1989.
- [48] E. L. Kramer e M. E. Noz, *CT-SPECT fusion for analysis of radiolabeled antibodies: applications in gastrointestinal and lung carcinoma*, Nucl. Med. Biol., Vol.18, No.1, 1991, pp.27-42.
- [49] P. A. Kenny, D. J. Dowsett, D. Vernon, et al., *A technique for digital image registration used prior to subtraction of lung images in nuclear medicine*, Phys. Med. Biol., Vol.35, No.5, 1990, pp.679-685.
- [50] D. A. Perednia e R. G. White, *Automatic registration of multiple skin lesions by use of point pattern matching*, Computerized Medical Imaging and Graphics, Vol.16, No.3, 1992, pp.205-216.
- [51] M. Merickel, *3D reconstruction: the registration problem*, CVGIP, Vol.42, 1988, pp.206-219.
- [52] R. Y. Tsai e T. S. Huang, *Estimating three-dimensional motion parameters of a rigid planar patch*, IEEE Trans. Acous., Speech, & Signal Process., Vol.29, No.6, 1981, pp.1147-1152.
- [53] G. Q. Maguire, M. E. Noz, E. M. Lee, et al., *Correlation methods for tomographic images using two and three dimensional techniques*, in IPMI, S. L. Bacharach ed., Martine Nijhoff, 1986, pp.266-279.
- [54] R. Boesecke, T. Bruckner, e G. Ende, *Landmark based correlation of medical images*, Phys. Med. Biol., Vol.35, No.1, 1990, pp.121-126.
- [55] R. Y. Tsai, T. S. Huang, e W. L. Zhu, *Estimating three-dimensional motion parameters of a rigid planar patch, II: singular value decomposition*, IEEE Trans. Acous, Speech, & Signal Process., Vol.30, No.4, 1982, pp.525-534.

- [56] R. Y. Tsai e T. S. Huang, *Estimating three-dimensional motion parameters of a rigid planar patch, III: finite point correspondences and the three-view problem*, IEEE Trans. Acous, Speech, & Signal Process., Vol.32, No.2, 1984, pp. 264-274.
- [57] A. C. Evans, S. Marrett, L. Collins, et al., *Anatomical-functional correlative analysis of the human brain using three dimensional imaging systems*, SPIE Vol.1092, Medical Imaging III: Image Processing, 1989, pp.264-274.
- [58] S. Marrett, A. C. Evans, L. Collins, et al., *A volume of interest (VOI) atlas for the analysis of neurophysiological image data*, SPIE Vol.1092, Medical Imaging III: Image Processing, 1989, pp.467-477.
- [59] D. J. Hawkes, D. L. Hill, et al., *Preliminary work on the interpretation of SPECT images with the aid of registered MR images and an MR derived 3D neuro-anatomical atlas*, in 3D Imaging in Medicine, K. H. Hohne eds., Springer-Verlag, 1990, pp.241-251.
- [60] I. Kapouleas, A. Alavi, et al., *Registration of three dimensional MR and PET images of the human brain without markers*, Radiology, Vol.181, No.3, 1991, pp.731-739.
- [61] S. Umeyama, *Least-squares estimation of transformation parameters between two point patterns*, IEEE Trans. PAMI, Vol.13, No.4, 1991, pp.376-380.
- [62] D. L. G. Hill, D. J. Hawkes, J. E. Crossman, et al., *Registration of MR and CT images for skull base surgery using point-like anatomical features*, The British Journal of Radiology, Vol.64, No.767, 1991, pp.1030-1035.
- [63] M. K. Hu, *Visual pattern recognition by moment invariants*, IRE Trans. Information Theory, Vol.8, pp179-187, Feb., 1962.
- [64] S. Maitra, *Moment invariants*, IEEE Proceedings, Vol.67, No.4, 1979, pp697-699.
- [65] F. W. Smith e M. H. Wright, *Automatic ship photo interpretation by the method of moments*, IEEE Trans. Computer, Vol.20, Sept, 1971, pp1089-1094.
- [66] S. A. Dudani, K. J. Breeding, e R. B. McGhee, *Aircraft identification by moment*

- invariants*, IEEE Trans. Computer, Vol.26, No.1, 1977, pp39-45.
- [67] C. H. Teh e R. T. Chin, *On image analysis by the methods of moments*, IEEE Trans. PAMI, Vol.10, No.4, 1988, pp496-513.
- [68] Y. S. Abu-Mostafa e D. Psaltis, *Recognitive aspects of moment invariants*, IEEE Trans. PAMI, Vol.6, No.6, 1984, pp698-706.
- [69] M. O. Freeman e B. E. A. Saleh, *Moment invariants in the space and frequency domains*, J. Opt. Soc. Am. A, Vol.5, No.7, 1988, pp1073-1084.
- [70] S. A. Dudani, *The distance-weighted k-nearest-neighbour rule*, IEEE Trans. Syst. Man. Cybern., Vol.6, No.4, 1976, pp325-327.
- [71] R. O. Duda e P. E. Hart, *Pattern Classification and Scene Analysis*, New York, Wiley, 1973.
- [72] A. Gamboa-Aldeco e G. T. Y. Chen, *Correlation of 3D surfaces from multiple modalities in medical imaging*, SPIE Proceedings, 1986, pp460-466.
- [73] N. M. Alpert, J. F. Bradshaw, D. Kennedy, e J. A. Correia, *The principal axes transformation - a method for image registration*, J. Nucl. Med., Vol.31, No.10, 1990, pp1717-1722.
- [74] M. R. Teague, *Image analysis via the general theory of moments*, J. Opt. Soc. Am., Vol. 70, No.8, 1980, pp920-930.
- [75] S. S. Reddi, *Radial and angular moment invariants for image identification*, IEEE Trans. PAMI, Vol.3, No.2, 1981, pp240-242.
- [76] J. S. Wierak, *Moment invariants in theory and practice*, Image and Vision Computing, Vol.1, No.2, 1983, pp79-83.
- [77] R. Safaee-Rad, K. C. Smith, B. Benhabib, e I. Tchoukanov, *Application of moment and Fourier descriptors to the accurate estimation of elliptical shape parameters*, Conference Proceed., IEEE Signal Processing, Toronto, 1991, pp2465-2468.
- [78] B. V. K. V. Kumar, *Geometric moments computed from the Hartley transform*, Optical Engineering, Vol.25, No.12, 1986, pp1327-1332.

- [79] A. P. Reeves, R. J. Prokop, S. E. Andrews, e et al., *Three-dimensional shape analysis using moments and Fourier descriptor*, IEEE Trans. PAMI, Vol.10, No.6, 1988, pp937-943.
- [80] B. G. Mertzios e K. D. Tsirikolias, *Fast shape discrimination using one-dimensional moments*, Confer. Proceed, IEEE Signal Processing, Toronto, 1991, pp2473-2475.
- [81] K. D. Toennies, J. K. Udupa, G. T. Herman, et al., *Registration of 3D objects and surfaces*, IEEE Computer Graphics & Applications, May, 1990, pp.52-53.
- [82] G. T. Bartoo e W. A. Hanson, *Multi-modality image registration using centroid mapping*, IEEE 11th Annual Int. Conf. Eng. Med. & Biol. pp.550-551.
- [83] A. Goshtasby, *Template matching in rotated images*, IEEE Trans. PAMI, Vol.7, No.3, 1985, pp.338-344.
- [84] D. W. Ro, L. Axel, G. T. Herman, and et al., *Computed masks in coronary subtraction imaging*, IEEE Trans. Med. Imaging, Vol.6, No.4, 1987, pp.297-300.
- [85] R. Sugita, S. Takahashi, K. Ishii, and et al., *Brain CT and MR findings in hyperphenylalaninemia due to dihydropteridine reductase deficiency (variant of phenylketonuria)*, J. Comput. Assist. Tomogr., 1990, Vol.14, No.5, pp.699-703.
- [86] C. C. Meltzer, R. N. Bryan, et al., *Anatomical localization for PET using MR imaging*, J. Computer Assisted Tomography, Vol.14, 3, 1990, pp.418-426.
- [87] B. L. Holman, R. E. Zimmerman, et al., *Computer-assisted superimposition of magnetic resonance and high-resolution Technetium-99m-HMPAO and Thallium-201 SPECT images of brain*, J. Nuclear Medicine, Vol.32, 8, 1991, pp.1478-1484.
- [88] G. Nagy, *Digital image-processing activities in remote sensing for earth resources*, Proceedings of the IEEE, Vol.60, No.10, 1972, pp.1177-1200.
- [89] R. M. Haralick, *Automatic remote sensor image processing*, in *Digital Picture Analysis*, Ed. A. Rosenfeld, Springer-Verlag, 1976, pp.5 - 66.
- [90] D. A. Landgrebe, *Analysis technology for land remote sensing*, Proceedings of the

- IEEE, Vol.69, No.5, 1981, pp.628 – 642.
- [91] D. H. McMahon, G. L. Johnson, et al., *A hybrid optical computer processing technique for fingerprint identification*, IEEE Trans. Computers, Vol.24, No.4, 1975, pp.358 - 369.
- [92] C. V. K. Rao, *On fingerprint pattern recognition*, Pattern Recognition, Vol.10, 1978, pp.15 - 18.
- [93] D. K. Isenor e S. G. Zaky, *Fingerprint identification using graph matching*, Pattern Recognition, Vol.19, No.2, 1986, pp.113 - 122.
- [94] B. M. Mehtre, N. N. Murthy, S. Kapoor, *Segmentation of fingerprint images using the directional image*, Pattern Recognition, Vol.20, No.4, 1987, 429 - 435.
- [95] L. O'Gorman e J. V. Nickerson, *An approach to fingerprint filter design*, Pattern Recognition, Vol.22, No.1, 1989, pp.29 - 38.
- [96] B. Bhanu e O. D. Faugeras, *Shape matching of two-dimensional objects*, IEEE Trans. PAMI, Vol.6, No.2, 1984, pp.137 - 155.
- [97] D. Vernon, *Two-dimensional object recognition using partial contours*, Image and Vision Computing, Vol.5, No.1, 1987, pp.21 - 27.
- [98] H. C. Liu, e M. D. Srinath, *Partial shape classification using contour matching in distance transformation*, IEEE Trans. PAMI, Vol.12, No.11, 1990, pp.1072 - 1079.
- [99] David Casasent e Demetri Psaltis. *Position, Rotation, and scale-invariant optical correlation*. *Applied Optics*, 15(7): 1795 – 1799, July 1976.
- [100] B. S. Reddy e B. N. Chatterji. *An fft-based technique for translation, rotation, and scale-invariant image registration*. *IP*, 5(8): 1266 – 1277, August 1996.
- [101] Q. S. Chen, M. Defrise, e F. Deconinck. *Symmetrical phase – only matched filtering of Fourier – Mellin transform for image registration and recognition*. *PAMI*, 16(12): 1156 – 1168, December 1994.
- [102] Francisco J. Restivo, *Processamento Digital de Sinal*, Tópicos, Ano lectivo 1998/99.

<http://www.fe.up.pt/~fjr/ensino/20022003/pds.html>

- [103] Roman Kuc, *Introduction to Digital Signal Processing*, Electrical Engineering Series, McGraw Hill International Editions, 1982, pp. 65 – 156.
- [104] Rafael C. Gonzalez e Richard E. Woods, *Digital Image Processing*, Addison – Wesley Publishing company, 1993, pp. 81 – 117.
- [105] Kenneth R. Castleman, *Digital Image Processing*, Prentice Hall, 1996, pp.171 – 206.
- [106] A. V. Oppenheim, R. W. Schaffer, *Digital Signal Processing*, Prentice Hall, 1975.
- [107] Estação Fotogramétrica Digital Educacional.
<http://e-foto.sourceforge.net/cap4.pdf>
- [108] Joseph J. K. Ó Ruanaidh e Thierry Pun *Rotation, Scale and Translation Invariant Digital Image Watermarking*, 21 August 1997
<http://citeseer.nj.nec.com/183296.html>
- [109] Steven W. Smith, *The Scientist and Engineers Guide to Digital Signal Processing*, California Technical Publishing, 1997, pp. 123 – 242.
- [110] Dongarra, J. Bunch, C. Moler, e G. W. Stewart. *LINPACK User's Guide*. SLAM, Philadelphia, P.A., 1979.
- [111] B. Smith, J. Boyle, Y. Ikebe, V. Klema, e C. Moler, *Matrix Eigensystem Routines: EISPACK Guide*, Springer – Verlag, New York, NY, Second ed., 1970.
- [112] Peter Kovesi – Department of Computer Science & Software Engineering – University of Western Australia – pk@cs.uwa.edu.au – December 2002.
www.cs.uwa.edu.au/~pk

