

Faculdade de Engenharia e Faculdade de Ciências
da Universidade do Porto



*Emparelhamento de Objectos
Representados em Imagens
Usando Técnicas de
Optimização*

Francisco Paulo Marques de Oliveira

Fevereiro – 2008

Emparelhamento de Objectos Representados em Imagens Usando Técnicas de Optimização

Dissertação submetida para satisfação parcial dos requisitos do grau
de Mestre em Métodos Computacionais em Ciências
e Engenharia pela Universidade do Porto

Francisco Paulo Marques de Oliveira

Licenciado em Ensino de Matemática pela
Universidade do Minho (2001)

Dissertação realizada sob a orientação de
Professor Doutor João Manuel R. S. Tavares
Departamento de Engenharia Mecânica e Gestão Industrial
Faculdade de Engenharia da Universidade do Porto

Porto, Fevereiro – 2008

AGRADECIMENTOS

À Faculdade de Engenharia da Universidade do Porto por ter aceitado o meu reingresso num ano em que não houve edição do curso de Mestrado em Métodos Computacionais em Ciências e Engenharia

Ao professor João Tavares pelas orientações e sugestões que me deu no desenvolvimento deste trabalho.

RESUMO

O tema desta Dissertação insere-se no domínio da Visão Computacional e na área da análise de objectos deformáveis representados em imagens. Mais concretamente, tenta-se determinar a correspondência entre elementos homólogos de objectos representados em imagens. Estas correspondências podem ser associadas a um “custo”, sendo este uma medida de similaridade entre os objectos a emparelhar. O custo considerado pode ser utilizado para reconhecimento/identificação de objectos, quando, por exemplo, é realizada a comparação com modelos base. Por outro lado, a determinação das correspondências permite o seguimento do movimento/deformação que o objecto possa ter sofrido. Este tipo de tarefas de análise de objectos em imagens tem elevada importância em sectores do conhecimento como a Medicina e a Indústria, por exemplo.

O trabalho aqui apresentado vem na sequência de trabalho desenvolvido e coordenado pelo Orientador desta Dissertação e centra-se especialmente na correspondência entre conjuntos de pontos que definem os contornos exteriores de dois objectos a emparelhar. Este trabalho tem como principais objectivos: impedir correspondências cruzadas verificadas em metodologias anteriormente desenvolvidas, utilizar técnicas de optimização para obtenção das correspondências e respectivos custos, desenvolver metodologias alternativas para determinação de matrizes de custo de emparelhamento.

Esta Dissertação inicia com uma análise do estado da arte na área do emparelhamento e reconhecimento de objectos representados em imagens. Seguidamente, no terceiro capítulo estuda-se o problema da ordenação dos pontos de um contorno, pois com a atribuição de uma ordem coerente pode-se mais facilmente desenvolver metodologias de optimização das correspondências entre contornos nas quais não sejam permitidos emparelhamentos cruzados. Ainda no terceiro capítulo, apresentam-se três formulações para a determinação de uma ordem para os pontos dos contornos baseadas em programação linear com *branch-and-bound*, programação inteira e *simulated annealing*. Posteriormente, apresentam-se os resultados da nossa implementação da terceira formulação proposta, os quais foram bons em termos de eficácia e velocidade.

No capítulo seguinte é apresentada uma metodologia para determinar a transformação rígida que melhor alinha dois contornos, utilizando informação relativa à ordem dos pontos. Posteriormente, no quinto capítulo são propostas duas formulações para determinar um emparelhamento global entre pontos de dois contornos, respeitando a ordem dos pontos. A primeira baseia-se num modelo de programação linear inteira e a segunda na técnica de optimização de programação dinâmica. Esta segunda solução foi implementada e testada usando matrizes de custo de emparelhamento determinadas usando uma plataforma computacional de desenvolvimento e ensaio onde já estava integrada uma implementação da metodologia baseada em modelação geométrica proposta por Shapiro. Os resultados obtidos foram melhores, quer em termos de qualidade dos emparelhamentos quer em termos de velocidade, do que os obtidos pelas metodologias de optimização baseadas em modelos de afectação clássicos previamente integrados na referida plataforma: método *Húngaro*, *Simplex* para problemas de fluxo e *LAPm*.

O sexto capítulo é dedicado ao desenvolvimento, implementação e teste de metodologias de emparelhamento de contornos ordenados, utilizando informação de curvatura e/ou de distância dos pontos aos respectivos centróides. Para emparelhar os pontos com base nestas características, foram ensaiadas metodologias de emparelhamento do tipo *um-para-um* e do tipo *um-para-vários* (ou *vários-para-um*). Os resultados globais obtidos demonstram a adequação destas metodologias para o emparelhamento de contornos definidos por conjuntos de pontos ordenados. Finalmente, no último capítulo são apresentadas as principais conclusões e as perspectivas de trabalho futuro.

ABSTRACT

The subject of this Master Thesis is in the Computational Vision domain, in particularly in the analysis of deformable objects represented in image sequences. To be specific, it tries to determine the correspondence between homologous elements of two objects represented in images. These correspondences can be associated to a “cost”, being this cost a measure of similarity between the objects. Then, the total matching cost can be used to recognize/identify objects, when they are compared with base models, for instance. On the other hand, the determination of the correspondences allows the tracking of the movement/deformation that the matched object might have suffered. Thus, these types of tasks have high importance in many practical areas as in Medicine and Industry, for example.

The work presented here comes in the sequence of the work previously developed and coordinated by the supervisor of this Thesis that focuses especially in the search for the correspondence between two groups of points that define the external contours of two objects. Its main objectives are to avoid crossed correspondences verified in previous methodologies, to employ optimization techniques to obtain the correspondences and respective costs, to develop new methodologies for determination of the matching cost matrices.

The work developed here begins with an analysis of the state-of-the-art in matching and recognition of objects represented in images.

Afterwards, in the third chapter we study the problem of ordination of contour points, because with the attribution of a coherent order we can more easily develop methodologies of optimization of correspondences between two contours in which crossed matches are not allowed. In this chapter, we present three formulations to attribute a coherent order to the points. The first one is based in linear programming with *branch-and-bound*, the second one in integer programming and the last one in *simulated annealing*. Afterwards, we present the results of our implementation of the third formulation proposed, which were good in terms of effectiveness and speed.

In the following chapter, we present a methodology to determine the rigid transformation that best aligns two contours, using information of points order.

Subsequently, in the fifth chapter, we propose two formulations to determine the correspondences among points of two contours with order restriction, one based in a model of integer linear programming and another one based in the optimization technique of dynamic programming. We implemented and tested the second solution. For such, we used the build matching cost matrices using the geometric modeling proposed by Shapiro. These matrices were determined using a computational platform of development and test, already developed. The experimental results obtained were better in matching quality and speed in comparison with the results obtained by the optimization methodologies based in models of classic assignment algorithms, also integrated in the computational platform referred: *Hungarian* method, *Simplex* for flow problems and *LAPm*.

In the sixth chapter, we present the development, implementation and testing of new matching methodologies for contours shapes defined by groups of ordered points, using curvature information and/or distance of the points to the respective centre. Based in these characteristics, we developed methodologies to match contours points allowing correspondences of types *one-to-one* and *one-to-several* (or *several-to-one*). The fine results obtained demonstrated the good adaptation of these methodologies to the matching of contours defined by groups of ordered points. Finally, in the last chapter we present the conclusions and plans for further work.

ÍNDICE

CAPÍTULO I: INTRODUÇÃO À DISSERTAÇÃO E SUA ESTRUTURA.....	1
1.1 <i>Introdução.....</i>	3
1.2 <i>Objectivos e abordagem seguida.....</i>	3
1.3 <i>Principais contribuições.....</i>	5
1.4 <i>Estrutura organizativa.....</i>	6
CAPÍTULO II: MÉTODOS DE EMPARELHAMENTO DE OBJECTOS REPRESENTADOS EM IMAGENS.....	9
2.1 <i>Introdução.....</i>	11
2.2 <i>Métodos baseados no sinal.....</i>	12
2.3 <i>Métodos baseados em características da forma</i>	15
2.3.1 <i>Formas representadas por conjuntos de pontos não ordenados</i>	16
2.3.2 <i>Formas representados pelo contorno.....</i>	19
2.4 <i>Métodos baseados na estrutura da forma</i>	22
2.5 <i>Sumário.....</i>	27
CAPÍTULO III: ORDENAÇÃO DOS PONTOS QUE DEFINEM UM CONTORNO.....	29
3.1 <i>Introdução.....</i>	31
3.2 <i>Definição do problema</i>	31
3.3 <i>Problema do caixeiro-viajante</i>	33
3.4 <i>Ordenação dos pontos de um contorno usando o simplex e a técnica de branch-and-bound... 34</i>	34
3.5 <i>Ordenação dos pontos de um contorno usando uma formulação de programação inteira</i>	38
3.6 <i>Aplicação do método simulated annealing à ordenação dos pontos de um contorno.....</i>	39
3.6.1 <i>Apresentação do método.....</i>	39
3.6.2 <i>Implementação</i>	41
3.6.3 <i>Análise de resultados</i>	43
3.7 <i>Determinação do sentido pelo qual o contorno está definido.....</i>	47
3.8 <i>Sumário e conclusões</i>	49
CAPÍTULO IV: DETERMINAÇÃO DA TRANSFORMAÇÃO RÍGIDA QUE MELHOR ALINHA DOIS CONTORNOS.....	51
4.1 <i>Introdução.....</i>	53

4.2	<i>Centróide e translação</i>	53
4.3	<i>Escala</i>	54
4.4	<i>Ângulo de rotação</i>	56
4.5	<i>Resultados</i>	57
4.6	<i>Sumário e conclusões</i>	61
CAPÍTULO V: DETERMINAÇÃO DE UM EMPARELHAMENTO ÓPTIMO ENTRE DOIS CONTORNOS COM BASE NUMA MATRIZ DE CUSTOS		63
5.1	<i>Introdução</i>	65
5.2	<i>Definição do problema</i>	66
5.3	<i>Formulação de programação inteira</i>	68
5.4	<i>Formulação de programação dinâmica</i>	73
5.4.1	<i>Formulação geral</i>	73
5.4.2	<i>Algoritmo e implementação</i>	76
5.4.3	<i>Custo computacional</i>	78
5.5	<i>Algoritmo de programação dinâmica versus algoritmos de afectação clássicos</i>	79
5.5.1	<i>Metodologia utilizada para obtenção de uma matriz de custos ou afinidade</i>	79
5.5.2	<i>Comparação de resultados</i>	83
5.5.2.1	<i>Definição dos parâmetros utilizados nos ensaios</i>	83
5.5.2.2	<i>Resultados</i>	85
5.5.2.3	<i>Análise de resultados</i>	91
5.6	<i>Emparelhamentos do tipo um-para-vários</i>	93
5.6.1	<i>Definição</i>	93
5.6.2	<i>Baseado numa matriz de custos</i>	94
5.6.3	<i>Baseado na minimização da distância euclidiana entre pontos</i>	95
5.7	<i>Sumário e conclusões</i>	96
CAPÍTULO VI: EMPARELHAMENTOS BASEADOS EM INFORMAÇÃO DE CURVATURA E DISTÂNCIA AO CENTRÓIDE		99
6.1	<i>Introdução</i>	101
6.2	<i>Emparelhamento baseado em informação de curvatura</i>	102
6.2.1	<i>Princípio da metodologia</i>	102
6.2.2	<i>Resultados do emparelhamento usando programação dinâmica</i>	104

6.2.3 Análise de resultados	112
6.3 Emparelhamento baseado na distância ao centróide	114
6.3.1 Princípio da metodologia	114
6.3.2 Resultados dos emparelhamentos usando programação dinâmica.....	115
6.3.3 Análise de resultados	125
6.4 Emparelhamento baseado em informação de curvatura e distância ao centróide.....	126
6.4.1 Princípio da metodologia	126
6.4.2 Resultados dos emparelhamentos usando programação dinâmica.....	128
6.4.3 Análise de resultados	132
6.5 Emparelhamento com ajuste local	133
6.5.1 Princípio da metodologia	133
6.5.2 Resultados e análise dos emparelhamentos.....	135
6.6 Resultados dos emparelhamentos do tipo um-para-vários	137
6.6.1 Baseado numa matriz de custos	137
6.6.2 Baseado na minimização da distância euclidiana entre pontos a emparelhar.....	139
6.6.3 Análise de resultados	141
6.7 Sumário e conclusões	142
CAPÍTULO VII: CONCLUSÕES FINAIS E PERSPECTIVAS DE TRABALHO FUTURO	145
7.1 Conclusões finais	147
7.1.1 Conclusões gerais.....	147
7.1.2 Conclusões relativas aos algoritmos desenvolvidos	149
7.2 Perspectivas de trabalho futuro	155
BIBLIOGRAFIA	159

Tabela 3.1: Resultados da ordenação de três dos contornos usados nos ensaios, em função do valor inicial do parâmetro T (temperatura). (\bar{X} representa a média da distância entre os pontos do contorno.)....	45
Tabela 4.1: Resultados relativos à transformação rígida existente entre dois contornos, determinada pela metodologia proposta.	59
Tabela 4.2: Representação do alinhamento dos pares de contornos considerados nos ensaios nos quais foi aplicada apenas uma determinada transformação rígida.....	59
Tabela 4.3: Representação do alinhamento de pares de contornos considerados nos ensaios nos quais foi aplicada uma transformação rígida e uma deformação local reduzida.	60
Tabela 4.4: Representação do alinhamento de pares de contornos considerados nos ensaios nos quais foi aplicada uma transformação rígida e uma deformação local mais acentuada.	60
Tabela 5.1: Custos mínimos guardados pelo algoritmo de programação dinâmica em função do estágio e do estado para o primeiro problema de minimização relativo ao exemplo em estudo.....	78
Tabela 5.2: Comparação da velocidade de execução do algoritmo de programação dinâmica relativamente aos algoritmos de afectação: método Húngaro, Simplex para problemas de fluxo e LAPm.....	91
Tabela 6.1: Resultados dos emparelhamentos dos contornos ilustrados nas Figuras 6.2 a 6.11, usando a metodologia baseada em informação de curvatura e optimização com o algoritmo de programação dinâmica.....	110
Tabela 6.2: Resultados dos emparelhamentos dos contornos ilustrados nas Figuras 6.15 a 6.26, usando a metodologia baseada nas distâncias aos centróides e optimização com o algoritmo de programação dinâmica.....	123
Tabela 6.3: Resultados do emparelhamento dos diversos contornos de ensaio, usando a metodologia baseada em informação de curvatura e distância ao centróide, e optimização com o algoritmo de programação dinâmica.....	130
Tabela 6.4: Custos e tempos de execução relativos a emparelhamentos do tipo um-para-vários usando a metodologia baseada numa matriz de custos.	139

Figura 2.1: Representação tridimensional do gráfico da função Φ definida pela expressão (2.1), relativamente ao contorno do rectângulo vermelho.....	21
Figura 2.2: Exemplo de duas formas e respectivos eixos médios no seu interior. (Retirados de [Sebastian, 2004].).....	22
Figura 2.3: Forma geométrica a cinzento-claro com contorno a preto e respectivo eixo médio a cinzento-escuro, representados numa grelha de <i>pixels</i> . Os valores indicados representam a distância de cada <i>pixel</i> ao contorno, utilizando a métrica induzida pela norma do máximo.....	24
Figura 2.4: Construção do eixo médio de um contorno fechado. O eixo médio é definido pelo centro dos discos máximos contidos na forma e que incidem em pelo menos dois pontos do contorno.	24
Figura 2.5: Imagens das primitivas utilizadas em [Zhu, 1996]. Estrutura tubular (a), círculo (b), deformações da estrutura tubular e do círculo e consequente adaptação à forma final em questão (c) e (d).	25
Figura 2.6: Uma forma poligonal e o respectivo <i>shock-graph</i> no seu interior.	26
Figura 2.7: Duas formas poligonais e o respectivo eixo médio no seu interior. A forma da direita resulta da introdução de uma pequena perturbação na forma da esquerda, originando um eixo médio significativamente diferente.	27
Figura 3.1: Dois contornos distintos definidos pelo mesmo conjunto de pontos.	32
Figura 3.2: Exemplo do processo de inversão da ordem na sequência de pontos de um contorno. Neste caso, foi invertida uma sequência formada por 4 pontos.....	43
Figura 3.3: Exemplo do processo de troca de posição de uma secção da sequência de pontos de um contorno. Neste caso, os <i>pontos 2 e 3</i> , que estavam entre os <i>pontos 5 e 1</i> na sequência inicial, foram colocados entre os <i>pontos 1 e 4</i>	43
Figura 3.4: Exemplos de três dos contornos utilizados para fazer os testes do algoritmo de ordenação dos pontos: (a) “ <i>heart3</i> ”; (b) “ <i>tree1</i> ”; (c) “ <i>heartB1</i> ”. Em cada figura, à esquerda, os pontos; ao centro os pontos unidos pela sequência inicial e, à direita, os pontos unidos após ordenação.	46
Figura 3.5: Exemplo de um contorno ordenado no sentido contrário ao dos ponteiros do relógio.	48
Figura 3.6: Duas configurações do mesmo contorno. À esquerda o contorno original e à direita o mesmo contorno após uma rotação.	48
Figura 4.1: Dois contornos de dois quadrados exactamente iguais. À esquerda o centróide foi calculado como o centro de massa e à direita o centróide foi calculado pelo método proposto.	54
Figura 5.1: Sequência de pontos 1, 3, 4, 6, 7, 9 dispostos sobre uma circunferência.....	67

Figura 5.2: Emparelhamento entre os contornos “heart5” e “heart6”, sem optimização global das correspondências. Os contornos são definidos por 81 e 83 pontos, respectivamente. (A seta representada assinala uma região onde foram deixados pontos por corresponder.)	82
Figura 5.3: Emparelhamento dos contornos da Figura 5.2 com optimização global das correspondências usando o algoritmo <i>Simplex</i> para problemas de fluxo. (A seta representada assinala uma região de emparelhamentos cruzados.)	82
Figura 5.4: Parâmetros definidos na plataforma computacional <i>CMIS</i> para determinar a matriz de afinidade entre os pontos que definem dois contornos a emparelhar.....	84
Figura 5.5: Configuração definida por defeito na plataforma computacional <i>CMIS</i> para o algoritmo de optimização baseado no algoritmo <i>Simplex</i> para problemas de fluxo.....	84
Figura 5.6: Exemplo da janela de visualização do tempo de execução para cada passo de todo o processo emparelhamento, na plataforma computacional <i>CMIS</i> . (No interior do rectângulo representado a preto está a parte que tem mais interesse para este estudo.).....	84
Figura 5.7: Emparelhamento dos contornos da Figura 5.2 usando programação dinâmica.....	85
Figura 5.8: Dois contornos de um coração: à esquerda “heart1” e à direita “heart1a”, definidos por 28 pontos cada um.	85
Figura 5.9: Emparelhamento dos contornos da Figura 5.8: à esquerda usando os algoritmos de afectação clássicos e à direita usando o algoritmo de programação dinâmica.....	86
Figura 5.10: Dois contornos de um coração: à esquerda “heartA1” e à direita “heartA2”, definidos por 36 pontos cada um.	86
Figura 5.11: Emparelhamento dos contornos da Figura 5.10: à esquerda usando os algoritmos de afectação clássicos e à direita usando o algoritmo de programação dinâmica.....	86
Figura 5.12: Dois contornos de uma caixa torácica: à esquerda “rib1” e à direita “rib2”, definidos por 46 pontos cada um.	87
Figura 5.13: Emparelhamento dos contornos da Figura 5.12: à esquerda usando os algoritmos de afectação clássicos e à direita usando o algoritmo de programação dinâmica.....	87
Figura 5.14: Dois contornos obtidos de imagens de pedobarografia dinâmica: à esquerda “footBar1” e à direita “footBar6”, definidos por 51 e 58 pontos, respectivamente.	87
Figura 5.15: Emparelhamento dos contornos da Figura 5.14: à esquerda usando os algoritmos de afectação clássicos e à direita usando o algoritmo de programação dinâmica.....	87

Figura 5.16: Dois contornos de um avião: à esquerda “ <i>airplane2</i> ” e à direita “ <i>airplane12</i> ”, definidos por 57 e 86 pontos, respectivamente.	88
Figura 5.17: Emparelhamento dos contornos da Figura 5.16: à esquerda usando os algoritmos de afectação clássicos e à direita usando o algoritmo de programação dinâmica.	88
Figura 5.18: Dois contornos obtidos de imagens de pedobarografia dinâmica: à esquerda “ <i>foot2</i> ” e à direita “ <i>foot13</i> ”, definidos por 67 e 233 pontos, respectivamente.	88
Figura 5.19: Emparelhamento dos contornos da Figura 5.18: à esquerda usando os algoritmos de afectação clássicos e à direita usando o algoritmo de programação dinâmica. (A seta representada indica um emparelhamento cruzado.)	88
Figura 5.20: Dois contornos obtidos de imagens de pedobarografia dinâmica: à esquerda, “ <i>foot13</i> ” e à direita “ <i>foot14</i> ”, definidos por 233 e 253 pontos, respectivamente.	89
Figura 5.21: Emparelhamento dos contornos da Figura 5.20: à esquerda usando os algoritmos de afectação clássicos e à direita usando o algoritmo de programação dinâmica.	89
Figura 5.22: Dois contornos de um coração e artéria aorta: à esquerda “ <i>heartB3</i> ” e à direita “ <i>heartB2</i> ”, definidos por 389 e 139 pontos, respectivamente.	89
Figura 5.23: Emparelhamento dos contornos da Figura 5.22: à esquerda usando os algoritmos de afectação clássicos e à direita usando o algoritmo de programação dinâmica.	89
Figura 5.24: Dois contornos de um coração e artéria aorta: à esquerda “ <i>heartB3</i> ” e à direita “ <i>heartB4</i> ”, definidos por 389 e 417 pontos, respectivamente.	90
Figura 5.25: Emparelhamento dos contornos da Figura 5.24: à esquerda usando os algoritmos de afectação clássicos e à direita usando o algoritmo de programação dinâmica.	90
Figura 5.26: Exemplo de emparelhamento do tipo <i>um-para-vários</i> . O traço grosso representa as secções dos contornos após aplicação da transformação rígida. O traço mais fino contínuo representa os emparelhamentos do tipo <i>um-para-um</i> iniciais. O traço interrompido representada o novo emparelhamento.	96
Figura 6.1: Exemplos de duas secções de dois contornos para ilustrar a forma de determinação do ângulo associado a cada ponto.	103
Figura 6.2: Emparelhamento baseado em informação de curvatura dos contornos “ <i>heart1</i> ” e “ <i>heart1a</i> ”, definidos por 28 pontos cada um.	105

Figura 6.3: Emparelhamento baseado em informação de curvatura dos contornos “ <i>heart3</i> ” e “ <i>heart4</i> ”, definidos por 25 pontos cada um.	105
Figura 6.4: Emparelhamento baseado em informação de curvatura dos contornos “ <i>heartA1</i> ” e “ <i>heartA2</i> ”, definidos por 35 pontos cada um.	106
Figura 6.5: Emparelhamento baseado em informação de curvatura dos contornos “ <i>heart1</i> ” e “ <i>heart6</i> ”, definidos por 28 e 84 pontos, respectivamente.	106
Figura 6.6: Emparelhamento baseado em informação de curvatura dos contornos “ <i>heart1</i> ” e “ <i>heart5</i> ”, definidos por 28 e 81 pontos, respectivamente.	107
Figura 6.7: Emparelhamento baseado em informação de curvatura dos contornos “ <i>heart6</i> ” e “ <i>heart5</i> ”, definidos por 84 e 81 pontos, respectivamente.	107
Figura 6.8: Emparelhamento baseado em informação de curvatura dos contornos “ <i>rib1</i> ” e “ <i>rib2</i> ”, definidos por 46 pontos cada um.	108
Figura 6.9: Emparelhamento baseado em informação de curvatura dos contornos “ <i>airplane2</i> ” e “ <i>airplane12</i> ”, definidos por 57 e 86 pontos, respectivamente.	108
Figura 6.10: Emparelhamento baseado em informação de curvatura dos contornos “ <i>foot6</i> ” e “ <i>foot7</i> ”, definidos por 58 e 67 pontos, respectivamente.	109
Figura 6.11: Emparelhamento baseado em informação de curvatura dos contornos “ <i>heartB1</i> ” e “ <i>heartB2</i> ”, definidos por 135 e 139 pontos, respectivamente.	109
Figura 6.12: Emparelhamento baseado em informação de curvatura dos contornos “ <i>airplane1a</i> ” e “ <i>airplane2</i> ” definidos por 48 e 57 pontos, respectivamente.	111
Figura 6.13: Emparelhamento baseado em informação de curvatura dos contornos “ <i>airplane1b</i> ” e “ <i>airplane2</i> ” definidos por 43 e 57 pontos, respectivamente.	111
Figura 6.14: Emparelhamento baseado em informação de curvatura dos contornos “ <i>foot6</i> ” e “ <i>foot7a</i> ” definidos por 58 e 43 pontos, respectivamente.	112
Figura 6.15: Emparelhamento baseado na distância ao centróide dos contornos “ <i>heart3</i> ” e “ <i>heart4</i> ”, definidos por 25 pontos cada um.	116
Figura 6.16: Emparelhamento baseado na distância ao centróide dos contornos “ <i>heart1</i> ” e “ <i>heart1a</i> ”, definidos por 28 pontos cada um.	117
Figura 6.17: Emparelhamento baseado na distância ao centróide dos contornos “ <i>heart1</i> ” e “ <i>heart5</i> ”, definidos por 28 e 81 pontos, respectivamente.	117

Figura 6.18: Emparelhamento baseado na distância ao centróide dos contornos “ <i>heart1</i> ” e “ <i>heart6</i> ”, definidos por 28 e 84 pontos, respectivamente.....	118
Figura 6.19: Emparelhamento baseado na distância ao centróide dos contornos “ <i>heart1</i> ” e “ <i>heart9</i> ”, definidos por 28 e 243 pontos, respectivamente.....	118
Figura 6.20: Emparelhamento baseado na distância ao centróide dos contornos “ <i>heart6</i> ” e “ <i>heart5</i> ”, definidos por 84 e 81 pontos, respectivamente.....	119
Figura 6.21: Emparelhamento baseado na distância ao centróide dos contornos “ <i>heartA1</i> ” e “ <i>heartA2</i> ”, definidos por 35 pontos cada um. (A seta representada assinala um emparelhamento possivelmente a melhorar ou a excluir.).....	119
Figura 6.22: Emparelhamento baseado na distância ao centróide dos contornos “ <i>rib1</i> ” e “ <i>rib2</i> ”, definidos por 46 pontos cada um.	120
Figura 6.23: Emparelhamento baseado na distância ao centróide dos contornos “ <i>airplane2</i> ” e “ <i>airplane12</i> ”, definidos por 57 e 86 pontos, respectivamente. (A seta representada assinala um emparelhamento local a melhorar.).....	120
Figura 6.24: Emparelhamento baseado na distância ao centróide dos contornos “ <i>foot6</i> ” e “ <i>foot7</i> ”, definidos por 58 e 67 pontos, respectivamente.....	121
Figura 6.25: Emparelhamento baseado na distância ao centróide dos contornos “ <i>foot7</i> ” e “ <i>foot13</i> ”, definidos por 67 e 233 pontos, respectivamente. (A seta representada assinala uma região de emparelhamentos locais a melhorar.).....	121
Figura 6.26: Emparelhamento baseado na distância ao centróide dos contornos “ <i>heartB1</i> ” e “ <i>heartB2</i> ”, definidos por 135 e 139 pontos, respectivamente.....	122
Figura 6.27: Emparelhamento baseado na distância ao centróide dos contornos “ <i>airplane1a</i> ” e “ <i>airplane2</i> ” definidos por 48 e 57 pontos, respectivamente.....	123
Figura 6.28: Emparelhamento baseado na distância ao centróide dos contornos “ <i>airplane1b</i> ” e “ <i>airplane2</i> ” definidos por 43 e 57 pontos, respectivamente.....	124
Figura 6.29: Emparelhamento baseado na distância ao centróide dos contornos “ <i>foot6</i> ” e “ <i>foot7a</i> ” definidos por 58 e 43 pontos, respectivamente.....	124
Figuras 6.30: Emparelhamento baseado em informação de curvatura e distância ao centróide dos contornos “ <i>heart1</i> ” e “ <i>heart6</i> ”, definidos por 28 e 84 pontos, respectivamente.....	128

Figura 6.31: Emparelhamento baseado em informação de curvatura e distância ao centróide dos contornos “ <i>heartA1</i> ” e “ <i>heartA2</i> ”, definidos por 36 pontos cada um.	129
Figura 6.32: Emparelhamento baseado em informação de curvatura e distância ao centróide dos contornos “ <i>airplane2</i> ” e “ <i>airplane12</i> ”, definidos por 57 e 86 pontos, respectivamente.	129
Figura 6.33: Emparelhamento baseado em informação de curvatura e distância ao centróide dos contornos “ <i>foot7</i> ” e “ <i>foot13</i> ”, definidos por 67 e 233 pontos, respectivamente.	130
Figura 6.34: Emparelhamento baseado em informação de curvatura e distância ao centróide dos contornos “ <i>airplane1a</i> ” e “ <i>airplane2</i> ” definidos por 48 e 57 pontos, respectivamente.	131
Figura 6.35: Emparelhamento baseado em informação de curvatura e distância ao centróide dos contornos “ <i>airplane1b</i> ” e “ <i>airplane2</i> ” definidos por 43 e 57 pontos, respectivamente.	131
Figura 6.36: Emparelhamento baseado em informação de curvatura e distância ao centróide dos contornos “ <i>foot6</i> ” e “ <i>foot7a</i> ” definidos por 48 e 57 pontos, respectivamente.	132
Figura 6.37: Exemplo fictício de um emparelhamento de duas secções de dois contornos com ajuste local. O traço grosso indica os contornos e o traço fino os emparelhamentos. À esquerda os emparelhamentos iniciais e à direita os emparelhamentos após ajuste local.	135
Figura 6.38: Emparelhamento dos contornos “ <i>airplane2</i> ” e “ <i>airplane12</i> ”. À esquerda usando apenas informação de curvatura e à direita usando informação de curvatura com ajuste local.	136
Figura 6.39: Emparelhamento dos contornos “ <i>heart1</i> ” e “ <i>heart6</i> ”. À esquerda usando apenas informação de curvatura e à direita usando informação de curvatura com ajuste local.	136
Figura 6.40: Emparelhamento dos contornos “ <i>heart1</i> ” e “ <i>heart5</i> ”. À esquerda usando apenas informação da distância ao centróide, à direita usando informação da distância ao centróide com ajuste local.	136
Figura 6.41: Emparelhamento dos contornos “ <i>foot7</i> ” e “ <i>foot13</i> ”. À esquerda usando informação de curvatura e distância ao centróide, à direita usando informação de curvatura e distância ao centróide com ajuste local.	136
Figura 6.42: Emparelhamento do tipo <i>um-para-vários</i> dos contornos “ <i>heart1</i> ” e “ <i>heart6</i> ”, com base numa matriz de custos.	137
Figura 6.43: Emparelhamento do tipo <i>um-para-vários</i> dos contornos “ <i>heart1</i> ” e “ <i>heart9</i> ”, com base numa matriz de custos.	138
Figura 6.44: Emparelhamento do tipo <i>um-para-vários</i> dos contornos “ <i>airplane2</i> ” e “ <i>airplane12</i> ”, com base numa matriz de custos.	138

Figura 6.45: Emparelhamento do tipo <i>um-para-vários</i> dos contornos “ <i>foot7</i> ” e “ <i>foot13</i> ”, com base numa matriz de custos.....	138
Figura 6.46: Emparelhamento do tipo <i>um-para-vários</i> dos contornos “ <i>heart1</i> ” e “ <i>heart6</i> ”, com base na minimização da distância euclidiana entre os pontos a emparelhar.	140
Figura 6.47: Emparelhamento do tipo <i>um-para-vários</i> dos contornos “ <i>heart1</i> ” e “ <i>heart9</i> ”, com base na minimização da distância euclidiana entre os pontos a emparelhar.	140
Figura 6.48: Emparelhamento do tipo <i>um-para-vários</i> dos contornos “ <i>airplane2</i> ” e “ <i>airplane12</i> ”, com base na minimização da distância euclidiana entre os pontos a emparelhar.	140
Figura 6.49: Emparelhamento do tipo <i>um-para-vários</i> dos contornos “ <i>foot7</i> ” e “ <i>foot13</i> ”, com base na minimização da distância euclidiana entre os pontos a emparelhar.	141

CAPÍTULO I:
INTRODUÇÃO À DISSERTAÇÃO E SUA ESTRUTURA

1.1 Introdução

O reconhecimento de objectos representados em imagens é um dos problemas centrais em Visão Computacional. É uma tarefa desafiadora, principalmente devido ao elevado número de possíveis variações de projecção de um objecto numa imagem, como por exemplo, as mudanças de posição em relação à câmara e as condições iluminação, acrescentando-se ainda as deformações que o próprio objecto possa ter sofrido. Também a forma como é adquirida a imagem do objecto pode ter muita influência na capacidade de reconhecimento do mesmo, [Sebastian, 2004].

A Visão Computacional tem diversas aplicações na área do processamento e análise de imagem. A vigilância de áreas florestais para detecção de fogos; a monitorização e seguimento de pessoas, automóveis, navios; a identificação automática de pessoas; o controlo de bagagens em aeroportos; o controlo do tráfego rodoviário; a análise automática de imagens de tomografia axial computadorizada (TAC), de ressonância magnética nuclear (RMN), de ecografias; a identificação e inspecção de peças e o controlo de qualidade são alguns dos exemplos de aplicação.

1.2 Objectivos e abordagem seguida

Seguidamente, são enunciados os principais objectivos definidos inicialmente para esta Dissertação:

- a) Determinação de correspondências entre conjuntos de dados pontuais extraídos de contornos de objectos representados em imagens, usando modelação física ou geométrica. Os conjuntos de dados pontuais extraídos das imagens a emparelhar poderão ser constituídos por igual ou diferente número de pontos.
- b) Utilização de uma medida capaz de quantificar adequadamente a afinidade entre os dados pontuais extraídos dos objectos a emparelhar.
- c) Utilização de técnicas de optimização das correspondências com a inclusão de restrições adequadas; tais como, dados vizinhos deverão manter-se vizinhos, não deverão ser permitidos emparelhamentos cruzados, deve ser respeitada a coerência do movimento ao longo do tempo, etc.

Em relação à estratégia seguida:

- a) Em primeiro lugar foi feito um estudo bibliográfico sobre as metodologias usuais

para emparelhamento de objectos em Visão Computacional e sobre os últimos desenvolvimentos desta temática bem como das suas aplicações. Aqui, foram analisadas várias metodologias capazes de converter características dos objectos a emparelhar em coeficientes de um espaço funcional, dados pontuais, eixos médios ou outros que poderão ser posteriormente utilizados para determinar a correspondência entre os objectos representados em imagens.

- b) Seguidamente, foi efectuado um estudo sobre técnicas de optimização, em especial as relacionadas com problemas de afectação, usando programação linear, programação inteira, grafos bipartidos, programação dinâmica e *simulated annealing*.
- c) Atendendo que neste trabalho se considerou o emparelhamento de contornos definidos por dados pontuais, foi estudado o problema da ordenação dos pontos que definem um contorno, no sentido destes formarem uma sequência ordenada coerente.
- d) O passo seguinte foi o desenvolvimento de metodologias de emparelhamento de dados pontuais representativos de contornos de objectos usando técnicas de optimização, e desenvolvimento de uma metodologia de ordenação dos dados pontuais que definem os contornos a emparelhar.
- e) Com base nos emparelhamentos obtidos pelas metodologias de emparelhamento desenvolvidas, foram também desenvolvidas metodologias de determinação da transformação rígida que melhor alinha os contornos considerados.
- f) Implementação em ambiente *Microsoft Visual C++*, [Rodrigues, 2003], das diversas metodologias desenvolvidas. Para melhor visualizar os resultados dos emparelhamentos, recorreu-se a funções da biblioteca de domínio público *VTK – The Visualization Toolkit*, [Schroeder, 1999].
- g) Estudo da plataforma computacional de processamento e análise de imagem – *CMIS*, [Tavares, 2000a, 2002, 2003]. Implementação, na referida plataforma, do algoritmo de programação dinâmica desenvolvido para optimização do emparelhamento de contornos definidos por dados pontuais.
- h) Finalmente, foram realizados vários ensaios experimentais para validar as metodologias desenvolvidas e comparar os respectivos resultados. Concluiu-se o trabalho com a escrita desta Dissertação.

1.3 Principais contribuições

As principais contribuições alcançadas com esta Dissertação são as seguintes:

- a) Desenvolvimento e implementação de um algoritmo baseado na técnica de otimização de *simulated annealing* para ordenar os pontos extraídos do contorno exterior de um objecto representado numa imagem.
- b) Desenvolvimento e implementação de um algoritmo baseado na técnica de otimização de programação dinâmica para determinar a correspondência óptima entre os pontos que definem dois contornos ordenados, partindo de uma matriz de custos de emparelhamentos previamente calculada. Este algoritmo pode ser aplicado a contornos definidos por igual ou diferente número de pontos, alcançando sempre uma correspondência de custo mínimo que respeita a ordem dos pontos emparelhados, eliminando assim emparelhamentos cruzados.
- c) Desenvolvimento de três metodologias para determinar uma matriz de custos de emparelhamento entre os pontos que definem dois contornos ordenados: a primeira metodologia baseia-se em informação de curvatura, a segunda baseia-se na comparação da distância dos pontos do contorno ao seu centróide e a terceira é uma combinação das duas metodologias anteriores.
- d) Desenvolvimento e implementação de duas metodologias para corresponder os pontos excedentários, possibilitando assim, emparelhamentos do tipo *um-para-vários* ou *vários-para-um*, após o emparelhamento inicial do tipo *um-para-um* entre pontos de dois contornos definidos por diferente número de pontos.
- e) Desenvolvimento e implementação de uma metodologia para determinação da transformação rígida que melhor alinha dois contornos emparelhados.
- f) Desenvolvimento de uma formulação para ordenar os pontos de um contorno, baseada no algoritmo *simplex* para variáveis contínuas e a técnica de *branch-and-bound*. Apresentação de uma formulação baseada em programação inteira para resolver o mesmo problema de ordenação dos pontos de um contorno.
- g) Desenvolvimento de uma formulação baseada em programação linear inteira para determinar o emparelhamento óptimo entre dois contornos com base numa matriz de custos previamente determinada, respeitando a ordem dos pontos emparelhados. Assim, nesta formulação não são permitidos emparelhamentos cruzados.

Foram, ainda, realizadas várias tentativas de melhorar o algoritmo de emparelhamento de objectos representados por um conjunto de dados pontuais, utilizando modelação geométrica e análise modal da forma, proposto por Shapiro, [Shapiro, 1992a, 1992b], implementado em [Tavares, 2000a]. As várias metodologias ensaiadas, nomeadamente, para definir automaticamente e de forma eficaz o número de vectores próprios a utilizar e o respectivo sinal dos mesmos, não apresentaram resultados melhores do que os registados pela implementação actual existente na plataforma computacional *CMIS* já referida. Assim, estas não serão apresentadas.

1.4 Estrutura organizativa

Como já referido, o trabalho desenvolvido durante esta Dissertação enquadra-se no tema da Visão Computacional; mais precisamente, no emparelhamento e reconhecimento de objectos representados em imagens, usando técnicas de optimização. Estando o presente documento dividido em mais seis capítulos, seguindo a seguinte organização:

- **CAPÍTULO II: MÉTODOS DE EMPARELHAMENTO DE OBJECTOS REPRESENTADOS EM IMAGENS**

Neste capítulo são indicados alguns métodos de emparelhamento e reconhecimento de objectos representados em imagens. Estes foram divididos em três grupos fundamentais, conforme a representação da informação, sobre os quais são referidos os pressupostos base para cada grupo. Para cada tipo base de métodos, são apresentados alguns pontos fortes e menos fortes dos mesmos. Para cada método, são referidos alguns trabalhos importantes e algumas suas aplicações práticas identificadas.

- **CAPÍTULO III: ORDENAÇÃO DOS PONTOS QUE DEFINEM UM CONTORNO**

Com o intuito de garantir uma ordem coerente dos pontos que definem os contornos a emparelhar, neste capítulo são apresentadas três metodologias de ordenação dos pontos que definem o contorno exterior de um objecto representado numa imagem. A primeira baseia-se numa formulação adaptada para o algoritmo *simplex* conjugado com a técnica de *branch-and-bound*. A segunda baseia-se num modelo de programação inteira. A terceira tem por base a técnica de optimização *simulated annealing*. Esta última metodologia foi implementada e testada neste trabalho.

- CAPÍTULO IV: DETERMINAÇÃO DA TRANSFORMAÇÃO RÍGIDA QUE MELHOR ALINHA DOIS CONTORNOS

Neste capítulo é apresentada uma metodologia para determinar a transformação rígida que melhor alinha dois contornos ordenados, usando informação relativa à ordem dos pontos que definem os contornos a alinhar. Vários ensaios experimentais realizados usando esta metodologia são apresentados e analisados neste capítulo; no entanto, no sexto capítulo, aquando da apresentação de metodologias de determinação dos referidos emparelhamentos, são apresentados mais resultados experimentais obtidos, nomeadamente para analisar o ângulo de rotação obtido.

- CAPÍTULO V: DETERMINAÇÃO DE UM EMPARELHAMENTO ÓPTIMO ENTRE DOIS CONTORNOS COM BASE NUMA MATRIZ DE CUSTOS

Neste capítulo são desenvolvidas duas metodologias para determinar o emparelhamento óptimo entre os pontos que definem dois contornos ordenados com base numa matriz de custos previamente determinada. Estas duas metodologias incluem a restrição de respeito pela ordem dos pontos a emparelhar. A primeira utiliza uma formulação de programação linear inteira. A segunda metodologia baseia-se na técnica de optimização de programação dinâmica. Apenas esta última foi implementada neste trabalho.

Para validar e comparar o algoritmo de optimização das correspondências, baseado na técnica de programação dinâmica, com algoritmos de optimização clássicos baseados em modelos de afectação, incluídos na plataforma computacional *CMIS*; o algoritmo baseado em programação dinâmica foi implementado na referida plataforma, apresentando-se os resultados neste capítulo. São ainda apresentadas duas metodologias para realizar emparelhamentos do tipo *um-para-vários* (ou *vários-para-um*), partindo de um emparelhamento inicial do tipo *um-para-um*.

- CAPÍTULO VI: EMPARELHAMENTOS BASEADOS EM INFORMAÇÃO DE CURVATURA E DISTÂNCIA AO CENTRÓIDE

Neste capítulo são apresentadas três metodologias para obter uma matriz de custos de emparelhamento entre dois contornos definidos por igual ou diferente número de pontos. A primeira metodologia baseia-se em informação de curvatura, a segunda metodologia baseia-se na comparação das distâncias dos pontos dos contornos aos respectivos centróides e a terceira metodologia utiliza em simultâneo a informação de curvatura e a informação da distância dos pontos aos respectivos centróides. Ainda neste capítulo, são

apresentados resultados de emparelhamentos do tipo *um-para-vários* com base em dois algoritmos desenvolvidos e apresentados no capítulo anterior.

- **CAPÍTULO VII: CONCLUSÕES FINAIS E PERSPECTIVAS DE TRABALHO FUTURO**

Neste capítulo são apresentadas algumas conclusões finais sobre o trabalho desenvolvido e resultados experimentais obtidos. Também são indicadas algumas perspectivas de desenvolvimento futuro, as quais poderão ser consideradas no prosseguimento do trabalho realizado.

CAPÍTULO II:
MÉTODOS DE EMPARELHAMENTO DE OBJECTOS
REPRESENTADOS EM IMAGENS

2.1 Introdução

O reconhecimento de objectos representados em imagens é um problema central em Visão Computacional. Este problema tem-se revelado de difícil resolução devido, por exemplo, ao elevado número de possíveis variações de projecção de um objecto numa imagem, às deformações que o próprio objecto possa sofrer e a situações de oclusão, [Sebastian, 2004].

Muito relacionado com o problema de reconhecimento de objectos representados em imagens está o problema de identificar elementos correspondentes entre objectos representados em imagens, geralmente definidos por conjuntos de pontos. Esta identificação torna-se mais difícil quando, regra geral, essas relações não são bijectivas: elementos de um objecto podem estar representados numa imagem e não estar representados noutra imagem. Este problema e a falta de uma teoria unificadora tornam o problema da correspondência entre elementos de objectos representados em imagens num problema de difícil resolução, que tem levado a um aparecimento de vários métodos para o resolver. Este problema pode ser considerado de diferentes modos, dependendo da representação da informação: desde baixo nível para alto nível. Assim, esses métodos podem ser divididos em correspondência de sinal (baixo nível), correspondência de características e correspondência estrutural (alto nível), [Starink, 1995].

Na correspondência de sinal, as imagens são tratadas como sinais 2D consistindo na distribuição de uma escala de cinzento ou de cor. Estes métodos fazem uso das técnicas bem desenvolvidas de processamento de sinal, como por exemplo em [Daugman, 2003].

Nos métodos de correspondência das características são utilizados elementos da imagem, tais como pontos e contornos, [Shapiro, 1992a, 1992b], [Starink, 1995], [Sclaroff, 1995], [Belongie, 2002]. Estes são detectados em ambas as imagens e depois correspondidos.

Nos métodos de correspondência estrutural, também referido como método de correspondência relacional, as estruturas e as suas relações com estruturas vizinhas são descritas por representações simbólicas, geralmente organizadas em grafos. Em grande parte destes últimos métodos é utilizado o eixo médio da forma, extraído da silhueta da mesma, [Blum, 1967], [Ogniewicz, 1995], [Zhu, 1996], [Chung, 2000], [Sebastian, 2004].

Na correspondência de formas, os métodos mais comuns para aplicações de reconhecimento e emparelhamento são os dois últimos anteriormente referidos. Assim, as técnicas que representam a forma por um conjunto de pontos, por uma linha de contorno

ou pelo seu eixo médio têm dominado o campo de investigação nesta área, [Sebastian, 2004].

Cada um destes tipos de métodos será analisado em mais pormenor nas secções seguintes, fazendo-se a divisão sugerida: métodos baseados no sinal da imagem, métodos baseados em características da forma e métodos baseados na estrutura da forma. Estudos sobre o estado da arte relativo a esta área da análise de objectos em imagens podem ser encontrados, por exemplo, em [Veltkamp, 2000] e [Zhang, 2004].

Antes de se passar para o estudo dos métodos de reconhecimento de objectos ou formas representas em imagem, importa realçar alguns pressupostos que estão por detrás da definição de uma medida de similaridade entre duas formas. Quando se deseja efectuar o reconhecimento de uma forma, o sistema computacional envolvido deve ser capaz de devolver uma medida de similaridade ou de diferença que indique o quanto a forma em estudo é idêntica à forma modelo, [Arkin, 1991]. Considerando o valor 0 (zero) como indicador de mínima diferença (geralmente referida como “distância mínima” entre as formas, por analogia com a métrica euclidiana), esta medida deve satisfazer uma série de propriedades:

1. Deve ser uma métrica:
 - i. Se duas formas são iguais, a sua medida de diferença deve ser nula.
 - ii. A medida de diferença entre a forma A e a forma B tem de ser igual à medida de diferença entre a forma B e a forma A.
 - iii. A medida de diferença entre a forma A e a forma C tem que ser menor ou igual do que a soma da medida de diferença entre as formas A e B com a medida de diferença entre as formas B e C.
2. Deve ser invariante a translações, rotações e mudanças de escala.
3. Deve ser computável em tempo útil.
4. Deve fazer a correspondência entre as formas de acordo com a intuição humana.

2.2 Métodos baseados no sinal

Em [Belongie, 2002], os métodos baseados no sinal são divididos em duas categorias básicas: os que procuram quantificar a similaridade entre as imagens construindo as correspondências e os que quantificam a similaridade sem construírem explicitamente as correspondências.

Belongie, na primeira categoria, colocou os métodos que explicitamente procuram as correspondências usando os valores da escala de cinzentos. Como métrica, pode ser utilizada uma medida de correlação dos níveis de cinzento, [Starink, 1995].

Neste grupo de métodos, Belongie destacou o trabalho de Yuille, [Yuille, 1991], o qual apresentou um método muito flexível em que invariâncias para uma certa categoria de transformações podem ser construídas como modelo de medição de semelhança, mas sofrem da necessidade de modelos predefinidos pelo homem e sensibilidade aos parâmetros de inicialização.

Em [Lades, 1993] é usado um emparelhamento elástico gráfico, num método que envolve ambas as características, geométricas e fotométricas, como forma de um descritor local baseado na derivada *Gaussiana*.

Em alguns métodos são extraídos *pixels* característicos numa imagem e depois procuram-se os correspondentes na outra imagem. Nestes métodos podem-se enquadrar os métodos que utilizam o gradiente da variação da intensidade de brilho ou de cor da imagem. Para cada imagem, é construído um campo de vectores que representa o gradiente da imagem, sendo depois medida a similaridade entre as imagens ou correspondidos os pontos característicos das mesmas através da comparação dos respectivos campos de vectores gradiente, [Scharstein, 1994].

O problema de reconhecimento de faces em imagens continua a ser encarado por grande parte dos investigadores como um problema de emparelhamento, pois em grande parte das aplicações existentes não estão disponíveis imagens estéreo da face da mesma pessoa. Nesta área têm ocorrido grandes progressos através do desenvolvimento de métodos de segmentação, extracção de características e reconhecimento de faces em imagens de intensidade, [Zhao, 2003]. Em [Vetter, 1997] são comparados os valores da intensidade, atendendo-se primeiro à deformação de uma imagem em relação à outra usando um campo denso de correspondências.

A segunda categoria de métodos incluiu os que constroem as classificações sem explicitamente encontrarem as correspondências. Uma parte destes métodos recorre a algoritmos capazes de aprender através da aquisição de invariantes em muitos exemplos de análise, isto é, constroem uma base de dados de características invariantes de uma imagem através da comparação de várias imagens modelo. Outros algoritmos utilizam a decomposição espectral da imagem para depois a poderem comparar com outras, [Zhang, 2004]. Exemplos destes algoritmos serão apresentados mais à frente.

Existem várias técnicas que transformam a informação da cor de uma imagem numa

informação de frequência e amplitude da mesma, sendo depois esta informação usada para comparar imagens. Embora estes métodos não detectem explicitamente a forma, eles capturam as regiões fronteira da mesma, pois estas são frequentemente regiões de transição. A transformada de *Fourier* é um desses métodos, pois consegue aproximar um sinal periódico qualquer por uma série de funções trigonométricas (senos e co-senos), sendo apenas necessário uma série de coeficientes para se poder analisar o sinal, reconstruí-lo ou mesmo compará-lo com um outro sinal. No entanto, na reconstrução do sinal da imagem, podem surgir problemas, pois não é possível recuperar a fase das diversas componentes, [Soares, 1997], [Zhang, 2004].

Outro método usual e ainda mais poderoso para decompor um sinal (ou função) é baseado na teoria de onduletas (*wavelet*). Existem vários tipos de onduletas, todas capazes de decompor e descrever outras funções, de forma que essas funções decompostas podem ser estudadas em frequência e tempo (ou fase). Além disso, é possível reconstruir integralmente o sinal original, pois a informação da fase deste não foi perdida, [Soares, 1997].

Tal como a transformada de *Fourier*, a transformada de onduleta tem as variantes discreta e contínua. Graças à sua capacidade de decompor as funções tanto no domínio da frequência quanto no domínio do tempo, estas classes de funções são ferramentas eficazes para a análise de sinais e compressão de dados. Assim, a comparação dos coeficientes de frequência e tempo das onduletas permitem medir a similaridade entre dois sinais e por consequência, permitem comparar imagens quando estas estão representadas por um sinal eléctrico, por exemplo, um sinal que represente os níveis de cinzento ao longo da imagem.

Os métodos que fazem uso da análise espectral do sinal para medir a similaridade entre duas imagens não têm, portanto, em consideração a forma representada na imagem, mas sim toda a imagem, fazendo com que aspectos da mesma sem relevo para o reconhecimento ou comparação de formas possam influenciar fortemente os resultados. No entanto, há muitas aplicações em Visão Computacional nas quais a forma representada na imagem é irrelevante, interessando apenas a variação de atributos da imagem como intensidade, cor e textura, [Zhang, 2004]. Frequentemente, os métodos de reconhecimento de pessoas através da íris baseiam-se nesta técnica da análise espectral do sinal. Inicialmente, é utilizado um algoritmo para identificar e seleccionar apenas a região da íris no olho humano. Seguidamente, o sinal representativo da mesma é decomposto usando um tipo de transformada de onduletas conhecida por *Onduletas de Gabor*. Finalmente, os valores de frequência e tempo obtidos são comparados com os registados numa base de

dados, [Daugman, 2003].

O método de reconhecimento de pessoas através da íris é altamente fiável, pois a formação da textura da mesma é um processo aleatório, não dependendo de características genéticas. Além disso, como o olho é um órgão muito protegido, não sofre as consequências da idade de uma forma tão acentuada como a face, por exemplo, mantendo-se quase inalterável ao longo da vida. Um método de reconhecimento e codificação da íris foi apresentado em 1993 por Daugman. Com base neste, são várias as instituições, desde grandes empresas multinacionais a aeroportos, que o utilizam com enorme sucesso. A probabilidade de um terço dos códigos de fase de duas íris da mesma pessoa não corresponderem é de um em cerca de dezasseis milhões, [Daugman, 2003]. Deste modo, este método é praticamente infalível.

Em [Keysers, 2007], para corresponder uma forma representada numa imagem com uma forma modelo representada noutra imagem, as imagens são subdivididas. Em cada parte resultante é realizada uma PCA (*Principal Component Analysis*), em que as componentes principais são extraídas usando um detector de pontos de interesse baseado em onduletas. Depois, é criado um histograma das partes para representar a imagem total. Em relação às técnicas que trabalham com a imagem na totalidade, esta última tem a vantagem de poder resolver de forma mais eficaz o problema da oclusão de partes de um objecto numa imagem.

2.3 Métodos baseados em características da forma

Como já referido, nestes métodos são utilizadas características da forma, tais como pontos característicos, linhas de contorno, pontos de elevada curvatura, etc. Para tal, em primeiro lugar é necessário que sejam extraídas das imagens as características pretendidas. Existem muitos algoritmos para o efeito, grande parte dos quais baseados em descritores de orlas que usam filtros *Gaussianos*.

Em [Zhang, 2004], os métodos de descrever as formas foram divididos em duas classes fundamentais: baseados apenas no contorno e baseados em toda a região (contorno e respectivo interior). Esta classificação deriva da região da forma de onde são extraídas as características que a descrevem. Tanto o primeiro como o segundo métodos podem ser subdivididos em estruturais e globais, conforme a forma seja representada por partes (ou secções) da mesma ou seja representada globalmente, respectivamente. Conforme o

descriptor de forma utilizado, o método de emparelhamento ou de reconhecimento tem de ser adaptado à situação em questão.

As técnicas baseadas no contorno recolhem informação apenas deste, podendo o mesmo ser tratado como uma linha contínua, como uma série de pontos ou de segmentos de recta. Para quantificar a semelhança entre duas formas, podem ser utilizadas várias técnicas. Em quase todas elas, em primeiro lugar são extraídas da mesma as características que serão utilizadas para as comparar, por exemplo, distribuição dos pontos, informação de curvatura, coeficientes (amplitude, frequência ou fase) de um descriptor espectral baseado em transformadas de *Fourier* ou de onduletas, etc. Depois é necessário definir uma função para quantificar a similaridade entre as características determinadas, [Veltkamp, 2000].

Para comparar as formas com base em toda a região, contorno e interior, podem ser utilizados, por exemplo, descritores baseados na área, uma matriz de forma (por exemplo: emparelhamento modal, [Shapiro, 1992a, 1992b], [Sclaroff, 1995]) e eixos médios, [Zhang, 2004].

2.3.1 Formas representadas por conjuntos de pontos não ordenados

Frequentemente, quando duas formas são representadas por conjuntos de pontos característicos, não necessariamente representativos dos seus contornos, após extracção das características a utilizar, para quantificar a similaridade das mesmas, é inicialmente realizado o emparelhamento dessas características por um algoritmo de afectação, sendo que a similaridade resulta da soma dos custos de todos os emparelhamentos determinados. Geralmente, estes métodos apresentam o problema de basear as correspondências em aspectos locais, não garantindo deste modo a coerência da forma. Para determinar os emparelhamentos, são frequentemente utilizados algoritmos baseados em modelos de programação linear, grafos bipartidos, programação dinâmica, etc. Em [Maciel, 2001], por exemplo, é utilizada programação côncava para determinar o emparelhamento óptimo nos exemplos em estudo, aproveitando desse modo a estrutura esparsa especial da matriz de correlação.

O trabalho de Sclaroff e Pentland, [Sclaroff, 1995], é representativo dos métodos baseados em vectores próprios ou emparelhamento modal. Nestes métodos, os pontos amostrados são vistos como “molas elásticas” e as correspondências são encontradas por comparação dos modos de vibração.

Em [Tavares, 2000a] foram apresentados e implementados alguns métodos de

emparelhamento de objectos representados em imagens e definidos por conjuntos de pontos não estruturados; nomeadamente, baseados no princípio da distância mínima, modelação geométrica e análise modal, e modelação física e análise modal. Muito simplesmente, em todos estes métodos, parte-se dos conjuntos de dados pontuais que definem ambos os objectos a emparelhar e depois determina-se uma medida de afinidade entre os pontos que caracterizam uma imagem com os pontos da outra imagem. Com base nessa medida, determinaram-se os emparelhamentos. Todas as metodologias implementadas em [Tavares, 2000a] deram bons resultados. Nessas implementações, um ponto de um objecto só emparelhava com um ponto de outro objecto quando ambos “reclamavam” um pelo outro. Como consequência, acontecia frequentemente que alguns pontos não eram emparelhados.

Para aumentar o número de pontos emparelhados, em [Bastos, 2003] são implementados três algoritmos de optimização global das correspondências, baseados no método *Húngaro*, *Simplex* para problemas de fluxo e *LAPm*. Estes partem da matriz de afinidade obtida pelas metodologias baseadas em modelação geométrica e física implementadas em [Tavares, 2000a] na plataforma computacional de desenvolvimento e ensaio *CMIS*, [Tavares, 2000a, 2002, 2003], procurando depois o emparelhamento global cuja soma dos custos fosse mínima.

Todos os três algoritmos de optimização implementados na referida plataforma computacional *CMIS* obtêm bons resultados no sentido em que aumentam o número de bons emparelhamentos em relação à anterior metodologia, que tinha um carácter puramente local. Da análise de toda esta metodologia global – determinação da matriz de afinidade e determinação do emparelhamento global óptimo em termos de custos – constata-se que a mesma apresenta algumas fraquezas, especialmente quando aplicada ao emparelhamento de contornos. Esses aspectos insatisfatórios são, por exemplo:

- a) A ordem dos pontos nos contornos nem sempre é respeitada, isto é, aparecem por vezes emparelhamentos cruzados.
- b) As metodologias de determinação das matrizes de correlação ou afinidade são muito sensíveis à escolha do número de vectores próprios a utilizar. Em alguns contornos obtêm-se bons emparelhamentos com a utilização de 25% dos vectores próprios, por exemplo; enquanto noutros não se obtêm bons emparelhamentos para os mesmos 25%. Nos testes efectuados, não foi encontrada uma forma automática de definir o número de modos a utilizar que garantisse sempre bons resultados.

- c) Na construção das matrizes de afinidade, há por vezes ambiguidade na escolha do sinal dos vectores próprios devido a simetrias das formas envolvidas. O critério utilizado, embora tenha revelado bons resultados, por vezes não é capaz de escolher o sinal adequado, provocando assim emparelhamentos sem sentido (o objecto dobra-se sobre si mesmo).
- d) O tempo dispendido para determinar o emparelhamento dos pontos é muito elevado. Tal deve-se à necessidade de resolver inicialmente um problema de determinação de valores e vectores próprios de matrizes de elevada dimensão, ao algoritmo de escolha do sinal dos vectores próprios, ao cálculo da matriz de afinidade e ainda aos algoritmos de optimização utilizados.

Para minimizar o problema da incoerência de emparelhamentos entre formas, foram desenvolvidas metodologias que têm em atenção a forma global, sendo capturada a localização relativa dos pontos. Um exemplo destes métodos é apresentado em [Carcassoni, 2003]. Nesse trabalho, utiliza-se a metodologia proposta por Shapiro, [Shapiro, 1992a, 1992b], sendo depois o emparelhamento hierarquizado. Em primeiro lugar, utilizando métodos probabilísticos e os valores dos modos obtidos pelo método de Shapiro, dividem-se os pontos em *clusters*, isto é, em subconjuntos de pontos com afinidade entre eles. Deste modo, obtêm-se subestruturas da forma. Seguidamente, esses *clusters* são correspondidos, sendo que, só depois, os pontos dos *clusters* emparelhados são correspondidos entre si.

Belongie, em [Belongie, 2002], utilizou um descritor de forma em que para cada ponto eram determinados todos os vectores que o unem aos restantes pontos da forma. Deste modo, associado a cada ponto há informação adicional além da sua posição num determinado referencial; nomeadamente, informação da forma global e da posição do ponto na respectiva forma. Definiu esses vectores em coordenadas polares, isto é, raio (ou comprimento) e amplitude do ângulo relativamente a um sistema de eixos previamente definido. Para cada ponto, foi então construído um histograma com base no ângulo e no logaritmo do raio. Para determinar o custo de emparelhamento entre dois pontos, utilizou a distribuição de χ^2 , sendo que posteriormente o emparelhamento foi realizado no sentido de minimizar a soma dos custos dos pontos emparelhados.

2.3.2 Formas representados pelo contorno

A grande pesquisa em similaridade de formas tem sido feita usando a silhueta dos objectos representados em imagens (contorno exterior). Desde que os objectos não apresentem contornos interiores, o contorno associado ao objecto representado na imagem é convenientemente representado por uma curva fechada.

As técnicas baseadas no contorno recolhem informação apenas deste, podendo o mesmo ser tratado como uma linha contínua ou como uma série de pontos e segmentos de recta. Para quantificar a semelhança entre duas formas, podem ser utilizadas várias técnicas. Em quase todas elas, em primeiro lugar são extraídas da mesma as características que serão utilizadas para as comparar ao alinhar; por exemplo, distribuição dos pontos, informação de curvatura, coeficientes (amplitude, frequência ou fase) de um descritor espectral baseado nas transformadas de *Fourier* ou de onduletas, etc. Depois é necessário definir uma função para quantificar a similaridade entre as características determinadas, [Zhang, 2004].

No domínio discreto, o problema de minimização é frequentemente transformado num problema de emparelhamento em que atributos como, por exemplo, curvatura (ou ângulo de viragem), orientação absoluta e coeficientes de uma transformada são considerados. Estando determinada uma matriz de similaridade entre os pontos dos contornos, o problema de determinar o melhor emparelhamento global pode ser encarado com um problema de afectação ou problema de emparelhamento de grafos bipartidos, tal como para corresponder pontos característicos de imagens, onde não está definida qualquer ordem na sequência de pontos. Assim, a medida de similaridade entre formas é frequentemente considerada como a soma dos custos de emparelhamento das características extraídas, [Bastos, 2003, 2006], [Scott, 2006].

Em [Scott, 2006] são proposta soluções baseadas em programação dinâmica para determinar os emparelhamentos, com base numa matriz de custos previamente determinada, considerando a restrição de respeito pela ordem dos pontos.

Outras técnicas comuns para determinar a similaridade entre dois contornos discretos são a distância de Hausdorff¹ e o cálculo da soma das distâncias entre os pontos correspondentes dos dois contornos. Neste caso é necessário alinhar previamente os dois contornos, [Zhang, 2004].

As metodologias de emparelhamento desenvolvidas ao longo desta Dissertação enquadram-se dentro da correspondência de objectos representados por contornos

discretos. Para determinar uma matriz de custos é utilizada informação de curvatura e informação da distância dos pontos dos contornos ao seu centróide. Para determinar o melhor emparelhamento, é utilizada uma filosofia de procura do melhor emparelhamento global que respeita a ordem dos pontos, sendo implementado um algoritmo baseado em programação dinâmica para esse efeito. Estas metodologias desenvolvidas neste trabalho utilizam contornos definidos por conjuntos de pontos ordenados. Assim, não estaria totalmente errado classificar estas mesmas metodologias como estruturais.

No domínio contínuo, os contornos são definidos frequentemente por funções implícitas ou paramétricas. Nestas situações, para determinar a similaridade das mesmas, pode-se recorrer à determinação da melhor transformação, dentro do grupo de transformações admissíveis, que minimiza, por exemplo, a área contida entre os dois contornos. No caso das curvas estarem alinhadas e parametrizadas, a distância de Fréchet² é também recorrentemente utilizada para determinar a similaridade entre as duas formas, [Veltkamp, 2000]. Outro método também utilizado é o cálculo da energia elástica necessária para transformar uma curva na outra, [Manay, 2006].

Em [Rosenhahn, 2006], a similaridade entre dois contornos é calculada do seguinte modo: Sejam Ω uma região do plano que contenha o contorno e $\Phi : \Omega \subset R^2 \rightarrow R$, uma função. Sendo C um contorno, a função Φ pode ser definida da seguinte modo:

¹**Distância de Hausdorff:** Sejam A e B dois conjuntos de pontos que definem os contornos de duas formas. A distância de Hausdorff do conjunto A para o conjunto B define-se como *o máximo da distância entre os pontos do conjunto A e os pontos mais próximo do conjunto B* . Em notação:

$$h(A, B) = \max_{a \in A} \left(\min_{b \in B} \|\vec{ab}\| \right).$$

Como a distância de Hausdorff não é simétrica, isto é, $h(A, B)$ e $h(B, A)$ não são necessariamente iguais, define-se a distância de Hausdorff entre os conjuntos A e B do seguinte modo:

$$H(A, B) = \max(h(A, B), h(B, A)).$$

²**Distância de Fréchet:** Sejam A e B duas curvas ou contornos de duas formas, $\alpha(t)$ uma parametrização de A e $\beta(t)$ uma parametrização de B ; onde $A(\alpha(t))$ representa um ponto de A , e $B(\beta(t))$ representa um ponto de B para cada $t \in [0; 1]$. A distância de Fréchet representa *o mínimo, para todas as parametrizações, da máxima distância entre os pontos das curvas*. Em notação:

$$F(A, B) = \min_{\alpha(t), \beta(t)} \left\{ \max_t \|\vec{A(\alpha(t))B(\beta(t))}\| \right\}.$$

$$\Phi(x) = \begin{cases} d(x, C), & \text{se } x \text{ está no interior do contorno} \\ -d(x, C), & \text{se } x \text{ está no exterior do contorno} \\ 0, & \text{se } x \text{ pertence ao contorno} \end{cases} \quad (2.1)$$

onde d representa, por exemplo, a distância euclidiana.

Da representação gráfica 3D função Φ anteriormente definida pode-se observar claramente o eixo médio da forma. Assim, esta função também pode ser utilizada como descritor deste tipo de estrutura. A título de exemplo, a Figura 2.1 representa o gráfico da função Φ em relação ao contorno de um rectângulo representado a vermelho. A projecção do traço representado a preto na referida figura sobre o plano xOy corresponde ao eixo médio do rectângulo.

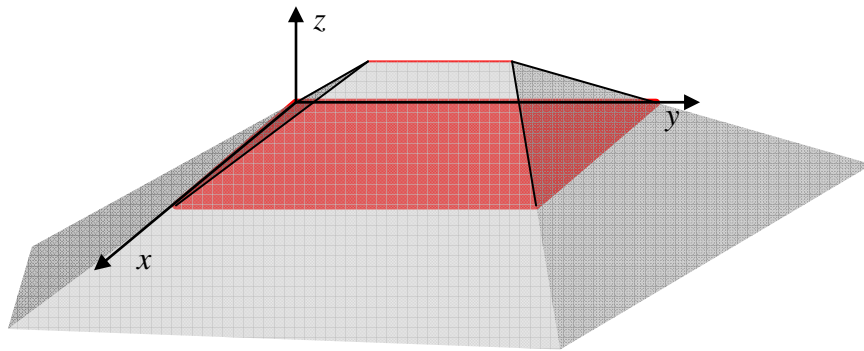


Figura 2.1: Representação tridimensional do gráfico da função Φ definida pela expressão (2.1), relativamente ao contorno do rectângulo vermelho.

Dados dois contornos, C_1 e C_2 , e as respectivas funções, Φ_1 e Φ_2 , definidas como em (2.1), o quadrado da diferença entre os dois contornos, dif^2 , pode ser determinado do seguinte modo:

$$dif^2(C_1, C_2) = \int_{\Omega} (\Phi_1(x) - \Phi_2(x))^2 dx.$$

Fazendo, por exemplo, Φ_2 depender de um grupo transformações consideradas válidas, o emparelhamento óptimo será obtido quando o valor da diferença entre os dois contornos for mínimo, podendo essa diferença mínima ser usada como medida de similaridade.

Em [Bala, 2004], o reconhecimento de objectos é efectuado através da determinação dos coeficientes de uma transformada de onduleta dos pontos que definem o contorno da imagem. Esta transformada e a transformada de *Fourier* tanto podem ser utilizadas no

meio contínuo como no meio discreto.

Em jeito de conclusão, pode-se dizer que os métodos baseados em contornos definidos por pontos, sofrem frequentemente de pelo menos um dos seguintes inconvenientes: assimetria no tratamento das formas; sensibilidade à amostragem; falta de invariância relativamente ao escalamento ou rotação. Os métodos que trabalham com o contorno no domínio contínuo são mais difíceis de aplicar; no entanto, não sofrem de sensibilidade à amostragem. Tanto uns como outros, como trabalham com a totalidade da moldura exterior da forma, são sensíveis articulações e oclusões de partes, [Sebastian, 2004].

Como comparação entre os métodos baseados em toda a forma e os métodos baseados apenas no contorno da mesma, pode-se dizer que as silhuetas (contornos exteriores) são limitadas como descritores para a generalidade dos objectos, pois ignoram contornos interiores e são difíceis de extrair de imagens reais; no entanto, em grande parte das aplicações revelam-se muito eficazes. Os métodos que tratam a forma como um conjunto de pontos 2D (não ordenados), de um modo geral, não preservam a coerência da forma, no entanto podem utilizar todos os pontos relevantes da imagem e a sua extracção desta é mais simples.

2.4 Métodos baseados na estrutura da forma

As formas têm também sido representadas pela sua estrutura. De entre alguns métodos de obtenção da estrutura de uma forma, os que têm merecido mais destaque pelos investigadores são os que usam o eixo médio da mesma, também conhecido por esqueleto, e uma variante deste conhecida por *shock-graph*. Na Figura 2.2 podem-se observar dois exemplos de eixos médios de duas formas.

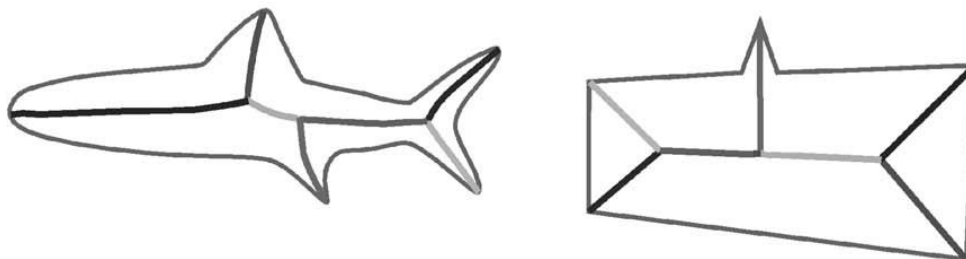


Figura 2.2: Exemplo de duas formas e respectivos eixos médios no seu interior.

(Retirados de [Sebastian, 2004].)

Esta representação tem a vantagem de transformar uma forma, geralmente caracterizada como uma estrutura 2D numa outra mais simples do tipo 1D. Um dos principais inconvenientes, é o facto de formas muito diferentes poderem apresentar eixos médios com estruturas topológicas iguais.

O eixo médio de uma forma pode ser definido de diferentes modos, [Blum, 1967], [Ogniewicz, 1995]:

- Lugar geométrico dos pontos interiores da forma equidistantes a dois pontos do seu contorno;
- Pela analogia da “queima da erva”, introduzida por Blum: “*O contorno é todo incendiado ao mesmo tempo, sendo que as zonas interiores do mesmo onde pelo menos duas frentes do fogo se encontram definem o respectivo eixo médio*”;
- Resultado do “emagrecimento” ou estreitamento máximo da forma de modo que esta mantenha ainda a mesma estrutura topológica;
- Lugar geométrico dos centros dos círculos máximos contidos na forma.

Através da definição do eixo médio como o lugar geométrico dos pontos equidistantes a dois pontos do contorno, podem ser induzidos diversos métodos como, por exemplo, o seguinte:

1. Divide-se a forma numa grelha, sendo calculadas as distâncias de cada ponto interior da forma ao limite da mesma (contorno);
2. Utilizando um algoritmo de procura e seguimento, define-se o eixo médio como a linha que une os máximos locais, com as regiões de diminuição mais lenta da amplitude.

Na Figura 2.3 está uma representação de um eixo médio de uma forma, determinado com recurso à métrica induzida pela norma do máximo, isto é, dados dois pontos $A(a_1, a_2)$ e $B(b_1, b_2)$ do plano, a distância entre esses pontos é determinada por:

$$D_{\max}(A, B) = \max\{|a_1 - b_1|, |a_2 - b_2|\}.$$

Nesta implementação só consideramos vizinhas as células que têm um lado em comum. Assim, na Figura 2.3, todas a células assinaladas a cinzento-escuro são máximos locais.

Jae-Moon Chung e Noboru Ohnishi, em [Chung, 2000], definiram o eixo médio como

o lugar geométrico de todos os centros dos discos máximos contidos no interior do contorno da forma e tangentes em pelo menos dois pontos do contorno, Figura 2.4. Para corresponder as formas, para cada ponto do contorno, determinaram todos os vectores que o relacionam com os restantes pontos da forma. Cada um desses vectores é formado por três componentes: a amplitude do ângulo definido por esses dois pontos e cujo vértice é o centro do respectivo círculo máximo; o raio do círculo máximo e o quociente entre o comprimento mínimo do contorno entre os dois pontos e o comprimento total do contorno. A correspondência é determinada através da comparação dos respectivos vectores de ambas as formas no sentido de minimizar a soma das diferenças entre eles, segundo uma métrica em que as três componentes dos vectores têm pesos diferenciados.

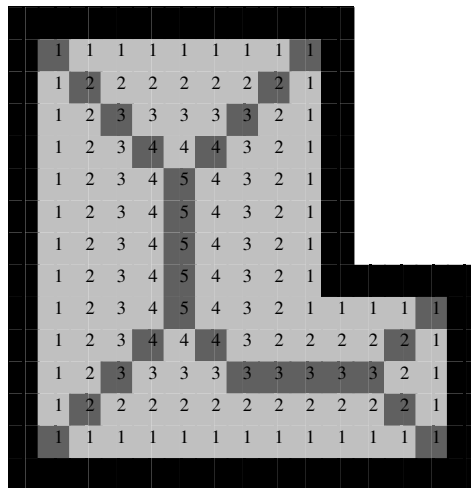


Figura 2.3: Forma geométrica a cinzento-claro com contorno a preto e respectivo eixo médio a cinzento-escuro, representados numa grelha de *pixels*. Os valores indicados representam a distância de cada *pixel* ao contorno, utilizando a métrica induzida pela norma do máximo.

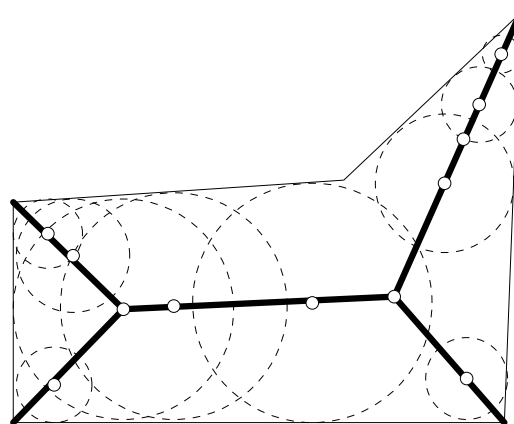


Figura 2.4: Construção do eixo médio de um contorno fechado. O eixo médio é definido pelo centro dos discos máximos contidos na forma e que incidem em pelo menos dois pontos do contorno.

Zhu e Yuille, no seu modelo apresentado em [Zhu, 1996], usaram duas primitivas deformáveis: uma estrutura tubular, *worm*, e um círculo. Estas estruturas adaptam-se à forma em estudo, extraíndo o seu eixo médio. Posteriormente, a forma é decomposta nas suas componentes, extraídas das derivações das estruturas tubulares. Por exemplo, a forma humana poderá ser decomposta em cabeça, tronco e membros. Na Figura 2.5 pode ser observada uma representação das duas primitivas utilizadas.

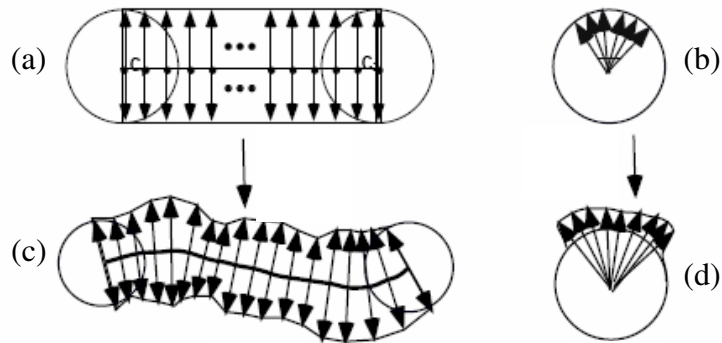


Figura 2.5: Imagens das primitivas utilizadas em [Zhu, 1996]. Estrutura tubular (a), círculo (b), deformações da estrutura tubular e do círculo e consequente adaptação à forma final em questão (c) e (d).

No modelo de Zhu e Yuille, em [Zhu, 1996], para calcular a afinidade entre duas formas, tenta-se, com as partes de uma forma, construir a outra forma, sendo associado um custo a cada operação. Quando numa forma há uma parte que não tem correspondente nas partes da outra forma, também há um custo para esta situação. Para iniciar a reconstrução de uma forma no sentido de obter a outra, começa-se por, por exemplo, por uma zona de derivação do eixo médio (ou vértice, em teoria de grafos) da forma completa e tenta-se encontrar na outra, componentes que encaixem nessa estrutura. A técnica de optimização utilizada foi a programação dinâmica. Tem-se que esta metodologia de emparelhamento é capaz de emparelhar com facilidade duas formas quando uma é obtida da outra por movimentos articulados ou por oclusão de uma parte da forma original. No entanto, por vezes podem ser obtidos emparelhamentos sem sentido. Além disso, a escolha do vértice inicial tem uma grande importância na velocidade de execução do algoritmo e qualidade do emparelhamento obtido.

No modelo proposto em [Sebastian, 2004], o eixo médio da forma, ou esqueleto, é enriquecido com a definição do sentido do fluxo, chamando a esta nova estrutura *shock-graph*. Utilizando a analogia de “queima da erva” de Blum, [Blum, 1967], o sentido do

fluxo é definido da região onde as frentes de fogo chocam mais rapidamente para a região onde demoram mais tempo a chocar. Assim, o sentido do fluxo será sempre da região mais estreita da forma para a região mais larga da mesma. A Figura 2.6 apresenta um exemplo de uma forma e o respectivo *shock-graph* no seu interior. Dos oito pontos assinalados, cinco ligam apenas a uma aresta e três ligam a três arestas cada um. Estes últimos ainda se dividem em dois tipos conforme a orientação do fluxo das arestas que lhe estão ligadas.

De um modo muito simplificado, para cada contorno de um objecto, é extraído um conjunto de linhas orientadas e pontos. As linhas orientadas são os eixos médios da forma e os pontos são os cruzamentos das linhas, as extremidades das linhas ou os pontos de mudança de sentido do fluxo. Pode ser feito o paralelo entre este conjunto de linhas e pontos com a teoria de grafos, considerando os pontos como os vértices dos grafos e as linhas orientadas como as suas arestas orientadas.

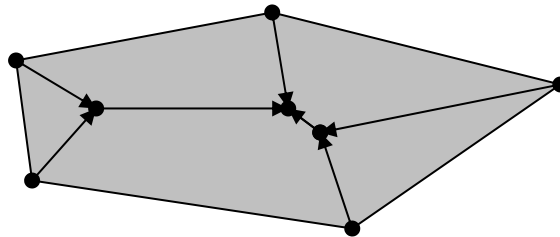


Figura 2.6: Uma forma poligonal e o respectivo *shock-graph* no seu interior.

Quando duas formas são similares, elas ficarão representadas pelo mesmo número de vértices de cada tipo ligando às respectivas arestas, seguindo ambas a mesma ordem de ligação. Em termos de grafos, elas ficarão representadas por grafos isomorfos.

Como no modelo de Sebastian os eixos médios foram enriquecidos com a definição de uma orientação, há mais um factor a influenciar o emparelhamento entre as diversas componentes da forma, pelo que os resultados apresentados em [Sebastian, 2004] são superiores aos apresentados em [Zhu, 1996].

Em comparação com os métodos baseados nas características da forma, os métodos baseadas no eixo médio da forma lidam melhor com situações de oclusão e deformação da forma; no entanto, são mais difíceis de implementar, pois a qualidade do eixo médio extraído condiciona muito a qualidade do emparelhamento obtido. Veja-se, por exemplo, a Figura 2.7, onde uma pequena diferença entre as duas formas origina uma diferença acentuada entre os seus eixos médios. Em relação à velocidade de obtenção dos emparelhamentos, tem-se que as metodologias baseadas no eixo médio da forma apresentam, em geral, menores velocidades.

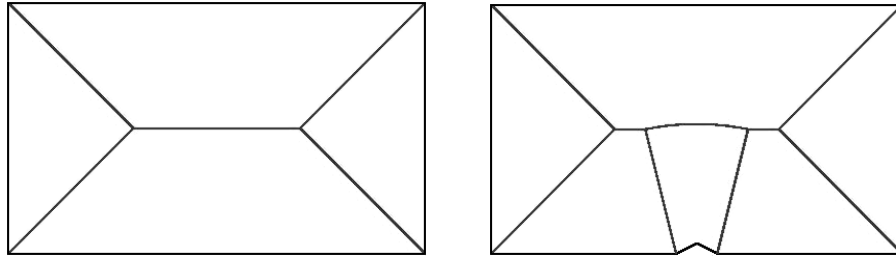


Figura 2.7: Duas formas poligonais e o respectivo eixo médio no seu interior. A forma da direita resulta da introdução de uma pequena perturbação na forma da esquerda, originando um eixo médio significativamente diferente.

2.5 Sumário

Neste capítulo fez-se uma breve abordagem às técnicas de reconhecimento, comparação e emparelhamento de objectos representados em imagens. Referiram-se metodologias que utilizam níveis de cinzento ou de cor das imagens para as comparar ou corresponder, depois foram referidas as metodologias que se baseiam na forma extraída da imagem.

As metodologias baseadas na forma foram divididas em dois grupos principais: as que utilizam características da forma, como por exemplo, pontos e contornos, e as que determinam a estrutura da forma para posteriormente realizarem tarefas de reconhecimento ou emparelhamento. Ao longo deste capítulo foram também realizadas algumas comparações entre as diversas metodologias consideradas, referindo-se pontos fortes e menos fortes de cada uma.

CAPÍTULO III:
ORDENAÇÃO DOS PONTOS QUE DEFINEM UM CONTORNO

3.1 Introdução

Nesta Dissertação é essencialmente abordado o problema do emparelhamento de contornos fechados definidos por um conjunto de pontos. Para evitar emparelhamentos cruzados, garantindo assim uma maior coerência dos emparelhamentos obtidos, é importante definir uma ordem para os pontos que definem cada um dos contornos a emparelhar, de modo que a união sucessiva desses pontos, por essa ordem, represente os respectivos contornos de forma adequada.

Na secção seguinte é feita uma introdução/descrição geral do problema da ordenação de pontos bem como as consequências que podem advir da não definição de uma ordem. Dada a semelhança entre este problema e o do caixeiro-viajante, na secção 3.3 é feita uma referência a este último e à complexidade computacional que a sua resolução envolve.

Este capítulo continua expondo duas formulações para a resolução do problema da ordenação dos pontos de contornos, uma recorre ao algoritmo *simplex* em conjunto com a técnica de *branch-and-bound* e a outra é baseada num modelo de programação inteira. Estas formulações não foram implementadas pois requerem elevado esforço computacional, em especial quando comparadas com o método de ordenação baseado em *simulated annealing* apresentado na secção 3.6. Assim, na secção referida é estudado este último método de optimização de um modo geral bem como a sua adaptação para a resolução do problema da ordenação dos pontos que definem um contorno.

Finalmente, termina-se o capítulo com a apresentação de um algoritmo para determinação do sentido pelo qual os contornos estão definidos: sentido dos ponteiros do relógio ou sentido contrário. Esta informação é depois utilizada para garantir que, aquando do emparelhamento, os dois contornos a corresponder são definidos no mesmo sentido.

3.2 Definição do problema

Conforme o método utilizado para extracção dos pontos que definem um contorno, a sequência de coordenadas dos pontos extraídos pode não estar devidamente ordenada, ou seja, a união sucessiva da sequência de pontos pode não definir correctamente o contorno. Neste caso, pode ser importante, dependendo do método de emparelhamento de objectos representados em imagens a usar, ordenar os pontos para que mais informação sobre o objecto possa ser extraída, como por exemplo: ângulo de curvatura entre três pontos

consecutivos ou perímetro do contorno.

Surge assim um novo problema a resolver: *Como definir a ordem dos pontos extraídos de modo que a sua união sucessiva por segmentos de recta defina o contorno esperado pelo ser humano através da simples observação da representação dos mesmos no plano?* Este problema nem sempre tem uma resposta única, veja-se por exemplo a Figura 3.1. Com base nos mesmos 5 pontos considerados não é possível dizer qual dos contornos corresponde ao do objecto original.

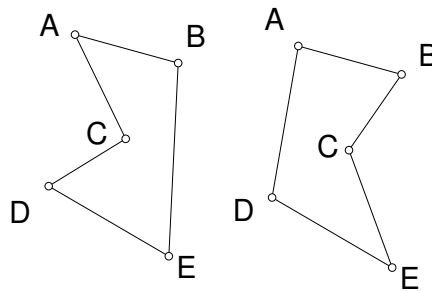


Figura 3.1: Dois contornos distintos definidos pelo mesmo conjunto de pontos.

O problema anterior, em termos práticos, geralmente não se coloca, pois são extraídos muitos pontos do contorno da forma, eliminando assim a ambiguidade que o observador poderia sentir na definição da ordem. Por exemplo, ainda em relação à Figura 3.1, se nela estivessem marcados muitos pontos que pertencessem ao segmento de recta [EB], não teríamos dúvidas em afirmar que o contorno representado do lado direito estaria incorrecto.

Para definir um contorno fechado simples, isto é, uma linha contínua fechada sem cruzamentos entre si e percorrida uma só vez, pensou-se como o ser humano usualmente define o contorno pela observação de um conjunto de pontos extraídos do mesmo. Assim, concluiu-se o seguinte:

- a) O contorno tem que passar por todos os pontos;
- b) Se possível, um ponto deve ligar ao seu vizinho mais próximo;
- c) Um ponto não deverá ser incluído duas vezes na sequência de pontos que define o contorno, pois assim provocará um cruzamento da linha do contorno.

Atendendo às condições anteriores, observou-se que se estava perante um problema equivalente ao tradicional problema do caixeiro-viajante.

3.3 Problema do caixeiro-viajante

Basicamente, este problema consiste no seguinte: *Um vendedor tem um conjunto de n cidades que tem de percorrer para vender os seus produtos. Está interessado em partir de uma delas, percorrê-las todas uma única vez e regressar à cidade de partida de forma a minimizar o percurso total.* No caso da definição da ordem dos pontos do contorno, cada cidade corresponde a um ponto do contorno, a distância percorrida para ligar todas as cidades e regressar ao ponto de partida corresponde ao perímetro do contorno, a ordem pela qual as cidades serão percorridas representa a ordem pela qual os pontos deverão ser considerados para definirem o contorno.

Note-se que o problema do caixeiro-viajante satisfaz as condições impostas ao contorno. Obviamente o contorno passará por todos os pontos (cidades); sempre que possível um ponto ligará ao seu vizinho, caso contrário a distância a percorrer não será minimizada. Devido à desigualdade triangular, cada cidade será visitada apenas uma vez, caso contrário o comprimento do percurso será maior ou, na melhor das hipóteses, igual ao que se obtém fazendo esse mesmo percurso passando apenas uma vez por cada cidade.

Este problema pode ser resolvido fazendo todas as permutações possíveis, calculando a distância do percurso definido por cada uma dessas permutações e depois escolher um percurso que seja menor ou igual que todos os outros. Assim, considerando n o número de cidades, temos inicialmente $n!$ caminhos diferentes. Mas como não interessa por qual das cidades se começa, pois o circuito é fechado (o caixeiro-viajante tem que regressar ao ponto de partida), as hipóteses de percursos são reduzidas para $(n-1)!$. Como, por outro lado, não interessa o sentido do percurso, apenas são considerados $(n-1)!/2$ percursos. Deste modo, este tipo de algoritmos só é computacionalmente praticável para valores de n reduzidos. Note-se que, por exemplo, para 50 cidades há aproximadamente $3,0414 \times 10^{62}$ percursos diferentes que teriam de ser considerados.

Este é um problema de minimização combinatória que pertence a uma classe de problemas conhecidos por problemas *NP-hard* (*Nondeterministic Polynomial-time*), cujo tempo de computação mínimo para uma solução exacta cresce com n do modo e^{kn} , sendo k uma constante positiva, [Press, 2002]. Assim, rapidamente o crescimento de n torna o problema impraticável. Mais concretamente, usando programação dinâmica, pode ser obtida uma solução exacta do problema num tempo de $O(n^2 2^n)$, [Held, 1962]. Existem também algoritmos de *branch-and-bound* e de programação inteira que resolvem este

problema, embora devido ao número de variáveis e restrições envolvidas não sejam computacionalmente viáveis para valores elevados de n . Nas secções seguintes serão apresentados dois exemplos dessas formulações.

Caso não seja fundamental encontrar sempre uma solução óptima, podem ser considerados outros algoritmos, havendo na literatura um leque considerável de propostas: *Ant Colonies*, [Dorigo, 1997]; *Simulated Annealing*, [Press, 2002]; *Nearest Neighbor Algorithms*, *Greedy Algorithms*, *Christofides Algorithms*, *Genetic Algorithms*, [Johnson, 1997, 2002]. Em alguns desses algoritmos pode-se até controlar o erro máximo entre o valor exacto do percurso mínimo e o valor obtido para o percurso encontrado.

O problema do caixeiro-viajante pertence também à classe dos problemas de minimização para os quais a função a minimizar, ou função objectivo, tem vários mínimos locais. Em alguns casos práticos é muitas vezes suficiente ser capaz de escolher um desses mínimos locais, ainda que não absoluto, pois não podem ser significativamente melhorados.

Como exemplo muito simples de um algoritmo não exacto (neste caso: *Nearest Neighbor*), pode-se pensar num percurso que inicia numa cidade qualquer e depois vai saltando para a cidade mais próxima que ainda não tenha sido visitada. Rapidamente este algoritmo encontra um trajecto relativamente curto, mas dificilmente será óptimo. Em [Rosenkrantz, 1977] mostra-se que, no plano euclidiano, este algoritmo apresenta como resultados médios um comprimento de $1,26 \times (\text{comprimento do caminho óptimo})$. Este algoritmo ainda pode ser melhorado se, por exemplo, forem calculados n percursos mínimos pela técnica anterior, um a sair de cada uma das cidades, e depois escolhido o mínimo de entre eles.

O método *Simulated Annealing*, ou simulação de têmpera de metais, encontra um mínimo global, ou um mínimo local muito próximo do mínimo global, em tempo da ordem de uma baixa potência de n , [Press, 2002]. Deste modo, para valores de n não demasiado pequenos, a velocidade deste algoritmo será muito superior à dos algoritmos que encontram sempre uma solução óptima.

3.4 Ordenação dos pontos de um contorno usando o *simplex* e a técnica de *branch-and-bound*

Consideremos um contorno definido por n pontos, obviamente $n \geq 3$. Numeremos os

pontos, isto é, i representa o *ponto* que na nossa ordem arbitrária ocupa a i -ésima posição, com $i = 1, 2, \dots, n$. Seja $D = \begin{bmatrix} d_{ij} \end{bmatrix}$ a matriz das distâncias entre os pontos do contorno, onde d_{ij} representa a distância entre o *ponto* i e o *ponto* j . Seja $X = \begin{bmatrix} x_{ij} \end{bmatrix}$ uma matriz de variáveis de decisão binárias, em que:

$$x_{ij} = \begin{cases} 1, & \text{se o contorno vai directamente do ponto } i \text{ para o ponto } j; \\ 0, & \text{caso contrário.} \end{cases}$$

Podemos definir a função objectivo a minimizar como:

$$f = \sum_{i=1}^n \sum_{j=1}^n x_{ij} d_{ij}. \quad (3.1)$$

Observe-se que $d_{ii} = 0$, pois representa a distância entre o *ponto* i e ele mesmo. Assim, como um ponto não poderá ligar a ele mesmo, os elementos d_{ii} da matriz D serão substituídos por um valor positivo muito elevado, isto é, $d_{ii} = M$ onde M é um valor consideravelmente superior aos restantes elementos da matriz D .

Passemos às restrições a considerar. Como o contorno tem que sair uma só vez de cada ponto, tem-se que:

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, 2, \dots, n. \quad (3.2)$$

Por outro lado, como só há uma chegada a cada ponto, tem-se que:

$$\sum_{i=1}^n x_{ij} = 1, \quad j = 1, 2, \dots, n. \quad (3.3)$$

Como as variáveis de decisão só podem assumir os valores 0 (zero) ou 1 (um), tem-se ainda:

$$x_{ij} \in \{0, 1\}, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, n. \quad (3.4)$$

Este problema pode ser relaxado, no sentido em que a restrição (3.4) pode ser substituída pela seguinte:

$$x_{ij} \geq 0, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, n. \quad (3.5)$$

Como a matriz das restrições é totalmente unimodular¹ e os valores do segundo membro das restrições são todos inteiros, então pelo Teorema de Hoffman e Kruskal², pode-se concluir que os vértices do espaço solução do problema são inteiros, pelo que está garantida a convergência do *simplex* relaxado para uma solução óptima inteira³. Por outro lado, nunca se terá $x_{ij} \geq 1$ qualquer que sejam i e j , atendendo às restrições dadas pelas condições (3.2) e (3.3).

Atendendo ao referido anteriormente, este problema, inicialmente de programação inteira, pode ser resolvido como um problema de programação linear, o que trás vantagens em termos de tempo de computação. Da resolução do problema tal como está formulado, pode acontecer uma das seguintes situações:

1. É encontrada uma solução óptima que define um percurso contínuo percorrendo sucessivamente todos os pontos. Neste caso a solução encontrada corresponde ao pretendido.
2. A solução óptima encontrada define vários subcontornos desconexos. Por exemplo, se $x_{92} = x_{29} = 1$ obtém-se um subcontorno que sai do *ponto 9*, vai para o *ponto 2* e depois regressa ao *ponto 9*. Deste modo, esta secção do contorno é desconexa das restantes e portanto a solução encontrada não satisfaz a condição de continuidade do contorno.

No caso da segunda situação, ter-se-á que reformular o problema inicial, dividindo-o em subproblemas. Por exemplo, para o caso de se ter $x_{92} = x_{29} = 1$, é necessário dividir o problema inicial em dois subproblemas. No primeiro define-se explicitamente que $x_{92} = 0$, impedindo deste modo que o contorno possa ir directamente do *ponto 9* para o *ponto 2*. No segundo subproblema impõe-se que $x_{29} = 0$, para que o contorno não vá directamente do *ponto 2* para o *ponto 9*. Note-se que estes dois subproblemas no seu conjunto admitem que

¹Uma matriz diz-se totalmente unimodular se o determinante das suas submatrizes quadradas for -1, 0 (zero) ou 1.

²Teorema de Hoffman e Kruskal: Seja A uma matriz de números inteiros de dimensão $m \times n$, então A é totalmente unimodular se e só se, para todo o vector $b \in Z^n$, o poliedro $P(A, b) = \{x \in \mathfrak{R}_+^n : Ax \leq b\}$ é inteiro, [Tavares, 2005].

³Note-se que o algoritmo *simplex* procura uma solução óptima apenas nos vértices do espaço solução.

que o contorno vá directamente do *ponto 2* para o *9* ou vice-versa, o que não permitem é que o contorno vá simultaneamente do *ponto 9* para o *2* e depois do *ponto 2* para o *ponto 9*.

Seguidamente, resolvem-se estes subproblemas e os que possam ainda surgir em consequência destes. Depois de todos os subproblemas resolvidos, de entre todas as soluções válidas escolhe-se uma que defina um contorno com um perímetro mínimo.

No caso da resolução do problema definido inicialmente originar vários subcontornos desconexos, independentemente do comprimento destes (número de pontos que os constituem), pode-se sempre proceder do modo indicado anteriormente, dividindo o problema inicial em tantos subproblemas quantos forem necessários.

Pelo acabado de descrever, o processo anteriormente referido conduz sempre à solução óptima pretendida, mas este processo tanto poderá ser relativamente rápido como extremamente lento.

No âmbito do trabalho desenvolvido, foi implementada em C++, [Rodrigues, 2003], uma formulação idêntica a esta, sendo realizadas algumas experiências, embora não se tenha considerado a parte do *branch-and-bound*. Os resultados obtidos não foram suficientemente motivadores, pois associada a alguma lentidão na obtenção de uma solução, esta frequentemente definia vários subcontornos desconexos. Para impedir o surgimento de subcontornos de comprimento 2, às restrições anteriores foram acrescentadas as seguintes restrições:

$$x_{ij} + x_{ji} \leq 1, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, n, \quad i \neq j. \quad (3.6)$$

Agora, o esforço computacional passou a ser superior, pois no total, a condição (3.6) representa $n(n-1)/2$ restrições. Além disso, embora os elementos da matriz das restrições sejam apenas os valores 0 (zero) ou 1, esta deixou de ser totalmente unimodular, portanto, não está garantida a obtenção de soluções óptimas inteiras. Apesar de tudo, foram realizadas algumas experiências com esta nova formulação, mas como era de esperar, os subcontornos de comprimento maior do que 2 continuaram a aparecer e em algumas situações as variáveis deixaram de ser inteiras.

Para corrigir os problemas surgidos, poder-se-ia acrescentar mais restrições para eliminar os subcontornos de comprimento 3, 4, ..., $n/2$ e aplicar um algoritmo de programação inteira, mas então o processo de obtenção de uma solução passaria a ser demasiadamente dispendioso em termos computacionais.

3.5 Ordenação dos pontos de um contorno usando uma formulação de programação inteira

Na pesquisa efectuada, nomeadamente em [Winston, 1994], foi encontrada uma formulação de programação inteira, onde não é necessário efectuar a divisão do problema inicial em subproblemas, a qual vamos seguidamente apresentar.

Consideremos as variáveis e matriz D definidas na secção anterior. Temos então a seguinte formulação do problema:

Função objectivo a minimizar:

$$f = \sum_{i=1}^n \sum_{j=1}^n x_{ij} d_{ij}. \quad (3.7)$$

Restrições:

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, 2, \dots, n; \quad (3.8)$$

$$\sum_{i=1}^n x_{ij} = 1, \quad j = 1, 2, \dots, n; \quad (3.9)$$

$$u_i - u_j + nx_{ij} \leq n - 1, \quad \text{para } i \neq j, \quad i = 2, 3, \dots, n, \quad j = 2, 3, \dots, n; \quad (3.10)$$

$$x_{ij} \in \{0, 1\}, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, n; \quad (3.11)$$

$$u_j \geq 0, \quad j = 1, 2, \dots, n. \quad (3.12)$$

A restrição dada pela condição (3.10) vai impedir a formação de subcontornos desconexos. Uma justificação da mesma pode ser consultada em [Winston, 1994].

Esta formulação não foi implementada, pois é computacionalmente muito dispendiosa. Note-se que são necessárias $n^2 + n - 1$ variáveis e $2n + (n - 1)(n - 2)$ restrições. Além disso, n^2 variáveis são inteiras. Como consequência, o esforço computacional será muito significativo. Por outro lado, como não se verificam as condições que garantam a existência de apenas soluções óptimas inteiras utilizando o *simplex* relaxado, pois a matriz das restrições não é totalmente unimodular, terão de ser utilizados algoritmo específicos para programação inteira.

3.6 Aplicação do método *simulated annealing* à ordenação dos pontos de um contorno

3.6.1 Apresentação do método

O método de simulação de têmpera de metais, *simulated annealing*, é uma técnica que atraiu uma significativa atenção pela sua aplicabilidade a problemas de otimização em grande escala, especialmente aqueles onde é necessário encontrar um extremo global que se encontra entre muitos extremos locais. Foi verificado, em aplicações práticas, que este método resolve eficientemente o problema do caixeiro-viajante. É muito utilizado, por exemplo, para desenhar circuitos integrados, onde é necessário dispor centenas de milhares de componentes de modo a minimizar a interferência entre os “fios” que os ligam, [Press, 2002].

Na base do método *simulated annealing* está a analogia com a termodinâmica, mais especificamente com o modo como os líquidos congelam e cristalizam, ou arrefecimento e têmpera de metais. Quando a altas temperaturas, as moléculas de um líquido movem-se livremente umas em relação às outras. Se o líquido é arrefecido lentamente, a mobilidade térmica é diminuída. Os átomos são muitas vezes capazes de eles próprios se organizarem ou ordenarem até formarem um cristal completamente ordenado, sendo a dimensão deste cristal milhões de vezes superior à dimensão de um simples átomo. Além disso, o cristal é o estado de energia mínima do sistema.

Um facto fundamental é que, para sistemas de arrefecimento lento, a natureza é capaz de encontrar o estado de energia mínima. Se um metal no estado líquido for arrefecido rapidamente ele não alcança esse estado, mas termina com a formação de vários pequenos cristais ou num estado amorfo em que muitas vezes ainda tem muita energia. A essência do processo é o arrefecimento lento, permitindo assim que os átomos tenham tempo para se organizarem à medida que perdem energia e portanto a sua mobilidade. Esta é a definição técnica de *annealing*, e é essencial para garantir que um estado de baixa energia seja alcançado, [Press, 2002].

A natureza tem o seu próprio algoritmo de minimização de energia que é baseado num procedimento algo aleatório: a chamada distribuição de probabilidade de Boltzmann, [Press, 2002]:

$$P(E) \approx e^{-\frac{E}{kT}} .$$

Esta fórmula dá a probabilidade de um sistema estar num determinado estado de energia E , em função do seu estado de energia, da temperatura T a que o sistema se encontra e de uma quantidade k (constante de Boltzmann) da natureza que relaciona a temperatura com a energia.

Pela observação da expressão da distribuição de probabilidade de Boltzmann, conclui-se que à medida que a temperatura baixa (temperatura absoluta) a probabilidade do sistema estar num determinado estado elevado de energia também baixa. No entanto, existe uma possibilidade, muito pequena, do sistema estar num estado elevado de energia. Este facto permite que o sistema possa sair de um estado local mínimo de energia e encontrar outro globalmente melhor.

Em 1953, Metropolis e colaboradores incorporaram estes princípios inicialmente em cálculos numéricos, [Press, 2002]. Oferecendo uma sucessão de opções, o sistema de simulação termodinâmica foi assumido para alterar a sua configuração do estado de energia E_1 para o estado de energia E_2 com probabilidade:

$$P = e^{-\frac{E_2 - E_1}{kT}} .$$

Repare-se que esta expressão representa um valor maior do que 1 se $E_2 < E_1$; nestes casos, a probabilidade é arbitrariamente assumida como 1, isto é, o sistema toma sempre a mesma opção predefinida. Este esquema geral, de aceitar sempre um passo na direcção da descida, embora por vezes dê um passo na direcção da subida, ficou conhecido por algoritmo de Metropolis. Para utilizar este algoritmo noutros sistemas diferentes dos termodinâmicos, temos que inicialmente proporcionar os seguintes elementos:

- a) A descrição das possíveis configurações do sistema.
- b) Um gerador aleatório de mudanças na configuração; estas mudanças são as opções fornecidas ao sistema.
- c) Uma função objectivo E (análoga à energia) cuja minimização é o objectivo do procedimento.
- d) Um parâmetro de controlo T (análogo à temperatura) e um registo dos níveis de energia obtidos que traduza quanto baixou dos altos para os baixos valores; por exemplo, quantas mudanças aleatórias na configuração foram necessárias para cada passo de descida para cada valor de T , e qual a amplitude dessa descida.

Muito simplesmente, um algoritmo geral de *simulated annealing* consiste no seguinte:

1. Parte-se de uma situação de energia e temperatura iniciais.
2. Gera-se uma série de valores aleatórios de arranjo, aceitando todos os que levam à diminuição do estado de energia e aceitando com probabilidade de Boltzmann os que levam ao aumento do estado de energia. Aqui, para cada aplicação em particular, pode ser definido um processo de arranjos mais eficaz do que o puramente aleatório.
3. Com base no registo, verifica-se se já se pode terminar o processo. Caso contrário, baixa-se a temperatura e regressa-se ao passo 2.

3.6.2 Implementação

Após a pesquisa efectuada foi decidido testar uma rotina proposta em [Press, 2002] para resolver o problema do caixeiro-viajante, baseado em *simulated annealing*. Esta rotina foi por nós adaptada e implementada para resolver o problema de ordenação dos pontos de um contorno. Assim, e atendendo ao já referido na subsecção anterior aquando da descrição do método, para aplicação desta rotina apenas foi necessário definir uma ordem inicial, neste caso, a ordem pela qual são lidos os pontos do ficheiro, e indicar as suas coordenadas. A rotina devolve o comprimento do melhor percurso encontrado e a ordem respectiva pela qual os pontos devem ser percorridos.

A rotina, tal como estava definida inicialmente em [Press, 2002], não encontrava sempre o percurso óptimo. Por exemplo, para um dos contornos testados constituído por 135 pontos, a sua eficácia era aproximadamente 80%, isto é, a rotina só definia uma ordem correcta em 80% dos casos. Nos outros 20%, apresentava uma solução muito próxima da óptima. No sentido de aumentar a sua eficácia e melhorar a sua velocidade, foram realizadas várias experiências, tendo-se chegado à formulação que se descreve de seguida.

Sejam n o número de pontos do contorno e $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ a sequência de pontos do contorno pela ordem de leitura dos mesmos, tem-se:

Algoritmo:

1. Lêem-se os pontos do ficheiro e numeram-se pela ordem de leitura.
2. Calcula-se o valor da função objectivo E , que muito simplesmente é a soma das distâncias entre os pontos na sequência a estudar:

$$E = \sum_{i=1}^n \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$$

assumindo que o *ponto* (x_{i+1}, y_{i+1}) é o que vem a seguir ao *ponto* (x_i, y_i) na sequência de pontos do contorno.

3. Calcula-se a média das distâncias entre todos os pontos, \bar{X} .
4. Define-se um valor inicial para o parâmetro positivo k , definindo seguidamente o valor de T (temperatura): $T = k \times \bar{X}$.
5. Escolhe-se um dos dois movimentos predefinidos.
6. Calcula-se ΔE (variação do comprimento do caminho) com base nesse movimento e consulta-se o algoritmo de Metropolis em função desse ΔE e T para decidir se é aceite ou não esse novo arranjo. Em caso afirmativo efectiva-se esse arranjo e actualiza-se o valor de E .
7. Enquanto não se completarem $4n^2$ permutações (ou movimentos) nem se encontrarem $10n$ reconfigurações bem sucedidas, regressa-se ao passo 5.
8. Enquanto ainda houver movimentos bem sucedidos e não se tenham completado 100 diminuições da temperatura, baixa-se a temperatura em 10% e regressa-se ao passo 5.
9. Finalmente, apresentam-se os resultados.

Na rotina implementada, as trocas de posições entre os diversos pontos na sequência definida para o contorno não são totalmente aleatórias, embora pudessem sê-lo, com a desvantagem de poder tornar o processo de convergência muito lento. Os dois movimentos definidos para aumentar a velocidade com que o algoritmo alcança um percurso óptimo foram sugeridos em [Lin, 1965] e são seguidamente apresentados:

- a) Uma secção do contorno é escolhida aleatoriamente de modo que haja pelo menos 3 pontos que não lhe pertencem. Depois é removida e substituída por ela mesma mas percorrida em sentido contrário. Deste modo, poderão ser facilmente eliminados laços, isto é, cruzamento da linha de contorno. Assim, conseguem-se diminuições significativas no comprimento do contorno, Figura 3.2.
- b) Tal como no movimento anterior, é escolhida uma secção do caminho que é removida e depois recolocada entre outros dois pontos consecutivos escolhidos aleatoriamente de entre os que fazem parte da secção não removida, Figura 3.3.

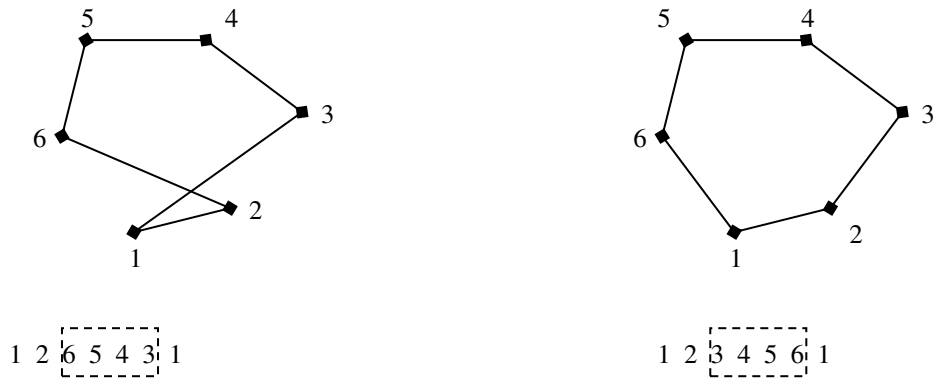


Figura 3.2: Exemplo do processo de inversão da ordem na sequência de pontos de um contorno. Neste caso, foi invertida uma sequência formada por 4 pontos.

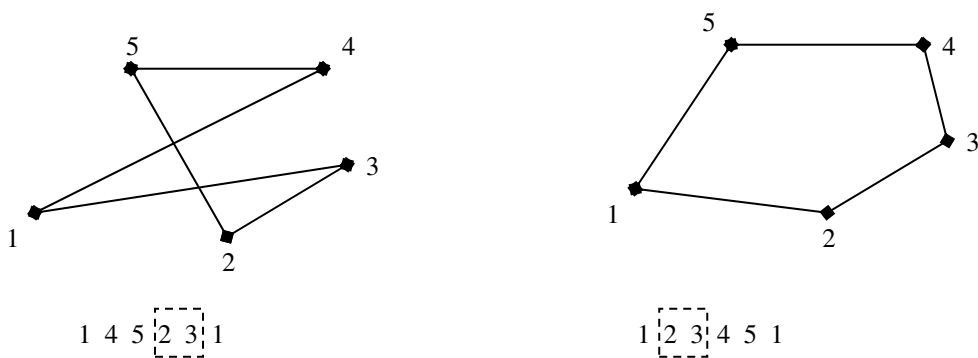


Figura 3.3: Exemplo do processo de troca de posição de uma secção da sequência de pontos de um contorno. Neste caso, os *pontos* 2 e 3, que estavam entre os *pontos* 5 e 1 na sequência inicial, foram colocados entre os *pontos* 1 e 4.

3.6.3 Análise de resultados

Como já referido, a rotina proposta em [Press, 2002] apresentou resultados insatisfatórios para contornos definidos por elevado número de pontos, razão pela qual teve de ser alterada. Mas antes de se chegar à formulação final para a mesma, foi necessário realizar várias de experiências com formulações diferentes. Da análise dos resultados das mesmas, verificou-se que de facto, valores iniciais para T muito baixos originaram por vezes percursos não óptimos. Valores elevados de T diminuíram a velocidade de execução, sem no entanto melhorarem significativamente a qualidade dos percursos obtidos. Verificou-se também que para o controlo da eficácia do algoritmo em termos de qualidade do percurso encontrado e velocidade de execução, o número de permutações permitidas para cada valor de T é mais importante do que o valor inicial de T .

Após as experiências iniciais efectuadas, verificou-se que a mudança do número de permutações para $4n^2$ em vez das $100n$ proposta na rotina inicial (n representa o número de pontos do contorno) para cada valor de T originou resultados óptimos em termos de percurso e também velocidades de execução óptimas para baixos valores iniciais de T . Para valores iniciais de T muito baixos, verificou-se até que o aumento do número de permutações tornou o algoritmo mais rápido, ao contrário do que era inicialmente esperado. A explicação deste comportamento consiste no seguinte:

- Para contornos definidos por reduzido número de pontos, a expressão $4n^2$ representa um conjunto menor de permutações do que a expressão $100n$, mas este conjunto de permutações é suficiente para alcançar um percurso óptimo.
- Para contornos definidos por um elevado número de pontos, a expressão $4n^2$ representa um maior número de permutações do que a expressão $100n$. No entanto, o aumento do número de permutações permitiu encontrar um melhor percurso para cada sucessivo valor de T , fazendo deste modo que o percurso óptimo seja encontrado sem necessidade de proceder a tantas reduções do parâmetro T como as necessárias anteriormente.

Após as experiências iniciais e consequentes alterações na rotina, foram realizadas várias experiências com diversos contornos para os seguintes valores iniciais de T : $T = \bar{X}/k$, onde \bar{X} representa a média da distância entre todos os pontos e $k = 1, 10^2, 10^3, 10^4, 10^5$.

Na Tabela 3.1, são apresentados resultados apenas para três contornos diferentes, representativos da dimensão dos conjuntos de pontos que definem os contornos utilizados. Como se pode observar nesta tabela, em todas as experiências efectuadas com a nova formulação indicada, os contornos encontrados correspondem ao que seria de esperar pela observação da distribuição do conjunto de pontos. Para garantir que o algoritmo é realmente eficaz, antes de cada experiência a ordem dos pontos do contorno a utilizar era baralhada aleatoriamente com recurso à função *rand* do C++. Os tempos indicados são tempos médios, pois como para cada novo teste a ordem dos pontos era diferente da ordem dos testes anteriores e, além disso, o processo de ordenação tem uma componente aleatória considerável, ocorrem ligeiras variações no tempo de execução. O algoritmo foi desenvolvido em ambiente *Microsoft Visual C++*, [Rodrigues, 2003], e testado num PC equipado com um processador *Intel Pentium III* a 1.0 GHz, com 256MB de RAM.

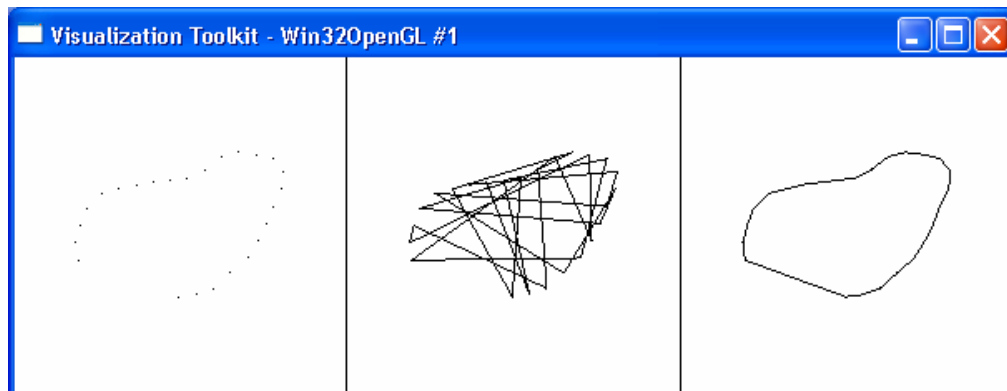
Tabela 3.1: Resultados da ordenação de três dos contornos usados nos ensaios, em função do valor inicial do parâmetro T (temperatura). (\bar{X} representa a média da distância entre os pontos do contorno.)

Contorno	Número de pontos	Valor do parâmetro T	Tempo de execução [s]	N.º de ordenações incorrectas em 100 testes
"heart3"	25	$T = \bar{X}$	0,085	0
		$T = \bar{X} / 10$	0,028	0
		$T = \bar{X} / 100$	0,009	0
		$T = \bar{X} / 1000$	0,012	0
		$T = \bar{X} / 10000$	0,011	0
		$T = \bar{X} / 100000$	0,010	0
"tree1"	62	$T = \bar{X}$	0,653	0
		$T = \bar{X} / 10$	0,430	0
		$T = \bar{X} / 100$	0,066	0
		$T = \bar{X} / 1000$	0,086	0
		$T = \bar{X} / 10000$	0,078	0
		$T = \bar{X} / 100000$	0,073	0
"heartB1"	135	$T = \bar{X}$	3,229	0
		$T = \bar{X} / 10$	2,560	0
		$T = \bar{X} / 100$	0,368	0
		$T = \bar{X} / 1000$	0,440	0
		$T = \bar{X} / 10000$	0,394	0
		$T = \bar{X} / 100000$	0,383	0

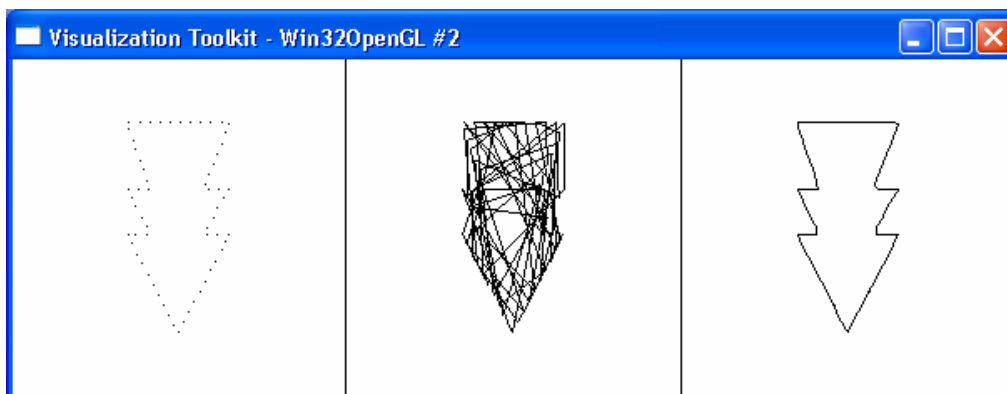
Pela observação da Tabela 3.1, verifica-se que a velocidade de convergência do algoritmo, como era de esperar, é tanto mais lenta quanto maior for o número de pontos do contorno. Quanto menor for o valor inicial da temperatura, mais rápida é a execução do algoritmo, excepto para valores de T muito baixos, onde se verifica uma estabilização da velocidade e em alguns casos um ligeiro aumento no tempo de execução. O valor do parâmetro T que proporcionou maior velocidade foi $T = \bar{X} / 100$. Observa-se, ainda, que o algoritmo alcançou sempre um caminho óptimo para percorrer os pontos do contorno, obtendo sempre um percurso de comprimento mínimo. Neste aspecto, a sua eficácia foi 100% nas 1800 experiências realizadas e registadas na Tabela 3.1.

Nas Figuras 3.4 (a), (b) e (c) são apresentadas imagens representativas dos três contornos utilizados para extrair os dados apresentados na Tabela 3.1. Para cada uma das três imagens apresentadas, à esquerda o conjunto de pontos, ao centro o contorno obtido

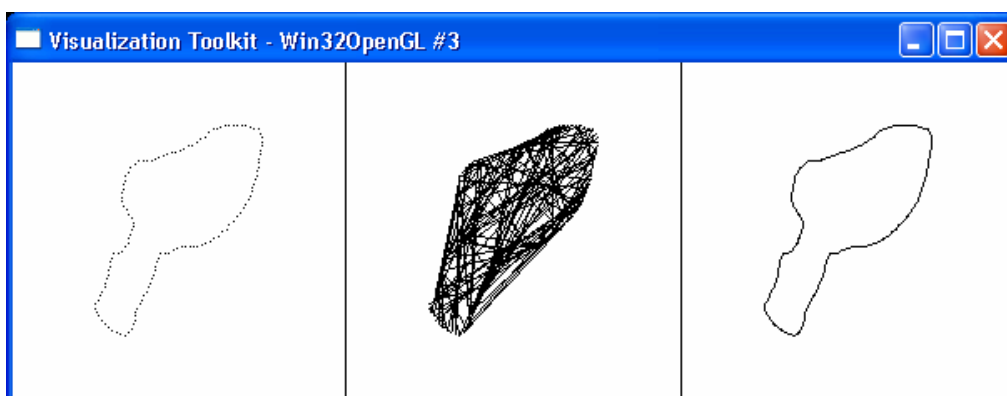
após a perturbação da ordem dos pontos e à direita o contorno após ordenação dos pontos, unindo os pontos consecutivos.



(a)



(b)



(c)

Figura 3.4: Exemplos de três dos contornos utilizados para fazer os testes do algoritmo de ordenação dos pontos: (a) “*heart3*”; (b) “*tree1*”; (c) “*heartB1*”. Em cada figura, à esquerda, os pontos; ao centro os pontos unidos pela sequência inicial e, à direita, os pontos unidos após ordenação.

Note-se que o processo de perturbação da ordem dos pontos era aleatório. Assim, para cada teste, a imagem que representa o contorno após essa perturbação era diferente.

Portanto, as imagens apresentadas nas Figuras 3.4 (a), (b) e (c) dizem respeito apenas a uma experiência em particular.

3.7 Determinação do sentido pelo qual o contorno está definido

Da aplicação do algoritmo de ordenação dos pontos de um contorno apresentado na secção anterior, o sentido pelo qual os pontos do contorno são considerados é totalmente indiferente, pelo que a ordenação encontrada tanto pode definir um percurso no sentido dos ponteiros do relógio como em sentido contrário. No entanto, para o método de determinação da transformação rígida que melhor alinha dois contornos, para algoritmo de optimização global das correspondências e para as metodologias de emparelhamento de contornos que serão apresentados nos capítulos quarto, quinto e sexto, respectivamente, é fundamental que ambos os contornos a alinhar ou emparelhar sejam definidos no mesmo sentido, independentemente de serem ambos definidos no sentido dos ponteiros do relógio ou ambos em sentido contrário. Assim, foi desenvolvido um algoritmo capaz de determinar o sentido pelo qual os contornos estão definidos e em caso de terem sentidos diferentes trocar a orientação de um deles.

Para detectar o sentido, o algoritmo procura os pontos extremos do contorno: o ponto com menor ordenada, o ponto com maior abcissa, o ponto com maior ordenada e finalmente o ponto com menor abcissa. Para cada um desses quatro pontos, o algoritmo lê a posição desses pontos na sequência de pontos do contorno, ou seja, a sua posição numa sequência de 1 a n , para um contorno definido por n pontos.

Sejam a , b , c e d as posições respectivas, na sequência de pontos do contorno, indicadas pela ordem referida no parágrafo anterior. Assim, se o contorno estiver a ser definido no sentido contrário ao dos ponteiros do relógio e pelo menos três das posições anteriormente referidas forem diferentes, apenas uma e só uma das seguintes quatro condições é falsa: $a \leq b$, $b \leq c$, $c \leq d$ e $d \leq a$. Caso contrário, o contorno está a ser definido no sentido dos ponteiros do relógio. Observe-se uma exemplificação representada na Figura 3.5.

Da observação da Figura 3.5, considerando as posições na sequência dos pontos extremos, menor ordenada, maior abcissa, maior ordenada e menor abcissa, por esta ordem, tem-se: 5, 6, 1, 3. Assim, só uma das condições $5 \leq 6$, $6 \leq 1$, $1 \leq 3$ e $3 \leq 5$ é falsa, pelo que o contorno está a ser percorrido no sentido contrário ao dos ponteiros do relógio.

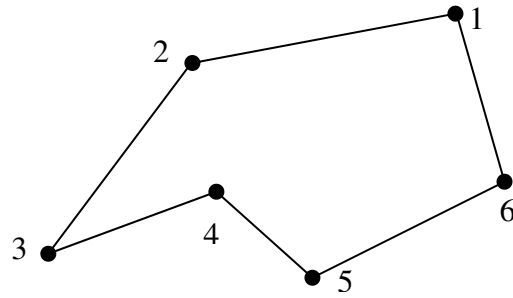


Figura 3.5: Exemplo de um contorno ordenado no sentido contrário ao dos ponteiros do relógio.

Pode acontecer, embora pouco provável, que das posições dos pontos extremos referidas, apenas hajam duas diferentes. Neste caso, a aplicação de uma rotação adequada ao contorno, permitirá encontrar pelo menos três pontos extremos distintos. Na Figura 3.6 apresenta-se um exemplo de um contorno para o qual o algoritmo terá de realizar uma rotação do mesmo para determinar o sentido pelo qual está a ser definido. Ainda em relação à mesma figura, considerando as posições dos pontos extremos (menor ordenada, maior abcissa, maior ordenada e menor abcissa) na sequência, o contorno da esquerda só tem dois pontos extremos distintos, enquanto o da direita já possui três pontos extremos distintos.

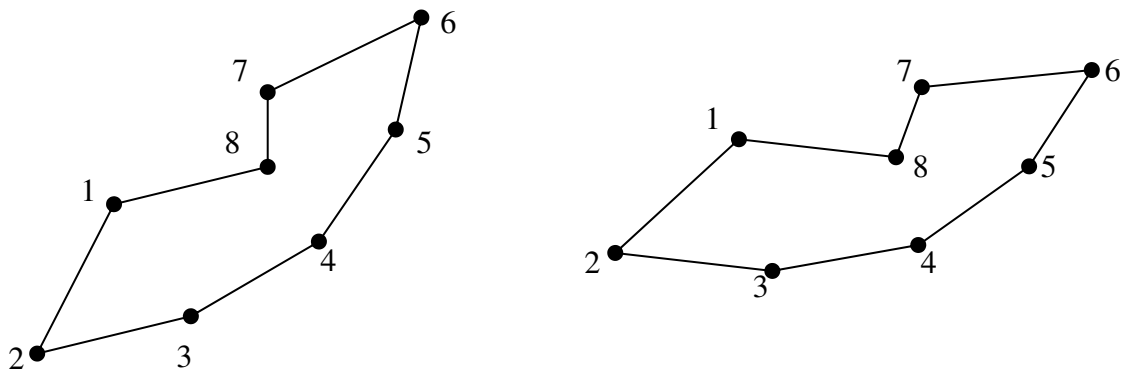


Figura 3.6: Duas configurações do mesmo contorno. À esquerda o contorno original e à direita o mesmo contorno após uma rotação.

Caso as soluções anteriores não encontrem o sentido em que o contorno está definido, então, provavelmente o conjunto de pontos não define um contorno, aceitando-se por defeito a ordem predefinida.

Após determinado o sentido pelo qual cada um dos contornos a emparelhar está definido, para inverter o sentido de um deles, basta reordenar os pontos desse contorno.

Por exemplo, se o contorno a inverter for definido por n pontos numerados de 1 a n , o *ponto 1* passa a ser o *ponto* que antes ocupava a posição n , o *ponto 2* passa a ser o *ponto* que antes ocupava a posição $n - 1$, e assim sucessivamente.

3.8 Sumário e conclusões

Neste capítulo abordou-se o problema da ordenação dos pontos que definem um contorno. Fez-se inicialmente a analogia deste problema com o problema do caixeiro-viajante. Foram apresentadas três formulações capazes de resolver este problema, mas apenas a terceira foi implementada, pois é a de menor custo computacional. A primeira formulação utiliza o algoritmo *simplex* relaxado (para variáveis contínuas) para obter uma primeira ordenação. Se essa ordenação definir mais do que um subcontorno fechado, então, usando uma técnica de *branch-and-bound*, o problema inicial é dividido em dois subproblemas e assim sucessivamente.

A segunda formulação resolve o problema de uma só vez, mas como é uma formulação de programação inteira e a matriz das restrições não é totalmente unimodular, o *simplex* relaxado não poderá ser aplicado, pelo que terá que ser aplicada uma formulação específica para programação inteira, o que tornará o processo de obtenção de uma solução mais lento.

Continuou-se este capítulo com a apresentação de um método de optimização baseado na técnica *simulated annealing*, fazendo-se uma adaptação do mesmo para resolver o problema da ordenação dos pontos que definem um contorno. Os resultados obtidos foram bons, quer em termos de velocidade de execução quer em termos de eficiência na obtenção de uma ordenação óptima. Ao contrário dos outros métodos descritos, este não é determinista, portanto, uma certa margem de erro está associada ao mesmo. Mas, após as adaptações efectuadas neste, verificou-se um sucesso de 100% na obtenção de uma ordenação óptima em mais de 1800 experiências efectuadas com diversos contornos, sendo que para cada experiência a ordem dos pontos era inicialmente perturbada aleatoriamente.

Finalmente, atendendo que as ordenações dos contornos obtidas podem definir um percurso no sentido dos ponteiros do relógio ou em sentido contrário, apresentou-se um algoritmo capaz de detectar o sentido pelo qual o contorno está definido e invertê-lo caso seja pretendido.

CAPÍTULO IV:
DETERMINAÇÃO DA TRANSFORMAÇÃO RÍGIDA QUE
MELHOR ALINHA DOIS CONTORNOS

4.1 Introdução

Dois objectos dizem-se semelhantes quando têm a mesma forma, independentemente da sua posição, rotação ou dimensão. Deste modo, para alinhar dois objectos semelhantes é necessário identificar a transformação de semelhança (translação, rotação e escala) que melhor transforma um no outro.

Neste capítulo é apresentada uma metodologia para determinar a transformação de semelhança (transformação rígida) que melhor alinha dois contornos definidos por conjuntos de pontos ordenados. São realizados e analisados vários ensaios experimentais para validar a referida metodologia. No entanto, no sexto capítulo serão apresentados mais resultados experimentais, especialmente relacionados com o valor do ângulo de rotação, pois este depende da obtenção de emparelhamentos prévios.

4.2 Centróide e translação

Para determinar a translação, começa-se por determinar os centróides dos contornos a alinhar. Estes são determinados por um processo em que o peso de cada ponto é ponderado conforme a sua distância aos seus pontos vizinhos na sequência de pontos do contorno. Sendo que, para um determinado ponto, quanto mais próximos estão os seus dois vizinhos, menor será o peso desse ponto para o cálculo das coordenadas do centróide. Assim, evita-se o deslocamento do centro do contorno para as regiões onde há maior densidade de pontos. Deste modo, diferenças na densidade da amostragem em diferentes regiões do contorno não terão influência na determinação do centróide. Vejamos a sua formulação matemática.

Sejam P_1, P_2, \dots, P_n uma sequência de pontos que define um determinado contorno. Assim, temos que:

$$\frac{1}{2} \times \left(\left\| \overrightarrow{P_{i-1}P_i} \right\| + \left\| \overrightarrow{P_iP_{i+1}} \right\| \right),$$

define a média da distância do *ponto* $P_i = (x_i, y_i)$ aos seus dois vizinhos, P_{i-1} e P_{i+1} , na sequência de pontos considerados. As coordenadas do centróide serão então dadas por:

$$(x_c, y_c) = \frac{\sum_{i=1}^n \left(\frac{1}{2} \times \left(\|\overrightarrow{P_{i-1}P_i}\| + \|\overrightarrow{P_iP_{i+1}}\| \right) \cdot (x_i, y_i) \right)}{\sum_{i=1}^n \|\overrightarrow{P_iP_{i+1}}\|},$$

onde $P_{n+1} = P_1$.

Note-se que determinar o centróide de um contorno fechado ordenado é algo diferente de determinar o centro de massa de um conjunto de pontos. Pelo método aqui definido, o centróide de um quadrado, por exemplo, será sempre bem calculado, independentemente dos seus lados poderem ser definidos por diferentes amostragens, o que não acontecia se estivéssemos a calcular o centróide como o centro de massa dos pontos, Figura 4.1.

Após a determinação dos centróides dos dois contornos, o vector translação é determinado pela diferença das coordenadas dos centróides dos mesmos.

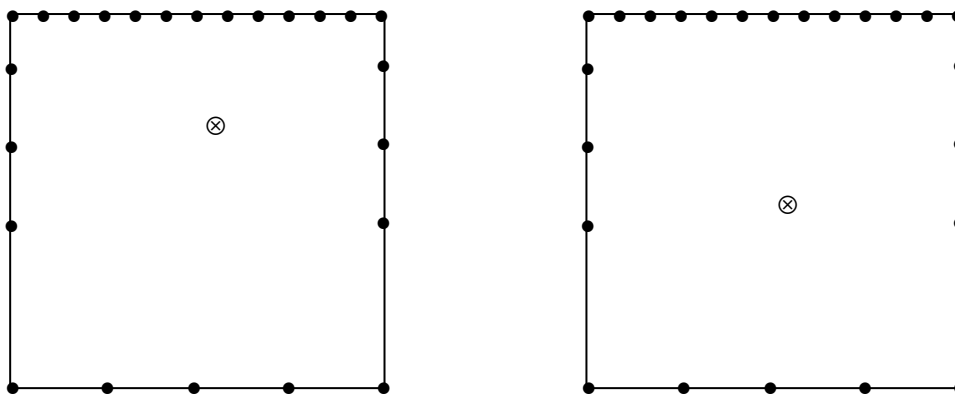


Figura 4.1: Dois contornos de dois quadrados exactamente iguais. À esquerda o centróide foi calculado como o centro de massa e à direita o centróide foi calculado pelo método proposto.

4.3 Escala

Para determinar a razão de semelhança (escala) entre dois contornos semelhantes foram inicialmente considerados três métodos. O primeiro calcularia a escala como a raiz quadrada da razão das áreas internas dos contornos. O segundo calcularia a escala como o quociente entre as medidas dos perímetros dos dois contornos. Finalmente, no terceiro método a escala seria calculada como o quociente entre a média ponderada das distâncias dos pontos do contorno ao seu centróide.

O primeiro método não foi implementado, pois a determinação da área dos contornos pode ser um problema algo complexo quando os contornos não são figuras convexas. O

segundo método é muito simples, pois basta somar as distâncias entre os pontos consecutivos para obter o perímetro de um contorno. Assim, após calcular o perímetro de cada contorno, o quociente entre eles dá a escala. Este método foi implementado e testado.

O terceiro método proposto também foi implementado. Este consiste basicamente no cálculo da média ponderada das distâncias dos pontos que definem o contorno ao seu centróide. Tal como para o cálculo do centróide indicado na secção 4.2, o peso de cada ponto para a média das distâncias ao centróide depende da incidência de pontos na sua vizinhança. Numa secção do contorno onde os pontos estejam muito próximos, cada um deles terá uma menor influência. Numa secção do contorno onde o espaçamento entre os pontos seja elevado, o peso de cada ponto será maior. Esta ponderação é calculada do mesmo modo que no caso do cálculo dos centróides na secção 4.2. Finalmente, a escala é determinada pelo quociente entre as médias ponderadas das distâncias aos centróides de cada contorno.

Formulemos matematicamente o método de determinação da escala baseado na distância ponderada dos pontos ao centróide. Sejam P_1, P_2, \dots, P_n a sequência de pontos que define um determinado contorno. Assim, tal como definido na secção anterior, temos que:

$$\frac{1}{2} \times \left(\left\| \overrightarrow{P_{i-1}P_i} \right\| + \left\| \overrightarrow{P_iP_{i+1}} \right\| \right),$$

define a média da distância do *ponto* $P_i = (x_i, y_i)$ aos seus dois vizinhos, P_{i-1} e P_{i+1} , na sequência de pontos. Seja $P_c = (x_c, y_c)$ o centróide do contorno definido pela sequência de pontos P_1, P_2, \dots, P_n e determinado pelo método proposto na secção anterior. Assim, a média ponderada das distâncias ao centróide, \bar{X} , é dada por:

$$\bar{X} = \frac{\sum_{i=1}^n \left(\frac{1}{2} \times \left(\left\| \overrightarrow{P_{i-1}P_i} \right\| + \left\| \overrightarrow{P_iP_{i+1}} \right\| \right) \times \left\| \overrightarrow{P_iP_c} \right\| \right)}{\sum_{i=1}^n \left\| \overrightarrow{P_iP_{i+1}} \right\|},$$

onde $P_{n+1} = P_1$. Deste modo, o quociente entre as distâncias médias ponderadas aos respectivos centróides representa a escala.

Pelo acabado de descrever, os dois métodos implementados para determinar a escala entre dois contornos semelhantes são insensíveis à amostragem dos contornos considerados.

Em [Tavares, 2000a], a escala é determinada pelo quociente da raiz quadrada dos desvios quadráticos das coordenadas dos dois objectos relativamente aos centróides correspondentes. Este processo é de fácil aplicação, no entanto, sofre de sensibilidade à amostragem, no sentido em que regiões onde a amostragem é mais fina, vão influenciar mais o valor final da escala do que regiões onde a amostragem é mais grosseira. No entanto, este método tem a vantagem de poder ser aplicado a qualquer conjunto de pontos ou contornos.

4.4 Ângulo de rotação

Para determinar o ângulo de rotação que melhor alinha dois contornos, é necessário que inicialmente sejam determinadas as correspondências entre os pontos que definem os contornos. Seguidamente, centram-se os dois contornos na origem e aplica-se a razão de semelhança (escala) previamente determinada a um dos contornos de modo que ambos fiquem com idêntica dimensão.

O problema agora consiste em encontrar um bom estimador para o ângulo de rotação envolvido, isto é, o ângulo que minimiza a distância entre pontos correspondidos. Este problema colocado assim globalmente é de difícil resolução, pois a função que representa a soma das distâncias entre os pontos correspondidos em função do ângulo de rotação poderá ter muitos mínimos locais, pelo que determinar o mínimo absoluto é um problema complexo. Para ultrapassar este problema, para cada par de pontos emparelhados, será calculado o ângulo de rotação que minimiza a distância entre os mesmos, isto é, considerando o ponto (x_i, y_i) do contorno 1 e o ponto (x_j, y_j) , o seu correspondente do contorno 2, pretende-se determinar β tal que $f(\beta) = (x_i - x_j(\beta))^2 + (y_i - y_j(\beta))^2$, seja mínimo, onde:

$$\begin{bmatrix} x_j(\beta) \\ y_j(\beta) \end{bmatrix} = \begin{bmatrix} \cos \beta & -\sin \beta \\ \sin \beta & \cos \beta \end{bmatrix} \begin{bmatrix} x_j \\ y_j \end{bmatrix}.$$

Estamos agora perante um problema de minimização de uma função de uma variável real com apenas um mínimo num qualquer intervalo de amplitude 2π , aberto num extremo e fechado no outro. Para determinar este mínimo recorreu-se a um algoritmo de minimização baseada na sequência de *Fibonacci*, [Fernandes, 1998].

Após o cálculo dos diferentes valores para os ângulos de rotação, um valor para cada par de pontos emparelhados, calcula-se a média e o desvio-padrão de todos os ângulos obtidos. Seguidamente, do conjunto de ângulos determinados, rejeitam-se os *outsiders* (neste caso, aqueles cuja distância à média seja superior ao desvio-padrão). Finalmente, volta-se a calcular a média dos ângulos de rotação sem os *outsiders*, a qual será considerada para valor final do ângulo de rotação.

No algoritmo implementado, conforme a situação, considera-se $\beta \in [0, 2\pi[$ ou $\beta \in [-\pi, \pi[$. Esta mudança do domínio da função a minimizar é importante, pois veja-se por exemplo a seguinte situação com $\beta \in [0, 2\pi[$:

- Imaginemos que obtemos os seguintes resultados para ângulos de rotação: o *ponto i* rodou $0,03 \text{ rad}$ (aproximadamente 2°) e o *ponto j* rodou $6,25 \text{ rad}$ (aproximadamente 358°). A média de rotação dá aproximadamente $\pi \text{ rad}$ (180°), o que é absurdo, pois em qualquer dos casos podemos considerar a rotação muito próxima de 0 rad . Com a mudança de domínio para $\beta \in [-\pi, \pi[$, o *ponto i* roda $0,03 \text{ rad}$ e o *ponto j* roda aproximadamente $-0,03 \text{ rad}$, sendo a média aproximadamente 0 rad , o que já faz sentido.

Para escolher o domínio da função que determina os sucessivos valores dos ângulos de rotação, inicialmente considera-se por defeito o domínio $[0, 2\pi[$. Após o cálculo de todos os valores para o ângulo de rotação, um valor por cada par de pontos emparelhados, verifica-se se a diferença entre o menor valor e o maior valor é superior a um determinado valor limite ψ . Apenas em caso afirmativo, o domínio da função de cálculo dos ângulos de rotação passa para $[-\pi, \pi[$, sendo necessário calcular novamente todos os valores dos diversos ângulos de rotação. Na implementação deste algoritmo que será utilizada no sexto capítulo, o valor limite considerado foi $\psi = \frac{4\pi}{3} \text{ rad}$, pois nos testes efectuados revelou sempre bons resultados. Finalmente, aplica-se a rotação determinada ao contorno que se deseja rodar, obtendo-se assim dois contornos com o mesmo centróide, com idênticas dimensões e alinhados em termos de rotação.

4.5 Resultados

Seguidamente, apresentam-se resultados de alguns ensaios experimentais realizados,

relativos à determinação da transformação rígida que melhor alinha dois contornos. Para determinar a escala foi utilizada a metodologia baseada na comparação da média ponderada da distância dos pontos do contorno ao seu centróide, apresentada na secção 4.3. Para aplicar a metodologia de determinação do ângulo de rotação proposta na secção anterior, é necessário um emparelhamento prévio. Assim, para esse efeito, recorreu-se à metodologia de emparelhamento baseada em informação de curvatura e de distância dos pontos aos respectivos centróides que será apresentada na secção 6.4.

Na Tabela 4.1 são apresentados os resultados obtidos em alguns ensaios realizados. Os contornos “*treea*”, “*treeb*”, “*treec*” e “*treed*” resultaram da aplicação da transformação rígida indicada na Tabela 4.1 ao contorno “*tree*”, sendo para esse efeito usada a plataforma computacional *CMIS*, [Tavares, 2000a, 2002, 2003]. Na Tabela 4.2 podem ser observados os pares de contornos a alinhar nas posições iniciais e os mesmos após alinhamento.

Para testar a robustez do algoritmo de determinação da transformação rígida desenvolvido, optou-se por provocar deformações locais retirando pontos de um dos contornos a alinhar. Esta opção deve-se ao facto de assim existir uma referência para os valores de rotação, escala e translação esperados.

Assim, os contornos “*treea3-1*”, “*treeb3-1*”, “*treec3-1*” e “*treed3-1*” resultam de retirar um ponto após cada grupo de três pontos aos contornos “*treea*”, “*treeb*”, “*treec*” e “*treed*”, respectivamente. Note-se que no total foram retirados aproximadamente 25% dos pontos. Os contornos “*treed2-1*”, “*treed1-2*”, “*treed1-3*” e “*treed1-4*” resultam também de retirar pontos ao contorno “*treed*”. Para se obter o contorno “*treed2-1*”, retirou-se um ponto a seguir a cada conjunto de dois pontos do contorno “*treed*”; para se obter o contorno “*treed1-2*”, retirou-se dois pontos a seguir a cada ponto do contorno “*treed*”; para se obter o contorno “*treed1-3*”, retirou-se três pontos a seguir a cada ponto do contorno “*treed*”; para se obter o contorno “*treed1-4*”, retirou-se quatro pontos a seguir a cada ponto do contorno “*treed*”. Pelo exposto, os contornos “*treed2-1*”, “*treed1-2*”, “*treed1-3*” e “*treed1-4*” têm aproximadamente 67%, 33%, 25% e 20% dos pontos do contorno “*treed*”; respectivamente. Os resultados obtidos podem ser observados nas Tabelas 4.1, 4.3 e 4.4.

Em todas as figuras apresentadas nas Tabelas 4.2, 4.3 e 4.4, a azul, em cima e em baixo, está representado o contorno original, a vermelho e em cima está o contorno após aplicação da transformação rígida e/ou deformação considerada, a vermelho e em baixo está o contorno transformado após aplicação da transformação rígida que melhor o alinha determinada usando a metodologia proposta neste capítulo.

Antes de analisar os resultados obtidos, é importante referir que a plataforma

computacional *CMIS*, após aplicação de uma transformação rígida a um contorno, arredonda os resultados para valores inteiros, pois trabalha em coordenadas *pixels*. Deste modo, pode ser introduzido um ligeiro erro nas coordenadas dos pontos aquando da aplicação da transformação rígida pretendida.

Tabela 4.1: Resultados relativos à transformação rígida existente entre dois contornos, determinada pela metodologia proposta.

Contornos		Transformação rígida aplicada			Transformação rígida determinada		
Original	Alterado	Rotação [°]	Escala	Translação (x, y) [pixel]	Rotação [°]	Escala	Translação (x, y) [pixel]
“tree”	“treea”	65,0	1,00	(0, 0)	65,0	1,00	(0, 1)
“tree”	“treeb”	0,0	1,30	(0, 0)	0,0	1,30	(0, 0)
“tree”	“treec”	0,0	1,00	(40, 40)	0,0	1,00	(40, 40)
“tree”	“treed”	120,0	1,40	(-30, 20)	120,0	1,40	(-32, 17)
“tree”	“treea3-1”	65,0	1,00	(0, 0)	65,1	1,01	(2, 1)
“tree”	“treeb3-1”	0,0	1,30	(0, 0)	0,0	1,31	(0, 1)
“tree”	“treec3-1”	0,0	1,00	(40, 40)	0,0	1,01	(40, 40)
“tree”	“treed3-1”	120,0	1,40	(-30, 20)	119,9	1,41	(-32, 17)
“tree”	“treed2-1”	120,0	1,40	(-30, 20)	119,5	1,41	(-34, 18)
“tree”	“treed1-2”	120,0	1,40	(-30, 20)	119,3	1,46	(-33, 18)
“tree”	“treed1-3”	120,0	1,40	(-30, 20)	120,3	1,46	(-35, 18)
“tree”	“treed1-4”	120,0	1,40	(-30, 20)	124,2	1,47	(-35, 15)

Tabela 4.2: Representação do alinhamento dos pares de contornos considerados nos ensaios nos quais foi aplicada apenas uma determinada transformação rígida.

Contornos		Contornos		Contornos		Contornos	
“tree”,	“treea”	“tree”,	“treeb”	“tree”,	“treec”	“tree”,	“treed”
N.º de pontos		N.º de pontos		N.º de pontos		N.º de pontos	
62	62	62	62	62	62	62	62

Tabela 4.3: Representação do alinhamento de pares de contornos considerados nos ensaios nos quais foi aplicada uma transformação rígida e uma deformação local reduzida.

Contornos		Contornos		Contornos		Contornos	
"tree",	"treea3-1"	"tree",	"treeb3-1"	"tree",	"treec3-1"	"tree",	"treed3-1"
N.º de pontos		N.º de pontos		N.º de pontos		N.º de pontos	
62	47	62	47	62	47	62	47

Tabela 4.4: Representação do alinhamento de pares de contornos considerados nos ensaios nos quais foi aplicada uma transformação rígida e uma deformação local mais acentuada.

Contornos		Contornos		Contornos		Contornos	
"tree"	"treed2-1"	"tree"	"treed1-2"	"tree"	"treed1-3"	"tree"	"treed1-4"
N.º de pontos		N.º de pontos		N.º de pontos		N.º de pontos	
62	42	62	21	62	16	62	13

Apresentam-se seguidamente as principais conclusões relativas aos vários testes

experimentais efectuados ao longo deste trabalho sobre a metodologia desenvolvida:

- Verificou-se que o método de determinação da escala com base na distância média ponderada ao centróide revelou-se mais adequado; especialmente para determinar os emparelhamentos com base na metodologia baseada na comparação da distância dos pontos que definem os contornos ao respectivo centróide, a apresentar no sexto capítulo.
- O processo de determinação da transformação rígida existente entre dois contornos representados por uma sequência de pontos ordenados revelou-se sempre eficaz. Quando os dois contornos a alinhar diferem entre si apenas de uma transformação rígida, a escala, a translação e rotação calculadas foram sempre adequadas.
- Se os contornos forem semelhantes e definidos por diferente número de pontos, a escala e translação continuam a ser adequadas. Neste caso, a determinação da rotação poderá ser sensível à qualidade dos emparelhamentos obtidos. Na maior parte dos casos, nos quais a qualidade dos emparelhamentos obtidos não foi satisfatória, a rotação determinada foi ainda de boa qualidade, consequência da eliminação dos *outsiders* e utilização de um valor médio para ângulo de rotação.
- Quando existe uma transformação local mais acentuada entre os dois contornos a alinhar, esta metodologia revelou ainda bons resultados. Nestes casos, a metodologia permitiu estimar adequadamente a essência da transformação rígida envolvida.

No sexto capítulo podem ser observados mais resultados relativos ao alinhamento de contornos, aquando da apresentação de metodologias de determinação de emparelhamento de contornos. Os ensaios a apresentar incluem diversos contornos onde a deformação envolvida é ainda mais acentuada do que a verificada nos exemplos apresentados nesta secção.

4.6 Sumário e conclusões

Neste capítulo foi apresentada uma metodologia para determinar a melhor transformação rígida que alinha dois contornos fechados definidos por um conjunto de pontos ordenados. No cálculo do centróide e razão de semelhança (escala) foi utilizada uma metodologia que

pondera o peso de cada ponto em função da incidência de pontos na sua vizinhança, conseguindo deste modo que estas duas grandezas sejam insensíveis à amostragem usada nos contornos envolvidos.

Para determinar a rotação, parte-se dos emparelhamentos obtidos por um qualquer método, excluindo os emparelhamentos cujos ângulos de rotação a eles associados se afastam da média um valor superior ao desvio-padrão.

Os vários ensaios experimentais realizados demonstraram que a metodologia desenvolvida se revela adequada, mesmo quando há deformação local acentuada entre os dois contornos envolvidos.

CAPÍTULO V:
DETERMINAÇÃO DE UM EMPARELHAMENTO ÓPTIMO ENTRE
DOIS CONTORNOS COM BASE NUMA MATRIZ DE CUSTOS

5.1 Introdução

O problema de optimização das correspondências entre pontos que definem dois contornos sofre de uma restrição importante: a monotonia da ordem relativa dos pontos emparelhados deve ser mantida para garantir a coerência da forma. Este problema é difícil de resolver, pois a ordem dos pontos não é absoluta.

Neste capítulo, vamos apresentar duas formulações capazes de encontrar uma correspondência óptima do tipo *um-para-um* entre os pontos que definem dois contornos a emparelhar, respeitando a ordem dos mesmos. Estas duas metodologias partem de dois pressupostos: os pontos dos contornos estão ordenados de forma que a união desses mesmos pontos pela ordem considerada defina o contorno; uma matriz de custos ou afinidade foi previamente calculada.

Caso os pontos não estejam ordenados, é necessário proceder à sua ordenação. Para tal, poderá ser utilizando o algoritmo baseado em *simulated annealing* apresentado na secção 3.6 seguido do algoritmo de verificação do sentido de definição do contorno apresentado na secção 3.7, procedendo-se à inversão do sentido de um dos contornos se necessário.

Na secção 5.2 far-se-á uma alusão ao problema da manutenção da ordem dos pontos emparelhados numa correspondência. Na secção seguinte será apresentada uma formulação de programação inteira capaz de resolver o problema referido. Esta formulação não foi implementada devido ao custo computacional esperado. Na secção 5.4 será apresentada uma formulação de programação dinâmica capaz de resolver o problema apresentado. Esta solução foi implementada em C++, [Rodrigues, 2003], tendo originado óptimos resultados em termos de velocidade de execução. Quanto à qualidade da correspondência global obtida, obviamente, em todos os casos testados foi sempre alcançada a correspondência global de menor custo que mantém a restrição da ordem dos pontos emparelhados.

Na secção 5.5 serão apresentados os resultados dos emparelhamentos entre vários pares de contornos partindo de matrizes de custo ou afinidade calculadas utilizando a metodologia baseada em modelação geométrica proposta por Shapiro, [Shapiro, 1992a, 1992b]. Para optimização das correspondências, serão utilizados o algoritmo de programação dinâmica desenvolvido ao longo desta Dissertação, apresentado na secção 5.4, e os algoritmos de afectação tradicionalmente usados para resolver este tipo de problemas, [Dell' Amico, 2000]: método *Húngaro*, [Hillier, 1995]; *Simplex* para problemas de fluxo, [Löbel, 2000], e algoritmo *LAPm*, [Volgenant, 1996]. Nos emparelhamentos a

considerar, será dada especial ênfase à comparação entre os resultados obtidos por estes algoritmos, quer em termos da qualidade dos emparelhamentos obtidos quer em termos de velocidade de execução dos mesmos.

Os ensaios foram realizados na plataforma computacional de desenvolvimento e ensaio *CMIS*, [Tavares, 2000a, 2002, 2003], pois nesta já se encontrava uma implementação dos algoritmos de afectação referidos e também metodologias de obtenção de uma matriz de afinidade baseadas em modelação física ou geométrica, [Tavares, 2000a], [Bastos, 2003]. Para comparar os dois tipos de métodos de optimização das correspondências, serão utilizadas as matrizes de afinidade obtidas usando a metodologia baseada em modelação geométrica proposta por Shapiro, [Shapiro, 1992a, 1992b]. A opção pela utilização da matriz de afinidade do modelo de Shapiro para comparar os algoritmos de optimização deve-se ao facto desta ter originado bons emparelhamentos na sua implementação na referida plataforma computacional; no entanto, qualquer outra matriz de custos poderia ser utilizada.

Para concluir o capítulo, na secção 5.6 serão apresentadas duas metodologias para fazer o emparelhamento do tipo *um-para-vários* ou *vários-para-um*, partindo inicialmente do emparelhamento do tipo *um-para-um*. A primeira metodologia faz os emparelhamentos minimizando os custos e a segunda metodologia faz os emparelhamentos minimizando a distância entre pontos.

5.2 Definição do problema

Começemos por definir o que nesta Dissertação significa ordem relativa e ordem absoluta dos pontos que definem um contorno. Consideremos a seguinte sequência de pontos: 1, 3, 4, 6, 7, 9. Esta sequência é monótona crescente. Observemos agora a sequência de pontos: 4, 6, 7, 9, 1, 3. Esta sequência não é monótona. No entanto, quando representamos os pontos que definem a primeira sequência sobre uma circunferência, verifica-se que a disposição relativa dos pontos é exactamente a mesma que se obteria se fossem representados os pontos da segunda sequência, considerando o mesmo sentido de rotação, Figura 5.1. Assim, diremos que a primeira sequência respeita a *ordem absoluta* e a segunda respeita apenas a *ordem relativa*. Obviamente, se uma sequência respeita a ordem absoluta também respeita a ordem relativa.

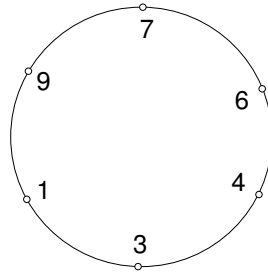


Figura 5.1: Sequência de pontos 1, 3, 4, 6, 7, 9 dispostos sobre uma circunferência.

Para ilustrar a complexidade do problema de emparelhar os pontos de dois contornos mantendo a ordem relativa dos pontos emparelhados, para evitar cruzamentos, comecemos por analisar os dois exemplos seguintes:

- a) Suponhamos que temos dois contornos, ambos definidos por 4 pontos cada um e ordenados de 1 a 4. Observem-se as duas correspondências seguintes, definidas por coluna:

$$f = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix} \text{ e } g = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 1 & 2 \end{pmatrix}.$$

A correspondência f satisfaz a ordem absoluta, mas a correspondência g não, pois a ordem dos pontos do segundo contorno, representados na segunda linha, não é absoluta. No entanto, a ordem relativa está correcta, pois a seguir ao *ponto 1* vem o *ponto 2*, a seguir ao *ponto 2* vem o *ponto 3* (considerando em círculo) e assim sucessivamente.

- b) Suponhamos agora que temos dois contornos, um definido por 4 pontos e o outro por 7 pontos, numerados de 1 a 4 e de 1 a 7, respectivamente. Observem-se as correspondências:

$$h = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 5 & 7 \end{pmatrix}, t = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 6 & 1 \end{pmatrix} \text{ e } p = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 6 & 7 & 4 & 5 \end{pmatrix}.$$

Todas respeitam a ordem relativa, mas apenas a correspondência h respeita a ordem absoluta.

Quando os contornos são definidos por igual número de pontos, o emparelhamento pode ser facilmente realizado. Basta observar que se o *ponto i* do *contorno 1* corresponder ao *ponto j* do *contorno 2*, então o *ponto i + 1* (neste caso $i + 1$ representa o ponto que vem

a seguir ao *ponto* i em termos de ordem relativa) de *contorno* 1 tem de corresponder ao *ponto* $j+1$ do *contorno* 2, e assim sucessivamente. Deste modo, considerando que ambos os contornos estão definidos por n pontos, há apenas n hipóteses de emparelhamento global mantendo a ordem relativa:

$$\begin{pmatrix} 1 & 2 & 3 & \dots & n \\ 1 & 2 & 3 & \dots & n \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 & \dots & n \\ 2 & 3 & 4 & \dots & 1 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 & \dots & n \\ 3 & 4 & 5 & \dots & 2 \end{pmatrix}, \dots, \begin{pmatrix} 1 & 2 & 3 & \dots & n \\ n & 1 & 2 & \dots & n-1 \end{pmatrix}.$$

Atendendo ao referido, basta calcular o custo de cada uma das n correspondências globais e depois escolher uma que origine um custo mínimo.

Para contornos definidos por diferente, ou igual, número de pontos, serão apresentadas seguidamente duas formulações: uma de programação inteira e outra de programação dinâmica, ambas capazes de encontrar o melhor emparelhamento global mantendo a ordem absoluta dos pontos emparelhados. Depois, através da reordenação dos pontos, conseguir-se-á determinar o melhor emparelhamento global respeitando a ordem relativa.

5.3 Formulação de programação inteira

Suponhamos, sem perda de generalidade, que temos dois contornos, o *contorno* 1 definido por n pontos e o *contorno* 2 definido por m pontos, com $n \leq m$. Após aplicação de uma “métrica” no sentido de determinar a “proximidade” (função que indica os custos de emparelhamento entre cada par de pontos) entre os pontos que definem os dois contornos, obtém-se a matriz $C = [c_{ij}]$ representativa desses custos, onde c_{ij} representa o custo de emparelhar o *ponto* i do *contorno* 1 com o *ponto* j do *contorno* 2.

Vamos impor que a correspondência do *contorno* 1 para o *contorno* 2 seja uma função injectiva, isto é, cada ponto do *contorno* 1 corresponde a um único ponto do *contorno* 2 e dois pontos distintos do *contorno* 1 têm que corresponder dois pontos distintos do *contorno* 2. Deste modo, estamos a impor uma correspondência do tipo *um-para-um*. Se $n = m$, a correspondência será sobrejectiva, caso contrário não será.

Seja $X = [x_{ij}]$ uma matriz de variáveis de decisão binárias, em que:

$$x_{ij} = \begin{cases} 1, & \text{se o ponto } i \text{ do contorno 1 corresponde ao ponto } j \text{ do contorno 2;} \\ 0, & \text{caso contrário.} \end{cases}$$

Podemos agora definir a função objectivo a minimizar:

$$f = \sum_{i=1}^n \sum_{j=1}^m x_{ij} c_{ij}. \quad (5.1)$$

Passemos às restrições. Como cada ponto do *contorno 1* tem que corresponder a um único ponto do *contorno 2*, temos:

$$\sum_{j=1}^m x_{ij} = 1, \quad i = 1, 2, \dots, n. \quad (5.2)$$

Por outro lado, como cada ponto do *contorno 2* corresponde a um ou nenhum ponto do *contorno 1*, tem-se que:

$$\sum_{i=1}^n x_{ij} \leq 1, \quad j = 1, 2, \dots, m. \quad (5.3)$$

As condições já impostas garantem a obtenção de uma correspondência global de custo mínimo, mas não garantem a ordem absoluta nem relativa dos pontos correspondidos.

A formulação clássica dos algoritmos de afectação é ligeiramente diferente desta, pois são definidos pontos fictícios no contorno constituído por menor número de pontos, sendo 0 (zero) o custo de os emparelhar com um ponto do outro contorno. Deste modo, a matriz de custos passa a ser quadrada e o problema passa a ser um problema de afectação clássico. Na formulação aqui apresentada, a técnica dos pontos fictícios não pode ser utilizada, pois estes pontos não respeitarão qualquer ordem nos emparelhamentos.

Para garantir a ordem absoluta dos emparelhamentos na formulação apresentada, ainda falta acrescentar as seguintes $n - 1$ restrições lineares à formulação inicial:

$$\sum_{j=1}^m jx_{1j} < \sum_{j=1}^m jx_{2j} < \dots < \sum_{j=1}^m jx_{nj}. \quad (5.4)$$

Com estas últimas condições, garante-se que se o *ponto i* do *contorno 1* corresponder com o *ponto j* do *contorno 2*, então o *ponto i + 1* do *contorno 1* tem que corresponder com um *ponto j + k* do *contorno 2*, com $k \geq 1$.

Para melhor explicar a acção destas novas restrições, vamos recorrer a um exemplo: Suponhamos que $x_{36} = 1$, isto é, o *ponto 3* do *contorno 1* corresponde ao *ponto 6* do *contorno 2*. Atendendo que cada ponto do *contorno 1* corresponde a um só ponto do

contorno 2, condição (5.2), tem-se que $x_{3j} = 0$ para todo $j \neq 6$. Assim, temos:

$$\sum_{j=1}^m jx_{3j} = 6.$$

Suponhamos, por absurdo, que $x_{45} = 1$, isto é, o ponto 4 do contorno 1 corresponde ao ponto 5 do contorno 2 (note-se que o ponto 4 do contorno 1 vem a seguir ao ponto 3, portanto deveria corresponder a um ponto do contorno 2 que viesse a seguir ao ponto 6). Então, pelo mesmo argumento usado anteriormente, tem-se:

$$\sum_{j=1}^m jx_{4j} = 5.$$

Mas neste caso não se verifica a condição (5.4), pois temos:

$$\sum_{j=1}^m jx_{3j} < \sum_{j=1}^m jx_{4j} \Leftrightarrow 6 < 5,$$

que é obviamente falso. Esta situação hipotética apresentada levaria à definição de uma correspondência do tipo da seguinte, que obviamente não respeita a ordem absoluta:

$$\begin{pmatrix} 1 & 2 & 3 & 4 & \dots \\ \dots & \dots & 6 & 5 & \dots \end{pmatrix}.$$

Para melhorar a eficiência do algoritmo, muitas das variáveis de decisão podem ser consideradas nulas, como resultado da restrição dada pela condição (5.4). Para uma matriz de custos de dimensão $n \times m$ com $n \leq m$, pode-se definir antecipadamente:

$$x_{1j} = 0, \text{ com } j = m - n + 2, \dots, m; \quad (5.5)$$

$$x_{21} = x_{2j} = 0, \text{ com } j = m - n + 3, \dots, m; \quad (5.5)$$

$$x_{31} = x_{32} = x_{3j} = 0, \text{ com } j = m - n + 4, \dots, m; \quad (5.5)$$

(...);

$$x_{nj} = 0, \text{ com } j = 1, \dots, n - 1. \quad (5.5)$$

Assim, tem-se no total $n \times (m - n + 1)$ variáveis de decisão em vez das $n \times m$ que se teria inicialmente. A título de exemplo, a matriz das variáveis de decisão para uma matriz

de custos de dimensão 4×6 será:

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} & 0 & 0 & 0 \\ 0 & x_{22} & x_{23} & x_{24} & 0 & 0 \\ 0 & 0 & x_{33} & x_{34} & x_{35} & 0 \\ 0 & 0 & 0 & x_{44} & x_{45} & x_{46} \end{bmatrix}.$$

A formulação anterior garante a obtenção da melhor correspondência global que mantém a ordem absoluta dos pontos emparelhados. No entanto, existem muitas correspondências globais que respeitam a ordem relativa que não estão a ser consideradas. Assim, para determinar a correspondência global ótima que respeita a ordem relativa, vamos supor três situações:

- a) Suponhamos que é conhecido um emparelhamento entre dois pontos, por exemplo, o *ponto i* do *contorno 1* corresponde ao *ponto j* do *contorno 2*. Então, reordenam-se os pontos dos contornos, sendo agora o *ponto 1* do *contorno 1* aquele que antes era o *ponto i*, o *ponto 2* aquele que antes era o *ponto i + 1* (note-se que $i + 1$ representa a ordem relativa) e assim sucessivamente. Procede-se do mesmo modo para o *contorno 2*, isto é, o *ponto j* passa a ser o *ponto 1*, o *ponto j + 1* passa a ser o *ponto 2*, e assim sucessivamente. Agora, basta aplicar a formulação anteriormente definida. Mas, para mais rápida resolução, pode-se fazer:

$$x_{11} = 1 \text{ e } x_{1j} = 0 \text{ para } j = 2, \dots, m. \quad (5.6)$$

Assim, retiram-se $m - n + 1$ variáveis. Também se podem retirar as condições:

$$\sum_{j=1}^m x_{1j} = 1, \sum_{i=1}^n x_{i1} \leq 1 \text{ e } \sum_{j=1}^m jx_{1j} < \sum_{j=1}^m jx_{2j}, \quad (5.7)$$

pois são automaticamente verdadeiras em função da definição dos valores das variáveis da expressão (5.6).

Tem-se assim no total $(n - 1) \times (m - n + 1)$ variáveis de decisão e $n + m + (n - 1) - 3$ restrições. Para o exemplo da matriz de custos anterior, a matriz das variáveis de decisão será:

$$X = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & x_{22} & x_{23} & x_{24} & 0 & 0 \\ 0 & 0 & x_{33} & x_{34} & x_{35} & 0 \\ 0 & 0 & 0 & x_{44} & x_{45} & x_{46} \end{bmatrix}.$$

- b) Suponhamos que é conhecido um conjunto de fortes candidatos a emparelhamento. Então, para cada par de candidatos a emparelhamento, aplica-se a formulação definida em a). De todas as soluções obtidas, uma para cada par de candidatos, escolhe-se a de menor custo total.
- c) Finalmente, na pior das hipóteses, suponhamos que não é conhecido qualquer conjunto de fortes candidatos a emparelhamento. Então, será necessário resolver m problemas (tantos quantos os pontos do contorno com mais pontos). Para o primeiro problema, supõe-se que o *ponto 1* do *contorno 1* corresponde ao *ponto 1* do *contorno 2*; para o segundo problema, supõe-se que o *ponto 1* do *contorno 1* corresponde ao *ponto 2* do *contorno 2*; para o terceiro problema, supõe-se que o *ponto 1* do *contorno 1* corresponde ao *ponto 3* do *contorno 2*, e assim sucessivamente. Para resolver cada um destes problemas, utiliza-se a formulação definida em a). Finalmente, de entre todas as m soluções encontradas, escolhe-se uma que corresponda ao custo mínimo.

Repare-se que cada nova ordem absoluta determinada vai corresponder a uma ordem relativa, relativamente à ordem inicial dos pontos. Assim, determinam-se todas as ordens relativas do segundo contorno e conseqüentemente todos os emparelhamentos globais de custo mínimo que respeitam essas sucessivas ordens.

Esta formulação não foi implementada, pois, na pior das hipóteses, será necessário resolver m problemas de minimização, sendo cada um destes computacionalmente bastante dispendioso, pois cada um requerer $(n-1) \times (m-n+1)$ variáveis de decisão e $n+m+(n-1)-3$ restrições, como calculado em a). Além disso, se for utilizado o algoritmo *simplex*, o problema não pode ser relaxado como se de um problema de variáveis contínuas se tratasse, pois a matriz das restrições não é totalmente unimodular, portanto, não está garantida a convergência para uma solução inteira. Deste modo, será necessário aplicar, conjuntamente com o algoritmo *simplex*, a técnica de *branch-and-bound* ou de corte fraccionário, por exemplo, o que ainda trará maiores custos computacionais.

5.4 Formulação de programação dinâmica

5.4.1 Formulação geral

Começamos esta secção com um exemplo simples, o qual será considerado ao longo de toda a secção. Suponhamos que temos o *contorno 1* e o *contorno 2* definidos por 4 e 6 pontos respectivamente e a seguinte matriz de custos C :

$$C = \begin{bmatrix} 1 & 0 & 1 & 4 & 5 & 1 \\ 0 & 3 & 1 & 5 & 2 & 1 \\ 6 & 1 & 2 & 4 & 0 & 8 \\ 3 & 2 & 7 & 5 & 4 & 1 \end{bmatrix},$$

onde c_{ij} representa o custo de emparelhar o *ponto i* do *contorno 1* com o *ponto j* do *contorno 2*.

Para evitar emparelhamentos cruzados, vamos impor obrigatoriedade de preservar a ordem absoluta dos pontos emparelhados. Deste modo, impomos a monotonia da sequência de emparelhamentos, isto é, se o *ponto i* do *contorno 1* corresponde ao *ponto j* do *contorno 2*, então o *ponto i + 1* do *contorno 1* tem que corresponder a um *ponto j + k* do *contorno 2* com k inteiro e $k \geq 1$. Assim, temos, por exemplo, de entre outras, as seguintes correspondências válidas:

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 3 & 4 & 5 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 6 \end{pmatrix} \text{ e } \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 5 & 6 \end{pmatrix},$$

com os respectivos custos: 11, 10, 6 e 7.

No total, para as hipóteses impostas, temos exactamente 15 correspondências possíveis, pois contar as hipóteses de emparelhamento é equivalente a contar quantos subconjuntos de quatro elementos distintos se conseguem formar com os 6 elementos do *contorno 2*. Portanto, o número de correspondências globais que respeitam a ordem absoluta é:

$$C_4^6 = \frac{6!}{(6-4)!2!} = 15.$$

De uma forma geral, se um contorno é definido por n pontos e o outro m pontos, com $n \leq m$, há exactamente C_n^m (combinações) hipóteses de emparelhamento respeitando a

ordenação absoluta. Respeitando a ordenação relativa, há exactamente mC_n^m hipóteses, pelas razões já explicadas na secção anterior.

Usando uma notação usual em programação dinâmica, [Norman, 1975] e [Winston, 1994], para o exemplo em estudo apresentado, vamos definir 4 estágios. No estágio 1, escolhe-se o emparelhamento de menor custo para o *ponto 1* do *contorno 1*, atendendo às hipóteses de emparelhamento. No estágio 2, escolhe-se o melhor emparelhamento para o *ponto 2* do *contorno 1*, com base nas hipóteses de emparelhamento e custos derivados do estágio 1, e assim sucessivamente. É fundamental referir que a definição de um emparelhamento entre dois pontos num determinado estágio, irá afectar as hipóteses de emparelhamento nos estágios seguintes.

Para ajudar a compreender a situação anteriormente descrita, vejamos o seguinte. No exemplo apresentado, para ser mantida a ordem absoluta e cada ponto do *contorno 1* corresponder a um ponto distinto do *contorno 2*, o *ponto 1* do *contorno 1* pode emparelhar apenas com os *pontos 1, 2* ou *3* do *contorno 2*, mas, por exemplo, se o *ponto 1* do *contorno 1* emparelhar com o *ponto 3* do *contorno 2*, então o *ponto 2* do *contorno 1* só poderá emparelhar com o *ponto 4* do *contorno 2*. Assim, de um modo geral, conforme os emparelhamentos já efectuados nos estágios anteriores, num determinado estágio k , o *ponto k* do *contorno 1* vai emparelhar com apenas um ponto dos seguintes conjuntos de pontos do *contorno 2*: $\{k\}$, $\{k, k + 1\}$ ou $\{k, k + 1, k + 2\}$.

Para indicar se um ponto do *contorno 1* tem 1, 2, ou 3 pontos do *contorno 2* disponíveis para emparelhamento, vamos definir a variável de estado s . Temos, assim, que $s \in \{1, 2, 3\}$. Se num determinado estágio k se tem $s = 1$, então o *ponto k* do *contorno 1* só tem uma opção de emparelhamento (com o *ponto k* do *contorno 2*); se $s = 2$, então o *ponto k* do *contorno 1* tem duas hipóteses de emparelhamento (com os *pontos k* ou $k + 1$ do *contorno 2*), e assim sucessivamente.

Definamos agora a função do custo mínimo $f_k(s)$, onde a variável s é a variável de estado já definida, k representa o estágio e $f_k(s)$ representa o custo mínimo para corresponder os *pontos 1, 2, 3, ..., k* do *contorno 1* quando o *ponto k* do *contorno 1* tem s hipóteses de emparelhamento à escolha.

No exemplo em estudo apresentado tem-se que, por exemplo, $f_3(2)$ representa o custo de emparelhar os *pontos 1, 2* e *3* do *contorno 1* quando o *ponto 3* do *contorno 1* tem apenas dois candidatos do *contorno 2* para emparelhamento (os *pontos 3* ou *4* do *contorno 2*).

Para melhor elucidar, vamos aplicar esta formulação ao exemplo em estudo. Vamos construir sucessivamente uma correspondência ótima que respeita a ordem absoluta dos pontos. À esquerda colocaremos os custos para cada estágio em função do estado e da matriz de custos C e à direita os emparelhamentos definidos:

$$\begin{aligned}
 f_1(1) &= c_{11} = 1 && \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\
 f_1(2) &= \min\{c_{11}, c_{12}\} = 0 && \begin{pmatrix} 1 \\ 2 \end{pmatrix} \\
 f_1(3) &= \min\{c_{11}, c_{12}, c_{13}\} = 0 && \begin{pmatrix} 1 \\ 2 \end{pmatrix} \\
 f_2(1) &= c_{22} + f_1(1) = 3 + 1 = 4 && \begin{pmatrix} 1 & 2 \\ 1 & 2 \end{pmatrix} \\
 f_2(2) &= \min\{c_{22} + f_1(1), c_{23} + f_1(2)\} = 1 && \begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix} \\
 f_2(3) &= \min\{c_{22} + f_1(1), c_{23} + f_1(2), c_{24} + f_1(3)\} = 1 && \begin{pmatrix} 1 & 2 \\ 2 & 3 \end{pmatrix} \\
 f_3(1) &= c_{33} + f_2(1) = 2 + 4 = 6 && \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix} \\
 f_3(2) &= \min\{c_{33} + f_2(1), c_{34} + f_2(2)\} = 5 && \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \end{pmatrix} \\
 f_3(3) &= \min\{c_{33} + f_2(1), c_{34} + f_2(2), c_{35} + f_2(3)\} = 1 && \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 5 \end{pmatrix} \\
 f_4(1) &= c_{44} + f_3(1) = 11 && \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{pmatrix} \\
 f_4(2) &= \min\{c_{44} + f_3(1), c_{45} + f_3(2)\} = 9 && \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \end{pmatrix} \\
 f_4(3) &= \min\{c_{44} + f_3(1), c_{45} + f_3(2), c_{46} + f_3(3)\} = 2 && \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 5 & 6 \end{pmatrix}
 \end{aligned}$$

Como no total há 4 estágios, no quarto estágio não há restrições, razão pela qual não seria necessário calcular $f_4(1)$ e $f_4(2)$ se apenas se pretendesse o custo mínimo, mas como é necessário guardar informação relativa à correspondência, tal tem de ser feito. Tem-se, assim, que o custo mínimo para emparelhar os 4 pontos do *contorno 1* com 4 pontos do *contorno 2*, respeitando a ordem absoluta dos pontos, é 2 e a correspondência óptima que respeita a ordem absoluta é a última assinalada.

De uma forma geral, para uma matriz de custos C de dimensão $n \times m$ com $n \leq m$, $k \leq n$ e $s \in \{1, 2, \dots, m - n + 1\}$, $f_k(s)$ representa o custo mínimo de corresponder os pontos $1, 2, \dots, k$ do *contorno 1*, quando o ponto k tem s hipóteses de emparelhamento. Com esta formulação garante-se a obtenção do melhor emparelhamento global mantendo a ordem absoluta. Para se obter o melhor emparelhamento global mantendo a ordem relativa, pode-se proceder como descrito na secção anterior, sendo na pior das hipóteses necessário resolver m problemas de emparelhamento. No caso do exemplo em estudo considerado, a correspondência de custo mínimo que respeita a ordem relativa dos pontos é ainda a anteriormente apresentada.

5.4.2 Algoritmo e implementação

Antes de apresentarmos o algoritmo, observemos novamente o exemplo apresentado na subsecção anterior. Neste, tem-se, por exemplo que:

$$f_3(3) = \min\{c_{33} + f_2(1), c_{34} + f_2(2), c_{35} + f_2(3)\}.$$

Aparentemente, temos que calcular três valores e depois compará-los no sentido de escolher o menor. Tal não é necessariamente obrigatório, pois os valores $c_{33} + f_2(1)$ e $c_{34} + f_2(2)$ já foram calculados e $c_{34} + f_2(2) \leq c_{33} + f_2(1)$. Pelo que basta calcular o valor de $c_{35} + f_2(3)$ e compará-lo com $c_{34} + f_2(2)$. Assim, em cada estágio apenas é efectuada uma operação de soma e uma de comparação para cada estado, se $s > 1$.

O algoritmo a seguir apresentado parte do pressuposto que não é conhecido *a priori* um emparelhamento nem um conjunto de fortes candidatos a emparelhamento, pelo que vai determinar todos os emparelhamentos globais possíveis que respeitam a ordem relativa e posteriormente escolher o melhor deles.

Algoritmo:

1. Ler a dimensão dos contornos, ler a matriz C , definir o valor de n e m de modo que $n \leq m$. Se necessário ($n > m$) determinar a transposta da matriz de custos, C .
2. Repetir m vezes:
 - i. Para $k = 1, 2, \dots, n$ e $s = 1, 2, \dots, m - n + 1$, calcular os respectivos valores $f_k(s)$, tendo em consideração o referido anteriormente para que não sejam repetidos cálculos. Guardar os sucessivos valores de $f_k(s)$ numa tabela de n linhas por $m - n + 1$ colunas, ou seja, tantas linhas quantos os estágios e tantas colunas quantos os estados possíveis (ver Tabela 5.1).
 - ii. Determinar o custo mínimo, o qual corresponde ao valor guardado na posição $(n, m - n + 1)$ da tabela de valores (no exemplo em estudo, é o valor guardado na posição $(4, 3)$ da Tabela 5.1).
 - iii. Definir a correspondência global de custo mínimo, fazendo uma pesquisa na tabela construída. Começa-se pela última linha (índice n), seleccionando, de entre todas as células de valor mínimo, a célula que corresponder à coluna de menor índice (a que está mais à esquerda). Seguidamente, passa-se para a penúltima linha (índice $n - 1$); seleccionando-se, de entre todas as células de índice de coluna menor ou igual ao índice de coluna da célula seleccionada no passo anterior e de valor mínimo para esse grupo de células, a célula que corresponder à coluna de menor índice. Continua-se este processo até percorrer todas as linhas.
 Note-se que, a selecção de uma determinada célula (i, j) significa que o ponto i do contorno 1 vai corresponder o ponto $i + j - 1$ do contorno 2 (veja-se, na Tabela 5.1, as células utilizadas para definir os emparelhamentos no exemplo em estudo).
 - iv. Reordenar as colunas da matriz C de modo que agora a primeira coluna passa a ser a que antes era a segunda, a segunda coluna passa a ser a que antes era a terceira e assim sucessivamente.
3. Procurar, de entre as m correspondências determinadas, a de menor custo e os respectivos emparelhamentos.

Na Tabela 5.1 estão representados os valores de $f_k(s)$ para o primeiro dos 6 problemas de minimização que o algoritmo de programação dinâmica tem que resolver, para o exemplo em estudo apresentado. As células destacadas são as utilizadas pelo algoritmo para definir a correspondência de custo mínimo no caso do primeiro dos 6 problemas considerados.

Tabela 5.1: Custos mínimos guardados pelo algoritmo de programação dinâmica em função do estágio e do estado para o primeiro problema de minimização relativo ao exemplo em estudo.

Estados (s) \ Estágios (k)	1	2	3
1	$f_1(1) = 1$	$f_1(2) = 0$	$f_1(3) = 0$
2	$f_2(1) = 4$	$f_2(2) = 1$	$f_2(3) = 1$
3	$f_3(1) = 6$	$f_3(2) = 5$	$f_3(3) = 1$
4	$f_4(1) = 11$	$f_4(2) = 9$	$f_4(3) = 2$

5.4.3 Custo computacional

Considerando um contorno definido por n pontos e o outro definido por m pontos com $n \leq m$, para cada emparelhamento global respeitando a ordem absoluta tem-se n estágios e $m - n + 1$ estados. Contabilizando apenas as operações definidas no algoritmo, para cada estágio é efectuada uma soma por estado e para cada estado maior do que 1 só é efectuada uma comparação. Se a variável de estado for 1, não é efectuada qualquer comparação. Assim, tem-se no total $n \times (m - n + 1)$ somas e $n \times (m - n)$ comparações.

Para obter o melhor emparelhamento global respeitando a ordem relativa tem-se que resolver m problemas, portanto $m \times n \times (m - n + 1)$ somas e $m \times n \times (m - n)$ comparações. Para escolher a melhor correspondência global de entre todas as globais determinadas, tem-se mais $m - 1$ comparações.

Consideremos a expressão relativa ao número de somas anteriormente apresentada, pois é a que origina um maior número de operações. Fixando m , facilmente se conclui que o número mínimo de somas será atingido com $n = 1$ (note que n é um número natural) e

com $n = m$. Substituindo n por 1 ou por m na expressão referida, obtém-se o valor m^2 para o número total de somas. Por outro lado, temos que:

$$m \times n \times (m - n + 1) = -m n^2 + (m^2 + m)n.$$

Considerando o segundo membro da expressão anterior como um polinómio sobre a variável n , e fixando m , temos que este polinómio atinge o máximo para $n = \frac{m-1}{2}$.

Portanto, o número máximo de somas do algoritmo é:

$$m \times n \times (m - n + 1) = m \times \frac{m-1}{2} \times \left(m - \frac{m-1}{2} + 1 \right) = \frac{m^3 + 2m^2 - 3m}{4}.$$

Assim, pelo acabado de expor, a ordem de complexidade deste algoritmo será m^2 , se $n = m$, e inferior a m^3 se $n < m$. Com estes dados, pode-se concluir que o tempo de execução será tanto maior quanto maior for o número de pontos que definem os contornos e tanto menor quanto menor for a diferença entre o número de pontos que definem os dois contornos.

Como exemplo, para emparelhar os pontos de dois contornos de 57 e 86 pontos, respectivamente, partindo de uma matriz de custos previamente calculada, são necessárias no total 147060 somas e 142243 comparações. Este valor é pouco significativo para os computadores actuais. A título de exemplo, na sua implementação na plataforma computacional *CMIS* a correr num computador equipado com um processador *Intel Pentium III* a 1.0 GHz com 256MB de RAM, para efectuar este cálculo e procurar na tabela a correspondência global óptima que respeita a ordem relativa foi dispendido um tempo de 0,01 segundos. Este exemplo e outros poderão ser observados na secção seguinte.

5.5 Algoritmo de programação dinâmica *versus* algoritmos de afectação clássicos

5.5.1 Metodologia utilizada para obtenção de uma matriz de custos ou afinidade

Nesta subsecção será brevemente apresentada a metodologia baseada em modelação geométrica proposta por Shapiro, [Shapiro, 1992a, 1992b], pois esta será utilizada na obtenção das matrizes de afinidade a utilizar nos ensaios que se seguirão. Tal deve-se ao

facto de que a plataforma computacional *CMIS*, já referida, ter uma implementação da mesma. No entanto, poderia ser utilizada outra qualquer metodologia de obtenção de uma matriz de custo de emparelhamento entre os pontos de ambos os contornos.

Passemos então à apresentação da metodologia baseada em modelação geométrica proposta por Shapiro. Esta metodologia pode ser utilizada para determinar uma matriz de afinidade entre qualquer par de formas definidas por um conjunto de dados pontuais, independentemente desses conjuntos definirem contornos ou não. No entanto, como nos ensaio que serão realizados apenas serão utilizados contornos definidos por dados pontuais, iremos referir-nos às formas a emparelhar apenas como contornos.

Consideremos, então, *contorno 1* e o *contorno 2* definidos por n e m pontos, respectivamente. Para cada um dos contornos, calculemos as respectivas matrizes de proximidade H_1 e H_2 , respectivamente, onde cada elemento h_{ij} de cada matriz representa a distância *Gaussiana* entre o *ponto i* e o *ponto j* do respectivo contorno, calculada do seguinte modo:

$$h_{ij} = e^{-\frac{\|\overrightarrow{X_i X_j}\|^2}{2\sigma_x^2}},$$

onde $\|\overrightarrow{X_i X_j}\|$ representa a distância euclidiana entre o *ponto i* e o *ponto j* do contorno em questão. O parâmetro σ_x ($\sigma_x > 0$) funciona como um filtro. Valores de σ_x elevados fazem com que os pontos vizinhos percam influência uns sobre os outros. Assim, as características locais do contorno perdem influência em detrimento das características globais do mesmo. Por outro lado, valores de σ_x baixos fazem precisamente o contrário, isto é, aumenta a influência das características locais do contorno e diminui a influência das características globais.

Para cada uma das matrizes de proximidade calculadas, é determinado o conjunto dos seus valores próprios e um respectivo conjunto de vectores próprios unitários associados aos respectivos valores próprios, chamados de *modos*. Seguidamente, os valores próprios são ordenados por ordem decrescente do seu valor absoluto, ordenando-se também os respectivos vectores próprios que lhes estão associados.

O passo seguinte consiste na determinação da matriz de correlação Z (também chamada matriz de afinidade) entre as componentes que definem os vectores próprios de ambos os contornos. Para determinar a correlação entre os contornos não é necessário utilizar todos

os vectores próprios de ambas as matrizes de proximidade, basta considerar uma percentagem dos mesmos, [Tavares, 2000a]. Cada elemento z_{ij} da matriz Z representa a “afinidade” entre *ponto* i do *contorno* 1 e o *ponto* j do *contorno* 2. Na implementação efectuada, z_{ij} é calculado do seguinte modo:

Consideremos que serão utilizados apenas k vectores próprios de cada uma das matrizes de proximidade H_1 e H_2 . Sejam T a matriz em que as colunas são definidas pelos k primeiros vectores próprios da matriz H_1 e U a matriz em que as linhas são constituídas pelos k primeiros vectores próprios da matriz H_2 . Em ambas as matrizes, T e U , é respeitada a ordenação dos vectores próprios previamente definida em função do valor dos valores próprios que lhe estão associados. Assim, temos:

$$z_{ij} = \sum_{p=1}^k |t_{ip} - u_{pj}|$$

onde t_{ip} e u_{pj} são elementos das matrizes T e U , respectivamente.

Atendendo que os vectores próprios utilizados são unitários, tem-se que $0 \leq z_{ij} \leq 2$. Valores de z_{ij} próximos de 0 (zero) indicam uma grande afinidade modal entre o *ponto* i do *contorno* 1 e o *ponto* j do *contorno* 2. Valores de z_{ij} próximos de 2 indicam exactamente o contrário. Dado que associado a um valor próprio de uma matriz existem dois vectores próprios unitários e simétricos entre si, na implementação efectuada, está incorporado um algoritmo de escolha do sinal dos vectores próprios, com o intuito de aumentar a afinidade global entre os pontos de ambas as formas.

Em [Tavares, 2000a], a solução considerada para determinar os emparelhamentos tinha um carácter puramente local, no sentido em que dois pontos só eram correspondidos se, para cada um deles, o outro era o que lhe estava mais próximo em termos de custo de emparelhamento. Deste modo, acontecia frequentemente que alguns pontos não eram correspondidos e por vezes apareciam emparelhamentos cruzados, Figura 5.2.

Em [Bastos, 2003] foram implementados na plataforma computacional *CMIS* três algoritmos baseados em modelos de afectação tradicionalmente utilizados neste tipo de situação, [Dell’ Amico, 2000]: método *Húngaro* [Hillier, 1995]; *Simplex* para problemas de fluxo, [Löbel, 2000]; e algoritmo *LAPm*, [Volgenant, 1996]. O objectivo era aumentar o número de emparelhamentos minimizando o custo global dos mesmos, definido o custo global como a soma dos valores da matriz de afinidade Z associados a cada

emparelhamento. O problema de determinar o melhor emparelhamento global foi formulado como um problema de afectação clássico, não sendo considerada qualquer restrição de ordem dos pontos emparelhados. Os resultados obtidos, aquando da aplicação destes algoritmos às matrizes de afinidade calculadas recorrendo às modelações físicas ou modelações geométricas implementadas na plataforma computacional *CMIS*, melhoraram significativamente quando comparados com os obtidos com a anterior metodologia de determinação das correspondências baseada em critérios puramente locais, [Bastos, 2003, 2006]. No entanto, continuaram a aparecer, por vezes, emparelhamentos cruzados em alguns ensaios, Figura 5.3.

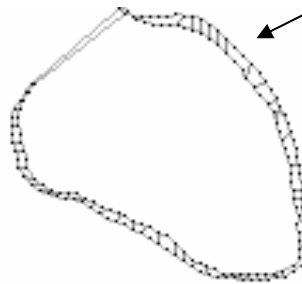


Figura 5.2: Emparelhamento entre os contornos “heart5” e “heart6”, sem optimização global das correspondências. Os contornos são definidos por 81 e 83 pontos, respectivamente. (A seta representada assinala uma região onde foram deixados pontos por corresponder.)



Figura 5.3: Emparelhamento dos contornos da Figura 5.2 com optimização global das correspondências usando o algoritmo *Simplex* para problemas de fluxo. (A seta representada assinala uma região de emparelhamentos cruzados.)

Pormenores dos algoritmos e critérios de escolha do valor de σ_X , número de modos a utilizar e algoritmo de escolha do sinal dos vectores próprios podem ser consultados em [Tavares, 2000a]. Pormenores sobre a implementação dos algoritmos de optimização e respectivos resultados podem ser consultados em [Bastos, 2003]. Refira-se apenas que os algoritmos de afectação utilizados em [Bastos, 2003] são algoritmos de afectação simples, isto é, procuram o melhor emparelhamento global possível sem atenderem a restrições como por exemplo a do respeito pela ordem dos pontos emparelhados.

A integração do algoritmo de programação dinâmica na plataforma computacional *CMIS* foi simples, considerou-se a matriz de afinidade, Z , previamente calculada pela referida plataforma computacional com base no par de contornos escolhidos, como a matriz de custos. Assim, o algoritmo de programação dinâmica devolve o emparelhamento global cuja soma dos custos associados a esse emparelhamento seja mínima de entre todos os emparelhamentos globais que respeitam a ordem relativa dos pontos emparelhados.

Note-se que este algoritmo de optimização das correspondências baseado em programação dinâmica foi desenvolvido para emparelhar contornos fechados definidos por conjuntos de pontos ordenados, pelo que antes de determinar a matriz de afinidade Z é necessário garantir a ordenação dos pontos dos contornos a emparelhar, para tal pode ser usado, por exemplo, o algoritmo proposto no terceiro capítulo.

Após a fase inicial para verificar da correcta integração do algoritmo de programação dinâmica com restrição de ordem (APDCRO), foram realizados diversos ensaios de emparelhamento de contornos, partindo das matrizes de afinidade calculadas pela metodologia de Shapiro já referida e por outras metodologias existentes na referida plataforma. Para comparar o desempenho do APDCRO com os algoritmos de afectação sem restrição de ordem (AASRO) referidos, foi utilizada a matriz de afinidade obtida pelo método de Shapiro. Os resultados são apresentados na subsecção seguinte.

5.5.2 Comparação de resultados

5.5.2.1 Definição dos parâmetros utilizados nos ensaios

Para permitir comparar os algoritmos de optimização baseados no método *Húngaro*, *Simplex* para problemas de fluxo e *LAPm* com o novo algoritmo de optimização baseado em programação dinâmica, é fundamental que o processo de determinação da matriz de custos associada aos pontos que definem ambos os contornos seja exactamente o mesmo. Assim, em todos os ensaios realizados foi aceite a configuração definida por defeito na plataforma computacional usada para determinação da matriz de afinidade pelo método de Shapiro implementado na mesma, Figura 5.4.

Em relação às definições do algoritmo *Simplex* para problemas de fluxo integrado na plataforma, foi também utilizada a configuração definida por defeito, pois é de um modo geral a mais rápida, Figura 5.5. Em relação à contagem de tempo dispendido por cada um dos algoritmos de optimização, foi utilizada uma função já disponível para esse efeito na

plataforma usada, Figura 5.6.

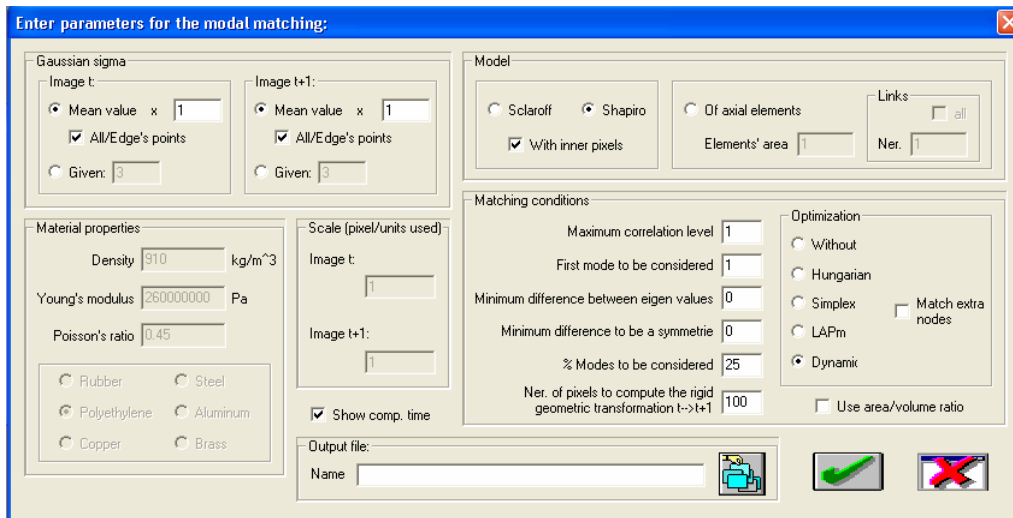


Figura 5.4: Parâmetros definidos na plataforma computacional *CMIS* para determinar a matriz de afinidade entre os pontos que definem dois contornos a emparelhar.

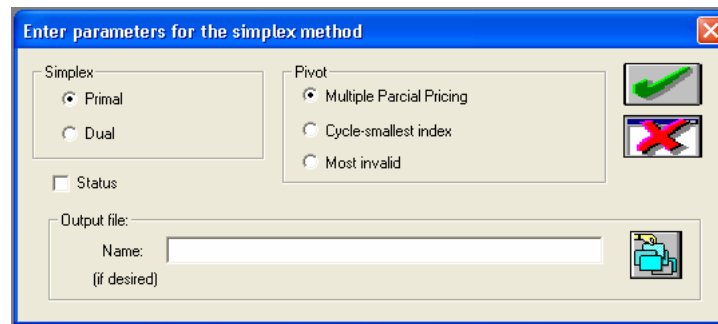


Figura 5.5: Configuração definida por defeito na plataforma computacional *CMIS* para o algoritmo de optimização baseado no algoritmo *Simplex* para problemas de fluxo.

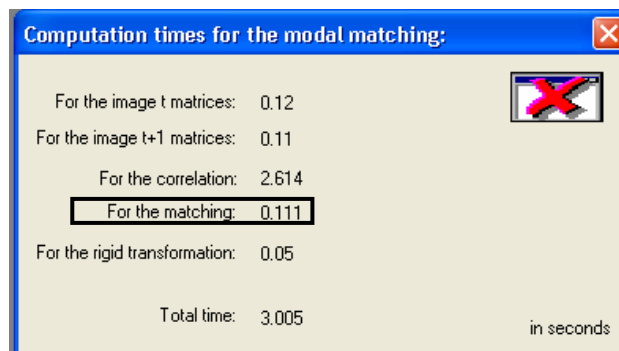


Figura 5.6: Exemplo da janela de visualização do tempo de execução para cada passo de todo o processo emparelhamento, na plataforma computacional *CMIS*. (No interior do rectângulo representado a preto está a parte que tem mais interesse para este estudo.)

5.6.2.2 Resultados

Em relação aos emparelhamentos obtidos utilizando modelação geométrica proposta por Shapiro e as configurações já definidas, os dois tipos de algoritmos apresentaram exactamente os mesmos emparelhamentos em grande parte dos contornos testados. A diferença surgiu quando os AASRO apresentaram emparelhamentos cruzados, o que obviamente não aconteceu no APDCRO. Para ilustrar as diferenças dos emparelhamentos dos dois tipos de algoritmos, em diversas situações, observem-se as Figuras 5.7 a 5.25. Nas Figuras 5.14, 5.18 e 5.20 são utilizados contornos de imagens extraídas de ensaios de pedobarografia dinâmica. Pormenores sobre as mesmas podem ser consultados em [Tavares, 2000b].

Só em casos muito particulares há mais do que uma correspondência global de custo mínimo. Assim, os emparelhamentos apresentados pelos três algoritmos de afectação clássicos foram sempre iguais. Deste modo, optou-se por apresentar apenas as imagens dos emparelhamentos obtidos com o algoritmo *Simplex* para problemas de fluxo para representar os três algoritmos de afectação sem restrição de ordem considerados.



Figura 5.7: Emparelhamento dos contornos da Figura 5.2 usando programação dinâmica.

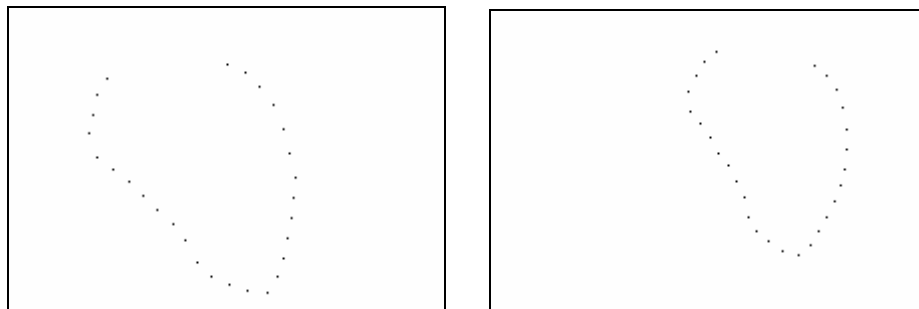


Figura 5.8: Dois contornos de um coração: à esquerda “heart1” e à direita “heart1a”, definidos por 28 pontos cada um.



Figura 5.9: Emparelhamento dos contornos da Figura 5.8: à esquerda usando os algoritmos de afectação clássicos e à direita usando o algoritmo de programação dinâmica.

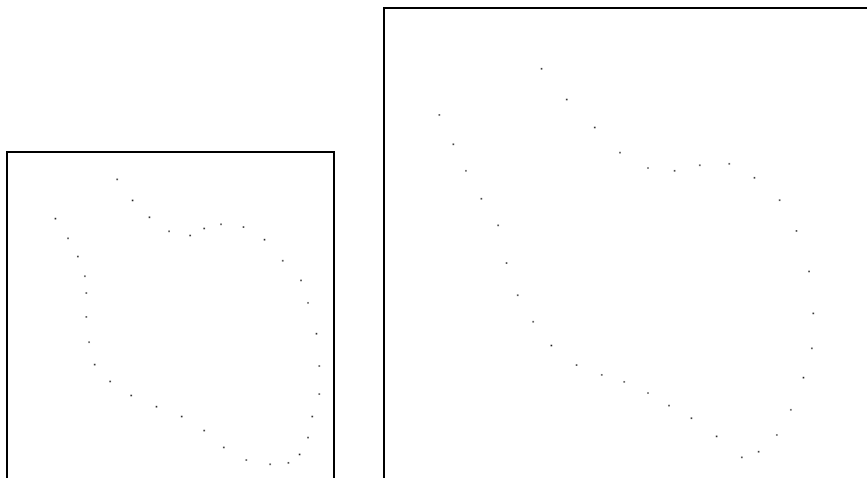


Figura 5.10: Dois contornos de um coração: à esquerda “heartA1” e à direita “heartA2”, definidos por 36 pontos cada um.

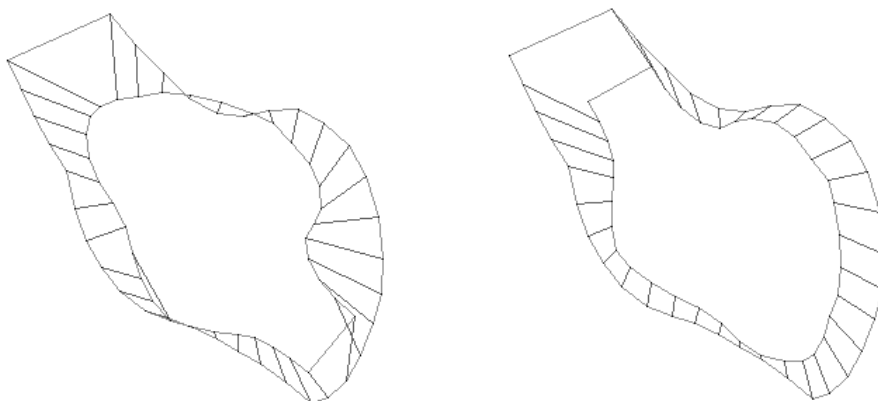


Figura 5.11: Emparelhamento dos contornos da Figura 5.10: à esquerda usando os algoritmos de afectação clássicos e à direita usando o algoritmo de programação dinâmica.

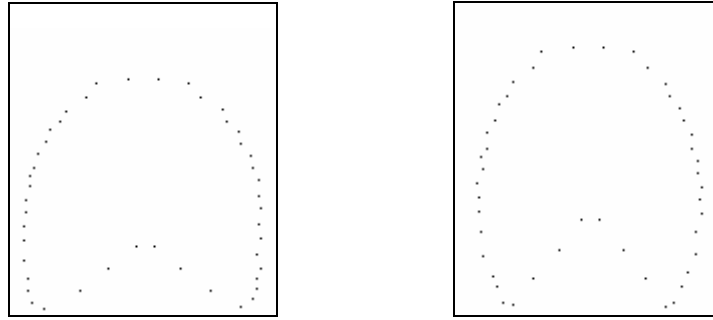


Figura 5.12: Dois contornos de uma caixa torácica: à esquerda “*rib1*” e à direita “*rib2*”, definidos por 46 pontos cada um.

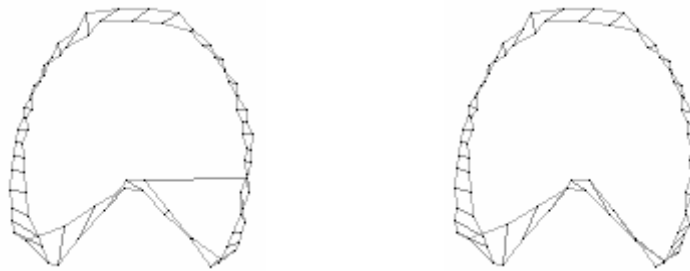


Figura 5.13: Emparelhamento dos contornos da Figura 5.12: à esquerda usando os algoritmos de afectação clássicos e à direita usando o algoritmo de programação dinâmica.

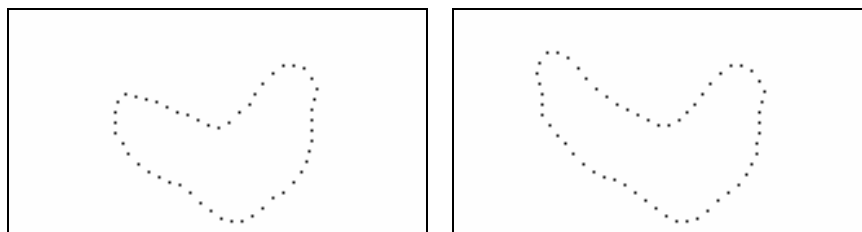


Figura 5.14: Dois contornos obtidos de imagens de pedobarografia dinâmica: à esquerda “*footBar1*” e à direita “*footBar6*”, definidos por 51 e 58 pontos, respectivamente.



Figura 5.15: Emparelhamento dos contornos da Figura 5.14: à esquerda usando os algoritmos de afectação clássicos e à direita usando o algoritmo de programação dinâmica.

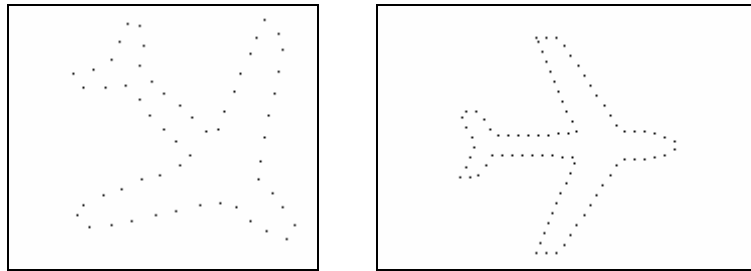


Figura 5.16: Dois contornos de um avião: à esquerda “*airplane2*” e à direita “*airplane12*”, definidos por 57 e 86 pontos, respectivamente.

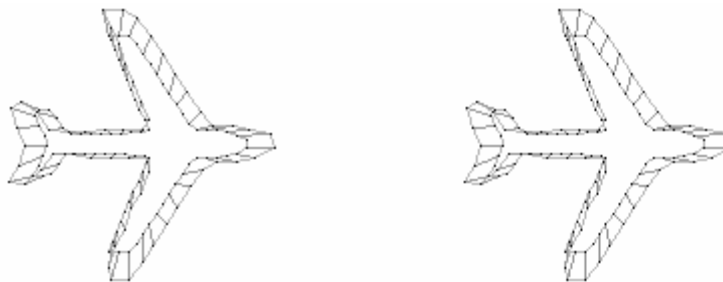


Figura 5.17: Emparelhamento dos contornos da Figura 5.16: à esquerda usando os algoritmos de afectação clássicos e à direita usando o algoritmo de programação dinâmica.



Figura 5.18: Dois contornos obtidos de imagens de pedobarografia dinâmica: à esquerda “*foot2*” e à direita “*foot13*”, definidos por 67 e 233 pontos, respectivamente.



Figura 5.19: Emparelhamento dos contornos da Figura 5.18: à esquerda usando os algoritmos de afectação clássicos e à direita usando o algoritmo de programação dinâmica. (A seta representada indica um emparelhamento cruzado.)

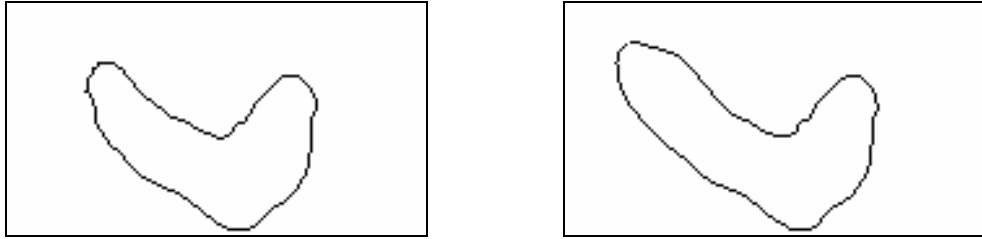


Figura 5.20: Dois contornos obtidos de imagens de pedobarografia dinâmica: à esquerda, “*foot13*” e à direita “*foot14*”, definidos por 233 e 253 pontos, respectivamente.



Figura 5.21: Emparelhamento dos contornos da Figura 5.20: à esquerda usando os algoritmos de afectação clássicos e à direita usando o algoritmo de programação dinâmica.

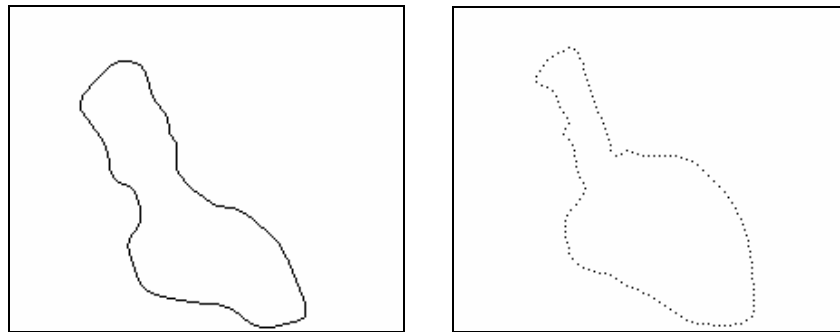


Figura 5.22: Dois contornos de um coração e artéria aorta: à esquerda “*heartB3*” e à direita “*heartB2*”, definidos por 389 e 139 pontos, respectivamente.

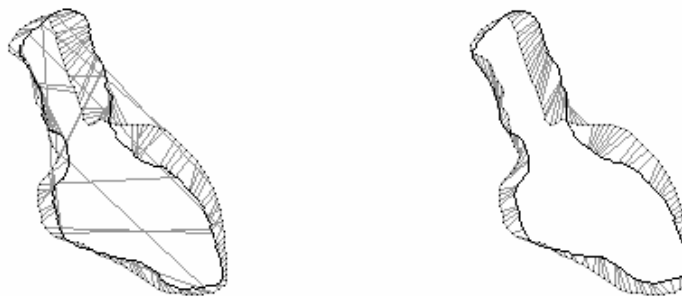


Figura 5.23: Emparelhamento dos contornos da Figura 5.22: à esquerda usando os algoritmos de afectação clássicos e à direita usando o algoritmo de programação dinâmica.



Figura 5.24: Dois contornos de um coração e artéria aorta: à esquerda “*heartB3*” e à direita “*heartB4*”, definidos por 389 e 417 pontos, respectivamente.

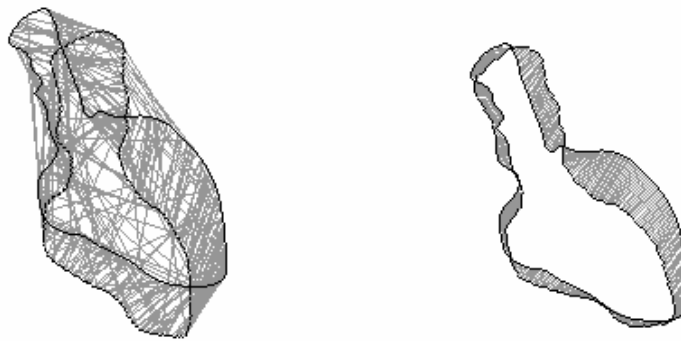


Figura 5.25: Emparelhamento dos contornos da Figura 5.24: à esquerda usando os algoritmos de afectação clássicos e à direita usando o algoritmo de programação dinâmica.

Em todas as imagens de emparelhamentos resultantes foi aplicada a translação e rotação determinadas pela plataforma computacional para alinhar os contornos. Essa metodologia de alinhamento dos contornos não é a definida no quarto capítulo, mas sim uma metodologia previamente integrada na plataforma *CMIS*, [Tavares, 2000a]. Para determinar a rotação, a plataforma parte dos emparelhamentos obtidos, assim, maus emparelhamentos como os verificados nas Figuras 5.11 e 5.25, relativas ao emparelhamento com os AASRO, deram origem a um valor incorrecto para o ângulo de rotação, sendo por este motivo que os contornos das figuras referidas estão não estão correctamente alinhados.

Na Tabela 5.2 são apresentados os tempos de execução para determinar os emparelhamentos dos contornos apresentados nas figuras anteriores e os respectivos custos. Note-se que os custos dependem dos elementos da matriz de afinidade e esta por sua vez depende, além dos contornos, dos valores dos parâmetros considerados para o seu cálculo. É ainda importante referir que os valores assinalados para os tempos de execução

são valores aproximados, pois foram registadas ligeiras variações. Os ensaios foram realizados num computador equipado com um processador *Intel Pentium III* a 1.0 GHz, com 256MB de RAM.

Tabela 5.2: Comparação da velocidade de execução do algoritmo de programação dinâmica relativamente aos algoritmos de afectação: método Húngaro, Simplex para problemas de fluxo e LAPm.

Número de pontos e “nome” do contorno		Custos globais		Tempos de execução [s]			
<i>Contorno 1</i>	<i>Contorno 2</i>	Húngaro/ Simplex/ LAPm	Dina- mico	Hún- garo	Sim- plex	LAPm	Dinâ- mico
28, “heart1”	28, “heart1a”	0,0040	0,0040	4,23	0,02	0,01	0
36, “heartA1”	36, “heartA2”	2,6424	11,203	>60	0,04	3,325	0
46, “rib1”	46, “rib2”	3,6397	4,0664	>60	0,06	2,774	0
51, “foot1”	58, “foot6”	1,3802	1,3802	>60	0,09	0,751	0,01
86, “airplane2”	57, “airplane12”	1,7452	1,7452	>60	0,19	1,772	0,01
81, “heart5”	84, “heart6”	5,7903	6,7061	>60	0,21	2,34	0
233, “foot13”	67, “foot2”	6,0508	6,1126	>60	1,372	23,554	0,24
233, “foot13”	253, “foot14”	50,264	57,741	>60	2,013	>60	0,151
389, “heartB3”	139, “heartB2”	24,215	25,858	>60	4,387	>60	2,164
389, “heartB3”	417, “heartB4”	123,65	150,89	>60	6,961	>60	0,621

5.6.2.3 Análise de resultados

Uma conclusão fundamental que se pode extrair dos resultados obtidos é a adequação do APDCRO para optimização dos emparelhamentos de contornos de objectos representados em imagens. O conjunto – metodologia de determinação dos emparelhamentos entre dois contornos baseada na modelação geométrica de Shapiro e algoritmo de programação dinâmica – criou uma metodologia global mais robusta do que as já existentes na plataforma computacional *CMIS*. Basta observar que, para a configuração base representada na Figura 5.4, esta nova metodologia global alcançou sempre óptimos emparelhamentos nos exemplos apresentados; enquanto as metodologias de optimização globais já implementadas na plataforma determinaram maus emparelhamentos em alguns pares de contornos e um emparelhamento sem sentido num par de contornos, Figura 5.11.

Embora aqui não seja apresentado, por ser irrelevante para a validação do algoritmo de programação dinâmica desenvolvido e sua comparação com os algoritmos de afectação clássicos considerados, houve um conjunto muito reduzido de situações em que não foi obtido um emparelhamento coerente por qualquer um dos algoritmos de optimização

utilizados. Obviamente, esta situação não se deve aos algoritmos de optimização considerados mas à sim metodologia de determinação da matriz de custo e respectivos valores dos parâmetros utilizados.

Embora não tenha sido convenientemente testado, é espectável que as outras metodologias de determinação de uma matriz de afinidade (modelação baseada em princípios físicos, por exemplo), também já existente na referida plataforma computacional, em conjunto com o algoritmo de optimização baseado em programação dinâmica originem uma metodologia global muito mais robusta do que as metodologias globais já implementadas na plataforma.

Passemos agora a especificar em mais detalhe as diferenças verificadas entre os dois tipos de algoritmos de optimização das correspondências: AASRO e APDCRO. Em relação à qualidade dos emparelhamentos obtidos, os três algoritmos de afectação clássicos apresentaram sempre o mesmo emparelhamento de custo mínimo, o que era de esperar, pois todos estão sujeitos às mesmas restrições. Quanto à comparação da qualidade dos emparelhamentos alcançados com os AASRO e com o APDCRO, os resultados permitem concluir o seguinte:

- Sempre que os AASRO alcançavam um bom emparelhamento sem correspondências cruzadas, o APDCRO alcançava o mesmo emparelhamento; portanto, o custo global dos emparelhamentos era exactamente o mesmo.
- Quando os AASRO alcançavam um emparelhamento com cruzamentos, o APDCRO alcançava um emparelhamento idêntico mas sem cruzamentos. Obviamente o custo associado teria que ser superior, pois a restrição da ordem obrigou a que emparelhamentos de menor custo fossem substituídos por emparelhamentos de maior custo mas mais coerentes.
- Em algumas situações em que os AASRO alcançaram um emparelhamento com pouco sentido ou sem sentido, como nas Figuras 5.25 e 5.11, o APDCRO alcançou um emparelhamento coerente. Obviamente, nesta situação os custos apresentados por ambos os tipos de algoritmos são muito diferentes. Refira-se que estes emparelhamentos incorrectos, em particular, poderiam desaparecer com a alteração de alguns parâmetros definidos por defeito na plataforma para a determinação da matriz de afinidade. No entanto, outros emparelhamentos inadequados poderiam surgir no emparelhamento de outros contornos.

Em relação ao tempo de execução, o APDCRO foi sempre bastante mais rápido do que

os AASRO, independentemente dos contornos serem definidos por igual ou diferente número de pontos, ou por muitos ou poucos pontos. Como se observa na Tabela 5.2, até houve muitas situações em que a plataforma indicou tempos de execução de 0 (zero) segundos para o APDCRO, o que significa tempos realmente muito baixos.

Pode-se constatar que os tempos de execução do APDCRO variaram de acordo com previsto na subsecção 5.4.3, ou seja, o tempo de execução aumentou com o aumento do número de pontos que definem os contornos, mas diminuiu à medida que a diferença entre os números de pontos que definem os dois contornos diminuiu.

5.6 Emparelhamentos do tipo *um-para-vários*

5.6.1 Definição

As formulações apresentadas nas secções 5.3 e 5.4 permitem obter para o emparelhamento óptimo do tipo *um-para-um* que respeita a ordem relativa dos pontos dos contornos a emparelhar. Acontece que, em grande parte das situações, os contornos a emparelhar são definidos por diferente número de pontos. Assim, foram desenvolvidos dois algoritmos que após o emparelhamento do tipo *um-para-um* fazem o emparelhamento dos pontos excedentários, ou seja, aqueles que não foram correspondidos aquando do emparelhamento inicial do tipo *um-para-um*.

Neste trabalho, quando um contorno está definido por maior número de pontos do que o outro contorno que com ele vai emparelhar, iremos chamar *emparelhamento do tipo um-para-vários* a um qualquer emparelhamento onde cada ponto do contorno definido por maior número de pontos corresponde a um só ponto do contorno definido por menor número de pontos, e cada ponto deste último contorno corresponde a um só ponto do contorno definido por maior número de pontos. Assim, quer sejam os pontos do contorno de partida que correspondem a vários pontos do contorno de chegada quer seja o contrário, chamaremos apenas emparelhamento do tipo *um-para-vários*.

Para fazer o emparelhamento dos pontos não correspondidos, podem ser utilizados diversos critérios. Neste trabalho, num dos algoritmos, cada ponto não correspondido na correspondência do tipo *um-para-um* passará a ficar correspondido com o ponto do outro contorno que lhe está mais próximo em termos de matriz de custos. No outro algoritmo, passará a ficar correspondido com o ponto do outro contorno que lhe está mais próximo em termos de distância euclidiana, após aplicação da transformação rígida que melhor alinha

os contornos. Obviamente, em ambos os algoritmos existirá a restrição de respeito pela ordem dos pontos.

5.6.2 Baseado numa matriz de custos

Este algoritmo desenvolvido para emparelhamento do tipo *um-para-vários* é baseado na minimização da soma dos custos associados a cada emparelhamento, naturalmente com a imposição de respeito pela ordem relativa dos pontos. Passemos à definição do algoritmo. Suponhamos que temos dois contornos, o *contorno 1* e o *contorno 2*, definidos por n e m pontos respectivamente, com $n < m$. Suponhamos ainda que já foi calculada a matriz de custos C e já foi determinado o emparelhamento do tipo *um-para-um* que minimiza a soma dos custos globais respeitando a ordem dos pontos. De uma forma simplificada, o algoritmo desenvolvido consiste nos seguintes passos:

Algoritmo:

1. De $i = 1$ até n :
 - i. Determinar os pontos do *contorno 2* que estão entre os pontos que emparelham com os *pontos* i e $i + 1$ do *contorno 1* (o *ponto* $n + 1$ é o *ponto* 1).
 - ii. Se o número de pontos determinados anteriormente é maior do que zero:
Determinar uma divisão em dois grupos dos pontos não emparelhados determinados no passo anterior (possivelmente um desses grupos pode ser vazio). Um dos grupo de pontos vai emparelhar com o *ponto* i do *contorno 1* e outro grupo vai emparelhar com o *ponto* $i + 1$ do *contorno 1*. A divisão dos pontos respeita a ordem dos mesmos e é feita de modo a minimizar a soma dos custos dos novos emparelhamentos relativos aos pontos excedentários.
2. Calcular os novos custos globais e representar os emparelhamentos.

A título de exemplo, suponhamos que temos dois pontos não emparelhados entre os *pontos* j e $j + 3$ do *contorno 2* (os *pontos* $j + 1$ e $j + 2$). Suponhamos ainda que o *ponto* j corresponde ao *ponto* i e o *ponto* $j + 3$ corresponde ao *ponto* $i + 1$, obviamente estes pontos são do *contorno 1*. Neste caso, apenas três situações de novos emparelhamentos podem acontecer respeitando a ordem dos pontos:

- i. Os pontos $j+1$ e $j+2$ irão ambos corresponder ao ponto i ;
- ii. O ponto $j+1$ irá corresponder ao ponto i e o ponto $j+2$ irá corresponder ao ponto $i+1$;
- iii. Os pontos $j+1$ e $j+2$ irão ambos corresponder ao ponto $i+1$.

Das três opções de emparelhamento anteriores, o algoritmo irá escolher a que representar no total menor custo global associado a esses emparelhamentos, ou seja, aquela opção cuja soma dos custos dos dois novos emparelhamentos seja mínima.

Como já referido, a metodologia de emparelhamento do tipo *um-para-vários* apresentada nesta subsecção parte de um emparelhamento inicial do tipo *um-para-um*. Assim, optou-se por apresentar resultados do emparelhamento do tipo *um-para-vários* apenas no capítulo seguinte, aquando da apresentação de metodologias de obtenção de uma matriz de custo de emparelhamento e consequente emparelhamento global do tipo *um-para-um*.

5.6.3 Baseado na minimização da distância euclidiana entre pontos

O algoritmo implementado para realizar os emparelhamentos dos pontos excedentários com base na minimização da soma das distância entre os pontos a emparelhar é idêntico ao apresentado na subsecção anterior. Há apenas duas diferenças: a primeira consiste na necessidade de determinar e aplicar a transformação rígida que melhor alinha os dois contornos; a segunda tem a ver com os custos, que neste caso são as distâncias entre os pontos dos dois contornos.

Para ajudar a perceber este algoritmo vamos recorrer a um exemplo. Suponhamos que já temos os emparelhamentos do tipo *um-para-um* definidos na Figura 5.26, e a transformação rígida que melhor alinha os contornos já foi aplicada. Observa-se que, inicialmente, o ponto 22 do contorno 2 não está correspondido. Para não provocar cruzamentos, esse ponto apenas pode corresponder aos pontos 6 ou 7 do contorno 1. Como está mais próximo do ponto 7 do que do ponto 6, será então correspondido ao ponto 7, mantendo-se todas as outras correspondências determinadas anteriormente pela metodologia do tipo *um-para-um*.

Pelas razões apresentadas na subsecção anterior, os resultados obtidos com a metodologia proposta nesta subsecção ficará para o capítulo seguinte.

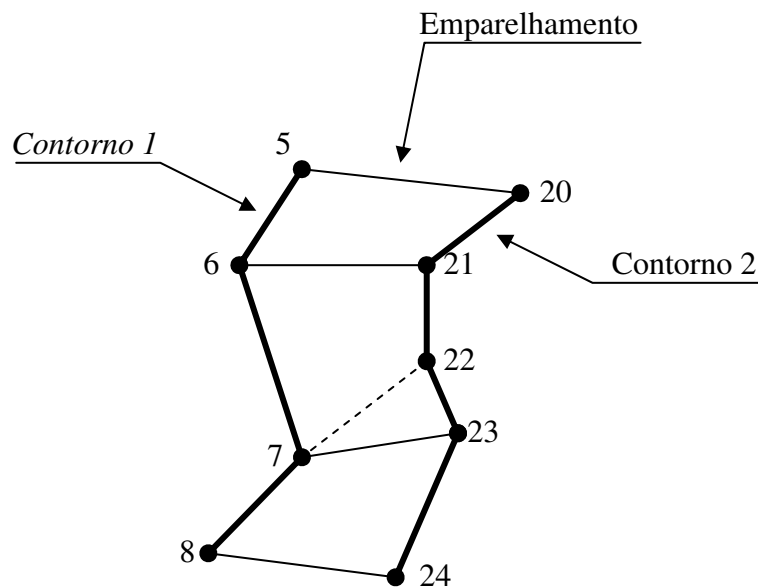


Figura 5.26: Exemplo de emparelhamento do tipo *um-para-vários*. O traço grosso representa as secções dos contornos após aplicação da transformação rígida. O traço mais fino contínuo representa os emparelhamentos do tipo *um-para-um* iniciais. O traço interrompido representada o novo emparelhamento.

5.7 Sumário e conclusões

Neste capítulo estudou-se o problema de emparelhar os pontos que definem dois contornos ordenados no sentido de minimizar o custo global, partindo-se de uma matriz de custos previamente calculada, respeitando a ordem dos pontos emparelhados para evitar emparelhamentos cruzados.

Foram apresentadas duas formulações. A primeira baseada num modelo de programação inteira que pode ser implementada com recurso ao algoritmo *simplex* adaptado para modelos de programação inteira. Atendendo ao custo computacional que esta solução envolve, optou-se pela sua não implementação. A segunda formulação utiliza a técnica de optimização de programação dinâmica. O custo computacional teórico revelou que esta alcançaria bons resultados em termos de velocidade.

Para confirmar as expectativas, a formulação do problema de optimização como um problema de programação dinâmica foi desenvolvida na linguagem de programação C++ e implementada numa plataforma computacional de desenvolvimento e ensaio para validação e comparação com outras metodologias de optimização da correspondência

global. A metodologia usada para obter as matrizes de afinidade utilizadas nos ensaios foi baseada em modelação geométrica proposta por Shapiro, razão pela qual foi efectuada uma apresentação desta metodologia.

Seguidamente, foi efectuada uma comparação entre os dois tipos algoritmos de optimização do emparelhamento global entre os pontos de dois contornos: o algoritmo de programação dinâmica com restrição de ordem, já referido, e os algoritmos de afectação clássicos sem restrição de ordem: método *Húngaro*, *Simplex* para problemas de fluxo e *LAPm*. Os aspectos a comparar foram a qualidade dos emparelhamentos, o custo global dos emparelhamentos e os tempos de execução dos dois tipos de algoritmos, sendo para tal utilizados contornos de diversos objectos definidos por igual ou diferente número de pontos.

Os resultados obtidos comprovam que, para o emparelhamento de contornos ordenados, o algoritmo de programação dinâmica desenvolvido apresenta sempre resultados tão bons ou melhores do que os algoritmos de afectação clássicos utilizados na comparação, em termos de qualidade de emparelhamento. Em relação à velocidade de execução, o algoritmo de programação dinâmica mostrou excelente desempenho, determinando sempre o emparelhamento global num tempo muito inferior ao dos seus competidores.

Outra conclusão importante que se pode retirar é o grande aumento de robustez do método de emparelhamento baseado modelação geométrica proposto por Shapiro, quando aplicado a contornos ordenados em conjunto com o algoritmo de programação dinâmica. Obviamente, esta robustez advém da imposição da restrição de ordem dos pontos emparelhados.

Finalmente, concluiu-se este capítulo com o desenvolvimento de duas metodologias para realizar emparelhamentos do tipo *um-para-vários*, as quais só podem ser aplicadas depois de se determinar um emparelhamento global do tipo *um-para-um*. A primeira metodologia faz os emparelhamentos em função dos custos de emparelhamento previamente calculados e a segunda metodologia faz os emparelhamentos em função da distância euclidiana entre pontos candidatos a emparelhamento.

CAPÍTULO VI:
EMPARELHAMENTOS BASEADOS EM INFORMAÇÃO DE
CURVATURA E DISTÂNCIA AO CENTRÓIDE

6.1 Introdução

Neste capítulo serão apresentadas e testadas metodologias para determinar uma matriz de custos de emparelhamento entre pontos que definem dois contornos ordenados de um objecto, ou de dois objectos, representados em imagens. Serão também apresentados resultados de emparelhamento do tipo *um-para-vários* usando as metodologias apresentadas na secção 5.6.

Assim, na secção 6.2 será apresentada uma metodologia de determinação da matriz de custos baseada em informação de curvatura. Ainda nesta secção, serão realizados vários ensaios experimentais usando a metodologia proposta. A secção seguinte será para apresentar uma metodologia baseada na comparação da distância dos pontos que definem os contornos ao respectivo centróide e apresentar os resultados dos ensaios realizados utilizando a mesma. Na secção 6.4 será apresentada outra metodologia de determinação de uma matriz de custos. Esta utiliza informação de curvatura e informação da distância dos pontos ao respectivo centróide. Todas estas três metodologias podem ser aplicadas a contornos ordenados definidos por igual ou diferente número de pontos. Para determinar o emparelhamento global que minimiza os custos, é utilizado o algoritmo de programação dinâmica apresentado na secção 5.4.

Nas secções 6.2, 6.3 e 6.4, além da análise dos emparelhamentos obtidos, serão também analisados resultados obtidos pela metodologia de determinação da transformação rígida existente entre dois contornos a emparelhar, apresentada no quarto capítulo.

Na secção 6.5 será apresentada uma metodologia para proceder a reajustes locais nos emparelhamentos, após um emparelhamento inicial, a qual pode ser aplicada no emparelhamento de contornos definidos por diferente número de pontos.

A secção 6.6 será para apresentar resultados das duas metodologias de emparelhamento do tipo *um-para-vários* apresentadas na secção 5.6, partindo de um emparelhamento inicial do tipo *um-para-um*. A primeira metodologia de tipo *um-para-vários* a apresentar será baseada na informação dos custos de emparelhamento obtidos aquando da determinação da matriz de custos. A segunda metodologia efectuará os emparelhamentos do tipo *um-para-vários* no sentido de minimizar a distância euclidiana entre os pontos a corresponder. Obviamente, nesta última, terá de ser tomada em consideração a transformação rígida que melhor alinha os dois contornos a emparelhar. Estas duas metodologias são utilizadas para corresponder os pontos excedentários, aquando do emparelhamento do tipo *um-para-um* de

contornos definidos por diferente número de pontos.

6.2 Emparelhamento baseado em informação de curvatura

6.2.1 Princípio da metodologia

As transformações rígidas ou, usando a designação matemática, transformações de semelhança, podem provocar alterações num objecto quer a nível da sua localização, dimensão ou rotação, mas não provocam alterações na forma do mesmo. Um invariante entre dois objectos quando um é obtido a partir do outro por uma transformação rígida é a curvatura. No caso dos polígonos, esse invariante é a amplitude dos ângulos do polígono.

Sendo os dois objectos a emparelhar definidos por um conjunto de pontos ordenados que definem o polígono associado ao seu contorno, vamos considerar como ângulo de curvatura, ou ângulo de viragem, o ângulo definido por cada conjunto de três pontos consecutivos.

Para calcular o ângulo associado a cada ponto, é também importante considerar o sentido em que o contorno está definido. A hipótese dos pontos serem ordenados é fundamental, pois caso contrário não seria possível definir o contorno e por consequência o seu polígono associado (ver capítulo III).

Consideremos então dois contornos arbitrários: o *contorno 1* e o *contorno 2*, cada um definido por uma sequência de pontos ordenados de dimensão n e m , respectivamente. Podemos supor, sem perda de generalidade, que $n \leq m$.

Seguidamente, para cada contorno, vamos construir uma sequência de ângulos, que passaremos a designar por ângulos de curvatura associados a cada ponto. Assim, ao *ponto* i do *contorno 1* vai corresponder a amplitude do ângulo α_i e ao *ponto* j do *contorno 2* vai corresponder a amplitude do ângulo θ_j . A amplitude dos ângulos de curvatura, considerando o sentido de rotação, é determinada do seguinte modo: Consideremos três pontos consecutivos do *contorno 1*, por exemplo: P_{i-1} , P_i e P_{i+1} . Consideremos ainda os vectores $\overrightarrow{P_i P_{i-1}}$ e $\overrightarrow{P_i P_{i+1}}$, ambos aplicados em P_i , e:

$$\alpha = \arccos \left(\frac{\overrightarrow{P_i P_{i-1}} \cdot \overrightarrow{P_i P_{i+1}}}{\left\| \overrightarrow{P_i P_{i-1}} \right\| \left\| \overrightarrow{P_i P_{i+1}} \right\|} \right).$$

Assim, α representa a amplitude do ângulo formado pelos dois vectores, não considerando o sentido do ângulo, isto é, não considerando um sentido de rotação. Deste modo, tem-se que $\alpha \in [0, \pi]$. Para ser possível determinar a amplitude do ângulo atendendo ao sentido de rotação, é necessário verificar se os vectores $\overrightarrow{P_i P_{i-1}}$ e $\overrightarrow{P_i P_{i+1}}$ formam uma base directa do espaço vectorial \mathbb{R}^2 . Assim, se o determinante da matriz:

$$\begin{bmatrix} \overrightarrow{P_i P_{i-1}} & \overrightarrow{P_i P_{i+1}} \end{bmatrix},$$

for positivo, a base formada pelos vectores é directa, sendo então $\alpha_i = \alpha \text{ rad}$. Se o determinante for negativo, a base não é directa, pelo que faremos $\alpha_i = (2\pi - \alpha) \text{ rad}$. Se o determinante for nulo, os vectores são colineares de sentidos opostos, pelo que $\alpha_i = \alpha = \pi \text{ rad}$. Observe-se a Figura 6.1, onde estão representadas duas secções de dois contornos. Pelo algoritmo implementado, considera-se o sentido de rotação e o sentido em que os contornos são definidos. Assim $\hat{A}BC = 270^\circ$ e $\hat{D}EF = 90^\circ$. Por outro lado, se não fossem considerados os sentidos, teríamos $\hat{A}BC = 90^\circ = \hat{D}EF$.

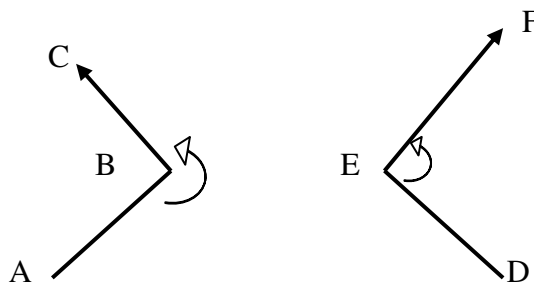


Figura 6.1: Exemplos de duas secções de dois contornos para ilustrar a forma de determinação do ângulo associado a cada ponto.

Deste modo, construíram-se duas sequências de ângulos orientados: $\alpha_1, \alpha_2, \dots, \alpha_n$ e $\theta_1, \theta_2, \dots, \theta_n, \dots, \theta_m$. Note-se que na metodologia implementada, se três pontos são colineares, o ângulo de curvatura no seu ponto central será $\pi \text{ rad}$ (180°). Refira-se, ainda, que as sequências de ângulos obtidas dependem apenas da forma dos contornos e do sentido pelo qual o contorno está definido, não da sua dimensão ou localização na imagem respectiva.

Agora, calcula-se a matriz C , matriz de custos angulares, sendo que c_{ij} representa a

diferença entre a amplitude do ângulo α_i do *contorno 1* e o ângulo θ_j do *contorno 2*. Esta matriz tem portanto dimensão $n \times m$:

$$C = \begin{bmatrix} |\alpha_1 - \theta_1| & |\alpha_1 - \theta_2| & \dots & |\alpha_1 - \theta_m| \\ |\alpha_2 - \theta_1| & |\alpha_2 - \theta_2| & \dots & |\alpha_2 - \theta_m| \\ \dots & \dots & \dots & \dots \\ |\alpha_n - \theta_1| & |\alpha_n - \theta_2| & \dots & |\alpha_n - \theta_m| \end{bmatrix}.$$

6.2.2 Resultados do emparelhamento usando programação dinâmica

Após a determinação da matriz de custos angulares, foi utilizado o algoritmo de programação dinâmica, definido na secção 5.4, para procurar o melhor emparelhamento global entre os pontos de ambos os contornos. Assim, é passada à rotina desenvolvida a matriz de custos e o número de pontos que define cada um dos contornos. Esta devolve o custo total do melhor emparelhamento obtido que respeita a ordem relativa dos pontos e o respectivo emparelhamento.

O algoritmo global de emparelhamento, usando informação de curvatura e optimização das correspondências com recurso à rotina de optimização baseada na técnica de programação dinâmica, foi desenvolvido em ambiente *Microsoft Visual C++*, [Rodrigues, 2003], sendo utilizadas funções da biblioteca de domínio público *VTK – Visualization Toolkit*, [Schroeder, 1999], para facilitar a visualização dos emparelhamentos obtidos.

Nas Figuras 6.2 a 6.11 podem ser observados alguns emparelhamentos obtidos entre os pares de contornos apresentados, sendo o *contorno 1* representado a azul e *contorno 2* representado a vermelho. Para cada figura, à esquerda e em cima, apresentam-se os contornos nas posições originais e à esquerda e em baixo, os contornos após aplicação da transformação rígida determinada pelo método descrito no quarto capítulo. À direita e em cima, os contornos nas posições originais e os respectivos emparelhamentos a verde e à direita e em baixo uma perspectiva 3D dos emparelhamentos e contornos após aplicação da transformação rígida determinada. A opção pela representação dos emparelhamentos numa perspectiva 3D é para permitir uma melhor visualização de alguns pormenores dos emparelhamentos obtidos. A representação dos dois contornos a emparelhar após aplicação da transformação rígida que melhor os alinha tem dois objectivos: facilitar a observação das deformações não rígidas de um contorno em relação ao outro e aferir da qualidade da transformação rígida calculada segundo a metodologia proposta no quarto capítulo.

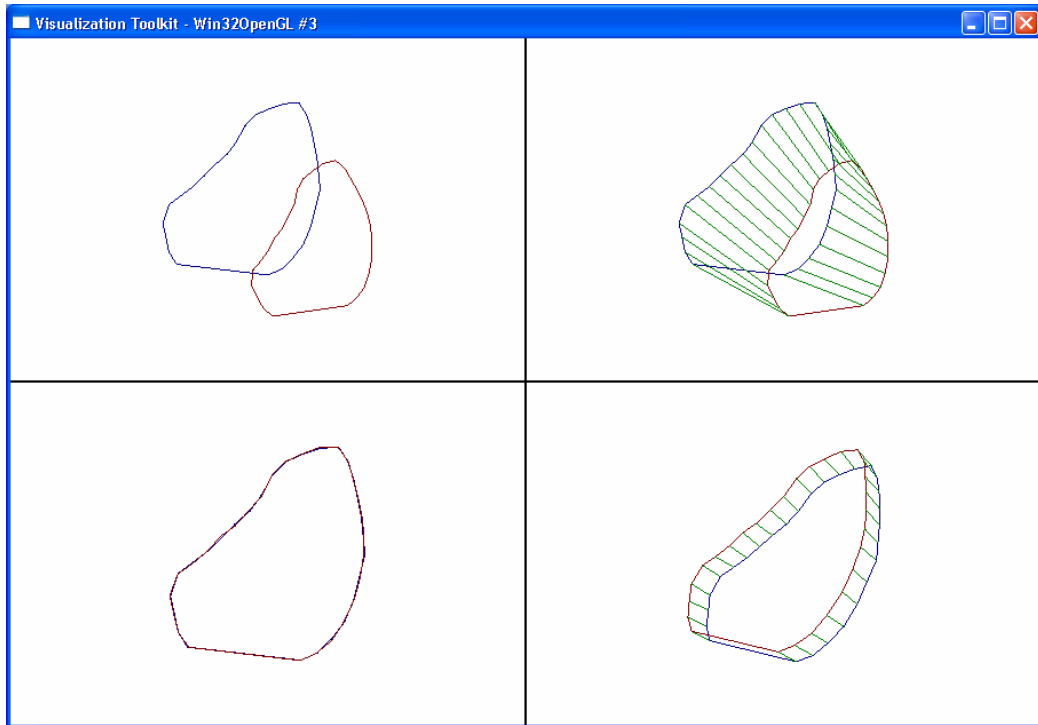


Figura 6.2: Emparelhamento baseado em informação de curvatura dos contornos “heart1” e “heart1a”, definidos por 28 pontos cada um.

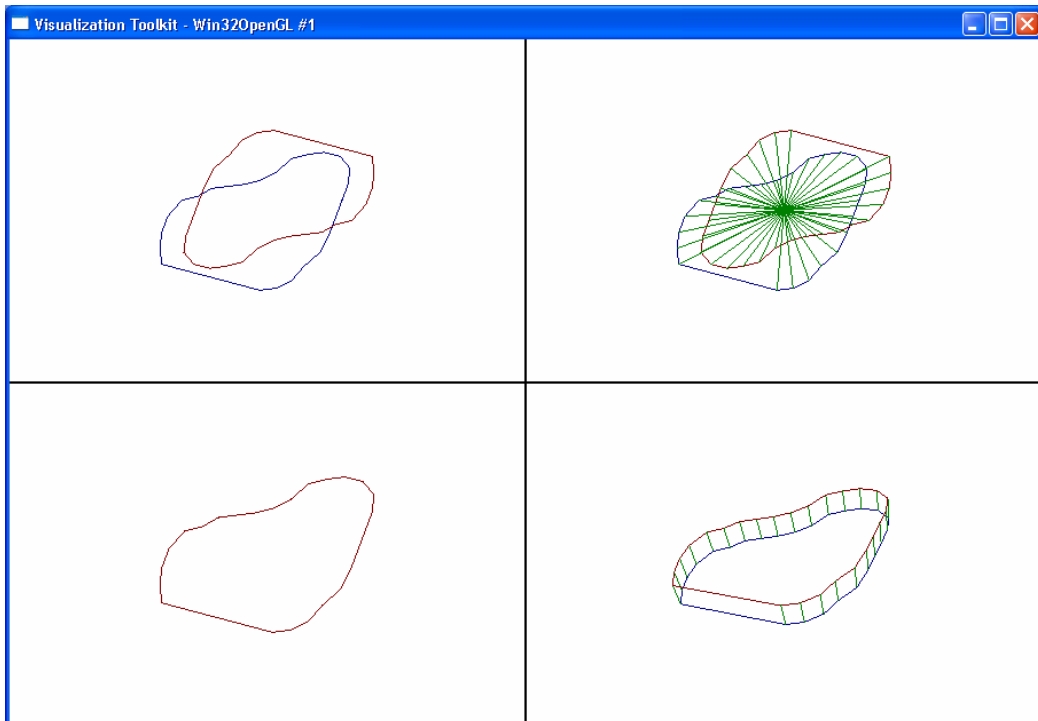


Figura 6.3: Emparelhamento baseado em informação de curvatura dos contornos “heart3” e “heart4”, definidos por 25 pontos cada um.

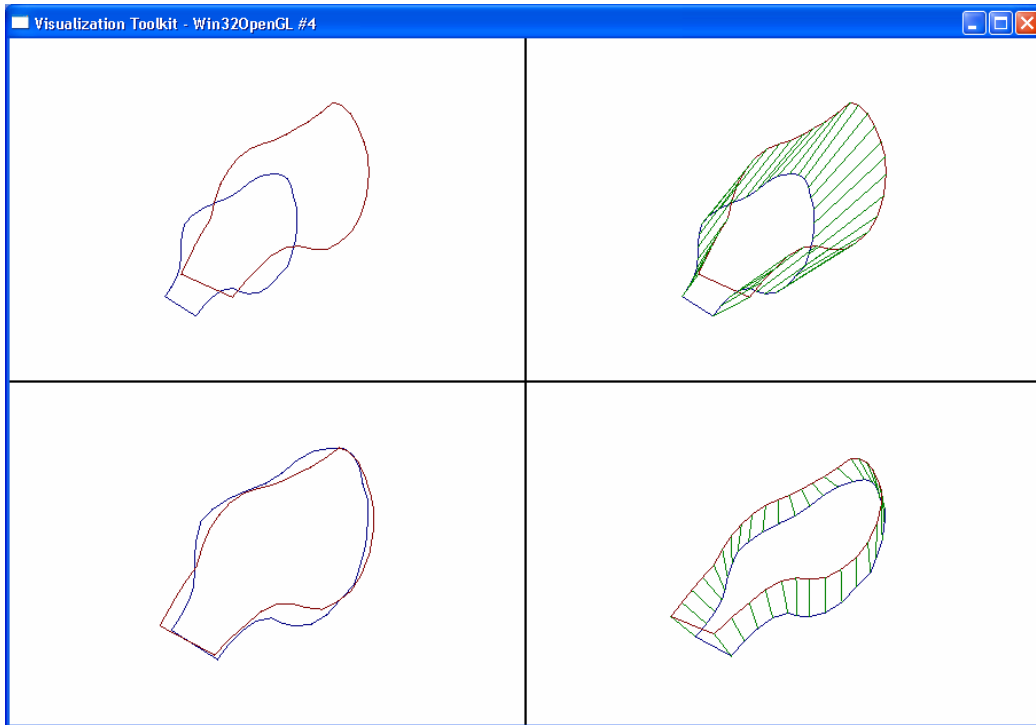


Figura 6.4: Emparelhamento baseado em informação de curvatura dos contornos “heartA1” e “heartA2”, definidos por 35 pontos cada um.

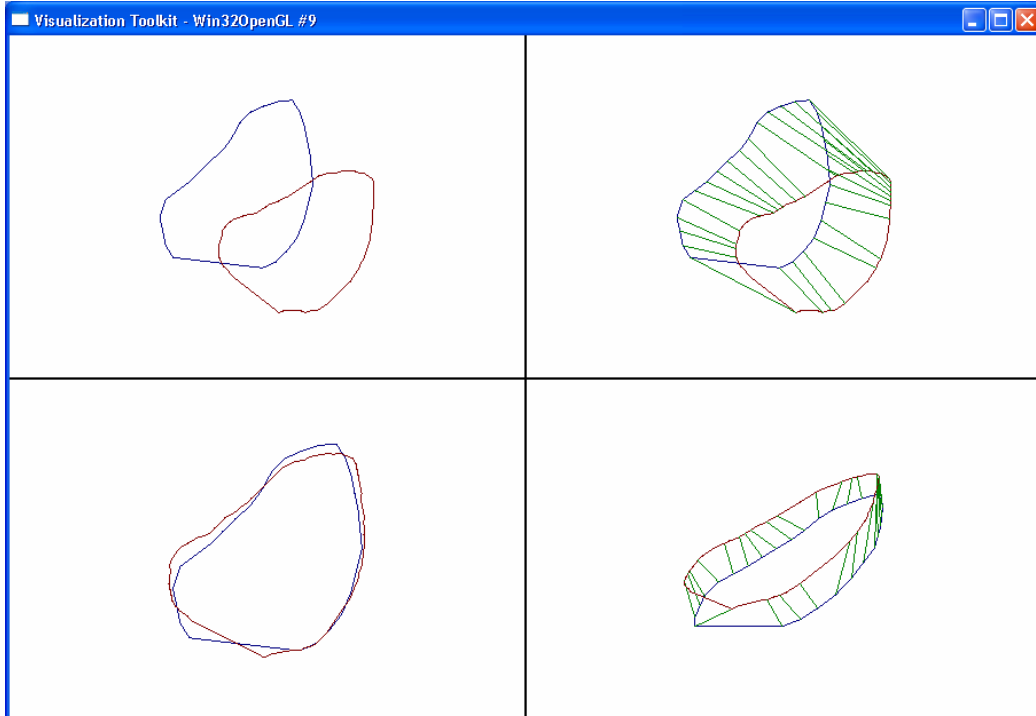


Figura 6.5: Emparelhamento baseado em informação de curvatura dos contornos “heart1” e “heart6”, definidos por 28 e 84 pontos, respectivamente.

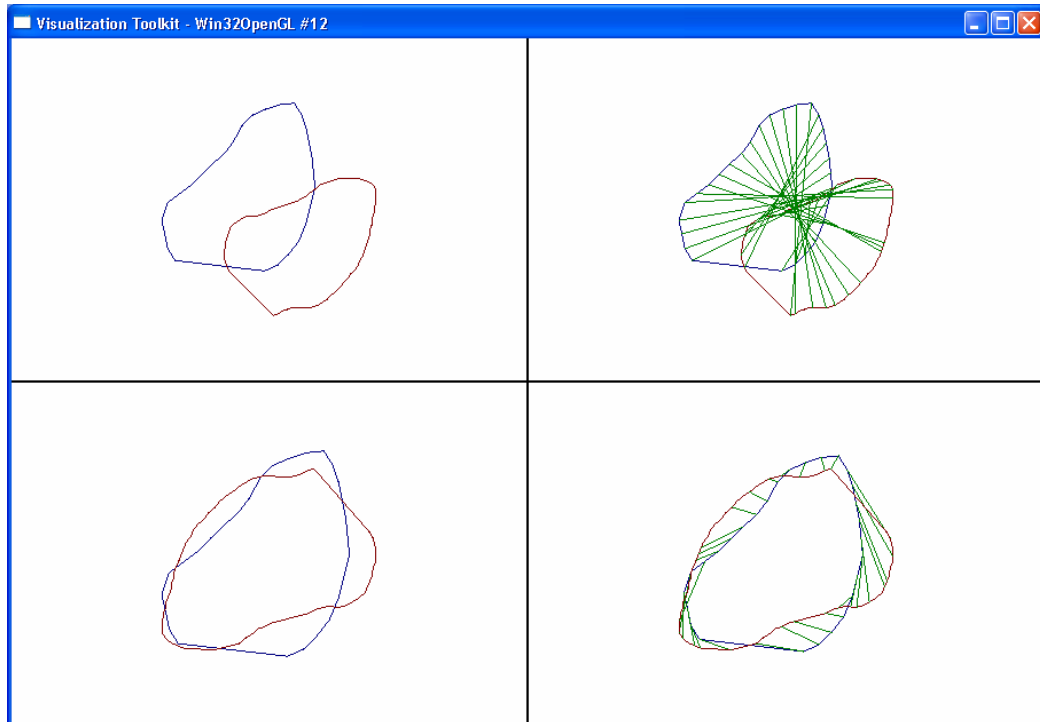


Figura 6.6: Emparelhamento baseado em informação de curvatura dos contornos “heart1” e “heart5”, definidos por 28 e 81 pontos, respectivamente.

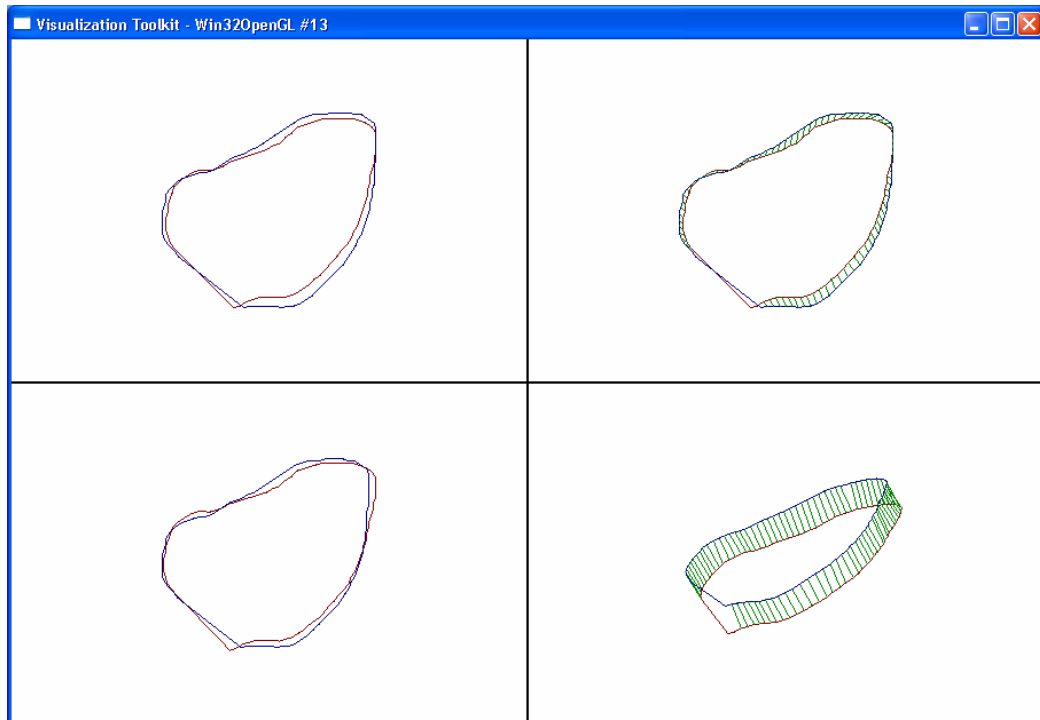


Figura 6.7: Emparelhamento baseado em informação de curvatura dos contornos “heart6” e “heart5”, definidos por 84 e 81 pontos, respectivamente.

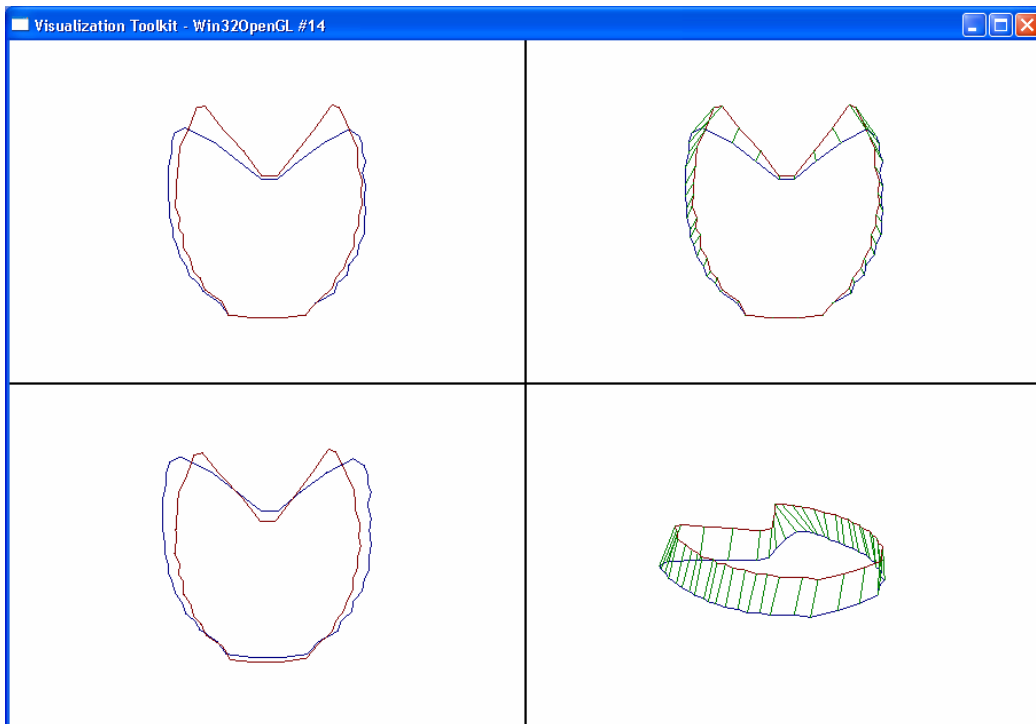


Figura 6.8: Emparelhamento baseado em informação de curvatura dos contornos “*rib1*” e “*rib2*”, definidos por 46 pontos cada um.

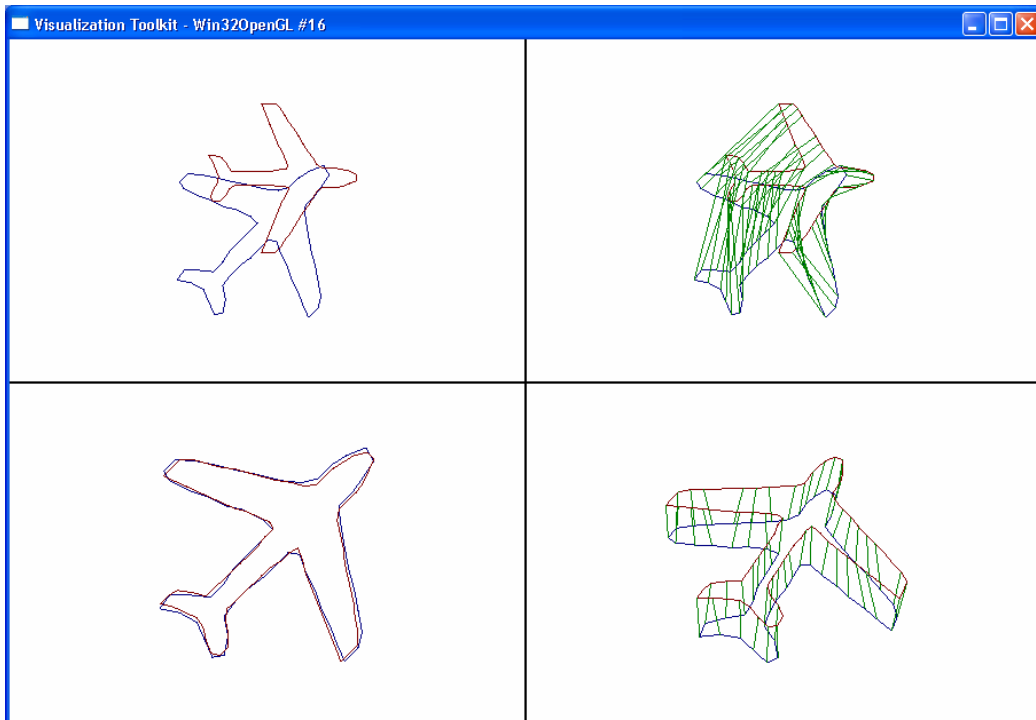


Figura 6.9: Emparelhamento baseado em informação de curvatura dos contornos “*airplane2*” e “*airplane12*”, definidos por 57 e 86 pontos, respectivamente.

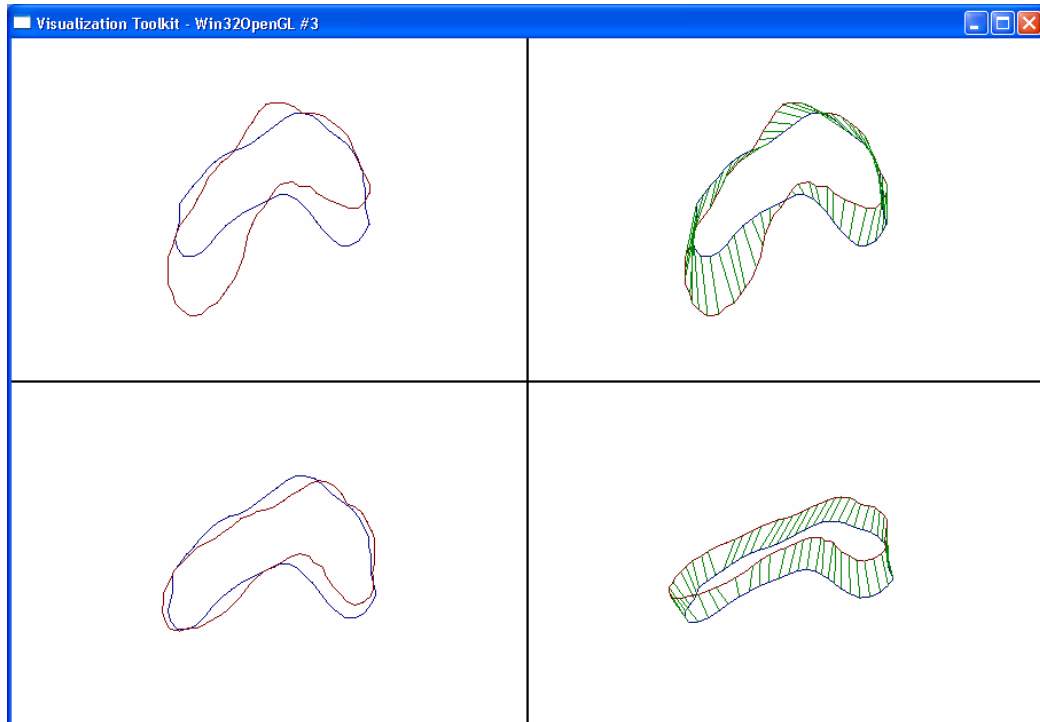


Figura 6.10: Emparelhamento baseado em informação de curvatura dos contornos “*foot6*” e “*foot7*”, definidos por 58 e 67 pontos, respectivamente.

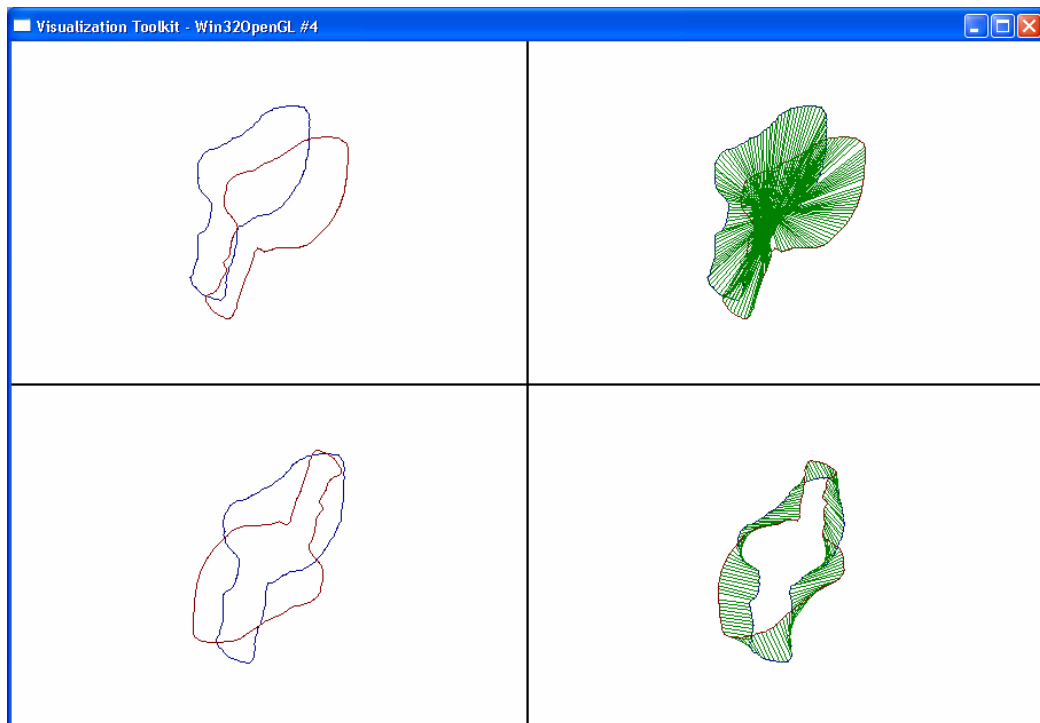


Figura 6.11: Emparelhamento baseado em informação de curvatura dos contornos “*heartB1*” e “*heartB2*”, definidos por 135 e 139 pontos, respectivamente.

Os tempos de execução totais para determinar a matriz de custos e os respectivos emparelhamentos são apresentados na Tabela 6.1. Na mesma, apresentam-se também os resultados relativos ao custo médio de emparelhamento por ponto e a amplitude do ângulo de rotação determinado pela metodologia apresentada no quarto capítulo. Os ensaios foram realizados num computador equipado com um processador *Intel Pentium III* a 1.0 GHz, com 256MB de RAM.

Tabela 6.1: Resultados dos emparelhamentos dos contornos ilustrados nas Figuras 6.2 a 6.11, usando a metodologia baseada em informação de curvatura e optimização com o algoritmo de programação dinâmica.

Número de pontos e “nome” do contorno		Custo médio de emparelhamento por ponto	Ângulo de rotação do contorno 2 em relação ao contorno 1 [°]	Tempo de execução [s]
Contorno 1	Contorno 2			
28, “heart1”	28, “heart1a”	0,096209	-14,9139	0
25, “heart3”	25, “heart4”	0	180,002	0
35, “heartA1”	35, “heartA2”	0,17929	-3,78273	0
28, “heart1”	84, “heart6”	0,076526	11,1027	0,01
28, “heart1”	81, “heart5”	0,07149	176,255	0,01
84, “heart6”	81, “heart5”	0,190316	-1,2454	0,01
46, “rib1”	46, “rib2”	0,17127	1,15815	0
57, “airplane2”	86, “airplane12”	0,161899	43,6113	0,01
58, “foot6”	67, “foot7”	0,174745	-25,6506	0
135, “heartB1”	139, “heartB2”	0,298391	180,594	0,01

A metodologia baseada em informação de curvatura é totalmente indiferente a transformações rígidas. Por outro lado, como cada elemento da sequência de ângulos depende apenas dos seus vizinhos mais próximos na sequência de pontos (antecessor e sucessor), o que acontece noutra região do contorno não terá qualquer influência no valor do ângulo de curvatura nesse ponto. Deste modo, uma oclusão de uma parte de um dos contornos a emparelhar não influenciará os valores da curvatura no restante contorno. Assim, é de esperar que esta metodologia apresente alguma robustez ao problema de oclusão de parte de um contorno. Apenas para comprovar este facto, são apresentadas as Figuras 6.12, 6.13 e 6.14 que ilustram essa situação. Assim, observe-se que, em cada uma destas figuras, foi retirada uma secção de um dos contornos envolvidos.

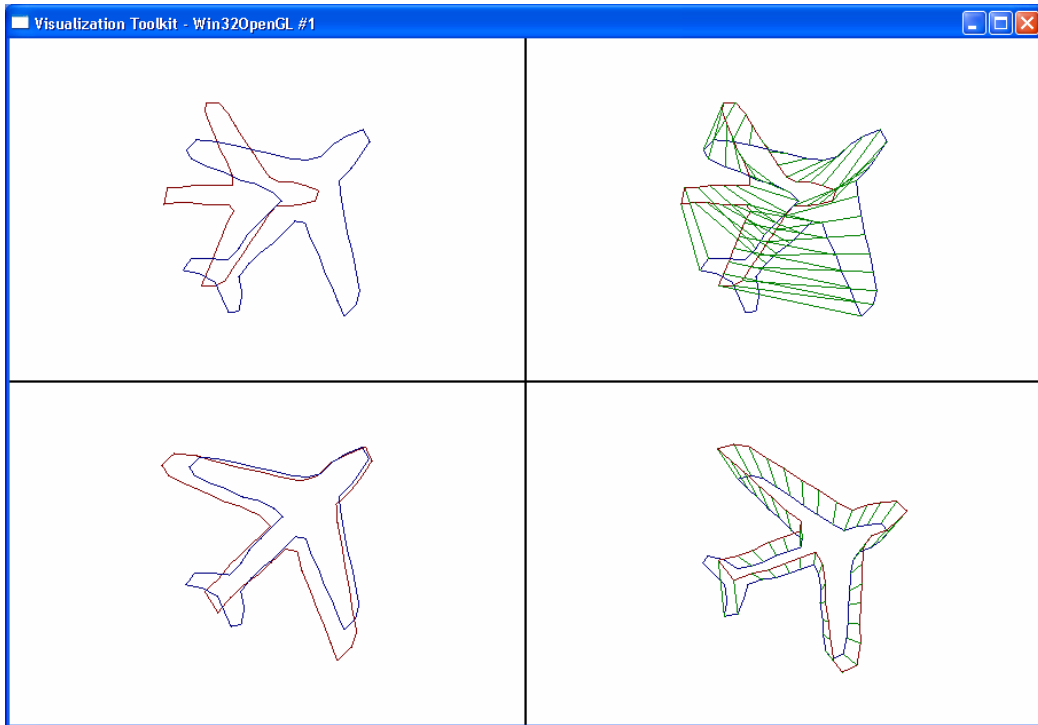


Figura 6.12: Emparelhamento baseado em informação de curvatura dos contornos “*airplane1a*” e “*airplane2*” definidos por 48 e 57 pontos, respectivamente.

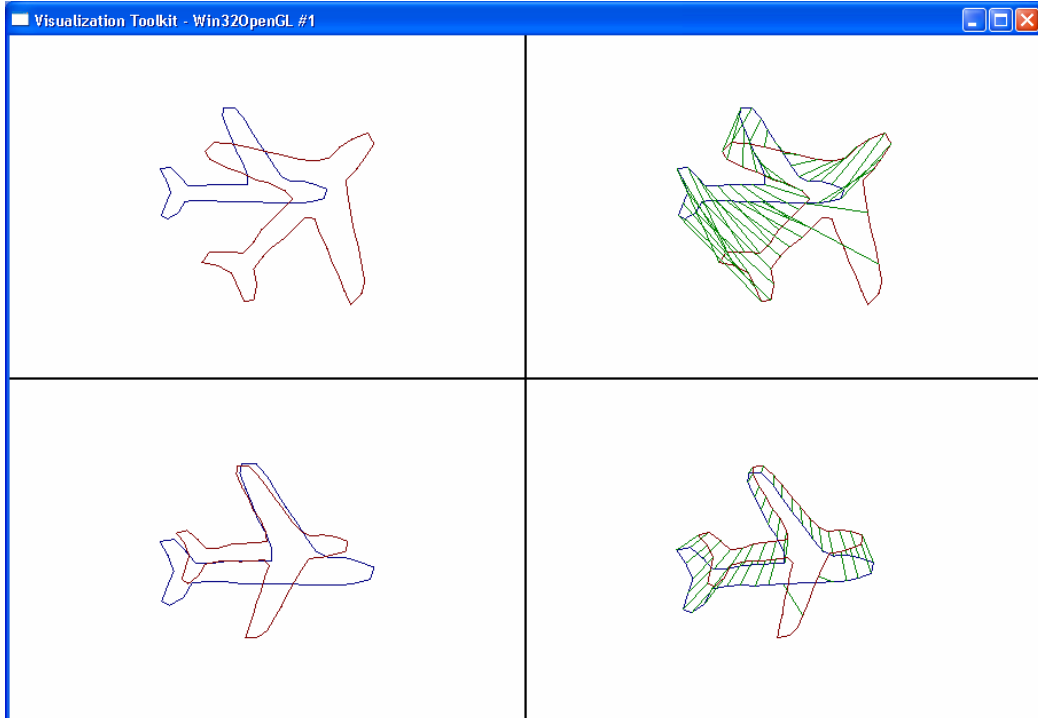


Figura 6.13: Emparelhamento baseado em informação de curvatura dos contornos “*airplane1b*” e “*airplane2*” definidos por 43 e 57 pontos, respectivamente.

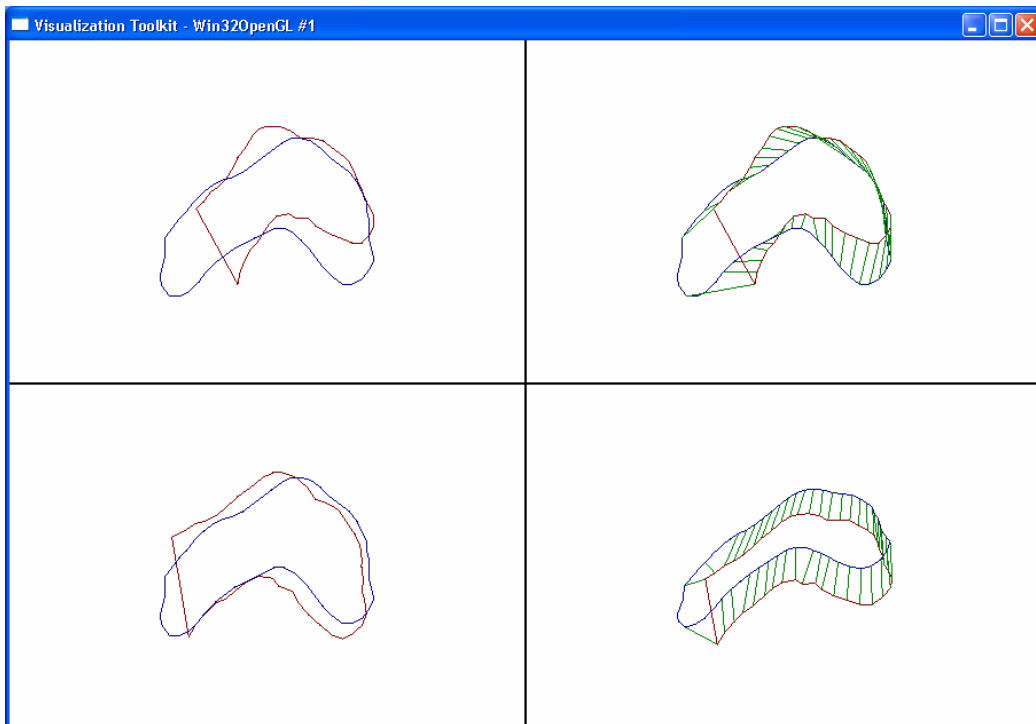


Figura 6.14: Emparelhamento baseado em informação de curvatura dos contornos “*foot6*” e “*foot7a*” definidos por 58 e 43 pontos, respectivamente.

6.2.3 Análise de resultados

De um modo geral, nas várias experiências realizadas, verificou-se que há um factor fundamental para a qualidade dos emparelhamentos obtidos: *a razão entre o número de pontos que define cada um dos contornos*. Além desta conclusão fundamental, podem referir-se as seguintes relativas à qualidade dos emparelhamentos obtidos:

- Sempre que os contornos foram definidos por conjuntos de pontos de reduzida dimensão e onde a diferença entre a dimensão dos mesmos não era acentuada, os emparelhamentos obtidos foram de boa qualidade. Mesmo no caso da oclusão de parte de um contorno, os emparelhamentos obtidos foram bons. À medida que a diferença entre o número de pontos que define cada um dos contornos aumentou, verificou-se uma tendência para a diminuição da qualidade dos emparelhamentos obtidos.
- No caso dos exemplos ilustrados nas Figuras 6.6 e 6.11, observam-se dois emparelhamentos sem sentido. Situações idênticas a estas só ocorreram quando um contorno estava definido por um número de pontos de cerca de duas vezes ou

mais o número de pontos que define o outro contorno ou os contornos estavam definidos por um elevado número de pontos cada um. No entanto, a diferença acentuada entre o número de pontos que define cada um dos contornos não obriga a que os emparelhamentos sejam necessariamente maus, veja-se o exemplo da Figura 6.5. Do mesmo modo, um elevado número de pontos a definir cada contorno também não obriga os emparelhamentos a serem de má qualidade.

- Os factos anteriores, permitem ainda concluir que quando o número de pontos que define um contorno é cerca de duas vezes ou mais o número de pontos que define o outro contorno, há alguma instabilidade nos emparelhamentos obtidos, isto é, tanto podem aparecer emparelhamentos sem sentido como emparelhamentos razoáveis. A instabilidade associada aos emparelhamentos de contornos definidos por elevado número de pontos verificou-se em contornos definidos por valores próximos ou superiores a uma centena de pontos.

Esta instabilidade nos emparelhamentos quando há uma diferença acentuada entre o número de pontos que define cada contorno pode ser explicada com o facto de que, em média, o contorno definido por maior número de pontos apresentar menor diferença entre os valores dos ângulos de curvatura do que o contorno definido por menor número de pontos. Deste modo, há uma suavização do polígono associado ao contorno definido por maior número de pontos. Assim, forçosamente há uma diferença muito significativa entre as amplitudes das sequências de ângulos definidas por cada contorno. Esta mesma suavização justifica ainda a instabilidade dos emparelhamentos de contornos definidos por elevado número de pontos.

Este algoritmo baseado em informação de curvatura revela alguma robustez ao problema da oclusão de partes de um contorno, estando garantidas as condições anteriormente referidas, ou seja, contornos definidos por reduzido número de pontos e reduzida diferença entre o número de pontos que define cada um dos contornos a emparelhar, pois estas condições são importantes para garantir estabilidade aos emparelhamentos obtidos.

Em relação ao cálculo dos custos dos emparelhamentos, pode-se observar que quando os contornos são definidos por um elevado número de pontos, os valores apresentados podem não fazer sentido. Veja-se o exemplo das Figuras 6.5 e 6.6. Na Figura 6.5 o emparelhamento é razoável e na Figura 6.6 o emparelhamento não faz sentido; no entanto, o custo indicado para o emparelhamento ilustrado na Figura 6.5 é superior ao custo do

emparelhamento ilustrado na Figura 6.6. Além disso, os contornos e respectivos números de pontos são idênticos.

Para contornos definidos por reduzido número de pontos e onde não haja uma diferença significativa entre esses valores, o custo indicado já pode ser utilizado como uma medida de similaridade entre contornos.

Em relação à velocidade de execução para determinar a matriz de custos e respectivo emparelhamento, verificou-se que esta foi sempre muito elevada, quer para contornos definidos por igual ou diferente número de pontos. Tal pode ser observado na Tabela 6.1.

Quanto à qualidade da transformação rígida obtida, sempre que a qualidade dos emparelhamentos foi no mínimo razoável, o ângulo de rotação determinado pareceu sempre coerente. Quanto à translação e escala determinadas, a observação das diversas imagens permitem concluir que foram sempre obtidos valores de boa qualidade.

Em relação à determinação da transformação rígida no caso de oclusão de parte de um contorno, os valores da translação e escala foram obviamente influenciados, pois dependem de todo o contorno, Figuras 6.12, 6.13 e 6.14. Em relação à rotação, apesar de na metodologia utilizada para o cálculo desta ser necessário aplicar previamente a translação e escala, esta mostrou-se mesmo assim correcta, consequência dos bons emparelhamentos obtidos.

6.3 Emparelhamento baseado na distância ao centróide

6.3.1 Princípio da metodologia

Quando é aplicada a uma forma apenas uma rotação ou uma translação, a distância de cada ponto ao centro da respectiva forma mantém-se invariante. Deste modo, quando dois contornos são semelhantes, depois de determinada a escala que transformou um contorno no outro e aplicada a um deles de forma que fiquem com igual dimensão, a propriedade enunciada pode ser usada para fazer o emparelhamento entre os pontos que define cada um dos contornos.

Para apresentar esta metodologia, suponhamos que temos dois contornos, o *contorno 1* definido por n pontos e o *contorno 2* definido por m pontos. Sejam \bar{X}_1 a média ponderada das distâncias ao centróide do *contorno 1*, calculadas como descrito na secção 4.3, e (x_c, y_c) as coordenadas do centróide do *contorno 1*, calculados como descrito na secção

4.2. Consideremos a sequência das distâncias ponderadas dos pontos do *contorno 1* ao seu centróide: $d1_1, d1_2, d1_3, \dots, d1_n$. Para um *ponto i* do *contorno 1* de coordenadas (x_i, y_i) , tem-se:

$$d1_i = \frac{\sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}}{\bar{X}_1}. \quad (6.1)$$

Do modo idêntico, constrói-se a sequência das distâncias ponderadas ao centróide do *contorno 2*: $d2_1, d2_2, d2_3, \dots, d2_m$, obviamente considerando o centróide do *contorno 2* e a respectiva média da distância entre os pontos que o definem e o seu centróide.

Seguidamente, calcula-se a matriz de custos C , a qual será dada pelo valor absoluto das diferenças entre as duas sequências de distâncias já construídas. Assim, temos:

$$C = \begin{bmatrix} |d1_1 - d2_1| & |d1_1 - d2_2| & \dots & |d1_1 - d2_m| \\ |d1_2 - d2_1| & |d1_2 - d2_2| & \dots & |d1_2 - d2_m| \\ \dots & \dots & \dots & \dots \\ |d1_n - d2_1| & |d1_n - d2_2| & \dots & |d1_n - d2_m| \end{bmatrix}.$$

Na fórmula (6.1), a divisão pela média das distâncias (\bar{X}_1 ou \bar{X}_2 , conforme o contorno) visa anular o efeito das diferentes dimensões (escala) dos contornos.

Refira-se que o cálculo das médias das distâncias ponderadas dos pontos ao centróide do respectivo contorno, fará com que o peso de cada ponto no cálculo desta média seja directamente proporcional à distância entre esse ponto e os seus dois vizinhos na sequência de pontos. Assim, diferentes amostragens não produzirão qualquer influência no cálculo da respectiva média.

Quando há deformações não rígidas num contorno em relação ao outro, a escala determinada poderá não ser correcta. Neste caso, para pequenos erros na escala, os emparelhamentos não serão influenciados. Contudo, para erros de escala consideráveis já se podem prever maus emparelhamentos. Em qualquer um dos casos anteriores, os valores dos custos de emparelhamento serão sempre influenciados.

6.3.2 Resultados dos emparelhamentos usando programação dinâmica

Com base na matriz de custos obtida pela metodologia apresentada na subsecção anterior,

recorreu-se ao algoritmo de programação dinâmica desenvolvido e apresentado na secção 5.4 para determinar o emparelhamento global que minimiza a soma dos custos associados aos pontos emparelhados.

O algoritmo de determinação de uma matriz de custos baseado na comparação da distância dos pontos dos contornos ao respectivo centróide foi desenvolvido em ambiente *Microsoft Visual C++*, com recurso à biblioteca de domínio público *VTK – Visualization Toolkit*.

Nas Figuras 6.15 a 6.26 podem ser observados alguns emparelhamentos entre os pares de contornos apresentados, *contorno 1* a azul e *contorno 2* a vermelho. Para cada figura indicada, à esquerda e em cima, apresentam-se os contornos nas posições originais e à esquerda e em baixo, os contornos após aplicação da transformação rígida, determinada pelo método descrito no capítulo quatro, ao *contorno 2*. À direita, em cima, os contornos nas posições originais e os respectivos emparelhamentos a verde e à direita, em baixo, uma perspectiva 3D dos emparelhamentos e contornos após aplicação da transformação rígida determinada. A opção pela representação dos emparelhamentos numa perspectiva 3D e apresentação dos contornos após aplicação da transformação rígida que melhor os alinha deve-se às razões já enunciadas na secção 6.2.

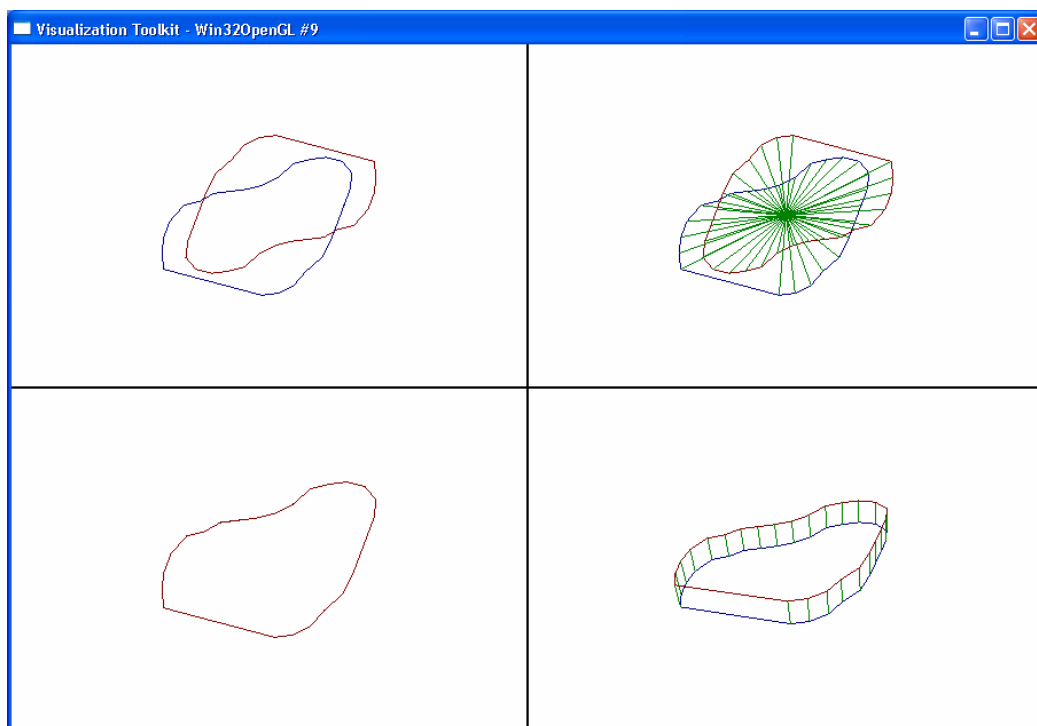


Figura 6.15: Emparelhamento baseado na distância ao centróide dos contornos “*heart3*” e “*heart4*”, definidos por 25 pontos cada um.

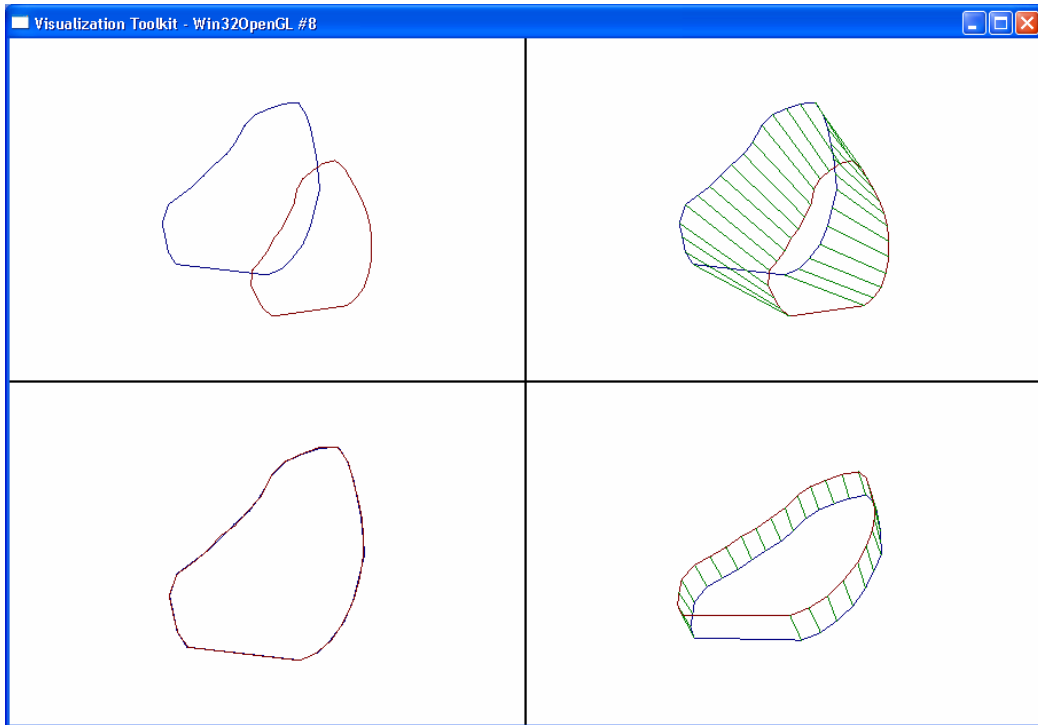


Figura 6.16: Emparelhamento baseado na distância ao centróide dos contornos “heart1” e “heart1a”, definidos por 28 pontos cada um.

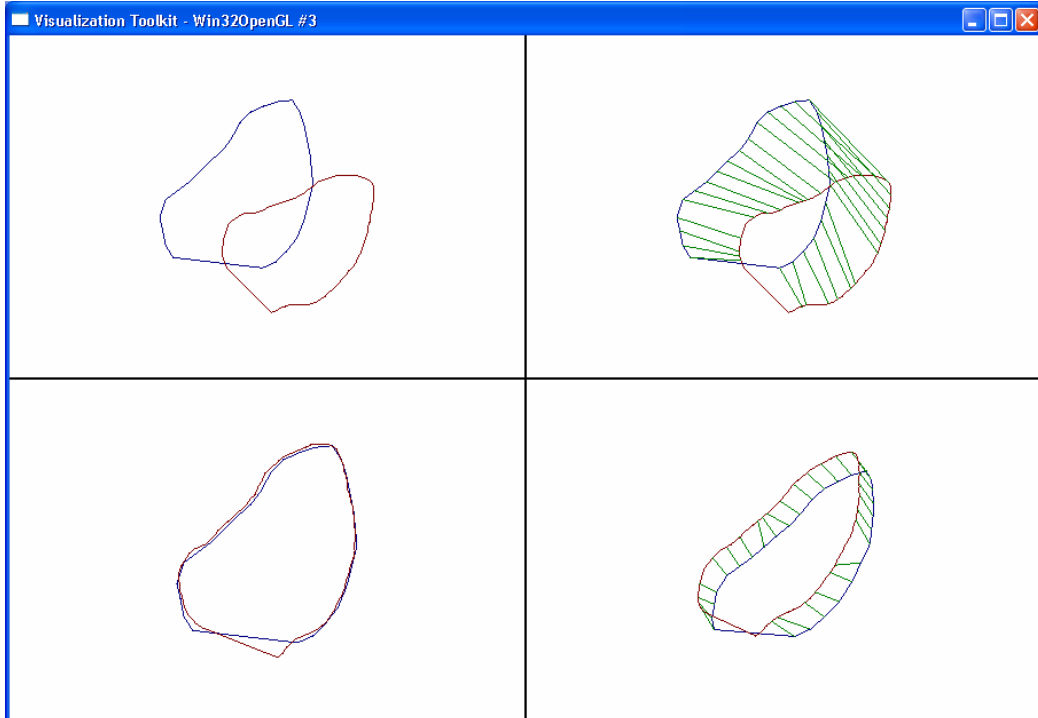


Figura 6.17: Emparelhamento baseado na distância ao centróide dos contornos “heart1” e “heart5”, definidos por 28 e 81 pontos, respectivamente.

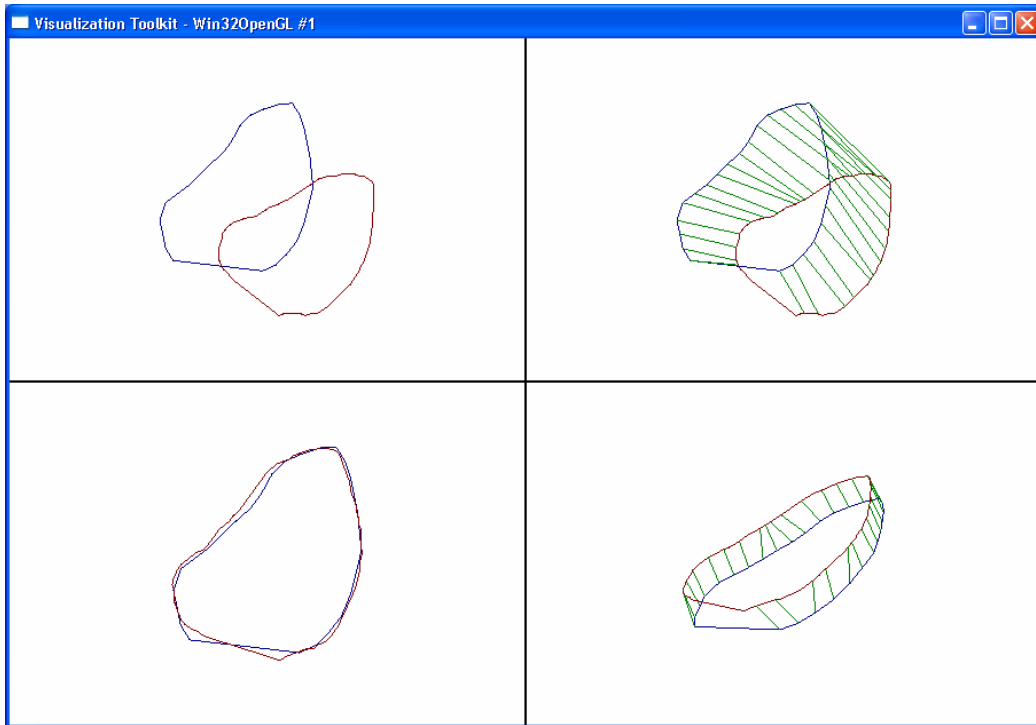


Figura 6.18: Emparelhamento baseado na distância ao centróide dos contornos “*heart1*” e “*heart6*”, definidos por 28 e 84 pontos, respectivamente.

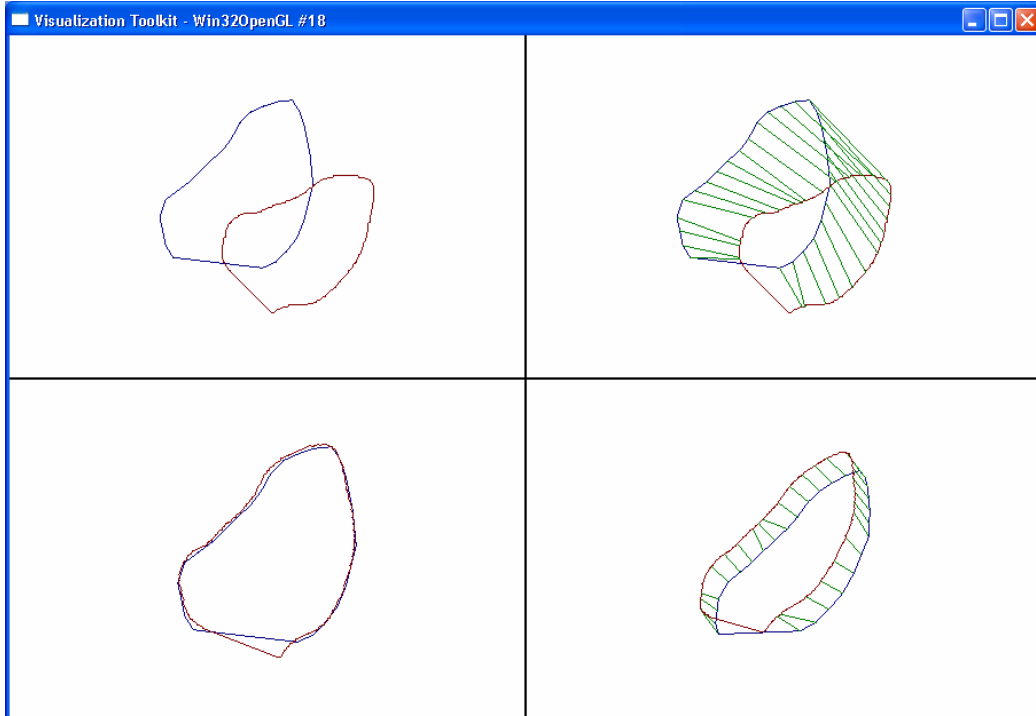


Figura 6.19: Emparelhamento baseado na distância ao centróide dos contornos “*heart1*” e “*heart9*”, definidos por 28 e 243 pontos, respectivamente.



Figura 6.20: Emparelhamento baseado na distância ao centróide dos contornos “heart6” e “heart5”, definidos por 84 e 81 pontos, respectivamente.

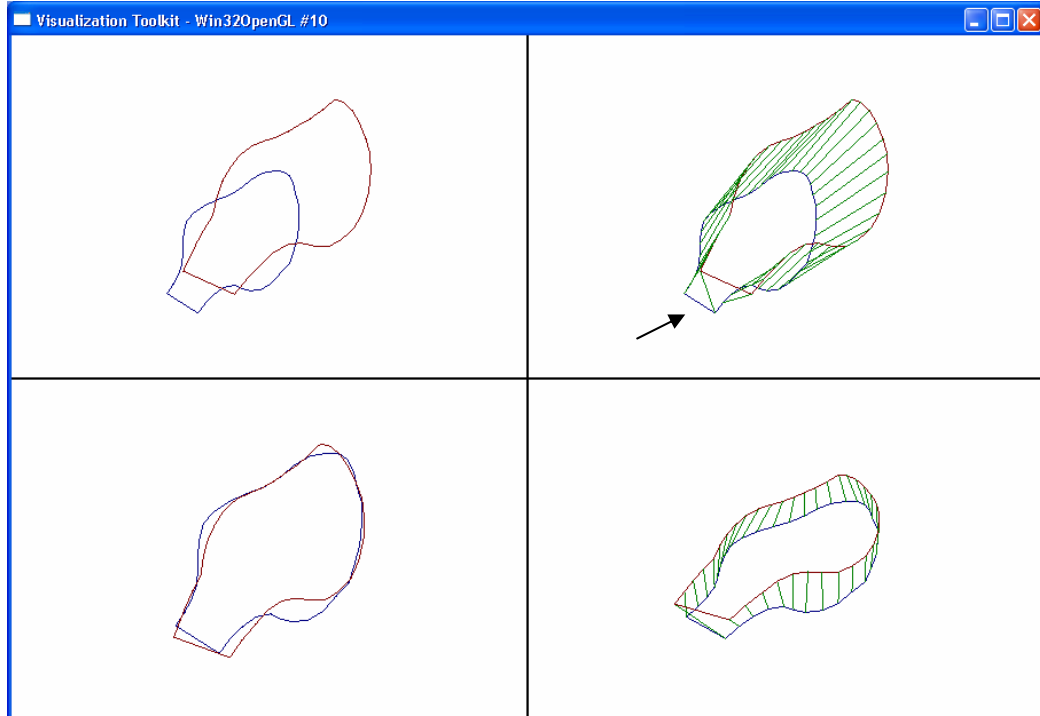


Figura 6.21: Emparelhamento baseado na distância ao centróide dos contornos “heartA1” e “heartA2”, definidos por 35 pontos cada um. (A seta representada assinala um emparelhamento possivelmente a melhorar ou a excluir.)

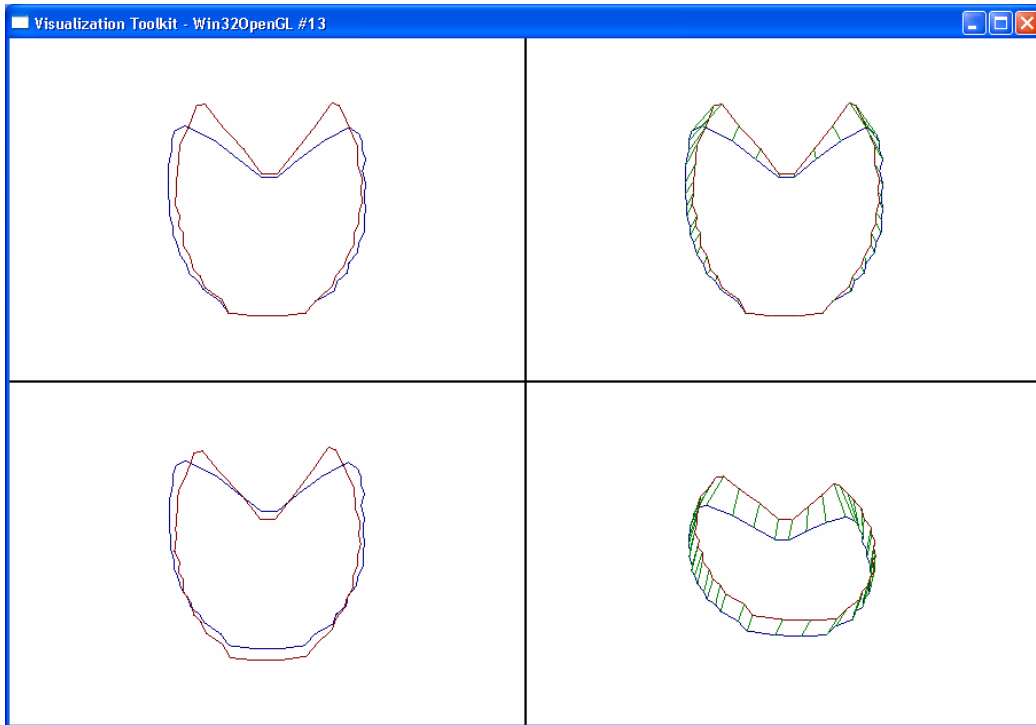


Figura 6.22: Emparelhamento baseado na distância ao centróide dos contornos “*rib1*” e “*rib2*”, definidos por 46 pontos cada um.

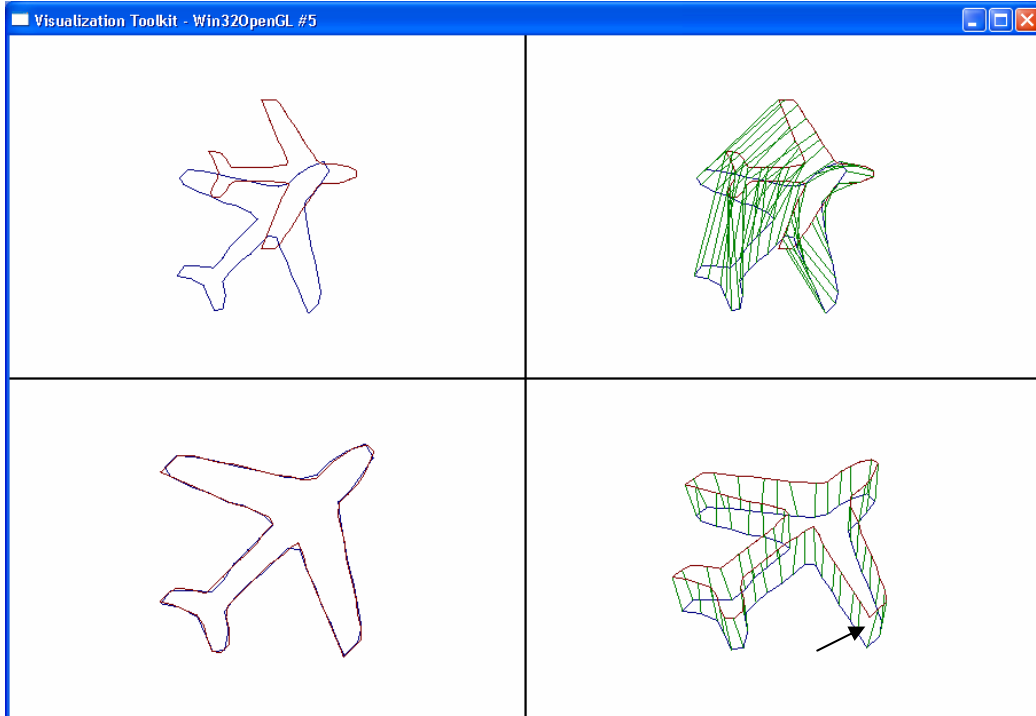


Figura 6.23: Emparelhamento baseado na distância ao centróide dos contornos “*airplane2*” e “*airplane12*”, definidos por 57 e 86 pontos, respectivamente. (A seta representada assinala um emparelhamento local a melhorar.)

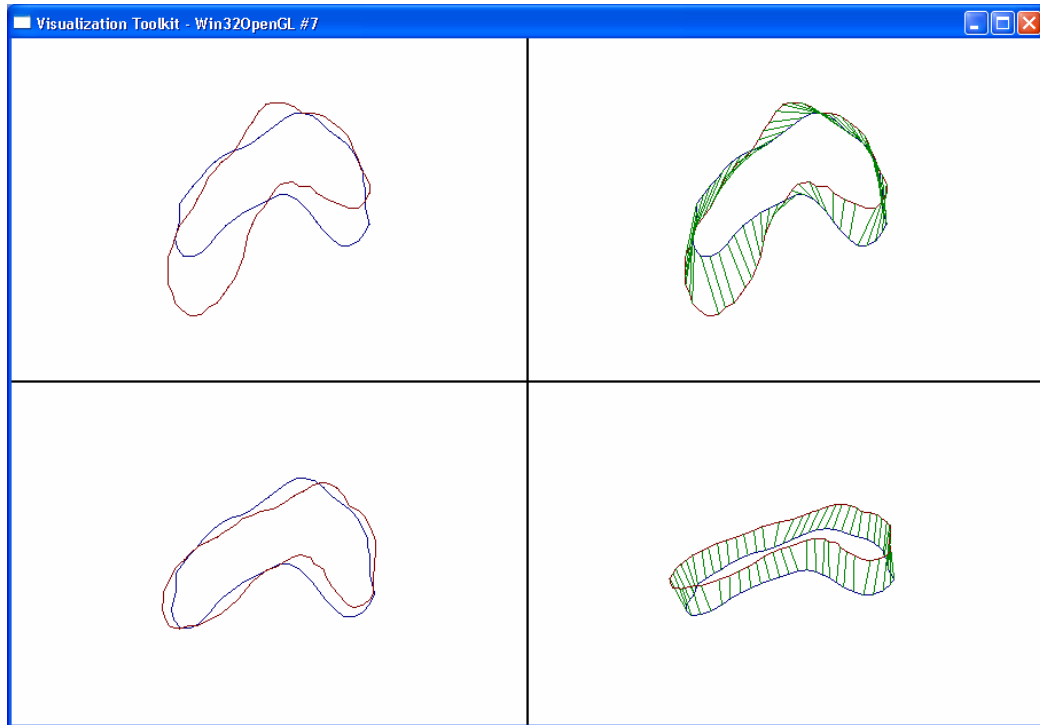


Figura 6.24: Emparelhamento baseado na distância ao centróide dos contornos “*foot6*” e “*foot7*”, definidos por 58 e 67 pontos, respectivamente.

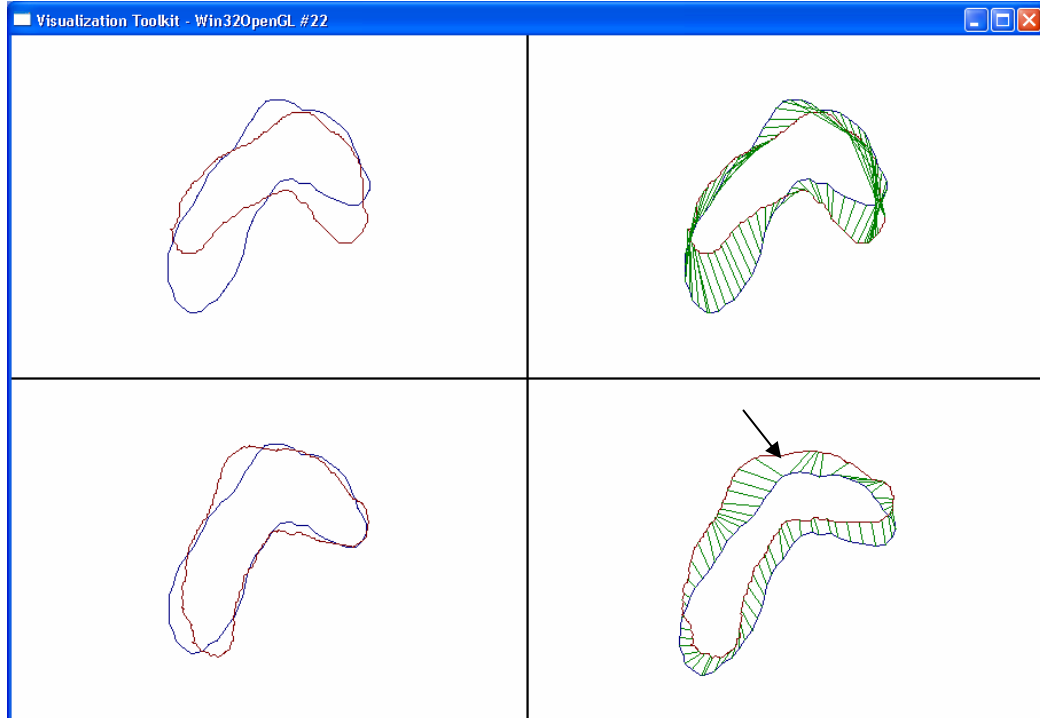


Figura 6.25: Emparelhamento baseado na distância ao centróide dos contornos “*foot7*” e “*foot13*”, definidos por 67 e 233 pontos, respectivamente. (A seta representada assinala uma região de emparelhamentos locais a melhorar.)

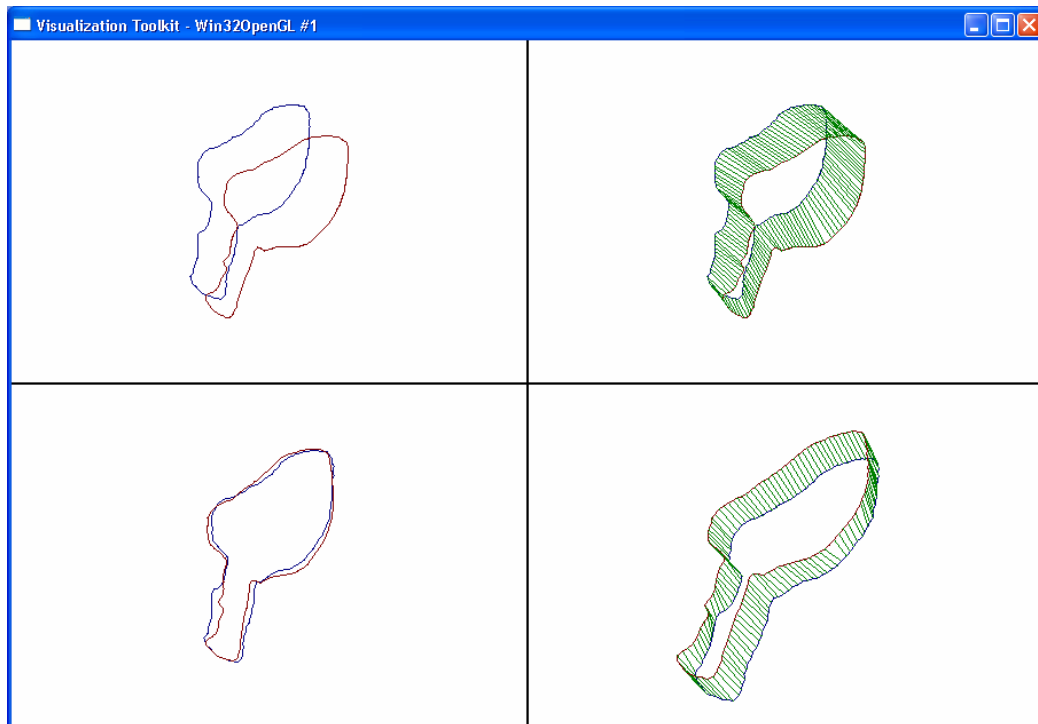


Figura 6.26: Emparelhamento baseado na distância ao centróide dos contornos “*heartB1*” e “*heartB2*”, definidos por 135 e 139 pontos, respectivamente.

Os tempos de execução totais para determinar a matriz de custos e os respectivos emparelhamentos são apresentados na Tabela 6.2. Na mesma tabela apresentam-se também os resultados relativos ao custo médio de emparelhamento por ponto e o ângulo de rotação determinado pela metodologia apresentada no quarto capítulo. Os ensaios foram realizados num PC equipado com um processador *Intel Pentium III* a 1.0 GHz, com 256MB de memória RAM.

Atendendo que na metodologia apresentada nesta secção a determinação do centróide e da escala são essenciais para a qualidade dos emparelhamentos obtidos, é de prever que esta metodologia apresente sensibilidade a problemas de oclusão de partes de um contorno. Tal deve-se ao facto de que a oclusão de uma parte de um contorno irá provocar um desvio no centróide e uma alteração para o valor da escala. Assim, foram realizadas algumas experiências para comprovar ou desmentir esta suposição. As Figuras 6.27, 6.28 e 6.29 ilustram os emparelhamentos obtidos utilizando os mesmos contornos que foram utilizados na secção anterior para o mesmo efeito. Deste modo, além de verificar a sensibilidade ou robustez ao problema da oclusão de partes do contorno por parte do algoritmo de emparelhamento baseado na comparação das distâncias aos respectivos centróides, pode-se também comparar os seus resultados com os apresentados pelo algoritmo baseado em

informação de curvatura, apresentado na secção anterior.

Tabela 6.2: Resultados dos emparelhamentos dos contornos ilustrados nas Figuras 6.15 a 6.26, usando a metodologia baseada nas distâncias aos centróides e otimização com o algoritmo de programação dinâmica.

Número de pontos e “nome” do contorno		Custo médio de emparelhamento por ponto	Ângulo de rotação do contorno 2 em relação ao contorno 1 [°]	Tempo de execução [s]
Contorno 1	Contorno 2			
25, “heart3”	25, “heart4”	0	180,003	0
28, “heart1”	28, “heart1a”	0,00639189	-14,9139	0
28, “heart1”	81, “heart5”	0,0130413	23,3915	0,01
28, “heart1”	84, “heart6”	0,015704	20,267	0,01
28, “heart1”	243, “heart9”	0,00638918	23,7785	0,13
84, “heart6”	81, “heart5”	0,0222509	3,65928	0
35, “heartA1”	35, “heartA2”	0,0710701	4,25485	0
46, “rib1”	46, “rib2”	0,0485709	1,15815	0
57, “airplane2”	86, “airplane12”	0,0252144	44,5453	0,02
58, “foot6”	67, “foot7”	0,0473267	-27,198	0
67, “foot7”	233, “foot13”	0,0185692	35,2316	0,23
135, “heartB1”	139, “heartB2”	0,0194957	9,85302	0,01

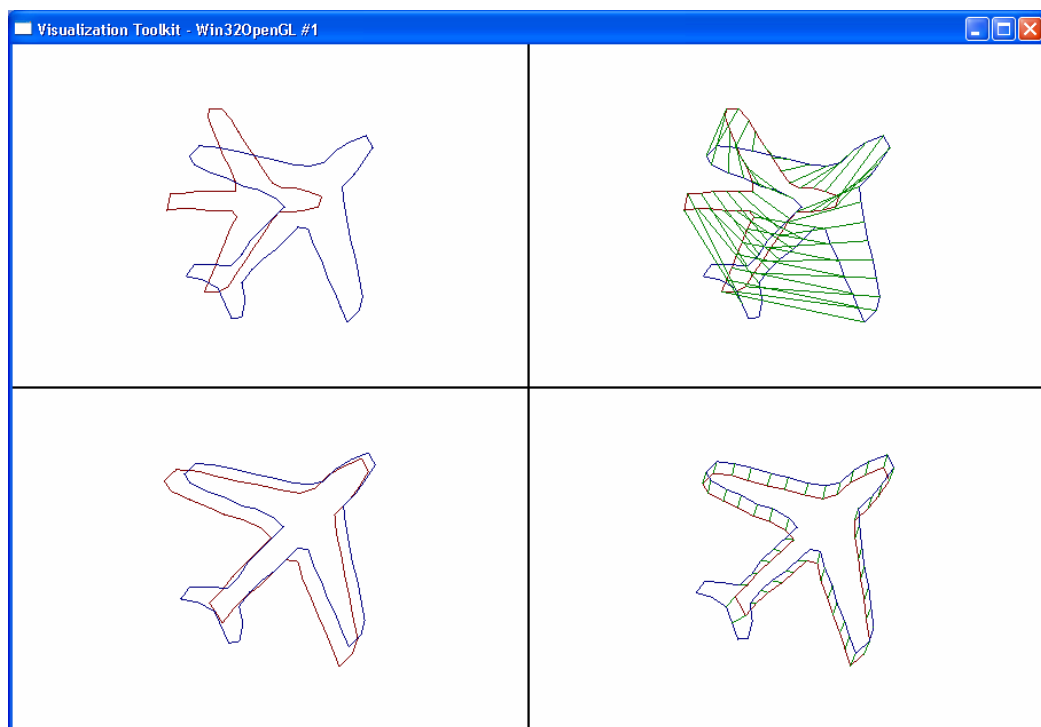


Figura 6.27: Emparelhamento baseado na distância ao centróide dos contornos “airplane1a” e “airplane2” definidos por 48 e 57 pontos, respectivamente.

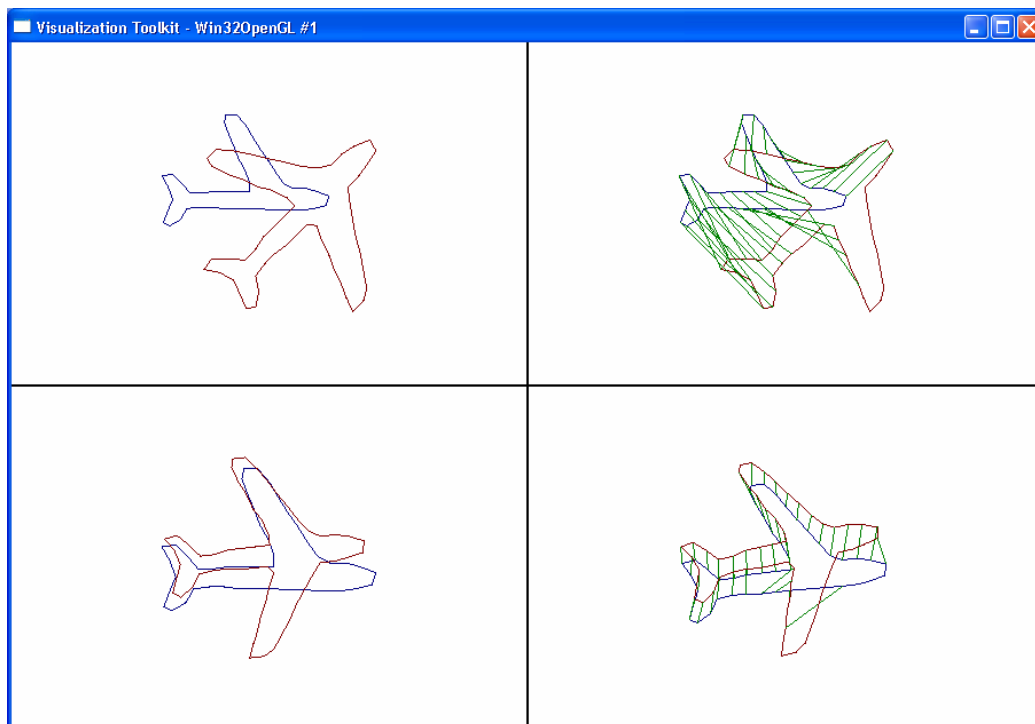


Figura 6.28: Emparelhamento baseado na distância ao centróide dos contornos “*airplane1b*” e “*airplane2*” definidos por 43 e 57 pontos, respectivamente.

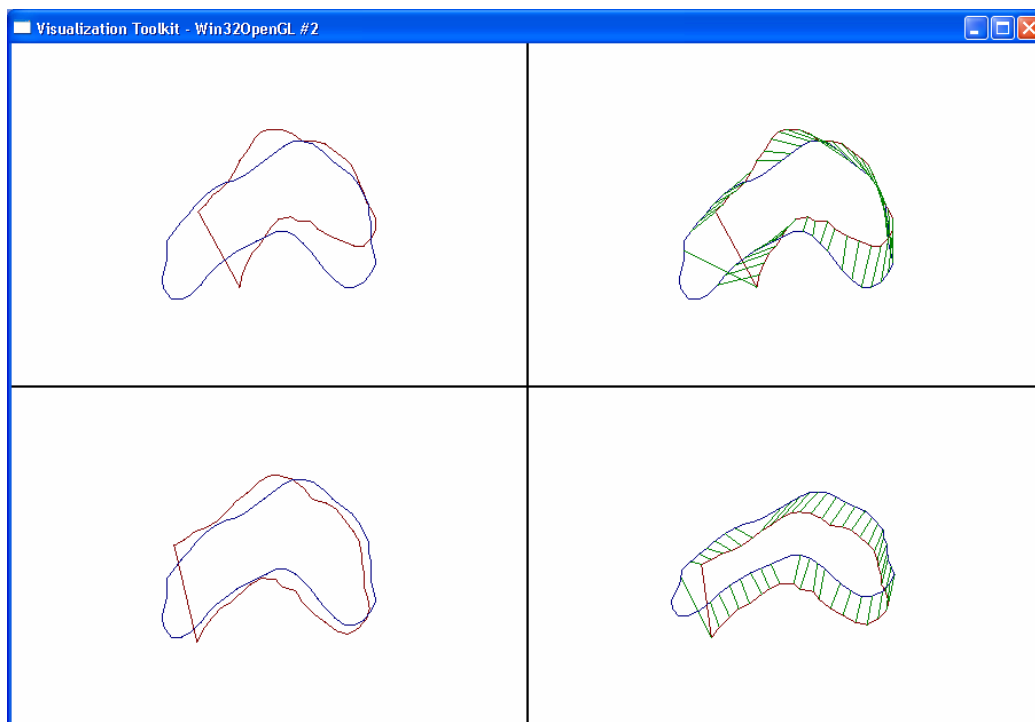


Figura 6.29: Emparelhamento baseado na distância ao centróide dos contornos “*foot6*” e “*foot7a*” definidos por 58 e 43 pontos, respectivamente.

6.3.3 Análise de resultados

Vamos analisar inicialmente os resultados verificados nas situações onde não houve oclusão de parte do contorno, Figuras 6.15 a 6.26. Assim, como se pode observar pelas figuras apresentadas na subsecção anterior, em todos os ensaios apresentados, o algoritmo alcançou sempre um emparelhamento global de boa qualidade para os contornos definidos, quer por muitos ou poucos pontos, onde a diferença entre o número de pontos que define cada um dos contornos não era acentuada. No caso dos emparelhamentos ilustrados nas Figuras 6.19 e 6.25, apesar da acentuada diferença entre o número de pontos que define cada um dos contornos, a qualidade dos emparelhamentos foi ainda de boa qualidade.

No caso das Figuras 6.23 e 6.25, verifica-se que há emparelhamentos locais que poderiam ser melhorados. Em relação à Figura 6.21, observa-se um emparelhamento local pouco coerente. Neste caso, não é fácil decidir se esse emparelhamento deve ser mudado, sendo que neste caso todos os outros emparelhamentos serão afectados pois os contornos são definidos pelo mesmo número de pontos. Outra hipótese seria a exclusão desse emparelhamento *a posteriori*. Escolher qual a melhor opção parece ser muito subjectivo.

Com estes dados, pode-se concluir que esta metodologia é adequada para o emparelhamento de contornos ordenados definidos por igual ou diferente número de pontos, sendo que por vezes alguns emparelhamentos locais parecem não ser os mais adequados. Outra conclusão a considerar é robustez desta metodologia ao ruído, que estava associado aos contornos definidos por um elevado número de pontos. Nesta situação, entenda-se por ruído: pequenas perturbações no alinhamento dos pontos, com pouco significado global mas bastante influentes ao nível local.

Passemos agora a analisar os valores apresentados para o custo médio dos emparelhamentos. Da observação das Figuras 6.15 a 6.18, pode-se concluir que a diferença entre os dois contornos apresentados em cada uma delas vai aumentando ligeiramente. O mesmo aconteceu ao valor dos custos apresentados pelo algoritmo, registados na Tabela 6.2. Em relação às restantes figuras apresentadas, não se consegue definir claramente, pela observação das mesmas, que pares de contornos são mais similares. Deste modo, não serão comentados os valores dos custos médios assinalados pelo algoritmo.

Em relação à velocidade de execução, esta foi boa em toda a diversidade de contornos utilizados nos ensaios. Quanto à qualidade da transformação rígida obtida que alinha os dois contornos a emparelhar, pela observação de cada uma das figuras apresentadas, conclui-se que esta foi sempre de boa qualidade.

Finalmente, nas situações nas quais houve oclusão de uma parte de um dos contornos a emparelhar, pela observação das respectivas figuras constata-se que, ao contrário do esperado, o algoritmo ainda conseguiu responder minimamente. No entanto, no caso da Figura 6.28, onde a parte retirada de um dos contornos já tem um peso considerável no mesmo, já aparecem mais emparelhamentos incorrectos do que os que surgiram usando a metodologia baseada em informação de curvatura apresentada na secção anterior.

6.4 Emparelhamento baseado em informação de curvatura e distância ao centróide

6.4.1 Princípio da metodologia

A ideia base desta metodologia é juntar a informação relativa à distância dos pontos ao centróide com informação de curvatura do contorno nos respectivos pontos que o definem. A informação da distância ao centróide garantirá alguma estabilidade global nos emparelhamentos e a informação de curvatura melhorará a sua qualidade ao nível local.

De uma forma simplificada, o algoritmo implementado procede como referido na subsecção 6.3.1 para construir as sequências das distâncias ponderadas ao centróide. Seguidamente, constroem-se as sequências de ângulos de curvatura como referido na secção 6.2.1. Deste modo, associa-se a cada ponto P de um qualquer contorno um vector de IR^2 , cujas componentes são: *distância ponderada de P ao centróide e ângulo de curvatura em P* .

Seguidamente, para cada ponto do *contorno 1* e cada ponto do *contorno 2*, é calculada a diferença das distâncias ponderadas ao centróide e as diferenças dos ângulos de curvatura. Seguidamente, são somados os dois valores, ponderando um peso adequado para cada um. Mais concretamente, tem-se o seguinte:

Sejam $d1_1, d1_2, d1_3, \dots, d1_n$ e $d2_1, d2_2, d2_3, \dots, d2_m$ duas sequências das distâncias ponderadas aos centróides dos respectivos *contornos 1* e *2*, calculadas como referido na subsecção 6.3.1. Sejam $\alpha_1, \alpha_2, \dots, \alpha_n$ e $\theta_1, \theta_2, \dots, \theta_n, \dots, \theta_m$ as sequências de ângulos de curvatura dos *contornos 1* e *2*, respectivamente, calculados como referido na subsecção 6.2.1.

Seja *medAng1* a média ponderada dos ângulos de curvatura do *contorno 1*. Esta média é calculada com a mesma ponderação utilizada para a determinação do centróide dos

contornos, descrito na secção 4.2. Seguidamente, constrói-se a matriz de custos C do modo que a seguir se expõe.

Seja $p \in [0, 1]$ o peso a atribuir à componente da diferença das distâncias ao centróide na matriz de custos. Tem-se que $(1-p)$ representa o peso a atribuir à diferença dos ângulos de curvatura. Assim, cada valor c_{ij} da matriz de custos C é calculado como:

$$c_{ij} = p \times |d1_i - d2_j| + (1-p) \times \frac{|\alpha_i - \theta_j|}{medAng1}.$$

Note-se que os valores $d1_i$ e $d2_j$ vão variar em torno do valor 1, basta observar a metodologia utilizada na subsecção 6.3.1. Tem-se ainda que:

$$\frac{|\alpha_i - \theta_j|}{medAng1} = \left| \frac{\alpha_i}{medAng1} - \frac{\theta_j}{medAng1} \right|.$$

Assim, os valores:

$$\frac{\alpha_i}{medAng1} \text{ e } \frac{\theta_i}{medAng1},$$

variam também em torno do valor 1, considerando que os dois contornos são semelhantes. Repare-se que se fossem determinados os quocientes dos valores das sequências de ângulos $\alpha_1, \alpha_2, \dots, \alpha_n$ e $\theta_1, \theta_2, \dots, \theta_n, \dots, \theta_m$ utilizando diferentes divisores, estar-se-ia a provocar artificialmente diferenças entre os valores dos ângulos.

Após vários ensaios experimentais preliminares com diversos pares de contornos definidos por conjuntos de pontos de diferentes dimensões, chegou-se às seguintes conclusões:

- Para contornos definidos por reduzido número de pontos e pequena diferença entre o número de pontos que define cada um dos contornos, um valor baixo do parâmetro p (o que corresponde a uma grande influência da informação de curvatura) tem tendência para originar melhores emparelhamentos.
- Para contornos definidos por um elevado número de pontos ou grande diferença entre o número de pontos que define cada um, um grande valor do parâmetro p (o que corresponde a uma reduzida influência da informação de curvatura) tem tendência para originar melhores emparelhamentos.

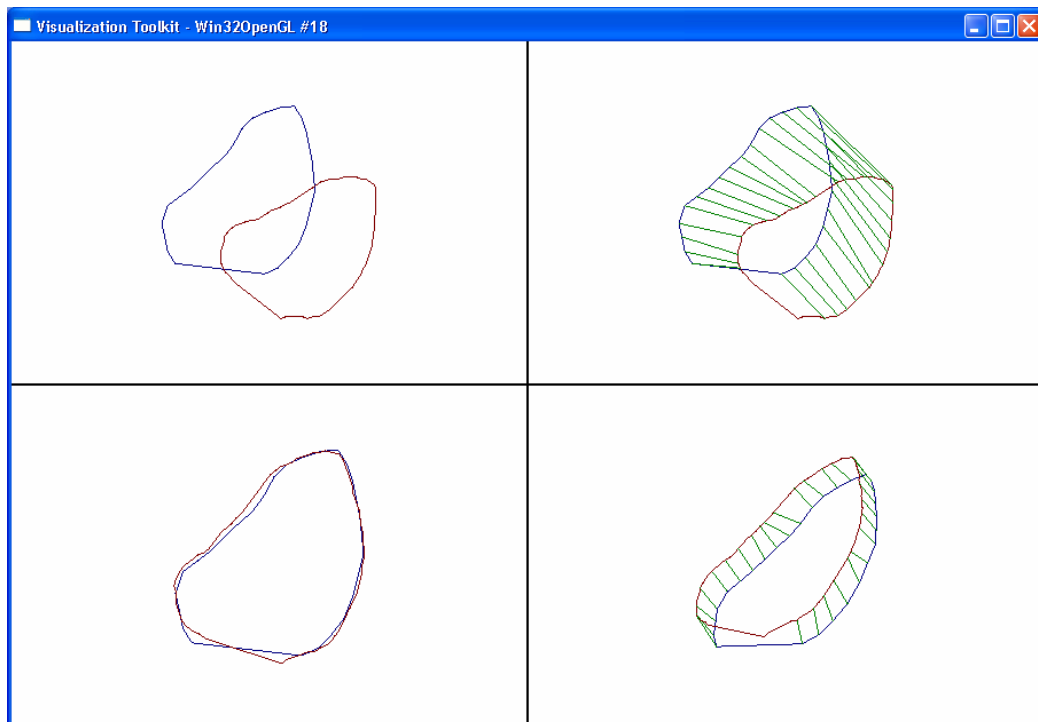
Os exemplos apresentados na secção seguinte foram obtidos com $p = 0,8$. Este valor

faz com que a informação das distâncias aos centróides tenha uma influência de cerca de 80% nos emparelhamentos, enquanto que a informação de curvatura tem uma influência de cerca de 20%. Com este valor, conseguiu-se a estabilidade da metodologia baseada nas distâncias aos centróides, garantido também bons emparelhamentos locais através da informação de curvatura, como se poderá constatar na subsecção seguinte.

6.4.2 Resultados dos emparelhamentos usando programação dinâmica

Tal como nas secções 6.2 e 6.3, para determinar o emparelhamento global que minimiza a soma dos custos de emparelhamento associados a cada par de pontos emparelhados, recorreu-se ao algoritmo de programação dinâmica com restrição de ordem apresentado na secção 5.4.

As Figuras 6.30 a 6.33 visam ilustrar alguns dos emparelhamentos obtidos. Optou-se por apresentar, apenas, imagens dos emparelhamentos onde foi possível observar diferenças em relação aos emparelhamentos obtidos pela metodologia apresentada na secção anterior. No entanto, os resultados do tempo de execução e dos custos globais de emparelhamento são todos apresentados na Tabela 6.3.



Figuras 6.30: Emparelhamento baseado em informação de curvatura e distância ao centróide dos contornos “heart1” e “heart6”, definidos por 28 e 84 pontos, respectivamente.

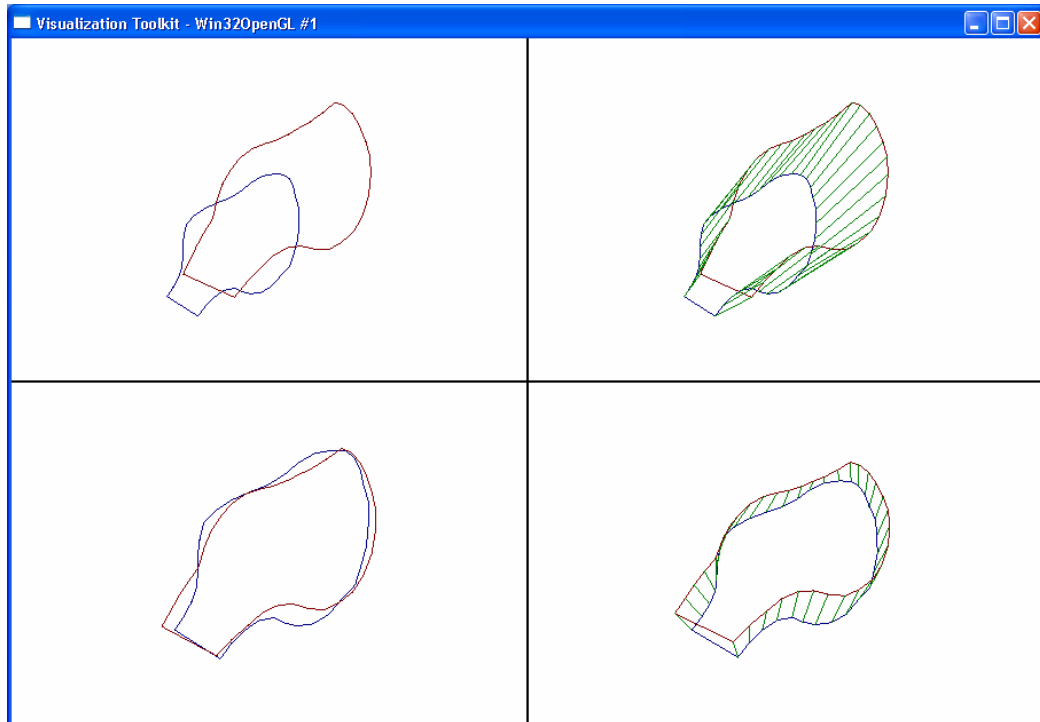


Figura 6.31: Emparelhamento baseado em informação de curvatura e distância ao centróide dos contornos “heartA1” e “heartA2”, definidos por 36 pontos cada um.

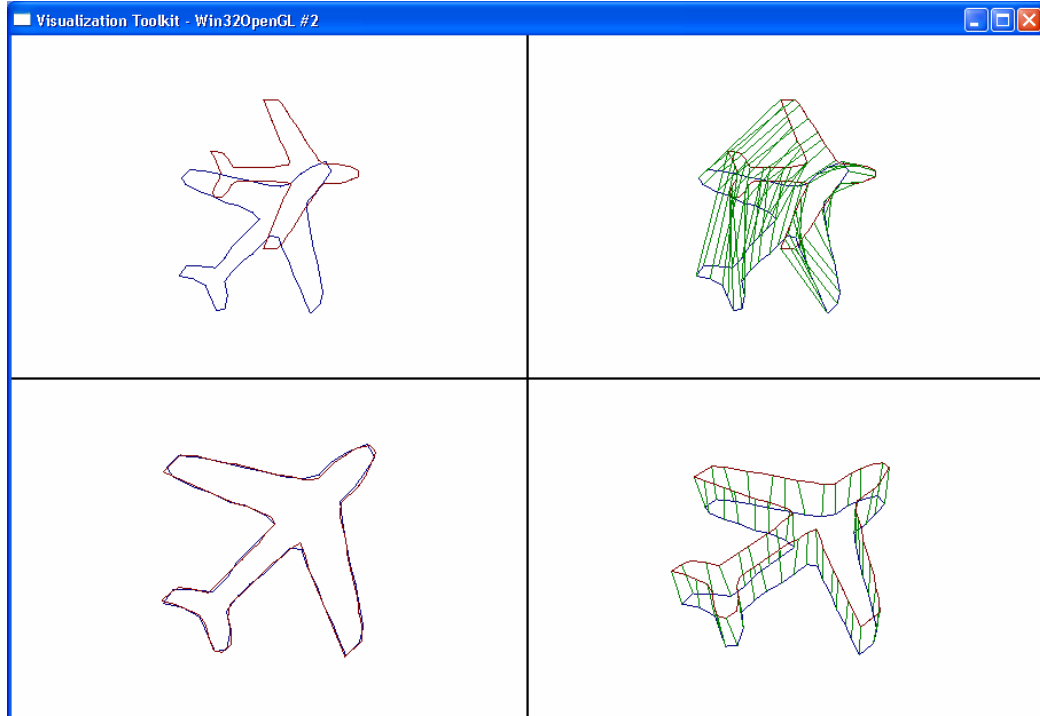


Figura 6.32: Emparelhamento baseado em informação de curvatura e distância ao centróide dos contornos “airplane2” e “airplane12”, definidos por 57 e 86 pontos, respectivamente.

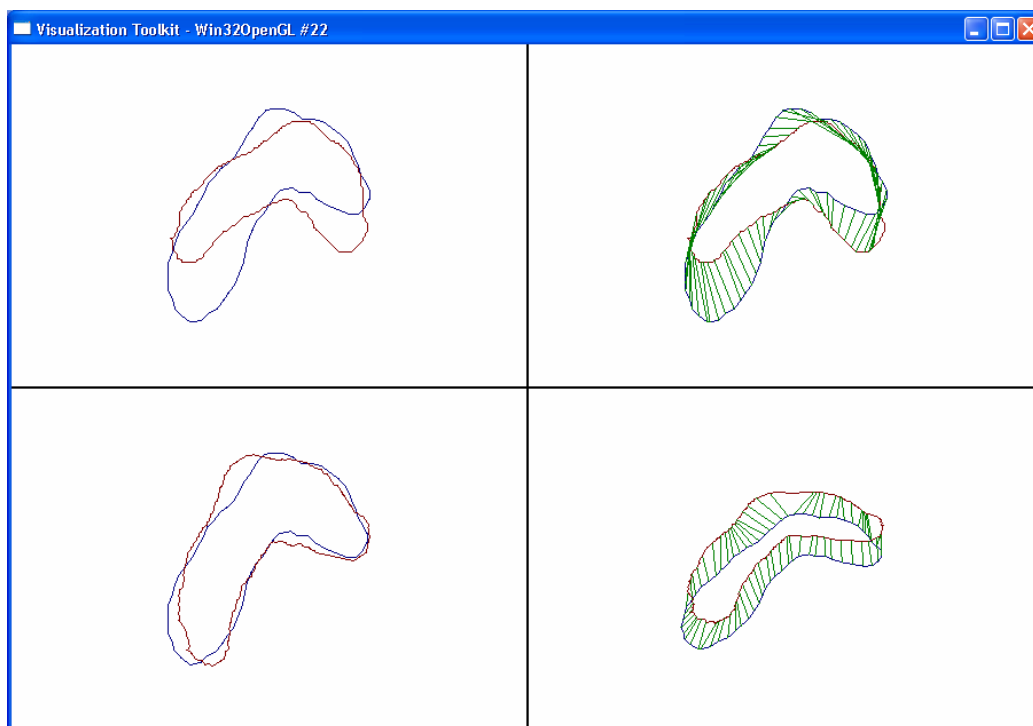


Figura 6.33: Emparelhamento baseado em informação de curvatura e distância ao centróide dos contornos “foot7” e “foot13”, definidos por 67 e 233 pontos, respectivamente.

Tabela 6.3: Resultados do emparelhamento dos diversos contornos de ensaio, usando a metodologia baseada em informação de curvatura e distância ao centróide, e otimização com o algoritmo de programação dinâmica.

Número de pontos e “nome” do contorno		Custo médio de emparelhamento por ponto	Ângulo de rotação do contorno 2 em relação ao contorno 1 [°]	Tempo de execução [s]
Contorno 1	Contorno 2			
25, “heart3”	25, “heart4”	0	180,003	0
28, “heart1”	28, “heart1a”	0,0107135	-14,9139	0
28, “heart1”	81, “heart5”	0,0223536	22,9437	0,01
28, “heart1”	84, “heart6”	0,0232339	19,6839	0,01
28, “heart1”	243, “heart9”	0,0178225	23,7402	0,13
84, “heart6”	81, “heart5”	0,032681	4,12871	0,01
35, “heartA1”	35, “heartA2”	0,0744347	-3,78273	0
46, “rib1”	46, “rib2”	0,0492854	1,15815	0
57, “airplane2”	86, “airplane12”	0,0324011	44,3622	0,02
58, “foot6”	67, “foot7”	0,0547642	-27,8792	0
67, “foot7”	233, “foot13”	0,0371012	32,7792	0,25
135, “heartB1”	139, “heartB2”	0,0386885	9,84156	0,01

As Figuras 6.34, 6.35 e 6.36 ilustram os emparelhamentos obtidos nas situações de oclusão de parte de um contorno, para os mesmos pares de contornos considerados nas duas secções anteriores.

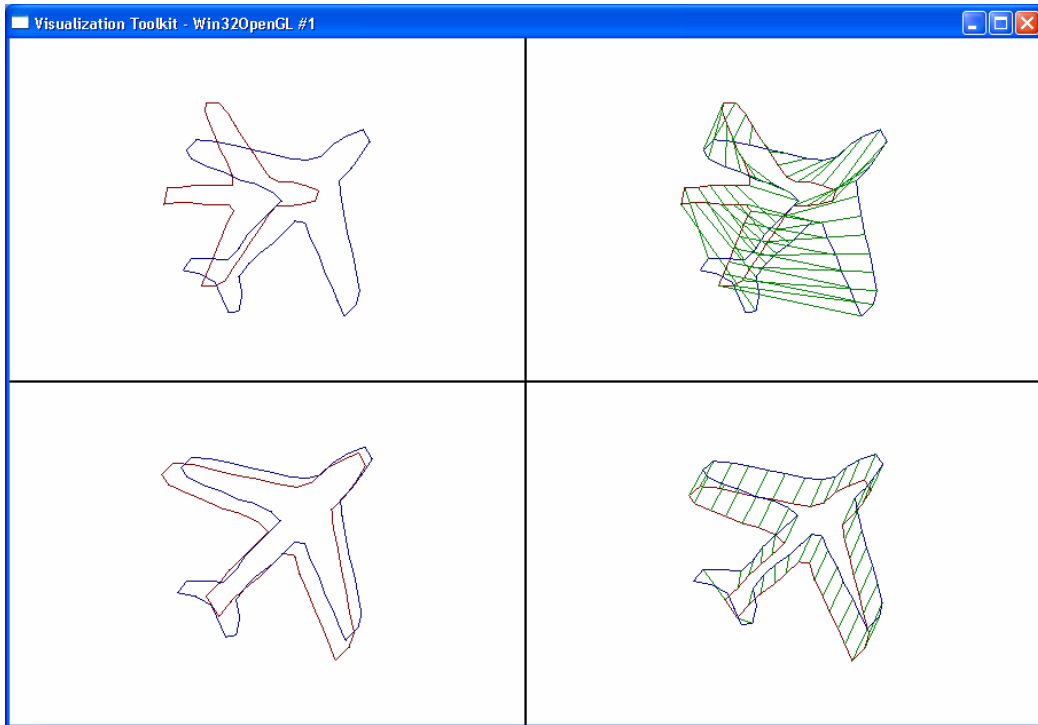


Figura 6.34: Emparelhamento baseado em informação de curvatura e distância ao centróide dos contornos “*airplane1a*” e “*airplane2*” definidos por 48 e 57 pontos, respectivamente.

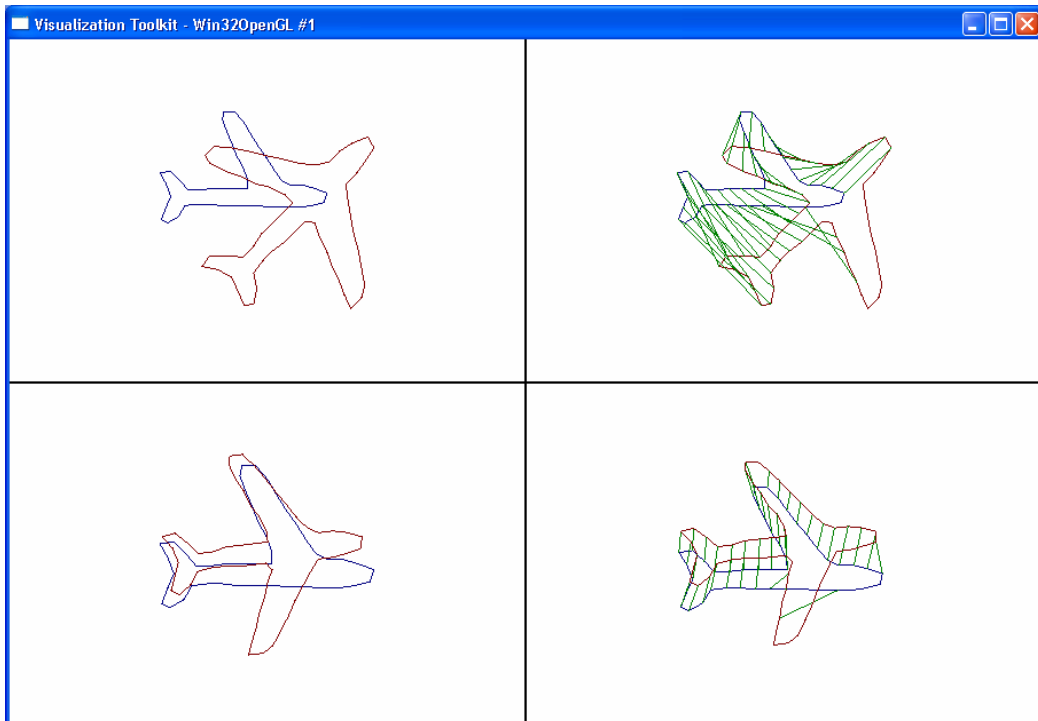


Figura 6.35: Emparelhamento baseado em informação de curvatura e distância ao centróide dos contornos “*airplane1b*” e “*airplane2*” definidos por 43 e 57 pontos, respectivamente.

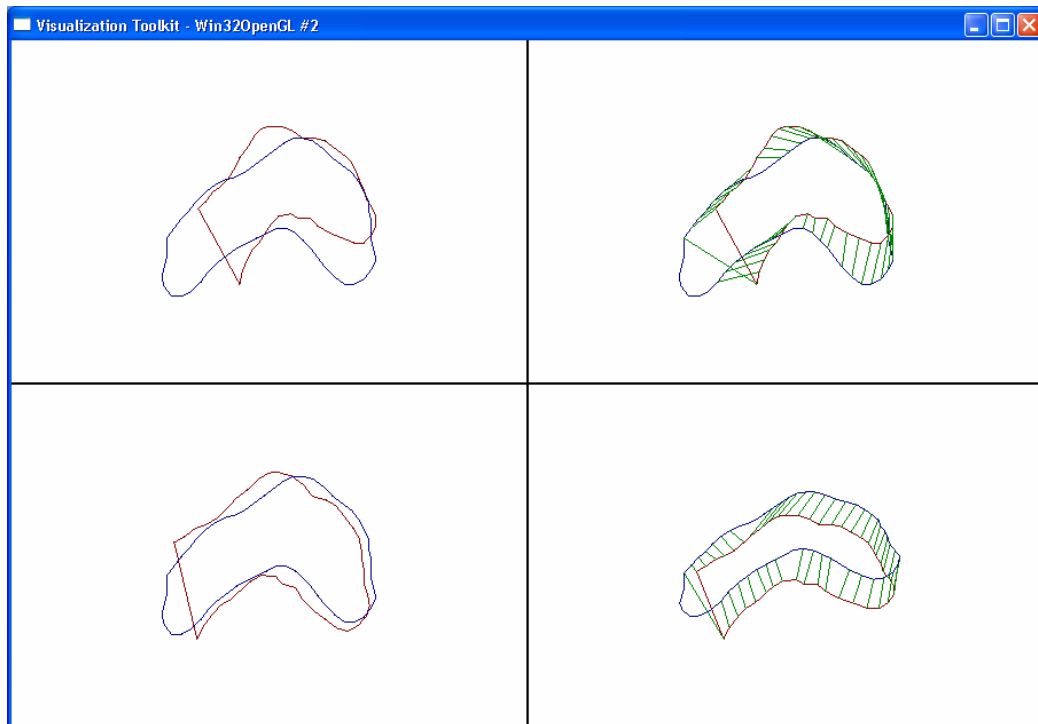


Figura 6.36: Emparelhamento baseado em informação de curvatura e distância ao centróide dos contornos “*foot6*” e “*foot7a*” definidos por 48 e 57 pontos, respectivamente.

6.4.3 Análise de resultados

Para a maior parte dos pares de contornos testados com esta metodologia baseada em informação de curvatura e distância dos pontos ao respectivo centróide, obtiveram-se os mesmos emparelhamentos, ou com diferenças imperceptíveis, que os emparelhamentos obtidos pela metodologia baseada apenas na comparação da distância dos pontos ao respectivo centróide, apresentada na secção anterior. Nos exemplos apresentados, houve apenas duas situações onde foi possível observar claramente melhorias em relação aos resultados apresentados na secção anterior, Figuras 6.32 e 6.33. Em relação às Figuras 6.30 e 6.31, aparentemente o emparelhamento obtido parece melhor do que o obtido na secção anterior para o mesmo par de contornos, no entanto esta observação parece bastante subjectiva.

Os exemplos aqui apresentados e os restantes ensaios realizados mostraram claramente que esta metodologia apresenta emparelhamentos adequados. De um modo geral, os emparelhamentos são melhores do que os da metodologia baseada apenas na comparação das distâncias ao centróide e muito mais consistentes do que os apresentados pela metodologia que utiliza apenas a informação de curvatura. Pode-se concluir, ainda, que tal

como pretendido, esta metodologia aproveita o que cada uma das metodologias apresentadas nas duas secções anteriores tem de melhor.

Em relação ao valor médio do custo de emparelhamento, as conclusões são as mesmas da secção anterior, isto é, parece haver uma tendência significativa para aumento dos custos médios de emparelhamento com o aumento das diferenças observadas nos respectivos contornos. Em relação à velocidade de execução do algoritmo, os valores indicados na Tabela 6.3 confirmam que esta é boa.

Quanto à determinação do ângulo de rotação que melhor alinha os dois contornos, exceptuando o caso do emparelhamento ilustrado na Figura 6.31, tem-se que os valores apresentados são os mesmos ou com ligeiríssimas diferenças em relação aos obtidos pela metodologia baseada apenas na comparação da distância dos pontos ao respectivo centróide. No caso do exemplo apresentado na Figura 6.31 há uma diferença significativa entre o valor do ângulo de rotação determinado com base na metodologia apresentada nesta secção e o determinado com base na metodologia apresentada na secção anterior. Dizer qual dos dois é mais correcto implica escolher qual dos dois emparelhamentos é melhor, e portanto, como já referido, esta questão parece algo subjectiva.

Finalmente, em relação aos emparelhamentos nas situações de oclusão de parte de um contorno, os resultados obtidos foram idênticos aos obtidos na secção anterior. Este facto era de esperar, pois o valor do parâmetro p (peso) utilizado foi de 0,8. Isto significa que as distâncias ao centróide têm um peso de cerca de 80% nos emparelhamentos, enquanto que a informação de curvatura tem um peso de cerca de 20%.

6.5 Emparelhamento com ajuste local

6.5.1 Princípio da metodologia

Em alguns exemplos de emparelhamentos, onde a diferença entre o número de pontos que define cada um dos contornos a emparelhar é muito acentuada, ver Figuras 6.30 e 6.33, parece ainda haver uma ligeira margem para possíveis melhoramentos. Assim, foi desenvolvido um algoritmo para realizar ajustes locais nos emparelhamentos obtidos inicialmente por uma qualquer metodologia do tipo *um-para-um*.

A ideia base é partir de um emparelhamento global do tipo *um-para-um*, alinhar os contornos aplicando a transformação rígida determinada e depois proceder a reajuste locais nos emparelhamentos no sentido de promover os emparelhamentos dos pontos em função

da sua proximidade em termos de distância euclidiana.

Suponhamos, então, que temos dois contornos, o primeiro definido por n pontos e o segundo definido por m pontos, em que $n < m$. Suponhamos ainda que já foi determinada a correspondência entre os pontos do *contorno 1* e os pontos do *contorno 2*, ficando $m - n$ pontos do *contorno 2* por emparelhar, pois são excedentários.

Para reajustar os emparelhamentos, com base nos emparelhamentos já obtidos, determina-se a transformação rígida que melhor alinha os dois contornos e procede-se à sua aplicação a um dos contornos. A translação, rotação e escala são determinadas conforme indicado no quarto capítulo. Agora, aplica-se o algoritmo de ajuste local, que consiste no seguinte:

Algoritmo:

1. Determina-se a soma das distâncias euclidianas entre os pares de pontos emparelhados.
2. Para todo i de 1 até n :
Se a distância do *ponto i* do *contorno 1* a um vizinho livre do seu par (ou parceiro de emparelhamento) é menor do que a distância ao seu par, então o *ponto i* do *contorno 1* passa a corresponder a essa ponto vizinho e o seu antigo par passa agora a estar livre.
3. Calcula-se a nova soma das distâncias entre os pontos agora emparelhados. Se for menor do que a anterior, regressa-se ao passo 2; caso contrário, termina-se a execução.

A Figura 6.37 ilustra o funcionamento do algoritmo desenvolvido, isto é, após o emparelhamento inicial que minimiza os custos com base numa matriz de custos, alguns emparelhamentos são modificados no sentido de minimizar a soma das distâncias entre pontos emparelhados. No entanto, só ocorreu mudança porque havia pontos por emparelhar na vizinhança dos pontos anteriormente emparelhados. Note-se que neste tipo de emparelhamento, emparelhamentos com mais afinidade, por exemplo, em termos de curvatura, podem ser substituídos por outros com menos afinidade mas de menor distância euclidiana.

Na Figura 6.37 pode ser observado um exemplo fictício deste tipo de emparelhamento. À esquerda, um emparelhamento obtido por uma qualquer metodologia inicial. À direita, o novo emparelhamento obtido após ajuste local dos emparelhamentos, utilizando a

metodologia apresentada nesta secção.

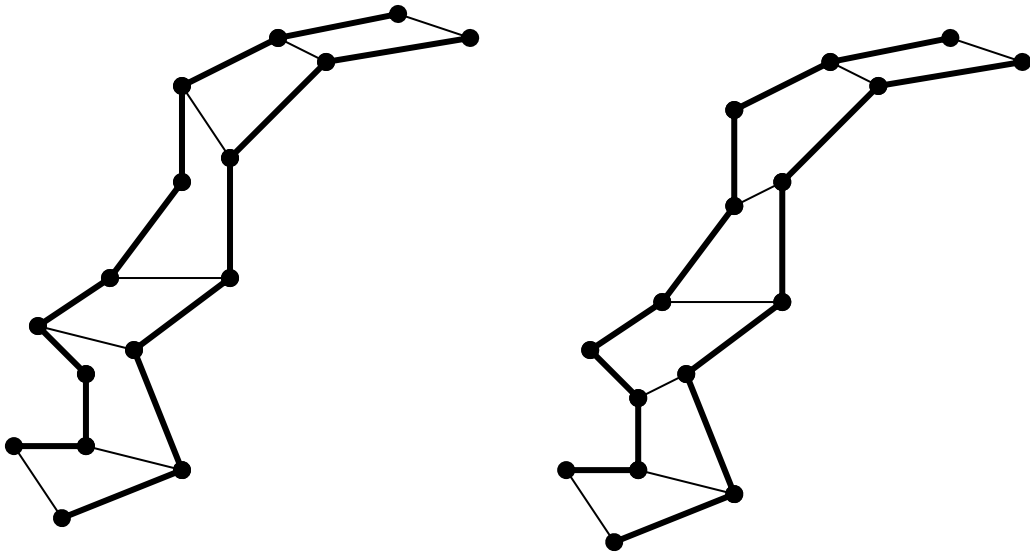


Figura 6.37: Exemplo fictício de um emparelhamento de duas secções de dois contornos com ajuste local. O traço grosso indica os contornos e o traço fino os emparelhamentos. À esquerda os emparelhamentos iniciais e à direita os emparelhamentos após ajuste local

6.5.2 Resultados e análise dos emparelhamentos

Este algoritmo parte de um emparelhamento do tipo *um-para-um* previamente determinado e baseia-se na minimização da distância entre os pontos após aplicação da transformação rígida que alinha os contornos. Assim, é fundamental para a obtenção de bons resultados que a metodologia inicial de emparelhamento do tipo *um-para-um* apresente emparelhamentos adequados. Caso contrário, o valor do ângulo de rotação que alinha os contornos poderá não fazer sentido e conseqüentemente o ajuste local também não será correcto.

As Figuras 6.38 a 6.41, a seguir apresentadas, ilustram alguns resultados deste tipo de emparelhamento, partindo de emparelhamentos iniciais obtidos com as metodologias baseadas em informação de curvatura e/ou distâncias aos centróides. Em todas estas figuras, as imagens representam uma perspectiva 3D dos contornos já alinhados e respectivos emparelhamentos.

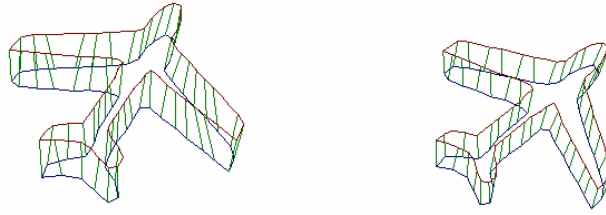


Figura 6.38: Emparelhamento dos contornos “*airplane2*” e “*airplane12*”. À esquerda usando apenas informação de curvatura e à direita usando informação de curvatura com ajuste local.



Figura 6.39: Emparelhamento dos contornos “*heart1*” e “*heart6*”. À esquerda usando apenas informação de curvatura e à direita usando informação de curvatura com ajuste local.



Figura 6.40: Emparelhamento dos contornos “*heart1*” e “*heart5*”. À esquerda usando apenas informação da distância ao centróide, à direita usando informação da distância ao centróide com ajuste local.



Figura 6.41: Emparelhamento dos contornos “*foot7*” e “*foot13*”. À esquerda usando informação de curvatura e distância ao centróide, à direita usando informação de curvatura e distância ao centróide com ajuste local.

Os emparelhamentos apresentados nas Figuras 6.38 a 6.41 mostram que esta metodologia de ajuste local apresenta bons resultados. Estes resultados tornam-se mais evidentes quando a metodologia base apresenta alguns maus emparelhamentos mas insuficientes para provocarem um erro considerável no ângulo de rotação estimado, Figuras 6.38 e 6.39. Conclui-se também que, obviamente, quanto melhores foram os emparelhamentos iniciais, menores alterações este algoritmo provocará. Finalmente, embora aqui não tenham sido apresentados exemplos que o justifiquem, por ser algo evidente, quanto menor for a diferença entre o número de pontos que define cada um dos contornos, menor alterações este algoritmo de ajuste poderá provocar.

6.6 Resultados dos emparelhamentos do tipo *um-para-vários*

6.6.1 Baseado numa matriz de custos

O algoritmo utilizado para efectuar o emparelhamento do tipo *um-para-vários*, tal como referido na sua apresentação na subsecção 5.6.2, parte de um emparelhamento inicial do tipo *um-para-um*, obtido por uma metodologia deste último tipo. Nas imagens a seguir apresentadas, Figuras 6.42 a 6.45, partiu-se sempre inicialmente do emparelhamento do tipo *um-para-um* obtido com a metodologia que usa simultaneamente as distâncias aos centróides e informação de curvatura. Em cada figura indicada, a linha azul assinala o *contorno 1* e a linha vermelha assinala o *contorno 2*, sendo representado à esquerda os contornos originais, ao centro os contornos e os respectivos emparelhamentos a verde e à direita uma perspectiva 3D dos contornos e emparelhamentos após aplicação da transformação rígida calculada.

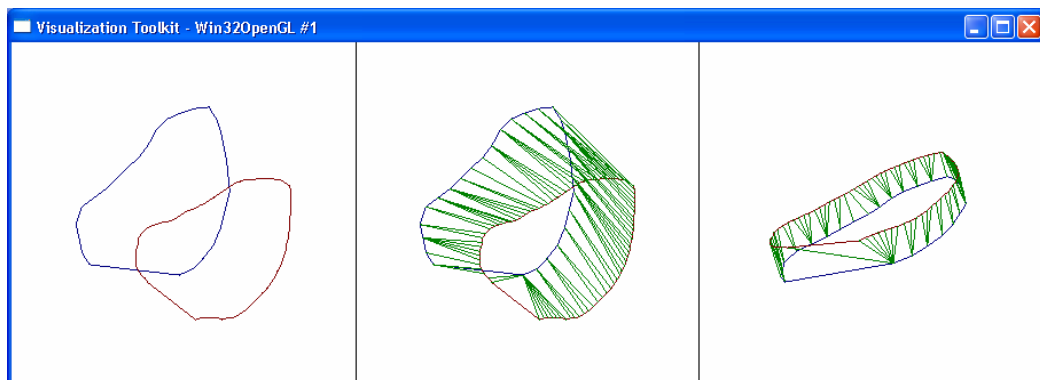


Figura 6.42: Emparelhamento do tipo *um-para-vários* dos contornos “*heart1*” e “*heart6*”, com base numa matriz de custos.

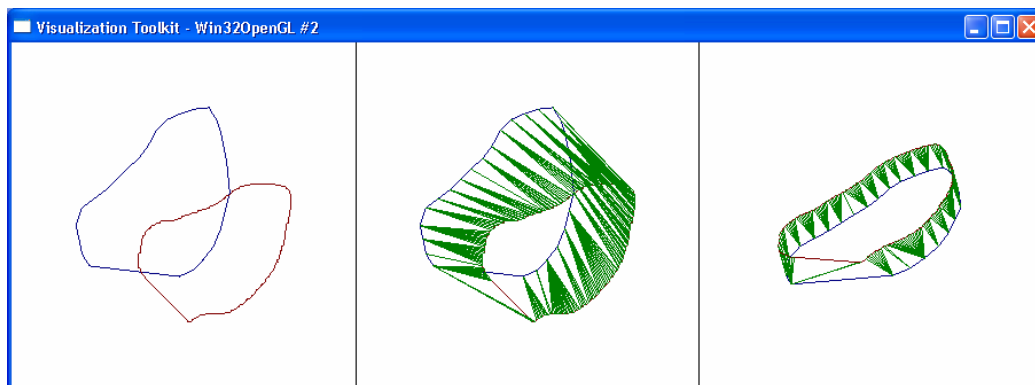


Figura 6.43: Emparelhamento do tipo *um-para-vários* dos contornos “heart1” e “heart9”, com base numa matriz de custos.



Figura 6.44: Emparelhamento do tipo *um-para-vários* dos contornos “airplane2” e “airplane12”, com base numa matriz de custos.

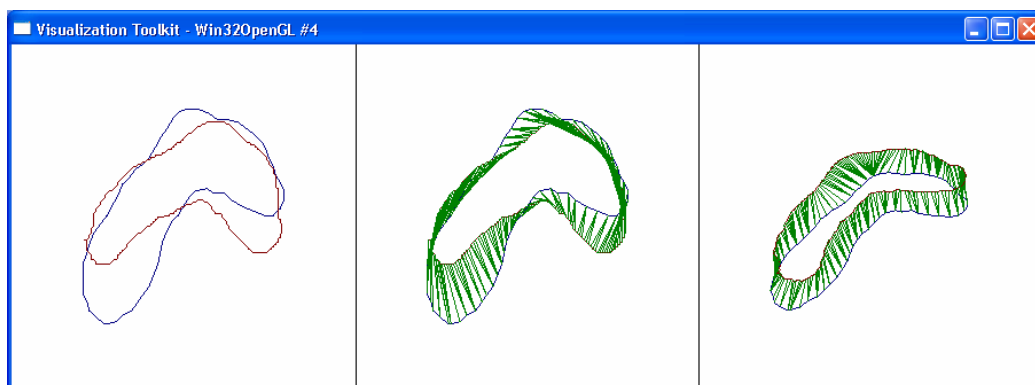


Figura 6.45: Emparelhamento do tipo *um-para-vários* dos contornos “foot7” e “foot13”, com base numa matriz de custos.

Na Tabela 6.4 estão apresentados os tempos de execução para efectuar o emparelhamento do tipo *um-para-vários* com base na metodologia baseada na matriz de

custos, já referida. O custo médio de emparelhamento por ponto apresentado é a média do custo de todos os emparelhamentos. O tempo de execução é total, isto é, inclui o tempo necessário para fazer o emparelhamento inicial do tipo *um-para-um* e o tempo para fazer o emparelhamento do tipo *um-para-vários*. Optou-se por utilizar os mesmos pares de contornos que foram utilizados na Tabela 6.3, pois deste modo será possível comparar os novos valores dos custos e tempos de execução. Como, na metodologia considerada, os emparelhamentos do tipo *um-para-vários* só são efectuados entre contornos definidos por diferente número de pontos, não são apresentados resultados para pares de contornos definidos pelo mesmo número de pontos.

Tabela 6.4: Custos e tempos de execução relativos a emparelhamentos do tipo *um-para-vários* usando a metodologia baseada numa matriz de custos.

Número de pontos e “nome” do contorno		Custo médio de emparelhamento por ponto	Tempo de execução [s]
Contorno 1	Contorno 2		
28, “heart1”	81, “heart5”	0,0369167	0,01
28, “heart1”	84, “heart6”	0,036442	0,02
28, “heart1”	243, “heart9”	0,0484862	0,13
84, “heart6”	81, “heart5”	0,033431	0
57, “airplane2”	86, “airplane12”	0,0447946	0,02
58, “foot6”	67, “foot7”	0,0582986	0,01
67, “foot7”	233, “foot13”	0,0553837	0,251
135, “heartB1”	139, “heartB2”	0,0383312	0,02

6.6.2 Baseado na minimização da distância euclidiana entre pontos a emparelhar

Tal como nos ensaios descritos na subsecção anterior, nos resultados a seguir apresentados partiu-se inicialmente dos emparelhamentos do tipo *um-para-um* obtidos pela metodologia que utiliza simultaneamente as distâncias dos pontos aos respectivos centróides e informação de curvatura. Depois foi utilizado o algoritmo de emparelhamento do tipo *um-para-vários* apresentado na subsecção 5.6.3, o qual faz os emparelhamentos em função da distância euclidiana entre pontos a emparelhar, após aplicação da transformação rígida que melhor alinha os dois contornos em questão. Para determinar a transformação rígida usou-se a metodologia proposta no quarto capítulo. As Figuras 6.46 a 6.49 ilustram alguns dos resultados obtidos. A disposição dos elementos das referidas figuras é a mesma da

subsecção anterior.

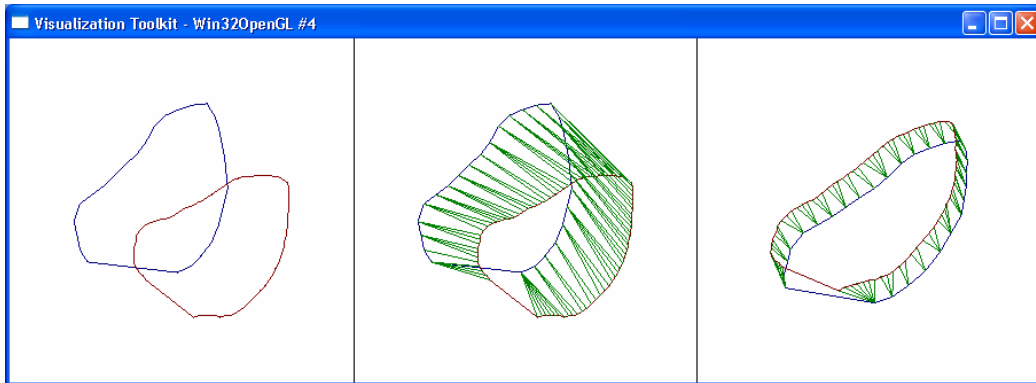


Figura 6.46: Emparelhamento do tipo *um-para-vários* dos contornos “heart1” e “heart6”, com base na minimização da distância euclidiana entre os pontos a emparelhar.

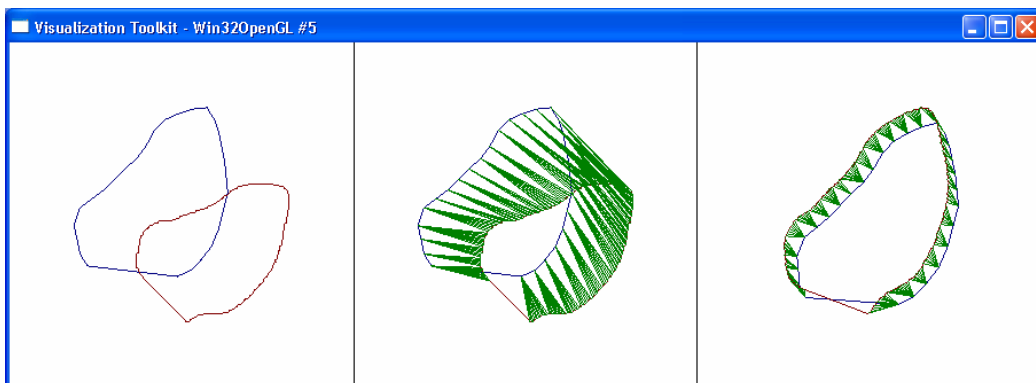


Figura 6.47: Emparelhamento do tipo *um-para-vários* dos contornos “heart1” e “heart9”, com base na minimização da distância euclidiana entre os pontos a emparelhar.

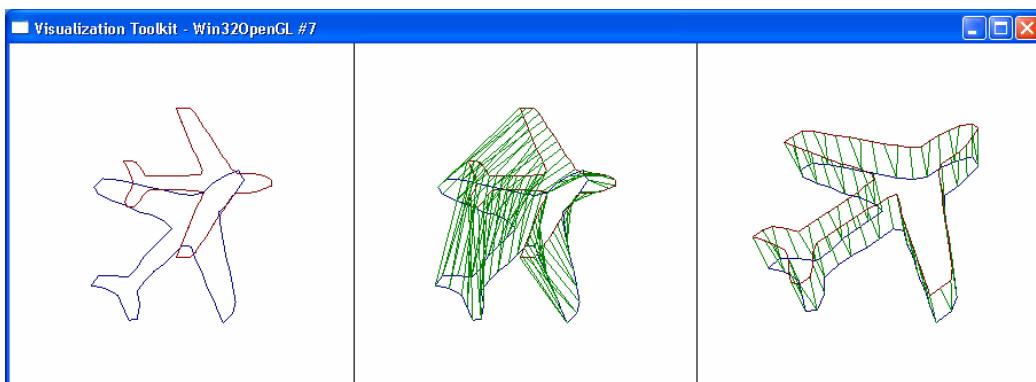


Figura 6.48: Emparelhamento do tipo *um-para-vários* dos contornos “airplane2” e “airplane12”, com base na minimização da distância euclidiana entre os pontos a emparelhar.

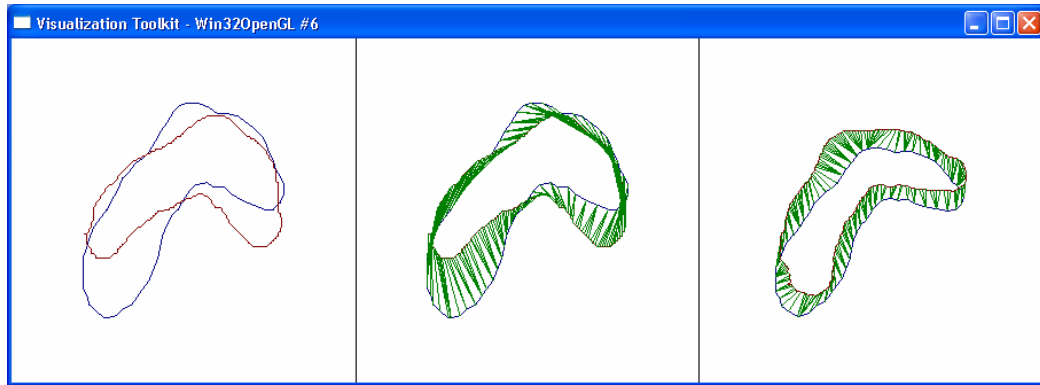


Figura 6.49: Emparelhamento do tipo *um-para-vários* dos contornos “foot7” e “foot13”, com base na minimização da distância euclidiana entre os pontos a emparelhar.

6.6.3 Análise de resultados

Pela observação das Figuras 6.42 a 6.49, apresentadas nas duas subsecções anteriores, conclui-se que ambas as metodologias de emparelhamento do tipo *um-para-vários* parecem adequadas. No entanto, comparando os resultados das mesmas, observa-se que a distribuição dos pontos excedentários é mais equilibrada no emparelhamento que considera a minimização da distância entre pontos a emparelhar como critério de emparelhamento.

A metodologia que utiliza a informação de custos guardada na respectiva matriz tem a vantagem de possibilitar a utilização destes custos para comparar a similaridade entre formas. Na Tabela 6.4 representam-se os custos de emparelhamento globais para esta metodologia e respectivos tempos de execução. Comparando esses valores de custos, apenas para pares de contornos definidos por diferente número de pontos, com os apresentados na Tabela 6.3, pode-se concluir o seguinte:

- O emparelhamento do tipo *um-para-um* apresenta custos inferiores ao da metodologia do tipo *um-para-vários* para todos os pares de contornos excepto para um caso, “heartB1” e “heratB2”.

Este facto pode ser explicado facilmente. Nos contornos onde o custo aumentou, os novos emparelhamentos eram, em média, de maior custo do que os obtidos pela metodologia do tipo *um-para-um*. No caso dos contornos “heartB1” e “heartB2”, os novos emparelhamentos apresentavam, em média, um custo inferior.

No emparelhamento do tipo *um-para-vários* com base na minimização da distância entre pontos, deixou de fazer muito sentido utilizar a medida de custos, obtida com base na

respectiva matriz previamente calculada, para quantificar a similaridade dos emparelhamentos obtidos, pois os novos emparelhamentos podem não ser de custo mínimo. No entanto, apesar de não se apresentarem valores neste trabalho, ganhou-se informação sobre a distância entre os pontos emparelhados após alinhamento dos dois contornos. Deste modo, esta grandeza poderá ser também utilizada como medida de similaridade entre contornos.

6.7 Sumário e conclusões

Neste capítulo foram apresentadas três metodologias do tipo *um-para-um* para determinar o emparelhamento entre dois contornos ordenados, sendo que qualquer uma delas pode ser aplicada a contornos definidos por igual ou diferente número de pontos. A primeira metodologia apresentada, secção 6.2, baseia-se em informação de curvatura (ou ângulo de viragem) do polígono definido pelos respectivos pontos. A segunda metodologia, secção 6.3, baseia-se na comparação da distância dos pontos que definem os contornos aos respectivos centróides. Na secção 6.4 foi apresentada uma metodologia de emparelhamento de contornos baseada simultaneamente em informação de curvatura e em informação da distância dos pontos ao respectivo centróide.

Em qualquer uma das três metodologias apresentadas para determinar o melhor emparelhamento global respeitando a ordem dos pontos emparelhados, foi utilizado o algoritmo de programação dinâmica apresentado nesta Dissertação, secção 5.4. Em relação à qualidade dos emparelhamentos, a metodologia baseada apenas em informação de curvatura apresentou alguma instabilidade, especialmente no emparelhamento de contornos definidos por elevado número de pontos. As duas metodologias que utilizam informação da distância dos pontos ao respectivo centróide apresentaram emparelhamentos adequados nos diversos contornos testados; no entanto, das duas, a metodologia que também utiliza também informação de curvatura apresentou, em alguns exemplos, melhores emparelhamentos ao nível local. Foram realizados, também, alguns ensaios para aferir da robustez, destas três metodologias referidas, a situações de oclusão de parte de um contorno. Os resultados obtidos foram satisfatórios, sendo que a metodologia que se baseia apenas em informação de curvatura revelou melhores resultados.

Na secção 6.5 foi apresentado e testado um algoritmo de emparelhamento do tipo *um-para-um* com ajuste local, o qual pode ser aplicada a contornos definidos por diferente

número de pontos. Este algoritmo parte de um emparelhamento prévio, também do tipo *um-para-um*, fazendo reajustes nos emparelhamentos no sentido de promover o emparelhamento de pontos próximos em termos de distância euclidiana, após alinhamento dos dois contornos.

O capítulo continuou com a apresentação de exemplos de emparelhamentos do tipo *um-para-vários*, utilizando os dois algoritmos definidos na secção 5.6. O primeiro algoritmo emparelha os pontos excedentários (não correspondidos na metodologia do tipo *um-para-um* utilizada) com os que lhe estão mais próximos em termos de matriz de custos, determinada aquando do emparelhamento inicial do tipo *um-para-um*. O segundo algoritmo emparelha os pontos excedentários com os que lhe estão mais próximos em termos de distância euclidiana, após alinhamento dos dois contornos. Os resultados apresentados por ambos os algoritmos foram bons, especialmente quando a metodologia de emparelhamento do tipo *um-para-um* inicialmente utilizada foi a baseada em informação de curvatura e comparação da distância dos pontos ao respectivo centróide.

CAPÍTULO VII:
CONCLUSÕES FINAIS E PERSPECTIVAS DE
TRABALHO FUTURO

7.1 Conclusões finais

7.1.1 Conclusões gerais

O tema central desta Dissertação insere-se no domínio da Visão Computacional, mais especificamente no reconhecimento/emparelhamento de objectos representados em imagens. Para os respectivos emparelhamentos foram considerados contornos definidos por dados pontuais, extraídos da representação dos objectos em imagens.

Os principais objectivos inicialmente traçados para este trabalho foram os seguintes:

1. Determinação de correspondências entre conjuntos de dados pontuais extraídos de contornos de objectos representados em imagens, usando modelação física ou geométrica. Os conjuntos de dados pontuais extraídos das imagens a emparelhar poderão ser constituídos por igual ou diferente número de pontos.
2. Utilização de uma medida capaz de quantificar a afinidade entre os dados pontuais extraídos dos objectos a emparelhar.
3. Utilização de técnicas de optimização das correspondências com inclusão de restrições, tais como: dados vizinhos deverão manter-se vizinhos, não deverão ser permitidos emparelhamentos cruzados, deve ser respeitada a coerência do movimento ao longo do tempo, etc.

Para concretizar os objectivos propostos, foram desenvolvidas as seguintes tarefas fundamentais:

- a) Em primeiro lugar foi feito um estudo bibliográfico sobre metodologias de emparelhamento de objectos em Visão Computacional, bem como das suas aplicações. Assim, foram analisadas várias metodologias capazes de traduzir características dos objectos a emparelhar em coeficientes de um espaço funcional, dados pontuais, eixos médios ou outros que poderão ser posteriormente utilizados para determinar a correspondência entre os objectos representados em imagens.
- b) Seguidamente, foi efectuado um estudo sobre técnicas de optimização, em especial sobre as relacionadas com problemas de afectação, usando programação linear, programação inteira, grafos bipartidos, programação dinâmica e *simulated annealing*.
- c) Atendendo que neste trabalho se iria trabalhar com o emparelhamento de contornos definidos por dados pontuais, foi também estudado o problema da

ordenação dos pontos que definem um contorno, no sentido destes formarem uma sequência ordenada coerente.

- d) O passo seguinte foi o desenvolvimento de metodologias de emparelhamento de dados pontuais representativos de contornos em imagens usando técnicas de optimização. Também foi desenvolvida uma metodologia de ordenação dos dados pontuais que definem os contornos a emparelhar.
- e) Com base nos emparelhamentos obtidos pelas metodologias de emparelhamento desenvolvidas, foram seguidamente desenvolvidas metodologias de determinação da transformação rígida que melhor alinha os dois contornos considerados.
- f) Posteriormente, foi realizada a implementação em ambiente *Microsoft Visual C++*, [Rodrigues, 2003], das diversas metodologias desenvolvidas. Para melhor visualizar os resultados dos emparelhamentos, recorreu-se a funções da biblioteca de domínio público *VTK – The Visualization Toolkit*, [Schroeder, 1999].
- g) O estudo da plataforma computacional de processamento e análise de imagem – *CMIS*, [Tavares, 2000a, 2002, 2003], e implementação na mesma do algoritmo de programação dinâmica desenvolvido para emparelhamento de contornos definidos por dados pontuais foram os passos seguintes.
- h) Finalmente, foram realizados os ensaios necessários para validar as metodologias desenvolvidas e analisar e comparar os respectivos resultados. Concluiu-se o trabalho com a escrita desta Dissertação.

Nesta hora de balanço, pode-se dizer que o trabalho desenvolvido permitiu alcançar os objectivos propostos. De entre os vários contributos dados nesta área do emparelhamento de objectos representados por contornos, destacam-se os seguintes:

1. Para a determinação de emparelhamentos de contornos de objectos representados em imagens, foram desenvolvidas três metodologias de carácter geométrico, fáceis de implementar e de reduzido custo computacional. A primeira metodologia é baseada em informação de curvatura do contorno. A segunda é baseada na comparação das distâncias dos pontos do contorno ao seu centróide. A terceira metodologia utiliza em simultâneo informação de curvatura e distância dos pontos ao centróide. Além das referidas três metodologias de emparelhamento do tipo *um-para-um*, foram também desenvolvidas duas metodologias de emparelhamento do tipo *um-para-vários*, uma baseada nos

custos de emparelhamento previamente guardados na respectiva matriz e a outra baseada na minimização da distância euclidiana entre pontos a emparelhar.

2. Em relação à determinação de uma medida capaz de quantificar a similaridade entre os objectos a emparelhar, nas três metodologias referidas no ponto anterior está definida uma medida de afinidade ou de custos de emparelhamento.
3. Finalmente, em relação ao terceiro objectivo enunciado, talvez o mais importante, este foi atingido plenamente. Para determinar emparelhamentos globais garantindo que dados pontuais vizinhos se mantenham vizinhos e não hajam emparelhamentos cruzados, foi desenvolvido e implementado um algoritmo baseado na técnica de optimização de programação dinâmica que resolve este problema. No entanto, este algoritmo parte do pressuposto que os contornos são definidos por conjuntos de pontos ordenados. Assim, para garantir esta exigência do algoritmo de optimização, foi também implementado um algoritmo de ordenação dos pontos baseado na técnica de optimização de *simulated annealing*.
4. Implementação do algoritmo de optimização baseado em programação dinâmica já referido na plataforma computacional de desenvolvimento e ensaio *CMIS*.

Além das realizações já referidas, foram também desenvolvidos outros trabalhos de menor importância, nomeadamente:

5. Desenvolvimento e implementação de uma metodologia específica para determinação da transformação rígida que melhor alinha dois contornos definidos por conjuntos de pontos ordenados. Para determinar a rotação envolvida é necessário um emparelhamento prévio.
6. Apresentação, ao nível da formulação, de outras metodologias para ordenar pontos de contornos, uma baseada na técnica *branch-and-bound* e outra em programação inteira.
7. Desenvolvimento, ao nível da formulação, de uma metodologia de programação inteira para determinar o emparelhamento óptimo entre dois contornos com base numa matriz de custos previamente determinada, respeitando a ordem dos pontos emparelhados.

7.1.2 Conclusões relativas aos algoritmos desenvolvidos

Em relação ao desempenho dos algoritmos desenvolvidos, vamos seguidamente expor as

principais conclusões, seguindo a ordem pela qual foram apresentados ao longo dos diversos capítulos desta Dissertação.

a) Algoritmo de ordenação dos pontos que definem um contorno baseada em *simulated annealing*

Este algoritmo, após adaptação para a resolução do problema em causa, demonstrou-se muito eficiente. Em primeiro lugar, porque nas mais de 1800 experiências efectuadas com diversos contornos alcançou uma taxa de sucesso de 100% na obtenção de uma ordem correcta para os pontos dos contornos. Em segundo lugar, porque as ordenações dos contornos foram sempre obtidas em tempos reduzidos. A título de exemplo, para ordenar contornos com 25, 62 e 135 pontos, o tempo dispendido, em segundos, foi aproximadamente 0,01, 0,07 e 0,37, num PC equipado com um processador *Intel Pentium III* a 1.0 GHz, com 256MB de RAM.

b) Algoritmo de determinação da transformação rígida que melhor alinha dois contornos

Não foi efectuada uma análise exaustiva em relação à qualidade dos resultados apresentados pela metodologia de determinação da transformação rígida existente entre dois contornos. Esta situação deve-se ao facto de este algoritmo ter sido desenvolvido especialmente para facilitar a visualização dos emparelhamentos obtidos pelos métodos de emparelhamento desenvolvidos ao longo deste trabalho. Mesmo assim, pode-se concluir pela observação dos resultados apresentados no quarto capítulo e pelas diversas figuras apresentadas ao longo do sexto capítulo, que os resultados apresentados foram adequados, pois corresponderam ao que era esperado pela observação dos respectivos contornos.

c) Algoritmo de optimização baseado em programação dinâmica

Obviamente, o algoritmo de programação dinâmica com restrição de ordem (APDCRO) foi testado antes de ser implementado na plataforma computacional *CMIS*. No entanto, a sua implementação nesta plataforma foi crucial para comparar o seu desempenho com os algoritmos de afectação sem restrição de ordem (AASRO) clássicos: método *Húngaro*, *Simplex* para problemas de fluxo e *LAPm*. Antes das conclusões, é importante referir que os testes foram efectuados com vários pares de contornos definidos por igual ou diferente número de pontos, mas sempre em igualdade de circunstâncias entre os diversos algoritmos de optimização.

Uma conclusão fundamental a retirar é a adequação do APDCRO para optimização dos emparelhamentos de contornos de objectos representados em imagens. O conjunto – metodologia de determinação dos emparelhamentos entre dois contornos baseada na modelação geométrica de Shapiro, [Shapiro, 1992a, 1992b], e algoritmo de programação dinâmica – criou uma metodologia global mais robusta do que as já existentes na plataforma computacional *CMIS*. Basta observar que nos exemplos apresentados no quinto capítulo, para a configuração base definida por defeito para determinar a matriz de afinidade pelo método de Shapiro, esta nova metodologia global alcançou sempre óptimos emparelhamentos, enquanto que as metodologias de optimização globais já implementadas na plataforma determinaram maus emparelhamentos em alguns pares de contornos.

Embora não tenha sido testado, é espectável que as outras metodologias de determinação de uma matriz de afinidade (modelação baseada em princípios físicos, por exemplo), também já existente na referida plataforma computacional, em conjunto com o algoritmo de optimização baseado em programação dinâmica originem uma metodologia global muito mais robusta do que as metodologias globais já implementadas na plataforma.

Passemos agora a especificar as diferenças entre os dois tipos de algoritmos de optimização das correspondências – AASRO e APDCRO – quanto à qualidade dos emparelhamentos obtidos e velocidade de execução:

- Sempre que os AASRO alcançavam um bom emparelhamento sem correspondências cruzadas, o APDCRO alcançava o mesmo emparelhamento; portanto, o custo global dos emparelhamentos era exactamente o mesmo.
- Quando os AASRO alcançavam um emparelhamento com alguns cruzamentos, o APDCRO alcançava um emparelhamento idêntico mas sem cruzamentos. Obviamente o custo associado teria que ser superior, pois a restrição da ordem obrigou a que emparelhamentos de menor custo fossem substituídos por emparelhamentos de maior custo mas mais coerentes.
- Em algumas situações em que os AASRO alcançaram um emparelhamento com pouco sentido ou mesmo sem sentido, o APDCRO alcançou um emparelhamento coerente. Obviamente, nesta situação os custos apresentados por ambos os tipos de algoritmos são muito diferentes. Refira-se que estes maus emparelhamentos, em particular, poderiam desaparecer com a alteração de alguns parâmetros definidos por defeito na plataforma para a determinação da matriz de afinidade. No entanto, outros maus emparelhamentos poderiam surgir em outros contornos.

- A velocidade de execução do APDCRO foi sempre muito superior à dos AASRO.

d) Algoritmo de emparelhamento baseado em informação de curvatura usando programação dinâmica

Nesta metodologia, em primeiro lugar é determinada uma matriz de custos angulares com base em informação de curvatura. Depois, é utilizado o algoritmo de optimização de programação dinâmica para determinar os emparelhamentos. Esta metodologia global revelou-se robusta para alguns pares de contornos e algo instável para outros. Passemos a especificar:

- Para contornos definidos por poucos pontos e pequena diferença entre o número de pontos que define cada um deles, os emparelhamentos obtidos foram óptimos.
- Para contornos definidos por um elevado número de pontos ou onde havia uma grande diferença entre o número de pontos que define cada um dos contornos, verificou-se grande instabilidade. Em alguns pares de contornos os emparelhamentos eram bons, enquanto que para outros pares em condições ligeiramente diferentes já apareciam emparelhamentos sem sentido.
- A medida de similaridade entre contornos, baseada nos custos de emparelhamento médios, revelou-se algo instável.
- Apresentou elevada velocidade de execução.
- Verificou-se alguma robustez ao problema de oclusão de partes de um contorno.

A instabilidade verificada nos emparelhamentos de alguns pares de contornos pode ser atribuída aos seguintes factos:

- Quando há uma diferença acentuada entre o número de pontos que define cada um dos contornos, em média, o contorno definido por maior número de pontos tem tendência a apresentar menor diferença entre os valores dos ângulos de curvatura do que o contorno definido por menor número de pontos. Deste modo, há uma elevada suavização do polígono associado ao contorno definido por maior número de pontos. Assim, forçosamente se tem uma diferença considerável entre as amplitudes das sequências de ângulos definidas por cada contorno. Este mesmo efeito de suavização justifica ainda a instabilidade dos emparelhamentos de contornos definidos por elevado número de pontos.
- Os contornos definidos por elevado número de pontos que foram utilizados, em geral, apresentavam muito ruído, isto é, pequenas perturbações no alinhamento

dos pontos, com pouco significado global mas influentes ao nível local.

e) Algoritmo de emparelhamento baseado na distância ao centróide usando programação dinâmica

Nesta metodologia, em primeiro lugar é determinada uma matriz da diferença das distâncias dos pontos que definem os contornos aos respectivos centróides. Depois, é utilizado o algoritmo de optimização de programação dinâmica para determinar os emparelhamentos.

Esta metodologia global revelou-se robusta para todos os pares de contornos utilizados. Mais especificamente, conclui-se o seguinte:

- Esta metodologia alcançou sempre bons emparelhamentos para contornos definidos por igual ou diferente número de pontos, quer fossem definidos por um elevado ou reduzido número de pontos.
- Em alguns pares de contornos surgiram emparelhamentos ao nível local que poderiam ser melhorados.
- A medida de similaridade entre os contornos, baseada nos custos globais médios de emparelhamento, apresenta sentido.
- Verificou-se alguma robustez ao ruído no caso dos contornos definidos por elevado número de pontos que foram estudados.
- Apresenta elevada velocidade de execução.

f) Algoritmo de emparelhamento baseado em informação de curvatura e distâncias ao centróide usando programação dinâmica

Esta metodologia utiliza em simultâneo informação de curvatura e informação da distância dos pontos ao centróide para construir a matriz de custos. Depois, utiliza o algoritmo de programação dinâmica para determinar um emparelhamento global óptimo.

A referida metodologia obteve sempre bons emparelhamentos nos diversos pares de contornos testados. Portanto, revelou-se bastante robusta. Como conclusões mais específicas, temos:

- Obtenção de bons emparelhamentos globais e locais em todos os pares de contornos testados.
- A medida de similaridade entre os contornos, baseada nos custos globais médios de emparelhamento, tem sentido.

- Alguma robustez ao ruído no caso dos contornos definidos por elevado número de pontos que foram estudados.
- Apresenta elevada velocidade de execução.

Como última conclusão, pode-se ainda referir que esta metodologia aproveita a comparação das distâncias ao centróide para dar estabilidade e a informação de curvatura para realizar bons emparelhamentos locais.

g) Algoritmo de emparelhamento com ajuste local

Este algoritmo, que tem interesse para ser utilizado no emparelhamento de contornos definidos por diferente número de pontos, revelou-se muito eficiente na correcção de maus emparelhamentos locais. De facto, conseguiu realizar melhorias nos emparelhamentos obtidos, especialmente quando há uma grande diferença entre a dimensão dos conjuntos de pontos que definem os contornos. No entanto, apresenta uma fraqueza: para funcionar correctamente, é necessário que parta inicialmente de um emparelhamento prévio com uma qualidade aceitável, de modo a ser possível determinar razoavelmente o ângulo de rotação que melhor alinha os dois contornos.

Com o ajuste local dos emparelhamentos, o custo global baseado na matriz de custos perde algum interesse, pois os novos emparelhamentos podem não corresponder àqueles que foram utilizados para determinar esse valor; no entanto, pode ser utilizada a soma das distâncias entre os pares de pontos correspondidos para determinar a similaridade entre os dois contornos.

h) Algoritmo de emparelhamento do tipo *um-para-vários* com base na matriz de custos

Esta metodologia pode ser aplicada para emparelhar contornos definidos por diferente número de pontos, após determinação de um emparelhamento global do tipo *um-para-um*. Os pontos não correspondidos pela metodologia do tipo *um-para-um* passam a ser correspondidos com pontos do outro contorno, respeitando a ordem e minimizando os custos previamente determinados. Pela observação das diversas figuras apresentadas pode-se concluir que:

- Obtiveram-se bons emparelhamentos quando a qualidade dos emparelhamentos iniciais do tipo *um-para-um* era boa.
- Para contornos definidos por conjuntos de pontos de dimensões muito diferentes, os emparelhamentos por vezes não correspondiam ao esperado pelo observador.

- Verificou-se que a medida de similaridade entre contornos continuou a fazer sentido, sempre que já fazia sentido no emparelhamento do tipo *um-para-um*.
- Apresentou elevada velocidade de execução.

i) Algoritmo de emparelhamento do tipo *um-para-vários* com base na minimização da distância entre pontos emparelhados

Esta metodologia pode ser aplicada para emparelhar contornos definidos por diferente número de pontos, após determinação de um emparelhamento global do tipo *um-para-um*. Os pontos não correspondidos pela metodologia do tipo *um-para-um* passam a ser correspondidos com pontos do outro contorno, respeitando a ordem e minimizando a distância aos seus pares, após alinhamento dos dois contornos. Pela observação das diversas figuras apresentadas pode-se concluir que:

- Obtiveram-se bons emparelhamentos quando a qualidade dos emparelhamentos iniciais do tipo *um-para-um* era boa, correspondendo ao esperado pelo observador.
- A medida de similaridade entre contornos baseada na matriz de custos dos emparelhamentos globais do tipo *um-para-vários* deixa de fazer sentido, no entanto surge uma nova medida de similaridade baseada na distância entre os pontos emparelhados.
- Apresentou elevada velocidade de execução.

7.2 Perspectivas de trabalho futuro

Obviamente, há ainda muito trabalho a desenvolver no domínio do reconhecimento de objectos representados em imagens; no entanto, vai-se referir apenas aqueles que estão mais relacionados com o trabalho desenvolvido ao longo desta Dissertação.

O algoritmo de determinação da correspondência óptima baseado em programação dinâmica parte do pressuposto que todos os pontos do contorno definido por menor número de pontos são relevantes para o emparelhamento global. O cumprimento deste pressuposto provoca pelo menos as seguintes fraquezas neste algoritmo e em todos os algoritmos que o cumprem:

- Incapacidade para lidar correctamente com o emparelhamento de contornos quando há oclusões de partes de um contorno que não se verifica no outro contorno.

- Sensibilidade a elementos *outsiders* quando estes são pontos do contorno definido por menor número de pontos.

Assim, propõe-se o seguinte:

- Desenvolvimento de um algoritmo de optimização com o qual seja possível controlar a percentagem de pontos a utilizar do contorno definido pelo menor número de pontos. Contudo, o objectivo não será desenvolver um algoritmo que exclua os emparelhamentos que provocam maior custo, após determinação do emparelhamento global, pois nesta situação o emparelhamento global já foi influenciado por esses pontos. O objectivo é que os pontos que menos se ajustem ao emparelhamento global não sejam considerados para alcançar esse mesmo emparelhamento.

A utilização da metodologia de emparelhamento que utiliza a informação de curvatura desenvolvida ao longo deste trabalho demonstrou ter alguma capacidade de lidar com o problema da oclusão de partes de um contorno. Embora não testado, o seu princípio de funcionamento também garante alguma robustez a movimentos articulados, pois num movimento deste tipo, apenas uma parte do objecto sofre deformação, mantendo-se o restante do objecto com a mesma forma e, por consequência, os mesmos ângulos de curvatura. No entanto, como já referido, o emparelhamento baseado em informação de curvatura apresentado é instável no emparelhamento de contornos definidos por um elevado número de pontos ou quando há uma diferença acentuada no número de pontos que define cada um dos contornos. Assim, propõe-se o seguinte:

- Desenvolvimento de um filtro capaz de extrair apenas os pontos essenciais de um contorno, eliminando deste modo o ruído, mas garantido que a informação relevante não seja perdida.
- Testar a metodologia baseada em informação de curvatura após aplicação do filtro proposto em conjunto com o novo algoritmo de optimização também aqui proposto. Os testes deverão incluir pares de contornos onde seja possível testar a robustez a problemas de oclusão de parte de um dos contornos e onde hajam deformações causadas por movimentos articulados.

Outra proposta de trabalho que pode ser interessante é o teste das metodologias desenvolvidas ao longo desta Dissertação no reconhecimento de objectos, recorrendo às

grandes bases de imagens, cada vez mais disponíveis em domínio público.

BIBLIOGRAFIA

[**Arkin, 1991**] – Esther Arkin, L. Chew, D. Huttenlocher, K. Kedem, J. Mitchell
AN EFFICIENTLY COMPUTABLE METRIC FOR COMPARING POLYGONAL SHAPES
IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, v. 13, n. 3, p.
209-216, MARCH – 1991

[**Bala, 2004**] – Erdem Bala, A. Enis Cetin
**COMPUTATIONALLY EFFICIENT WAVELET AFFINE INVARIANT FUNCTIONS FOR SHAPE
RECOGNITION**
IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE , v. 26, n. 8, p.
1095-1099, AUGUST – 2004

[**Bastos, 2003**] – Luísa Bastos
**TESE DE MESTRADO: OPTIMIZAÇÃO DA DETERMINAÇÃO DAS CORRESPONDÊNCIAS ENTRE
OBJECTOS DEFORMÁVEIS NO ESPAÇO MODAL**
FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO, PORTUGAL, 2003

[**Bastos, 2006**] – Luísa Bastos, João Tavares
MATCHING OF OBJECTS NODAL POINTS IMPROVEMENT USING OPTIMIZATION
INVERSE PROBLEMS IN SCIENCE AND ENGINEERING, v. 14, n. 5, p. 529-541, 2006

[**Belongie, 2002**] – S. Belongie, J. Malik, J. Puzicha
SHAPE MATCHING AND OBJECT RECOGNITION USING SHAPE CONTEXTS
IEEE TRANSACTION ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, v. 24, n. 4, p.
509-522, APRIL – 2002

[**Blum, 1967**] – H. Blum
A TRANSFORMATION FOR EXTRACTING NEW DESCRIPTORS OF SHAPE
IN W. WATHEN-DUNN, EDITOR, MODELS FOR THE PERCEPTION OF SPEECH AND VISUAL
FORM, MIT PRESS, CAMBRIDGE MA, 1967

[**Carcassoni, 2003**] – Marco Carcassoni, Edwin Hancock
CORRESPONDENCE MATCHING WITH MODAL CLUSTERS
IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, v. 25, n. 12, p.
1609-1615, DECEMBER – 2003

[Chung, 2000] – Jae-Moon Chung, Noboru Ohnishi

MATCHING AND RECOGNITION OF PLANAR SHAPES USING MEDIAL AXIS PROPERTIES

BIO-MIMETIC CONTROL RESEARCH CENTER, THE INSTITUTE OF PHYSICAL AND CHEMICAL RESEARCH (RIKEN), 2000

[Daugman, 2003] – John Daugman

THE IMPORTANCE OF BEING RANDOM: STATISTICAL PRINCIPLES OF IRIS RECOGNITION

PATTERN RECOGNITION, v. 36, p. 279-291, 2003

[Dell' Amico, 2000] – M. Dell' Amico, P. Toth

ALGORITHMS AND CODES FOR DENSE ASSIGNMENT PROBLEMS: THE STATE OF THE ART

DISCRETE APPLIED MATHEMATICS, v. 100, p. 274-278, 2000

[Dorigo, 1997] – Marco Dorigo, Luca Gambardella

ANT COLONIES FOR THE TRAVELING SALESMAN PROBLEM

BIOSYSTEMS, n. 43, p. 73-81, 1997

[Fernandes, 1998] – Edite Fernandes

COMPUTAÇÃO NUMÉRICA, 2ª EDIÇÃO

UNIVERSIDADE DO MINHO, BRAGA, PORTUGAL, p. 301-315, 1998

[Held, 1962] – Michael Held, Richard Karp

A DYNAMIC PROGRAMMING APPROACH TO SEQUENCING PROBLEMS

JOURNAL OF THE SOCIETY FOR INDUSTRIAL AND APPLIED MATHEMATICS, v. 10, n. 1, p. 196-210, 1962

[Hillier, 1995] – F. Hillier, G. Lieberman

INTRODUCTION TO OPERATIONS RESEARCH

MCGRAW-HILL INTERNATIONAL EDITIONS, 1995

[Johnson, 1997] – D. Johnson, L. McGeoch.

THE TRAVELLING SALESMAN PROBLEM: A CASE STUDY IN LOCAL OPTIMIZATION

IN E. H. L. AARTS AND J. K. LENSTRA, EDITORS, LOCAL SEARCH IN COMBINATORIAL OPTIMIZATION, p. 215-310, JOHN WILEY & SONS, CHICHESTER, UK, 1997

[**Johnson, 2002**] – D. S. Johnson, L. A. McGeoch

EXPERIMENTAL ANALYSIS OF HEURISTICS FOR THE STSP

IN G. GUTIN AND A. PUNNEN, EDITORS, THE TRAVELING SALESMAN PROBLEM AND ITS VARIATIONS, P. 369-443, KLUWER ACADEMIC PUBLISHERS, 2002

[**Keyzers, 2007**] – Daniel Keyzers, Thomas Deselaers, Thomas Breuel

OPTIMAL GEOMETRIC MATCHING FOR PATCH-BASED OBJECT DETECTION

ELECTRONIC LETTERS ON COMPUTER VISION AND IMAGE ANALYSIS 6(1), P. 44-54, 2007

[**Lades, 1993**] – M. Lades

DISTORTION INVARIANT OBJECT RECOGNITION IN THE DYNAMIC LINK ARCHITECTURE

IEEE TRANSACTION COMPUTERS, V. 42, N. 3, P. 300-311, 1993

[**Lin, 1965**] – S. Lin

COMPUTER SOLUTIONS OF THE TRAVELING SALESMAN PROBLEM

BELL SYSTEM TECHNICAL JOURNAL, V. 44, P. 2245-2269, 1965

[**Löbel, 2000**] – A. Löbel

MFC – A NETWORK SIMPLEX IMPLEMENTATION

<http://www.zib.de/optimization/software/mfc>, 2000

[**Maciel, 2001**] – João Maciel

TESE DE DOUTORAMENTO: GLOBAL MATCHING: OPTIMAL SOLUTION TO CORRESPONDENCE PROBLEMS

INSTITUTO SUPERIOR TÉCNICO, PORTUGAL, 2001

[**Manay, 2006**] – S. Manay, D. Cremers, Byung-Woo Hong, A. Yezzi Jr., S. Soatto

INTEGRAL INVARIANTS FOR SHAPE MATCHING

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, V. 28, N. 10, OCTOBER – 2006

[**Norman, 1975**] – John Norman

ELEMENTARY DYNAMIC PROGRAMMING

EDWARD ARNOLD (PUBLISHERS), LONDON, 1975

[Ogniewicz, 1995] – R. Ogniewicz, O. Kübler

HIERARCHIC VORONOI SKELETONS

PATTERN RECOGNITION, v. 28, n. 3, p. 343-359, 1995

[Press, 2002] – W. Press, S. Teukolsky, W. Vetterling, B. Flannery

NUMERICAL RECIPES IN C: THE ART OF SCIENTIFIC COMPUTING, 2ND EDITION

CAMBRIDGE UNIVERSITY PRESS, USA, 2002

[Rodrigues, 2003] – Pimenta Rodrigues, Pedro Pereira, Manuela Sousa

PROGRAMAÇÃO EM C++ – CONCEITOS BÁSICOS E ALGORITMOS, 6.^a EDIÇÃO

FCA – EDITORA DE INFORMÁTICA, LISBOA, 2003

[Rosenhahn, 2006] – Bodo Rosenhahn, Thomas Brox, Daniel Cremers, Hans-Peter Seidel

A COMPARISON OF SHAPE MATCHING METHODS FOR CONTOUR BASED POSE ESTIMATION

IN “COMBINATORIAL IMAGE ANALYSIS, SPRINGER LNCS 4040”, R. REULKE ET AL. (EDS.), P.

263-276, BERLIN, GERMANY, JUNE – 2006

[Rosenkrantz, 1977] – D. Rosenkrantz, R. Stearns, P. Lewis

AN ANALYSIS OF SEVERAL HEURISTICS FOR THE TRAVELLING SALESMAN PROBLEM

SIAM JOURNAL ON COMPUTING, n. 6, p. 563-581, 1977

[Sebastian, 2004] – Thomas B. Sebastian, Philip N. Klein, Benjamin B. Kimia

RECOGNITION OF SHAPE BY EDITING THEIR SHOCK GRAPHS

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, v. 26, n. 5,

MAY – 2004

[Scharstein, 1994] – Daniel Scharstein

MATCHING IMAGES BY COMPARING THEIR GRADIENT FIELDS

PATTERN RECOGNITION, v. 1, p. 572-575, 1994

[Schroeder, 1999]– Will Schroeder, Ken Martin

THE VTK USER’S GUIDE

KITWARE INC., JUNE – 1999

[**Sciaroff, 1995**] – S. Sciaroff, A. Pentland

MODAL MATCHING FOR CORRESPONDENCE AND RECOGNITION

IEEE TRANSACTIONS PATTERN ANALYSIS AND MACHINE INTELLIGENCE, v. 17, n. 6, p. 545-561, JUNE – 1995

[**Scott, 2006**] – C. Scott, R. Nowak

ROBUST CONTOUR MATCHING VIA THE ORDER-PRESERVING ASSIGNMENT PROBLEM

IEEE TRANSACTIONS ON IMAGE PROCESSING, v. 15, n. 7, p. 1831-1838, JULY – 2006

[**Shapiro, 1992a**] – L. Shapiro, J. M. Brady

A MODAL APPROACH TO FEATURE-BASED CORRESPONDENCE

ROBOTICS RESEARCH GROUP, DEPARTMENT OF ENGINEERING SCIENCE, OXFORD UNIVERSITY, 1992

[**Shapiro, 1992b**] – L. Shapiro, J. M. Brady

FEATURE-BASED CORRESPONDENCE: AN EIGENVECTOR APPROACH

IMAGE AND VISION COMPUTING, v. 10, n. 5, p. 283-288, 1992

[**Soares, 1997**] – M. J. Soares

ONDULETAS E PROCESSAMENTO DE SINAL

"CONFERÊNCIA NACIONAL DE TELECOMUNICAÇÕES": ACTAS. AVEIRO: FUNDAÇÃO JOÃO JACINTO MAGALHÃES, p. 395-400, 1997

[**Starink, 1995**] – J. Pascual Starink, Eric Backer

FINDING POINT CORRESPONDENCES USING SIMULATED ANNEALING

PATTERN RECOGNITION, v. 28, n. 2, p. 231-240, 1995

[**Tavares, 2000a**] – João Tavares

TESE DE DOUTORAMENTO: ANÁLISE DE MOVIMENTO DE CORPOS DEFORMÁVEIS USANDO VISÃO COMPUTACIONAL

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO, PORTO, PORTUGAL, 2000

[Tavares, 2000b] – João Tavares, Jorge Barbosa, A. Jorge Padilha

A MATCHING IMAGE OBJECTS IN DYNAMIC PEDOBAROGRAPHY

RECPAD 2000 – 11TH PORTUGUESE CONFERENCE ON PATTERN RECOGNITION, PORTO, PORTUGAL, 2000

[Tavares, 2002] – João Tavares, Jorge Barbosa, A. Jorge Padilha

APRESENTAÇÃO DE UM BANCO DE DESENVOLVIMENTO E ENSAIO PARA OBJECTOS DEFORMÁVEIS

RESI – REVISTA ELECTRÓNICA DE SISTEMAS DE INFORMAÇÃO, EDIÇÃO 1, v. 1, n. 1, 2002

[Tavares, 2003] – João Tavares

RELATÓRIO INTERNO: INTRODUÇÃO À PLATAFORMA DE PROCESSAMENTO E ANÁLISE DE IMAGEM E COMPUTAÇÃO GRÁFICA – CMIS

INEB – INSTITUTO DE ENGENHARIA BIOMÉDICA, LABORATÓRIO SINAL E IMAGEM, INEGI – INSTITUTO DE ENGENHARIA E GESTÃO INDUSTRIAL, LABORATÓRIO DE ÓPTICA E MECÂNICA EXPERIMENTAL, UNIVERSIDADE DO PORTO, FACULDADE DE ENGENHARIA, DEPARTAMENTO DE ENGENHARIA MECÂNICA E GESTÃO INDUSTRIAL, 2003

[Tavares, 2005] – Paulo Tavares

TESE DE MESTRADO: MATRIZES TOTALMENTE E QUASE TOTALMENTE UNIMODULARES

FACULDADE DE CIÊNCIAS E TECNOLOGIA DA UNIVERSIDADE DE COIMBRA, PORTUGAL, 2005

[Veltkamp, 2000] – R. Veltkamp, M. Hagedoorn

STATE-OF-THE-ART IN SHAPE MATCHING

PRINCIPLES OF VISUAL INFORMATION RETRIEVAL, p. 87-119, SPRINGER-VERLAG, LONDON, UK, 2000

[Vetter, 1997] – T. Vetter, M. J. Jones, T. Poggio

A BOOTSTRAPPING ALGORITHM FOR LEARNING LINEAR MODELS OF OBJECTS CLASSES

IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR'97), p. 40-46, 1997

[Volgenant, 1996] – A. Volgenant

LINEAR AND SEMI-ASSIGNMENT PROBLEMS: A CORE ORIENTED APPROACH

COMPUTERS AND OPERATIONS RESEARCH, v. 23, n. 10, 1996

[Winston, 1994] – Wayne L. Winston

OPERATIONS RESEARCH: APPLICATIONS AND ALGORITHMS, 3RD EDITION

DUXBURY PRESS, USA, 1994

[Yuille, 1991] – A. L. Yuille

DEFORMABLE TEMPLATES FOR FACE RECOGNITION

J. COGNITIVE NEUROSCIENCE, v. 3, n. 1, p. 59-71, 1991

[Zhang, 2004] – D. Zhang, Guojun Lu

REVIEW OF SHAPE REPRESENTATION AND DESCRIPTION TECHNIQUES

PATTERN RECOGNITION, v. 37, p. 1-19, 2004

[Zhao, 2003] – W. Zhao, R. Chellappa, P. J. Phillips, A. Rosenfeld

FACE RECOGNITION: A LITERATURE SURVEY

ACM COMPUTING SURVEYS (CSUR), v. 35, p. 399-458, USA, DECEMBER – 2003

[Zhu, 1996] – Song Chun Zhu, A. L. Yuille

FORMS: FLEXIBLE OBJECT RECOGNITION AND MODELING SYSTEM

INT’L JOURNAL OF COMPUTER VISION, v. 20, n. 3, DECEMBER – 1996