

SISTEMA INTEGRADO DE DETECÇÃO E SEGUIMENTO DE CONTORNOS

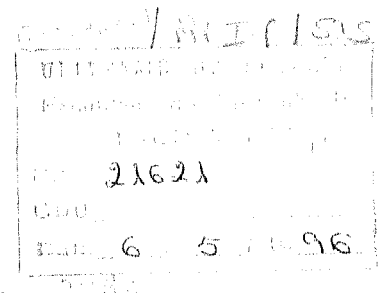
por

LUÍS PAULO GONÇALVES DOS REIS

Dissertação submetida ao Departamento de Engenharia Electrotécnica e de
Computadores para satisfação parcial dos requisitos para a obtenção do grau de

Mestre em Engenharia Electrotécnica e de Computadores

Perfil de Informática Industrial.



Orientador:

Joaquim Pontes Marques de Sá

Professor Associado do Departamento de Engenharia Electrotécnica e de Computadores da
Faculdade de Engenharia da Universidade do Porto

Co-orientador:

António Miguel Pontes Pimenta Monteiro

Professor Auxiliar do Departamento de Engenharia Electrotécnica e de Computadores da
Faculdade de Engenharia da Universidade do Porto

À MEMÓRIA DO MEU PAI

AGRADECIMENTOS

Gostaria de começar por agradecer ao meu orientador, o Prof. Joaquim Marques de Sá e co-orientador Prof. António Pimenta Monteiro, por me terem proposto este trabalho e me terem procurado proporcionar as melhores condições possíveis para a sua execução.

À Nela, pela leitura deste texto, pela concepção de grande parte das suas ilustrações, pelo amor, pela amizade, pelo incentivo, enfim, por tudo...

Aos meus familiares e amigos, nomeadamente à minha Mãe e ao Miguel, pela enorme paciência que tiveram durante esta fase atribulada da minha vida, em que muito pouco tempo tive para lhes dedicar.

Ao Pedro Moreira, companheiro de muitas “noitadas” de trabalho durante este mestrado, pela amizade, pela discussão de ideias, pela leitura deste texto e pelas sugestões fornecidas.

Ao Miguel Velhote e à Ana Paula que leram e criticaram esta tese, pelas suas críticas e conselhos. Ao Miguel gostaria ainda de agradecer a cedência de diversos algoritmos em linguagem C, utilizados neste trabalho.

Ao Instituto de Engenharia Biomédica e à Faculdade de Engenharia da Universidade do Porto, a minha gratidão por me terem disponibilizado espaços e equipamentos sem os quais não poderia ter realizado este trabalho.

A todos os meus colegas de trabalho no INEB, especialmente ao Pedro, Miguel, Jorge e Tavares, pelo excelente ambiente de trabalho proporcionado.

À Junta Nacional de Investigação Científica e Tecnológica pela Bolsa de Mestrado BM/1999, com a qual foi possível realizar este Mestrado.

O meu agradecimento vai também, pelos conhecimentos e incentivos que me transmitiram, para todos os professores da componente escolar deste mestrado, nomeadamente o Prof. Eugénio Oliveira, Prof. Aurélio Campilho, Prof. Raul Vidal, Prof. José Nuno Oliveira, Prof. António Pimenta Monteiro, Prof. Jorge Alves Silva, Prof. J. Borges Gouveia, Prof. Augusto Gomes, Prof. Armando Lage, Prof. José Caldas Pinto, Prof. José Távora e Prof. Rui Loureiro.

Finalmente ao meu Pai, que infelizmente não está entre nós para poder ver a apresentação desta dissertação, mas cuja vida e obra nela se encontram reflectidas. Um agradecimento profundo para alguém que, para além de ter sido meu pai biológico, foi também meu pai científico, ensinando-me a amar a investigação científica e a querer “fazer ciência”. Alguém cuja criatividade, imaginação, simpatia e bondade, um dia, eu gostaria de alcançar.

Sistema Integrado de Detecção e Seguimento de Contornos

por

Luís Paulo Gonçalves dos Reis

RESUMO

Nesta tese são apresentadas metodologias para a detecção e seguimento de contornos, utilizando contornos activos. Inicialmente, apresentam-se alguns conceitos relacionados com a detecção de contornos e arestas, referindo-se os métodos actualmente mais utilizados nestas tarefas. Introduzem-se os métodos baseados em contornos activos, analisando-se a modelização de um contorno, a definição da sua função energética, a solução numérica do método e alguns problemas comuns de implementação.

De forma a poder aplicar, na prática, as técnicas baseadas em contornos activos, é discutida a forma de projecto e implementação de um sistema integrado de detecção e seguimento de contornos. São definidos os requisitos para o sistema, proposta uma possível arquitectura e discutidas diversas questões de implementação. O sistema desenvolvido é descrito em termos da sua interface com o utilizador, operações de processamento de imagem disponibilizadas e sua utilização na detecção e seguimento de contornos. São ainda analisados os métodos e algoritmos de definição, detecção e seguimento de contornos desenvolvidos.

O sistema desenvolvido foi aplicado com sucesso na análise de imagens fictícias e reais, obtendo-se bons resultados tanto na detecção como no seguimento de contornos. Destacam-se os resultados obtidos na detecção e seguimento de contornos endocárdicos em sequências de imagens ecocardiográficas.

ÍNDICE

1. INTRODUÇÃO.....	1
1.1 MOTIVAÇÃO	1
1.2 OBJECTIVOS.....	1
1.3 DEFINIÇÃO E ENQUADRAMENTO	2
1.4 CONTRIBUIÇÕES.....	2
1.5 ORGANIZAÇÃO GLOBAL DA TESE	3
2. CONTORNOS ACTIVOS (“SNAKES”)	5
2.1 TÉCNICAS DE DETECÇÃO DE ARESTAS.....	5
2.1.1 <i>Introdução</i>	5
2.1.2 <i>Operadores Diferenciais e Afins</i>	6
2.1.2.1 <i>Deteção de Arestas Através dos Máximos do Gradiente</i>	7
2.1.2.2 <i>Deteção de Arestas Através dos Cruzamentos por Zero do Laplaciano</i>	9
2.1.3 <i>Critérios de Desempenho</i>	11
2.1.4 <i>Filtros para a Deteção de Arestas</i>	12
2.1.4.1 <i>Laplaciano do Gaussiano</i>	12
2.1.4.2 <i>Canny</i>	13
2.1.4.3 <i>Deriche</i>	14
2.1.4.4 <i>Sarkar e Boyer</i>	14
2.1.4.5 <i>Shen e Castan</i>	15
2.1.5 <i>Operações Complementares</i>	15
2.1.6 <i>Utilização dos Mapas de Arestas na Deteção de Contornos</i>	16
2.2 INTRODUÇÃO AOS CONTORNOS ACTIVOS.....	17
2.3 MODELIZAÇÃO DE UM CONTORNO ACTIVO.....	18
2.3.1 <i>Comportamento Básico de um Contorno Activo</i>	18
2.3.2 <i>Função de Energia a Minimizar</i>	19
2.3.3 <i>Forças Internas do Contorno</i>	19
2.3.3.1 <i>Força Elástica</i>	20
2.3.3.2 <i>Movimentação da “Snake” Sujeita à Força Elástica</i>	21
2.3.3.3 <i>Força de Flexão</i>	23
2.3.3.4 <i>Movimentação da “Snake” Sujeita à Força de Flexão</i>	24
2.3.3.5 <i>Movimentação da “Snake” Sujeita às Forças Elástica e de Flexão</i>	25

2.3.4	<i>Forças Externas</i>	26
2.3.4.1	Força da Intensidade da Imagem.....	26
2.3.4.2	Movimentação da “Snake” Sujeita à Força da Intensidade da Imagem	29
2.3.4.3	Força do Gradiente da Imagem	31
2.3.4.4	Movimentação da “Snake” Sujeita à Força do Gradiente da Imagem.....	32
2.3.4.5	Força do Detector de Arestas	34
2.3.4.6	Movimentação da “Snake” Sujeita à Força do Detector de Arestas	35
2.3.4.7	Força de Enchimento do Contorno	37
2.3.4.8	Movimentação da “snake” Sujeita à Força de Enchimento do Contorno.....	38
2.4	SOLUÇÃO NUMÉRICA	40
2.4.1	Formulação Matemática da Energia Total do Contorno.....	40
2.4.2	Discretização da Função Energética	40
2.4.3	Equação de Evolução.....	41
2.4.4	Discretização das Equações no Espaço	41
2.4.5	Equação de Evolução e sua Discretização no Tempo.....	44
2.5	PROBLEMAS COMUNS DE IMPLEMENTAÇÃO	46
2.5.1	Instabilidade Devido à Discretização no Tempo.....	47
2.5.2	Instabilidade Devido à Discretização no Espaço.....	49
2.5.3	Inicialização da “Snake”	50
2.5.4	Dificuldade na Detecção de Contornos Afastados.....	54
2.5.5	Optimização dos Coeficientes de Rigidez e Elasticidade	55
2.6	SEGUIMENTO DE CONTORNOS	56
2.6.1	Introdução	56
2.6.2	Esquema de Seguimento.....	56
2.7	APLICAÇÕES DE CONTORNOS ACTIVOS	58
3.	IMPLEMENTAÇÃO DO SISTEMA DE DETECÇÃO E SEGUIMENTO DE CONTORNOS.....	59
3.1	INTRODUÇÃO	59
3.2	PLATAFORMA DE DESENVOLVIMENTO SELECIONADA.....	59
3.2.1	Sistema Operativo Utilizado.....	59
3.2.2	Linguagem de Programação	60
3.2.3	Estrutura Básica de um Programa Windows	62
3.2.4	Aplicações em Visual C++.....	65
3.2.5	Utilização das “Microsoft Foundation Classes”	67
3.3	DESCRIÇÃO DO SISTEMA DESENVOLVIDO	69
3.3.1	Potencialidades Requeridas ao Sistema.....	69
3.3.2	Arquitectura do Sistema	70
3.3.3	Implementação do Sistema.....	72

1. INTRODUÇÃO

1.1 MOTIVAÇÃO

A principal motivação para a realização deste trabalho surgiu devido à constatação da cada vez maior aplicação das técnicas de processamento e análise de imagens médicas como ferramenta de diagnóstico. No entanto, poucas aplicações existem, disponíveis em ambientes de ampla divulgação, que incluam capacidades genéricas de processamento de imagem e capacidades específicas num determinado domínio, como seja o da detecção e seguimento de contornos. Desta forma, o desenvolvimento de um sistema que inclua capacidades genéricas de processamento e análise de imagem, e capacidades específicas de detecção, seguimento e análise de contornos, poderá constituir uma ferramenta de grande utilidade no âmbito da análise de imagem médica.

Outra motivação para a realização deste trabalho, prende-se com o crescente interesse do Instituto de Engenharia Biomédica nas técnicas de detecção e seguimento de contornos baseadas em contornos activos. Assim, este estudo surge como uma primeira análise da aplicabilidade destas técnicas na análise de imagens médicas, e como uma primeira abordagem à construção de um sistema integrado de detecção e seguimento de contornos, baseado em contornos activos.

1.2 OBJECTIVOS

Com o presente trabalho pretende-se estudar as técnicas de detecção e seguimento de contornos baseadas em contornos activos ("snakes"). Com base nesse estudo, pretende-se projectar e implementar um sistema com funcionalidades genéricas de processamento e análise de imagem, e funcionalidades específicas de detecção e seguimento de contornos baseadas em contornos activos. Outro objectivo prende-se com a aplicação do sistema desenvolvido à análise de imagens reais, nomeadamente de imagens médicas.

Daqui decorre que os objectivos gerais deste projecto podem ser enunciados da seguinte forma:

- Estudo dos métodos genéricos de detecção de contornos e arestas.

- Estudo dos métodos de detecção e seguimento de contornos baseados em contornos activos.
- Estudo e selecção de uma plataforma que permita o desenvolvimento de raiz de uma aplicação genérica de processamento e análise de imagem.
- Implementação de um sistema que permita efectuar operações genéricas de processamento e análise de imagem e que permita facilmente adicionar novos métodos e algoritmos.
- Inclusão no sistema desenvolvido de operações de detecção de contornos em imagens.
- Inclusão no sistema desenvolvido de operações de seguimento de contornos em sequências de imagens.
- Aplicação dos métodos de detecção e seguimento de contornos desenvolvidos, na análise de imagens reais, nomeadamente na análise de imagens médicas.

1.3 DEFINIÇÃO E ENQUADRAMENTO

O trabalho descrito nesta dissertação enquadra-se no domínio do Processamento e Análise de Imagem e Visão por Computador, abordando o tema específico da detecção e seguimento de contornos.

Tendo em conta que a aplicação das técnicas de Processamento e Análise de Imagem a problemas concretos, implica sempre a utilização de computadores e de técnicas de programação, o presente trabalho enquadra-se ainda nos domínios da Ciência dos Computadores e Engenharia de Software, mais especificamente da Análise de Sistemas e da Programação e Computadores.

Devido a grande parte das análises terem sido efectuadas utilizando imagens médicas - procurando utilizar as técnicas de Processamento e Análise de Imagem e Ciência dos Computadores em problemas médicos concretos - a presente dissertação enquadra-se ainda no domínio da Engenharia Biomédica.

1.4 CONTRIBUIÇÕES

As principais contribuições proporcionadas pelo trabalho desenvolvido foram as seguintes:

- Realização no Instituto de Engenharia Biomédica, do primeiro trabalho relacionado com as técnicas de detecção e seguimento de contornos baseadas em contornos activos.
- Projecto e implementação de raiz de um sistema genérico de processamento e análise de imagem, com objectivos específicos de detecção e seguimento de contornos, capaz ainda de realizar um vasto leque de operações genéricas de processamento digital de imagem.
- Definição de uma estrutura de dados inovadora adaptada a problemas de detecção de contornos e desenvolvimento de um método hierárquico de detecção de contornos, baseado em contornos activos, que opera sobre essa estrutura.
- Projecto e implementação de um novo método de seguimento temporal de contornos baseado em contornos activos, e sua aplicação na análise de exames ecocardiográficos.

Embora a maioria dos métodos e algoritmos utilizados neste trabalho, se baseiem em ideias e propostas de outros autores, a principal contribuição deste trabalho prende-se com a inclusão num único sistema integrado, de todos os algoritmos desenvolvidos. Este sistema, por si só, constitui a principal contribuição do trabalho, podendo ser utilizado como base, no futuro, para a construção de outros sistemas com maiores potencialidades ou com objectivos mais específicos

1.5 ORGANIZAÇÃO GLOBAL DA TESE

Esta dissertação compreende quatro capítulos, para além do primeiro capítulo que constitui a presente introdução.

O segundo capítulo incide sobre as técnicas de detecção de contornos e arestas. Neste capítulo é dada particular relevância às técnicas de detecção de contornos baseadas em contornos activos. É analisado o comportamento básico de um contorno activo e a sua relação com a função energética do contorno, sendo referidas e explicadas as diversas forças internas e externas a que um contorno pode estar sujeito. É ainda analisado o método iterativo de movimentação do contorno e problemas comuns na sua implementação. Neste capítulo é ainda descrita uma metodologia de seguimento de contornos em sequências de imagens e são referidas as principais aplicações actuais de contornos activos.

O capítulo seguinte, aborda o projecto e implementação do sistema de processamento e análise de imagem com funcionalidades específicas de detecção e seguimento de contornos. É analisada inicialmente a selecção da plataforma de

desenvolvimento. São abordados em seguida os requisitos, arquitectura e implementação do sistema, os formatos de imagem utilizados, a interface do sistema com o utilizador e as operações disponibilizadas pelo sistema. Segue-se uma descrição dos métodos de definição inicial de um contorno e sua movimentação de forma a determinar o contorno final. O capítulo é concluído com a descrição do método desenvolvido para efectuar o seguimento de contornos em sequências de imagens.

O quarto capítulo inclui uma análise dos resultados obtidos na análise de imagens fictícias e reais, utilizando o sistema desenvolvido. São analisados os comportamentos dos algoritmos de detecção e seguimento de contornos desenvolvidos, na análise de imagens fictícias (construídas propositadamente para este fim), imagens médicas (ecocardiográficas) e outras imagens reais provenientes de bibliotecas de imagem ou digitalizadas especialmente para esse efeito.

Finalmente, no capítulo final são retiradas as conclusões finais do trabalho e apresentadas perspectivas de desenvolvimento futuro.

2. CONTORNOS ACTIVOS ("SNAKES")

2.1 TÉCNICAS DE DETECÇÃO DE ARESTAS

2.1.1 Introdução

Uma das tarefas mais importantes na percepção visual, e consequentemente em qualquer sistema de visão por computador, é a detecção de regiões com declives acentuados de variação da intensidade luminosa dos pixels¹ das imagens digitalizadas, denominadas arestas ou orlas de intensidade.

As variações de intensidade numa imagem têm origem em diversos fenómenos físicos distintos, como sejam diferenças de iluminação (fontes de luz, sombras e espelhamentos), diferenças na orientação e distância das superfícies de objectos relativamente à câmara, e variações da reflexão de superfícies. Uma vez que as arestas ocorrem frequentemente em zonas da imagem representativas de rebordos de objectos, a detecção de arestas é extensivamente utilizada no apoio à segmentação de imagens e à detecção de contornos.

A representação de uma imagem através das suas arestas apresenta como vantagem adicional, o facto de a quantidade de dados ser significativamente reduzida, enquanto a informação estrutural contida na imagem é, na sua grande maioria, mantida intacta [Canny86].

Numa imagem de uma cena real encontra-se, normalmente, um misto de regiões nas quais o brilho varia de forma suave, e de regiões que apresentam variações acentuadas de brilho. Surgem assim diversos tipos de arestas, consoante o perfil espacial de variação da intensidade que lhe deu origem, representadas esquematicamente na Figura 1:

- Arestas tipo degrau
- Arestas tipo rampa
- Arestas tipo linha
- Arestas tipo telhado

¹ Do inglês "pixels"

Os principais requisitos colocados a um bom detector de arestas, consistem em que responda unicamente à verdadeira estrutura de arestas e seja relativamente insensível ao ruído. A realização computacional eficiente do detector de arestas é também um requisito a impor [Deriche87].

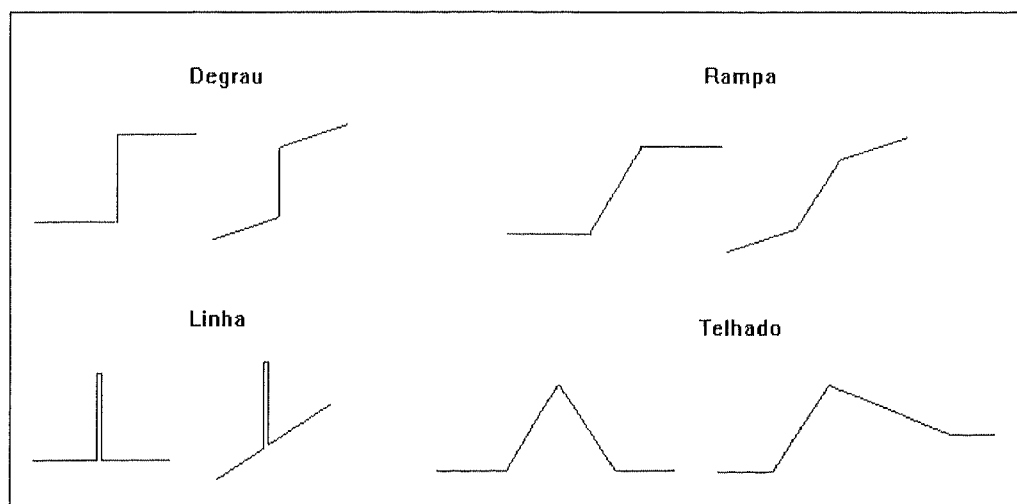


Figura 1: Classificação do tipo de arestas consoante o seu perfil espacial.

As técnicas de detecção de arestas podem dividir-se basicamente em três categorias: Métodos estocásticos [Huang88], ajuste de superfícies [Haralick84, Nalwa86] e filtragem [Canny86, Deriche87]. Os métodos de filtragem apresentam-se como os mais utilizados e de mais fácil implementação algorítmica. Por esta razão, foram os métodos seleccionados para uma descrição mais aprofundada, e para a utilização no sistema de processamento de imagem desenvolvido. Ao longo desta secção serão descritos os principais métodos, incluídos nesta categoria, utilizados actualmente.

2.1.2 Operadores Diferenciais e Afins

Diversos métodos foram propostos para realizar a tarefa da detecção de arestas e a grande maioria sugere a aplicação de operadores locais à imagem para extrair as arestas. Os operadores diferenciais, devido a conduzirem a valores elevados em locais onde o valor do nível de cinzento varia significativamente e de forma rápida, tornam-se bastante adequados à operação de extracção de arestas. Exemplos desses operadores são o gradiente e a segunda derivada de uma imagem, geralmente seguidos, respectivamente, por operações de detecção de máximos e de cruzamentos por zero [Rosenfeld82, Schalkof89].

2.1.2.1 Detecção de Arestas Através dos Máximos do Gradiente

A detecção de arestas através dos máximos da função gradiente baseia-se no esquema representado na Figura 2.

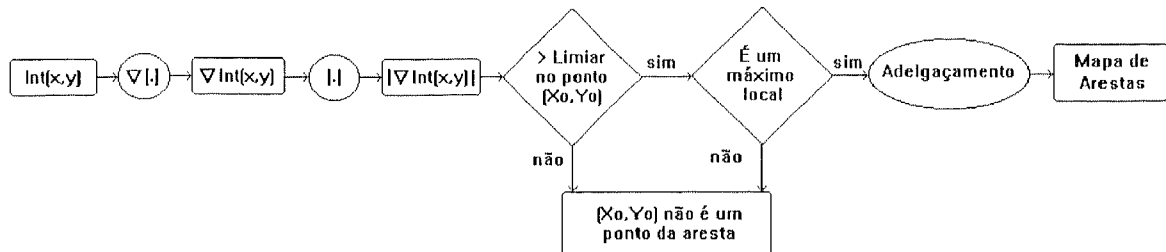


Figura 2: Esquema de funcionamento de um detector de arestas através dos máximos do gradiente.

Considerando o brilho dos pixels da imagem como uma função bidimensional $\text{Int}(x,y)$, a sua primeira derivada, corresponde ao gradiente dado por:

$$\nabla \text{Int}(x,y) = \frac{\partial \text{Int}(x,y)}{\partial x} \cdot \vec{u}_x + \frac{\partial \text{Int}(x,y)}{\partial y} \cdot \vec{u}_y \quad (1)$$

em que $\vec{u}_x$ e $\vec{u}_y$ são os vectores associados ao referencial XY da função imagem.

Partindo da definição do gradiente, é possível calcular a sua amplitude e direcção, da seguinte forma:

$$|\nabla \text{Int}(x,y)| = \sqrt{\left(\frac{\partial \text{Int}(x,y)}{\partial x}\right)^2 + \left(\frac{\partial \text{Int}(x,y)}{\partial y}\right)^2} \quad (2)$$

$$\angle(\nabla \text{Int}(x,y)) = \arctan\left(\frac{\frac{\partial \text{Int}(x,y)}{\partial y}}{\frac{\partial \text{Int}(x,y)}{\partial x}}\right) \quad (3)$$

Num detector por máximos do gradiente, os pontos nos quais a amplitude do gradiente seja superior a um determinado limiar arbitrado, e em que se verifique um máximo local, são assinalados como pontos candidatos a arestas. A selecção do limiar utilizado depende em larga medida da aplicação. À medida que este valor aumenta,

unicamente os pontos correspondentes a grandes variações na função brilho da imagem, são registados como candidatos a arestas.

No domínio discreto, o gradiente é vulgarmente aproximado por diferenças finitas (centradas), resultando:

$$\nabla \text{Int}(x, y) = \frac{\text{Int}(x + \Delta x) - \text{Int}(x - \Delta x)}{2\Delta x} \cdot \vec{u}_x + \frac{\text{Int}(y + \Delta y) - \text{Int}(y - \Delta y)}{2\Delta y} \cdot \vec{u}_y \quad (4)$$

A equação 4 pode ser implementada através da convolução da imagem com uma máscara bidimensional. Máscaras comuns de aproximação ao gradiente (Figura 3) são a máscara de Prewitt (que aproxima a equação 4), de Sobel (com pesos que realçam a contribuição do pixel central) e de Roberts (gradiente segundo as direcções ortogonais 45° e 135°) [Rosenfeld82].

<table><tr><td>-1</td><td>0</td><td>1</td></tr><tr><td>-1</td><td>0</td><td>1</td></tr><tr><td>-1</td><td>0</td><td>1</td></tr></table>	-1	0	1	-1	0	1	-1	0	1	<table><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>-1</td><td>-1</td><td>-1</td></tr></table>	1	1	1	0	0	0	-1	-1	-1	<table><tr><td>0</td><td>1</td></tr><tr><td>-1</td><td>0</td></tr></table> <table><tr><td>1</td><td>0</td></tr><tr><td>0</td><td>-1</td></tr></table>	0	1	-1	0	1	0	0	-1	<table><tr><td>-1</td><td>0</td><td>1</td></tr><tr><td>-2</td><td>0</td><td>2</td></tr><tr><td>-1</td><td>0</td><td>1</td></tr></table>	-1	0	1	-2	0	2	-1	0	1	<table><tr><td>1</td><td>2</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>-1</td><td>-2</td><td>-1</td></tr></table>	1	2	1	0	0	0	-1	-2	-1
-1	0	1																																														
-1	0	1																																														
-1	0	1																																														
1	1	1																																														
0	0	0																																														
-1	-1	-1																																														
0	1																																															
-1	0																																															
1	0																																															
0	-1																																															
-1	0	1																																														
-2	0	2																																														
-1	0	1																																														
1	2	1																																														
0	0	0																																														
-1	-2	-1																																														
Prewitt		Roberts		Sobel																																												

Figura 3: Máscaras 3x3 de aproximação ao gradiente de Prewitt, Roberts e Sobel.

A detecção de arestas utilizando a abordagem referida, apresenta três problemas essenciais:

- Se for utilizado unicamente o método de detecção de pontos com uma amplitude do gradiente superior a um determinado limiar, as arestas resultantes podem apresentar-se sob a forma de traços largos. Para transformar os resultados obtidos, num conjunto de arestas bem demarcadas e estreitas, é necessário aplicar uma operação de adelgaçamento à imagem resultante da limiarização. Esta operação pode ser realizada eliminando todos os pontos que não constituem máximos locais, preferencialmente segundo a direcção da aresta a que possam pertencer.
- Os contornos detectados através deste método, apresentam-se bastante descontínuos devido ao carácter direccionado das máscaras de convolução utilizadas. Normalmente, os pontos que apresentam uma amplitude elevada segundo uma das direcções, possuirão variações fracas segundo a direcção perpendicular. Sendo

assim, as arestas resultantes de contornos curvilíneos, poderão ser detectadas como pequenas linhas descontínuas.

- Pontos isolados marcados como pertencentes a arestas, que aparecem distribuídos aleatoriamente no mapa de arestas, resultam provavelmente de ruído na imagem original. Este problema pode ser minorado através da suavização prévia da imagem original ou através de algoritmos muito simples de remoção de ruído operando sobre o mapa de arestas obtido.

Os métodos apresentados nesta secção, podem ser melhorados de diversas maneiras. No entanto, métodos baseados no cálculo de alguma forma de gradiente ou diferenciação, são tipicamente muito sensíveis ao ruído. A presença de ruído degrada bastante, em todas as situações, os resultados que se podem obter com este tipo de métodos.

2.1.2.2 Detecção de Arestas Através dos Cruzamentos por Zero do Laplaciano

O objectivo dos algoritmos de detecção de arestas consiste em localizar regiões onde a intensidade dos pixels varia rapidamente. Na secção anterior foram analisados métodos baseados na verificação da amplitude da primeira derivada dessa intensidade. Outra possibilidade consiste em assumir que as arestas se encontram nos extremos locais da função gradiente, ou seja, nos cruzamentos por zero da segunda derivada da função intensidade.

O principal problema desta técnica, consiste no facto de que, assinalando como arestas os pontos correspondentes aos cruzamentos por zero, obtém-se como resultado, um elevado número de pontos aresta. Como não é efectuada nenhuma verificação da amplitude do gradiente, sendo apenas verificada a existência de um extremo local, qualquer pequena variação isolada é suficiente para gerar pontos de aresta. Devido a esta sensibilidade ao ruído, a aplicação de uma suavização prévia à imagem original é desejável neste tipo de métodos.

A expressão para a segunda derivada da função imagem, corresponde ao Laplaciano, dado por:

$$\nabla^2 \text{Int}(x,y) = \nabla(\nabla \text{Int}(x,y)) = \frac{\partial^2 \text{Int}(x,y)}{\partial x^2} + \frac{\partial^2 \text{Int}(x,y)}{\partial y^2} \quad (5)$$

Esta equação pode ser discretizada, substituindo as derivadas parciais de segunda ordem por diferenças finitas de segunda ordem, obtendo-se as máscaras de aproximação apresentadas na Figura 4.

0	1	0	1	1	1	-1	2	-1
1	-4	1	1	-8	1	2	-4	2
0	1	0	1	1	1	-1	2	-1

Figura 4: Máscaras 3x3 de aproximação ao Laplaciano.

Contrariamente aos máximos do gradiente, as linhas de cruzamento por zero do Laplaciano, tendem a ser contínuas, fechadas e estreitas, pois correspondem a linhas de nível da função gradiente. Desta forma, são desnecessárias operações de adelgaçamento e condições para garantir a continuidade das arestas detectadas. No entanto, para que o Laplaciano apresente resultados correctos [Torre86, Marr80], as variações do brilho na imagem devem ser lineares na sua direcção longitudinal. Se tal não se verificar (por exemplo nos cantos), os cruzamentos por zero aparecerão ligeiramente deslocados.

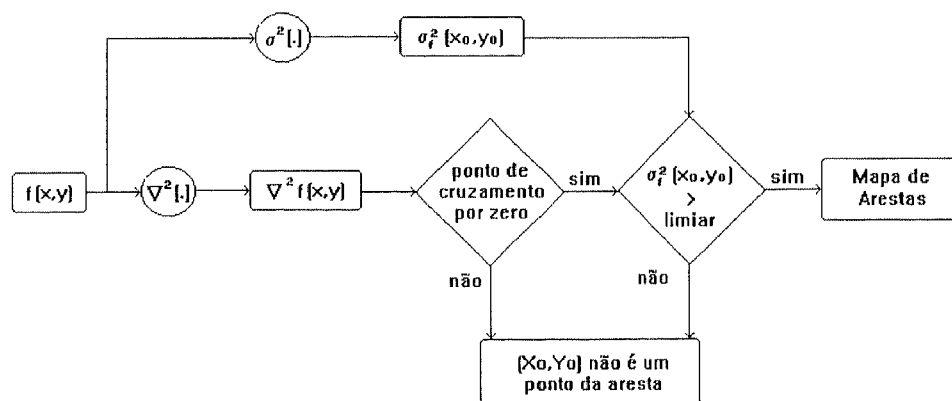


Figura 5: Esquema de funcionamento de um detector de arestas através dos cruzamentos por zero do Laplaciano.

Devido a não serem impostas condições de amplitude mínima aceitável, um grande número de pontos, correspondentes a pequenas variações de brilho, são erradamente assinalados como arestas. Em regiões em que a variância local da imagem é reduzida, este problema é ainda mais agravado, conduzindo qualquer pequena

variação da intensidade da imagem a um cruzamento por zero do Laplaciano. Um método utilizado para evitar estes falsos contornos (Figura 5) consiste em requerer que a variância local seja suficientemente elevada, para que um ponto possa ser assinalado como aresta.

2.1.3 Critérios de Desempenho

A detecção de arestas é um problema comum a muitas aplicações, sendo, por isso útil dispor de critérios de desempenho genéricos para os detectores de arestas, que sejam independentes das suas possíveis aplicações. Os critérios principais, referidos inicialmente por Marr e Hildreth [Marr80, Marr82] e formulados matematicamente por Canny [Canny86], estão relacionado com a detecção precisa, sem erros e única, das arestas contidas na imagem. Um determinado filtro, constituirá um bom detector de arestas se a sua convolução com a imagem original, permitir obter:

- **Detecção Correcta** - com baixa probabilidade de não detectar um ponto aresta correcto e elevada probabilidade de não assinalar falsas arestas. Uma vez que estas probabilidades decrescem monotonicamente com o aumento da relação Sinal-Ruído, um critério adequado consiste na maximização dessa relação.
- **Localização Precisa** - estando os pontos assinalados como arestas, tão próximo quanto possível do centro da verdadeira aresta. Este critério corresponde à minimização de uma medida dessa distância, tal como a variância da posição dos cruzamentos por zero.
- **Unicidade da Resposta** - não produzindo o detector múltiplas resposta à mesma aresta. Se for considerado o comportamento do filtro na vizinhança da aresta, a unicidade resulta da minimização do número de cruzamentos por zero ou limitação do número de máximos da resposta, nessa vizinhança.

Estes critérios, são normalmente formulados para o caso unidimensional (sinais) e para a detecção óptima de arestas do tipo degrau. A extensão para o caso bidimensional é derivada a partir da análise unidimensional. A formulação matemática destes critérios, pode ser encontrada em diversas referências [Canny86, Deriche87, Shen92], podendo-se encontrar uma síntese em [Correia95].

2.1.4 Filtros para a Detecção de Arestas

2.1.4.1 Laplaciano do Gaussiano

Um operador baseado na segunda derivada da imagem é o detector de arestas de Marr [Marr80]. Aqui, a segunda derivada é calculada utilizando o filtro Laplaciano do Gaussiano² (LoG). O Laplaciano tem a vantagem de ser uma medida isotrópica da segunda derivada da imagem, isto é, a amplitude das arestas é obtida independentemente da sua orientação, convoluindo a imagem unicamente com uma máscara.

No trabalho de Marr e Hildreth [Marr80], a ideia básica consiste na determinação das arestas em múltiplas escalas, através de filtros sintonizados para cada uma. Assume-se que as variações de intensidade nas imagens existem numa vasta gama de escalas espaciais e que essas variações têm uma extensão limitada. Desta forma, cada filtro deve ter um espectro de frequências de banda limitada de forma a detectar unicamente as variações numa dada escala.

O filtro de suavização óptimo é considerado nesta abordagem, de forma empírica, como sendo o Gaussiano, cuja expressão, para o caso unidimensional, é dada por:

$$h_G(x) = \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{\left(-\frac{x^2}{2\sigma^2}\right)} \quad (6)$$

ficando, para o caso bidimensional:

$$h_G(x, y) = \frac{1}{2\pi\sigma^2} \cdot e^{\left(-\frac{x^2+y^2}{2\sigma^2}\right)} \quad (7)$$

O filtro de detecção óptimo é obtido pela derivada de primeira ordem - para um detector por máximos - ou de segunda ordem - para um detector por cruzamentos por zero - da imagem convoluída com o Gaussiano. Neste último caso, temos então:

$$\nabla^2 [Int(x, y) \otimes h_G(x, y)] \quad (8)$$

ou, equivalentemente:

² Do inglês "Laplacian of Gaussian"

$$[\text{Int}(x,y) \otimes \nabla^2 h_G(x,y)] \quad (9)$$

em que:

$$\nabla^2 h_G(x,y) = \frac{1}{\sigma^3 \sqrt{2\pi}} \cdot \left(\frac{x^2 + y^2}{\sigma^2} - 2 \right) \cdot e^{\left(-\frac{x^2 + y^2}{2\sigma^2} \right)} \quad (10)$$

expressão do já referido Laplaciano do Gaussiano [Marr82].

A posição das arestas é dada pela posição dos cruzamentos por zero da imagem LoG obtida. A escala a que se pretende detectar as arestas pode ser controlada alterando a variância do Gaussiano.

2.1.4.2 Canny

Canny segue uma abordagem diferente [Canny86] baseando-se nos critérios que estabeleceu para um filtro óptimo de detecção de arestas tipo degrau (apresentados na secção 2.1.3). A sua abordagem baseia-se na necessidade de maximização simultânea dos dois primeiros critérios referidos. Desta forma Canny tenta minimizar o produto para simplificar a solução, constituindo o terceiro critério uma restrição à sua maximização. A solução geral que foi encontrada no intervalo $[0, W]$ consiste na soma de quatro exponenciais:

$$f_C(x) = a_1 \cdot e^{\alpha \cdot x} \cdot \text{sen}(\omega \cdot x) + a_2 \cdot e^{\alpha \cdot x} \cdot \text{cos}(\omega \cdot x) + a_3 \cdot e^{-\alpha \cdot x} \cdot \text{sen}(\omega \cdot x) + a_4 \cdot e^{-\alpha \cdot x} \cdot \text{cos}(\omega \cdot x) + C \quad (11)$$

extensível ao intervalo $[-W, W]$, sujeita às condições fronteira:

$$f_C(0) = 0 \quad f_C(W) = f_C(-W) = 0 \quad f'_C(0) = S \quad f'_C(W) = f'_C(-W) = 0 \quad (12)$$

onde **S** é uma constante desconhecida igual ao declive da função $f_C(x)$ na origem.

A forma do filtro obtida, para **k** máximo, é muito semelhante à forma da primeira derivada da função Gaussiana. Sendo assim, a primeira derivada da função Gaussiana, pode ser considerada como sendo um filtro sub-óptimo.

2.1.4.3 Deriche

Utilizando os critérios de Canny, e removendo a condição de suporte limitado, Deriche chegou a uma outra forma do detector óptimo [Deriche87], possuindo índices de desempenho superiores aos de Canny. Deriche modifica as condições fronteira anteriores para as de um filtro de resposta impulsional infinita, fazendo $W \rightarrow \infty$, obtendo as seguintes condições:

$$f(0) = 0 \quad f(+\infty) = 0 \quad f'(0) = S \quad f'(+\infty) = 0 \quad (13)$$

o que conduz à seguinte expressão para o filtro óptimo:

$$f_D(x) = -c \cdot e^{-\alpha|x|} \cdot \sin(\omega \cdot x) \quad (14)$$

O filtro fica totalmente definido por apenas dois parâmetros α e ω , sendo o valor de c determinado pela condição de ganho, que geralmente se pretende unitário.

Para o caso que, segundo Deriche, conduz ao melhor compromisso entre o cumprimento dos critérios de detecção correcta e precisa, ωx é muito pequeno, podendo ser efectuada a aproximação $\sin(\omega x) = \omega x$, transformando-se a expressão do filtro óptimo em:

$$f_D(x) = -c \cdot x \cdot e^{-\alpha|x|} \quad (15)$$

dependente apenas do parâmetro α , que regula a actuação do filtro no sentido de obter uma melhor detecção ou uma melhor localização.

2.1.4.4 Sarkar e Boyer

Sarkar e Boyer [Sarkar91] adoptam os critérios e a abordagem de Canny, excepto no que se refere à extensão espacial do filtro, condições fronteira e parâmetros de optimização. Optam pela extensão infinita do filtro e por condições fronteira idênticas às de Deriche, obtendo a seguinte expressão:

$$f_{SB}(x) = -a \cdot e^{-\alpha \cdot x} \cdot (\cos(\phi) - \cos(\beta \alpha x + \phi)) \quad ; \quad \alpha \geq 0 \quad (16)$$

Os parâmetros que influem na forma do filtro são ϕ e β . Os parâmetros a e α , dizem respeito à amplitude e escala pretendidas.

2.1.4.5 Shen e Castan

Shen e Castan [Shen92] apresentaram um detector óptimo, baseado em critérios de processamento de sinal, capaz de localizar com precisão arestas, quer no modelo de aresta única, quer no modelo estocástico de múltiplas arestas. Em ambos os casos obtêm uma expressão equivalente a:

$$f_{SC}(x) = c \cdot e^{-\alpha|x|} \quad (17)$$

para o filtro de suavização com vista à detecção de orlas do tipo degrau. Esta expressão é bastante simples, contendo unicamente um parâmetro ajustável α , sendo o valor de c determinado pela condição de ganho pretendido.

2.1.5 Operações Complementares

A aplicação de um dos filtros mencionados para a detecção de arestas numa imagem, constitui apenas uma das fases de processamento para a obtenção de um mapa de arestas de boa qualidade. Embora os filtros apresentados sejam todos óptimos com base nos critérios que estão na base da sua formulação, os mapas de arestas que obtêm estão longe de ser uma representação fidedigna das arestas da imagem original. Os resultados obtidos na detecção de arestas utilizando os filtros propostos, podem ser significativamente melhorados através da utilização de diversas operações complementares:

- **Supressão de não-máximos** - com remoção de todos os pontos que não sendo máximos de amplitude na resposta do filtro à imagem original, se situam na sua proximidade e apresentam ainda um nível elevado que poderá conduzir a uma localização imprecisa da aresta correspondente. Esta operação corresponde a um adelgaçamento das arestas encontradas.
- **Limiarização** - destinada a transformar a imagem numa representação unicamente das arestas mais representativas, removendo todas aquelas que correspondem a transições de pequena intensidade, normalmente associadas a ruído. É vulgar a utilização de uma limiarização com dois níveis de histerese [Canny86, Deriche87]. Esta técnica consiste em seleccionar todos os pontos com valor superior ao limiar mais elevado, juntamente com os pontos a eles conexos e com valor superior ao segundo limiar de valor mais reduzido.
- **Validação de passagens por zero** - procurando eliminar as passagens por zero que resultam de ruído nas imagens originais. Esta remoção pode ser efectuada em dois

passos [Shen92]. Em primeiro lugar é determinado o valor da amplitude do gradiente no ponto homólogo da imagem original e procede-se a uma limiarização de forma a remover, na imagem com os cruzamentos por zero, os pontos que apresentam um baixo gradiente. Em segundo lugar é verificada a concordância de sinal entre a primeira derivada e a transição da segunda derivada [Clark89, Shen92].

O tipo de filtragem utilizada - detecção por máximos ou por cruzamentos por zero - determina as operações aplicáveis. As operações de supressão de não-máximos e de limiarização só são aplicáveis a imagens resultantes de uma filtragem por detector de máximos da amplitude do gradiente. A operação de validação das passagens por zero, tal como o nome indica, só pode ser aplicada aos resultados da detecção por cruzamentos por zero da segunda derivada da imagem.

2.1.6 Utilização dos Mapas de Arestas na Detecção de Contornos

Embora a detecção de arestas desempenhe um papel inicial extremamente importante em qualquer sistema de visão por computador, as imagens resultado obtidas através dos detectores de arestas não têm a descrição de alto nível necessária a grande parte das tarefas da visão por computador [Lowe89].

Os resultados obtidos utilizando os detectores de arestas consistem em imagens contendo pontos classificados como pertencendo ou não a arestas, não fornecendo desta forma a informação de alto nível desejada sobre os contornos dos objectos contidos na imagem.

Para formar um contorno é necessário ligar os pontos das arestas que lhe pertencem, não se apresentando esta tarefa, na maioria dos casos, nada simples. No entanto, as arestas contêm a grande maioria da informação necessária à efectiva detecção de contornos, apresentando-se como um excelente ponto de partida para a construção de uma representação de alto nível dos contornos contidos na imagem. Desta forma, ao longo das próximas secções, serão analisadas as técnicas de detecção de contornos baseadas em contornos activos, sendo referida com detalhe a contribuição que os resultados obtidos pelos detectores de arestas, podem oferecer ao método de detecção de contornos.

2.2 INTRODUÇÃO AOS CONTORNOS ACTIVOS

A utilização de modelos de minimização de energia tem uma já antiga e rica história no campo da visão por computador, situando-se as primeiras aplicações no início dos anos 70 com o modelo estéreo de Sperling [Sperling70].

A utilização de contornos activos (também conhecidos como "snakes") na extracção de características de interesse em imagens, foi introduzida por Kass, Witkin e Terzopoulos em 1987 [Kass87, Kass88]. No seu trabalho, um contorno activo é definido como uma curva de minimização de energia guiada por forças externas e influenciada por forças da imagem que a impulsionam para características relevantes da imagem como linhas ou arestas.

Os contornos activos apresentam enormes possibilidades na área da visão computacional. Podem ser aplicados nas mais diversas tarefas de extracção de características em imagens, tais como:

- Detecção de linhas e arestas.
- Detecção de contornos subjectivos.
- Análise de movimento em sequências de imagens.
- Análise de correspondências em imagens estéreo.

Numa definição básica, uma "snake" pode ser considerada como um contorno sujeito a forças resultantes da sua própria elasticidade e rigidez, a uma força potencial dependente das características locais da imagem e possivelmente a uma força de pressão [Cohen91] que tende a insuflá-lo (ou eventualmente a contraí-lo).

Partindo de uma posição inicial grosseira, através de um processo iterativo, o contorno vai-se deslocando no sentido de minimizar a sua função energética. A posição final de equilíbrio corresponderá a um mínimo local da função energética definida anteriormente. Nesta posição, as forças a que o contorno está sujeito, anular-se-ão mutuamente, de forma a que o seu deslocamento será nulo.

Através da escolha correcta dos parâmetros de rigidez e elasticidade do contorno e da sua função energética é possível detectar contornos com perfis totalmente distintos correspondentes às mais diversas características da imagem (desde regiões com elevadas variações de brilho até regiões que correspondam a baixos ou elevados valores de brilho).

2.3 MODELIZAÇÃO DE UM CONTORNO ACTIVO

2.3.1 Comportamento Básico de um Contorno Activo

Um contorno activo na sua definição genérica é um mapeamento que faz corresponder a cada ponto (s) de uma linha $[0, 1]$, uma posição $v(s) = (x(s), y(s))$ no espaço $\mathbb{R}^2$:

$$\begin{aligned} \Omega = [0,1] &\rightarrow \mathbb{R}^2 \\ s &\rightarrow v(s) = (x(s), y(s)) \end{aligned} \quad (18)$$

Um contorno activo, numa definição mais simples e discreta, é constituído por um conjunto de pontos de controlo, efectivamente ligados entre si por linhas rectas. Cada ponto de controlo possui uma posição espacial dada pelas suas coordenadas (X,Y) na imagem. A localização de uma "snake" é desta forma, totalmente especificada pelo seu número de pontos de controlo e suas respectivas coordenadas. Ajustes na "snake" são efectuados através de movimentações sucessivas no espaço dos seus pontos de controlo.

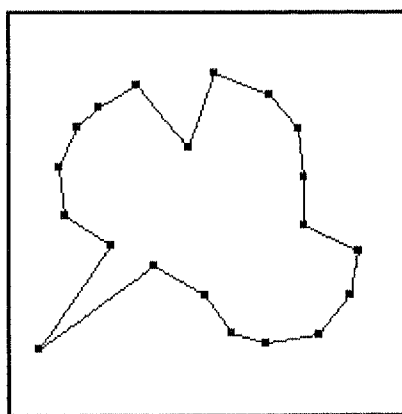


Figura 6: Exemplo de uma "snake" constituída por 20 pontos de controlo.

Na Figura 6 podemos observar uma "snake" constituída por vinte pontos de controlo. Embora a "snake" apresentada seja um contorno fechado, ou seja, o último ponto possui coordenadas idênticas às do primeiro, num caso genérico, uma "snake" não tem necessariamente de ser fechada. Pode simplesmente ser constituída por um conjunto de pontos ligados contendo um ponto inicial ("cabeça da cobra") e um ponto final ("cauda").

2.3.2 Função de Energia a Minimizar

A principal virtude da representação de contornos através de "snakes" reside no facto de ser possível especificar um vasto conjunto de propriedades do contorno através da definição da função designada por energia, por analogia com os sistemas físicos. O método iterativo de deslocamento da "snake" conduz o contorno no sentido de reduzir a sua energia. Desta forma, através da especificação de uma função energética apropriada, é possível deslocar a "snake" para as características da imagem desejadas, tais como arestas ou zonas de brilho elevado.

A função energética a minimizar é constituída por duas partes, a energia interna e a energia externa, ou seja:

$$E_{\text{Snake}}(v) = E_{\text{Interna}}(v) + E_{\text{Externa}}(v) \quad (19)$$

A energia interna é a parte dependente das propriedades intrínsecas da "snake" tais como o seu comprimento e a sua curvatura. A energia externa depende de factores tais como a estrutura da imagem e restrições impostas pelo utilizador.

Estendendo a analogia física, o movimento do contorno pode ser considerado como sendo provocado por forças simuladas que sobre ele actuam. Tal como as forças em sistemas físicos agem de forma a reduzir a energia desses sistemas, as forças simuladas agem sobre a "snake" obrigando-a a se movimentar de forma a reduzir a sua energia total.

2.3.3 Forças Internas do Contorno

O primeiro termo de energia da "snake" (E_{interna}) permite definir as propriedades mecânicas do contorno nomeadamente no que diz respeito à sua elasticidade e rigidez. Este termo é vulgarmente a soma de duas componentes, a energia elástica e a energia de flexão.

$$E_{\text{Interna}}(v) = E_{\text{Elastica}}(v) + E_{\text{Flexão}}(v) \quad (20)$$

A energia elástica permite definir a resistência do contorno à tensão, ou seja, ao alongamento do contorno enquanto que, a energia de flexão permite quantificar a resistência do contorno à curvatura.

2.3.3.1 Força Elástica

Para que os pontos de uma "snake" se mantenham ligados e não se afastem demasiado dos seus pontos vizinhos, é necessário definir uma função energética que aumente de valor com o afastamento entre pontos contíguos. Uma forma conveniente para tal função energética será a soma dos quadrados das distâncias entre pontos de controlo adjacentes. Utilizando os quadrados das distâncias, em vez dos seus valores simples, é aumentada a contribuição correspondente a pontos largamente espaçados, o que se apresenta desejável.

A consideração desta energia é basicamente o mesmo, em analogia com os sistemas físicos [Beer81, Branco85], que considerar que os pontos de controlo da "snake" se encontram ligados por molas obedecendo à lei de Hook [Branco85]. A diferença consistirá em que num sistema físico as molas não se contraem até um comprimento nulo, apresentando sempre um comprimento residual finito, enquanto no sistema simulado correspondente à "snake", o comprimento residual não existe.

Usualmente a soma dos quadrados das distância é multiplicada por uma constante correspondendo, numa analogia com um sistema físico, à força das molas utilizadas, surgindo assim a seguinte formulação para a energia elástica da "snake":

$$E_{\text{Elastica}}(v) = \int_{\Omega} w1(s) |v'(s)|^2 ds \quad (21)$$

Discretizando a equação 21 obtemos:

$$E_{\text{Elastica}}(v) = \sum_{i=1}^n w1(i) (\text{Dist}(v(i), v(i+1)))^2 \quad (22)$$

Na equação, n representa o número de pontos de controlo considerado e $w1(i)$ é um valor intrínseco do contorno que representa a elasticidade do contorno na região correspondente ao ponto i . Estes valores podem ser alterados livremente de acordo com o problema em análise. Como a "snake" em análise corresponde a um contorno fechado, a posição do ponto de controlo 0 será idêntica à do ponto de controlo n .

As distâncias entre pontos de controlo consecutivos são calculadas utilizando o teorema de Pitágoras, ou seja:

$$\text{Dist}(v(i), v(i+1)) = \sqrt{(x(i) - x(i+1))^2 + (y(i) - y(i+1))^2} \quad (23)$$

representando-se por $x(i)$ a coordenada segundo o eixo dos XX do i -ésimo ponto de controlo do contorno e por $y(i)$ a coordenada segundo o eixo dos YY do referido ponto.

A transformação da energia elástica na correspondente força elástica é efectuada através da diferenciação da equação 22 segundo as direcções X e Y . A força elástica obtida para o i -ésimo ponto segundo as direcções ortogonais X e Y será desta forma a seguinte:

$$\begin{cases} F_{\text{Elast}} x(i) = 2 * w1(i) * ((x(i-1) - x(i)) + (x(i+1) - x(i))) \\ F_{\text{Elast}} y(i) = 2 * w1(i) * ((y(i-1) - y(i)) + (y(i+1) - y(i))) \end{cases} \quad (24)$$

Uma força elástica definida tal como na equação 24, aplicada a todos os pontos de controlo do contorno tenderá a contrair e suaviza-lo, aproximando e alinhando os pontos de controlo.

2.3.3.2 Movimentação da "Snake" Sujeita à Força Elástica

Tendo deduzido na secção precedente as expressões matemáticas para a força elástica aplicada em cada ponto da "snake", a dinâmica de um contorno a ela sujeito pode muito facilmente ser implementada. Em cada instante de tempo, cada um dos pontos do contorno será deslocado de uma quantidade proporcional à força que sobre ele age. As equações de evolução do contorno assumem desta forma o seguinte formato:

$$\begin{cases} x(i, t+1) = x(i, t) + K * F_{x_{\text{Elastica}}}(i, t) \\ y(i, t+1) = y(i, t) + K * F_{y_{\text{Elastica}}}(i, t) \end{cases} \quad (25)$$

K é uma constante definida pelo utilizador que determina a velocidade de movimentação de um ponto para uma determinada força. Os valores $x(i, t)$ e $y(i, t)$ representam as posições segundo os eixos XX e YY do i -ésimo ponto do contorno no instante t . $F_{x_{\text{Elástica}}}(i, t)$ e $F_{y_{\text{Elástica}}}(i, t)$ correspondem às forças segundo os eixos dos XX e YY aplicadas ao ponto i , no instante temporal t .

Na Figura 7, podemos observar a evolução de um contorno com 20 pontos de controlo, ao longo de 30 iterações, quando lhe é aplicada unicamente uma força interna elástica constante ao longo de todo o contorno.

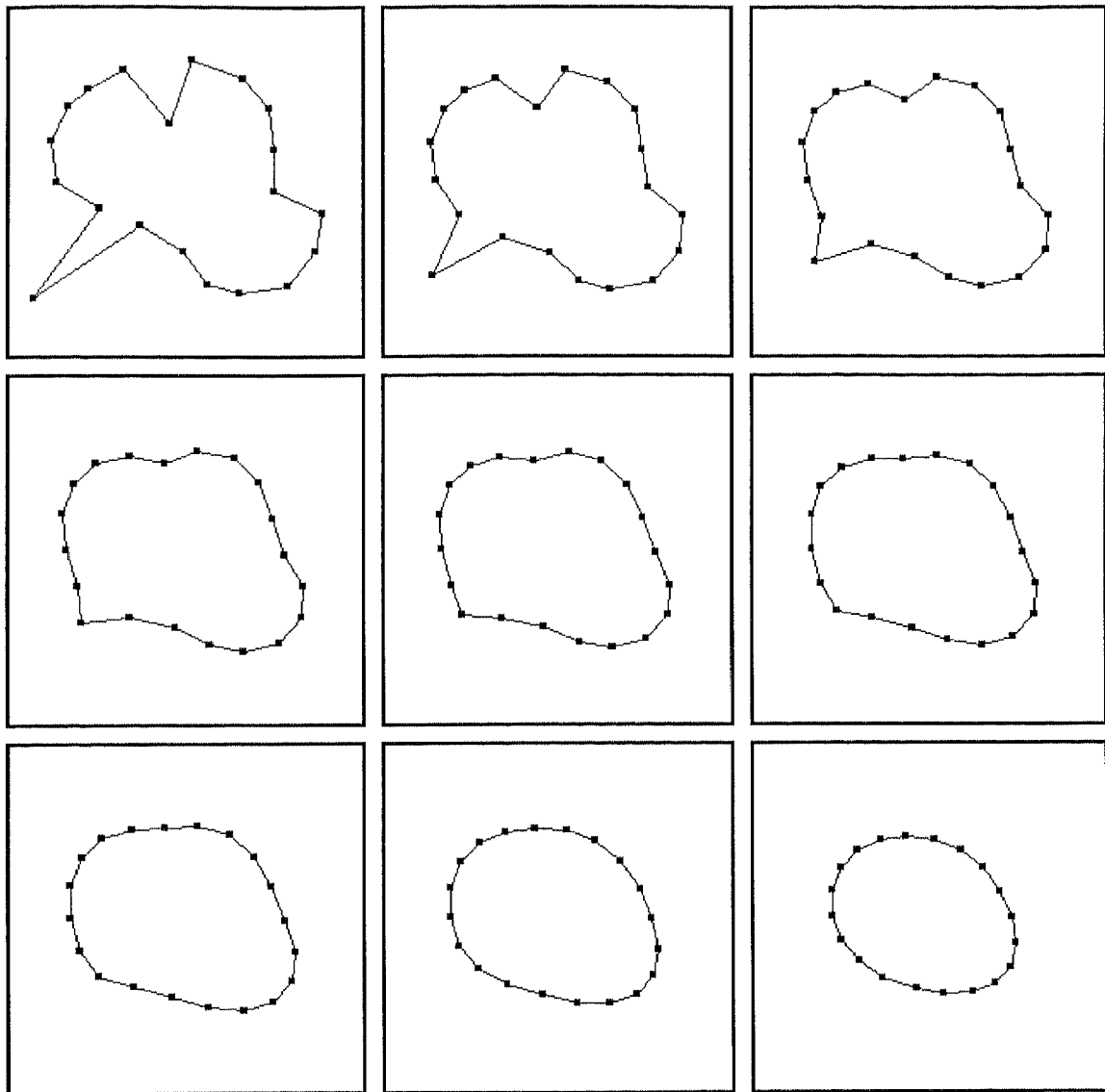


Figura 7: Evolução de um contorno simples (20 pontos) quando lhe é aplicada uma força elástica ($w_1=1/h^2$), iterações (0, 2, 4, 6, 8, 10, 12, 20 e 30).

A força elástica provoca um efeito de contracção e suavização simultânea do contorno. O efeito de contracção explica-se devido à própria analogia física desta força, ou seja, um conjunto de molas perfeitas ligando pontos adjacentes do contorno. O efeito de suavização é em parte derivado do efeito anterior, uma vez que os pontos do contorno são puxados pelos seus vizinhos, atenuando desta forma as irregularidades do contorno.

O efeito do parâmetro w_1 é o de controlar a intensidade da força elástica. Valores elevados do parâmetro w_1 , corresponderão a uma elevada força elástica, enquanto que valores reduzidos deste parâmetro corresponderão a uma reduzida intensidade da força elástica. Na Figura 8, podemos observar a evolução do contorno apresentado na Figura 6, quando lhe é aplicada uma força elástica de valor duplo da anterior.

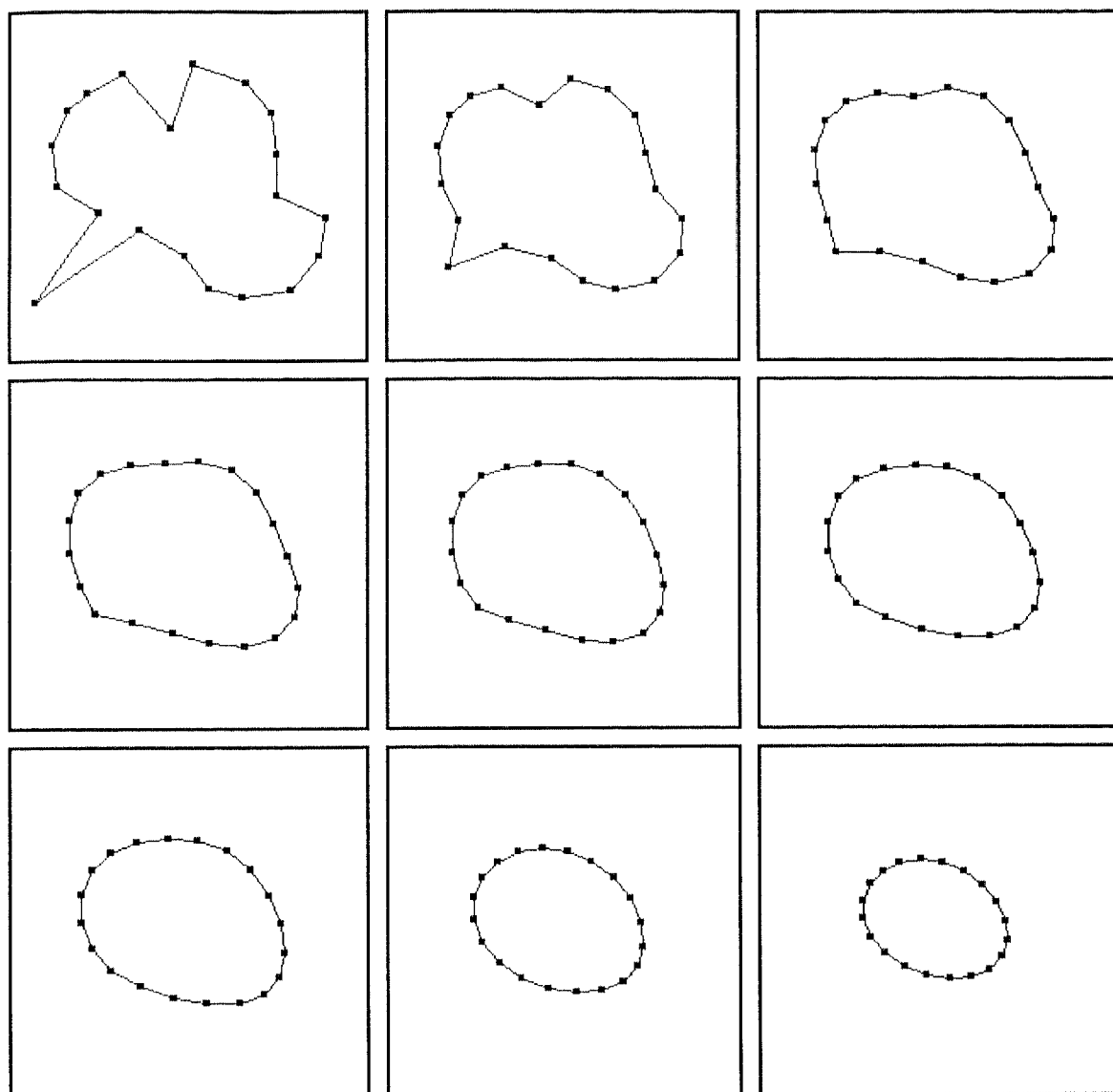


Figura 8: Evolução do contorno 1 ($w_1=2/h^2$), iterações (0, 2, 4, 6, 8, 10, 12, 20, 30).

Na figura é visível a muito mais rápida regularização e contracção do contorno relativamente à evolução apresentada na Figura 7. O contorno rapidamente tende para um formato circular e, no limite, uma vez que as molas foram consideradas como perfeitas (ou seja, sem comprimento residual), tenderá a contrair-se para um único ponto.

2.3.3.3 Força de Flexão

De forma a que o contorno apresente uma forma suave limitando o seu grau de curvatura (sem no entanto interferir na proximidade dos pontos adjacentes) é necessário definir uma outra função energética designada vulgarmente por energia de flexão. Com

esta função energética é conseguido para o contorno um comportamento semelhante a uma fina barra metálica (curva suave mas que não se contrai) em vez do comportamento tipo fio elástico obtido com a consideração da energia elástica.

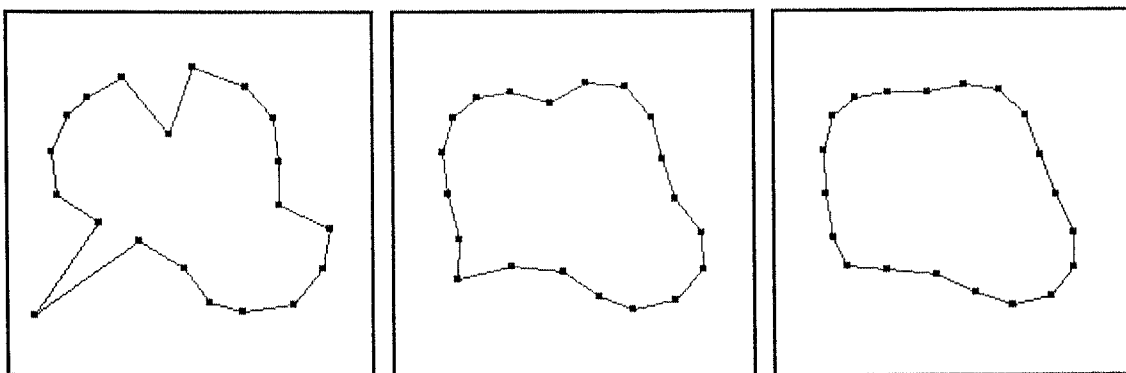
A equação que define uma força com as propriedades de resistência à flexão apresentadas, corresponde à integração ao longo do contorno do quadrado da segunda derivada da posição do contorno, pesado por um factor designado por w_2 que define a importância desta energia.

$$E_{\text{Flexão}}(v) = \int_{\Omega} w_2(s) |v''(s)|^2 ds \quad (26)$$

Discretizando a equação 26, a função energética de flexão é simplesmente constituída pela soma dos quadrados das curvaturas do contorno medidas nos pontos de controlo. Com esta função, quanto maiores forem os ângulos definidos pelos vizinhos mais próximos de um determinado ponto de controlo, maior será a curvatura nesse ponto e, conseqüentemente, maior será a contribuição energética para a função definida.

2.3.3.4 Movimentação da "Snake" Sujeita à Força de Flexão

A força de flexão tem como principal efeito, atenuar as curvaturas do contorno sem o contrair. De facto, esta força foi definida de forma aproximada, a partir de uma função energética proporcional à segunda derivada espacial da posição do contorno, ou seja, à sua flexão. A minimização da função energética referida terá conseqüentemente o efeito de reduzir a flexão do contorno tornando o seu perfil mais suave. Na Figura 9, pode ser observada a evolução, do contorno inicial apresentado na Figura 6, ao longo de 30 iterações quando sujeito unicamente a uma força de flexão constante ao longo de toda a sua extensão.



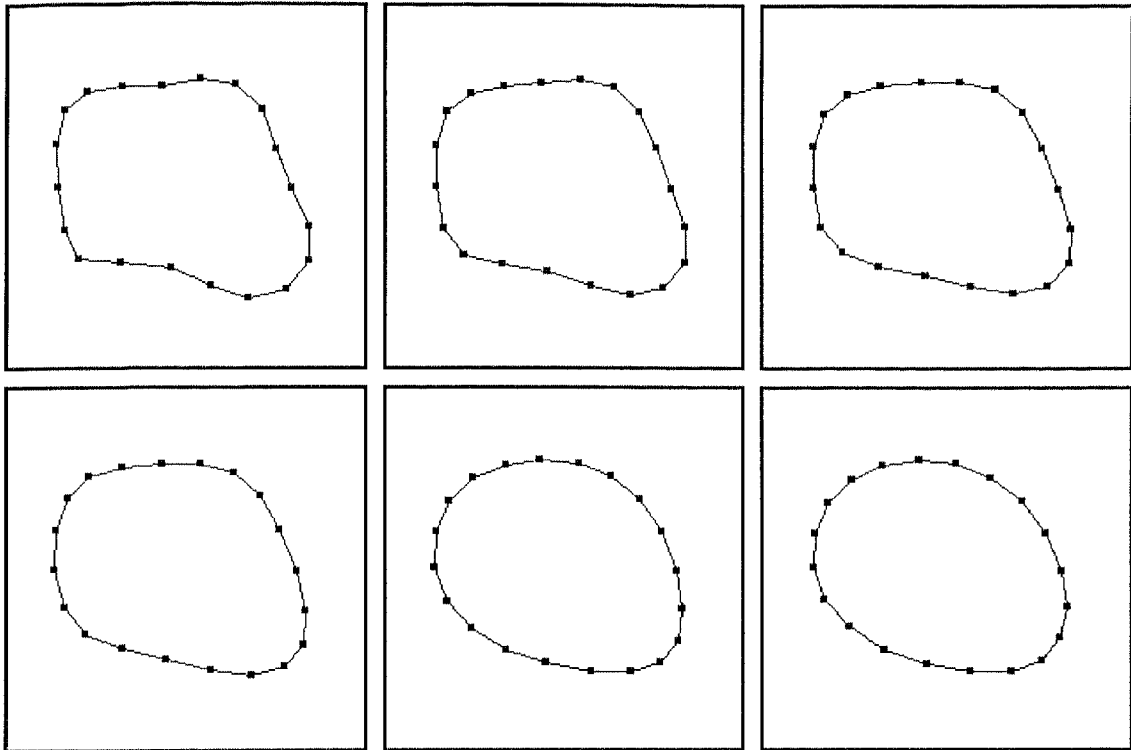


Figura 9: Evolução de um contorno simples (20 pontos) quando lhe é aplicada uma força de resistência à flexão ($w_2=1/h^4$), iterações (0, 2, 4, 6, 8, 10, 12, 20 e 30).

Comparando o resultado da Figura 9 com aquele obtido utilizando unicamente a força elástica podemos verificar que o efeito de suavização obtido com a consideração da força de flexão, é muito semelhante ao obtido com a força elástica, não se obtendo no entanto qualquer efeito de contracção do contorno.

2.3.3.5 Movimentação da "Snake" Sujeita às Forças Elástica e de Flexão

A utilização simultânea das Forças Elástica e de Flexão, faz com que o contorno apresente simultaneamente propriedades de resistência ao alongamento e de resistência à curvatura. A função energética resultante consiste em:

$$E_{\text{Interna}}(v) = E_{\text{Elastica}}(v) + E_{\text{Flexão}}(v) = \int_{\Omega} (w_1(s)|v'(s)|^2 + w_2(s)|v''(s)|^2) ds \quad (27)$$

ou seja, a soma das duas funções energéticas definidas nas secções precedentes.

Com esta função energética o contorno apresenta simultaneamente os comportamentos de fio elástico e de fina barra metálica, podendo a importância relativa destes dois comportamentos ser controlada utilizando os parâmetros **w1** e **w2**.

2.3.4 Forças Externas

De forma a que uma "snake" seja útil, necessitamos de funções energéticas que a atraiam para as características relevantes da imagem. A energia externa do contorno surge assim como um conjunto de funções energéticas com esse mesmo intuito. As características relevantes da imagem podem ser desde simples zonas de elevado brilho até descontinuidades no brilho da imagem (arestas).

Normalmente a definição da energia externa é efectuada em termos de potenciais definidos pelas características locais da imagem. A energia externa é desta forma definida como uma soma pesada de diversos potenciais:

$$E_{\text{Externa}}(v) = \int_{\Omega} \left(\sum_{i=1}^{np} K_i P_i(v(s)) \right) ds \quad (28)$$

As energias externas presentes na função energética apresentada são mais convenientemente manipuladas em termos de forças externas, definidas como o inverso do gradiente dos diversos potenciais:

$$F_i(v) = -\nabla P_i(v) \quad (29)$$

Ao longo desta secção serão analisadas a definição e a influência no comportamento da "snake", de quatro tipos distintos de forças externas, nomeadamente:

- Força da intensidade da imagem
- Força do gradiente da imagem
- Força do detector de arestas
- Força de enchimento do contorno

Para cada uma das forças consideradas será analisada a sua definição em termos de potencial e sua consequente transformação matemática em força, seguida por uma análise do efeito da sua inclusão no comportamento da "snake".

2.3.4.1 Força da Intensidade da Imagem

Supondo que desejamos atrair a "snake" para contornos que correspondam às zonas mais claras da imagem, a função energia poderá ser neste caso, simplesmente o simétrico da soma das intensidades dos pixels sobre os quais se situa o contorno:

$$E_{\text{Claro}}(v) = - \int_{\Omega} \text{Int}(v(s)) ds \quad (30)$$

em que "Int" representa a função brilho na imagem e $v(s)=(x(s),y(s))$ descreve a posição na imagem dos pontos constituintes do contorno.

Minimizar esta função energética, ou seja, torná-la mais negativa, corresponde a movimentar a "snake" para as zonas mais brilhantes da imagem (aquelas a que correspondem valores mais elevados da função brilho). Igual raciocínio pode ser utilizado na dedução da função intensidade com o objectivo de atrair o contorno para as zonas mais escuras da imagem (a que correspondem baixos valores da função brilho da imagem). Naturalmente que esta função será análoga à anterior mas com sinal inverso, ou seja:

$$E_{\text{Escuro}}(v) = \int_{\Omega} \text{Int}(v(s)) ds \quad (31)$$

A definição de uma função energética genérica que permita a atracção tanto para regiões claras como para regiões escuras da imagem, passa pela introdução de uma variável binária (b) que indique o sentido de atracção desejado:

$$\begin{cases} E_{\text{Intensidade}}(v) = b \int_{\Omega} \text{Int}(v(s)) ds \\ b \in \{-1, 1\} \end{cases} \quad (32)$$

Para $b=-1$ obtém-se a função 30 (E_{Claro}) e para $b=1$ obtém-se a função 31 (E_{Escuro}). Introduzindo um coeficiente de intensidade da força da imagem e discretizando a função genérica 32, obtém-se a seguinte função energética:

$$E_{\text{Intensidade}}(v) = K_{\text{Int}} * b * \sum_{i=1}^n \text{Int}(v(i)) \quad (33)$$

onde K_{Int} é o referido coeficiente (de valor positivo) que mede a intensidade da atracção e $v(i)$ representa a posição $(x(i),y(i))$ do i -ésimo ponto do contorno. Uma vez que o potencial da intensidade da imagem é:

$$P_{\text{Intensidade}}(v(s)) = \text{Int}(v(s)) \quad (34)$$

A força produzida por este potencial, apresenta uma definição extremamente simples:

$$F_{\text{Intensidade}}(v) = -\nabla \text{Int}(v) \quad (35)$$

o que pode ser discretizado, utilizando a aproximação mais simples possível para o gradiente (Figura 10), nas suas componentes **XX** e **YY** da seguinte forma:

$$\begin{cases} F_{X_{\text{Intensidade}}}(i) = -\frac{K_{\text{Int}}}{2} * b * (\text{Int}(x(i) + 1, y(i)) - \text{Int}(x(i) - 1, y(i))) \\ F_{Y_{\text{Intensidade}}}(i) = -\frac{K_{\text{Int}}}{2} * b * (\text{Int}(x(i), y(i) + 1) - \text{Int}(x(i), y(i) - 1)) \end{cases} \quad (36)$$

Supondo **b=-1** (**E_{claro}**), se o pixel na direcção crescente de **XX** é mais brilhante do que o pixel na direcção decrescente de **XX**, então o ponto de controlo é impelido no sentido positivo de **XX**, o mesmo se passando para a direcção **YY**. De forma mais resumida, o sentido da força em cada ponto de controlo (para **E_{claro}**) será idêntico ao sentido do gradiente da imagem que se verifica na região sobre a qual se encontra esse ponto.

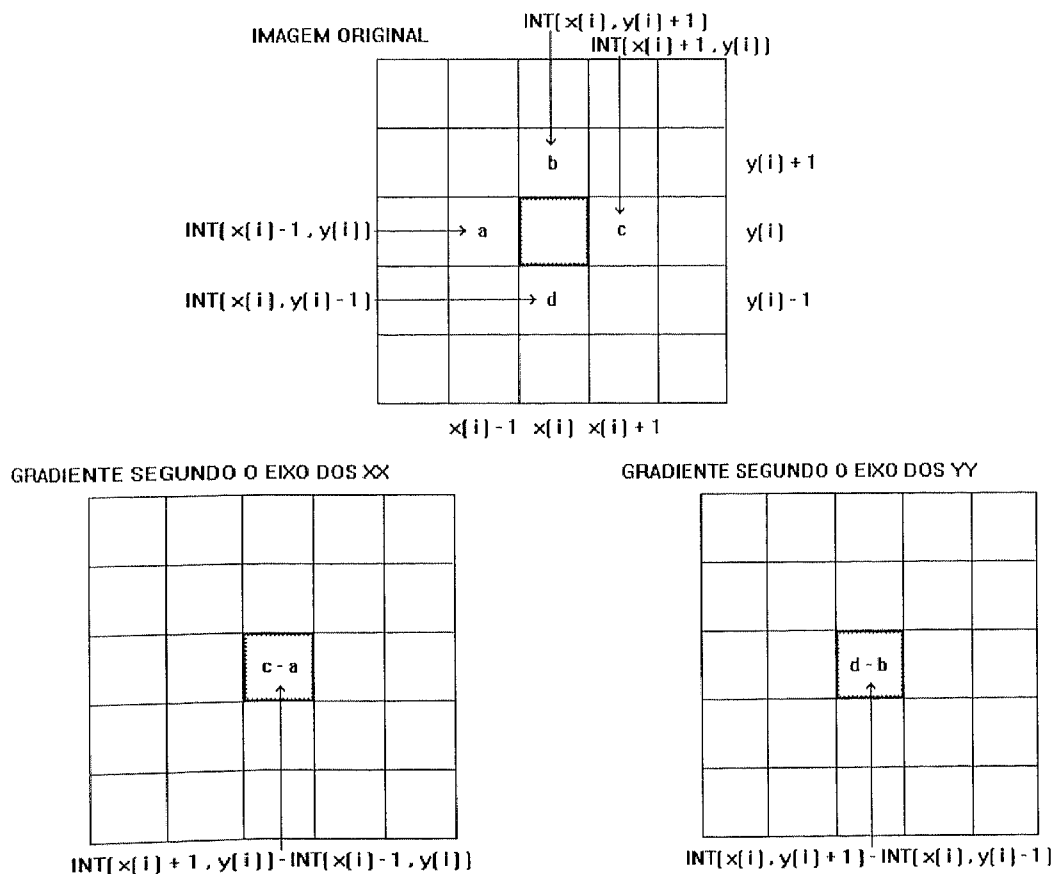


Figura 10: Aproximação do gradiente segundo os eixos **XX** e **YY**.

Na análise efectuada, outras aproximações mais robustas do gradiente da imagem poderiam ser utilizadas. No entanto, nesta fase do trabalho, tal não se apresenta ainda necessário sendo, no entanto, objecto de análise em secções posteriores.

2.3.4.2 Movimentação da "Snake" Sujeita à Força da Intensidade da Imagem

Para analisar a movimentação de uma "snake" sujeita unicamente à força da intensidade da imagem analisada na secção anterior foi criada a imagem simples apresentada na Figura 11.

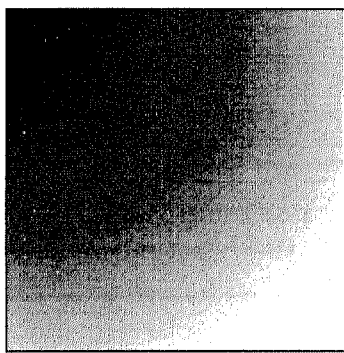
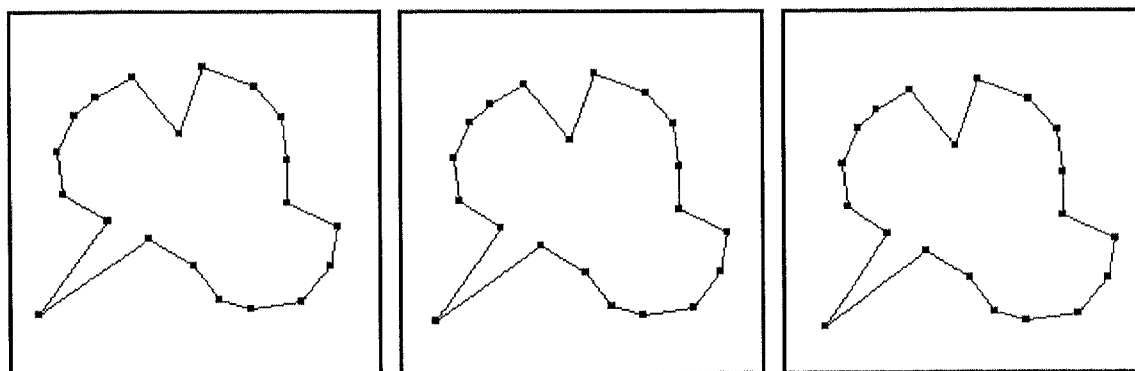


Figura 11: Imagem com gradiente constante com direcção (1,1)

A imagem criada possui um gradiente positivo constante de brilho na direcção diagonal (1,1). Desta forma, sujeitando a "snake" à força da intensidade de imagem definida anteriormente, utilizando os valores $b=-1$, $K_{int}=1$ (de forma à "snake" ser atraída para as zonas de maior luminosidade), obtemos a sequência apresentada na Figura 12.



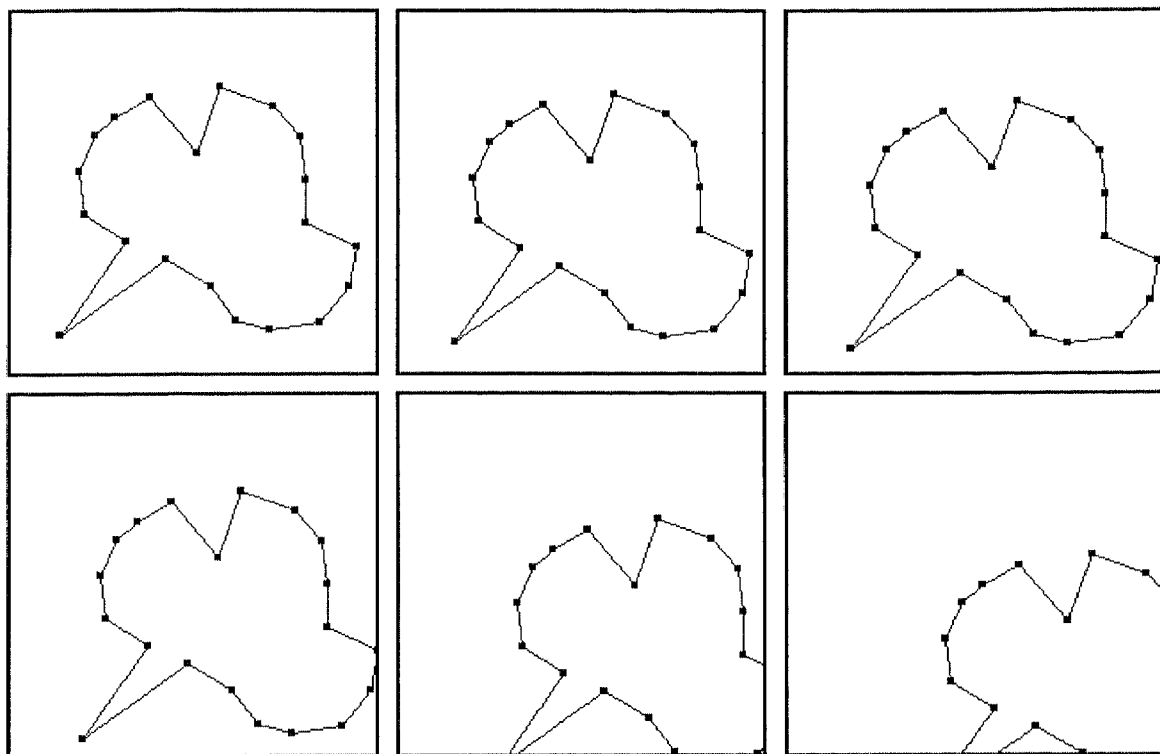


Figura 12: Evolução de um contorno sujeito unicamente à força da intensidade da imagem (iterações 0, 2, 4, 6, 8, 10, 12, 20 e 30)

Na Figura 12 é visível que o contorno, ao longo das 20 iterações, se desloca com velocidade constante na direcção $(1,1)$, ou seja, no sentido positivo dos eixos XX e YY . Todos os pontos do contorno sofrem em cada instante uma força idêntica (proporcional ao gradiente da imagem), não havendo desta forma lugar para qualquer deformação do contorno. O contorno sofre assim uma translação simples ao longo do processo iterativo na direcção da região mais brilhante da imagem correspondente ao canto inferior direito.

Em imagens reais, este tipo de força não apresenta grande utilidade dada a forma como foi definido o gradiente. A "snake", "vê" unicamente uma vizinhança de um pixel em redor dos seus pontos de controlo, estando desta forma extremamente dependente do ruído e de características locais da imagem. De forma a evitar uma grande dependência local e a oferecer ao contorno activo a possibilidade de encontrar as características desejadas na imagem (mesmo que estas não se encontrem muito próximas dos seus pontos de controlo iniciais), "snakes" mais poderosas utilizam outras definições da função gradiente. Definições alternativas consideram uma vizinhança maior do ponto de controlo no cálculo do gradiente ou efectuem uma suavização prévia (por exemplo Gaussiana) da imagem original.

2.3.4.3 Força do Gradiente da Imagem

Na análise de imagens reais um dos objectivos mais frequentes corresponde à detecção e quantificação do contorno de objectos nela incluídos. Os contornos de um objecto correspondem normalmente a regiões com declives acentuados de variação do brilho da imagem (gradientes elevados). Fica assim justificado o interesse de uma energia que seja inversamente proporcional à intensidade do gradiente da imagem. Uma definição simples para uma função energética deste tipo é a seguinte:

$$E_{\text{Gradiente}}(v) = - \int_{\Omega} |\nabla \text{Int}(v(s))| ds \quad (37)$$

obtendo assim para o potencial a definição:

$$P_{\text{Gradiente}}(v(s)) = -|\nabla \text{Int}(v(s))| \quad (38)$$

e para a força do gradiente num determinado ponto, a expressão:

$$F_{\text{Gradiente}}(v(s)) = |\nabla^2 \text{Int}(v(s))| \quad (39)$$

Como foi analisado anteriormente, é conveniente espalhar o efeito das forças da imagem a regiões adjacentes de forma a que cada ponto da "snake" possa "ver" uma maior parcela da estrutura da imagem. Seguindo este raciocínio é comum ser utilizada uma força do gradiente suavizada através da convolução prévia da imagem com um núcleo Gaussiano (de dimensão n):

$$F_{\text{Gradiente}}(s) = |\nabla^2 G(n) \otimes \text{Int}(v(s))| \quad (40)$$

Dependendo do grau de suavização será possível obter uma maior ou menor dispersão das forças do gradiente na imagem. A uma maior (menor) suavização da imagem, corresponderá uma maior (menor) suavização do gradiente e consequentemente uma força do gradiente mais (menos) dispersa. A suavização da imagem original implica no entanto uma redução da precisão obtida nos resultados finais devido à diminuição que tal implica na precisão de detecção das zonas que correspondem aos máximos locais da função gradiente.

2.3.4.4 Movimentação da "Snake" Sujeita à Força do Gradiente da Imagem

O efeito da força externa resultante do gradiente da imagem é o de atrair o contorno para as regiões da imagem onde se verificam maiores variações na intensidade luminosa dos pixels, ou seja, as arestas dos objectos. Um contorno sujeito unicamente a uma força deste tipo, verá os seus pontos serem atraídos para os máximos locais da função gradiente que se encontram mais próximos e simultaneamente dentro da distância de "visão" do contorno.

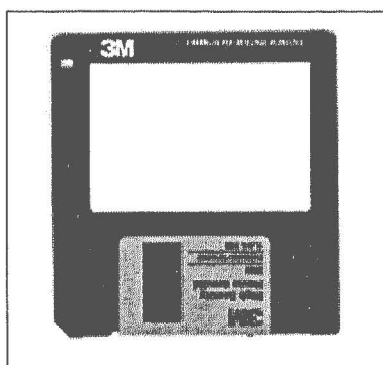


Figura 13: Imagem a 256 níveis de cinzento contendo uma disquete.

De forma a analisar a forma e influência deste tipo de forças, atente-se na imagem apresentada na Figura 13 Calculando o gradiente da imagem segundo os eixos **XX** e **YY** (utilizando um dos métodos descritos nas secções precedentes), obtém-se as imagens gradiente apresentadas na Figura 14.

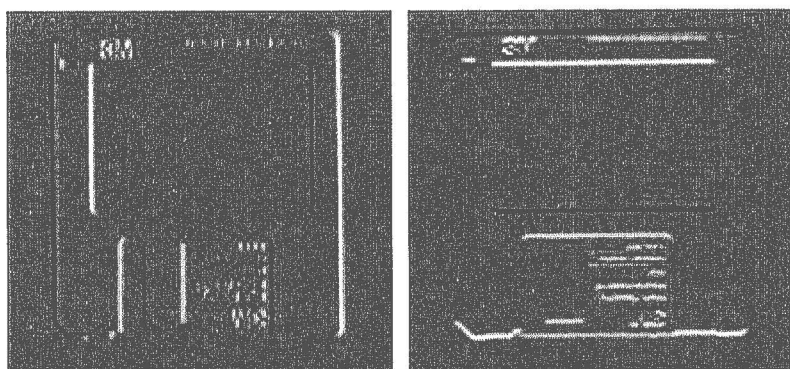


Figura 14: Gradientes segundo os eixos dos **XX** e **YY** da imagem exemplo.

É visível que as zonas correspondentes a aumentos no brilho da imagem original apresentam na imagem gradiente uma cor clara (zona de elevado gradiente), verificando-se exactamente o oposto para as zonas correspondentes a decréscimos no brilho da

imagem original. As zonas de brilho constante da imagem original, apresentam na imagem gradiente, uma cor cinzenta intermédia correspondente ao nível de cinzento 127 (metade da escala utilizada).

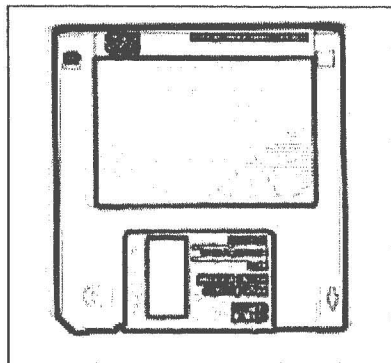


Figura 15: Inverso do módulo do gradiente da imagem exemplo.

A imagem correspondente ao inverso do módulo do gradiente para a imagem exemplo, encontra-se representada na Figura 15. Esta imagem, para maior facilidade de visualização foi invertida, correspondendo as regiões mais escuras aos valores mais elevados do módulo do gradiente e vice-versa. Na imagem é visível que as regiões correspondentes às arestas, apresentam elevados valores de gradiente, correspondendo consequentemente esta imagem a um excelente potencial para ser incluído na função energética externa do contorno. Calculando as forças segundo os eixos dos **XX** e **YY** correspondentes a essa inclusão na função energética obtém-se as imagens apresentadas na Figura 16.

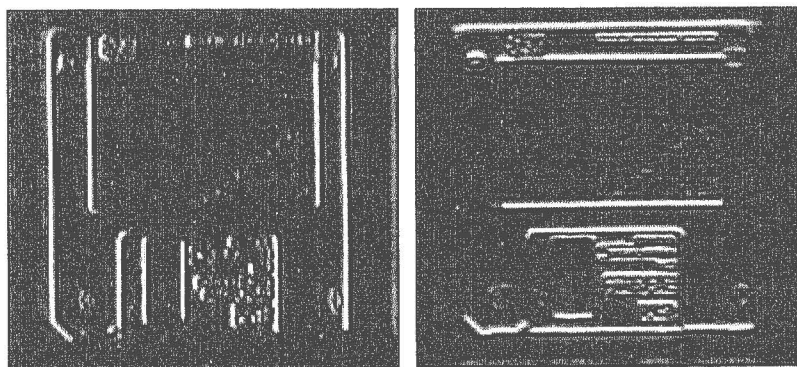


Figura 16: Forças do Gradiente da Imagem segundo os eixos dos XX e YY

Na imagem correspondente ao gradiente segundo o eixo dos **XX** (esquerda), verifica-se que valores elevados (claros) de brilho correspondem a forças no sentido positivo deste eixo (da esquerda para a direita) enquanto que valores reduzidos (escuras)

correspondem a forças opostas a estas. Situação análoga verifica-se na imagem contendo a força do gradiente da imagem segundo o eixo dos YY onde, valores claros correspondem a forças no sentido positivo do eixo (para baixo) e valores escuros a forças no sentido negativo (para cima).

2.3.4.5 Força do Detector de Arestas

A utilização de uma força calculada a partir de um mapa de arestas extraído utilizando um detector de arestas convencional, permite combinar a qualidade de um bom detector de arestas com o modelo dos contornos activos. Como será analisado em secções posteriores deste trabalho, esta associação permite em muitas imagens, obter resultados de melhor qualidade, do que se obteriam utilizando unicamente as forças descritas anteriormente.

A força do detector de arestas corresponde normalmente a um potencial obtido utilizando um detector de arestas convencional (Canny, Deriche, Shen e Castan, Spacek, Sarkar e Boyer, etc.). Partindo da imagem de arestas obtida, são eliminadas as arestas mais fracas e efectuada uma normalização para o intervalo **[0,1]**. A imagem obtida contém um potencial simulado cujo valor é elevado nos pontos correspondentes às arestas e reduzido nos restantes pontos.

$$P_{\text{Arestas}}(v(s)) = |Int_{\text{Arestas}}(v(s))| \quad (41)$$

A imagem potencial é depois convoluída com um núcleo Gaussiano (de forma a obter um potencial menos concentrado) após o que, é calculado o seu gradiente de forma a obter as imagens força segundo ambos os eixos:

$$F_{\text{Arestas}}(s) = |\nabla G(n) \otimes Int_{\text{Arestas}}(v(s))| \quad (42)$$

Alterando os parâmetros do detector de arestas utilizado é possível modificar a resolução de detecção das arestas. Este procedimento permite utilizar a força do detector de arestas para controlar o tipo de contorno a detectar: Contorno forte (baixa resolução no detector de arestas) ou contorno fraco (elevada resolução no detector de arestas).

Modificando a dimensão do núcleo Gaussiano a convoluir com a imagem resultante do detector de arestas é possível estabelecer um compromisso entre a precisão e o espalhamento das forças resultantes. Maiores núcleos Gaussianos, conduzem a uma maior suavização da imagem força e consequentemente a um maior

espalhamento das forças. Menores núcleos conduzem a uma localização mais precisa das forças, situando-se estas unicamente em regiões muito próximas das arestas detectadas.

2.3.4.6 Movimentação da "Snake" Sujeita à Força do Detector de Arestas

O efeito da força do detector de arestas é naturalmente o de "puxar" o contorno para as regiões onde se encontram as arestas extraídas pelo detector de arestas.

De forma a analisar o efeito da força do detector de arestas no movimento da "snake", é utilizada, como exemplo, a imagem da Figura 13. Calculando a força do detector de arestas através do filtro de Shen e Castan com **alfa=2.0**, os resultados obtidos para as imagens de potenciais e de forças são apresentados nas Figuras 17 e 18.

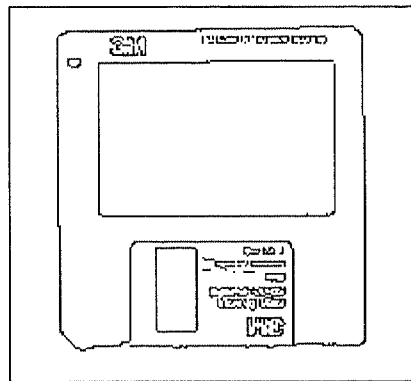


Figura 17: Imagem potencial resultante da aplicação do detector de arestas de Shen e Castan (alfa=2.0) seguida de limiarização.

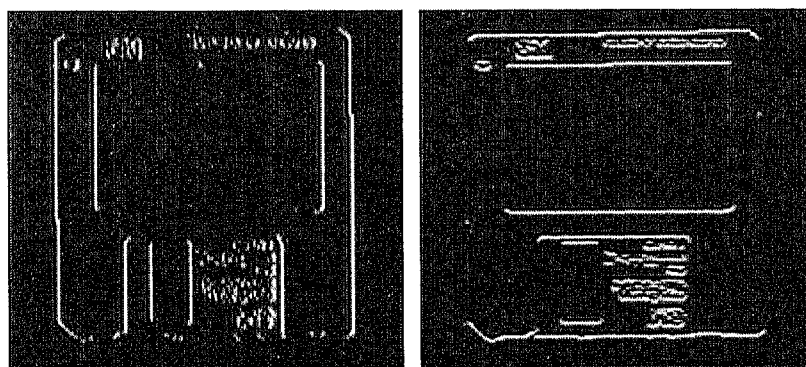


Figura 18: Força do detector de arestas segundo os eixos XX e YY.

É visível que o detector de arestas utilizado permite extrair com razoável grau de precisão as arestas da disquete contida na imagem. No entanto, a utilização da imagem

contendo as forças resultantes não será de grande utilidade a não ser que a definição inicial do contorno se encontre extremamente próxima da localização real do contorno. Esta observação pode ser comprovada experimentalmente utilizando o contorno inicial apresentado na Figura 19.

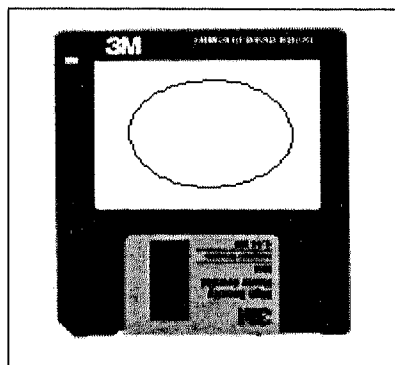


Figura 19: Contorno inicial que permanece imóvel ao longo do processo iterativo quando sujeito unicamente à força do detector de arestas.

A utilização unicamente da força do detector de arestas, tal como foi anteriormente definida, não resulta em qualquer movimentação da "snake" uma vez que esta se encontra demasiado distante da localização das forças contidas na imagem. Desta forma, o contorno permanece imóvel ao longo do processo iterativo.

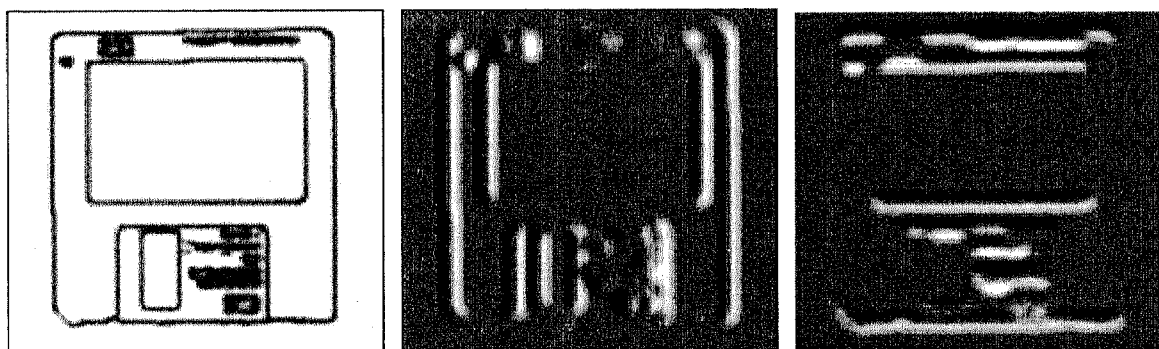


Figura 20: Resultados suavizados do detector de arestas e forças segundo os eixos.

Uma solução passa por suavizar previamente a imagem de forças do detector de arestas, utilizando simultaneamente as forças de elasticidade e flexão. Desta forma, desde que a definição inicial do contorno possua pontos em contacto com zonas onde a força do detector de arestas assuma valores relevantes, esses pontos serão "puxados", encarregando-se as forças internas do contorno do resto do trabalho.

2.3.4.7 Força de Enchimento do Contorno

Se um contorno estiver sujeito unicamente às suas forças internas, a sua elasticidade irá fazer com que ele se contraia e, no limite, se reduza a um único ponto [Cohen91]. Explorando mais profundamente este raciocínio, se o contorno estiver sujeito não só às suas forças internas, como também às forças externas do gradiente e detector de arestas, a contracção pode-se verificar de forma muito semelhante, desde que o contorno se situe em regiões de brilho constante.

As forças do gradiente e detector de arestas são normalmente muito fracas em grande parte da superfície da imagem, apresentando valores fortes unicamente nas zonas correspondentes a contornos de objectos. Um contorno activo, se inicializado relativamente longe da solução desejada, tenderá na grande maioria dos casos a ser atraído por uma parcela reduzida do contorno, reduzindo-se em seguida devido às suas forças internas. Exemplos deste procedimento podem ser analisados na Figura 21 onde as inicializações efectuadas para o contorno não conduzem ao resultado desejado, ou seja, à detecção da mancha da imagem.

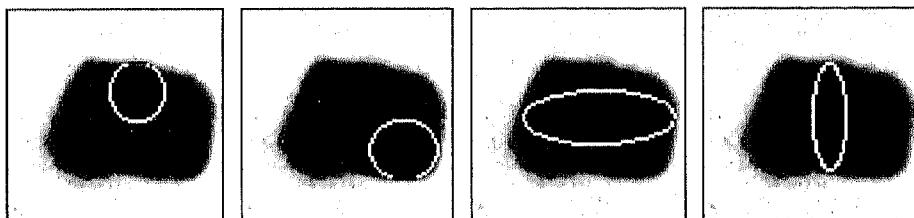


Figura 21: Exemplos de inicializações do contorno que não conduzem à solução correcta.

Exceptuando as situações em que o contorno inicial está muito próximo (Figura 22) da solução desejada, os resultados obtidos não são correctos. Uma solução para este problema, como foi analisado anteriormente, consiste na suavização das imagens contendo as forças do gradiente da imagem e do detector de contornos. No entanto esta solução para além de nem sempre ser eficaz, pode também conduzir à obtenção de contornos finais muito pouco precisos.

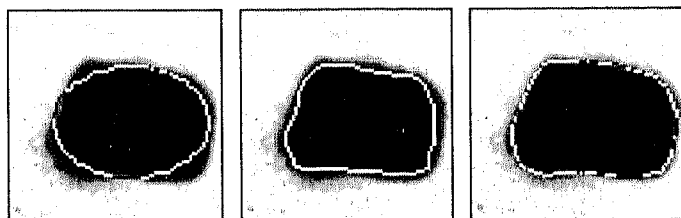


Figura 22: Obtenção de uma solução correcta através de inicializações próximas do contorno desejado e respectivos resultados (direita).

A melhor solução para o problema descrito foi proposta por Laurent Cohen [Cohen91, Cohen93] e adaptada por outros autores [Maurincomme93, Maurincomme94]. Consiste na utilização de uma força de enchimento que funciona como uma pressão interna. O seu efeito faz com que o contorno se comporte como um balão que é insuflado. Ao passar pelas regiões onde se encontram arestas, o movimento do contorno pode ser travado, se a aresta for forte, ou continuado o enchimento, se a aresta for fraca relativamente à força insufladora.

Os principais benefícios da força de enchimento são o aumento da insensibilidade da solução final, no que diz respeito às condições iniciais, e a diminuição da probabilidade do contorno ser travado por pontos isolados de elevado gradiente, mas que na realidade constituem falsas arestas.

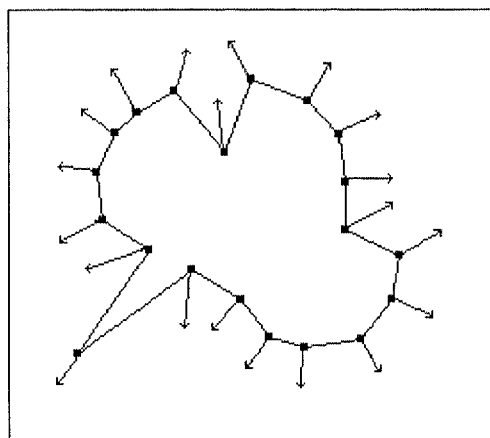


Figura 23: Força de enchimento

A definição matemática para a força de enchimento é simplesmente um vector normal ao contorno em toda a sua extensão (Figura 23), ou seja:

$$\mathbf{F}_{\text{Enchimento}} = K_{\text{Ench}} \vec{n}(v(s)) \quad (43)$$

onde K_{Ench} é um coeficiente indicativo da intensidade desta força de pressão e $\vec{n}(v(s))$ é um vector unitário normal ao contorno no ponto $v(s)$. Se o coeficiente K_{Ench} tiver sinal negativo, a força de enchimento transforma-se numa força de deflação em vez de insuflação.

2.3.4.8 Movimentação da "snake" Sujeita à Força de Enchimento do Contorno

A força de enchimento transforma o movimento do contorno num movimento tipo balão que é insuflado até encontrar forças suficientemente fortes da imagem que o façam

parar. Se for utilizada unicamente a força de enchimento (sem quaisquer outras forças internas ou externas), o resultado obtido consiste no deslocamento de todos os pontos do contorno, em direcção normal a este como pode ser visualizado na Figura 24.

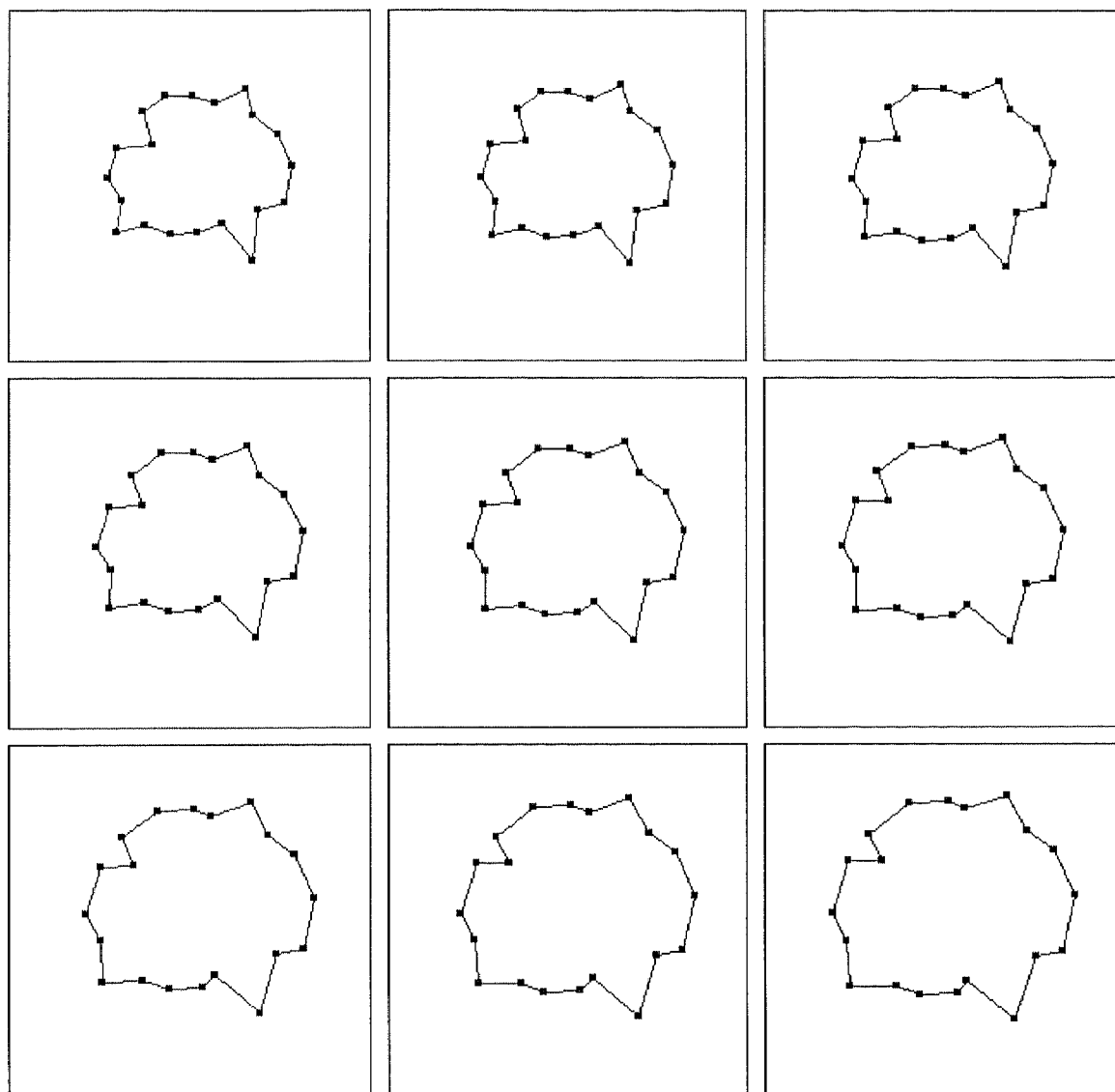


Figura 24: Movimentação de um contorno sujeito unicamente à Força de Enchimento (iterações 0, 2, 4, 6, 8, 10, 12, 14, 16)

O único efeito da Força de Enchimento consiste numa progressiva abertura do contorno que só será travada pela presença de outras forças contrárias tais como as Forças Internas, a Força do Gradiente ou a Força do Detector de Arestas. Esta situação é extremamente positiva desde que a "snake" seja inicializada no interior do contorno a detectar. Neste caso, a Força de Enchimento encarrega-se de ir abrindo progressivamente o contorno até encontrar as regiões com elevadas Forças da Imagem

que se encarregam de o parar. Desta forma é possível obter bons resultados finais utilizando contornos iniciais muito mais simples e mais distantes da solução desejada.

2.4 SOLUÇÃO NUMÉRICA

Embora ao longo das secções precedentes, a discretização das diversas equações e a forma de obtenção das equações de evolução tenha já sido abordada, ao longo desta secção, estes tópicos serão abordados com maior detalhe sendo dada uma maior relevância aos métodos numéricos utilizados.

2.4.1 Formulação Matemática da Energia Total do Contorno

Utilizando a formulação matemática definida ao longo da secção 2.3 para a função energética da "snake" e considerando **Ad** como o espaço de deformações admissíveis para o contorno, temos:

$$\begin{aligned} E: \text{Ad} &\rightarrow \mathbb{R} \\ v &\rightarrow E(v) = E_{\text{Interna}}(v) + E_{\text{Externa}}(v) = \\ &= \int_{\Omega} (w1(s)|v'(s)|^2 + w2(s)|v''(s)|^2 + P_{\text{Externo}}(v(s)))ds \end{aligned} \quad (44)$$

A energia externa do contorno resulta da integração do potencial externo que por sua vez pode ser definido como uma soma pesada dos potenciais externos de Intensidade da Imagem, do Gradiente da Imagem, do Detector de Contornos e de Enchimento.

2.4.2 Discretização da Função Energética

Se os coeficientes de elasticidade e de rigidez (**w1(s)** e **w2(s)**) forem constantes ao longo de todo o contorno, ou seja, se o contorno tiver elasticidade e rigidez uniformes, é bastante simples discretizar directamente a equação 44:

$$E(v) = \sum_{i=1}^n (E_{\text{Interna}}(i) + E_{\text{Externa}}(i)) \quad (45)$$

onde **n** representa o número total de pontos em que o contorno foi discretizado.

Aproximando as derivadas com diferenças finitas [Conte81, Segerlind84], podemos decompor o termo **E_{Interna}**, obtendo a definição:

$$\begin{cases} E_{\text{Interna}}(v) = \sum_{i=1}^n \left(w1 \frac{|v(i) - v(i-1)|}{2h^2} + w2 \frac{|v(i-1) - 2v(i) + v(i+1)|}{2h^4} \right) \\ h = 1/n \end{cases} \quad (46)$$

onde h representa o espaço de discretização e são impostas as condições fronteira $v(0)=v(n)$ e $v(n+1)=v(1)$.

Como a energia externa do contorno apresenta um formato bastante variável e é bastante mais fácil de trabalhar em termos de força (por exemplo não faz sentido falar de uma Energia de Enchimento), a sua discretização no espaço será efectuada unicamente quando for analisada a resolução da equação de evolução do contorno.

2.4.3 Equação de Evolução

Para minimizar a função energética 44, um contorno $v(s)$ terá que ser um mínimo local da função $E(v)$, satisfazendo desta forma a equação Euler-Lagrange associada:

$$\begin{cases} -\frac{\partial(w1v')}{\partial s} + \frac{\partial^2(w2v'')}{\partial s^2} + \nabla P_{\text{Externo}}(v) = 0 \\ v(0), v'(0), v(1) \text{ e } v'(1) \text{ conhecidos} \end{cases} \quad (47)$$

Nesta formulação, cada termo aparece como uma força que é aplicada ao contorno. A equação 47, equivale a dizer que o contorno se encontra em equilíbrio, ou seja, que o somatório das forças que lhe são aplicadas tem um valor nulo:

$$-\frac{\partial(w1v')}{\partial s} + \frac{\partial^2(w2v'')}{\partial s^2} = F_{\text{Externa}}(v) \quad (48)$$

2.4.4 Discretização das Equações no Espaço

Discretizando no espaço a equação 48 utilizando diferenças finitas (com passo $h=1/n$), obtêm-se as seguintes equações:

$$\begin{aligned}
& \frac{k1}{h} \left((v(i) - v(i-1)) + (v(i+1) - v(i)) \right) \\
& + \frac{k2}{h^2} \left((v(i-2) - 2v(i-1) + v(i)) - 2(v(i-1) - 2v(i) + v(i+1)) \right) \\
& + (v(i) - 2v(i+1) + v(i+2)) \Big) \\
& - F_{\text{Externa}}(v(i)) = 0
\end{aligned} \tag{49}$$

onde:

$$\begin{cases}
h = \frac{1}{n} \\
k1 = \frac{w1}{h} \\
k2 = \frac{w2}{h^2} \\
v(i) = (x(i), y(i))
\end{cases} \tag{50}$$

As equações de Euler apresentadas, podem ser rescritas na forma matricial:

$$A \cdot V = F_{\text{Externa}} \tag{51}$$

onde a matriz **A** apresenta um formato pentadiagonal simétrico, ou seja:

$$\begin{bmatrix}
a1 & a2 & a3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
a2 & a1 & a2 & a3 & 0 & 0 & 0 & 0 & 0 & 0 \\
a3 & a2 & a1 & a2 & a3 & 0 & 0 & 0 & 0 & 0 \\
0 & a3 & a2 & a1 & a2 & a3 & 0 & 0 & 0 & 0 \\
\cdot & \cdot & & & & & \cdot & \cdot & & \\
\cdot & \cdot & & & & & \cdot & \cdot & & \\
\cdot & \cdot & & & & & \cdot & \cdot & & \\
0 & 0 & 0 & 0 & a3 & a2 & a1 & a2 & a3 & 0 \\
0 & 0 & 0 & 0 & 0 & a3 & a2 & a1 & a2 & a3 \\
0 & 0 & 0 & 0 & 0 & 0 & a3 & a2 & a1 & a2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & a3 & a2 & a1
\end{bmatrix}
\begin{bmatrix}
v(1) \\
v(2) \\
v(3) \\
v(4) \\
\cdot \\
\cdot \\
\cdot \\
v(n-3) \\
v(n-2) \\
v(n-1) \\
v(n)
\end{bmatrix}
=
\begin{bmatrix}
(F_x(v(1)), F_y(v(1))) \\
(F_x(v(2)), F_y(v(2))) \\
(F_x(v(3)), F_y(v(3))) \\
(F_x(v(4)), F_y(v(4))) \\
\cdot \\
\cdot \\
\cdot \\
(F_x(v(n-3)), F_y(v(n-3))) \\
(F_x(v(n-2)), F_y(v(n-2))) \\
(F_x(v(n-1)), F_y(v(n-1))) \\
(F_x(v(n)), F_y(v(n)))
\end{bmatrix} \tag{52}$$

em que:

$$\begin{cases} a1 = \frac{w2}{h^4} \\ a2 = -\left(\frac{w1}{h^2} + 4\frac{w2}{h^4}\right) \\ a3 = \left(2\frac{w1}{h^2} + 6\frac{w2}{h^4}\right) \end{cases} \quad (53)$$

De forma a obter um contorno fechado, será necessário alterar a equação 52, impondo um conjunto de condições fronteira. Como as aproximações efectuadas na discretização das equações, utilizam no cálculo de um determinado ponto unicamente os seus quatro vizinhos mais próximos, as condições fronteira a impor, de forma a garantir um contorno fechado, serão as seguintes:

$$\begin{cases} v(-1) = v(n-1) \\ v(0) = v(n) \\ v(n+1) = v(1) \\ v(n+2) = v(2) \end{cases} \quad (54)$$

ou seja, o vizinho direito ($v(0)$) do ponto $v(1)$ é o ponto $v(n)$, e o seu vizinho direito ($v(-1)$) é por sua vez o ponto $v(n-1)$. Por outro lado, o vizinho esquerdo ($v(n+1)$) do ponto $v(n)$ corresponde ao ponto $v(1)$ e o seu vizinho esquerdo ($v(n+2)$) será $v(2)$. Na Figura 25 são mais facilmente visíveis as condições fronteira consideradas.

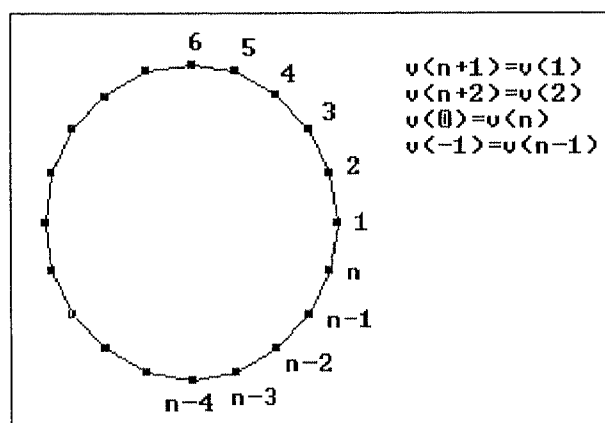


Figura 25: Condições fronteira para tornar o contorno fechado.

Utilizando as condições fronteira definidas é possível [Kass88, Gladwell79] obter uma nova definição para a equação 52:

$$\begin{bmatrix} a1 & a2 & a3 & 0 & 0 & 0 & \dots & 0 & a3 & a2 \\ a2 & a1 & a2 & a3 & 0 & 0 & \dots & 0 & 0 & a3 \\ a3 & a2 & a1 & a2 & a3 & 0 & \dots & 0 & 0 & 0 \\ 0 & a3 & a2 & a1 & a2 & a3 & \dots & 0 & 0 & 0 \\ \cdot & \cdot & & & & & & \cdot & \cdot & \\ \cdot & \cdot & & & & & & \cdot & \cdot & \\ \cdot & \cdot & & & & & & \cdot & \cdot & \\ 0 & 0 & 0 & \dots & a3 & a2 & a1 & a2 & a3 & 0 \\ 0 & 0 & 0 & \dots & 0 & a3 & a2 & a1 & a2 & a3 \\ a3 & 0 & 0 & \dots & 0 & 0 & a3 & a2 & a1 & a2 \\ a2 & a3 & 0 & \dots & 0 & 0 & 0 & a3 & a2 & a1 \end{bmatrix} \begin{bmatrix} v(1) \\ v(2) \\ v(3) \\ v(4) \\ \cdot \\ \cdot \\ \cdot \\ v(n-3) \\ v(n-2) \\ v(n-1) \\ v(n) \end{bmatrix} = \begin{bmatrix} (F_x(v(1)), F_y(v(1))) \\ (F_x(v(2)), F_y(v(2))) \\ (F_x(v(3)), F_y(v(3))) \\ (F_x(v(4)), F_y(v(4))) \\ \cdot \\ \cdot \\ \cdot \\ (F_x(v(n-3)), F_y(v(n-3))) \\ (F_x(v(n-2)), F_y(v(n-2))) \\ (F_x(v(n-1)), F_y(v(n-1))) \\ (F_x(v(n)), F_y(v(n))) \end{bmatrix} \quad (55)$$

Os valores $F_x(v(i))$ e $F_y(v(i))$ correspondem ao somatório de todas as forças externas aplicadas, decomposto nas suas componentes segundo os eixos **XX** e **YY**.

2.4.5 Equação de Evolução e sua Discretização no Tempo

A equação de evolução para o problema apresentado, assume o seguinte formato:

$$\begin{cases} \frac{\partial v}{\partial t} - \frac{\partial(w1v')}{\partial s} + \frac{\partial^2(w2v'')}{\partial s^2} = F_{\text{Externa}}(v) \\ v(t,0) = v(t,1) \\ v'(t,0) = v'(t,1) \\ v(0,s) = v_0(s) \end{cases} \quad (56)$$

Dado um determinado contorno inicial ($v(0,s)=v_0(s)$) e impostas as condições fronteira de forma a tornar o contorno fechado ($v(t,0)=v(t,1)$ e $v'(t,0)=v'(t,1)$), a variação na posição de cada ponto vai depender do somatório das forças internas e externas

aplicadas nesse ponto. No limite é obtida a solução para o problema estático quando o termo $d\mathbf{v}/dt$ tende para 0 e a solução $\mathbf{v}(t)$ estabiliza.

Utilizando a notação matricial, esta equação de evolução, pode ser obtida a partir da equação:

$$\mathbf{A}\mathbf{V} - \mathbf{F}_{\text{Externa}} = 0 \quad (57)$$

simplesmente colocando o lado direito da equação igual à multiplicação de um passo temporal pela derivada do lado esquerdo. No entanto, devido ao facto de que a utilização da derivada das forças externas implicaria a alteração da matriz $\mathbf{A}$ em cada iteração, iterações mais rápidas são obtidas assumindo que a força externa permanece constante ao longo de cada intervalo de tempo. Utilizando diferenças finitas [Conte81] no tempo (passo τ) e no espaço (passo h), obtém-se:

$$\mathbf{A}\mathbf{V}(t) - \mathbf{F}_{\text{Externa}}(\mathbf{V}(t-1)) = -\frac{1}{\tau}(\mathbf{V}(t) - \mathbf{V}(t-1)) \quad (58)$$

$\Leftrightarrow$

$$(\mathbf{I}_d + \tau\mathbf{A})\mathbf{V}(t) = \mathbf{V}(t-1) + \tau\mathbf{F}_{\text{Externa}}(\mathbf{V}(t-1)) \quad (59)$$

onde $\mathbf{I}_d$ corresponde à matriz identidade. Resolvendo em ordem a $\mathbf{V}(t)$, obtemos o seguinte esquema iterativo:

$$\mathbf{v}(t) = (\mathbf{I}_d + \tau\mathbf{A})^{-1}(\mathbf{v}(t-1) + \tau\mathbf{F}_{\text{Externa}}(\mathbf{v}(t-1))) \quad (60)$$

que pode ser decomposto nas suas componentes segundo os eixos dos $\mathbf{XX}$ e $\mathbf{YY}$:

$$\begin{cases} x(t) = (\mathbf{I}_d + \tau\mathbf{A})^{-1}(x(t-1) + \tau\mathbf{F}_{\text{Externa}}(v(t-1))) \\ y(t) = (\mathbf{I}_d + \tau\mathbf{A})^{-1}(y(t-1) + \tau\mathbf{F}_{\text{Externa}}(v(t-1))) \end{cases} \quad (61)$$

onde a matriz $(\mathbf{I}_d + \tau\mathbf{A})$ é pentadiagonal sendo facilmente invertida através de uma decomposição LU [Chapra85, Press92]. Utilizando as condições fronteira definidas de forma a garantir que o contorno seja fechado, obtém-se a seguinte definição para a matriz $(\mathbf{I}_d + \tau\mathbf{A})$:

$$\begin{bmatrix}
 1 + \tau a_1 & \tau a_2 & \tau a_3 & 0 & 0 & 0 & \dots & 0 & \tau a_3 & \tau a_2 \\
 \tau a_2 & 1 + \tau a_1 & \tau a_2 & \tau a_3 & 0 & 0 & \dots & 0 & 0 & \tau a_3 \\
 \tau a_3 & \tau a_2 & 1 + \tau a_1 & \tau a_2 & \tau a_3 & 0 & \dots & 0 & 0 & 0 \\
 0 & \tau a_3 & \tau a_2 & 1 + \tau a_1 & \tau a_2 & \tau a_3 & \dots & 0 & 0 & 0 \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
 0 & 0 & 0 & \dots & \tau a_3 & \tau a_2 & 1 + \tau a_1 & \tau a_2 & \tau a_3 & 0 \\
 0 & 0 & 0 & \dots & 0 & \tau a_3 & \tau a_2 & 1 + \tau a_1 & \tau a_2 & \tau a_3 \\
 \tau a_3 & 0 & 0 & \dots & 0 & 0 & \tau a_3 & \tau a_2 & 1 + \tau a_1 & \tau a_2 \\
 \tau a_2 & \tau a_3 & 0 & \dots & 0 & 0 & 0 & \tau a_3 & \tau a_2 & 1 + \tau a_1
 \end{bmatrix} \quad (62)$$

Uma vez calculada a inversa da matriz $(I_d + \tau A)$, supondo que a rigidez e a elasticidade do contorno se mantêm constantes ao longo do processo, o método de resolução baseia-se unicamente no cálculo sucessivo das posições dos pontos $(\mathbf{x}(t))$ e $\mathbf{y}(t)$ a partir dos pontos da iteração anterior $(\mathbf{x}(t-1))$ e $\mathbf{y}(t-1))$ e das forças externas que se verificam nas regiões onde se encontram esses pontos $(F_x(\mathbf{x}(t-1), \mathbf{y}(t-1)))$ e $F_y(\mathbf{x}(t-1), \mathbf{y}(t-1))$). O método iterativo é finalizado quando a diferença verificada na posição dos pontos entre iterações sucessivas assume um valor inferior a um determinado limiar pré-definido.

2.5 PROBLEMAS COMUNS DE IMPLEMENTAÇÃO

A implementação do método dos contornos activos seguindo a formulação apresentada leva a diversos problemas, destacando-se essencialmente cinco:

- Instabilidade devido à discretização no tempo.
- Instabilidade devido à discretização no espaço.
- Dificuldade na inicialização da "snake".
- Dificuldade na detecção de contornos afastados da solução inicial.
- Selecção e optimização dos coeficientes de rigidez e elasticidade.

Ao longo desta secção serão analisados e discutidos os problemas apresentados e propostos alguns métodos para a sua resolução.

2.5.1 Instabilidade Devido à Discretização no Tempo

As forças externas, resultantes do gradiente da imagem ou do detector de arestas, foram definidas em secções precedentes da seguinte forma:

$$F(v) = -\nabla P(v) \quad (63)$$

No caso da Força do Gradiente da Imagem, a função potencial corresponde ao inverso do módulo do gradiente da imagem. Sendo assim, esta força tenderá a atrair o contorno para as zonas que correspondem a mínimos da função potencial, ou seja, para as zonas de mais elevado gradiente.

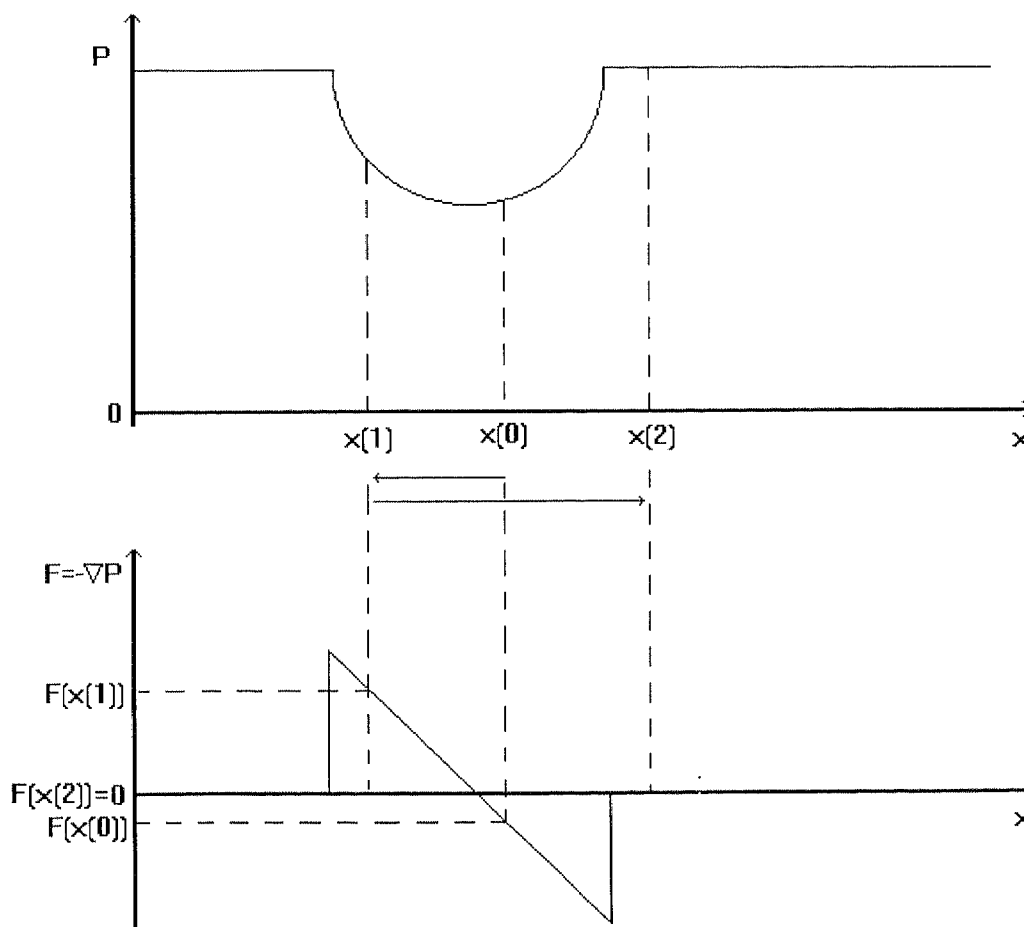


Figura 26: Instabilidades devido à discretização no tempo.

Para a Força do Detector de Arestas, a função potencial corresponde ao resultado (após filtragem) obtido utilizando um determinado detector de arestas. Neste caso, os mínimos da função potencial correspondem aos pontos onde foram encontradas

arestas pelo detector, sendo o contorno atraído para as regiões correspondentes a esses pontos.

Em ambos os casos, a direcção da força $\mathbf{F}$ é sempre no sentido descendente da função potencial e as regiões para as quais o contorno é atraído correspondem aos mínimos das respectivas funções potencial. O equilíbrio é atingido nas regiões onde $\mathbf{P}$ é mínimo na direcção normal ao contorno, correspondendo estes aos cruzamento por zero da função força da imagem.

Mesmo que o contorno inicial se situe perto da solução desejada, podem ocorrer instabilidades devido à discretização no tempo e no espaço do problema de evolução. Em cada iteração, o efeito das forças externas é o de deslocar cada ponto do contorno ao longo do vector $\tau \mathbf{F}_{\text{externa}}(\mathbf{v}(t-1))$. Se $\tau \mathbf{F}_{\text{externa}}(\mathbf{v}(t-1))$ for muito elevado, um ponto pode ser movimentado para longe do mínimo local desejado e nunca regressar (Figura 26).

Na figura é visível que o valor elevado de τ leva a que o ponto $\mathbf{x}(0)$ seja movimentado sucessivamente para $\mathbf{x}(1)=\mathbf{x}(0)+\tau \mathbf{F}(\mathbf{x}(0))$ e $\mathbf{x}(2)=\mathbf{x}(1)+\tau \mathbf{F}(\mathbf{x}(1))$. Em $\mathbf{x}(2)$, uma vez que $\mathbf{F}(\mathbf{x}(2))$ é nula (pois o potencial aqui é constante), o ponto estabiliza. Desta forma, devido a um passo temporal demasiado elevado, o mínimo local da função potencial não foi encontrado.

Uma solução para o problema apresentado consiste na utilização de passos temporais reduzidos, por exemplo fazendo com que $\tau \mathbf{F}(\mathbf{v}(t-1))$ seja sempre inferior a um pixel [Cohen91]. No entanto, esta solução apresenta como contrapartidas, tornar o processo iterativo muito lento e o efeito das forças externas muito reduzido em comparação com o das internas, fazendo com que apenas alguns pontos onde o gradiente é muito elevado atraiam efectivamente o contorno.

Outra solução mais eficaz proposta por Cohen [Cohen91] consiste em normalizar a força externa em vez de actuar sobre o passo temporal (τ), fazendo:

$$\begin{cases} \mathbf{F} = -k \frac{\nabla P}{\|\mathbf{P}\|} \\ \tau k \approx 1 \end{cases} \quad (64)$$

Utilizando esta definição, $\mathbf{F}$ nunca é superior a um pixel e pequenos gradientes atraem o contorno, devido à normalização da função potencial. Desta forma, gradientes baixos e elevados atraem igualmente o contorno, estabilizando este nos mínimos da função gradiente do potencial que correspondem aos cruzamentos por zero da função força.

Uma solução mais elaborada consiste em estabelecer um limiar inferior para o valor da função potencial, abaixo do qual a força resultante seja considerada nula:

$$\begin{cases} \text{Se } \|P\| > \text{Lim} & F = -k \frac{\nabla P}{\|P\|} \\ \text{Se } \|P\| \leq \text{Lim} & F = 0 \\ \tau k \approx 1 \end{cases} \quad (65)$$

A utilização deste limiar inferior faz com que muito reduzidos potenciais, provavelmente resultantes de ruído na imagem, não contribuam para o deslocamento do contorno gerando forças que sobre ele actuem.

2.5.2 Instabilidade Devido à Discretização no Espaço

O problema da instabilidade devido à discretização no espaço surge devido às forças da imagem só serem conhecidas numa rede discreta de pontos, correspondendo à imagem em análise. Este problema torna muito provável a existência de uma oscilação na posição de um determinado ponto tal como é visível na Figura 27.

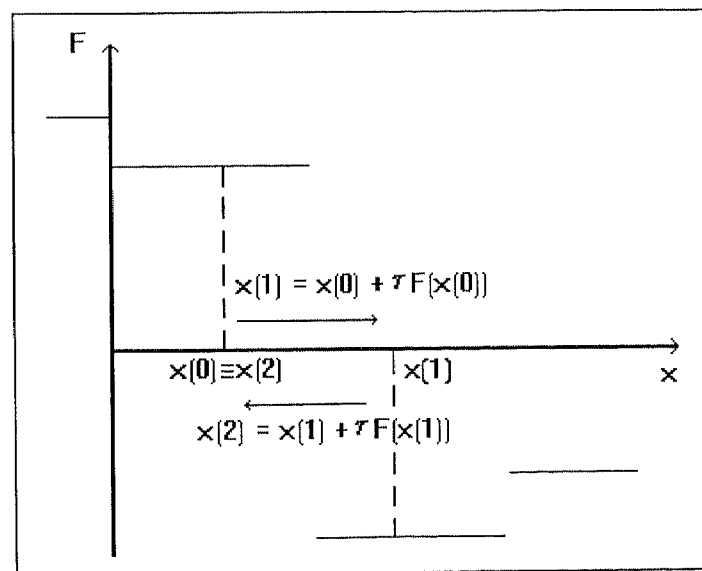


Figura 27: Instabilidade devido à discretização no espaço (utilização de uma força F conhecida apenas numa rede discreta).

A oscilação verificada deve-se a que o ponto considerado é sujeito, em iterações sucessivas, a forças da imagem de igual intensidade mas sentidos opostos (devido à

rede discreta em que é conhecida a força da imagem). O cruzamento por zero da força encontra-se entre os dois pontos da imagem sem ser no entanto visível na rede discreta. O problema descrito pode ser facilmente resolvido através de uma interpolação bilinear da força F em posições que não sejam inteiras.

Através da interpolação descrita, obtém-se um campo contínuo para a Força da Imagem. Desta forma não ocorrem instabilidades uma vez que os pontos do contorno convergirão para os cruzamentos por zero da força F .

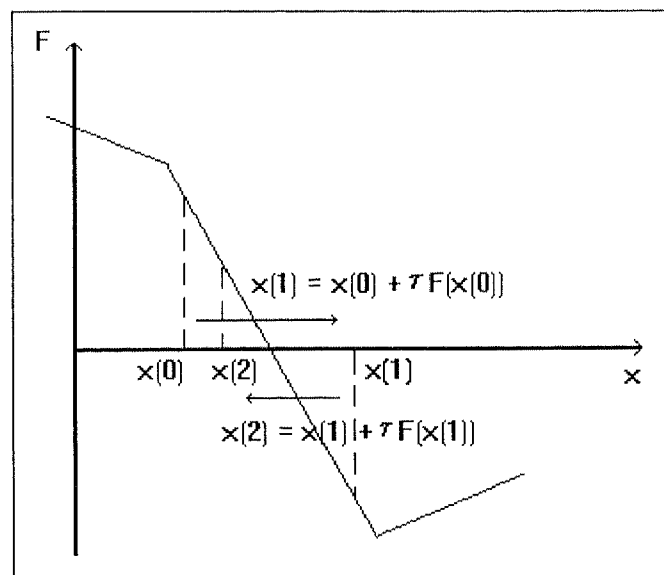


Figura 28: Resolução do problema da Instabilidade devido à discretização no espaço utilizando uma interpolação bilinear da força F em pontos não inteiros.

Tal como é visível na Figura 28, a interpolação bilinear transforma o campo de forças da imagem num campo contínuo, implicando a fácil convergência, ao longo do processo iterativo, dos pontos do contorno, para os cruzamentos por zero desse campo.

2.5.3 Inicialização da "Snake"

A inicialização do contorno é um dos problemas mais complexos na implementação do método dos contornos activos [Henricson94, Neuenschwnder94]. O resultado final obtido está intimamente ligado com a inicialização do contorno efectuada. Dependendo do contorno inicial, pode ser obtida uma infinidade de contornos distintos finais, correspondendo unicamente alguns à detecção dos contornos do objecto pretendido.



Figura 29: Imagem contendo sete objectos distintos.

A Figura 29, apresenta uma imagem contendo diversos objectos, nomeadamente uma esferográfica, uma disquete, um clipe, uma chave, uma borracha e duas tampas de objectivas. Se o objectivo da análise efectuada for o de detectar os contornos da tampa fotográfica (com a inscrição "olympus"), naturalmente que o contorno terá de ser inicializado na região correspondente a esse objecto (Figura 30).



Figura 30: Inicialização de um contorno para detecção do objecto "tampa".

Se o contorno for inicializado em zonas correspondentes aos outros objectos, naturalmente que será impossível obter como resultado final a detecção do objecto pretendido. Por exemplo, a inicialização demonstrada na Figura 31, conduz no final do processo de minimização energética à detecção de um contorno que não corresponde a nenhum dos objectos apresentados.



Figura 31: Inicialização de um contorno que não permite detectar o objecto tampa.



Figura 32: Resultado obtido com a inicialização apresentada na figura 31.

O contorno inicializado na Figura 31, encontra-se no início do processo iterativo, localizado entre os objectos “tampa” e “borracha”, sendo atraído pelas arestas destes objectos. Devido à abertura do contorno provocada pela força de enchimento e pela adaptação do contorno ao objecto “tampa”, este aproxima-se do objecto “disquete”. A proximidade desse objecto, faz com que o contorno seja atraído pela sua muito forte aresta superior. O contorno encontra desta forma o seu equilíbrio energético, numa posição intermédia entre os três objectos, ilustrada na Figura 32.

A selecção do contorno inicial depende fundamentalmente de três factores:

- Características gerais da imagem em análise
- Contorno a detectar, sua localização, tipo e forma
- Parâmetros da “snake” e do algoritmo seleccionados

Em primeiro lugar é necessário ter em atenção o tipo de imagens em análise, nomeadamente no que diz respeito à averiguação da possibilidade de selecção automática do contorno inicial ou da necessidade de efectuar operações de pré-processamento que permitam uma mais fácil selecção desse contorno.

O tipo de contorno a detectar, a sua localização na imagem e a sua forma apresentam uma enorme importância no processo de inicialização. Existem grandes vantagens em definir um contorno inicial com uma forma semelhante à desejada e que se situe bastante próximo do contorno final desejado.

Os parâmetros da "snake" e do algoritmo seleccionados também condicionam fortemente o tipo de contorno inicial a definir. Por exemplo, se for seleccionada uma força de enchimento considerável, o contorno inicial terá de ser definido no interior do objecto cujo contorno se pretende detectar.

No que diz respeito à forma de definição do contorno, existem basicamente três formas de o fazer:

- Definição automática
- Definição semi-automática
- Definição manual

A definição automática do contorno inicial baseia-se em métodos bastante variados que, no global, permitem detectar automaticamente um contorno que sirva garantidamente como ponto de partida para o método iterativo (por exemplo que esteja próximo do contorno desejado ou seguramente no interior deste). Este tipo de inicialização só é possível em alguns tipos de imagens, sendo totalmente impossível em outros casos.

Na definição semi-automática, o programa sugere uma possível localização para o contorno inicial podendo o utilizador em seguida corrigir essa definição inicial (antes do início do processo iterativo). Este tipo de definição é útil na maioria das situações em que é simples determinar automaticamente um contorno inicial e, através de pequenas correcções do utilizador, se podem conseguir muito melhores resultados no processo iterativo.

A definição manual é útil sobretudo em situações em que é totalmente impossível definir automaticamente um contorno inicial (como no exemplo analisado anteriormente). Nestas situações, a definição manual do contorno pode ser efectuada de diversas formas, tais como a definição de um conjunto de pontos chave inicial, seguida de interpolação (para determinar os pontos restantes) ou a definição de um contorno inicial

com uma forma simples, seguida da possibilidade de efectuar ajustes simples sobre esse contorno.

2.5.4 Dificuldade na Detecção de Contornos Afastados

Ao longo das secções precedentes ficou bem patente que um dos principais problemas do método dos contornos activos consiste nas suas grandes dificuldades de detecção eficaz de contornos que se encontrem distantes da solução inicial. Um exemplo desta dificuldade pode ser analisado na Figura 33.

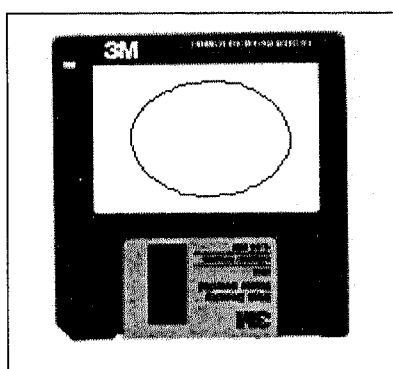


Figura 33: Inicialização de um contorno que não permite detectar os contornos do objecto disquete.

Como é visível na figura, se o contorno desejado for o contorno exterior do objecto representado (disquete), a inicialização efectuada certamente não irá conduzir a bons resultados. O movimento do contorno é travado pelos contornos interiores da etiqueta da disquete, correspondendo o resultado final (Figura 34) à detecção precisamente desta etiqueta.

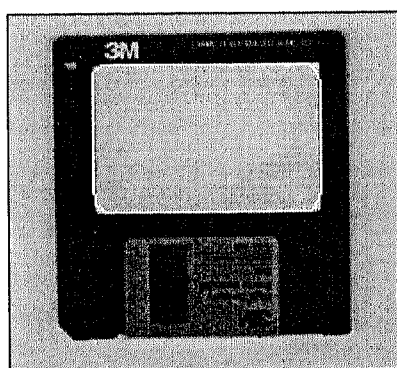


Figura 34: Resultados obtidos utilizando a inicialização apresentada na figura 33, com consequente detecção da etiqueta.

Embora exista sempre uma grande dependência do método relativamente ao contorno inicial, esta pode ser minorada através da utilização de diversas técnicas auxiliares, tais como:

- Interação com o utilizador na definição do contorno inicial de forma a obter um contorno de partida para o processo iterativo mais próximo da solução final desejada.
- Definição de um contorno inicial no interior do objecto cujo contorno se pretende detectar e utilização de uma força de enchimento com um valor suficiente para insuflar o contorno até à posição final pretendida (desde que não existam no interior do objecto arestas suficientemente fortes para parar o movimento do contorno).
- Utilização de algoritmos hierárquicos que utilizem inicialmente imagens força muito suavizadas e que vão focando gradualmente essas imagens força de forma a obter um contorno final mais preciso.

A utilização destas e de outras técnicas será objecto de uma análise mais detalhada em secções posteriores deste trabalho.

2.5.5 Optimização dos Coeficientes de Rigidez e Elasticidade

A escolha dos coeficientes de rigidez (**w2**) e de elasticidade (**w1**) da "snake" apresenta uma importância vital nos resultados finais. Cohen [Cohen91, Cohen93], baseado em considerações de análise numérica, sugere que os coeficientes referidos assumam valores próximos de $\mathbf{w1=h^2}$ e $\mathbf{w2=h^4}$, de forma a que todos os coeficientes da matriz **A** possuam valores da mesma ordem de grandeza. Uma escolha mais criteriosa dos parâmetros **w1** e **w2** deve ser guiada tendo em conta as propriedades mecânicas pretendidas para o contorno final a detectar.

Na maioria dos casos, existe um conflito entre os parâmetros idealizados tendo em conta as características finais desejadas para o contorno solução e os parâmetros que conduzem a melhores resultados no processo iterativo. De facto, os contornos finais pretendidos apresentam normalmente uma forma pouco suave sendo consequentemente necessárias baixas resistências à flexão e ao alongamento por parte do contorno. No entanto, uma especificação inicial deste tipo para os parâmetros **w1** e **w2** conduz a que durante a fase inicial do processo iterativo, os seus pontos se agrupem em "clusters" ao serem atraídos pelas zonas onde se verificam forças da imagem mais intensas. As baixas resistências ao alongamento e flexão não são suficientes para manter o contorno suave.

Uma solução para o problema apresentado consiste na variação ao longo do processo iterativo dos parâmetros $w1$ e $w2$ [Bajcsy89, Maurincomme94]. À medida que a "snake" vai encontrando uma solução final estável é possível diminuir a sua rigidez e elasticidade de forma a que ela se possa adaptar com maior precisão ao contorno

2.6 SEGUIMENTO DE CONTORNOS

2.6.1 Introdução

A aplicação de contornos activos a problemas de seguimento de contornos em sequências de imagens, foi executada com sucesso por diversos autores [Takahata92, Fujimura93, Delagnes95]. Os resultados mais satisfatórios foram obtidos em aplicações em que:

- Os objectos se deslocam sobre um fundo relativamente uniforme [Fujumura93].
- A forma dos objectos a seguir, encontra-se bem adaptada à utilização de contornos activos [Takahata92, Leymarie93] (forma suave tipo elipse).

Mesmo em aplicações em que os objectos possuem formas não elípticas, com contornos pouco suaves e movem-se sobre um fundo complexo (contendo múltiplos contornos), alguns autores têm também alcançado resultados bastante satisfatórios [Delagnes95].

2.6.2 Esquema de Seguimento

Um possível esquema de seguimento de contornos consiste em três operações básicas (Figura 31) [Delagnes95]:

- Predição da posição do contorno
- Ajustamento da posição do contorno
- Estimação do movimento

Tendo em conta uma determinada descrição $v(s,t)$ de um contorno no instante t , a sua posição no instante $v(s,t+1)$ pode ser prevista tendo em conta a informação respeitante ao movimento estimado na imagem no instante t . Partindo do contorno previsto e utilizando o algoritmo dos contornos activos, tal como descrito anteriormente, podemos corrigir a posição do contorno utilizando a informação espacial disponível no

instante $t+1$, obtendo assim uma definição muito mais precisa do contorno. Os principais problemas que se colocam a este método consistem em:

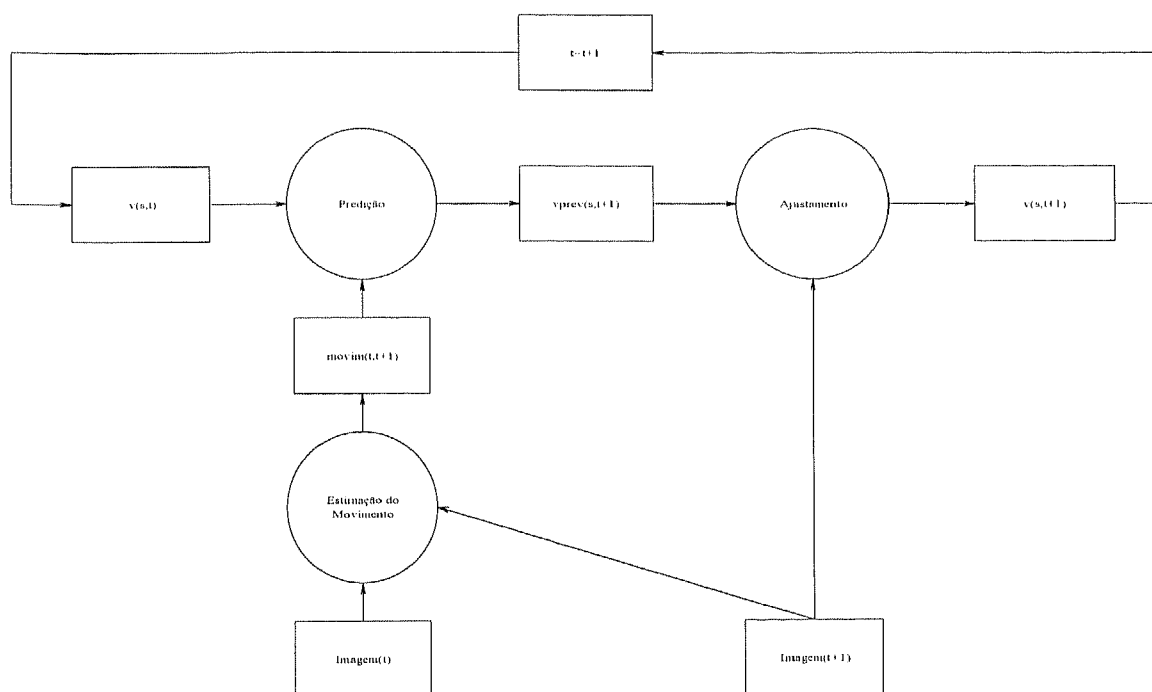


Figura 31: Esquema geral de seguimento de contornos em sequências de imagens.

- Detectar o contorno inicial pois, nesta fase do processo, não existe nenhuma informação sobre o contorno solução para o instante de tempo anterior.
- Definir o método de estimação do movimento a utilizar. Este método deverá detectar as variações temporais da imagem e estimar a rapidez e direcção dessas variações.

Uma análise muito detalhada destes problemas, nomeadamente do problema da estimação do movimento, envolve diversos conceitos que fogem um pouco do âmbito deste trabalho, pelo que não será apresentada neste texto.

No que diz respeito à operação de predição da posição do contorno no instante seguinte, esta pode ser efectuada simplesmente, deslocando cada um dos pontos, de um valor proporcional à sua velocidade estimada no instante anterior ou à velocidade global do contorno estimada no mesmo instante.

Após determinada a posição prevista para o contorno no instante seguinte, o seu ajuste é efectuado utilizando o processo iterativo descrito nas secções precedentes. O esforço computacional envolvido neste processo é bastante reduzido, uma vez que dispomos de uma muito boa aproximação da solução final desejada (obtida pelo preditor).

2.7 APLICAÇÕES DE CONTORNOS ACTIVOS

Actualmente verifica-se um grande esforço de investigação e uma aplicação já relativamente generalizada das técnicas de detecção de contornos baseadas em modelos de minimização de energia. As principais aplicações situam-se na área da análise de imagem médica e na reconstrução de informação tridimensional. Entre as aplicações e trabalhos mais conhecidos destacam-se:

- Reconstrução 3D em imagens médicas através da extracção de secções cruzadas sucessivas (ventrículo esquerdo do coração [Cohen91]).
- Segmentação de imagens de ultra-sons intravasculares [Maurincomme93, Maurincomme94]
- Segmentação de imagens radiográficas [Radeva93a, Radeva93b]
- Segmentação de imagens de detecção remota [Cohen93, Neuenschwander94]
- Reconstrução 2D e 3D de imagens médicas [Cohen91, Cohen92, Goshtasby93]
- Determinação de movimento e seguimento de contornos [Takahata92, Leymarie93, Fujimura93, Baumberg93, Baumberg94, Bascle94 e Delagnes95]
- Análise e síntese de expressões faciais [Kass88, Yuille92, Neuenschwander94, Essa95]
- Reconhecimento de caracteres [Lai94a, Lai94b]

Embora a lista de campos de aplicação apresentada, não seja minimamente exaustiva, é visível que as técnicas de detecção e seguimento de contornos baseadas em contornos activos, têm sido já aplicadas, com sucesso, a imensos problemas na área do processamento e análise de imagem. Desta forma fica comprovada a utilidade prática que apresenta o desenvolvimento de um pacote de software que permita a fácil aplicação dessas técnicas, e que sirva de base para o desenvolvimento de aplicações específicas de detecção e seguimento de contornos.

3. IMPLEMENTAÇÃO DO SISTEMA DE DETECÇÃO E SEGUIMENTO DE CONTORNOS

3.1 INTRODUÇÃO

Como objectivo de efectuar a aplicação prática dos métodos de detecção e seguimento de contornos baseados em contornos activos, foi desenvolvido, de raiz, um pacote de software de processamento e análise de imagem. O software foi desenvolvido em ambiente Windows [Blackwell95], utilizando a linguagem C++ [Holzner94]. A interface com o utilizador apresenta as características comuns de uma aplicação Windows, baseando-se em janelas sobreponíveis, menus, botões e caixas de diálogo.

No pacote de software desenvolvido foram incluídas rotinas básicas de tratamento de ficheiros (contendo imagens ou definição de contornos) e de visualização e edição de imagens coloridas (em janelas de dimensão variável). Foram ainda incluídas rotinas diversas de tratamento de imagem, tais como: operações lógicas, aritméticas e morfológicas, convoluções, cálculo de gradientes, limiarizações e filtragens. Diversos métodos tradicionais de detecção de arestas são ainda disponibilizados no software.

As rotinas essenciais do programa servem de base à aplicação do método dos contornos activos. São disponibilizadas rotinas que permitem definir interactivamente um contorno inicial, movimentar esse contorno utilizando os parâmetros especificados para o algoritmo de movimento e visualizar os resultados obtidos, numa grande variedade de formas. É ainda possível efectuar a análise quantitativa dos contornos finais obtidos através da extracção de diversas características. Outras potencialidades incluem o armazenamento e visualização de filmes com a movimentação do contorno ao longo do processo iterativo.

3.2 PLATAFORMA DE DESENVOLVIMENTO SELECIONADA

3.2.1 Sistema Operativo Utilizado

O sistema de detecção de contornos foi desenvolvido sobre o sistema operativo Windows, procurando utilizar ao máximo as potencialidades gráficas deste sistema. A escolha do ambiente Windows baseou-se essencialmente em 5 factores:

- Disponibilidade de hardware adequado ao trabalho, compatível com o ambiente Windows (computador, impressora, placa de aquisição de imagem e "scanner").
- Disponibilidade de linguagens de programação neste sistema, adequadas ao desenvolvimento de raiz de um sistema de processamento de imagem (Visual C++, Borland C++, Turbo Pascal para Windows, Borland Pascal, MATLAB, etc).
- Potencialidades gráficas disponibilizadas e facilidade de utilização de rotinas gráficas próprias do sistema (definição de menus, caixas de diálogo, botões, tratamento de janelas, etc).
- Facilidade de conversão de rotinas disponíveis em outros ambientes de desenvolvimento (MS-DOS¹, UNIX, X-Windows) para o ambiente Windows
- Vantagens do Windows a nível de interacção com o utilizador quando comparado com outros sistemas concorrentes (essencial, por exemplo, na fase de definição do contorno inicial, por parte do utilizador).

Acrescido a estes factores, foi ainda tomada em consideração a cada vez maior generalização e facilidade de utilização do ambiente Windows e a sua crescente superiorização a nível de quota de mercado, com consequentes benefícios ao nível do número de potenciais utilizadores do sistema desenvolvido.

3.2.2 Linguagem de Programação

Para o desenvolvimento do sistema de processamento de imagem e detecção de contornos foi seleccionada a linguagem Visual C++ 1.5 da Microsoft. Esta escolha baseou-se na análise das linguagens mais poderosas disponíveis² no ambiente de desenvolvimento seleccionado (Microsoft Windows). Nesta análise verificou-se que o Visual C++ apresentava como principais vantagens:

- Utilização de todas as potencialidades oferecidas pela linguagens de programação C e C++.
- Facilidade de adaptação, para esta linguagem de programação e ambiente, de rotinas básicas de processamento de imagem, desenvolvidas para outros ambientes (XITE³, MegaWave2⁴, etc).

¹ Microsoft Disk Operating System

² Em Fevereiro de 1995, data de início da implementação do sistema

³ X-based Image Processing Tools and Environment

⁴ Sistema de processamento de imagem do INRIA (Institut National de Recherche en Informatique et en Automatique)

- Disponibilidade do “Application Wizard” que permite construir, automaticamente, uma aplicação Windows muito simples, com características definidas pelo programador e que serve de base para a construção de aplicações mais poderosas.
- Facilidades oferecidas pelo “Application Studio” que permite desenhar de forma extremamente simples, recursos Windows tais como menus, caixas de diálogo ou ícones.
- Facilidades concedidas pelo “Class Wizard” que permite construir classes e ligar as mensagens provenientes de controlos - tais como itens dos menus e botões - ao código do programa.
- Disponibilidade das “Microsoft Foundation Classes” (MFC) que funcionam como um repositório de componentes de software, reutilizáveis para as mais diversas funções (desde a manipulação de “strings”, listas e vectores até ao tratamento de bases de dados).
- Facilidade de implementação automática dos menus “standard” das aplicações Windows, ou seja, os menus “File”, “Edit”, “View”, “Window” e “Help”.
- Uma arquitectura de programação flexível, baseada em objectos do tipo documento que tratam dos dados, e em objectos de visualização que tratam da interacção do utilizador com os documentos.
- Possibilidade de utilização de classes que encapsulam a Windows API⁵: janelas, caixas de diálogo, “device contexts”, GDI⁶, objectos desenhados, controlos, etc.
- Disponibilidade de classes (no repositório MFC) com funções básicas de visualização de imagens em formato DIB (“Device Independent Bitmap”).
- Possibilidade e facilidade de implementação de: janelas subdivisíveis e com “scroll”, capacidades de impressão e previsão de impressão, barras de ferramentas e ajuda adequada ao contexto (“context-sensitive help”).
- Enormes potencialidades do ambiente a nível de teste, “Debug”⁷ e correcção de erros em programas de média dimensão.
- Facilidades de gestão de um projecto de software de média dimensão oferecidas pelo ambiente Visual C++.

Associando todas as características referidas, obtém-se um ambiente de programação com enormes potencialidades, muito fácil de utilizar, extremamente robusto

⁵ Do inglês “Application Program Interface”

⁶ Graphic Device Interface

⁷ Rastreamento

e que se apresenta ideal para o desenvolvimento de raiz de um sistema de processamento e análise de imagem.

3.2.3 Estrutura Básica de um Programa Windows

Um programa básico MS-DOS em linguagem C [Schildt90], possui sempre o requisito de possuir uma função designada por “main”. Quando o utilizador deseja executar o programa, o sistema operativo chama essa função e a partir desse ponto é possível utilizar qualquer estrutura ou função no programa. Se o programa necessitar de responder à pressão em teclas, ou utilizar serviços do sistema operativo, chama as funções apropriadas, tais como “getchar”.

Quando o sistema operativo Windows executa um programa, chama a função “Winmain” desse programa. Esta função, realiza um conjunto de tarefas específicas, sendo a mais importante a criação da janela principal do programa que deve conter código para processar as mensagens que o Windows lhe envia. Esta é a principal diferença entre uma aplicação Windows e um programa escrito para o sistema operativo MS-DOS. Em DOS, um programa chama o sistema operativo para obter os comandos do utilizador. Em Windows, um programa, processa os comandos do utilizador através de mensagens do sistema operativo.

Os programas (aplicações) em ambiente Windows são baseados no conceito de transmissão de mensagens [Adler90a], apresentando um ciclo de processamento das mesmas. Sempre que ocorre um determinado evento, o sistema gera uma mensagem e transmite-a a todas as aplicações eventualmente interessadas (Figura 32). Em conformidade com o objectivo do programa e a semântica da mensagem, são (ou não) efectuadas determinadas acções, sendo em seguida libertado o processador. O Windows é um sistema operativo multi-tarefa em que uma aplicação tem que libertar explicitamente o controlo da unidade central de processamento (CPU⁸) para que outra aplicação a ele possa ganhar acesso. É também por mensagens que as várias aplicações comunicam entre si, e que o próprio sistema operativo comunica com as aplicações. Cada aplicação assume o controlo quando lê, da fila de mensagens, uma mensagem a ela destinada. Liberta o controlo, transmitindo-o a outra aplicação quando termina a execução da acção correspondente à mensagem processada.

⁸ Control Process Unit

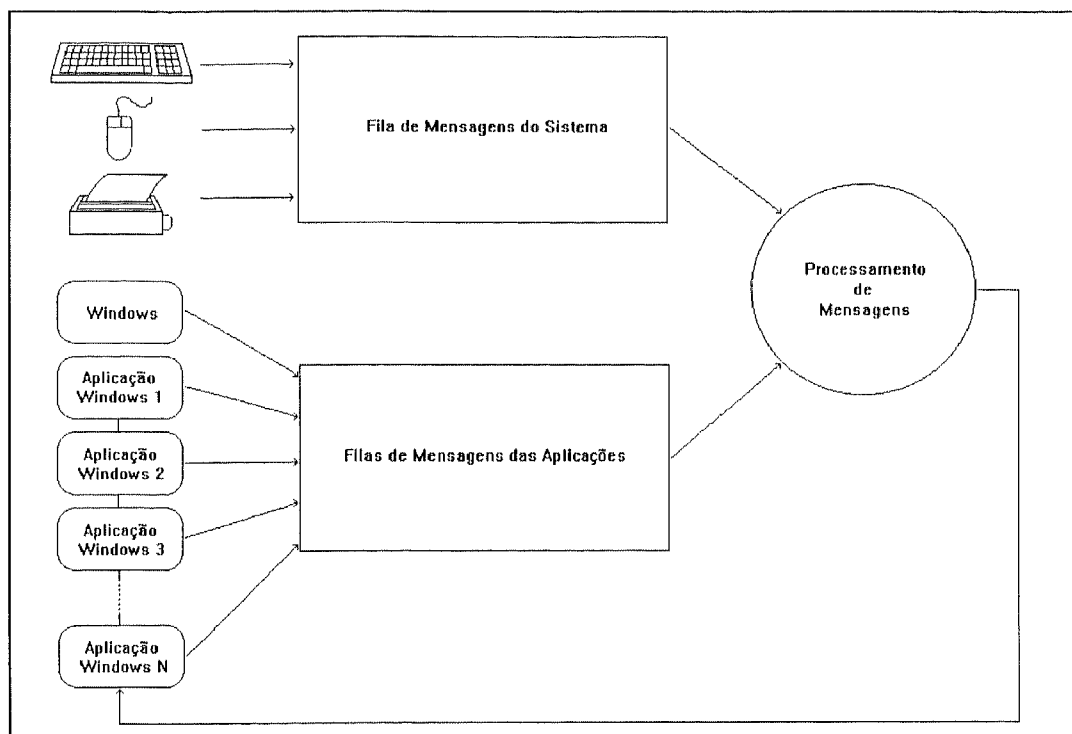


Figura 32: Processamento de Mensagens Windows.

Cada programa em Windows tem uma janela associada, onde é efectuada toda a interacção com o utilizador utilizando menus [Adler90b], janelas [Adler90c] e caixas de diálogo [Adler91a]. As janelas das aplicações (em Windows 3.1) são constituídas por diversas partes [Holzner94], onde se destacam (Figura 33): a barra de título, o menu de sistema, os botões de minimização e maximização, a barra de menu, a área de cliente e as barras de “scroll” horizontal e vertical.

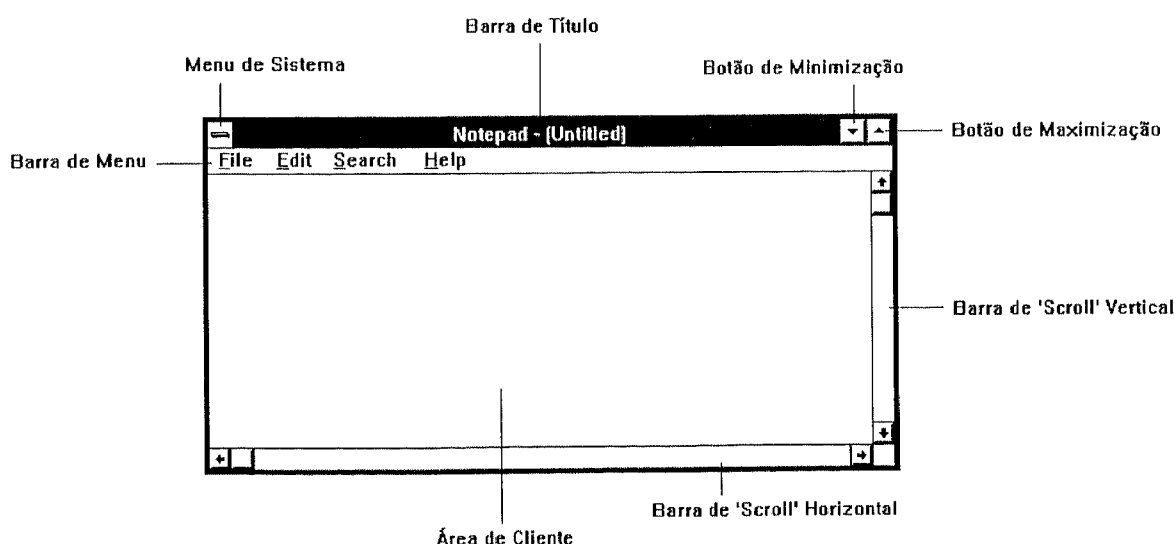


Figura 33: Componentes de uma Janela Windows.

Utilizando a denominada área de cliente da janela de aplicação - na janela associada ao programa - são visualizados todos os resultados do processamento efectuado pelo programa e das acções realizadas pelo utilizador. Esta área constitui praticamente a totalidade da janela que se situa abaixo da barra de menu (exceptuam-se as barras de "scroll"). É nesta área que podem ser colocados os botões, caixas de listagem, caixas de texto e outras partes do programa que em Windows se designam por "controles".

Exceptuando a área de cliente, tudo o que se situa na janela do Windows é designado por área não cliente. O sistema Windows é responsável pela manutenção de tudo o que se situa na área não cliente enquanto o utilizador assume a responsabilidade pela área de cliente.

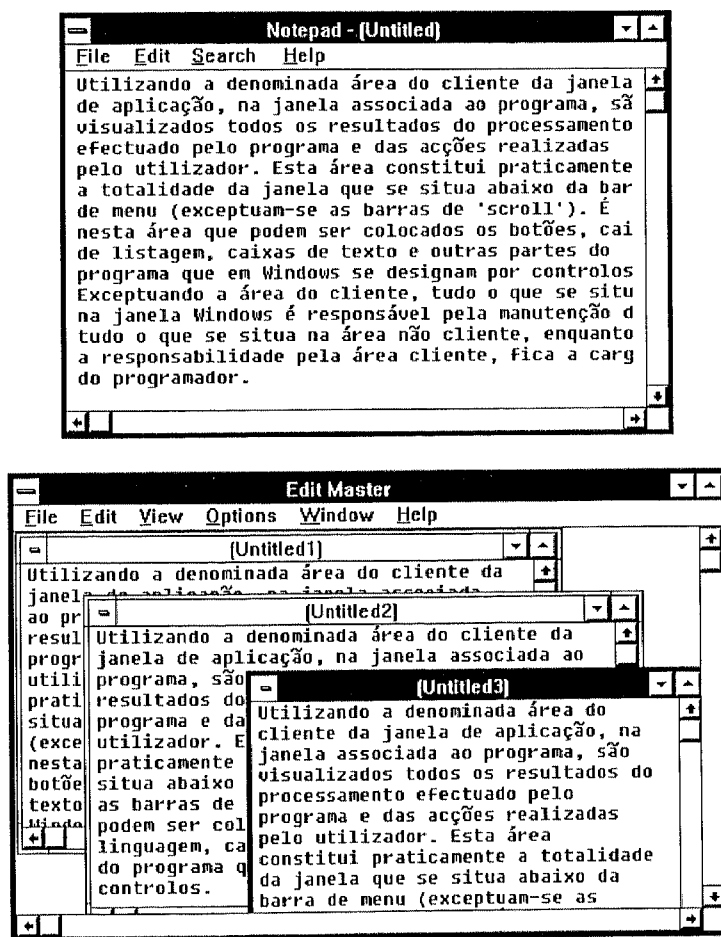


Figura 34: Distinção entre as aplicações SDI (em cima) e MDI (em baixo)

Uma aplicação Windows pode ser baseada numa única janela de interacção (interface SDI⁹), associada à área de cliente da janela de aplicação ou ter múltiplas janelas. Neste último caso (interface MDI¹⁰), a aplicação possui na área de cliente, uma janela para cada documento, denominada janela filha¹¹.

Na Figura 34 são visíveis as distinções entre as aplicações SDI e MDI. Na aplicação visualizada na parte superior da figura, não existem janelas na área de cliente efectuando-se a visualização de resultados directamente sobre esta. Na aplicação da parte inferior, múltiplos documentos podem ser visualizados simultaneamente sobre a área de cliente da aplicação.

Outra característica distintiva dos programas Windows encontra-se relacionada com a forma como são tratados os dispositivos tais como a placa gráfica ou impressora. Muitos programas MS-DOS, escrevem directamente sobre a memória vídeo e porta da impressora. O Windows introduz uma camada de abstracção designada por "Graphics Device Interface" (GDI) [Adler91b]. O Windows fornece todos os "drivers" para o écran e impressora, levando a que o utilizador não necessite de se preocupar com o tipo de placa gráfica ou impressora que se encontra ligada ao sistema [Adler91b, Adler91c]. Em vez de se relacionar directamente com o Hardware, um programa chama funções do GDI que referenciam uma estrutura designada por "device context". O Windows efectua automaticamente o mapeamento desta estrutura nos dispositivos físicos e executa as instruções de entrada/saída apropriadas.

3.2.4 Aplicações em Visual C++

Embora, ao utilizador comum, a interface Windows pareça simples e intuitiva, para o programador apresenta-se extensa e complicada. O programador Windows é confrontado com uma enorme quantidade de opções em termos de estilos de janelas, formas de executar o programa, dimensões de janelas, alocação de memória, etc. No entanto, a grande maioria dos programas não necessita dessa vastidão de opções, assumindo-se neste contexto o Visual C++ bastante indicado para as construir. Grande parte das selecções de opções podem ser executadas derivando novas classes a partir das classes "standard" fornecidas pela linguagem.

A arquitectura duma aplicação Windows utilizando a linguagem Visual C++ e biblioteca de classes MFC é baseada nos seguintes conceitos base:

⁹ "Single Document Interface"

¹⁰ "Multiple Document Interface"

¹¹ Do inglês "Child Window"

- A aplicação Windows assim construída constitui um objecto do tipo “Application Object”. Este objecto controla todos os documentos utilizados pelo programa e envia os comandos para os outros objectos do programa.
- Os objectos com que o utilizador trabalha são documentos. Os objectos deste tipo mantêm, lêem e gravam os seus dados.
- O utilizador interage com um documento através de uma vista desse documento. Uma vista do documento é uma janela embebida na área cliente da aplicação. Nesta janela pode ser visualizado, e eventualmente editado, o conteúdo do documento respectivo. As acções efectuadas com o rato ou com o teclado são transformadas em acções de edição e selecção na vista activa.
- Os objectos da interface do programa tais como menus e botões, enviam comandos para as vistas dos documentos e eventualmente para os outros objectos da aplicação. Estes objectos encarregam-se de executar os comandos respectivos.

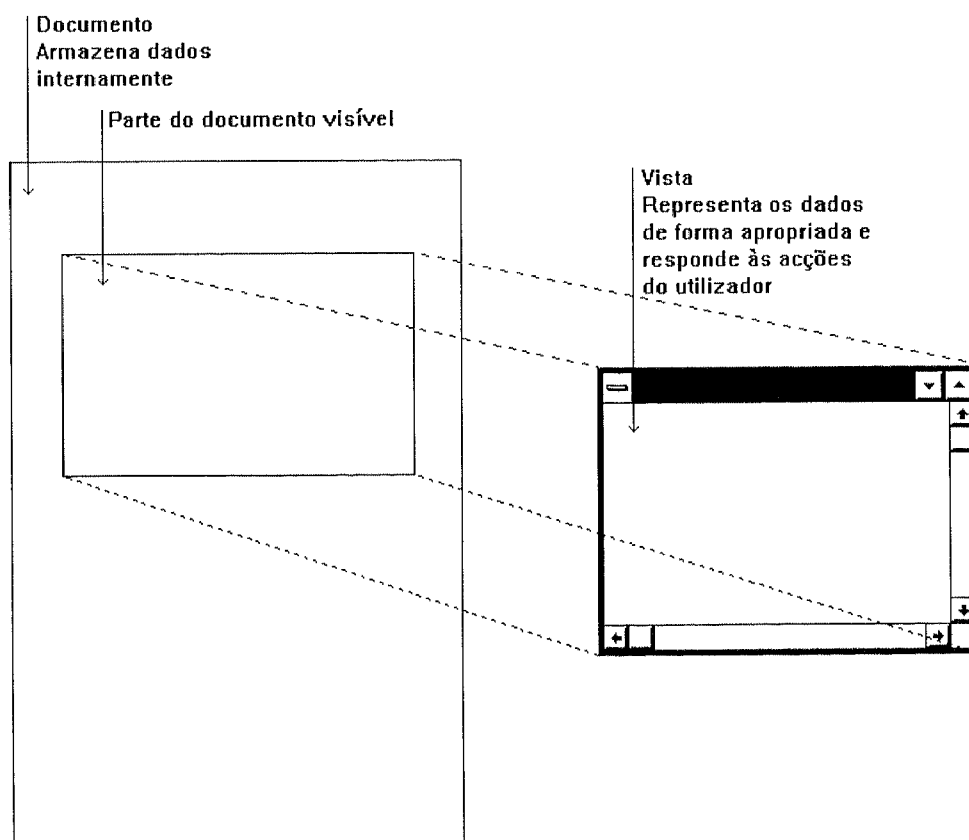


Figura 35: Relação entre um documento e as suas vistas.

A Figura 35 ilustra a relação entre um documento e as vistas desse documento. O documento armazena os dados no seu interior e encarrega-se de efectuar as acções de

manutenção, leitura e gravação de dados. Cada vista permite visualizar (e eventualmente editar) a totalidade ou unicamente uma determinada porção do documento.

3.2.5 Utilização das “Microsoft Foundation Classes”

A biblioteca “Microsoft Foundation Classes” [Kruglinsky93, Holzner94] providencia uma base sólida que permite ao programador construir facilmente a interface gráfica do programa. O programador pode assim dispendir a maior parte do seu tempo na construção do código de tratamento dos dados. A utilização desta biblioteca baseia-se no seguinte esquema temporal:

- Definição das classes tipo documento partindo dos dados que a aplicação deve tratar.
- Definição de como o utilizador irá visualizar e interactivar com os dados dentro das respectivas janelas de visualização.
- Conexão de menus, botões e outros controlos da interface aos respectivos comandos, definindo as funções que irão ser executadas por cada um desses comandos.

As classes genéricas principais da biblioteca MFC, relacionadas com a arquitectura de uma aplicação, encontram-se representadas na Figura 36. No topo da hierarquia situa-se a classe “CObject” que é a classe base principal da biblioteca MFC. Esta classe serve de base não só para todas as classes “standard” como também para as classes definidas pelo programador.

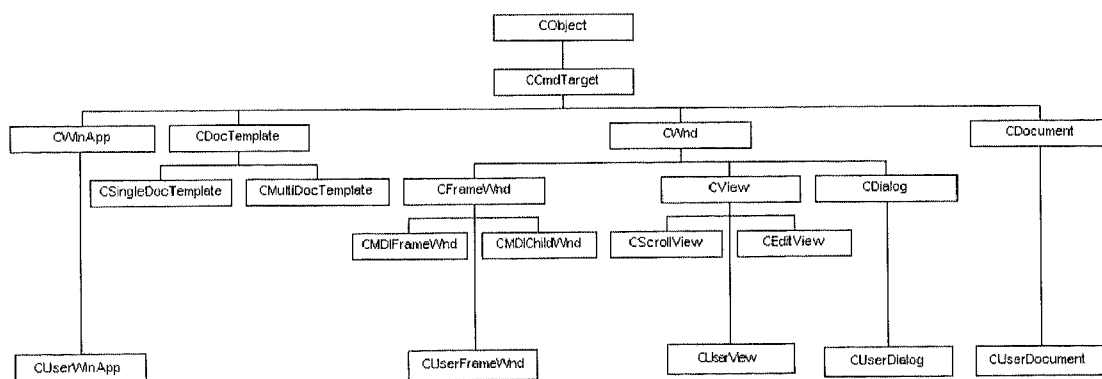


Figura 36: Hierarquia das principais MFCs relacionadas com a arquitectura de uma aplicação Windows.

A segunda classe “CCmdTarget” é a classe base para todas as classes de objectos que podem receber e responder a mensagens. Um mapa de mensagens deve ser definido de forma a encaminhar os comandos ou mensagens para as funções que

são definidas para as tratar. Derivadas desta classe existem quatro outras classes: "CWinApp", "CDocTemplate", "CDocument" e "CWnd".

A classe "CWinApp" é a classe base a partir da qual é derivada a aplicação Windows propriamente dita. Encapsula o código necessário para inicializar, executar e terminar a aplicação.

A classe "CDocTemplate" constitui uma base para documentos padrão. Controla a criação dos objectos derivados de "Document", "View" e "FrameWindow" e as relações entre eles. Cada aplicação, possui um "CDocTemplate" para cada tipo de documento que manipula. As suas subclasses, "CSingleDocTemplate" e "CMultiDocTemplate", coordenam a criação de documentos em interfaces SDI e MDI, respectivamente.

A classe "CDocument" constitui a base para as classes de documentos específicos da aplicação que são definidos pelo utilizador (documentos que são tipicamente abertos com o comando do menu "Ficheiro" denominado "Abrir"). Uma aplicação pode suportar mais do que um tipo de documento (por exemplo imagens e textos) e as respectivas classes derivadas de "CDocument" providenciam as funções básicas de tratamento desses documentos tais como criação, abertura, fecho, gravação e impressão.

A classe "CWnd" fornece a base para todas as classes que lidam com janelas. A sua subclasse "CView" fornece a base para as classes definidas pelo utilizador que trabalham com vistas. Uma vista encontra-se sempre associada a um determinado documento e actua como interface entre o documento e o utilizador. Uma vista coloca visível no écran ou na impressora um documento e interpreta os comandos que o utilizador realiza sobre este. Uma vista encontra-se sempre associada a um único documento embora este possa ser visualizado simultaneamente por múltiplas vistas (por exemplo numa interface MDI um documento pode ser visualizado em diversas janelas ao mesmo tempo). Uma vista pode ser responsável por processar diversos tipos de entradas de dados tais como pressão de uma tecla, comandos do rato, comandos de menus, botões da barra de ferramentas ou manipulação das barras de "scroll".

Três classes indispensáveis à construção de uma aplicação Windows utilizando a biblioteca MFC são as classes "CFrameWnd", "CMDIFrameWnd" e "CMDIChildWnd". A primeira destas classes é a classe base para criar a janela principal de uma aplicação com interface SDI enquanto a segunda desempenha a mesma função para uma interface MDI. Estas classes permitem controlar a janela da aplicação que contém o título do programa, os menus e as barras de ferramentas e de estado. Numa interface MDI, as janelas filhas são derivadas da classe "CMDIChildWnd". As janelas filha são muito semelhantes às janelas normais, mas possuem a particularidade de surgirem sempre no interior da área de cliente da janela principal da aplicação.

3.3 DESCRIÇÃO DO SISTEMA DESENVOLVIDO

3.3.1 Potencialidades Requeridas ao Sistema

O sistema de processamento e análise de imagem desenvolvido procurou conter uma série de funcionalidades genéricas de processamento e análise de imagem e algumas funcionalidades específicas de análise de imagens biomédicas, sendo de destacar as funcionalidades relacionadas com a detecção de arestas e contornos. As potencialidades genéricas que idealmente um sistema de processamento e análise de imagem deve possuir são basicamente as seguintes:

- Captar imagens (e sequências de imagens) para memória, utilizando placas de aquisição, “scanners” ou outros periféricos apropriados para o efeito.
- Visualizar simultaneamente diversas imagens num monitor gráfico, explorando ao máximo as características disponibilizadas pelo “hardware” disponível.
- Realizar operações genéricas de processamento de imagem tais como filtragens, alterações de luminosidade e contraste, operações lógicas, aritméticas ou morfológicas.
- Realizar operações genéricas de análise de imagem tais como a medição de comprimentos, áreas ou perímetros e a extração de parâmetros estatísticos.
- Armazenar as imagens (e sequências de imagens) após o seu processamento, em formato “standard” e compatível com outros sistemas.
- Possuir uma interface gráfica interactiva e intuitiva que permita total controlo, por parte do utilizador, de todo o funcionamento do programa.

Relativamente às potencialidades específicas de análise de Imagem Biomédica e detecção de Arestas e Contornos, os principais requisitos do sistema foram os seguintes:

- Realizar operações de detecção de arestas baseadas nos principais algoritmos utilizados nesta área: Canny, Deriche, Shen e Castan, Sarkar e Boyer, etc.
- Permitir a definição e edição interactiva sobre as imagens de contornos, com vista ao seu ajuste a características relevantes da imagem.
- Realizar operações de movimentação de contornos sobre as imagens, utilizando algoritmos baseados em contornos activos, de forma a permitir a adaptação dos contornos às características desejadas da imagem.
- Permitir a medição de características relevantes dos contornos obtidos tais como a sua área e perímetro.

- Visualizar filmes descrevendo a movimentação dos contornos ao longo do seu processo iterativo de ajuste.
- Permitir o ajuste de contornos em sequências de imagens e a visualização de filmes contendo os contornos ao longo da sequência.
- Armazenar os filmes contendo a movimentação dos contornos, permitindo a sua posterior reedição e ajuste.

O sistema construído possui a quase totalidade das potencialidades genéricas e específicas descritas. Infelizmente, devido a uma avaria irreparável da placa de aquisição de imagem que estava a ser utilizada neste projecto, foi impossível incluir no software as rotinas específicas de aquisição de imagem que tinham sido desenvolvidas para a placa em questão.

3.3.2 Arquitectura do Sistema

A arquitectura do sistema de processamento de imagem desenvolvido procurou adaptar, da melhor forma possível, o sistema às potencialidades genéricas e específicas pretendidas. Esta arquitectura pode ser visualizada na Figura 37.

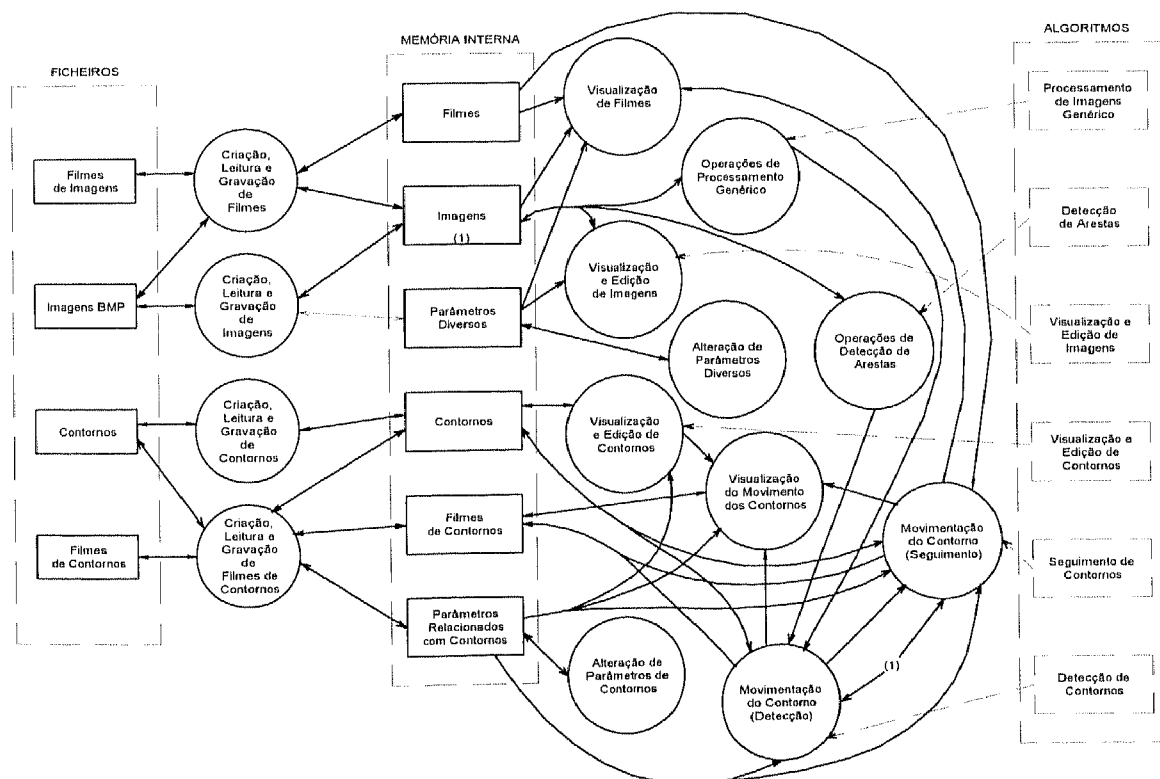


Figura 37: Arquitectura do sistema desenvolvido.

A arquitectura baseia-se na utilização de quatro tipos de ficheiros específicos:

- Imagens "Bitmap"
- Filmes de Imagens
- Contornos
- Filmes de Contornos

Quatro módulos independentes tratam das operações de criação, leitura e gravação dos quatro tipos de ficheiros. Sempre que se verifica uma operação de leitura (ou gravação) de um determinado filme, verifica-se a correspondente operação de leitura (ou gravação) de todas as imagens incluídas nesse filme. Da mesma forma, sempre que se verifica a leitura (ou gravação) de um ficheiro contendo um filme com a movimentação de um contorno, verifica-se a correspondente operação de leitura (ou gravação) de todos os contornos que compõe esse filme.

Na implementação prática do sistema, a arquitectura ilustrada na Figura 37 não foi totalmente respeitada em termos de independência dos ficheiros. Na realidade, como será analisado com maior profundidade em secções seguintes, devido à reduzida dimensão dos ficheiros contendo definições de contornos, estes são gravados como filmes de contornos. Desta forma, um filme com a movimentação de um contorno ao longo do processo iterativo, inclui no ficheiro que o define, todos os contornos que o compõem.

Todas as operações de leitura de imagens, filmes de imagens, contornos e filmes de contornos, encarregam-se de transformar as representações destes objectos, contidas nos ficheiros correspondentes, para as representações em memória respectivas, utilizadas pelo programa. Na memória interna do sistema, são ainda colocados diversos parâmetros de configuração global do sistema e dos seus algoritmos.

Todas as operações disponibilizadas pelo programa, utilizam os objectos e parâmetros contidos na memória do sistema, os algoritmos de processamento implementados e parâmetros diversos fornecidos pelo utilizador. Estas operações podem ser subdivididas em 8 grupos principais:

- Operações de processamento de imagem genéricas.
- Visualização e edição de imagens.
- Visualização de filmes de imagens.
- Operações de detecção de arestas.
- Operações de detecção de contornos.

- Operações de seguimento de contornos.
- Visualização e edição de contornos.
- Visualização de filmes contendo o movimento de contornos.

Para além destas operações, são ainda disponibilizadas ao utilizador, operações de alteração dos parâmetros de configuração do sistema e de configuração dos algoritmos de detecção e seguimento de contornos.

3.3.3 Implementação do Sistema

Tendo em conta a arquitectura demonstrada na secção anterior, ao longo desta secção será analisada a implementação do sistema utilizando a linguagem Visual C++. Serão analisadas as classes construídas a partir da biblioteca MFC e a sua interacção, procurando descrever o funcionamento global do sistema.

As classes principais implementadas no sistema (Figura 38) permitem executar a aplicação ("CSnakApp"), gerir a janela principal ("CSnakMainFrame"), gerir as janelas filhas ("CMDISnak"), manipular os documentos tipo imagem ("CSnakDoc") e tipo contorno ("CSnak2Doc") e manejar as vistas dos documentos tipo imagem ("CSnakView").

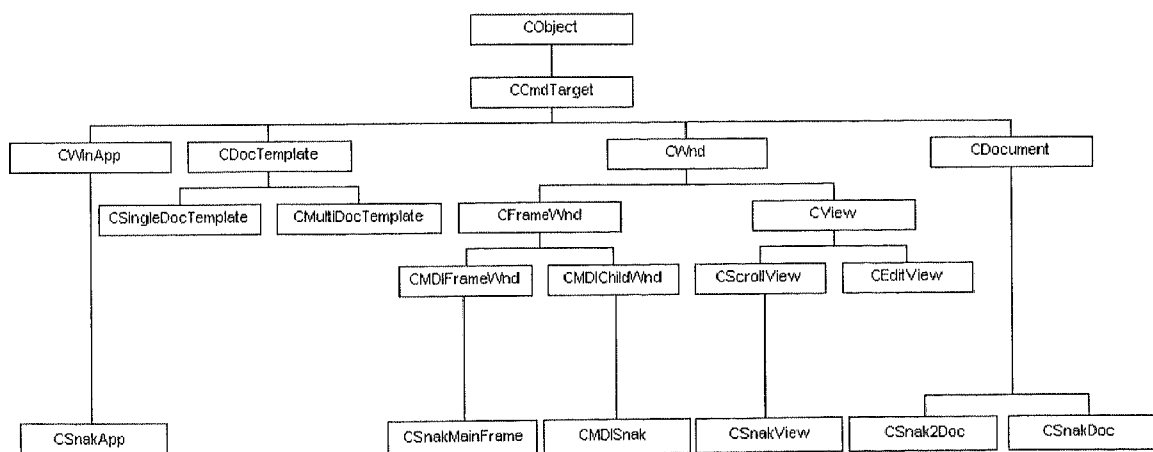


Figura 38: Hierarquia das principais classes implementadas no sistema.

Derivada da classe "CWinApp" foi criada a classe "CSnakApp", que efectua todas as tarefas necessárias à inicialização da aplicação, cria um documento padrão do tipo MDI e cria a janela principal da aplicação (também do tipo MDI).

A classe "CSnakMainFrame" descendente de "CMDIFrameWnd" é a classe associada à janela de aplicação, que permite gerir as diversas janelas filha numa interface do tipo MDI. Na sua criação é construída uma janela de aplicação com uma barra de ferramentas (contendo botões para acesso aos procedimentos mais frequentemente executado) e uma barra de estado (onde são visualizadas mensagens relativas ao funcionamento do programa). O construtor da classe "CSnakMainFrame" inclui código de inicialização das variáveis globais do programa, que são independentes do documento a processar. Nestas variáveis incluem-se o tamanho por defeito da janela de convolução e respectivos valores, e os parâmetros por defeito utilizados em diversos algoritmos de processamento. Outros parâmetros, considerados como variáveis globais e inicializados no construtor desta classe, estão relacionados com os filmes de contornos e de imagens. Na arquitectura do software foi definida a existência de um único filme de imagens e um único filme de contornos abertos em cada instante. Desta forma, toda a informação relativa a estes filmes, incluindo as suas definições e parâmetros, é incluída na classe "CSnakMainFrame".

Derivada de "CDocument" foi criada a classe "CSnakDoc", que serve de base a todos os objectos documento do tipo imagem (DIB). Nesta classe são declarados como variáveis as dimensões da imagem em pixels (segundo os dois eixos x e y), a tabela de cores na forma duma paleta¹² de cores com o formato Windows, e a própria imagem, na forma de um mapa de bits independente do dispositivo (DIB). Nesta classe foi colocado todo o código que permite efectuar a criação, leitura e gravação de imagens. Os filmes de imagens são armazenados em ficheiros comuns contendo os parâmetros do filme e a lista de imagens que o compõe, não tendo sido, por este motivo, necessário definir um objecto do tipo filme de imagens.

Baseada também na classe "CDocument" foi criada a classe "CSnak2Doc", que serve de base a todos os objectos documento do tipo filme de contornos. Nesta classe são declarados como variáveis o número de pontos do contorno, todas as suas propriedades, os parâmetros do algoritmo de movimentação que foi utilizado e, finalmente, a lista contendo a posição dos pontos do contorno ao longo do processo iterativo. Como os ficheiros contendo filmes de contornos têm sempre uma dimensão bastante reduzida, não foi definido um objecto tipo contorno. Os contornos fazem sempre parte do respectivo filme, podendo no entanto ser editados individualmente.

A classe "CMDISnak", descendente de "CMDIChildWnd", permite controlar cada uma das janelas filha da interface de múltiplos documentos. Nestas janelas são visualizadas as imagens contidas nos documentos tipo imagem processados pelo

¹² Do inglês "palette"

programa. Inicialmente, quando é aberto um determinado documento contendo uma imagem, é criada uma janela filha que permite visualizar essa imagem sem distorção. Efectuando alterações na dimensão dessa janela filha, tendo em conta o funcionamento (descrito em seguida) da classe "CSnakView" (responsável pela visualização) é possível visualizar a imagem ampliada ou apenas uma fracção desta.

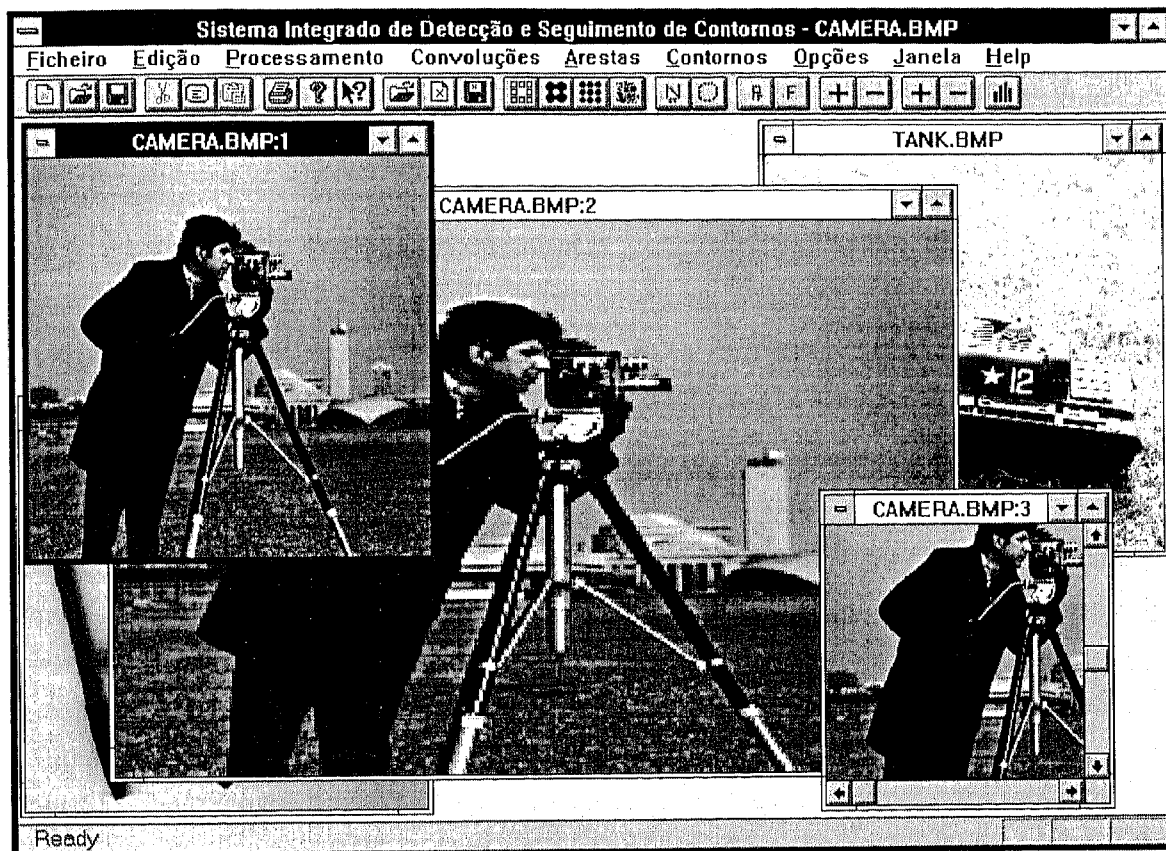


Figura 39: Formas de visualização de uma imagem.

Uma classe de grande importância no funcionamento do programa é "CSnakView", descendente de "CScrollView". Esta classe está associada à visualização, na área de cliente das diversas janelas filha, dos documentos que se encontram abertos na aplicação MDI. Com esta classe controla-se a visualização nas janelas filha dos documentos contendo imagens. As imagens podem ser visualizadas de três formas distintas (Figura 39):

- Totalmente e sem distorção (com uma relação de 1 para 1 entre os pixels da imagem e o monitor).
- Totalmente e ampliada (correspondendo cada pixel da imagem a diversos pixels do monitor, obtendo-se desta forma uma ampliação da imagem).

- Parcialmente (podendo ser utilizadas as barras de “scroll” horizontais e verticais da janela filha de forma a visualizar as partes ocultas da imagem).

Nesta classe encontra-se todo o código necessário para efectuar as conversões da imagem para a dimensão com que irá ser visualizada no monitor e para seleccionar correctamente as paletas de cores do sistema em função das paletas da imagem. É incluída também, a possibilidade de cortar, copiar e colar¹³, áreas rectangulares da imagem (definidas com o rato). Finalmente, uma das principais funções da classe de visualização, consiste em efectuar o processamento dos comandos da aplicação, introduzidos através dos menus e botões da barra de ferramentas.

3.3.4 Formatos de Imagem Utilizados

3.3.4.1 Formato DIB (“Device Independent Bitmap”)

O sistema de processamento e análise de imagem desenvolvido, como foi referido anteriormente, possui potencialidades de leitura, visualização, processamento e armazenamento de imagens. Para este efeito, foi necessário utilizar estruturas, denominadas formatos de imagem, que permitem manipular as imagens, tanto em memória como em disco. Estas estruturas, de forma a serem portáteis, devem seguir formatos “standard” e serem independentes dos recursos de vídeo. A resolução, características cromáticas e número de níveis de intensidade de cada pixel da imagem, deve ser uma imposição do sistema ou do processo de captação da imagem e não da estrutura que a representa. O formato de imagem não deve consequentemente impor restrições ao tipo de imagem, devendo permitir, pelo menos, a representação de imagens a cores (RGB), indexadas, 256 níveis de cinzento e preto e branco.

O “kit” de desenvolvimento de programas (SDK¹⁴) do Windows 3.1 oferece uma estrutura em memória com as características apresentadas, que é o formato DIB (“Device Independent Bitmap”) [Rimmer91, Charlap95]. Este é actualmente o formato “standard” de armazenamento de imagens em aplicações Windows, suportando imagens monocromáticas, 16 cores, 256 cores e 24 bits RGB. Devido à sua popularidade neste sistema, este formato de imagem começou também a ser comum em outros sistemas. Este formato é o equivalente em memória do formato BMP (“Bitmap”) [Rimmer91, Charlap95] que é vulgarmente utilizado neste sistema operativo para armazenar imagens e desenhos em disco.

¹³ Do inglês “cut”, “copy” e “paste”

¹⁴ “Software Development Kit”

O formato BMP ao contrário de formatos de imagem tais como o GIF ("CompuServe's Graphic Interchange File Format") ou o JPEG ("Joint Photographic Experts Group"), não foi projectado para ser portátil, compacto ou facilmente transmissível à distância [Rimmer91]. Em vez disso foi desenhado de forma a trabalhar facilmente com a Windows API [Rimmer92] utilizando as mesmas estruturas que as aplicações Windows utilizam para manipular mapas de bits em memória. Esta estrutura, devido a não apresentar, na sua forma mais simples, qualquer tipo de compressão de informação, permite uma grande rapidez de acesso às imagens. Como contrapartida, apresenta o facto de criar ficheiros que necessitam de enormes quantidades de espaço em disco e de necessitar de uma grande largura de banda para a sua transmissão em tempo útil.

Nesta secção serão descritas resumidamente as estruturas DIB (utilizada em grande parte dos processamentos internos de imagem realizados no sistema desenvolvido) e BMP (utilizada como forma de armazenamento de imagens em disco no sistema).

Um DIB é constituída pela estrutura "BITMAPHEADER" (Figura 40) que contém toda a informação necessária para descrever a imagem (dimensões, cores da imagem seguida opcionalmente por uma tabela de cores (paleta) "RGBQUAD" (Figura 41) e pela matriz de pixels que representam a imagem propriamente dita.

```
typedef struct
{
    LONG        biSize;
    LONG        biWidth;
    LONG        biHeight;
    INT         biPlanes;
    INT         biBitCount;
    LONG        biCompression;
    LONG        biSizeImage;
    LONG        biXPelsPerMeter;
    LONG        biYPelsPerMeter;
    LONG        biClrUsed;
    LONG        biClrImportant;
} BITMAPHEADER;
```

Figura 40: Estrutura BITMAPHEADER (cabeçalho do DIB).

A descrição dos campos da estrutura de cabeçalho, dum mapa de bits DIB é a seguinte:

- “biSize” define o tamanho do cabeçalho em bytes (assumindo sempre o valor 40).
- “biWidth” representa a largura da imagem, em pixels.
- “biHeight” representa a altura da imagem, em pixels.
- “biPlanes” especifica o número de planos da imagem, sendo sempre igual a um.
- “biBitCount” representa o número de bits por pixel da imagem, podendo assumir os valores 1, 4, 8, ou 24.
- “biCompression” especifica o tipo de compressão da imagem, podendo tomar o valor BI_RGB que determina um mapa de bits sem compressão, BI_RLE8 ou BI_RLE4 que especifica uma compressão do tipo RLE (“Run Length Encoding”) para imagens respectivamente com 8 ou 4 bits por pixel.
- “biSizeImage” representa o número de bytes ocupados pelos pixels da imagem, valor que é igual a $biWidth * biHeight * biBitCount / 8$.
- “biXPelsPerMeter” especifica a resolução horizontal da imagem, em pixels por metro, do dispositivo alvo para o mapa de bits.
- “biYPelsPerMeter” especifica a resolução vertical da imagem.
- “biClrUsed” especifica o número de cores, da tabela de cores, que realmente são utilizados pela imagem.
- “biClrImportant” especifica o número de cores, da tabela de cores, que são considerados importantes para uma visualização correcta da imagem.

```
typedef struct
{
    CHAR    rgbBlue;
    CHAR    rgbGreen;
    CHAR    rgbRed;
    CHAR    rgbfiller;
} RGBQUAD;
```

Figura 41: Estrutura RGBQUAD (tabela de cores).

A estrutura RGBQUAD é simplesmente um vector contendo a definição RGB das cores utilizadas na imagem. A cada valor possível de intensidade um pixel, irá corresponder um elemento deste vector.

A parcela final do DIB é constituída pela matriz de pixels que constituem a imagem. Esta é constituída por “nHeight” linhas em ordem inversa, isto é, a primeira linha do ficheiro corresponde à última linha da imagem. Cada linha tem “nWidth” pixels, o que corresponde em bits, à largura da imagem multiplicada pelo número de bits por pixel.

Este valor é ainda aproximado para o número superior múltiplo de 32, tornando as linhas alinhadas à palavra de 4 bytes.

```
typedef struct
{
    CHAR        bfld[2];
    LONG INT    bfFileSize;
    INT         bfReserved1 [2];
    LONG INT    bfHeaderSize;
} BITMAPFILEHEADER;
```

Figura 42: Estrutura BITMAPFILEHEADER.

Um ficheiro com o formato BMP tem exactamente a mesma estrutura referida para o DIB, antecedida por 14 bytes, representados no SDK pela estrutura de cabeçalho dum ficheiro BMP, "BITMAPFILEHEADER" (Figura 42). Esta estrutura contém a seguinte informação:

- "bfld" contém, em BMPs Windows¹⁵, os códigos ASCII das letra "B" e "M" (indicativos de "Bitmap")
- "bfFileSize" representa o tamanho do ficheiro, em bytes.
- "bfReserved" contém quatro bytes reservados.
- "bfHeaderSize" representa o número de bytes do cabeçalho do "Bitmap", ou seja, o número de bytes desde o início do ficheiro até ao início da matriz de pixels. Este valor corresponde à soma das dimensões das estruturas de cabeçalho dum ficheiro BMP (14) e dum mapa de bits DIB (40), com o tamanho da tabela de cores da imagem (dependente do número de bits por pixel e do número de cores usadas).

As estruturas "BITMAPFILEHEADER" e "BITMAPHEADER" têm um comprimento fixo, pelo que, funções de leitura básicas são suficientes para realizar o seu tratamento. A leitura da estrutura "RGBQUAD" depende do tipo de tabela de cores utilizada, apresentando-se também muito simples. Na maioria dos "Bitmaps", como foi analisado anteriormente, esta tabela é um vector de estruturas RGB e na leitura do "Bitmap" que se segue, cada valor de um pixel será um índice para o campo deste vector que indica a cor (RGB) que esse valor representa. A leitura da matriz de pixels final é também directa bastando tomar em consideração a inversão de linhas mencionada.

¹⁵ Em Bitmaps de outros sistemas contém outras designações

3.3.4.2 Formato BIFF (“Blab Image File Format”)

O formato BIFF¹⁶ foi utilizado no sistema desenvolvido de forma a obter uma total compatibilidade com rotinas desenvolvidas para o sistema de processamento de imagem XITE [Lonnestad92] que corre em ambiente UNIX. Embora todos os ficheiros de imagem sejam gravados externamente em disco seguindo o formato BMP, internamente muitas das operações do programa utilizam um formato muito semelhante ao BIFF, pelo que, será fornecida neste capítulo, uma breve descrição deste formato de imagem.

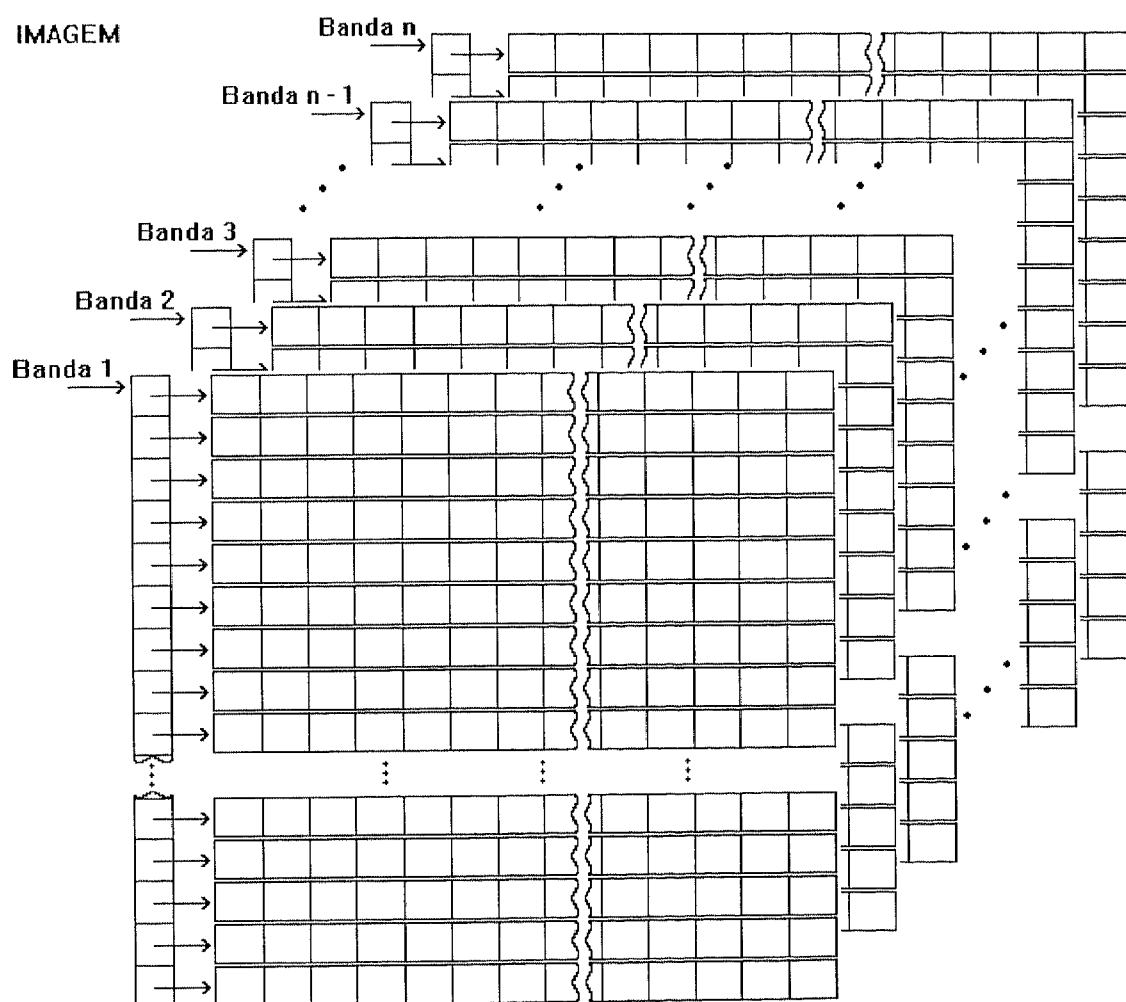


Figura 43: Constituição de uma imagem com formato BIFF.

No sistema XITE, uma imagem pode ser constituída por várias bandas (numeradas de 1 a n), podendo cada banda possuir dimensões e tipo de pixels próprios. Uma banda no contexto deste sistema é considerada como sendo uma matriz rectangular de pixels todos possuindo o mesmo tipo de dados. De forma a identificar

¹⁶ “Blab Image File Format”

cada pixel numa dada banda é utilizado um sistema de coordenadas básico com dois eixos (X e Y). O pixel superior esquerdo possui as coordenadas (1,1) e o pixel inferior direito, as coordenadas (Xsize,Ysize).

Ignorando as informações genéricas respeitantes às imagens e bandas (títulos, dimensões, sistemas de coordenadas, tipos de pixeis, etc), as imagens são armazenadas internamente em memória, como um vector de apontadores para bandas. Cada uma destas bandas (Figura 43), é por sua vez constituída por um vector de apontadores (com uma dimensão idêntica ao número de linhas da banda) para os vectores que constituem as linhas da banda. Cada um dos vectores representando uma linha da banda, possui uma dimensão idêntica ao número de colunas da banda e é constituído por elementos de igual tipo, onde se encontram armazenados os valores dos pixeis da imagem.

Muito frequentemente as imagens possuem mais do que uma banda. Diferentes bandas podem representar a mesma área espacial ou diferentes áreas espaciais, podendo ter idênticas ou distintas resoluções espaciais. No sistema XITE, esta informação é armazenada utilizando um sistema de coordenadas global. Para cada banda são utilizados os parâmetros Xmag e Ymag para descrever os factores de ampliação segundo as direcções X e Y de cada banda. São ainda utilizados os parâmetros Xstart e Ystart para descrever as coordenadas relativas, no sistema global de coordenadas, do ponto (1,1) de cada banda (Figura 44).

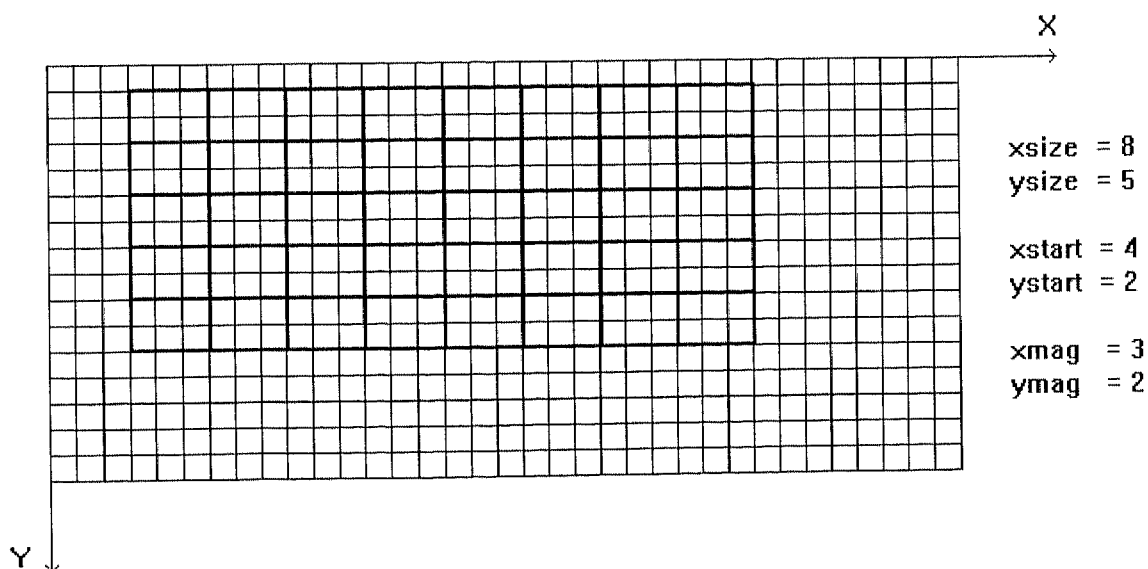


Figura 44: Especificação da relação entre o sistema de coordenadas de uma banda e da imagem.

Conceptualmente, no sistema XITE, uma imagem contém três partes principais [Lonnestad92], “info-data”, “band-data” e “block-data”. O bloco de dados “info-data” contém informação diversa, numérica e textual, acerca da imagem em análise. O bloco de dados “band-data” contém a informação numérica respeitante à intensidade dos pixels na imagem. Cada imagem pode ser constituída por várias bandas, podendo cada banda possuir dimensões e tipo de pixels próprios. Finalmente, o bloco “block-data” contém informação livremente introduzida pelo utilizador. Não existe qualquer restrição à dimensão ou tipo de informação colocada neste bloco.

A informação contida no campo “info-data”, descritiva da imagem e das suas bandas é a seguinte:

- TITLE Pequeno texto descrevendo a imagem.
- NBANDS Número de bandas contidas na imagem.
- PARAM[8] Oito parâmetros tipo inteiro que podem ser definidos e utilizados livremente pelo utilizador.
- BANDARRAY Informação numérica respeitante a cada banda, contendo:
 - PIXTYPE Tipo de pixels da banda. Pode ser um dos seguintes tipos: “unsigned byte”, “signed byte”, “unsigned short”, “signed short”, “integer”, “real”, “complex”, “double real” e “double complex”.
 - XSIZE Tamanho horizontal da banda.
 - YSIZE Tamanho vertical da banda.
 - XSTART Posicionamento horizontal da banda.
 - YSTART Posicionamento vertical da banda.
 - XMAG Ampliação horizontal da banda.
 - YMAG Ampliação vertical da banda.
- NCHARS Número de caracteres contidos no campo seguinte (“TEXTDATA”).
- TEXTDATA Descrição opcional mais completa da imagem.
- NBLOCKS Número de blocos de 512 bytes no “block-data”.

O bloco de informação “band-data” contém ordenadamente a informação respeitante aos valores dos pixels de cada uma das bandas da imagem. Existem na imagem “NBANDS” bandas, cada uma podendo possuir uma dimensão, tipo de pixels e

sistema de coordenadas próprio. O bloco “block-data” contém “NBLOCKS” de 512 bytes contendo informação da exclusiva responsabilidade do utilizador.

3.3.5 Interface com o Utilizador

A interface com o utilizador foi desenvolvida de acordo com as normas gerais de definição de interfaces homem-máquina [Cox93, Pyle95] , nomeadamente de acordo com as especificações para as interfaces Windows [Petzold90, Adler90b]:

- Informação visualizada graficamente totalmente compreensível, sem permitir interpretações erradas por parte do operador.
- Utilização de menus, caixas de diálogo e botões para permitir a introdução de comandos do utilizador.
- Utilização de janelas sobreponíveis na visualização da informação.
- Utilização simultânea do teclado e do rato na introdução de comandos.
- Utilização de terminologia “standard” em todos os menus e caixas de diálogo (aquela a que os utilizadores estão habituadas nas outras aplicações Windows).
- Entrada de dados robusta minimizando a introdução de valores errados.
- Permissão de total controlo do funcionamento da aplicação ao utilizador.
- Grande interactividade e intuitividade na introdução de comandos e realização de acções.
- Preocupações de clareza, associando cada comando a uma acção única.
- Preocupações estéticas de forma a tornar a interface agradável.
- Utilização de auxílio “on-line” e adequado ao contexto de forma a guiar o utilizador na utilização do programa.

O aspecto geral da interface implementada para o sistema pode ser visualizado na Figura 45.

A aplicação apresenta uma interface típica Windows para múltiplos documentos. A janela principal contém um menu de sistema, botões de minimização e maximização, barras de menu, de ferramentas e de estado, uma área de cliente contendo múltiplos documentos (imagens) e barras de “scroll” horizontais e verticais.

Na figura é visível a utilização de uma barra de menu típica, onde a grande maioria dos comandos são semelhantes aos comandos típicos de qualquer aplicação Windows. Entre estes destacam-se os comandos contidos nos menus “Ficheiro”,

“Edição”, “Opções”, “Janela” e “Help”. Em outros menus, os comandos são mais específicos do tipo de aplicação construída, ou seja, de uma aplicação com objectivos genéricos de processamento e análise de imagem e específicos de detecção de arestas e contornos.

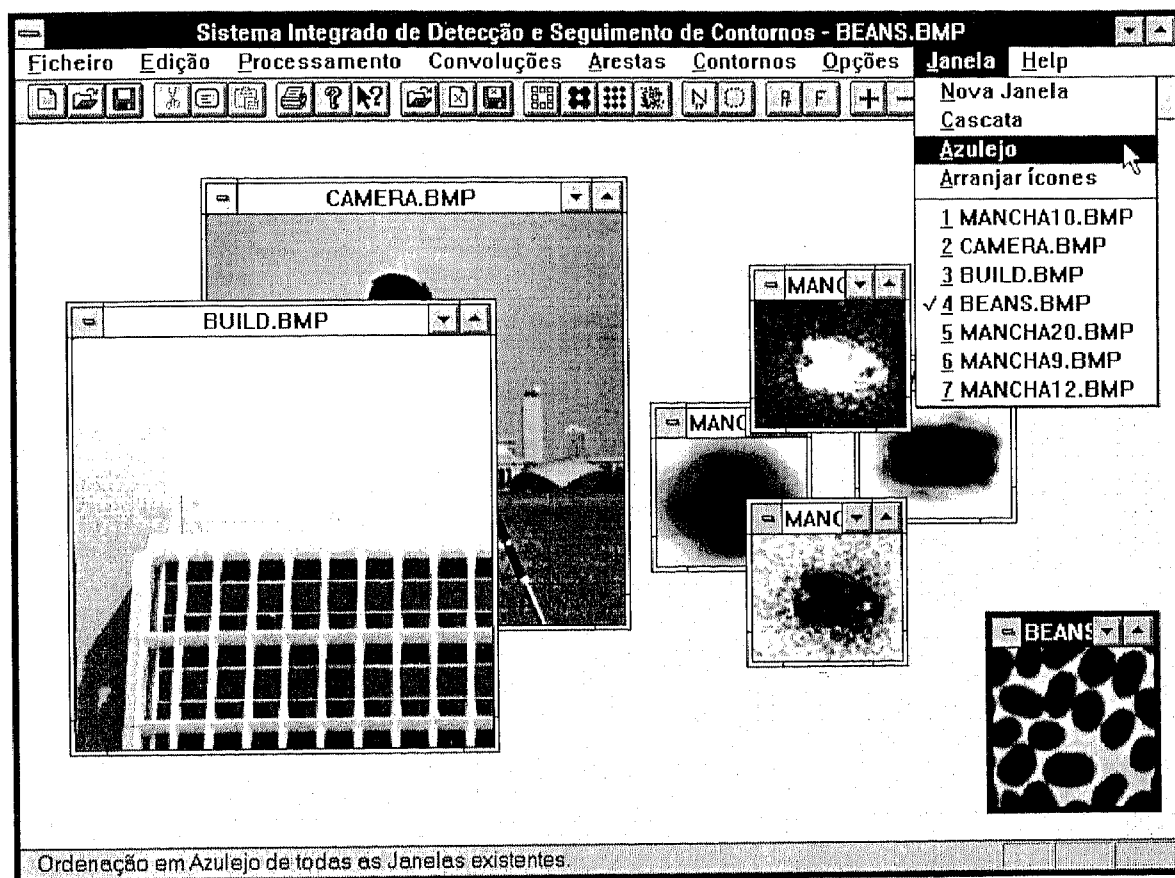


Figura 45: Aspecto geral do sistema de detecção de contornos implementado.

Situada imediatamente abaixo da barra de menus, situa-se a barra de ferramentas. Esta barra contém um conjunto de botões, agrupados de forma lógica, que permitem ao utilizador a realização automática das acções mais frequentes. Cada botão contém uma imagem apropriada ao comando que permite executar. Também nesta barra foi mantida a terminologia “standard”, através da utilização nos botões, de imagens comuns a outras aplicações Windows.

Na parte inferior da janela pode ser verificada a existência de uma barra de estado com funções semelhantes às suas equivalentes em outras aplicações Windows. Sempre que um determinado comando é iluminado, na barra de estado, aparece um auxílio relativo ao significado desse comando, permitindo ao utilizador uma decisão mais apoiada sobre a sua execução ou não. Sempre que uma determinada acção é executada

pelo programa, na barra de estado aparece uma mensagem indicativa do significado dessa acção.

Na área de cliente da aplicação podem ser visualizadas múltiplas imagens e sobre estas podem ser visualizados contornos e filmes de contornos. A dimensão de cada uma das imagens pode ser alterada livremente pelo utilizador. Para uma determinada imagem, podem ser abertas diversas janelas com vistas diferentes. Este procedimento é visível na Figura 39 onde a imagem contendo o fotógrafo, é visualizada em três janelas distintas. As imagens visualizadas na área de cliente podem ser dispostas em forma de cascata, azulejo (Figura 46) ou minimizadas para uma forma tipo ícone.

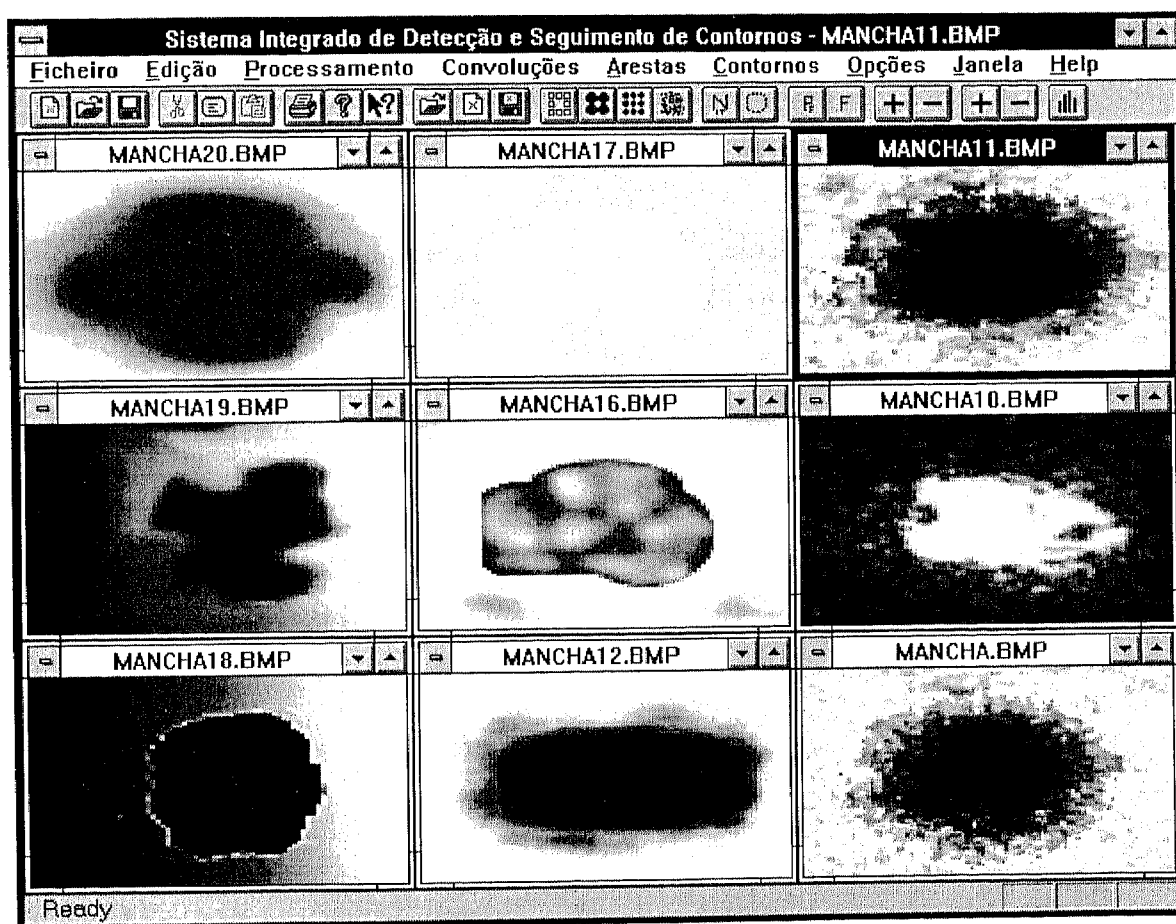


Figura 46: Disposição das imagens visualizadas no écran em azulejo.

Os menus disponibilizados na aplicação são os seguintes:

- Ficheiro: Contendo os comandos de criação, leitura e gravação de ficheiros e contornos e de configuração da impressora, previsão e realização de impressão de imagens.

- Edição: Contendo as operações típicas de qualquer aplicação Windows, "Undo", "Cut", "Copy" e "Paste".
- Processamento: Contendo diversas operações genéricas de processamento digital de imagem, descritas na secção seguinte.
- Convoluções: Menu de acesso rápido a operações de filtragem comuns tais como Gaussianos, Laplacianos, filtragem LoG, gradientes (de Sobel, Prewitt e Roberts).
- Arestas: Onde são disponibilizadas operações de detecção de arestas segundo diversos métodos.
- Contornos: Contendo operações de definição, visualização e movimentação de contornos, descritas em pormenor adiante.
- Opções: Contendo opções de visualização e de selecção de um operando genérico ou uma máscara para operações de processamento.
- Janela: Opções comuns de criação de janelas e disposição em cascata, azulejo ou ícone de janelas.
- "Help": Menu comum às aplicações Windows contendo auxílio genérico e adequado ao contexto e o quadro com a descrição da aplicação e do autor.

3.3.6 Operações Genéricas de Processamento de Imagem Disponibilizadas

Na versão actual do programa são disponibilizadas rotinas de processamento de imagem incluídas nas seguintes categorias:

- Operações Lógicas ("Not", "And", "Or" e "Xor" [Rosenfeld82])
- Operações Aritméticas (Adição, Subtracção, Multiplicação, Divisão [Rosenfeld82])
- Operações Morfológicas (Dilatação, Erosão, Abertura, Fecho, "Hit or Miss" [Serra82])
- Filtragens Não Lineares (Mediana, Mínimo, Máximo e "Closest" Máximo ou Mínimo [Schalkoff89])
- Filtragens Lineares (Convoluções com núcleo definido pelo utilizador, Gaussianos e Laplacianos de dimensão variável e Gradientes diversos, incluindo Sobel, Prewitt e Roberts [Schalkoff89])
- Limiarização (Seleção de Nível [Rosenfeld82], Otsu [Otsu78] e "Multi Level Reddi" [Sahoo88])
- Construção de Histogramas [Rosenfeld82].
- Alteração do Brilho e Contraste [Rosenfeld82].

- Esqueletização (“Bones” e “ThinKxK” [O’Gorman90])
- Introdução de Ruído (Gaussiano ou aleatório, aditivo ou multiplicativo).

Embora algumas das operações tenham sido desenvolvidas de raiz especificamente para o sistema (alteração de brilho e contraste, limiarizações, construção de histogramas e introdução de ruído), a maioria das operações disponibilizadas (filtragens, esqueletizações, operações lógicas, aritméticas e morfológicas) foram adaptadas do sistema XITE. Nesta perspectiva foram desenvolvidas operações de conversão de formatos e implementados quadros diversos de introdução de dados e visualização de resultados.

A relativa simplicidade das operações descritas nesta secção e o facto de constituírem operações genéricas comuns a qualquer sistema de processamento de imagem, conduz a que seja dispensável a apresentação de uma análise mais profunda da sua implementação. Nas referências apresentadas podem encontrar-se descrições suficientemente detalhadas destas operações para que a sua implementação possa ser facilmente compreendida.

3.3.7 Operações de Detecção de Arestas

Uma vez que as operações de detecção de arestas são de enorme utilidade na detecção de contornos, foram incluídas no sistema as seguintes operações:

- Detecção de arestas através de Filtragem LoG [Marr82].
- Detecção de arestas utilizando o filtro de Canny [Canny86].
- Detecção de arestas utilizando o filtro de Canny-Deriche [Deriche87].
- Detecção de arestas utilizando o filtro de Spacek [Spacek91].
- Detecção de arestas utilizando o filtro de Shen e Castan [Shen92].

Estas rotinas foram adaptadas do trabalho de Correia [Correia95] e do sistema XITE e seguem os métodos tradicionais de implementação mencionados no capítulo 2. Para estas operações foram desenvolvidos módulos de introdução de dados (de forma a definir interactivamente os parâmetros de cada filtro) e de pré e pós processamento.

3.3.8 Operações de Definição e Detecção de Contornos

O principal objectivo específico do sistema desenvolvido é exactamente a detecção e seguimento de contornos utilizando contornos activos. Desta forma, foram incluídas no sistema as seguintes funcionalidades:

- Definição inicial interactiva de um contorno.
- Definição de propriedades do contorno e do algoritmo de movimentação.
- Movimentação do contorno utilizando técnicas de contornos activos.
- Análise do contorno e extracção de medidas quantitativas.
- Visualização do filme com a movimentação do contorno.
- Armazenamento de filmes para posterior visualização e análise.

A análise detalhada do funcionamento de cada uma das operações mencionadas será realizada na secção seguinte.

3.4 DETECÇÃO E SEGUIMENTO DE OBJECTOS UTILIZANDO CONTORNOS ACTIVOS

3.4.1 Introdução

Nas secções anteriores foi analisado o software desenvolvido no que diz respeito às suas potencialidades genéricas, tendo sido abordados tópicos tais como a arquitectura e implementação do sistema, os formatos de imagem utilizados, a interface com o utilizador e as operações de processamento de imagem disponibilizadas. Nesta secção serão abordadas as funcionalidades específicas do software, no que concerne à detecção de contornos utilizando contornos activos.

3.4.2 Método de Definição do Contorno Inicial

A definição do contorno inicial, como foi analisado na secção 2.5.3, é uma operação com grande importância nos resultados finais obtidos. Embora a utilização de algoritmos hierárquicos e da força de enchimento de contorno permita minorar a dependência dos resultados finais, em relação à definição do contorno inicial, este tem sempre que ser definido de forma a que o seu movimento não possa ser travado por falsas arestas, que não pertençam ao contorno desejado.

No pacote de software desenvolvido foram incluídas duas formas de definição manual do contorno inicial:

- Definição de diversos pontos de controlo, seguida de uma aproximação poligonal dos pontos de forma a obter o contorno inicial.

- Definição de uma elipse inicial, ajustável posteriormente através de forças manuais de atracção e repulsão.

A definição de um contorno utilizando a primeira técnica encontra-se ilustrada na Figura 47. Nesta técnica é efectuada uma colocação sucessiva de diversos pontos de controlo sobre a imagem, seguida de uma indicação de finalização da definição do contorno. Nesta altura, os pontos de controlo definidos são utilizados para definir, através de uma interpolação, os restantes pontos de controlo desejados.

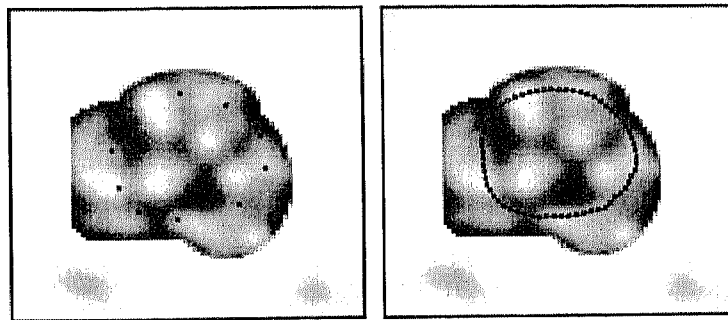


Figura 47: Definição de 10 pontos de controlo (esquerda) e interpolação de forma a obter os 64 pontos desejados (direita).

A segunda técnica de definição de um contorno baseia-se no traçado de uma forma simples inicial (elipse) seguida do seu ajuste progressivo através de forças de repulsão e atracção dos pontos desse contorno. Na Figura 48 é visível a forma de definição inicial do contorno e a sua discretização no número de pontos desejado, efectuada automaticamente pelo programa.

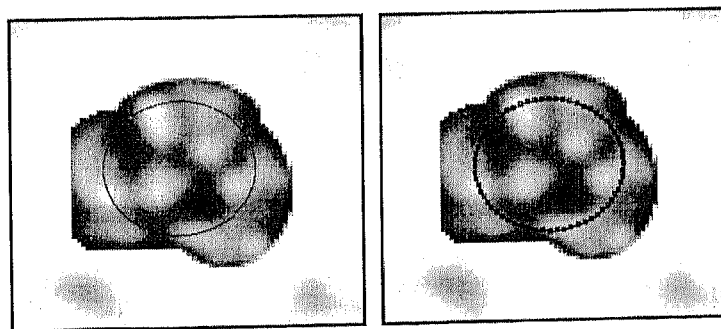


Figura 48: Definição de uma elipse inicial (esquerda) e sua discretização em 64 pontos formando o contorno inicial (direita).

Após discretizada a elipse, o seu ajuste pode ser efectuado utilizando forças de atracção (Figura 49) introduzidas através da pressão do botão esquerdo do rato e de forças de repulsão (Figura 50) introduzidas através da pressão do botão direito do rato.

Através da utilização sucessiva deste tipo de forças é possível definir facilmente um contorno inicial com a forma desejada.

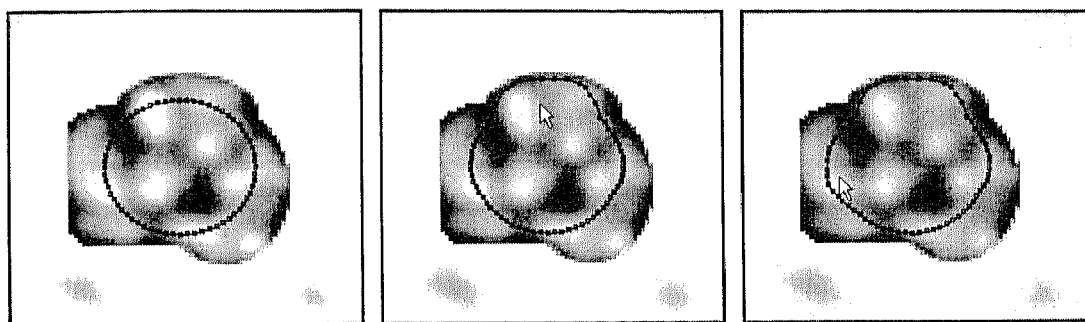


Figura 49: Introdução de forças de repulsão para ajuste do contorno, utilizando o rato.

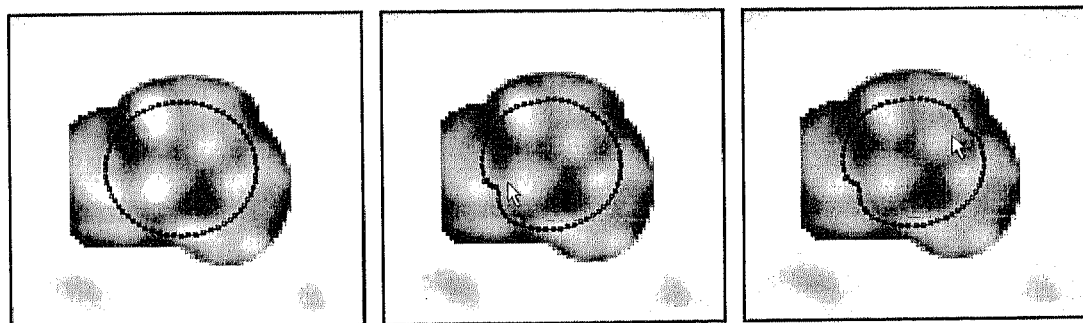


Figura 50: Introdução de forças de atracção para ajuste do contorno, utilizando o rato.

As forças de ajuste (repulsão e atracção) podem ser utilizadas em ambos os métodos de definição do contorno mencionados. A partir do momento em que o contorno inicial se encontra definido e discretizado, ambas as forças podem ser utilizadas para obter uma melhor definição do seu formato.

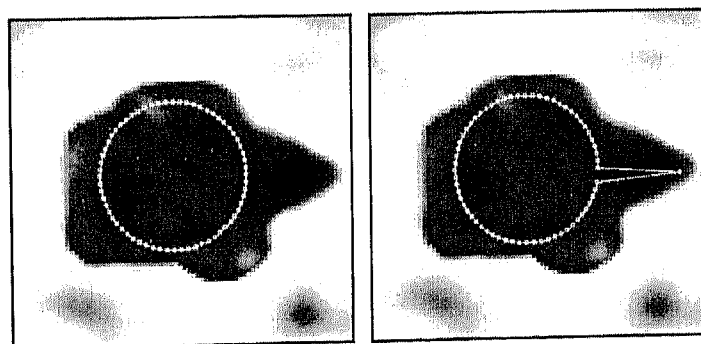


Figura 51: Movimentação directa dos pontos do contorno através do método "Drag and Drop".

Outro método de melhoria do contorno inicial consiste na movimentação directa dos pontos que o constituem. Esta movimentação pode ser executada, de forma muito simples, através do vulgar método “drag and drop”. Este método consiste em pressionar o botão esquerdo do rato sobre o ponto desejado, seguido da sua posterior movimentação para a localização pretendida (Figura 51).

Um método de grande utilidade no ajuste da posição inicial do contorno, consiste na interpolação do contorno definido, de forma a espaçar igualmente todos os seus pontos. Na Figura 52 é visível um exemplo de aplicação desta técnica. Após a definição de um contorno inicial elíptico, um dos seus pontos foi “arrastado” de forma a definir um “pico”, após o que, foi realizada uma operação de interpolação com espaçamento equitativo dos pontos.

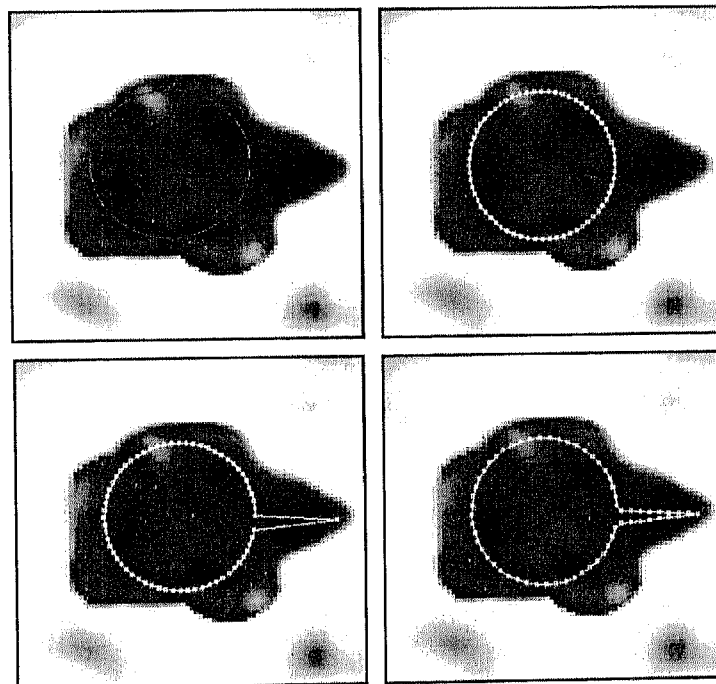


Figura 52: Espaçamento equitativo dos pontos constituintes do contorno: Definição de um contorno e sua discretização (em cima). Movimentação de um ponto e resultado obtido na interpolação (em baixo).

Utilizando em conjunto todas as técnicas descritas é extremamente simples definir interactivamente um contorno inicial de boa qualidade, adequado ao método iterativo de ajuste implementado.

3.4.3 Representação Interna de Contornos e Filmes de Contornos

Um contorno activo na implementação efectuada, foi definido como uma lista de pontos circular, duplamente ligada. Um filme contendo a movimentação de um contorno ao longo do processo iterativo é constituído por uma lista ligada de apontadores para contornos. A definição interna utilizada para os contornos e filmes pode ser visualizada na Figura 53.

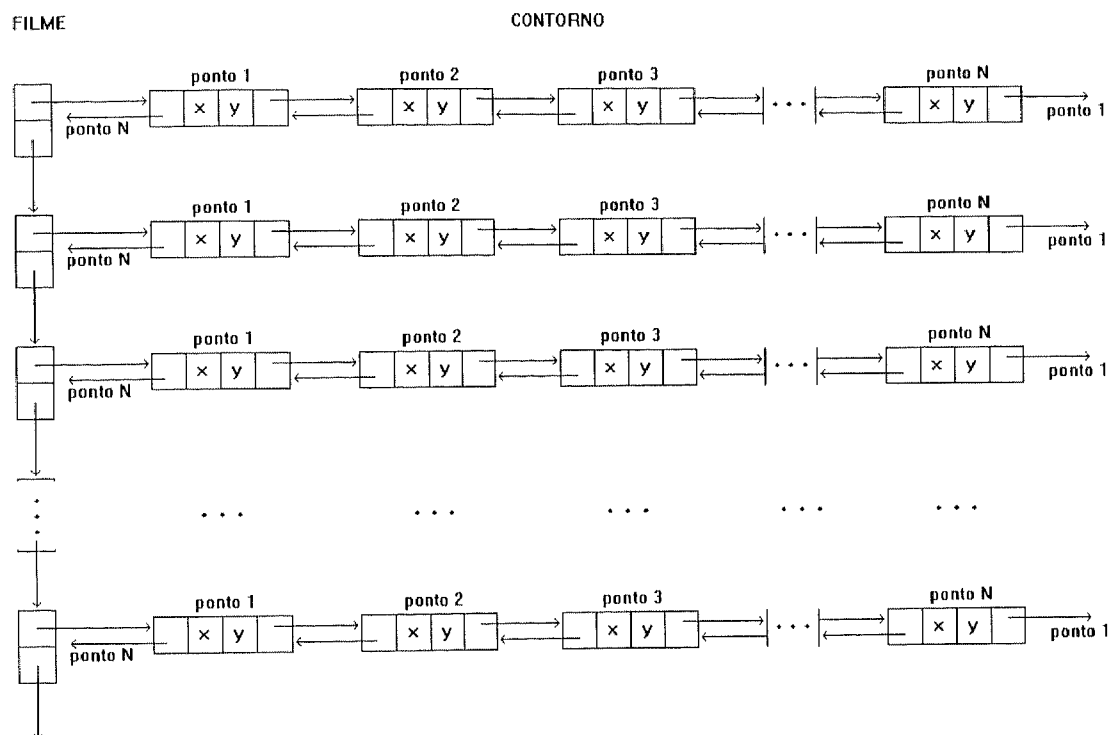


Figura 53: Definição interna de contornos e filmes.

Na figura é visível que cada ponto de um contorno contém as suas coordenadas (x,y) e apontadores para os pontos anterior e seguinte do contorno. Relativamente a cada contorno são ainda armazenados parâmetros relativos às suas propriedades físicas, à sua evolução temporal e ao algoritmo utilizado no seu movimento.

No que diz respeito aos filmes de contornos, cada filme é constituído por uma lista ligada de apontadores para contornos. Para além desta lista é ainda incluída informação relativa ao número de "frames" contidas no filme e à sua velocidade de projecção desejada.

3.4.4 Estrutura Global do Método de Detecção de Contornos

O método de detecção de contornos implementado no sistema baseou-se essencialmente nas ideias apresentadas no capítulo 2 deste trabalho. No entanto, foram definidas algumas características próprias com o intuito de o tornar mais robusto e adequado às imagens a processar. A estrutura global do método desenvolvido pode ser visualizada na Figura 54.

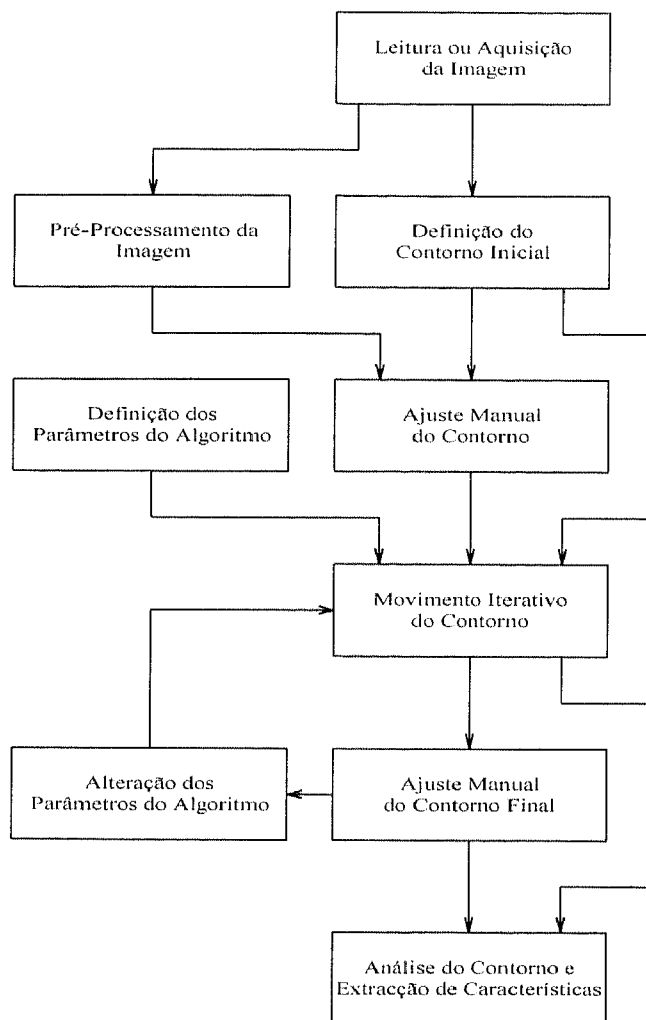


Figura 54: Estrutura global do método de detecção de contornos desenvolvido.

A primeira fase do método consiste na leitura da imagem e alteração do seu formato para um dos formatos internos de tratamento do programa (DIB ou BIFF). A

aquisição de imagens, devido ao problema referido anteriormente¹⁷, não foi incluída na versão final do software.

Após a leitura das imagens, são disponibilizadas operações de pré-processamento com vista a efectuar a sua adequação ao tratamento a que vão ser submetidas. De entre as operações com maior utilidade destacam-se os filtros não lineares e morfológicos. Por vezes é também desejável realizar aumentos ou diminuições de brilho e contraste de forma a realçar melhor à vista humana as características mais relevantes das imagens.

A primeira operação directamente relacionada com a detecção de contornos consiste em definir uma localização inicial do contorno. Após definida esta localização (da forma analisada na secção 3.4.2), o contorno é discretizado num conjunto finito de pontos que podem ser ajustados pelo utilizador, individualmente ou utilizando forças de atracção e repulsão.

A fase mais importante do método consiste na movimentação iterativa do contorno utilizando o algoritmo dos contornos activos. Esta fase é antecedida pela definição de todos os parâmetros do algoritmo de movimentação. Uma vez definidos estes parâmetros e fornecido o comando de início da operação de movimentação, o contorno vai-se deslocar sucessivamente adaptando-se às características da imagem.

No final do processo iterativo, o utilizador pode visualizar o filme contendo a movimentação do contorno e alterar interactivamente o contorno final obtido. Se o desejar, o utilizador pode ainda definir novos parâmetros para o algoritmo de movimentação e recommençar, a partir de uma determinada iteração, o processo de movimentação automática.

Após obtido um contorno final com as características desejadas, a fase final do processo, consiste na análise desse contorno e na extracção das medidas quantitativas desejadas.

3.4.5 Algoritmo de Movimentação do Contorno

O algoritmo de movimentação do contorno implementado opera de uma forma hierárquica sobre uma estrutura multi-resolução contendo informação relativa às forças do Gradiente da Imagem, Forças do Detector de Arestas, Forças de Enchimento e definições das Matrizes Características dos contornos.

¹⁷

Avaria da Placa de Aquisição de Imagem que se encontrava associada ao projecto.

A adopção de um algoritmo hierárquico, aliás já sugerida por Kass no seu trabalho precursor [Kass88] deve-se aos bons resultados que ao longo dos últimos anos tem fornecido estas estruturas nas diversas áreas do processamento e análise de imagem [Burt88a, Burt88b, Kervrann93]. As estruturas hierárquicas de dados têm também representado um papel de extrema importância em domínios tais como os sistemas gráficos, a inteligência artificial ou a robótica. Baseiam-se no princípio da divisão recursiva do espaço em segmentos de dimensão sucessivamente menor, permitindo assim focar a atenção para subconjuntos de dados de dimensão mais limitada [Campilho93].

A estrutura multi-resolução proposta (Figura 55) apresenta características diferentes das estruturas mais comuns aplicadas na área do processamento e análise de imagem. Baseia-se em utilizar nos níveis mais baixos da hierarquia, imagens força altamente suavizadas, altas forças de enchimento e contornos com elevada rigidez. Nos níveis hierárquicos mais elevados, as imagens força são muito pouco suavizadas, as forças de enchimento são baixas ou nulas e os contornos apresentam menor elasticidade e rigidez.

No final do processo de pré-processamento, obtemos uma estrutura multi-resolução contendo as seguintes subestruturas:

- Forças do Gradiente suavizadas a **n** níveis.
- Forças do Detector de Arestas a **m** escalas, suavizadas a **n** níveis.
- Matriz de Características do contorno a **p** níveis.
- Força de Enchimento do contorno a **q** níveis de intensidade.

A estrutura apresentada não é automaticamente calculada no início do processo. As matrizes de características e as forças de enchimento a diversos níveis vão sendo calculadas conforme são necessárias ao processo iterativo de movimentação do contorno. Por exemplo, no início do processo iterativo, não existem dados que permitam calcular as forças de enchimento para todo esse processo. De facto, estas forças, ao contrário do que acontece com as forças do gradiente e do detector de arestas, são definidas unicamente para cada um dos pontos do contorno e não de força global para a imagem em análise. No que diz respeito às matrizes de características do contorno, estas podiam facilmente ser calculadas no início do processo uma vez que dependem unicamente das características físicas desejadas para o contorno. No entanto, na implementação desenvolvida, optou-se pelo seu cálculo unicamente quando se apresentam necessárias para o processo iterativo de movimentação.

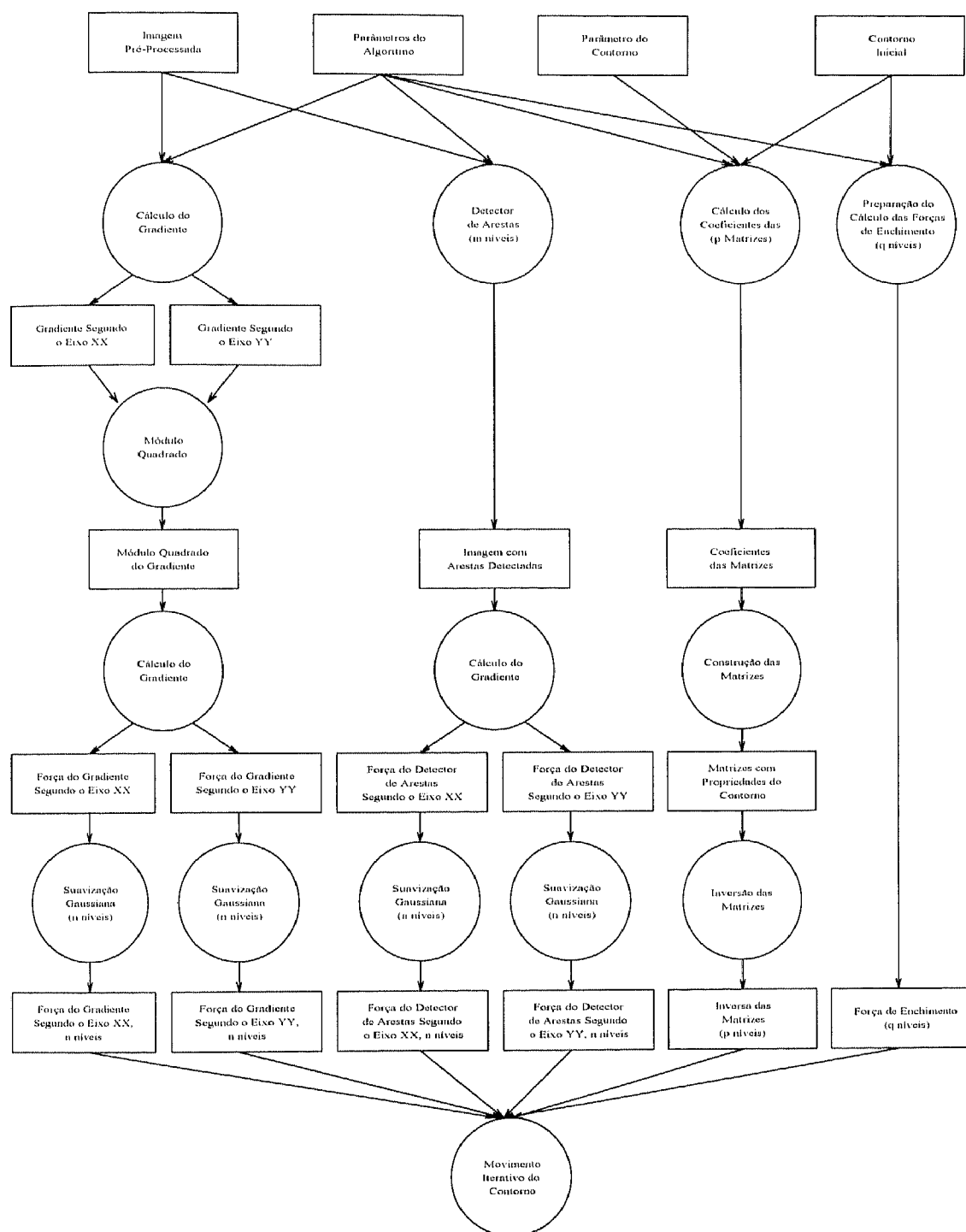


Figura 55: Cálculo da estrutura de dados multi-resolução proposta.

No que diz respeito às forças do gradiente e do detector de contornos, estas são calculadas no início do processo de acordo com os parâmetros especificados para o algoritmo de movimentação. Estas forças não dependem da definição do contorno nem das suas propriedades, podendo desta forma ser facilmente calculadas no início do processo de acordo com os parâmetros especificados para o algoritmo.

A utilização das estruturas definidas baseia-se fundamentalmente nos seguintes princípios:

- Contornos iniciais mais rígidos e elásticos conduzem a uma movimentação inicial mais adequada. Desta forma, convém iniciar o movimento com maiores coeficientes de elasticidade e de rigidez. Progressivamente, conforme o contorno se for aproximando da solução final, estes valores podem ser diminuídos, permitindo a adaptação do contorno às regiões de maiores curvaturas do contorno final.
- Forças de enchimento fortes são necessárias no início do processo de movimentação mas são indesejáveis no final do processo de movimentação. De facto, se a “snake” for inicializada no interior do objecto cujo contorno se pretende detectar e se, o brilho desse interior for relativamente constante, forças de enchimento consideráveis são necessárias para deslocar o contorno para próximo da solução desejada. No entanto, uma vez atingida a região onde se encontra a solução, a força de enchimento torna-se perfeitamente dispensável e até mesmo indesejável. Esta força leva a que o equilíbrio energético se produza para um contorno final ligeiramente mais alargado do que o desejado pois todos os pontos do contorno, no equilíbrio, recebem uma força extra que é perpendicular a este.
- Forças do gradiente da imagem suavizadas, utilizando uma máscara de grandes dimensões, de forma a se espalharem ao longo da imagem, são bastante adequadas ao início do processo. No entanto, na fase de detecção precisa do contorno final, é desejável que as forças do gradiente sejam localizadas com precisão nas regiões onde se verificam as maiores variações de intensidade na imagem.
- Forças do detector de arestas resultantes de detectores de maior escala¹⁸, bastante suavizadas, de forma a se espalharem ao longo da imagem, são bastante adequadas ao início do processo. No entanto, na fase de detecção precisa do contorno final, é desejável que as forças do detector de arestas sejam resultantes de detectores de menor escala¹⁹, com uma menor suavização. Desta forma, as forças serão localizadas de forma mais precisa nas arestas a detectar, permitindo na fase final do processo uma detecção com maior precisão do contorno.

Os parâmetros que o utilizador pode especificar para o algoritmo, como é visível na Figura 56, incluem:

- NRES Número de imagens resultado pretendidas (contendo a evolução do contorno ao longo do processo iterativo).

¹⁸ Com detecção unicamente das arestas mais fortes

¹⁹ Com detecção de uma maior parcela das arestas

- NITER Número de iterações a efectuar pelo algoritmo entre a obtenção de cada imagem resultado (o número de iterações final será igual a $NRES \cdot NITER$).
- ERROMAX Critério de paragem através da obtenção de um erro (movimento médio dos pontos entre duas iterações consecutivas) inferior ao máximo admissível.
- W1 Coeficiente de elasticidade do contorno.
- W2 Coeficiente de rigidez do contorno.
- H Passo de discretização no espaço.
- G Passo de discretização no tempo.
- NPONTOS Número de pontos do contorno.
- KFGRAD Coeficiente da força do gradiente da imagem.
- KFAREST Coeficiente da força do detector de arestas.
- KFENCH Coeficiente da força de enchimento.
- FACTSUAV Factor máximo de suavização das imagens força.
- FACTAREST Factor máximo de escala do detector de arestas.
- NNIVENCH Número de níveis da força de enchimento considerados.
- NNIVSUAV Número de níveis de suavização das imagens força.
- NNIVAREST Número de escalas do detector de arestas.
- NNIVRIG Número de níveis de rigidez do contorno considerados.
- KFENCHMIN Coeficiente da força de enchimento no último nível do processo.
- FACTSUAVMIN Factor mínimo de suavização das imagens força (no nível de resolução final).
- FACTARESTMIN Factor mínimo de escala do detector de arestas (no nível de resolução final).
- W1MIN Coeficiente de elasticidade do contorno no nível final de elasticidade (por defeito idêntico a W1).
- W2MIN Coeficiente de rigidez do contorno no nível final de rigidez (por defeito idêntico a W2)
- ERROSUAV Erro máximo admissível antes da passagem para o nível seguinte de suavização das imagens força.
- ERRORIG Erro máximo admissível antes da passagem para o nível seguinte de rigidez do contorno.

- **ERROENCH** Erro máximo admissível antes da passagem para o nível seguinte de força de enchimento do contorno.

Parâmetros do Contorno			
Coefficientes do Contorno:		Discretização:	
Elasticidade (w1):	2.433e-04	Espaço (h):	1.56e-002
Rigidez (w2):	5.922e-08	Tempo (g):	0.1
		Nº Pontos:	64
Processo Iterativo:		Coefficientes de Força:	
Nº Imag. Resultado:	10	Força da Imagem:	1.0
Nº Iterações:	20	Força Enchimento:	2.0
Erro Máximo Final:	1.0e-06	Força Det. Arestas:	1.0
Factores Espalhamento:		Localização:	
Factor de Suavização:	3	Cx:	40
Factor de Arestas:	0.5	Cy:	40
		Raio:	
		Rx:	30
		Ry:	30
		Parâmetros do Algoritmo Hierárquico:	
		Nº de Níveis do Algoritmo:	
		Nº Níveis Força Ench.	3
		Nº Níveis de Suavização	3
		Nº Níveis Det. Arestas	1
		Nº Níveis Rigidez	3
		Valores Finais do Algoritmo:	
		Força Enchimento Mínima:	0.0
		Factor de Suavização Mín.	0
		Factor de Det. Arest. Min.	0.5
		Elasticidade Mín. (w1min):	2.433e-04
		Rigidez Mín. (w2min):	1.217e-08
		Erros de Alteração:	
		Erro Dim. Força Enchimento	0.3
		Erro Dim. Fact. Suavização	0.3
		Erro Dim. Rigidez	0.3
OK Val. Defeito Par. Acons. Cancel			

Figura 56: Quadro de selecção dos parâmetros do algoritmo.

O algoritmo de movimentação propriamente dito, apresenta um formato muito simples. Inicialmente é calculada a estrutura multi-resolução contendo os vários níveis de resolução das imagens força e é copiado o contorno inicial definido pelo utilizador para a estrutura final que irá conter o filme da movimentação. É também construída a matriz de características do contorno e calculada a sua inversa.

O cálculo de cada um dos NRES contornos resultado é efectuado, calculando a movimentação do contorno ao longo das NITER iterações. Em cada uma das iterações, são calculadas em cada um dos pontos do contorno:

- Forças do gradiente da imagem segundo os eixos **XX** e **YY** (interpolação bilinear dos valores dos pixels vizinhos das imagens força correspondentes).
- Forças do detector de arestas segundo os eixos **XX** e **YY** (interpolação bilinear dos valores dos pixels vizinhos das imagens força correspondentes).
- Força de enchimento segundo os eixos **XX** e **YY** (através da decomposição nas coordenadas **x** e **y** do vector normal ao contorno no ponto considerado).

Todas as forças são pesadas utilizando os coeficientes definidos anteriormente. Para as forças do gradiente e detector de arestas, é utilizado o critério apresentado na

equação 65 de forma a ignorar a contribuição de gradientes muito reduzidos e normalizar a contribuição de gradientes consideráveis.

Cada um dos pontos é movimentado, utilizando as forças internas e externas a que se encontra sujeito, de acordo com o algoritmo dos contornos activos tal como descrito na secção 2.4.

No final do processo de movimentação é calculada a média de movimentação dos pontos do contorno na iteração considerada. Este valor é utilizado como base para os critérios de alteração do nível da estrutura de dados utilizado na movimentação. Sempre que o valor da média de movimentação é inferior a um dos valores de erro considerados (ERROSUAV, ERRORIG e ERROENCH), são alteradas em conformidade as estruturas de dados utilizadas no algoritmo.

Se o valor da média de movimentação, for inferior a ERROMAX ou o número máximo de iterações for atingido, o algoritmo termina a sua execução, sendo fornecida ao utilizador, a movimentação do contorno ao longo do processo iterativo, sob a forma de um filme. São ainda calculados diversos parâmetros quantitativos relativos ao contorno final obtido, destacando-se a sua área, cor média e perímetro.

3.4.6 Estrutura Global do Método de Seguimento de Contornos

O método de seguimento de contornos baseia-se essencialmente no método de detecção apresentado nas secções precedentes, associado ao método genérico de seguimento de contornos apresentado na secção 2.6. A estrutura do método desenvolvido encontra-se ilustrada na Figura 57.

Inicialmente é definido um contorno inicial ($v(s,1)$), sobre a primeira imagem da sequência (**Imagem(1)**). Esse contorno é ajustado aos contornos do objecto pretendido, utilizando o algoritmo de movimentação descrito nas secções precedentes.

A determinação do contorno do objecto na segunda imagem da sequência, é efectuada, utilizando como ponto de partida, o contorno obtido para a primeira imagem e aplicando o algoritmo desenvolvido.

A partir da terceira imagem, é possível corrigir a posição do contorno, utilizando informação relativa à sua posição em imagens precedentes. É efectuado o cálculo da velocidade instantânea de cada um dos pontos do contorno, verificada entre as duas imagens anteriores. Essa velocidade é utilizada para efectuar a correcção da posição dos pontos do contorno que vai ser utilizado como início para o processo iterativo seguinte. Utilizando a nova posição prevista, é executado o algoritmo de movimentação iterativa do contorno, de forma a obter a sua nova posição.

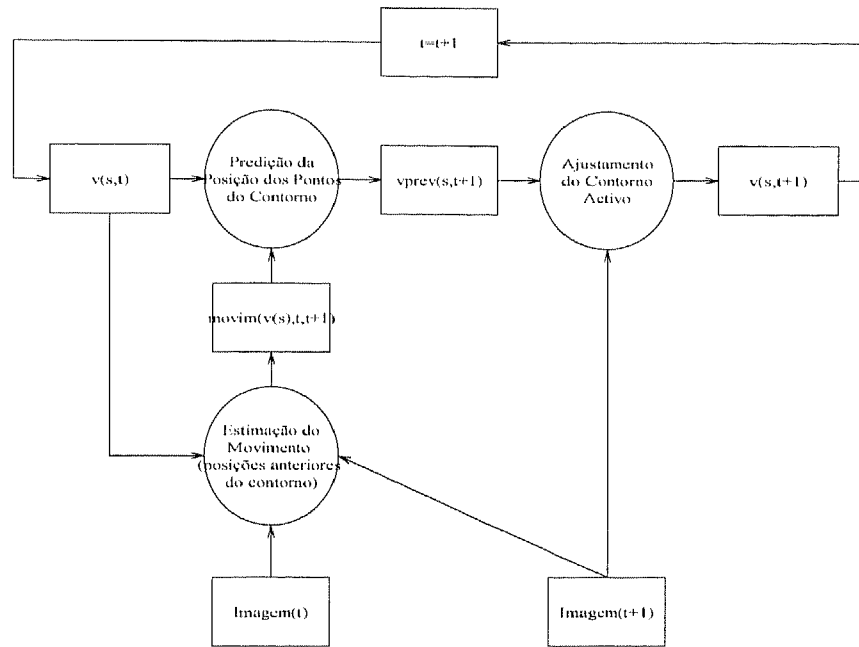


Figura 57: Estrutura global do método de seguimento de contornos desenvolvido.

O método desenvolvido possui uma estrutura bastante simples, não utilizando informação obtida por detectores de movimento convencionais, baseando-se unicamente nas diferenças entre os contornos de imagens sucessivas. Procura-se assim tirar partido da descrição de alto nível do contorno que é possível obter em cada imagem evitando-se deste modo ter de processar as imagens na sua globalidade para estimar campos de movimento globais.

3.4.7 Algoritmo de Seguimento do Contorno

O algoritmo de seguimento de um contorno implementado pode decompor-se em três fases temporais:

- Determinação do contorno na primeira imagem da sequência.
- Determinação do contorno na segunda imagem da sequência.
- Determinação dos contornos a partir da terceira imagem.

A determinação do contorno do objecto na primeira imagem da sequência é efectuada exactamente da forma descrita nas secções anteriores. O utilizador define um contorno inicial, sendo em seguida aplicado o algoritmo hierárquico de contornos activos desenvolvido de forma a determinar o contorno desejado.

Na segunda imagem da sequência, a determinação do contorno do objecto pode já, ser efectuada de forma muito mais apoiada. Existe já informação relativa à posição do

contorno na primeira imagem, podendo esta informação ser facilmente utilizada como ponto de partida para a determinação do contorno desejado para a segunda imagem. No entanto, se o objecto que se encontra a ser seguido, se deslocar a uma velocidade elevada, este método pode conduzir a um seguimento incorrecto.

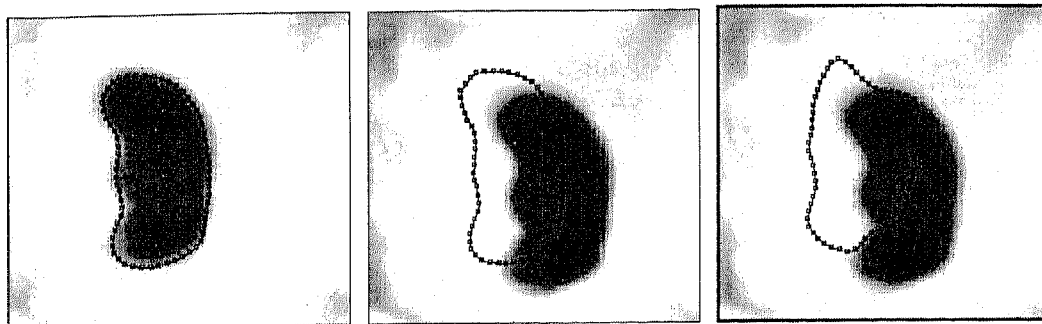


Figura 58: Seguimento incorrecto do objecto devido á utilização do contorno obtido na primeira imagem como contorno inicial da segunda imagem.

Na Figura 58 é visível, que utilizando o contorno obtido para a primeira imagem da sequência (esquerda), como contorno inicial da segunda imagem (centro), o contorno não se consegue adaptar rapidamente aos contornos do objecto (direita). Uma situação ainda mais nefasta surge se for utilizado um valor considerável de força de enchimento no algoritmo de detecção. Neste caso, o contorno tenderá a insuflar-se, não existindo, do seu lado esquerdo, qualquer aresta que o trave. Desta forma, não será possível a detecção correcta da mancha na segunda imagem da sequência, nem em nenhuma imagem a partir desse instante.



Figura 59: Operação de contracção do contorno obtido.

De forma a minorar o problema descrito, a determinação do contorno para a segunda imagem, é efectuada a partir de uma versão contraída do contorno obtido para a primeira imagem (Figura 59). Desta forma, existe uma muito maior probabilidade que o contorno inicial se situe no interior do objecto pretendido, conseguindo-se, desta forma, que o algoritmo de detecção desenvolvido, determine correctamente o contorno desejado na segunda imagem da sequência (Figura 60).

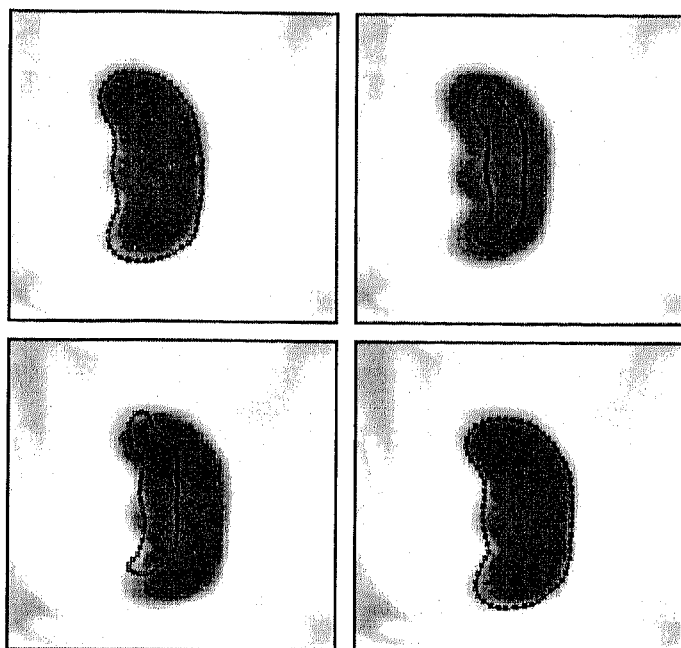


Figura 60: Contracção do contorno obtido na primeira imagem (em cima) e sua utilização como contorno inicial na segunda imagem (em baixo).

A partir da terceira imagem, o contorno inicial utilizado baseia-se na correcção, do contorno obtido para a imagem anterior, utilizando a velocidade instantânea de cada ponto. Esta velocidade é estimada utilizando os deslocamentos sofridos por cada um dos pontos do contorno entre as duas iterações precedentes (Figura 61).

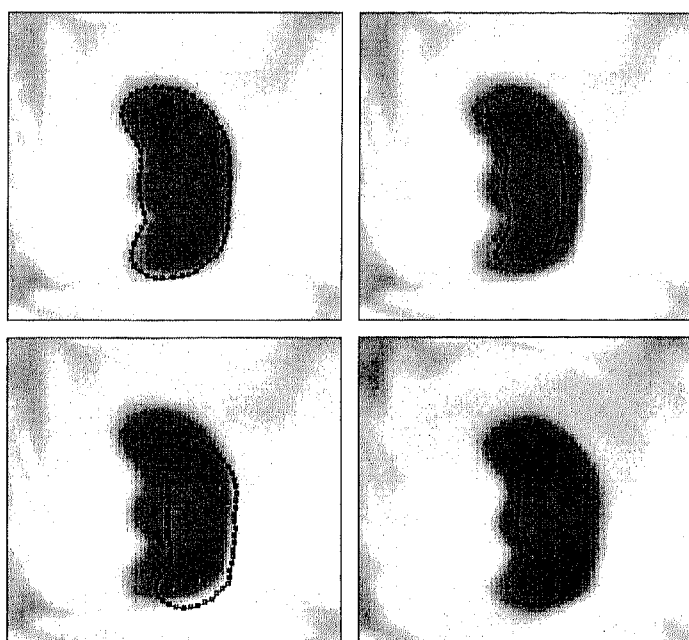


Figura 61: Correcção da posição do contorno utilizando a velocidade instantânea verificada em cada um dos pontos, e sua utilização como contorno inicial na imagem seguinte.

Para além da correcção efectuada utilizando a velocidade instantânea pontual, pode ser efectuada uma correcção global, baseada na velocidade média dos pontos do contorno. Este tipo de correcção consiste em aplicar a todos os pontos do contorno uma operação de translação proporcional ao deslocamento médio sofrido entre as imagens anteriores da sequência, pelos pontos que o constituem. Outra correcção aplicável ao contorno inicial (demonstrada nas figuras) consiste na sua ligeira contracção (tal como a efectuada para o contorno inicial da segunda imagem). Esta operação pode ser de grande utilidade em situações em que o objecto a seguir se desloca a alta velocidade e possui um brilho relativamente constante.

Os parâmetros do algoritmo de seguimento do contorno, são os seguintes:

- NFRAMES Número de imagens da sequência em análise.
- FCONTR2IM Força de contracção do contorno, antes da detecção do contorno na segunda imagem da sequência.
- FCONTR Força de contracção do contorno, antes da detecção do contorno numa imagem genérica da sequência.
- FAJUSTPONT Força de ajuste de cada ponto no cálculo do contorno inicial para a imagem seguinte da sequência.
- FAJUSTGLPO Força de ajuste global dos pontos do contorno relativamente à movimentação média dos pontos do contorno entre as imagens anteriores.

Acrescem a estes parâmetros, todos os parâmetros especificados para o algoritmo de detecção de contornos, que é utilizado na sua globalidade no processo de seguimento.

3.4.8 Conclusões

Ao longo desta secção foram apresentados diversos conceitos relacionados com o projecto e implementação do sistema de detecção e seguimento de contornos. Foram justificadas todas as opções tomadas a nível de sistema operativo, linguagem de programação, análise de requisitos, projecto e implementação do sistema. Foram ainda descritos os métodos de detecção e seguimento de contornos implementados.

O sistema integrado de detecção e seguimento de contornos desenvolvido, permite realizar, de forma extremamente simples e eficaz, análises com vista a detectar, seguir e quantificar contornos em imagens de diversos tipos. As suas potencialidades ao incluírem operações de gravação de imagens, filmes e contornos, permitem armazenar todos os resultados obtidos no processo de análise de forma extremamente simples.

Na secção seguinte, serão utilizadas exactamente as potencialidades específicas de detecção e seguimento de contornos do software de forma a analisar diversas imagens e sequências de imagens.

4. ANÁLISE DE RESULTADOS

4.1 INTRODUÇÃO

Os métodos de detecção de contornos incluídos no sistema desenvolvido, foram validados através da sua aplicação a um vasto conjunto de imagens fictícias contendo manchas com contornos de difícil definição, a imagens reais contendo diversos objectos e a imagens médicas ecocardiográficas.

A validação dos métodos de seguimento de contornos incluídos no sistema, realizou-se tendo como base imagens fictícias contendo o movimento de manchas deformáveis, imagens reais contendo o movimento de diversos objectos e imagens ecocardiográficas contendo o movimento das paredes ventriculares.

Ao longo desta secção será analisado o comportamento dos diversos algoritmos desenvolvidos e os resultados obtidos através da sua aplicação. Para o efectuar serão utilizados alguns dos resultados obtidos durante as fases de teste e validação do sistema e algoritmos desenvolvidos, nomeadamente:

- Detecção de contornos em imagens contendo manchas, geradas artificialmente.
- Detecção de contornos em imagens ecocardiográficas.
- Seguimento de manchas deformáveis em sequências de imagens geradas artificialmente.

De forma a não alongar em demasia este texto e tendo em conta que os três exemplos apresentados são suficientemente esclarecedores, as restantes análises realizadas no teste e validação dos algoritmos desenvolvidos, não são incluídas nesta dissertação.

4.2 RESULTADOS OBTIDOS NA DETECÇÃO DE CONTORNOS EM IMAGENS ARTIFICIAIS

4.2.1 Conjunto de Imagens Utilizado

O método de detecção de contornos desenvolvido foi testado utilizando o conjunto de 20 imagens contendo manchas apresentado na Figura 62.

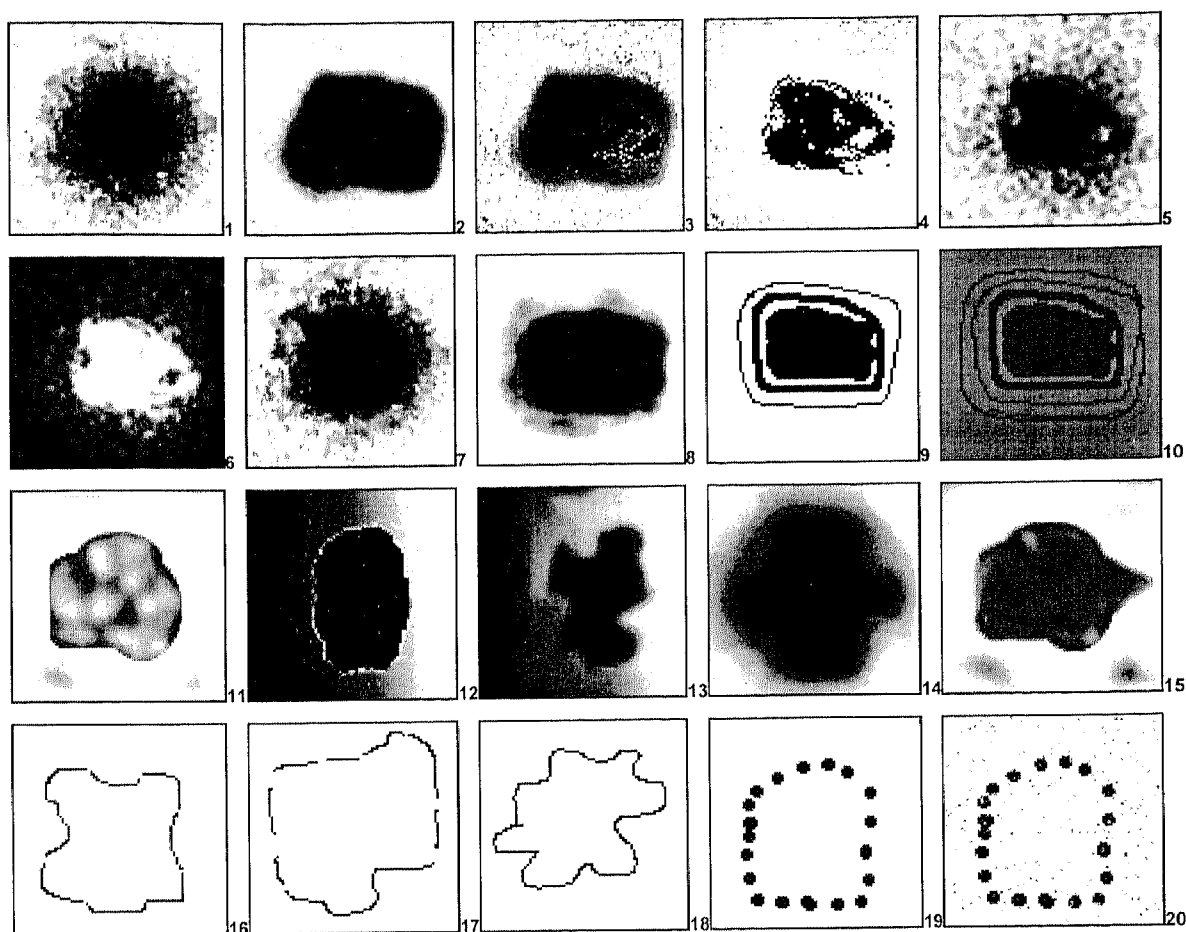


Figura 62: Conjunto de 20 imagens utilizado no teste do algoritmo de detecção de contornos.

Como é visível na figura, foram construídas imagens contendo contornos de diversos tipos e formas:

- Alguns dos contornos a detectar possuem uma forma extremamente simples (por exemplo nas imagens 2, 8 ou 12) enquanto outros possuem formas relativamente complexas (por exemplo os contornos contidas nas imagens 4, 13 e 18).
- Muitos dos contornos a detectar possuem um interior uniforme (2, 8, 12, 13, 14, 16, 17, 18 e 19) enquanto outros possuem um interior muito pouco uniforme (11, 15 e 20).
- A maioria dos contornos a detectar correspondem a arestas do tipo degrau (1 a 15), correspondendo outros a arestas do tipo linha (16, 17 e 18) e dois a contornos ilusórios (19 e 20). Nos contornos com arestas tipo linha, não foram utilizadas forças do detector de arestas (uma vez que os detectores utilizados neste trabalho são adequados a arestas tipo degrau).

- Alguns dos contornos não se encontram muito bem definidos, situando-se nas proximidades de outros contornos menos visíveis. Nestes casos, o contorno que irá ser detectado pelo algoritmo irá depender bastante da inicialização efectuada, podendo ser obtidos diversos resultados finais diferentes.

4.2.2 Evolução Típica de um Contorno ao Longo do Processo Iterativo

De forma a ilustrar a aplicação do algoritmo de detecção desenvolvido, podemos analisar na Figura 63, a evolução de uma “snake” com vista à detecção do contorno da mancha número seis apresentada na Figura 62. Os parâmetros utilizados no algoritmo de detecção foram os seguintes:

- | | |
|---------------------|----------------------|
| • NPONTOS = 64 | • NNIVENCH = 2 |
| • H = 1.56e-02 | • NNIVSUAV = 2 |
| • G = 0.1 | • NNIVAREST = 2 |
| • W1 = 2.43e-04 | • NNIVRIG = 2 |
| • W2 = 5.92e-08 | • KFENCH = 0.0 |
| • ERROMAX = 1.0e-04 | • FACTSUAVMIN = 1 |
| • NITER = 20 | • FACTARESTMIN = 1.5 |
| • KFIMAG = 1.0 | • W1MIN = 2.43e-04 |
| • KFAREST = 1.0 | • W2MIN = 2.96e-04 |
| • KFENCH = 2.0 | • ERROSUAV = 0.4 |
| • FACTSUAV = 3 | • ERRORIG = 0.4 |
| • FACTAREST = 0.5 | • ERROENCH = 0.4 |

Como facilmente pode ser concluído analisando os parâmetros apresentados, foram utilizados unicamente dois níveis hierárquicos para o algoritmo. A transição do primeiro para o segundo nível é efectuada quando o deslocamento médio dos pontos do contorno desce abaixo dos 0.4 pontos por iteração. Esta transição consiste na alteração simultânea da força de enchimento, rigidez do contorno, escala de detecção de arestas e factores de suavização das imagens força.

Na primeira fase do algoritmo, o contorno partindo de uma posição inicial bastante distante do resultado pretendido, é impulsionado pela força de enchimento, aproximando-se das arestas exteriores da mancha.

Algumas das arestas interiores, nomeadamente perto da extremidade direita do contorno, ainda exercem uma razoável força de atracção ao contorno. Esta força é derrotada pela acção conjunta das forças internas do contorno e força de enchimento (imagens 4 a 6 da sequência).

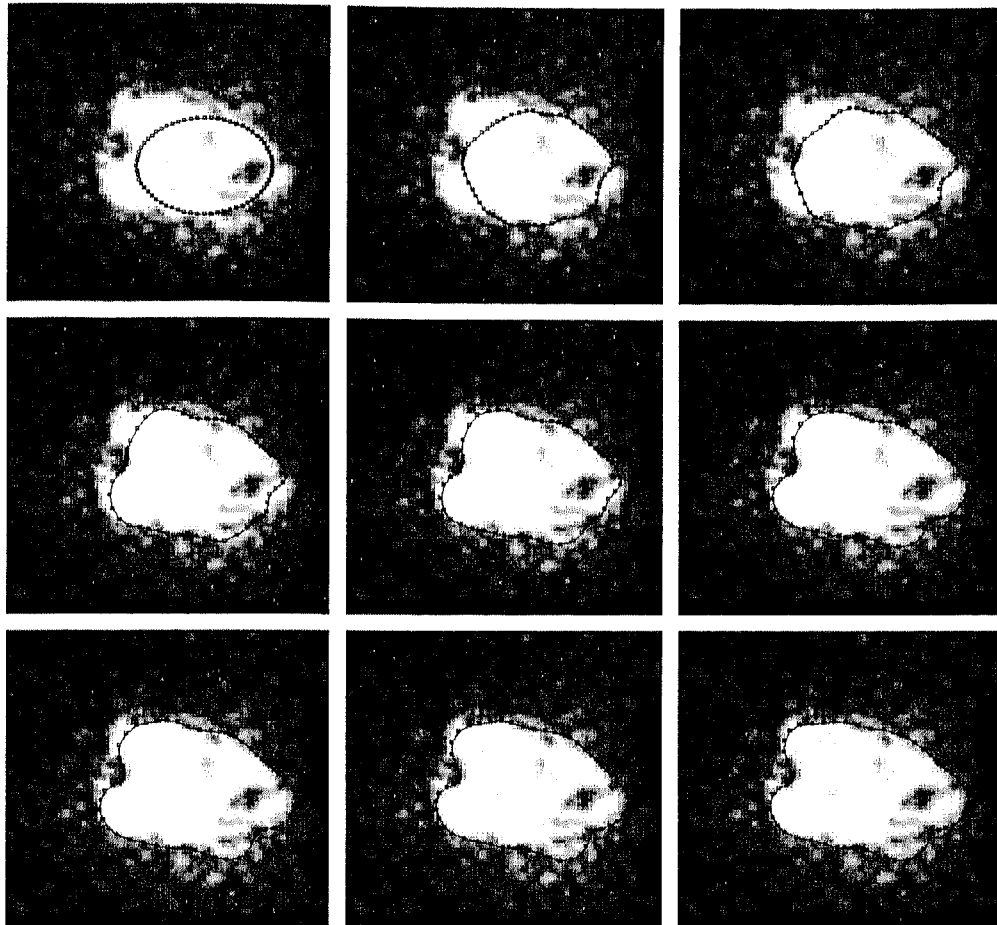


Figura 63: Evolução do contorno na primeira fase do algoritmo de detecção de contornos.

Na fase final do processo, o deslocamento do contorno é já muito baixo, anulando-se o somatório das forças aplicadas a cada ponto, sendo assim atingido o equilíbrio energético. A força de enchimento, sendo perpendicular ao contorno em todos os seus pontos, conduz a que o equilíbrio energético se verifique para um contorno ligeiramente mais aberto do que o desejado. Por sua vez, a alta rigidez do contorno e as imagens força muito suavizadas utilizadas no início do processo, não permitem que o contorno se adapte com precisão ao formato exterior sinuoso da mancha.

Surgem aqui as vantagens do algoritmo hierárquico! O baixo deslocamento provoca a alteração dos parâmetros do algoritmo, sendo alteradas as propriedades físicas do contorno, a força de enchimento e as condições de suavização e de escala do detector de arestas.

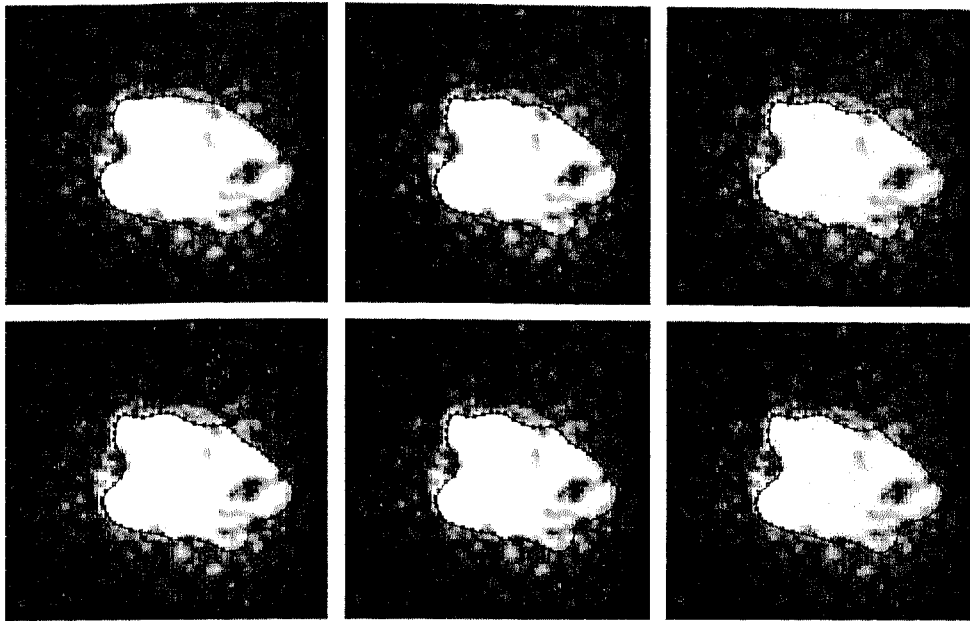


Figura 64: Evolução do contorno após diminuição simultânea rigidez do contorno, força de enchimento, suavização das imagens força e escala de detector de arestas.

A diminuição da rigidez da "snake" torna-a mais flexível permitindo-lhe ajustar-se melhor à forma da mancha contida na imagem. Acrescido a este facto, a diminuição da suavização das forças do detector de arestas e gradiente e da escala do detector de arestas, tornam as forças da imagem mais fortes e localizadas unicamente na região correspondente ao contorno. Como a "snake" já se encontra próxima desta região, este facto, não conduz a nenhum tipo de inconveniente, tendo como consequência um ajuste progressivo da "snake" à verdadeira forma da mancha. A anulação da força de enchimento verificada, provoca uma ligeira contracção no contorno, contribuindo para que o equilíbrio energético se verifique na região correcta. O contorno final obtido, corresponde com grande precisão e exactidão ao formato da mancha contida na imagem.

4.2.3 Detecção de Contornos no Conjunto de Imagens Artificiais

Foram realizadas análises com vista a testar o comportamento do algoritmo, na análise de todas as imagens teste apresentadas. Nas imagens teste ilustradas na figura 62, foram realizadas as inicializações apresentadas na figura 65.

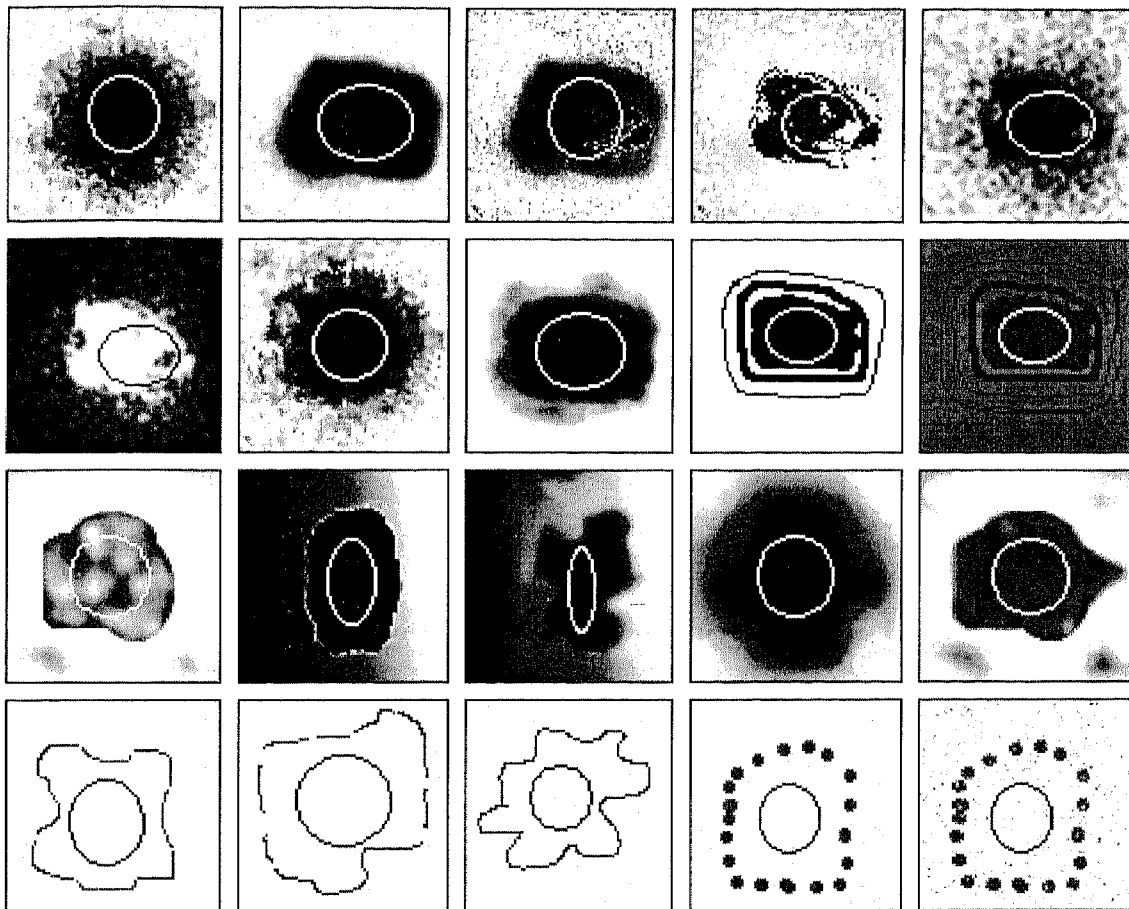


Figura 65: Inicializações dos contornos activos com vista a efectuar a detecção das manchas contidas nas imagens (demonstrados em alguns casos sobre imagens menos contrastadas).

Como é visível na figura, as inicializações foram na sua totalidade efectuadas utilizando o método mais simples disponibilizado pelo programa, ou seja, o traçado de um contorno elíptico. Esse contorno, foi desenhado perto da região central das manchas a detectar, incluído na totalidade no interior destas.

Na figura 66, podemos verificar as imagens resultantes da aplicação do detector de arestas de Shen e Castan (com $\alpha=0.5$, factor de escala utilizado no início do algoritmo para a maioria das imagens).

O detector de arestas de Shen e Castan consegue identificar na maioria das imagens, com razoável precisão, as arestas correspondentes às manchas a detectar. No entanto, em grande parte das imagens apresentadas, verifica-se a detecção de muitas outras arestas que não fazem parte dos contornos desejados.

Nas imagens da linha inferior, os resultados do detector de arestas não correspondem aos contornos desejados. Este facto deve-se a que nas imagens 16, 17 e

18, as arestas detectadas são do tipo linha. Sendo o detector de Shen e Castan formulado de forma a apresentar uma resposta óptima a arestas do tipo degrau, apresenta respostas múltiplas às linhas contidas nas imagens. Nas imagens 19 e 20, o detector responde às arestas correspondentes aos pequenos pontos que constituem o contorno ilusório.

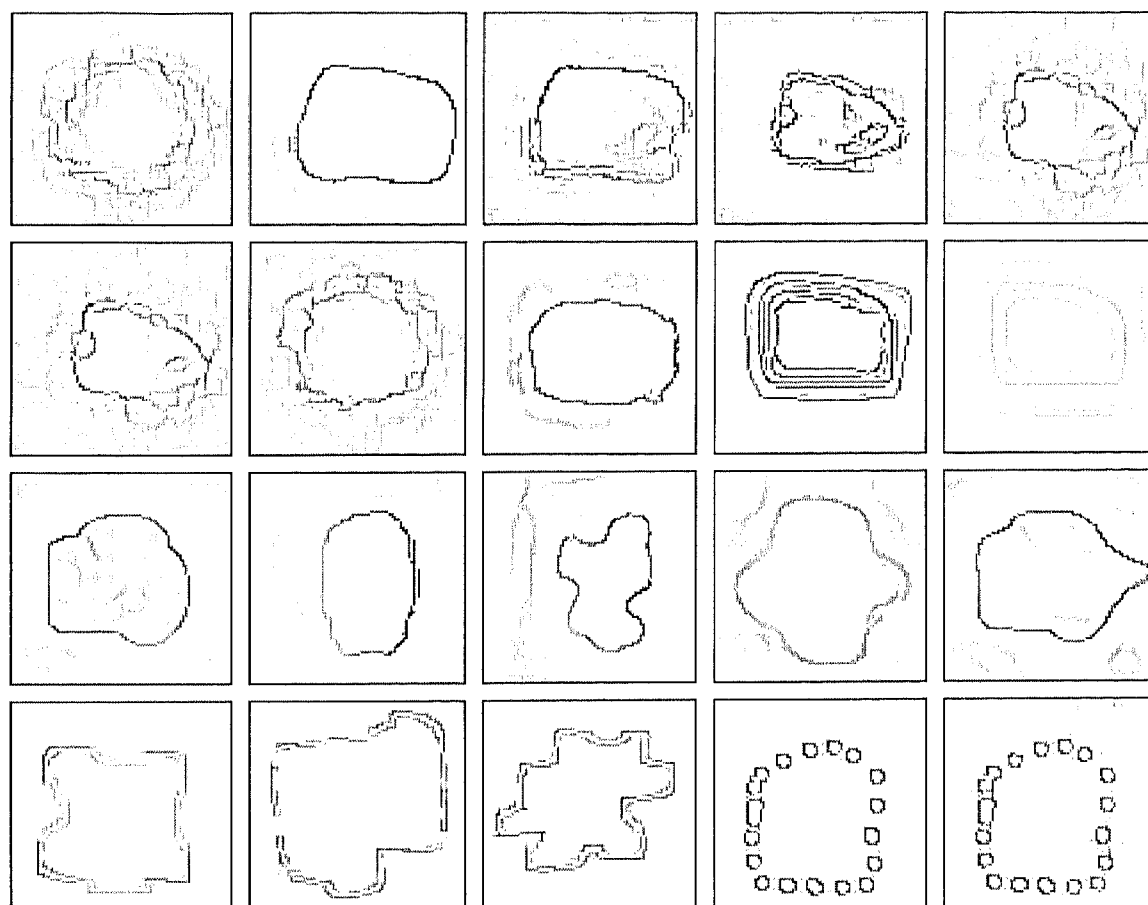


Figura 66: Resultados da aplicação do detector de arestas de Shen e Castan às 20 imagens teste.

Os resultados obtidos, utilizando os contornos iniciais apresentados e parâmetros adequados para o algoritmo hierárquico de detecção, encontram-se representados na Figura 67. Para todas as manchas, foi utilizado um algoritmo com três níveis, um contorno com 64 pontos e parâmetros de rigidez e elasticidade adequados à sua forma.

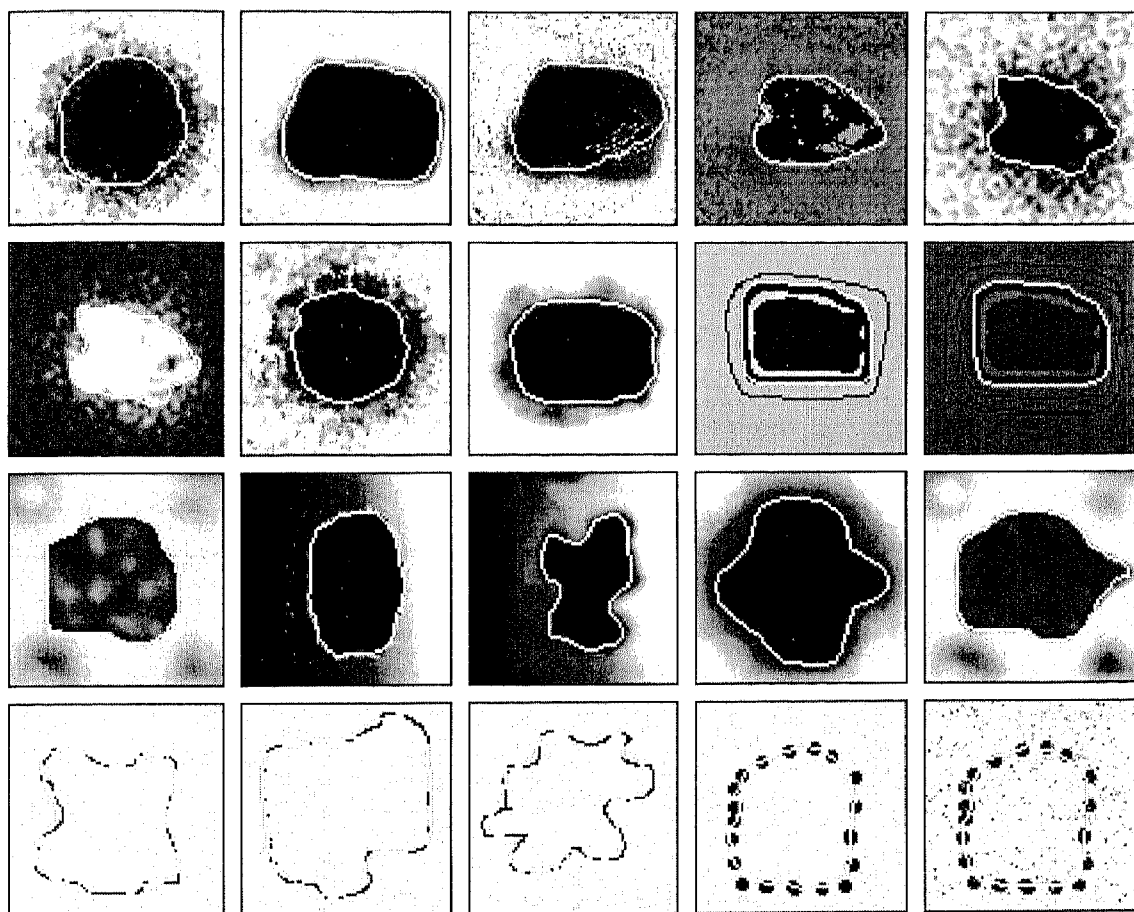


Figura 67: Conjunto de 20 manchas de teste e respectivos resultados obtidos utilizando o algoritmo hierárquico desenvolvido (demonstrados para uma melhor visualização, em alguns casos, sobre imagens com menor contraste).

Os resultados finais obtidos, como é visível na figura, correspondem a uma detecção exacta e precisa dos contornos das manchas reais e ilusórias, contidas nas diversas imagens.

4.3 RESULTADOS OBTIDOS NA ANÁLISE DE IMAGENS ECOCARDIOGRÁFICAS

4.3.1 Introdução

Nas últimas décadas, a aplicação da tecnologia dos ultra-sons na produção de imagens do coração, tem vindo a ser cada vez mais utilizada. Comparado com outras técnicas de visualização do coração, o exame ecocardiográfico é muito simples, seguro (pois trata-se de uma técnica não invasiva) e rápido (podendo ser efectuado em tempo

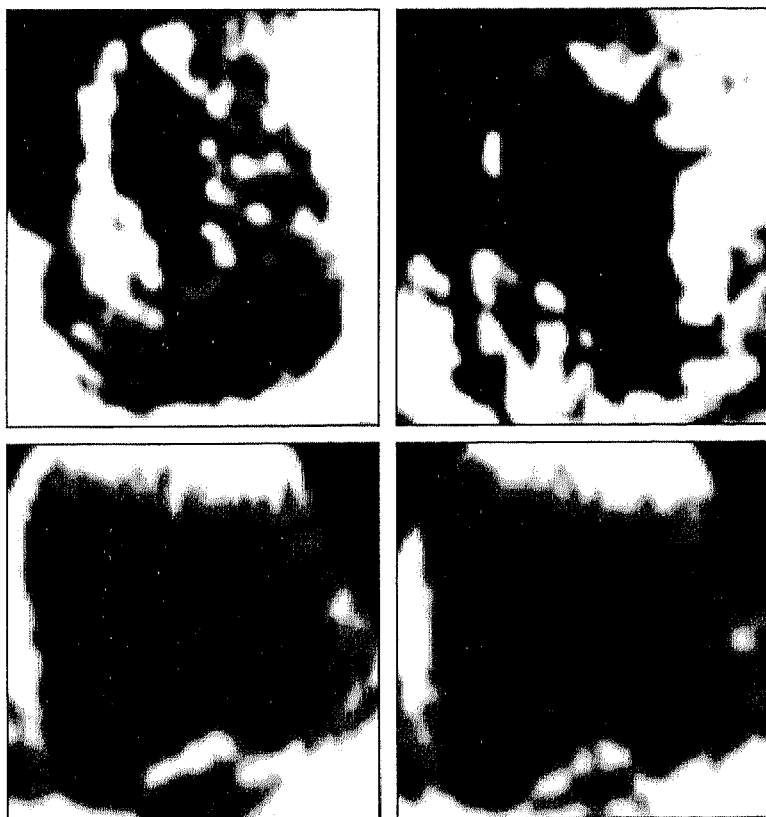
real). Acrescido a estes factores, a ecocardiografia é uma técnica portátil e cujos equipamentos são relativamente baratos.

Todas estas características, tornaram a ecocardiografia numa das ferramentas mais utilizadas no diagnóstico cardíaco dos nossos dias. Ao longo dos anos, tem existido um crescente esforço para efectuar a extracção de informação quantitativa e qualitativa, a partir dos exames ecocardiográficos. Contudo, a ecocardiografia não é uma técnica precisa, sendo as imagens obtidas com o seu auxílio, muito difíceis de analisar com as ferramentas de análise disponíveis no mercado.

Nesta secção, será abordada a forma de extrair informação quantitativa de exames ecocardiográficos utilizando os métodos de detecção e seguimento de contornos implementados no sistema desenvolvido.

4.3.2 Imagens Ecocardiográficas Utilizadas

Na realização deste estudo, foram utilizadas as imagens ecocardiográficas utilizadas por Monteiro no seu estudo [Monteiro94] (Figura 68).



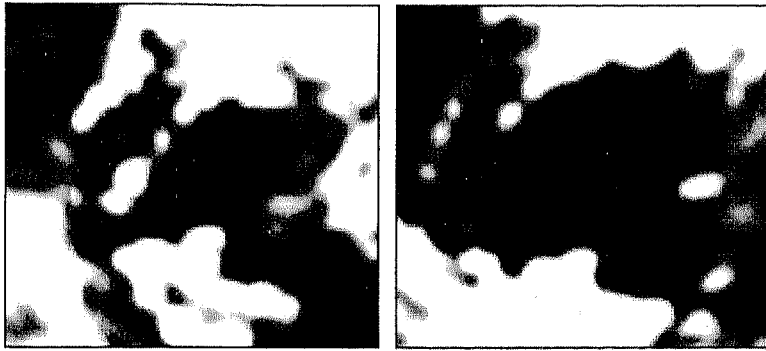


Figura 68: Conjunto de imagens ecocardiográficas utilizadas na análise. À esquerda encontram-se as imagens correspondentes à sístole, enquanto à direita encontram-se as imagens correspondentes à diástole.

Como é visível na figura, devido à indisponibilidade de imagens em estado bruto, foram utilizadas imagens pré-processadas utilizando o método desenvolvido por Monteiro [Monteiro94]. Este método baseia-se na utilização consecutiva de operações de “mascaramento” (para determinação da região de interesse), filtragem de média (para suavização), normalização dos níveis de cinzento (para aumento de contraste) e filtragem vertical Gaussiana (para eliminação de ruído tipo estrias presente nas imagens). As imagens disponíveis encontram-se desta forma já bastante suavizadas, o que embora não seja crítico, prejudica substancialmente os resultados obteníveis com os algoritmos desenvolvidos neste trabalho.

4.3.3 Detecção de Contornos em Imagens Ecocardiográficas

A análise das imagens ecocardiográficas foi efectuada traçando contornos iniciais muito simples nas imagens disponíveis (Figura 69).



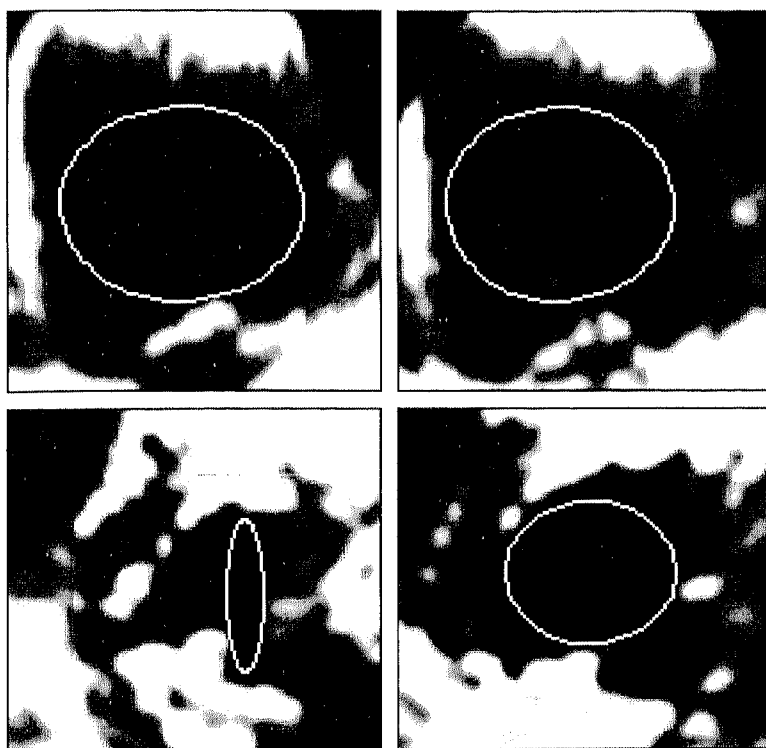


Figura 69: Contornos iniciais traçados sobre as imagens ecocardiográficas.

Os resultados obtidos na análise das imagens apresentadas, encontram-se ilustrados na Figura 70.

É visível a detecção com razoável precisão, tanto na sístole como na diástole, dos contornos da cavidade endocárdica das diversas imagens. No entanto, os contornos determinados, podem não corresponder totalmente aos contornos reais. Este facto deve-se a que os resultados obtidos encontram-se ilustrados sobre imagens pré-processadas utilizando o método descrito anteriormente e consequentemente pouco precisas.



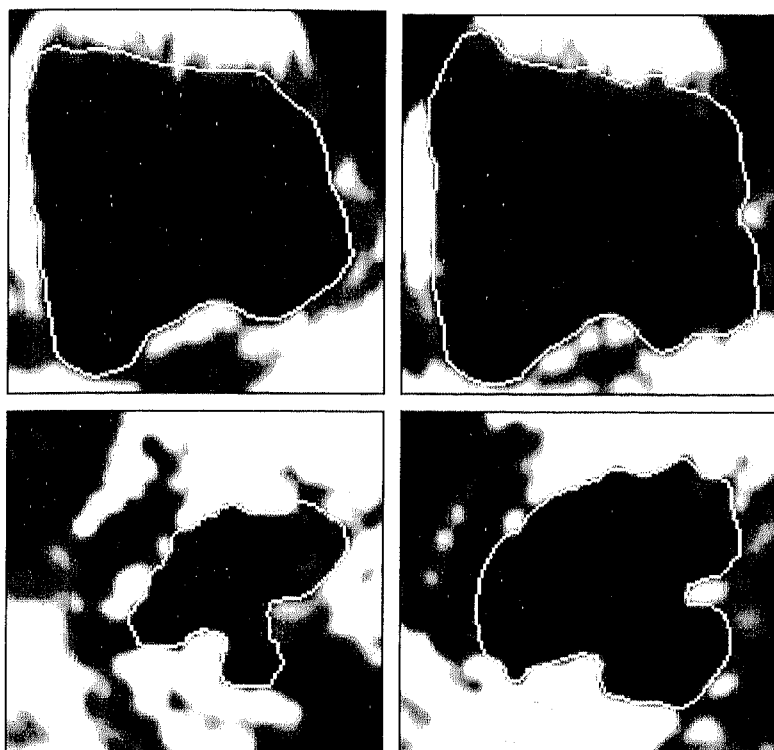


Figura 70: Resultados finais obtidos na detecção dos contornos endocárdicos.

Na análise de outras imagens ecocardiográficas disponíveis, os resultados obtidos foram também de bastante boa qualidade. No entanto, de forma a obter resultados correctos em parte das imagens, sobretudo naquelas com contornos sinuosos e pouco acentuados, é necessária uma inicialização um pouco mais cuidada do contorno inicial e uma correcta afinação dos parâmetros do algoritmo.

4.4 RESULTADOS OBTIDOS NO SEGUIMENTO DE CONTORNOS EM IMAGENS ARTIFICIAIS

4.4.1 Conjunto de Imagens Utilizado

A análise de movimento utilizando contornos activos foi efectuada utilizando diversas sequências de imagens geradas artificialmente. Na Figura 71 é apresentada uma das sequências teste utilizada.

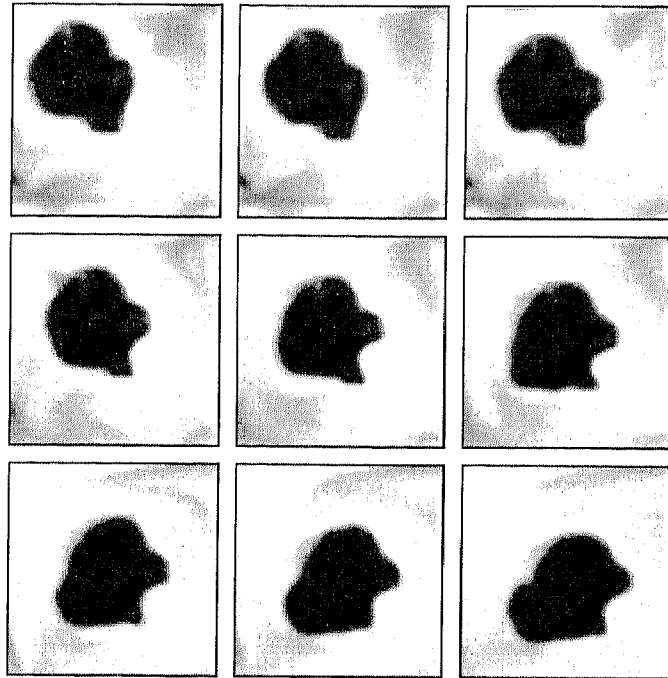


Figura 71: Sequência de 9 imagens contendo o movimento de uma mancha deformável.

A sequência de teste apresentada na Figura 71, contém uma mancha que se desloca do canto superior esquerdo para o canto inferior direito da imagem. No decurso do seu deslocamento, a mancha sofre uma deformação, apresentando na imagem final uma forma totalmente diferente da apresentada na imagem inicial da sequência.

4.4.2 Seguimento de Contornos em Sequências de Imagens Artificiais

Os principais problemas do método de seguimento de contornos implementado, situam-se no início do processo de seguimento, altura em que ainda não se encontra disponível informação respeitante ao deslocamento dos pontos.

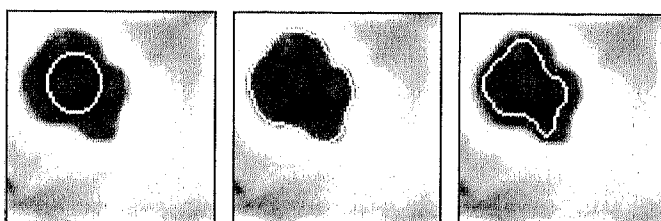


Figura 72: Definição do contorno inicial (esquerda), resultado obtido para a imagem inicial da sequência (centro) e contracção do contorno de forma a ser utilizado como contorno inicial da imagem seguinte da sequência (direita).

Na Figura 72 são visíveis os resultados obtidos pelo método de detecção implementado, utilizando um contorno inicial circular muito simples. É ainda visível o cálculo do contorno inicial para a segunda imagem da sequência, através da contracção do contorno final obtido para a primeira imagem.



Figura 73: Utilização da contracção do contorno obtido na imagem inicial (esquerda) como contorno inicial da segunda imagem(centro) e resultados obtidos (direita).

A utilização de uma contracção do contorno solução inicial, como posição inicial na segunda imagem, encontra-se ilustrada na Figura 73. O contorno contraído encontra-se no interior da mancha contida na imagem, resultando do processo de detecção um contorno final correcto.

A partir da segunda imagem da sequência, o cálculo da posição inicial do contorno é efectuado tendo em conta o seu deslocamento na iteração anterior. Os resultados apresentados na Figura 74, foram obtidos utilizando para o deslocamento dos pontos, correcções simultâneas de velocidade pontual e velocidade global do contorno, e utilizando uma ligeira contracção do contorno inicial. Os parâmetros utilizados para o algoritmo foram os seguintes:

- NFRAMES = 9
- FCONTR2IM = 1.0
- FCONTR = 0.2
- FAJUSTPONT = 1.0
- FAJUSTGLPO = 0.5

Devido à mancha contida na sequência de imagens, sofrer uma deformação ao longo do tempo, foi utilizada no algoritmo, uma força de ajuste pontual superior à força de ajuste global. Para efectuar o seguimento de objectos não deformáveis, não é necessária a utilização da força de ajuste pontual, bastando a utilização da força de ajuste proporcional ao deslocamento médio dos pontos do objecto.

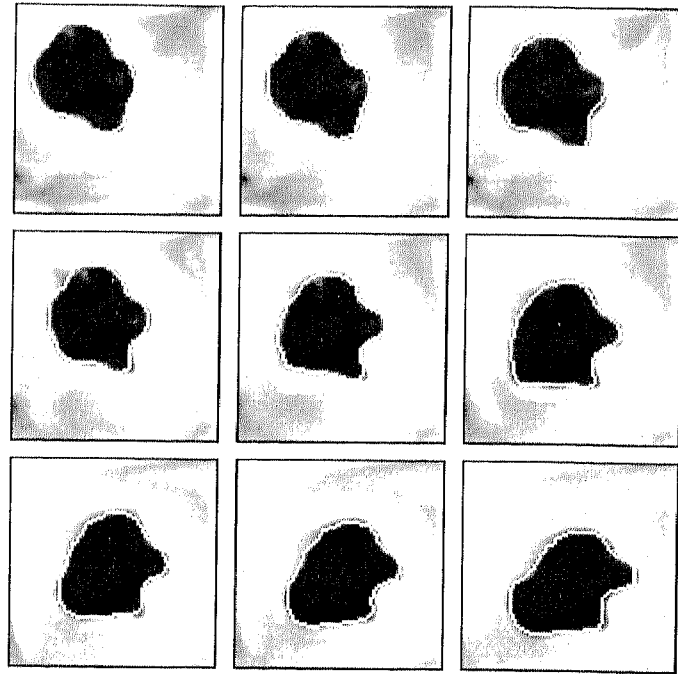


Figura 74: Resultados obtido no processo de seguimento de contornos para a sequência de imagens teste utilizada.

No algoritmo foi ainda utilizada uma ligeira força de contracção do contorno, no cálculo dos contornos iniciais. Foi comprovado em várias sequências artificiais que a utilização desta força de contracção, associada à utilização no início do algoritmo hierárquico de detecção, de uma ligeira força de enchimento, conduz à obtenção de processos iterativos mais simples e com maiores probabilidades de conduzirem a resultados correctos.

Analisando os resultados obtidos (Figura 74), é visível que o algoritmo de seguimento desenvolvido, através da utilização dos parâmetros referidos, consegue facilmente efectuar o seguimento do movimento e deformação da mancha presente na imagem, fornecendo resultados correctos para todas as imagens da sequência.

5. CONCLUSÕES E CONSIDERAÇÕES FINAIS

Uma primeira conclusão a retirar, consiste no facto dos objectivos propostos no início do trabalho que conduziu a esta dissertação, terem sido muito satisfatoriamente alcançados. Para os objectivos serem plenamente atingidos faltou a inclusão no sistema desenvolvido, do módulo de aquisição de imagens inicialmente implementado. Tal não foi possível devido à avaria irreparável da placa de aquisição de imagem que se encontrava adstrita a este trabalho.

Neste trabalho foi apresentada uma descrição das técnicas de detecção e seguimento de contornos utilizadas actualmente. Foram analisadas em maior detalhe, as técnicas baseadas em contornos activos, sendo referidos os principais problemas da sua aplicação prática em sistemas reais. Foi concluída a aplicabilidade prática destas técnicas na detecção e seguimento de contornos e foram referidas as principais aplicações actuais dos contornos activos.

Foi desenvolvido um sistema cujo objectivo consistiu em permitir a detecção e seguimento de contornos em imagens e sequências de imagens. O sistema projectado e implementado revelou-se, na prática, muito simples de utilizar, possuindo uma interface intuitiva e extremamente agradável. As potencialidades genéricas de processamento de imagem e específicas de detecção e seguimento de contornos tornam o sistema numa ferramenta bastante poderosa, nomeadamente na área da extracção de características em imagens médicas.

A plataforma de desenvolvimento seleccionada revelou-se indicada ao desenvolvimento de raiz de um sistema de processamento de imagem com as características desejadas. Os formatos de imagem utilizados (BMP e BIFF), revelaram-se adequados e de utilização bastante simples. No entanto, ao longo da execução deste projecto, foi por vezes necessária a utilização de sistemas comerciais de processamento de imagem, para realizar operações de conversão de formato em imagens provenientes de outros sistemas, de forma a que pudessem ser tratadas pelo sistema desenvolvido. A inclusão de filtros de conversão de formato revelou-se necessária em futuras versões do sistema.

Os métodos de definição inicial e ajuste manual do contorno demonstraram ser bastante adequados à realização dessas operações. Confrontados com estes métodos, os potenciais utilizadores do sistema revelaram total satisfação. A grande maioria dos utilizadores do sistema, revelaram apreciar particularmente o método de ajuste do contorno através de forças de atracção e repulsão.

O algoritmo hierárquico de detecção de contornos desenvolvido, demonstrou na prática conduzir a melhores resultados do que os algoritmos de contornos activos tradicionais. As inovações propostas na estrutura de dados utilizada, demonstraram-se adequadas ao algoritmo utilizado, embora impliquem um maior esforço computacional para o algoritmo. A versão final do algoritmo torna-se um pouco lenta, se forem utilizados diversos níveis de forças do gradiente e detector de arestas, ou diversas propriedades de rigidez e elasticidade para o contorno ao longo do processo iterativo. No entanto, na prática, a utilização de três ou quatro níveis revela-se suficiente pelo que o acréscimo no peso computacional torna-se pouco significativo.

O método de seguimento de contornos proposto demonstrou ser adequado ao seguimento de objectos com um interior constante que se deslocam a média ou reduzida velocidade. Se o interior dos objectos não for constante ou for confundido facilmente com o fundo da imagem, a qualidade dos resultados é bastante inferior. Se a velocidade dos objectos a seguir for muito elevada, o contorno não consegue “arrancar” e seguir o objecto nas primeiras imagens da sequência. Este facto, deve-se a que o método proposto não utiliza qualquer informação relativa ao campo de velocidades da imagem. Desta forma é incapaz de detectar o contorno na segunda imagem da sequência pois utiliza como ponto de partida uma versão contraída do contorno obtido para a primeira imagem. Conceder ao utilizador a permissão de definir manualmente o contorno na segunda imagem da sequência, pode ser uma boa forma de evitar este problema. Outra forma consiste na introdução de informação obtida por um detector de movimento genérico.

Como possíveis desenvolvimentos futuros deste trabalho, nomeadamente do sistema integrado de detecção de contornos implementado, sugerem-se os seguintes:

- Inclusão de um módulo de aquisição de imagem de forma a que o sistema fique funcionalmente completo de acordo com a especificação fornecida.
- Implementação de conversores entre formatos de imagens de forma a que o sistema possa tratar imagens provenientes de outros ambientes e sistemas.
- Melhoria das potencialidades de impressão do sistema com inclusão da possibilidade de colocação de títulos, cabeçalhos e rodapés nas páginas e impressão de várias imagens por página.
- Aumento das potencialidades ao nível do conjunto de operações genéricas de processamento de imagem disponibilizado, nomeadamente através da inclusão de transformações geométricas (escala, perspectiva, rotação, etc), operações de segmentação, realce e reconhecimento de padrões.

- Permissão ao utilizador de alteração na representação das imagens, através da alteração do tipo (escala de cinzentos, indexada, RGB, CMYK, preto e branco).
- Inclusão de potencialidades de edição de imagem pixel a pixel e da sobreposição de elementos gráficos tais como texto, linhas ou setas.
- Inclusão no sistema de novos métodos de análise de imagem baseados em contornos activos.
- Teste dos algoritmos de detecção e seguimento de contornos num conjunto mais alargado de imagens ecocardiográficas.
- Teste do algoritmo hierárquico de detecção de contornos desenvolvido, num maior número de imagens, derivadas de outras aplicações de análise de imagem médica.
- Introdução de métodos de predição baseados em filtros de Kalman do algoritmo de seguimento de contornos desenvolvido.
- Utilização de métodos de estimação do campo de velocidades em conjunto com o algoritmo de seguimento desenvolvido.
- Construção de sistemas específicos, baseados no sistema de detecção e seguimento de contornos implementado, adaptados a problemas médicos reais.

Muitos outros desenvolvimentos futuros deste trabalho são possíveis, tendo sido referidos nesta análise unicamente aqueles considerados principais. O sistema protótipo desenvolvido, constitui uma base sólida para a integração dos desenvolvimentos referidos, permitindo no futuro, com reduzido esforço, a evolução para um sistema mais completo e potente.

REFERÊNCIAS BIBLIOGRÁFICAS

[Adler90a] Adler, Marc

"Learning Windows Part I: The Message Based Paradigm"
Microsoft Systems Journal, pp. 85-91, Julho, 1990

[Adler90b] Adler, Marc

"Learning Windows Part II: Resources and Menuing System"
Microsoft Systems Journal, pp. 75-90, Setembro, 1990

[Adler90c] Adler, Marc

"Learning Windows Part III: Control Windows and MDI Support"
Microsoft Systems Journal, pp. 67-85, Novembro, 1990

[Adler91a] Adler, Marc

"Learning Windows Part IV: Integrating Controls and Dialog Boxes"
Microsoft Systems Journal, pp. 97-118, Janeiro, 1991

[Adler91b] Adler, Marc

"Learning Windows Part V: Exploring the Graphics Device Interface"
Microsoft Systems Journal, pp. 75-88, Março, 1991

[Adler91c] Adler, Marc

"Learning Windows Part VI: Fonts, Bitmaps and Printing"
Microsoft Systems Journal, pp. 115-131, Maio, 1991

[Bajcsy89] Bajcsy, R., Kovacic, S.

"Multi-Resolution Elastic Matching"
Computer Vision, Graphics and Image Processing, Vol. 46, pp. 1-21, 1989

[Bascle94] Bascle, Benedicte ; Bouthemy, Patrick ; Deriche, Rachid ; Meyer, Francois

"Suivi de Primitives Complexes sur une Déquences d'Images"
Institut National de Recherche en Informatique et en Automatique - N° 2468, Dezembro, 1994

[Baumberg93] Baumberg, A. M. ; Hogg, D. C.

"Learning Flexible Models from Image Sequences"
University of Leeds, Schools of Computer Studies - N° 93.36, Outubro, 1993

[Baumberg94] Baumberg, A. M. ; Hogg, D. C.

"An Efficient Method for Contour Tracking Using Active Shape Models"

University of Leeds, Schools of Computer Studies - Nº 94,11, Abril, 1994

[Berr81] Beer, F. P. ; Johnston Jr., E. R.

"Mecânica Vectorial para Engenheiros - Vol. I e II"

McGraw-Hill, 1981

[BlackWell95] BlackWell, Ken

"How to Write Fast Win32 Portable Applications"

Software Development 95 Proceedings, E.U.A., 1995

[Blake93] Blake Andrew; Curwen, Rupert ; Zisserman, Andrew

"A Framework for Spatiotemporal Control in the Tracking of Visual Contours"

International Journal of Computer Vision, Vol. 11, Nº2, pp. 127-145, 1993

[Branco85] Branco, Carlos Moura

"Mecânica dos Materiais"

Fundação Calouste Gulbenkian, 1985

[Brenda90] Brenda, Laurel

"The Art of Human-Computer Interface Design"

Addison Wesley, 1990

[Burt88a] Burt, P.

"Smart Sensing in Machine Vision"

Machine Vision Algorithms, Architectures and Systems, pp. 1-30, Academic Press, 1988

[Burt88b] Burt, P.

"Smart Sensing Within a Pyramid Vision Machine"

Proceedings IEEE, Vol. 76, Nº 8, pp. 1006-1015, 1988

[Campilho93] Campilho, Aurélio C.

"Métodos Hierarquicos na Representação e Análise de Imagem"

Publicação Interna Faculdade de Engenharia da Universidade do Porto, 1993

[Canny86] Canny, J.F.

"A Computational Approach to Edge Detection"

IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8(6), pp. 679-698, Novembro, 1986

[Chapra85] Chapra, Steven C. ; Canale, Raymond P.

"Numerical Methods for Engineers - With Personal Computer Applications"

McGraw-Hill, 1985

[Charlap95] Charlap, David

"The BMP File Format, Part 1"

Dr Dobb's Journal, Março, 1995

[Clark89] Clark, James J.

"Authenticating Edges Produced by Zero-Crossing Algorithms"

IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-11(1), pp. 43-57, Janeiro, 1989

[Cohen91a] Cohen, Laurent D.

"NOTE On Active Contour Models and Balloons"

CVGIP: Image Understanding, Vol. 53, Nº 2, pp. 211-218, Março, 1991

[Cohen91b] Cohen, Isaac ; Cohen, Laurent D. ; Ayache, Nicholas

"Introducing Deformable Surfaces to Segment 3D Images and Infer Differential Structures"

Institut National de Recherche en Informatique et en Automatique - Nº 1403, Março, 1991

[Cohen92] Cohen, Isaac ; Cohen, Laurent; Ayache, Nicholas

"Using Deformable Surface to Segment 3d Images and Infer Differential Structures"

Proceedings of the Second European Conf. on Computer Vision, pp. 648-652, Santa Margherita Ligure, Itália, Maio, 1992

[Cohen93] Cohen, Laurent; Ayache, Nicholas; Bardinet, Eric

"Surface Reconstruction Using Active Contour Models"

Rapports de Recherche, Nº 1824, Institut National de Recherche en Informatique et en Automatique, Fevereiro, 1993

[Conte81] Conte, Samuel D. ; Boor, Carl

"Elementary Numerical Analysis - An Algorithmic Approach"

McGraw-Hill, 1981

[Correia95] Correia, Miguel V.

"Análise de Movimento em Sequências de Imagens"

Tese de Mestrado, Faculdade de Engenharia da Universidade do Porto, Fevereiro, 1995

[Cox93] Cox, Kevin ; Walker, David

"User Interface Design"

Prentice Hall, Second Edition, 1993

[Delagnes95] Delagnes, Philippe; Benois, Jenny; Barba, Dominique

"Active Contours Approach to Object Tracking in Image Sequences with Complex Background"

Pattern Recognition Letters Vol. 16, pp. 171-178, 1995

[Deriche87] Deriche, Rachid

"Using Canny's Criteria to Derive a Recursively Implemented Optimal Edge Detector"

International Journal of Computer Vision, Vol. 1, Nº2, pp. 167-187, 1987

[Essa95] Essa, Irfan Aziz

"Analysis, Interpretation and Synthesis of Facial Expressions"

Dissertação de doutoramento, Massachusetts Institute of Technology - Fevereiro, 1995

[Fujimura93] Fujimura, K. ; Yokoya, N. ; Yamamoto, K.

"Motion Tracking of Deformable Objects by Active Contour Models Using Multiscale Dynamic Programming"

Journal of Visual Communication and Image Representation, Vol. 4, Nº4, pp. 382-391, 1993

[Gladwell79] Gladwell, I.

"A Survey of Numerical Methods for Partial Differential Equations"

Claredon, Oxford, 1979

[Goshtasby93] Goshtasby, Ardeshir

"Design and Recovery of 2-D and 3-D Shapes Using Rational Gaussian Curves and Surfaces"

International Journal of Computer Vision, Vol. 10, Nº3, pp. 233-256, 1993

[Haralick84] Haralick, R. M.

"Digital Step Edges From Zero-Crossings of the Second Directional Derivatives"

IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-6(1), pp. 58-68 Janeiro, 1984

[Henricson94] Henricson, Olof ; Neuenschwander, W.

"Controlling Growing Snakes by Using Key-Points"

IEEE 1051-4651/94, 1994

[Holzner94] Holzner, Steven

"Visual C++ Programming"

Brady Publishing, 1994

[Huang88] Huang, J. S. ; Tseng, D. H.

"Statistical Theory of Edge Detection"

Computer Vision, Graphics and Image Processing, Vol. 43, pp. 337-346, 1988

[Kass87] Kass, Michael, Witkin, Andrew and Terzopolus, Demetri

"Snakes: Active Contour Models"

Proceedings of International Conf. on Computer Vision, pp. 259-268, Londres, 1987

[Kass88] Kass, Michael, Witkin, Andrew and Terzopolus, Demetri

"Snakes: Active Contour Models"

International Journal of Computer Vision, Vol. 1, N°4, pp. 321-333, 1988

[Kervrann93] Kervrann, Charles ; Heitz, Fabrice

"A Hierarchical Statistical Framework for the Segmentation of Deformable Objects in Image Sequences"

Institut National de Recherche en Informatique et en Automatique - N° 2133, Dezembro, 1993

[Kruglinsky] Kruglinsky, David J.

"Inside Visual C++"

Microsoft Press, 1993

[Lai94a] Lai, Kok F. ; Chin, Roland T.

"On Classifying Deformable Contours Using the Generalised Active Contour Model"

M.I.T. Media Laboratory - 1994

[Lai94b] Lai, Kok F.

"Deformable Contours: Modelling, Extraction, Detection and Classification"

Dissertação de Doutorado, University of Wisconsin-Madison, 1994

[Leymarie93] Leymarie, F. ; Levine, M. D.

"Tracking Deformable Objects in the Plane Using an Active Contour Model"

IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-15(6), pp. 617-633, 1993

[Lonnestad92] Lonnestad, Tor ; Milvang, Otto

"XITE: X-based Image Processing Tools and Environment User's Manual"

University of Oslo, Outubro, 1992

[Lowe89] Lowe, David G.

"Organization of Smooth Image Curves at Multiple Scales"

International Journal of Computer Vision, Vol. 3, N°2, pp. 119-130, 1989

[Marr80] Marr, David; Hildreth, E.

"A Theory of Edge Detection"

Proceedings of the Royal Society of London Vol. B-107, pp. 187-217, 1980

[Marr82] Marr, David

"Vision: A Computational Investigation in the Human Representation and Processing of Visual Information"

W. H. Freeman and Company, 1982

[Maurincomme93] Maurincomme Eric; Friboulet, D.; Finet, G.; Magnin, I. e Reiber, J.

"ADDER: A Snake-Based Segmentation Approach for Intravascular Ultrasound Images"

DICTA-93 Conference Proceedings - Digital Image Computing: Techniques and Applications, edited by Fung, Ka Kit and Ginige, Athula, v.1 pp. 422-429, Sydney, 1993

[Maurincomme94] Maurincomme, Eric.

"Analysis and Segmentation of Intravascular Ultrasound Images"

PhD Dissertation, Institut National des Sciences Appliquées, França, Março, 1994

[Monteiro94] Monteiro, António P.

"Processamento e Análise Automática de Ecocardiogramas"

Tese de Doutoramento, Faculdade de Engenharia da Universidade do Porto, 1994

[Nalwa86] Nalwa, V. S. ; Binford, T. O.

"On Detecting Edges"

IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8(5) pp. 699-714, 1986

[Rimmer92] Rimmer, Steven W.

"Graphical User Interface Programming"

Windcrest Books, 1991

[Rosenfeld82] Rosenfeld, A.; Kak, A.

"Digital Picture Processing"

Academic Press, Vol. 2, Second Edition, 1982

[Sahoo88] Sahoo, P. ; Soltani, S. ; Wong, A. ; Chen, Y.

"A Survey of Thresholding Techniques"

Computer Vision, Graphics and Image Processing, Vol. 41, pp. 233-260, 1988

[Santos89] Santos, Maria B.

"Análise Assistida por Computador de Cineangiogramas do Ventrículo Esquerdo"

Tese de Doutorado, Universidade de Aveiro, Abril, 1989

[Sarkar91] Sarkar, S. ; Boyer, L.

"On Optimal Infinite Impulse Response Edge Detection Filters"

IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-13(11), pp. 1154-1171, Novembro, 1991

[Schalkof89] Schalkoff, Robert J.

"Digital Image Processing and Computer Vision"

John Willey and Sons, Inc., 1989

[Schildt89] Schildt

"C: The Complete Reference"

McGraw-Hill, 1990

[Segerlind84] Segerlind, Larry J.

"Applied Finite Element Analysis"

John Willey & Sons, Inc. - 1984

[Serra82] Serra, Jean

"Image Analysis and Mathematical Morphology"

Academic Press, London, 1982

[Shen92] Shen, J. ; Castan, S.

"An Optimal Linear Operator for Step Edge Detection"

CVGIP: Graphical Models and Image Processing, Vol. 54, Nº 2, pp. 112-133, 1992

[Sperling70] Sperling G.

"Binocular Vision: A Physical and a Neural Theory"

American Journal of Psychology, Vol. 83, pp. 461-534, 1970

[Torre86] Torre, V.; Poggio, T.

"On Edge Detection"

IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8(2), pp. 147-163 Março, 1986

[Yuille92] Yuille, Alan L. ; Hallinan, Peter W. ; Cohen, David S..

"Feature Extraction from Faces Using Deformable Templates"

International Journal of Computer Vision, Vol. 8, Nº2, pp. 99-111, 1992