



COPPE/UFRJ

CLASSIFICAÇÃO/SEGMENTAÇÃO DE IMAGENS EMPREGANDO
GRADIENTES DE DEFORMAÇÃO

Ricardo Inácio Machado

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador: Antônio Alberto Fernandes de Oliveira

Rio de Janeiro
Outubro de 2010

CLASSIFICAÇÃO/SEGMENTAÇÃO DE IMAGENS EMPREGANDO
GRADIENTES DE DEFORMAÇÃO

Ricardo Inácio Machado

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

Prof. Antônio Alberto Fernandes de Oliveira, D.Sc.

Prof. Ricardo Guerra Marroquim, D.Sc.

Prof. Cristina Nader Vasconcelos, D.Sc.

Prof. Paulo Sérgio Rodrigues, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
OUTUBRO DE 2010

Machado, Ricardo Inácio

Classificação/Segmentação de imagens empregando Gradientes de Deformação/Ricardo Inácio Machado. – Rio de Janeiro: UFRJ/COPPE, 2010.

XIII, 67 p.: il.; 29, 7cm.

Orientador: Antônio Alberto Fernandes de Oliveira

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2010.

Referências Bibliográficas: p. 64 – 67.

1. Classificação. 2. Segmentação. 3. Informação de formas a priori. I. Oliveira, Antônio Alberto Fernandes de. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

A Deus.

Agradecimentos

Gostaria de agradecer, nesse momento em que chego à conclusão de mais uma etapa da minha vida, primeiramente a Deus, pois tudo que tenho feito até agora foi porque Ele me ajudou durante todo o percurso, me trouxe até o final desse curso, foi Ele quem me deu a força necessária para continuar em frente, a esperança de conseguir alcançar o que para mim era inalcançável, Ele quem colocou as pessoas certas no meu caminho para me ajudar e sem ele nada disso teria sido possível.

Gostaria também de agradecer ao meu orientador prof. Antônio, por ter sido o homem mais paciente do mundo para com esse aluno que não entendia quase nada. Quando eu pensava em pesquisar o problema ele já tinha duas ou três soluções ótimas para o problema (de cabeça). Que Deus continue te abençoando e lhe recompensando pelo esfoço e dedicação que o senhor tem tido para com os alunos, muito obrigado.

Aos meus familiares, que por vezes não entendiam o porquê de eu ficar ausente de tudo, meus pais Josafá e Regina, meus sogros Damião e Raquel, meu cunhado Tiago, meu irmão Moisés, cunhadas Viviane e Cristiane sem o apoio de vocês eu não teria chegado até aqui, os momentos difíceis, a falta que eu fazia... mas em fim, momentos que agora ficaram para trás, e, para frente, temos agora a alegria de termos alcançado mais essa vitória.

Gostaria também de fazer um agradecimento, em especial, para minha esposa Daniele, que me apoiou quase que incondicionalmente em tudo que fosse necessário para que hoje pudéssemos estar aqui escrevendo esse agradecimento. Dani, você foi uma bênção de Deus na minha vida, sem sua ajuda não teria passado nem da metade desse trabalho, muito obrigado.

Gostaria de agradecer também aos meus companheiros de trabalho Odilon, Felipe e Moisés que, por vezes, cobriram minhas ausências, ainda mais nesse final de curso, onde as coisas ficaram realmente complicadas. Agradeço, também, ao Vincenzo e ao Elias os quais me ajudaram em tudo quanto lhes foi possível, desde o início do curso. Muito obrigado a todos pelo apoio.

Gostaria de agradecer, também, a todos que contribuíram direta ou indiretamente para a conclusão deste trabalho. Mesmo que seu nome não tenha sido citado, também lhe sou igualmente grato, pois, se fosse citar a todos, precisaria de muitas folhas e tempo para escrever, o que não é o caso.

Muito obrigado a todos.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

CLASSIFICAÇÃO/SEGMENTAÇÃO DE IMAGENS EMPREGANDO GRADIENTES DE DEFORMAÇÃO

Ricardo Inácio Machado

Outubro/2010

Orientador: Antônio Alberto Fernandes de Oliveira

Programa: Engenharia de Sistemas e Computação

Este trabalho versa sobre o uso de informação conhecida a *priori* sobre a forma do objeto no processo de segmentação e é constituído de duas partes:

Na primeira, se definem modos principais para uma classe de formas lineares. A questão de se estabelecer uma correspondência entre os pontos de duas formas com cardinalidade diferente- que precisa ser resolvida para que se possa definir uma matriz de covariância- é superada aproximando-se cada uma delas pelo resultado da aplicação de um gradiente de deformação aos pontos de uma curva média da classe.

Na segunda parte, uma metodologia de contorno ativo, que visa minimizar uma energia definida pelo produto dos histogramas da parte interna e externa do contorno, é apresentada. Essa metodologia é mais custosa que uma snake padrão mas assume-se que empregando-se as informações disponíveis a *priori* pode-se partir de uma solução próxima do resultado final.

A partir do resultado final são propostos procedimentos para detecção de partes oclusas e mesmo para a substituição dessas partes por partes correspondentes extraídas de um modelo. Grande parte desta abordagem foi baseada no trabalho de Guillaume Charpiat *et al.* em [1].

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

IMAGE CLASSIFICATION/SEGMENTATION EMPLOYING DEFORMATION GRADIENTS

Ricardo Inácio Machado

October/2010

Advisor: Antônio Alberto Fernandes de Oliveira

Department: Systems Engineering and Computer Science

The subject of this work is the use of a-priori information about the shape of an object in the segmentation process. It has two parts:

In the first one, principal modes of a class of linear shapes are determined. The problem of establishing a correspondence between the points of two class members with different cardinalities - which must be solved so that a covariance matrix can be computed- is solved by approximating each one by the shape obtained by applying a multiple of a deformation gradient function to the points of an average curve of the class.

In the second part, we introduce an active contour methodology that aims to minimize an energy defined by the product of the histograms of the inner and outer parts of the contour. That methodology is computationally heavier than a standard snake but it is assumed that using the available a priori information it is possible to start the evolution close to the target curve.

That methodology is adapted to deal with occlusion, both for detecting the occluded parts and even to replace those parts by corresponding ones extracted from a model. A larger part of this approach was based in the work of Guillaume Charpiat *et al.* at [1].

Sumário

Lista de Figuras	xi
1 Introdução	1
1.1 Proposta do trabalho	2
2 Estado da arte	3
2.1 Segmentação	3
2.1.1 Modelos deformáveis	4
2.2 Classificação de imagens	14
2.2.1 Classificação supervisionada	14
2.2.2 Classificação Não Supervisionada	17
3 Metodologia Empregada	20
3.1 Introdução - Shape Priors	20
3.2 Descrição da Metodologia Utilizada	21
3.3 Estatísticas da formas	22
3.3.1 Média Empírica	22
3.3.2 Covariância Empírica	23
3.4 Shape Priors	26
3.4.1 Contexto	26
3.4.2 Probabilidade da forma C pertencer à classe Λ	28
3.5 Aplicação a um conjunto de formas	30
3.6 Registro Rígido	31
3.7 O Processo de Segmentação	33
3.8 Alternativas para o Tratamento da Oclusão	43
4 Implementação	47
4.1 Algoritmos	47
4.1.1 Pré Processamento	49
4.1.2 Processo de Segmentação	51
4.1.3 Classificador	53
4.2 Interface	56

4.3 Ferramentas de Desenvolvimento	58
5 Conclusões e Trabalhos Futuros	61
5.1 Conclusões	61
5.2 Trabalhos Futuros	62
Referências Bibliográficas	64

Lista de Figuras

2.1	Plano xy, gráfico da função level set e a curva do nível zero. Fonte: http://people.esam.northwestern.edu/~chopp/	8
2.2	Evolução do método level set. Em vermelho o gráfico da função level set, em azul o plano xy em cinza a curva em si (interseção). Fonte: http://commons.wikimedia.org/wiki/File:Level_set_method.jpg	9
2.3	Exemplo de uma classe com 16 amostras do objeto que se deseja segmentar. (Fonte: Cootles <i>et al.</i> [2])	10
2.4	Imagem de amostra com os pontos de interesse (<i>Landmarks</i>) para a borda. (Fonte: Cootles <i>et al.</i> [2])	11
2.5	Modelo com os <i>landmarks</i> , conjunto de pontos e <i>Shape Free Patch</i> . (Fonte: Cootles e Taylor [3])	12
2.6	Processos básicos para uma classificação supervisionada. Fonte: http://www.fgel.uerj.br/labgis/gis_atualizada/pdi/06_classificacao.htm	15
3.1	Um pequeno conjunto de quatro retângulos semelhante com dois tipos de deformações : Linha de cima: as quatro formas que compõem a amostra. Meio, à esquerda: a média destas formas (em azul), com os dois primeiros modos principais (em roxo), sucessivamente. Meio, a direita: a nova imagem a ser segmentada, construída aproximadamente como uma nova combinação das deformações observadas previamente adicionando uma nova deformação não observada. Linha de baixo: segmentação com o conhecimento apenas da média, modos principais e autovalores, assumindo uma distribuição Guassiana que, obviamente, não é satisfeita (à esquerda: inicialização; Meio: algumas etapas da evolução; direita: resultado de convergência). (Fonte: Charpiat <i>et al.</i> [1])	32
3.2	Alguns exemplos do conjunto de aprendizagem da estrela. (Fonte: Charpiat <i>et al.</i> [1])	33

3.3	A média do conjunto de estrela, com seus seis primeiros modos principais. (Fonte: Charpiat <i>et al.</i> [1])	34
3.4	Esquerda sup: qualquer inicialização razoável para o critério histograma de intensidade regional. Uma parte da imagem foi apagada a fim de aumentar a dificuldade.Meio sup: A mudança automática para a média no mesmo local, com tamanho semelhante.Restante: algumas etapas do processo de segmentação com o conhecimento da forma média. Este critério rígido encontra a localização da estrela, mas não possui informações sobre como adaptar a forma final. Com efeito, para um menor peso dos anteriores. (Fonte: Charpiat <i>et al.</i> [1])	35
3.5	Linha superior: a mesma inicialização que a anterior. Restante: Alguns passos do processo de segmentação sem nenhum <i>shape prior</i> , para o mesmo critério de região. O resultado não tem a forma global da estrela e inclui as pequenas bolas brancas. (Fonte: Charpiat <i>et al.</i> [1])	36
3.6	Região a ser considerada para se computar a variação do histograma de parte interna e externa da curva quando o snaxel S_k e seus vizinhos são movimentados numa iteração.	37
3.7	Resultado hipotético de uma evolução, cujo o contorno inicial considerou uma faixa de pixels (valor 6) não pertencentes ao objeto o que gerou whiskers.	38
3.8	Objeto com textura interna consideravelmente distinta do resto do objeto. Fonte: http://animais.blogmaneiro.com/fotos/peixes-ornamentais . . .	41
3.9	Curva inicial, projetada no subespaço dos modos principais da classe do objeto.	42
3.10	Objeto sem oclusão.	42
3.11	Objeto com oclusão em três partes.	43
4.1	Fluxograma para a segmentação (direita) e Fluxograma do pré-processamento(esquerda)	48
4.2	Representação dos pontos de Γ_i	49
4.3	Sistema de vetores que descreve \vec{g}_{proj} , que é a projeção do gradiente \vec{g} no ponto p em um vetor \vec{e}_p , o qual é perpendicular à \vec{d}_p um vetor que parte do centro de gravidade Cg até o ponto p . Com isso, é possível a obtenção do ângulo de rotação α_p	50

4.4	Alguns modelos segmentados pelo processo descrito.	
	Fonte: http://apenas24horas.blogspot.com/2010_06_01_archive.html	
	Fonte: http://neo-ambiente.blogspot.com/2010/04/vida-aves.html	52
4.5	Resultados obtidos pelos detector de oclusão. Na primeira (direita), são detectados as partes oclusas, na segunda os trechos do contornos, e na terceira a substituição do trechos selecionados pelos correspondentes na figura de referência.	52
4.6	Um conjunto de imagens reais, no lado esquerdo o contorno inicial para a segmentação, do lado direito o contorno segmentado e a oclusão já tratada.	
	Fonte: http://apenas24horas.blogspot.com/2010_06_01_archive.html	
	Fonte: http://neo-ambiente.blogspot.com/2010/04/vida-aves.html	
	Fonte: http://www.baixaki.com.br/papel-de-parede/25477-estrela-do-mar.htm	53
4.7	Tela principal de visualização da ferramenta com os exemplos utilizando um conjunto de com 150 elementos. Primeiro com um elemento gerado pela própria função e o outro com uma estrela.	55
4.8	Tela principal de visualização da ferramenta com os exemplos utilizando um conjunto de com 150 elementos. Sendo um peixe como elemento a ser classificado.	56
4.9	Tela após o processamento geral	57
4.10	Tela de visualização detalhada da poligonal	58
4.11	Foi utilizada uma poligonal a partir do conjunto de modelos, ao fundo a matriz de Transformada da Distancia do Circulo Médio (Opção selecionada STD) e os gradientes da dada matriz	59
4.12	Foi utilizado a poligonal do circulo médio, ao fundo a matriz de Transformada da Distancia da poligonal selecionada (Opção selecionada TD) e os gradientes da dada matriz em exibição, e as normais em cada ponto da curva média também em exibição.	60

Capítulo 1

Introdução

O primeiro passo na análise/processamento de imagens é o processo de segmentação. Esse processo pode ser definido como em delimitar na imagem a porção relativa a cada objeto representado nela. Atualmente existem inúmeras técnicas e algoritmos para se alcançar esse objetivo, a melhor escolha depende do problema que se deseja resolver, recursos disponíveis, entre outros fatores.

Entre os tipos de segmentação, podemos destacar aqueles que detectam os objetos por descontinuidades entre os pixels, aqueles que aplicam detecção de bordas dos objetos, contornos ativos, e os que se utilizam de métricas estatísticas como o ASM [2], AAM [3] entre outros.

Já a classificação de imagens representa uma importante etapa para várias aplicações de processamento digital de imagens e tem sido aplicada a vários problemas práticos como Visão Computacional, Compressão de imagens, entre outros. Dentre os métodos existentes existem aqueles que se destacam pela sua larga utilização, dentre eles podemos citar, por exemplo, os método ISODATA [4], K-Means [5], Maximum likelihood [6], Parallelepiped [7] entre outros que são amplamente utilizados em sensoriamento remoto como classificação não supervisionada.

Outra categoria para a classificação são as metodologias ditas supervisionadas, onde existe o conhecimento prévio do objeto de interesse bem como um conjunto de imagens de exemplo do objeto (modelos). Estes modelos, serão usados como amostra para a definição de uma classe para esse objeto e, a partir da classe, serão aplicadas várias métricas estatísticas com o objetivo da posterior identificação do objeto em uma nova imagem.

Técnicas como PCA (Principal Component Analysis) são utilizadas para obter um conjunto compacto capaz de representar o objeto em uma nova base, segundo Field [8] o PCA realiza uma transformação que permite que os dados de entrada possam ser representados com um reduzido número de vetores. Como uma generalização do PCA, foi desenvolvido um método chamado ICA (independent-components analysis) [9].

O PCA assume que a distribuição dos dados tem estrutura gaussiana enquanto o ICA generaliza o PCA permitindo distribuições não gaussianas e bases não ortogonais. Duas limitações comuns dessas técnicas são que elas não permitem que o ruído seja modelado separadamente do sinal, e também não permitem códigos mais completos, para os casos em que há mais funções base que dimensões de entrada.

Neste trabalho será mostrado um método para segmentação e/ou classificação de objetos dentro de uma imagem. Serão utilizadas técnicas para segmentação de imagens para definição dos gradientes de deformação e a partir deles, de uma forma geral, segmentar o objeto mesmo em casos que exista oclusão parcial do objeto.

No Capítulo 3 será mostrado a metodologia empregada para a realização do trabalho assim como boa parte do embasamento teórico. O Capítulo 4 mostra a ferramenta implementada. No Capítulo 5 serão mostrados os resultados obtidos, conclusões e trabalhos futuros. O estado da Arte sobre os temas de segmentação e classificação, será descrito no Capítulo 2.

1.1 Proposta do trabalho

Apresentar um método que se utiliza de informação a priori sobre a forma do objeto no processo de segmentação, permitindo que o objeto mantenha a identidade como um membro da classe, mesmo que o objeto esteja parcialmente ocluso.

Capítulo 2

Estado da arte

Neste capítulo, será apresentada uma síntese de alguns estudos com o objetivo de demonstrar os trabalhos científicos que vêm sendo desenvolvidos, os quais possuem contexto ou mesmo alguma definição que sejam referentes ao tema tratado neste trabalho. Visto que o foco do trabalho é classificação/segmentação dos elementos de uma classe, serão abordados como referencial para esse capítulo dois assuntos: A segmentação e a classificação de imagens.

2.1 Segmentação

Uma das principais etapas do processamento digital de imagens é a de segmentação, que consiste em delimitar na imagem a porção relativa a cada objeto representado nela.

As diferentes metodologias para se obter a segmentação de uma imagem podem ser classificadas em dois grupos principais. No primeiro, leva-se em consideração principalmente o contorno do objeto a ser segmentado, sendo nestes, utilizadas técnicas para detectar descontinuidades bruscas entre pixels vizinhos. No segundo, se buscam similaridades entre os pixels vizinhos e encontrando-as, se expande o segmento. Primeiramente se define um modelo para caracterizar o segmento e se indica um determinado ponto (ou conjunto de pontos) como pertencente ao segmento. A partir desse segmento inicial, caso seja detectado que um pixel externo adjacente a ele satisfaz as condições do modelo dado, o mesmo é incorporado ao segmento fazendo com que novos pixels se tornem adjacentes a ele e assim sucessivamente até que uma dada condição de parada seja alcançada.

A Segmentação via detecção de borda baseia-se no fato de que existe uma variação brusca no valor dos atributos em pixels vizinhos quando os mesmos pertencem a objetos diferentes. Abordagens clássicas para esse problema buscam descontinuidades fazendo uso de operadores diferenciais de primeira e segunda ordem. Uma das mais eficientes nessa tarefa é a que emprega o algoritmo de Canny [10].

Para Canny os algoritmos de detecção de borda devem seguir três critérios de desempenho:

1 – Boa detecção: Deve ter uma baixa probabilidade de falha em marcar pontos de borda reais, e baixa probabilidade de marcar pontos que não estejam no entorno da borda.

2 – Boa Localização: Ao longo de um corte transversal a borda, e os pontos marcados como pertencentes à ela devem estar tão perto quanto possível da intersecção da borda real com o corte.

3 – Apenas uma resposta para uma borda simples, ou seja, o conjunto de pontos marcados deve ter apenas um pixel de largura.

Entre os procedimentos que expandem o segmento a partir de uma versão inicial e que se baseiam na similaridade entre pixels vizinhos, podemos citar os métodos de crescimento de regiões (*region growing*) [11], binarização [12], dividir e mesclar (*split and merge*) [13] e *watershed* [14], e métodos que se utilizam de estruturas baseadas em teoria dos grafos como o *Min-cut/Max-flow* [15], as Pirâmides [16] e a IFT (Transformada da Floresta de uma imagem) [17]. Todos eles segmentam a imagem utilizando propriedades locais para definir as regiões, considerando-se que elas representam objetos “contínuos”, ou seja, que os pixels de uma dada região formam necessariamente um conjunto conexo e possuem níveis similares dos atributos avaliados.

2.1.1 Modelos deformáveis

Dado que na metodologia em que nos iremos aprofundar nesse trabalho consideramos, essencialmente, contornos obtidos por um processo de deformação efetuado a partir de uma curva inicial, vamos tratar com mais especificidade a técnica de segmentação que usa modelos deformáveis tendo em vista os elementos comuns com a que será proposta aqui.

As técnicas de segmentação baseadas em modelos deformáveis podem ser divididas em duas categorias:

- Modelos de contornos ativos – que podem representar qualquer objeto na imagem.
- Modelos baseados nas formas – que podem representar uma classe de objetos na imagem, e possuem baixo grau de liberdade (imposto pela própria classe previamente processada).

Modelos de Contornos Ativos

Geometricamente, um contorno ativo é uma poligonal definida no domínio da imagem. Fisicamente, os vértices dessa poligonal interagem entre si com uma intensidade que depende de sua posição relativa e são afetados por forças externas que são função de sua posição absoluta. Contornos ativos possuem alto grau de liberdade permitindo que eles possam ser utilizados na segmentação de qualquer objeto.

Os modelos de contorno ativo mais utilizados hoje se baseiam no modelo de Snakes proposto inicialmente no trabalho de KASS *et al.*[18]. Esse modelo é capaz de se ajustar aos contornos salientes de uma imagem. Snakes são segmentos compostos por n vértices regidos por uma formulação de energias internas e externas, as quais fazem com que este se deforme no plano da imagem. A evolução da Snake consiste em encontrar um equilíbrio entre forças internas e externas que se aplicam aos seus snaxels. De forma equivalente pode-se dizer que ela busca um posicionamento para seus snaxels que minimize um dado funcional que representa a energia da snake. A energia interna é gerada a partir da geometria do segmento e a externa a partir de campo potencial obtido de uma propriedade da imagem na qual a snake está inserida. Sendo a representação paramétrica de uma snake dada por:

$$\Omega = [0, 1] \mapsto IR^2 \quad (2.1)$$

$$s \rightarrow v(s) = (x(s), y(s))$$

Podemos definir energia de uma snake conforme indicado abaixo:

$$E_{snake} = E_{int} + E_{image} + E_{con} \quad (2.2)$$

Sendo E_{int} a energia interna, E_{image} a energia do campo, isto é aquela obtida em função dos dados da imagem, e E_{con} se refere a energia introduzida por outros meios, seja pela ação do operador para fazer a snake escapar de um mínimo local de energia, seja para fixar as suas extremidades se for o caso, ou para fazê-la se ajustar a um modelo conhecido *a priori*, entre outras razões. Aqui por concisão, não consideraremos esse tipo de energia.

A energia interna tem usualmente a seguinte expressão

$$E_{int} = \oint ((w_1|v_s(s)|^2 + w_2|v_{ss}(s)|^2)/2)ds \quad (2.3)$$

onde w_1 e w_2 são pesos que controlam, respectivamente, a elasticidade e a rigidez da snake.

Como o objetivo é a detecção de bordas que são caracterizadas por um contraste

mais alto, devemos fazer com que a energia de campo decresça em regiões da imagem nas quais o contraste aumenta, de forma que as bordas se tornem curvas de mínima energia de imagem. A maneira mais simples de fazer isso é definir essa energia por

$$E_{image} = \frac{1}{2} \oint -\|\nabla(I(v(s)))\|^2 ds \quad (2.4)$$

Tendo em vista as expressões das energias interna e da energia da imagem dadas nas equações 2.3 e 2.4 respectivamente, a minimização do funcional de energia composto por elas é caracterizada pela equação de Euler correspondente que é dada por:

$$\begin{cases} -w_1 \frac{d^2 v}{ds^2}(s) + w_2 \frac{d^4 v}{ds^4}(s) - \nabla(\|\nabla(I(v(s)))\|^2) = 0 \\ v(0) = v(1), v'(0) = v'(1) \end{cases} \quad (2.5)$$

Quando a equação acima é satisfeita um mínimo local de energia da snake é atingido. Esse método de segmentação possui vantagens como o simples fato de manter como invariante a geração de curvas fechadas que é uma propriedade inerente a borda de um segmento. A incorporação de informações conhecidas *a priori* sobre os objetos a serem segmentados é também facilitada. Exemplos:

i) Características da forma adotada e analisada podem ser usadas para definir uma energia adicional que pode fazer com que os contornos gerados tendam a assumir essas características.

ii) A suavidade da borda pode ser favorecida aumentando-se o peso da energia interna.

Entretanto, existe uma sensibilidade muito grande quanto à sua inicialização exigindo uma boa estimativa inicial para garantir o funcionamento adequado do método. Uma snake pode não evoluir caso a energia potencial do contorno do objeto não seja suficientemente intensa para se propagar pela imagem e seu posicionamento inicial esteja muito afastado do contorno do objeto. Outro problema deste método é a limitação relativa à topologia do segmento limitando-se este à uma curva fechada ou a um segmento de reta, sendo incapazes de se auto-dividirem ou de duas delas se juntarem num contorno único.

Tendo em vista esses fatos, muitas formulações têm sido desenvolvidas com a finalidade de minimizar a dependência da snake à sua inicialização bem como fazer com que a mesma possa lidar com mudanças topológicas de forma adequada. Entre elas podemos citar:

Snakes balloon [19]: A proposta dessa formulação é modificar a força externa para alcançar resultados menos sensíveis ao ruído e a inicialização da snake. Basicamente, a idéia é fazer com que a curva se comporte como um balão que é inflado por uma força adicional normal ao segmento fazendo com que o mesmo possa evoluir

para a borda do objeto, mesmo que parta de pontos que não tenham sido alcançados pela energia potencial do contorno. Com isso, a inicialização da snake não necessariamente precisa estar tão perto da borda do objeto. Além disso, definindo-se uma força de campo de tamanho unitário aplicada apenas nos pixels onde o gradiente da imagem ultrapassa um dado limiar, o bloqueio a evolução da snake passa a ser dependente do tamanho do conjunto desses pixels que ela encontra, e não do valor do contraste neles. Assim, bordas verdadeiras que são sequências longas desses pixels conseguem segurar o balloon enquanto que segmentos eventualmente pequenos são ultrapassados, mesmo que o contraste neles seja grande

Há formulações de **Snakes** onde se redefinem as energias de campo de modo que elas possam exercer influência em uma maior área da imagem como, por exemplo, o GVF (*Gradient Vector Flow*) [20]. Nelas a energia externa é redefinida de modo que a área de influência do campo gradiente desta função aumente sem que a borda do objeto perca a definição ou resolução significativamente, ao contrário do que ocorreria se aplicássemos reiteradamente um filtro gaussiano para estender a área de influência das bordas.

Snakes que empregam programação dinâmica [21] são formulações que substituem o problema variacional que, segundo os autores, não possui garantias de encontrar o ótimo global para a solução. A proposta neste caso é a utilização da Programação dinâmica que garante encontrar o mínimo global, porém com altos custos computacionais [22]. Normalmente, a snake é evoluída até se posicionar no entorno da borda e só então a programação dinâmica é aplicada como se fosse um pós-processamento.

T-Snakes[23] bem como suas variantes, são formulações que visam resolver a questão topológica da snake. O método busca uma formulação paramétrica para a snake que possua habilidades topológicas. Baseia-se na conversão das questões topológicas para o contexto da Topologia Combinatória. Seus elementos básicos são: decomposição simplicial (triangulação) do domínio, a restrição dos snaxels se posicionarem apenas nas arestas dessa triangulação e uma *função característica*, definida nos nodos da triangulação, utilizado para identificar as partes internas e externas das snakes.

Outras metodologias, além dos modelos paramétricos, são as snakes geométricas ou implícitas [24] que empregam métodos do tipo *Level Set*, propostos inicialmente por James Setian [25] e *Fast Marching* [26].

Snakes geométricas evoluem no tempo de acordo com a medidas geométricas extraídas da própria imagem. Como mostra o trabalho de Malladi *et al.*[27] o contorno ativo é visto como a configuração de nível zero de uma função definida no domínio da imagem. Variando-se uma curva pertencente a essa configuração pode-se dividir em várias ou duas delas podem-se unir numa única. Assim, a snake pode mudar

de topologia permitindo a detecção de vários objetos. O movimento da snake por esse modelo em cada ponto tem sempre a direção da normal à curva no ponto e é determinado por uma função de velocidade, que depende da curvatura nesse ponto, logo, a formulação para o movimento segue a dinâmica Euleriana.

$$F = (L, G, I) \tag{2.6}$$

Onde, L corresponde às informações geométricas locais (curvatura e direção normal); G corresponde às propriedades globais da curva as quais dependem da sua localização, forma e posição e é calculado a partir das integrais de linha e/ou equações diferenciais associadas a elas; I corresponde às propriedades independentes da curva e referentes à própria imagem na qual a curva se desloca.

Métodos Level set consistem em inserir a curva atual em uma superfície que intercepta o plano xy exatamente na curva. Esta superfície é chamada de gráfico da função level set e a curva em si é chamada de curva do nível zero (ou do level set zero). Esses elementos são mostrados nas figuras 2.1 e 2.2

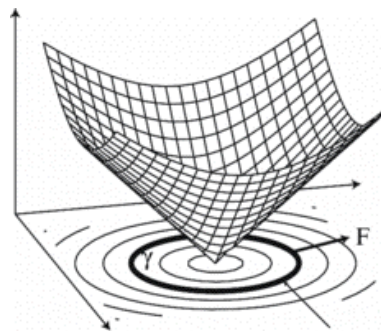


Figura 2.1: Plano xy , gráfico da função level set e a curva do nível zero.
 Fonte: <http://people.esam.northwestern.edu/~chopp/>

Métodos de Level set são utilizados para problemas nos quais a função velocidade pode ser positiva em alguns lugares e negativas em outros, dispensando a necessidade da curva inicial ser inteiramente interna ou inteiramente externa ao objeto que se quer segmentar, o que dá generalidade ao seu uso.

No contexto dos métodos *Fast Marching* a função level set é o tempo que a curva de nível zero levaria para atingir um pixel da imagem se submetida ao campo determinado pelos valores correntes da função de velocidade. Para computar o tempo de evolução, o método emprega um algoritmo de Dijkstra, o qual a cada iteração avalia o tempo - t^* - necessário para se atingir o pixel mais próximo - p^* - entre os que ainda não foram cobertos pela evolução da curva até aquele momento. Explorando a técnica de *Heapsort*, p^* pode ser identificado de maneira eficiente.

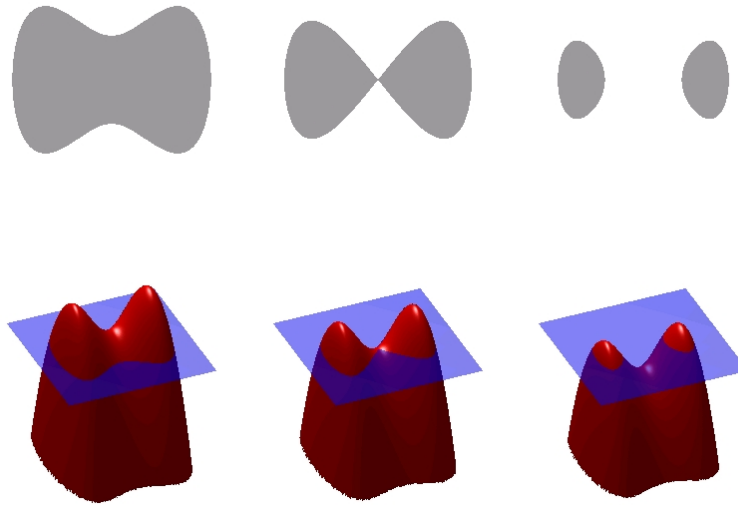


Figura 2.2: Evolução do método level set. Em vermelho o gráfico da função level set, em azul o plano xy em cinza a curva em si (interseção).

Fonte: http://commons.wikimedia.org/wiki/File:Level_set_method.jpg

Modelos Baseados na Forma

São metodologias que se baseiam na definição de um modelo a partir de um conjunto amostral do tipo de objeto que se deseja segmentar. Utilizando-se do conhecimento prévio da forma que esperada para o objeto a ser segmentado, pode-se obter segmentações de melhor qualidade sem fazer crescer o esforço computacional. O conjunto de amostras de um mesmo tipo define uma classe, conforme exemplificado na figura 2.3.

Para cada classe são definidas métricas e condições as quais permitem definir uma equação de evolução que mantém um modelo dentro dessa classe. Para definir essas métricas/condições consideram-se informações locais da imagem tais como curvatura e gradiente, e também, informações globais como tamanho ou características específicas. Nas abordagens mais largamente difundidas essas informações são utilizadas para construir uma função de probabilidade a posteriori e o modelo se movimenta de forma a aumentar essa probabilidade gradativamente convergindo para um máximo local. Em vista disso, essa abordagem é referenciada por MAP – maximum a posteriori probability. Essa técnica permite que o modelo tenha liberdade suficiente para segmentar um objeto na imagem mesmo que esse seja ligeiramente diferente dos modelos definidos a priori. Além disso, se o modelo mantém características da classe na qual ele foi definido pode-se segmentar um objeto mesmo que haja a oclusão do mesmo por outro objeto.

Entre os diversos modelos baseados em formas podemos citar dois tipos: *Active Shape Model* e *Active Appearance Model* (AAM).



Figura 2.3: Exemplo de uma classe com 16 amostras do objeto que se deseja segmentar. (Fonte: Cootles *et al.* [2])

Active Shape Models (ASM) são técnicas baseadas na modelagem estática de um conjunto de amostras para uma dada forma, de modo que, a partir dessa modelagem obtenha-se uma forma média, a qual será utilizada como contorno ativo inicial. Utilizando, então, métricas como PDM (Point Distribution Model), covariância e análise das componentes principais (PCA) a evolução desse contorno ativo pode ser realizada de modo a que ele atenda sempre às condições que caracterizam a classe a partir da qual sua primeira versão foi computada.

Conforme mencionado acima e proposto originalmente por Cootles *et al.*[2] o modelo baseia-se, primeiramente, em se definir uma forma média para a classe. Para tal, são definidos pontos de interesse (*landmarks*), isto é, pontos utilizados como referências, localizados principalmente em vértices e pontos relevantes de curvas. A poligonal gerada por esses pontos deve representar bem a borda da amostra. Dentro de uma classe, cada amostra deve ter o mesmo número de pontos de interesse e, em todas elas cada ponto deve representar uma mesma característica dos objetos da classe de forma a simplificar o estabelecimento de correspondências entre partes de duas amostras. Assim, no exemplo conforme é mostrado na figura 2.4 o ponto 0, por exemplo, deve se referir à extremidade superior do segmento mais a direita em todas as amostras e assim por diante.

Após a definição dos pontos de interesse, toda a classe é alinhada e normalizada, ou seja, são aplicadas operações de escala, rotação e translação que forem necessárias a cada elemento da classe. Essas operações são aplicadas visando minimizar o somatório das distâncias entre as posições de cada ponto de interesse em cada par de elementos da classe. Tendo um conjunto com amostras alinhadas, o ponto médio



Figura 2.4: Imagem de amostra com os pontos de interesse (*Landmarks*) para a borda. (Fonte: Cootles *et al.* [2])

\bar{x} de cada conjunto de pontos de interesse de uma mesma característica (mesmo índice) é calculado utilizando-se:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \quad (2.7)$$

O conjunto de médias dos pontos de interesse compõe a forma média para a classe. A partir disso, pode-se calcular o desvio em relação à média $dx_i = x_i - \bar{x}_i$ e a matriz de covariância $S = \frac{1}{N} \sum_{i=1}^N dx_i dx_i^T$.

Os eixos principais do conjunto de pontos de interesse fornecem os modos de variação dos pontos da curva, são descritos por $p_k (k = 1, 2, 3 \dots 2n)$. Cada p_k é o autovetor unitário de S , associado a um autovalor λ_k .

A partir daí cada poligonal com um número de vértices igual ao de “landmarks” da classe pode ser obtida adicionando-se à curva média uma combinação linear desses autovetores.

O algoritmo ASM tenta encontrar o posicionamento de um contorno de uma dada classe em uma nova imagem. Basicamente, seu procedimento consiste em alternar os seguintes passos:

- Localização de uma melhor posição para cada ponto dentro de uma dada vizinhança.
- Atualização dos parâmetros do modelo, que são substituídos por aqueles que melhor correspondem a estas novas posições encontradas.

A definição da melhor posição de um ponto pode ser a de maior contraste na vizinhança, valorizando-se assim as bordas. Os parâmetros a serem atualizados são os coeficientes relativos a um dado número de modos principais, da combinação linear de autovetores que reproduz a curva gerada pelo posicionamento corrente dos pontos de interesse.

Desde a introdução do ASM, uma série de trabalhos foram publicados focando o aprimoramento de seus procedimentos básicos tais como se fazer a definição dos pontos de interesse de forma automatizada [28], e ajustar o modelo de forma que ele considere também informações de textura, entre outras melhorias.

Por sua vez, o Active Appearance Model (AAM) [3] é uma generalização do ASM que inclui, também, informações de textura extraídas de imagens que representam as amostras. Esta técnica permite a modelagem estatística tanto da forma quanto da textura de um objeto. O processo utilizado no AAM é similar ao executado pelo ASM. Também se definem pontos de interesse e se calcula a curva média para o conjunto de modelos de cada classe. A informação de textura é obtida separada da informação de forma, fazendo-se um warping entre as regiões delimitadas pela borda do objeto - caracterizada por um subconjunto de pontos de interesse - e pela curva média das bordas dos objetos da classe. Transferem-se assim as texturas de todos os objetos da classe para a região delimitada pela curva média das bordas, obtendo o que o autor chama de “*shape-free patch*”. Para esse conjunto de texturas transferidas constrói-se, então, um modelo estatístico. A figura 2.5 ilustra a abordagem empregada pelo esquema AAM.

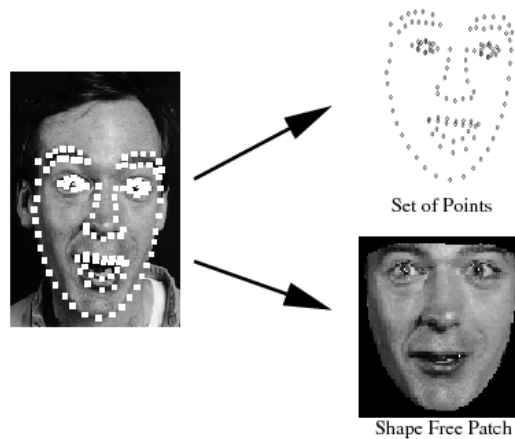


Figura 2.5: Modelo com os *landmarks*, conjunto de pontos e *Shape Free Patch*. (Fonte: Cootles e Taylor [3])

Cada textura transferida é amostrada nos mesmos pontos da região delimitada pela curva média das bordas para formar um vetor de textura g_{im} .

Para minimizar os efeitos da variação de luminosidade entre uma e outra imagem, é feita uma normalização das amostras aplicando-se a elas uma escala α e uma compensação β .

$$g = (g_{im} - \beta)/\alpha \quad (2.8)$$

Os valores de α e β são escolhidos de forma que cada vetor de textura passe a ter média zero e variância 1. Isso significa fazer β ser a média do vetor de textura e α o seu desvio padrão.

Aplicando-se a análise de componentes principais a esses dados normalizados

obtemos para cada vetor de textura uma representação da forma:

$$g = \bar{g} + P_g b_g \quad (2.9)$$

Onde \bar{g} é a média dos vetores de textura normalizados, P_g é um conjunto de modos de variação ortogonais e b_g é um conjunto de coeficientes reais.

A textura *shape-free* de cada objeto da classe pode ser gerada a partir desses coeficientes b_g e dos parâmetros de normalização α e β . Para se obter uma transformação linear - eliminando termos constantes - os parâmetros da normalização podem ser representados utilizando o vetor $u = (u_1, u_2)^T = (\alpha - 1, \beta)^T$.

Assim, uma textura *shape-free*(g_{im}) pode ser obtida de sua versão normalizada pela expressão:

$$T_u(g) = (1 + u_1)g + u_2 \mathbf{1} \quad (2.10)$$

Substituindo g na equação acima, por sua expressão dada em 2.9 podemos escrever:

$$g_{im} = T_u(\bar{g} + P_g b_g) = (1 + u_1)(\bar{g} + P_g b_g) + u_2 \mathbf{1} \quad (2.11)$$

A textura e a forma podem ser sumarizadas pelos vetores b_s e b_g . Desde que seja possível haver correlação entre a variação da textura e da forma. Aplica-se mais um PCA aos dados como se segue. Para cada exemplo é gerado um vetor concatenado.

$$b = \begin{pmatrix} W_s b_s \\ b_g \end{pmatrix} = \begin{pmatrix} W_s P_s^T (x - \bar{x}) \\ P_g^T (g - \bar{g}) \end{pmatrix} \quad (2.12)$$

Onde, W_s é uma matriz diagonal de pesos para cada parâmetro de forma, permitindo então a diferença em unidades entre as formas e as texturas (níveis de cinza). Aplica-se o PCA a esses vetores obtendo-se um modelo entendido.

$$b = P_c c \quad (2.13)$$

Onde, P_c são os autovetores e c é um vetor de parâmetros de textura, capazes de controlar a forma e os níveis de cinza da textura. Desde que os parâmetros da forma e da textura em níveis de cinza possuam média zero, c também será zero.

Logo, a forma e os níveis de cinza podem ser expressos diretamente como:

$$\begin{aligned} x &= \bar{x} + P_s W_s^{-1} P_{cs} c \\ g &= \bar{g} + P_g P_{cg} c \end{aligned} \quad (2.14)$$

Onde,

$$P_c = \begin{pmatrix} P_{cs} \\ P_{cg} \end{pmatrix} \quad (2.15)$$

Reduzindo,

$$\begin{aligned} x &= \bar{x} + Q_s c \\ g &= \bar{g} + Q_g c \end{aligned} \quad (2.16)$$

Onde,

$$\begin{aligned} Q_s &= P_s W_s^{-1} P_{cs} \\ Q_g &= P_g P_{cg} \end{aligned} \quad (2.17)$$

Uma imagem de exemplo pode ser sintetizada para um dado c pela geração das imagens em níveis de cinza *shape-free* a partir do vetor g e fazendo um warping utilizando os pontos de controle descritos por x .

2.2 Classificação de imagens

Podemos definir a classificação de imagens como o processo de extração e tratamento de informações contidas numa imagem com o objetivo de identificar padrões e objetos contidos nela que nos permitam associá-la - ou associar cada um de seus segmentos - a uma classe de um conjunto previamente definido.

Podemos dividir os processos de classificação em duas categorias. Supervisionada, é aquela em que o processo de classificação se baseia em um conjunto de amostras de cada uma das classes a serem identificadas. E, não-supervisionada, onde o processo ocorre sem conhecer previamente exemplos de representantes das classes.

2.2.1 Classificação supervisionada

Nesse tipo de metodologia, uma estratégia padrão é o usuário, em função de uma análise prévia, definir manualmente os pixels que serão utilizados para identificar as classes no conjunto de amostras. A partir desses pixels são extraídas informações estatísticas para as classes que serão utilizadas para posterior identificação das mesmas, dada uma imagem real. Para que o processo funcione corretamente, é necessária a correta utilização por parte do usuário para a definição das classes conforme o algoritmo, ou classificador, a ser empregado. O usuário deve identificar várias áreas representativas de cada classe no conjunto de amostras. É importante que cada uma dessas áreas seja uma amostra homogênea da respectiva classe. Homogênea,

neste caso, não quer dizer, necessariamente, uniforme, podendo significar, quando a classe está associada a um pattern característico, que pixels similares no pattern serão similares na amostra sem apresentar diferenças determinadas pela variação da luminosidade ou mesmo ruído. Entretanto, as amostras de cada classe devem incluir toda a variabilidade de valores que os pixels empregados para caracterizá-la, podem assumir. Além disso, deve-se observar que o número de pixels de treinamento empregados para que se possa reconhecer uma classe com uma dada precisão aumenta conforme a diferenciação entre as classes diminui. A figura 2.6 mostra a sequência de processos para uma classificação supervisionada dentro do contexto de sensoriamento remoto.

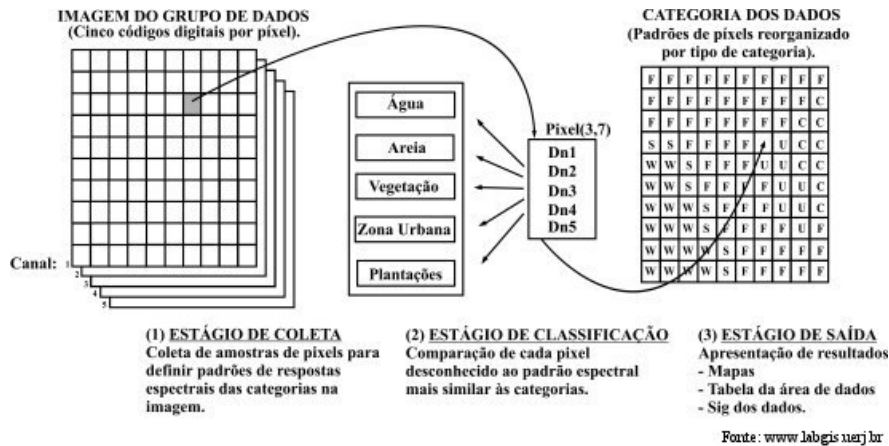


Figura 2.6: Processos básicos para uma classificação supervisionada.
 Fonte: http://www.fgel.uerj.br/labgis/gis_atualizada/pdi/06_classificacao.htm

Os principais algoritmos de classificação supervisionada utilizados são os seguintes:

Máxima verossimilhança (Maximum likelihood) [6], onde os pixels/áreas processados são associados à classe a qual possuam maior probabilidade de pertencerem. A forma convencional desse algoritmo assume que a distribuição de probabilidade para cada classe espectral é da forma de um modelo de normal multivariada com dimensões com o mesmo número de bandas espectrais. Podemos descrever o algoritmo conforme abaixo:

Seja x um vetor que representa o pixel/área, m_i o vetor médio para a classe i e Σ_i sua matriz de covariância. Σ_i é uma matriz $N \times N$, sendo N o número total de bandas da imagem e C o número total de classes.

$$x \in \omega_i \quad \text{se} \quad g_i(x) > g_j(x) \quad \text{para} \quad \text{todo} \quad j \neq i \quad (2.18)$$

Onde, $g(\cdot)$ é uma função de discriminação:

$$g(x) = -\ln|\Sigma_i| - (x - m_i)^t \Sigma_i^{-1} (x - m_i) \quad \text{para} \quad i = 1 \dots C \quad (2.19)$$

Linear discriminant analysis [29], é um método estatístico que consiste na construção de uma função discriminante definida a partir de dois grupos. E, é capaz de discriminar a qual grupo um dado elemento pertence. A função discriminante é formulada por uma combinação linear de variáveis características $D = a_0 + \sum_{i=1}^n a_i X_i$, sendo n o número de variáveis, X_i os valores das variáveis características e a_i são pesos estimados durante a fase de treinamento. O processo pode ser feito em dois estágios, o primeiro, chamado de *stepwise*, onde as variáveis características são avaliadas e identificadas as que efetivamente serão úteis para a formulação da função discriminante. O segundo, as variáveis selecionadas são utilizadas para a definição dos pesos para cada variável característica com o objetivo de alcançar a máxima separação entre as classes na função discriminante.

Paralelepípedo (Parallelepiped) [7], Esta é uma das abordagens mais simples para a classificação supervisionada, onde, a caracterização da classe é definida pelo fato de um pixel ou um conjunto de medidas obtidas para ele ou uma dada área estar dentro de um conjunto de definição simples como um paralelepípedo. Pode haver dois casos especiais para essa abordagem. No primeiro, o pixel/área pode pertencer a mais de uma classe e neste caso ele é colocado em uma nova classe caracterizada pela sobreposição de duas ou mais classes originalmente definidas. No segundo, o pixel/área pode não pertencer a nenhuma classe e neste caso ele é associado a uma classe nula.

Mínima distância (Minimum distance) [30], realiza a classificação de um dado da imagem baseado na menor distância encontrada para esse dado em relação ao conjunto de categorias (classes). Resumidamente o algoritmo Mínima distância pode ser definido da seguinte maneira:

Seja x a representação de uma amostra que se deseja classificar no espaço característico de uma dada classe ω_i , e z_i a representação de um protótipo dessa categoria nesse mesmo espaço, cuja dimensão representaremos por m . Seja ainda C o conjunto de categorias ω_i . A definição de que x pertence à classe ω_i pode ser dada por:

$$x \in \omega_i \iff d(x, z_i) = \min_j d(x, z_j), \quad j \in C \quad (2.20)$$

Sendo a distância euclidiana:

$$d(x, z_i) = \sqrt{\sum_{l=1}^m (x_l - z_{il})^2} \quad (2.21)$$

Para a distância, qualquer outra métrica pode ser utilizada. Uma variação deste algoritmo, utilizado por alguns autores, seria a utilização da métrica de Mahalanobis para a definição de distância. A distância de Mahalanobis é uma medida de distância baseada nas correlações entre variáveis tornando a métrica invariante à escala, ou

seja, não depende da escala das medições, e possui a característica de tornar possível a medição da distância entre variáveis não diretamente relacionadas, ou seja, medidas de características diferentes que não possuam uma relação direta. A única restrição da métrica é que os dois conjuntos de dados devem ter a mesma dimensão, entretanto o mesmo não é válido para o número de elementos dos conjuntos. Vale ressaltar que quando a matriz de covariância é a matriz identidade, a distância de Mahalanobis é igual à distância euclidiana.

A distância de Mahalanobis pode ser definida como:

$$D_m(x) = \sqrt{(x - \mu)^T \Sigma^{-1} (x - \mu)} \quad (2.22)$$

Onde, μ é a média, Σ é a matriz de covariância para o vetor multivariado x .

Entre dois vetores aleatórios x e y com mesma distribuição a distância pode ser calculada conforme equação abaixo:

$$d(x, y) = \sqrt{(x - y)^T \Sigma^{-1} (x - y)} \quad (2.23)$$

Para casos em que a matriz de covariância é diagonal, a medida de distância resultante é chamada distância euclidiana normalizada:

$$d(x, y) = \sqrt{\sum_{i=1}^p \frac{(x - y)^2}{\sigma_i^2}} \quad (2.24)$$

Sendo σ_i o desvio-padrão de x_i no conjunto amostral.

2.2.2 Classificação Não Supervisionada

O processo consiste em classificar os pixels/áreas sem a utilização de um conjunto de modelos para treinamento a priori. Utiliza-se de mecanismos de agrupamento ou clusterização baseados em medidas de similaridade entre pixels vizinhos.

Existem muitos algoritmos de clusterização que podem ser utilizados para a realização do agrupamento de pixels como por exemplo:

Simple One-Pass Clustering [5], consiste no agrupamento nos pixels baseado em alguma medida de similaridade e um *threshold*. O processo se inicia verificando as classes existentes. Caso não haja, o primeiro item que estiver sendo avaliado passa a compor a primeira classe. A partir daí, cada novo elemento para o qual a medida de similaridade em relação a qualquer das classes já existentes, não seja superior ao *threshold*, introduz uma nova classe. Em caso contrário ele é inserido na classe onde obtenha a maior similaridade. Em decorrência dessa inserção a medida de similaridade relativa a essa classe é atualizada, o que normalmente implica em pelo menos se atualizar o centróide dos elementos da classe no espaço de representação.

K-Means[5], permite que o usuário defina a priori a quantidade (K) de clusters que serão gerados, daí o nome do método. O processo consiste em definir centróides (inicialmente aleatórios) para cada cluster e considerar como pertencendo ao cluster os elementos que estiverem no polígono de Voronoi do centróide. Após a definição dos elementos de cada cluster seu centróide é atualizado dando origem a novos polígonos de Voronoi. Com isso, os elementos podem ser reclassificados seguidamente. O processo acima é iterado até que não hajam mais alterações nos centróides dos clusters.

Fuzzy C Means [31], permite que uma parte dos dados pertença a dois ou mais clusters e é baseado na minimização da seguinte função objetivo:

$$J_m = \sum_{i=1}^N \sum_{j=1}^C (u_{ij})^m \|x_i - c_j\|^2 \quad , \quad 1 \leq m < \infty \quad (2.25)$$

Onde m é um número real maior que 1, u_{ij} é a pertinência de x_i no cluster j , x_i é o i -ésimo elemento d-dimensional de x , c_j é o centro d-dimensional do cluster, e $\| * \|$ é uma norma que expressa uma medida de similaridade qualquer entre x_i e o centro do cluster.

A atualização dos valores de u_{ij} e dos centros dos clusters c_j é dada por:

$$u_{ij} = \frac{1}{\sum_{k=1}^C \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}} \quad c_j = \frac{\sum_{i=1}^N u_{ij}^m \cdot x_i}{\sum_{i=1}^N u_{ij}^m} \quad (2.26)$$

O critério de parada para a iteração dessas funções é $\max_{ij} \{|u_{ij}^{k+1} - u_{ij}^k|\} < \epsilon$ onde ϵ é um limite definido entre 0 e 1. Este procedimento converge para um mínimo local ou para um ponto de sela de J_m .

Minimum Distribution Angle [5], é um procedimento iterativo similar ao K-means. A diferença está na forma de cálculo da distância entre o elemento e o centro do cluster. Este método trata o centro do cluster e o elemento como vetores direcionais que partem da origem. Segundo o método, o ângulo entre os vetores fornece uma medida de similaridade da mistura das características (ignorando a magnitude). Este conceito é similar a uma mistura de tintas. Por exemplo. A mistura de tintas de cores vermelha e azul na mesma proporção produzem uma tinta púrpura, não importa a quantidade de tinta utilizada.

Isodata Classification (Iterative Self-Organizing Data Analysis Technique) [4], é um metodo que utiliza uma abordagem iterativa que incorpora procedimentos heurísticos (tentativa e erro) para computar as classes e procedimentos de subdivisão, combinação e descarte das classes preliminares para obter como resultado final a definição de um conjunto ótimo de classes. De um modo geral o ISODATA é similar ao K-means. O algoritmo analisa o conjunto de exemplos de entrada e determina o número inicial de classes e seus respectivos centróides. Para cada classe

considera como pertencendo à ela os elementos que estiverem no polígono de Voronoi do centróide da classe. Após a definição dos elementos de cada classe seu centróide é atualizado e em seguida o processo avalia o conjunto de classes correntemente produzido. Classes grandes podem ser divididas com base em uma série de fatores (desvio padrão máximo para a classe, distância média do conjunto de amostras de classes ao centróide da classe, número de elementos de amostras na classe). Se a distância entre os centróides de um par de classes ficar abaixo de um *threshold*, as classes podem ser combinadas. Se o número de elementos de uma classe ficar abaixo de outro *threshold* a classe pode ser descartada e seus elementos são associados à outras classes. O processo acima é iterado até que não hajam mais alterações nos centróides dos clusters, ou então, seja alcançado o número máximo de iterações definidas para o método. Essa metodologia se constitui numa das mais utilizadas na área.

Self-Organization Map [32], é basicamente uma rede neural não supervisionada que tenta imitar a organização neuronal que ocorre cérebro humano, onde os neurônios vizinhos são ativados por estímulos similares. Ela é normalmente utilizada para encontrar relacionamentos entre um conjunto de variáveis que descrevem um problema. A característica mais marcante desse método é que ele faz uma projeção não linear a partir de um espaço de dimensão alta - muitas variáveis - em uma grade de neurônios regular de baixa dimensão (geralmente 2d). Uma vez que esse tipo de rede é distribuída em um plano (estrutura bidimensional), pode ser concluído que as projeções preservam as relações topológicas enquanto simultaneamente promove uma redução dimensional do espaço de representação.

Capítulo 3

Metodologia Empregada

3.1 Introdução - Shape Priors

Antes de descrever a metodologia empregada neste trabalho vamos fazer um resumo do que foi abordado nas seções anteriores sobre a utilização da forma na detecção/classificação de objetos. Dentro dessa descrição mais compacta vamos procurar contextualizar a metodologia escolhida em relação às abordagens padrão para tratar esses problemas para os quais a forma é, sem dúvida, a mais relevante propriedade, para muitas categorias de objetos[33][34]. Em vista disso a utilização de critérios para identificação dos objetos de uma classe em função de características extraídas de sua forma vem sendo largamente utilizada desde há muito tempo. Esses critérios são estabelecidos numa fase de aprendizado, quando exemplos de elementos da classe são tratados para gerar um conjunto de parâmetros tal que a análise de seus valores permita distinguir a classe de outras também possíveis. Além de ser gerada a partir de imagens reais, essa caracterização pode incorporar à própria definição da classe, elementos característicos que permitam distingui-la de outras. A definição torna-se, assim, dependente de todo o conjunto de classes em questão. Deve-se considerar que pode ser mais difícil caracterizar as classes, tomando cada classe separadamente e explorando especificidades comuns aos seus elementos.

Entre muitas alternativas para caracterizar formas num contexto bi-dimensional, a mais natural seria a de particionar seu traçado num conjunto de segmentos de forma simples, comumente segmentos de retas. Entretanto, se as figuras da classe são mais complexas e, em especial, se apresentam alguma diversidade pode ser difícil escolher um traçado que represente a classe. O problema principal é que segmentos que são significativos num elemento da classe podem não ter correspondente em outro, além do fato de limites de segmentos que são nítidos em um contorno poderem ser imprecisos em outros. Se os objetos são tri-dimensionais, temos adicionalmente a questão de escolher um conjunto de vistas suficiente para cobrir, ainda que de

modo aproximado, todas as alternativas topológicas para a projeção em perspectiva desses objetos. Para tornar mais efetiva a caracterização da classe dada por esses modelos, se acrescentam a eles descrições de “características notáveis” como ângulos acentuados, trechos mais estreitos, extremidades de traçados abertos, que são referidas genericamente como marcas (*landmarks*). Além disso, relações entre o tamanho de segmentos ou entre distâncias de uma marca a outra ou ao centro de gravidade procuram tornar o processo de identificação imune à variação da escala.

A terminologia *shape priors* se refere, exatamente aos exemplos de formas dos elementos de uma classe que são utilizados para construir a sua caracterização. Um conjunto considerável de trabalhos tem sido apresentado, em diferentes contextos, descrevendo formas de se aproveitar a informação proveniente dos *priors* em metodologias correntes para tratar o problema de classificação, sejam elas de natureza estatística, baseadas em teoria dos grafos, empregando contornos ativos - nesse caso efetuando-se conjuntamente a segmentação do elemento - ou de modo mais genérico, propondo a resolução de algum problema de minimização de energia.

A abordagem escolhida aqui emprega essencialmente PCA - análise de componentes principais - obtendo modos principais para uma dada representação dos elementos de uma classe. A representação que ela utiliza é que é menos corrente. Ela assume que todos os membros de uma classe podem ser obtidos a partir de uma figura média por um processo de deformação. Assumindo que a variância dentro da classe não é alta. Esse processo de deformação pode ser substituído por uma aproximação linear, especificamente, a obtida quando ele começa, isto é, quando se parte da figura média. Essa aproximação é determinada computando-se em cada ponto da figura média gradientes de deformação.

Na formalização dessa abordagem apresentada na seção que segue, o processo de deformação é expresso por um sistema de equações diferenciais e mecanismos estatísticos padrão são adaptados empregado para obter os modos principais dos gradientes, nos pontos da curva média, das deformações que levam essa curva nos elementos de uma classe. Essa abordagem foi introduzida por Guillaume Charpiat *et al.* em [1] o qual serviu de base para a realização deste trabalho.

3.2 Descrição da Metodologia Utilizada

A questão tratada especificamente em [1] é encontrar o contorno de um objeto utilizando-se de informações estatísticas, vindas de um conjunto de exemplos supostamente pertencendo a classe do que se está segmentando. Como ele emprega processos de evolução de um contorno ativo, vamos inicialmente distingui-lo da abordagem usualmente empregada nessa área. Nela, as informações disponíveis são todas atributos da imagem, não havendo nenhuma especificação prévia sobre a forma do

objeto a ser segmentado. Conseqüentemente, durante a evolução de um contorno ativo padrão para se segmentar um objeto, a única restrição quanto à forma é a expressa por um termo de regularidade que impõe suavidade ao seu contorno.

No trabalho em questão, além de informações desse tipo, se consideram também informações estatísticas relativas à forma do contorno, já traduzidas de forma que possam ser empregadas diretamente por algoritmos de contorno ativo padrão. O trabalho é apresentado no contexto de curvas planas, mas a metodologia pode ser facilmente estendida a bordas de superfícies de objetos 3D. A abordagem que ele apresenta foi projetada para ser invariante a transformações rígidas, não exigir um conjunto denso de amostras, não ser cara computacionalmente e para que variabilidade da forma dentro de uma classe seja a mais bem coberta possível, obtendo-se resultados satisfatórios mesmo quando o conjunto de representantes conhecido é muito pequeno.

3.3 Estatísticas da formas

3.3.1 Média Empírica

A primeira questão é como definir e calcular a média de um conjunto de formas de modo que elas sejam invariantes a transformações rígidas. Entenda-se por forma qualquer contorno suave, independente de parametrização e escala. Nesse contexto, podemos dizer que um movimento rígido - isto é, um movimento que não modifica a forma - R é definido como a combinação de uma translação, uma rotação e uma escala centrada no centro de massa da forma. Assim, R pode ser representado por quatro parâmetros reais: ângulo de rotação, fator de escala e as coordenadas do vetor translação.

Dadas, então, N formas $(\Gamma_1, \dots, \Gamma_N)$ e uma energia $E(\cdot, \cdot)$ que expressa distância entre duas formas quaisquer, define-se como média empírica dessas formas, qualquer forma que atinja um mínimo local da função μ definida por:

$$\mu : \Gamma \rightarrow \frac{1}{N} \sum_{i=1}^N \inf_{R_i} E^2(\Gamma, R_i(\Gamma_i)) \quad (3.1)$$

Onde para cada forma Γ_i , o ínfimo é tomado sobre todos os movimentos rígidos R_i com os quatro graus de liberdade indicados acima.

Para o cálculo da média empírica de N formas, inicializam-se todos os movimentos rígidos de modo que todas as formas depois de movidas por eles ($R_i(\Gamma_i)$) tenham o mesmo centro de massa e raio médio. A partir de uma forma inicial $\Gamma(0)$,

resolve-se, então, o seguinte sistema de equações diferenciais parciais.

$$\partial_t \Gamma(t) = -\nabla_{\Gamma} \mu(\Gamma, R_i(\Gamma_i), \dots, R_N(\Gamma_N)) \quad (3.2)$$

$$\partial_t R_i = -\nabla_{R_i} E^2(\Gamma, R_i(\Gamma_i)) \quad (3.3)$$

A derivada parcial $\nabla_{\Gamma} \mu$ da energia μ relativo à forma Γ define um campo de deformação que aplica a cada ponto p de Γ um movimento segundo a direção normal a Γ em p . As duas equações diferenciais parciais são resolvidas simultaneamente. A evolução de Γ é feita utilizando-se um método de *level set* enquanto os parâmetros dos movimentos rígidos evoluem aplicando-se a eles um método do tipo *descending gradient*. Note-se pela definição de média dada, que podem existir várias médias se μ possuir diversos mínimos locais. Entretanto, optando-se para E , a aproximação diferenciável da distância de Hausdorff dada em Charpiat *et al* [35] a média empírica parece ser única, não se conhecendo contra-exemplos que contrariem essa afirmativa.

3.3.2 Covariância Empírica

Definida uma média de um conjunto de formas pode-se pensar em obter uma estrutura similar à matriz de covariância para um conjunto de N formas. Isso é obtido tomando-se para cada forma Γ_i o campo de deformação normal - isto é, onde cada ponto p da curva Γ_i é tracionado na direção ortogonal à curva em p - definido da seguinte maneira:

Considere o espaço H das curvas simples (Γ) de comprimento finito. Nesse caso, considerando que $\forall p \in R^2$, $E(p, R_i(\Gamma_i))$ é dada pelo mínimo das distâncias de p a pontos de $R_i(\Gamma_i)$, então $F_i(\Gamma) = E^2(\Gamma, R_i(\Gamma_i)) = \int_{\Gamma} E^2(p, R_i(\Gamma_i)) dp$ também será finito $\forall i$. Assuma adicionalmente que $p \rightarrow E^2(p, R_i(\Gamma_i))$ é continuamente diferenciável. Seja, então, $\nabla_{\Gamma} F_i(\hat{\Gamma})$ o gradiente da função F_i computado em $\hat{\Gamma}$.

As formas em questão podem ser assumidas como imagens de funções de um mesmo parâmetro $s \in [0, S]$ no R^2 e identificando as curvas com as funções que as definem, podemos fazer $H \subseteq L_2([0, S])$ - o espaço das funções de $[0, S]$ no R^2 cujas funções coordenadas tem quadrado integrável. Nesse caso, a soma das integrais em $[0, S]$ dos quadrados das funções coordenadas pode ser tomada como a norma de uma função em $L_2([0, S])$. Uma direção σ em $L_2([0, S])$ será caracterizada por uma função nesse espaço que tenha norma 1. Derivadas unidirecionais de $F_i(\Gamma)$ definidas por: $\partial_{\sigma} F_i(\Gamma) = \lim_{\lambda \rightarrow 0} ((F_i(\Gamma + \lambda\sigma) - F_i(\Gamma)) / \lambda)$ irão então existir $\forall \sigma$ e $\forall i$.

Definimos $\nabla_{\Gamma} F_i(\Gamma)$, o gradiente da função F_i computado em Γ , como sendo o valor da derivada unidirecional $\partial_{\sigma^*} F_i(\Gamma)$ na direção σ^* onde ele é maior, multiplicado

por σ^* . Para identificar σ^* observe que ele pode ser obtido por

$$\lim_{\lambda \rightarrow 0} (\operatorname{argmax}(F_i(\Gamma + \lambda\sigma) \quad \text{s.a.} \quad \|\sigma\| = 1)) \quad (3.4)$$

Para caracterizar uma solução do problema de otimização acima vamos reescrevê-lo explicitando todos os elementos envolvidos. Assumindo, inicialmente, que tanto Γ como σ são diferenciáveis ele toma então a forma:

$$\begin{aligned} \max(\int_0^S E^2((x(s) + \lambda\sigma_x(s), y(s) + \lambda\sigma_y(s)), R_i(\Gamma_i)). \\ \sqrt{(x'(s) + \lambda\sigma'_x(s))^2 + (y'(s) + \lambda\sigma'_y(s))^2} ds) \quad (3.5) \\ \text{sujeito a } \int_0^S (\sigma_x(s)^2 + \sigma_y(s)^2). ds = 1 \end{aligned}$$

onde, $(x(s), y(s))$ e $(\sigma_x(s), \sigma_y(s))$ são as coordenadas de $\Gamma(s)$ e $\sigma(s)$ respectivamente. $f'(s)$ identifica a derivada de f em relação a s , onde f pode ser uma das quatro funções coordenadas acima. Definindo $v(s) = x(s) + \lambda\sigma_x(s)$ e $w(s) = y(s) + \lambda\sigma_y(s)$, o problema acima se torna:

$$\begin{aligned} \max(\int_0^S E^2((v(s), w(s)), R_i(\Gamma_i)). \sqrt{(v'(s)^2 + w'(s)^2)} ds) \quad (3.6) \\ \text{sujeito a } \int_0^S ((v(s) - x(s))^2 + (w(s) - y(s))^2). ds = \lambda^2 \end{aligned}$$

Uma condição de otimalidade para esse último problema é dada pelas equações de Euler associadas a função

$$(\int_0^S E^2((v(s), w(s)), R_i(\Gamma_i)). \sqrt{(v'(s)^2 + w'(s)^2)} + \beta\{(v(s) - x(s))^2 + (w(s) - y(s))^2\} - \lambda^2) ds \quad (3.7)$$

onde β é um real qualquer.

Assumindo que w e v são duplamente diferenciáveis e fazendo:

- 1) $P(s) = E^2((v(s), w(s)), R_i(\Gamma_i))$ e
- 2) $N(s) = P(s)(v''(s)w'(s) - v'(s)w''(s)) + (v'(s)^2 + w'(s)^2)(\partial P/\partial w(s)v'(s) - \partial P/\partial v(s)w'(s))$,

as equações de Euler relativas a função dada em 3.7 são dadas por:

$$\begin{aligned} w'(s).N(s) + \beta(v(s) - x(s)) = 0 \quad e \\ -v'(s).N(s) + \beta(w(s) - y(s)) = 0 \end{aligned} \quad (3.8)$$

Substituindo $(v(s) - x(s), w(s) - y(s))$ por $\lambda\sigma(s)$ temos: $(w'(s), -v'(s)) = (-\beta\lambda/N(s))\sigma(s)$

Como $(w'(s), -v'(s))$ caracteriza a direção normal a $(v(s), w(s))$ e $(-\beta\lambda/N(s))$ é um escalar, verificamos que se σ é solução do problema dado em 3.4 então $\sigma(s)$ é ortogonal a $(v(s), w(s)) \forall s$ em $[0, S]$.

Finalmente, como tanto $w'(s), -v'(s) \rightarrow ((y'(s), -x'(s)))$ como $\sigma(s) \rightarrow \sigma^*(s)$ quando $\lambda \rightarrow 0$ e o mapeamento que leva um ponto de uma curva numa direção normal a curva nesse ponto é semi-contínuo inferiormente, temos que: $\sigma^*(s)$ deve ser ortogonal a Γ em $\Gamma(s)$.

Se w e v não forem duas vezes diferenciáveis, eles serão limites de sequências (w_n) e (v_n) com essa propriedade o que permitirá estender a afirmação acima para essa condição mais geral.

Portanto, $\nabla_\Gamma F_i(\Gamma)$, que é um campo bidimensional definido em $[0, S]$ - ou seja, uma função de $[0, S] \rightarrow \mathfrak{R}^2$ - tem a direção determinada em todo s onde $\Gamma(s)$ seja diferenciável. Num desses pontos- p - ele poderá então ser caracterizado por um único valor, a sua norma, já que sua direção estará determinada - é a da normal a Γ em p . Se Γ é uma curva simples $(\Gamma^{-1})(p)$ está perfeitamente identificada e podemos associar p a $\alpha_i(p) = \langle \nabla_\Gamma F_i(\Gamma)(\Gamma^{-1})(p), n(p) \rangle$, onde $n(p)$ é o vetor unitário normal a Γ em p que aponta para fora da curva. Definimos assim, nos pontos onde Γ é diferenciável, um campo unidimensional α_i e supondo que Γ é diferenciável, exceto em um número finito de vértices, podemos estender a definição de α , também para esses vértices, atribuindo a eles um valor qualquer. Isso não vai alterar o valor da integral ao longo de Γ de qualquer função dos campos α_i , $i = 1, \dots, N$. Desse modo, podemos considerar α_i definido para toda a curva Γ .

Observamos que $\nabla_\Gamma F_i(\Gamma)$ é a melhor deformação elementar que se deve aplicar a Γ para torná-la mais próxima de $R_i(\Gamma_i)$. O campo α_i indica a intensidade e o sentido dessa deformação em cada ponto p onde Γ é diferenciável.

Estes campos $\alpha_1, \alpha_2, \alpha_3 \dots \alpha_N$ pertencem ao mesmo espaço, o espaço dual à curva média, aquele constituído de funções dos pontos de Γ em \mathfrak{R} . Alternativamente, funções que levam cada ponto p de Γ num vetor normal a Γ em p cuja norma é dada por $|\alpha_i(p)|$, $i = 1, \dots, N$. A correlação entre dois quaisquer desses campos α_i e α_j é dada por:

$$\langle \alpha_i | \alpha_j \rangle_{L^2} = \int_{\hat{\Gamma}} \alpha_i(x) \alpha_j(x) dx \quad (3.9)$$

Tendo essas correlações, podemos realizar uma análise das componentes principais (PCA) para conjunto de campos α_i , $i = 1, \dots, N$ e obter novos campos de deformação instantânea β_k - que serão auto-vetores unitários da matriz de correlação- associados a desvios-padrão σ_k - o auto valor da matriz de correlação associado a β_k . Os auto vetores β_k , $k = 1, \dots, N$ formam uma base ortogonal do espaço de deformações gerado pelos α_i . Ortogonalidade no caso significa dizer que esses novos campos são

totalmente de-correlacionados entre si. Dado que os β_k são auto-vetores da matriz de correlação dos α_i se aplica a eles a nomenclatura **modos característicos** do espaço de deformações gerado por esses campos. Conforme mostrado na seqüência, os campos β_k , são muito sensíveis e expressam a variabilidade das formas das amostras do conjunto dado previamente.

Substituímos assim a correlação entre as curvas Γ_i pela correlação entre as deformações elementares que mais aproximam a curva Γ das curvas Γ_i . Obter uma correlação entre duas curvas é complicado por que precisamos estabelecer uma correspondência entre os pontos de uma e de outra cujos dados/coordenadas serão multiplicados. O resultado torna-se então uma função da correspondência escolhida. No caso das deformações elementares não temos esse problema - multiplicamos os valores dos dois campos de deformação em cada ponto d e Γ .

Suponha, finalmente, que as curvas Γ_i são relativamente próximas e seja $\hat{\Gamma}$ a forma média obtida a partir das Γ_i da forma indicada na seção anterior. Nesse caso, cada curva Γ_i estará razoavelmente representada pela curva χ_i , obtida aplicando-se à curva média $\hat{\Gamma}$ uma deformação $\lambda\alpha_i$, para um dado $\lambda > 0$ pequeno. $\lambda^2 < \alpha_i, \alpha_j >$ é uma formulação robusta para a correlação entre χ_i e χ_j que nas condições dadas acima aproximam bem Γ_i e Γ_j , respectivamente. Esse é o contexto em que a metodologia apresentada aqui faz perfeito sentido.

3.4 Shape Priors

Podem ser definidos diferentes *Shape Priors* cada um deles baseado numa forma de se expressar a distância de uma forma a um conjunto de outras ($\Gamma_1, \Gamma_2, \dots, \Gamma_N$).

3.4.1 Contexto

Considere, inicialmente, o problema de segmentação de imagens. Seja, então, C a curva de evolução que se deseja ajustar ao contorno do objeto numa nova imagem dada I , sabendo que a curva C representa um objeto de uma classe Λ . Qualquer problema de minimização de energia $-\min_x E(x)$ - pode ser transformado num de maximização de probabilidade - $\max (1/\pi) e^{-E(x)^2}$, por exemplo - sendo, portanto as duas abordagens essencialmente a mesma. Assim o problema de se conseguir o melhor ajuste para C dentro da imagem I , que dentro do contexto clássico dos contornos ativos, pode ser pensado como um problema de se minimizar a soma das energias interna e de campo ao longo da curva, pode ser transformado num problema de maximizar em C a probabilidade $P(C|I)$. Observe que no referido contexto clássico não temos informação prévia acerca da forma do contorno e por isso a influência da forma em $P(C|I)$ é determinada medindo-se de alguma maneira

a sua suavidade.

Em nosso caso, temos adicionalmente que a curva que procuramos pertence à classe Λ . Um trabalho estatístico feito com base em exemplos conhecidos dessa classe nos permite gerar uma probabilidade $P(C|\Lambda)$ de que a curva C pertença a Λ . Agora temos determinadas não só a imagem I mas também a classe Λ e é mais correto escrever a probabilidade que devemos encontrar como $P(C|I, \Lambda)$.

Por outro lado, a participação da imagem é obtida a partir de alguma medida de quanto a curva C pode ser um contorno nessa imagem - determinada, em geral, pela soma de alguma norma dos gradientes da imagem computados nos pixels atravessados pela curva. Eventualmente pode-se mensurar, via algum procedimento estatístico, a variação dos atributos nos pixels da região delimitada pela curva, quando se sabe que essa região deve apresentar uma textura uniforme ou já previamente conhecida. Essa probabilidade pode ser pensada como uma probabilidade a priori: Dada uma curva C , ela indica a probabilidade da imagem I conter uma representação de um objeto delimitado pela curva C . Como essa probabilidade independe da classe Λ podemos grafá-la simplesmente $P(I|C)$.

A probabilidade que queremos computar é a probabilidade a posteriori $P(C|I, \Lambda)$, que em nosso caso indica dada a imagem I e a classe Λ , qual a probabilidade da curva C ser um contorno em I e sua forma pertencer à classe Λ .

Pelo Teorema de Bayes a probabilidade a posteriori $P(C|I, \Lambda)$ pode ser obtida multiplicando-se a probabilidade a priori $P(I|C) = P(I|C, \Lambda)$ pela razão $P(C, \Lambda)/P(I, \Lambda)$ -Ver equação 3.10. Isso é verdade porque Λ restringe todas as probabilidades acima. O denominador dessa última razão é o mesmo, não importa qual seja a curva C , e portanto podemos descartá-lo sem alterar a solução ótima do problema de maximização. Assumindo então que $P(I|C)$ será calculado de uma forma clássica como as especificadas acima, resta, então, propor alternativas para o cômputo de $P(C, \Lambda)$.

$$P(C|I, \Lambda) = P(I|C) \frac{P(C, \Lambda)}{P(I, \Lambda)} \cong P(I|C)P(C|\Lambda) \quad (3.10)$$

Na seqüência $P(C, \Lambda)$ vai ser uma função assumindo valores positivos sem a propriedade de ter integral unitária requerida por uma probabilidade. Transformá-la numa probabilidade, entretanto, é mera questão de dividi-la pelo valor da integral. Mas isso não precisa ser feito, dado que, fazendo ou não essa divisão, a solução ótima do problema $\max_C P(C, \Lambda)$ é a mesma.

O problema de classificar o objeto representado numa imagem I pode ser pensado como um problema de maximização dupla, da forma $\max_\Lambda(\max_C P(C, \Lambda))$. Dessa maneira ele requer a resolução do problema de segmentação descrito acima.

3.4.2 Probabilidade da forma C pertencer à classe Λ

Dada a distância ou uma energia E que mensura a diferença entre duas formas, uma nova distância entre uma forma C e o conjunto de formas de uma classe Λ pode ser definida como, ou mais apropriadamente, como:

$$\sum_{i=1} \inf_{R_i} E^2(C, R_i(\Gamma_i)) \quad (3.11)$$

Mas o mínimo global desta energia, conforme já foi descrito é a forma média $\hat{\Gamma}$, de modo que essa energia não é realmente mais interessante do que $E(C, \hat{\Gamma})$. Em particular, as variações das formas em relação a $\hat{\Gamma}$ não são consideradas. Se elas variam muito ou pouco ou de que forma, nada disso altera o valor da energia de uma curva C , e em consequência não influem no cômputo de $P(C, \Lambda)$. Por isso descartamos o uso dessa formulação.

Modos Principais

Uma alternativa que leva em conta as variações em relação à curva média $\hat{\Gamma}$ é expressá-las por intermédio de deformações locais -isto é, aplicadas à própria curva média no começo de um processo de deformação que transforma essa curva em um membro qualquer da classe Λ - conforme foi descrito na Seção 3.3.2. Cada auto-valor σ_k da matriz de covariância entre os campos de deformação relativos as curvas membros da classe (Γ_i) , representa o desvio padrão relativo a uma nova variável, definida como uma combinação das variáveis originais canônicas ponderada pelas coordenadas dos auto-vetores unitários β_k , que caracterizam os modos da classe Λ . Se a classe tiver muitos elementos, o número de modos será grande e teremos que fazer restrições, considerando explicitamente apenas aqueles associados aos autovalores maiores.

Seja k , o número de modos que vamos considerar explicitamente. Para cada um deles seja β_k , o auto-vetor associado e dada uma curva C computamos sua projeção sobre a reta gerada por β_k , que devido a ortonormalidade dos β_k é obtida pelo produto interno $\langle C, \beta_k \rangle = \int_{\hat{\Gamma}} \alpha(C) \alpha(\beta_k)$, onde $\alpha(C)$ e $\alpha(\beta_k)$ são os campos de deformação iniciais aplicados a $\hat{\Gamma}$ em processos de transformação que as levam em C e β_k , respectivamente. Observe que, computando $\langle C, \beta_k \rangle$ dessa maneira nos livramos de ter de encontrar uma parametrização comum para C e β_k . Seja então C_K , a projeção ortogonal de C sobre o sub-espaco gerado pelos K autovetores associados aos maiores autovalores σ_k . Pela ortonormalidade dos β_k temos

$$C_K = \sum_{k=1}^K \langle C, \beta_k \rangle \cdot \beta_k \quad (3.12)$$

A contribuição dos demais modos é representada pelo vetor $Rem(C) = C - C_K$. No entanto, a questão de determinar o número de modos de interesse não é fundamental, pois a importância de cada modalidade está relacionada a seu desvio-padrão e, conseqüentemente, a consideração de alguns modos extra com baixo desvio-padrão não irá alterar significativamente a distribuição.

Essa decomposição supõe implicitamente que combinações lineares de campos de deformações iniciais α_i seja também um campo de deformação inicial válido - Essa é a expectativa de qualquer abordagem que emprega a decomposição em componentes principais. Vamos supor, adicionalmente que a distribuição da função de pertinência à classe Λ é gaussiana na reta gerada por qualquer β_k . Nesse caso:

1) A covariância entre as projeções de curvas da classe Λ relativas a β_k e β_j ($k \neq j$) são nulas.

2) A variância das projeções de curvas da classe Λ relativas a β_k é dada por σ_k^2 podemos obter uma expressão para $P(C, \Lambda)$ dada por

$$P(C, \Lambda) = P(\alpha(C)) = \prod_k e^{-\frac{\langle \beta_k | \alpha(C) \rangle_{L^2}^2}{2\sigma_k^2}} \cdot e^{-\frac{\|Rem.(\alpha(C))\|_2^2}{2\sigma_{noise}^2}} \quad (3.13)$$

O parâmetro σ_{noise} representa o desvio padrão associado aos modos não principais - que pode ser considerado um ruído - e pode ser escolhido, por exemplo, 0,01 vezes o menor autovalor (σ_K) considerado explicitamente.

Note que a distância de C à curva média - ($d(C, C_m)$) - que dá origem a essa distribuição de probabilidades - via a expressão, $P(C, \Lambda) = e^{-d(C, C_m)^2}$ - é exatamente:

$$\sqrt{\sum_k \frac{1}{\sigma_k^2} \langle \beta_k | \alpha(C) \rangle^2 + \frac{1}{\sigma_{noise}^2} \|Rem.(\alpha(C))\|_2^2} \quad (3.14)$$

Observe que no espaço de deformações α os level sets desta distância são elipsoidais. Esta distância é uma aproximação da distância de Mahalanobis, seria a obtida considerando-se explicitamente todos os auto-vetores expressa por $\sqrt{\sum_{k=1}^n \frac{1}{\sigma_k^2} \langle \beta_k | \alpha(C) \rangle^2}$.

Histogramas dos modos principais

Na seção anterior empregamos a hipótese de que a distribuição de $P(\lambda\beta_k, \Lambda)$ é Gaussiana em $\lambda \in \Re$ com média 0, o que não é necessariamente verdade. Além disso, os modos principais podem ser computados com ICA (análise de componentes independentes)[9] ao invés de PCA. Nessas condições podemos obter para cada modo principal k , o histograma normalizado h_k das projeções dos membros da classe Λ , observados no subespaço dos múltiplos de β_k . Os histogramas obtidos podem necessitar de alguma suavização, em especial, se a densidade da distribuição

for baixa.

Seja S , o mecanismo que efetua essa suavização. Então, a probabilidade $P(C, \Lambda)$ será expressa por:

$$P(C, \Lambda) = P(\alpha(C)) = \prod_k S(h_k(\langle \beta_k | \alpha(C) \rangle)) \quad (3.15)$$

Observe que, com exceção da primeira possibilidade, $\alpha(C)$ aparece na expressão da probabilidade para maximizar, o que implica que a derivada de $\alpha(C)$ com relação a C terá de ser computada: Essa derivada, dada a definição de $\alpha(C)$, é expressa por:

$$\nabla_C \alpha(C) = -\nabla_C \nabla_{\hat{\Gamma}} E^2(\hat{\Gamma}, R(C)) \quad (3.16)$$

Assim, é preciso calcular a derivada cruzada de segunda ordem para $E(\partial^2 E / \partial \hat{\Gamma} \partial C)$. Tendo em vista a complexibilidade desse cômputo, devemos observar que os cálculos poderiam ser, algumas vezes, muito mais simples se a energia E fosse baseada não em duas curvas, mas nas funções de distância com sinal de cada uma delas como é feito em [36], isto é, se considerarmos que o verdadeiro objeto de interesse é uma função definida no espaço R_2 e não só a curva que é o seu level set zero dessa função. Isso, independentemente de se tratar de uma função de distância ou não. O framework seria então semelhante ao de Leventon et al [37], que consiste na aplicação do PCA para as funções distância com sinal. Entretanto, segundo [1] tal abordagem é questionável, já que uma combinação linear de funções de distância com sinal geralmente não é uma função distância com sinal o que, por vezes, pode dar origem a um nível zero realmente inesperado. Por isso neste trabalho se recomenda manter a definição de E dada acima, apesar do eventual maior custo computacional.

3.5 Aplicação a um conjunto de formas

Para mostrar a força da abordagem delineada acima, em [1] se considera um pequeno conjunto de quatro figuras obtidas aplicando-se a um mesmo retângulo - que será a curva média da classe assim gerada - dois tipos de deformações aplicadas em dois sentidos - isto é, para fora e para dentro (figura 3.1). Para este exemplo particular, não há necessidade de se transformar rigidamente nenhuma das formas geradas, para calcular os campos de deformação. Elas já estão posicionadas de forma ótima em relação a curva média. Considere agora o *prior* onde se assume que a distribuição das projeções nos modos principais é Gaussiana e que pretendemos aplicar a metodologia à segmentação de uma nova forma obtida combinando duas deformações a uma terceira, nova. Neste caso o resultado será insatisfatório já que a nova deformação introduzida não poderá ser considerada. O procedimento proposto anteriormente

produz ao final uma projeção da curva sobre o subespaço gerado pelas deformações usadas para gerar os elementos da classe.

Um ponto interessante desse método é a característica de que não é preciso escolher os autovetores que serão considerados relevantes comparando-os com algum *threshold*. Pode-se definir, a priori, um número maior do que seria suficiente, dado que a importância de cada um deles é, naturalmente, descrito por seu desvio padrão associado. Para acentuar a influência dos principais modos característicos da deformação na solução obtida, o desvio padrão associado ao ruído foi escolhido para ser igual a 0,005 vezes o desvio padrão associado ao maior modo principal. Isso quer dizer que um campo de deformação “ruído” com componente nulo em cada modo relevante gasta uma energia 200 vezes maior do que a dispendida por um campo da mesma norma mas colinear com o primeiro modo. Para ser coerente, se existem modos principais com autovalores menores do que aquele associado ao ruído, então eles têm que ser esquecidos e considerados também como ruído. No caso da figura 3.1 os dois primeiros modos tem quase o mesmo autovalor e os outros dois tinham autovalores cerca de cem vezes menores. Na verdade, as segmentações com todos os modos ou apenas os dois primeiros foram as mesmas. Em problemas de segmentação de imagens, um *shape prior* é conjugado com medidas relacionadas a textura para compor uma função objetivo. Independentemente do valor do peso do *shape prior*, desde que ele seja suficientemente alto, o que ele determina no comportamento qualitativo da evolução do contorno ativo é que esse contorno seja “projetado” no subespaço das combinações lineares dos modos principais dos campos de deformação local $\alpha(C)$ aplicados à curva média para obter a forma da classe Λ . Ao longo do processo de otimização, os componentes nos modos principais (e a parte restante dada pelo ruído) serão progressivamente reduzidos de acordo com o seu desvio padrão. Como a distribuição que aqui é proposta é Gaussiana, qualquer aumento do peso do *shape prior* fará com que o resultado se aproxime da forma média. No entanto, no caso dos histogramas dos componentes, como proposto na Seção 3.4.2, este fenômeno pode não acontecer uma vez que a forma com a maior probabilidade a priori pode não ser média.

3.6 Registro Rígido

Uma série de 12 imagens de estrelas do mar (figura 3.2) foi segmentada à mão. Este passo poderia ser automatizado no caso geral, se os exemplos do conjunto da amostra fossem escolhidos de modo que as imagens fossem de fácil segmentação para os algoritmos usuais. A curva média do conjunto de estrela foi calculada com o framework invariante a transformações rígidas proposto na seção 3.2. A média e o seus primeiros modos principais são exibidos na figura 3.3 com amplitude

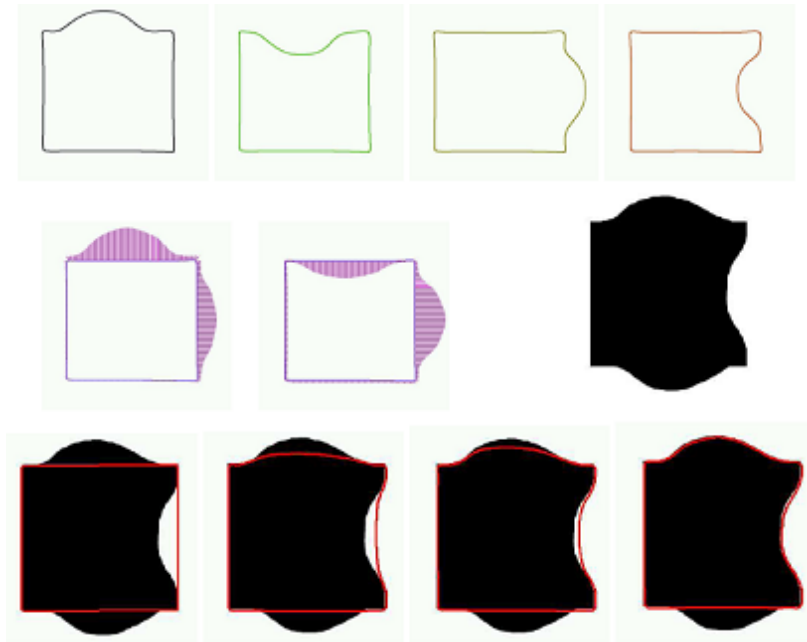


Figura 3.1: Um pequeno conjunto de quatro retângulos semelhante com dois tipos de deformações : Linha de cima: as quatro formas que compõem a amostra. Meio, à esquerda: a média destas formas (em azul), com os dois primeiros modos principais (em roxo), sucessivamente. Meio, a direita: a nova imagem a ser segmentada, construída aproximadamente como uma nova combinação das deformações observadas previamente adicionando uma nova deformação não observada. Linha de baixo: segmentação com o conhecimento apenas da média, modos principais e autovalores, assumindo uma distribuição Guassiana que, obviamente, não é satisfeita (à esquerda: inicialização; Meio: algumas etapas da evolução; direita: resultado de convergência). (Fonte: Charpiat *et al.*[1])

proporcional ao seus autovalores.

Consideraremos o *shape prior* rígido em uma nova tarefa de segmentação de imagem, onde C é a forma atual em evolução e R é uma transformação rígida. A forma C e os parâmetros de R são estimados simultaneamente dentro de um framework similar ao anterior.

Existem muitas abordagens para a segmentação da imagem na literatura de visão computacional, por exemplo, contornos ativos geodésicos [24] ou histogramas de região [38] [39].

Foi escolhido aqui um critério de histograma de intensidade da região. Mais precisamente, como uma imagem tem “naturalmente” 256 níveis de cinza, os histogramas de intensidade do interior h_i e o de fora h_0 do contorno C são funções reais com valor de um inteiro que pode ter 256 valores possíveis. Eles associam a qualquer nível de cinza o número desses pixels coloridos na região correspondente. Supõe-se que os dois histogramas são relativamente homogêneos e, muito diferentes um do outro como possível. O critério de segmentação para minimizar é uma soma



Figura 3.2: Alguns exemplos do conjunto de aprendizagem da estrela. (Fonte: Charpiat *et al.*[1])

ponderada do comprimento $|C|$ do contorno C mais a correlação entre uma versão ligeiramente suavizada das distribuições de probabilidade de intensidade das duas regiões e (onde $|I|$ é a área da região I):

$$\frac{1}{256} \sum_{a \in \llbracket 0, 255 \rrbracket} (G_\sigma * p_I)(a)(G_\sigma * P_O)(a) + |C| \quad (3.17)$$

onde $G_\sigma *$ é a suavização gaussiana com o parâmetro σ .

O resultado da conjugação deste critério com a *shape prior* rígido é mostrado na figura 3.4 : a localização da estrela do mar é encontrada, mas é claro que a variabilidade da forma do conjunto da amostra não foi tida em conta. Por conseguinte, cada resultado da segmentação é exatamente a média, sem qualquer deformação não rígida (se o peso da forma prévia for enorme), ou ele é a média mais uma pequena deformação qualquer (sem qualquer priors sobre esta deformação). Este algoritmo é simplista uma vez que o *prior* é uma forma fixa (até transformações rígidas), mas ajuda a encontrar um objeto não muito variável, bem como um objeto sob oclusão (comparar com a figura 3.5).

3.7 O Processo de Segmentação

Se os atributos do objeto a ser segmentado forem suficientemente distintos dos atributos do fundo, o produto dos histogramas do interior e exterior do contorno ativo pode ser utilizado para fazê-lo evoluir adequadamente até o contorno do segmento. Para isso, é preciso assumir que o contorno inicial envolve a maior parte do segmento desejado. Além disso, como o cômputo do histograma é realizado se fazendo contagens dos pixels que possuem o mesmo atributo é preciso identificar aqueles que pertencem ao interior e exterior do contorno vigente a cada nova interação. Isso, entretanto, pode se realizado computando-se os histogramas das regiões separadas pelo contorno inicial, depois atualizá-los determinando suas variações numa região em torno da borda. Em relação a isso existem dois aspectos a considerar.

- 1) Se os pixels do contorno ativo forem ligados por arestas de tamanho aleatório,

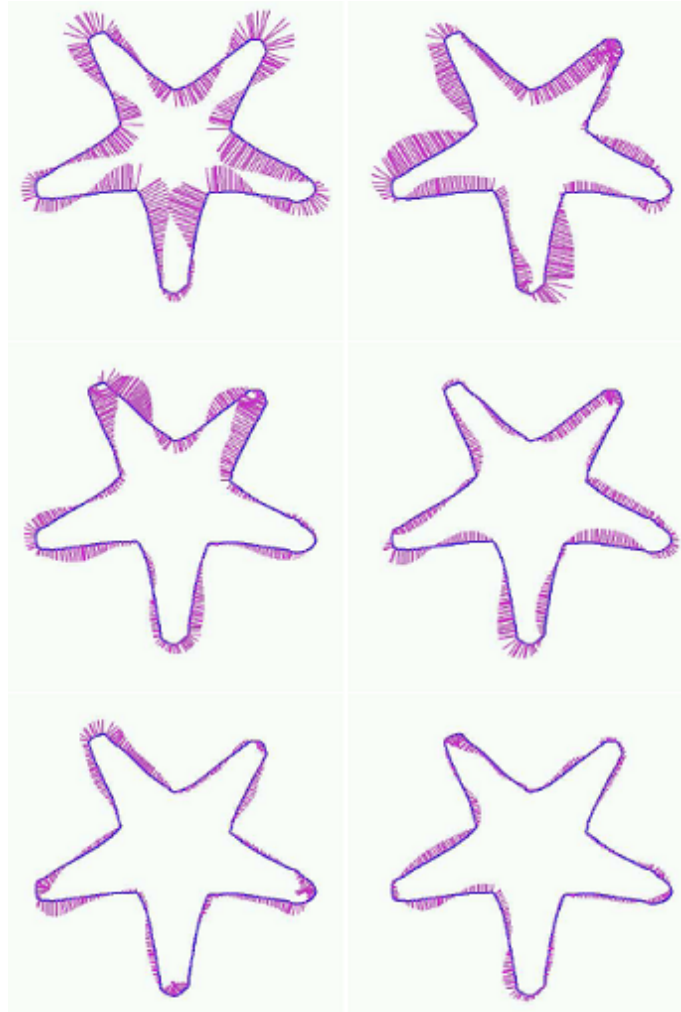


Figura 3.3: A média do conjunto de estrela, com seus seis primeiros modos principais. (Fonte: Charpiat *et al.*[1])

para avaliar alteração determinada pelo movimento de um snaxel seria preciso considerar toda região delimitada pelo posicionamento das arestas adjacentes ao snaxel, antes e depois do movimento (ver figura 3.6). Esse procedimento fica, naturalmente, simplificado se snaxels consecutivos ocuparem pixels vizinhos, o que é equivalente a dizer: se uma versão digital do contorno em evolução fosse computada a cada passo dela. Esse procedimento fica simplificado se os pixels ocuparem pixels vizinhos.

2) Argumentos similares justificam a opção de se realizar deslocamentos de snaxels de no máximo um pixel. Caso contrário avaliar a variação dos histogramas interno e externo vai requerer também que toda uma região em torno dos snaxels seja analisada. Assim, para simplificar o procedimento trabalhamos com versões “raster” dos contornos e efetuamos movimentos elementares de um pixel a cada iteração. Esclarecemos que nosso foco é o resultado da segmentação e não a performance com que ela é obtida. Além disso, vamos supor que cada snaxel se movimenta apenas na direção normal ao contorno na posição que ele ocupa. Dentro do contexto raster

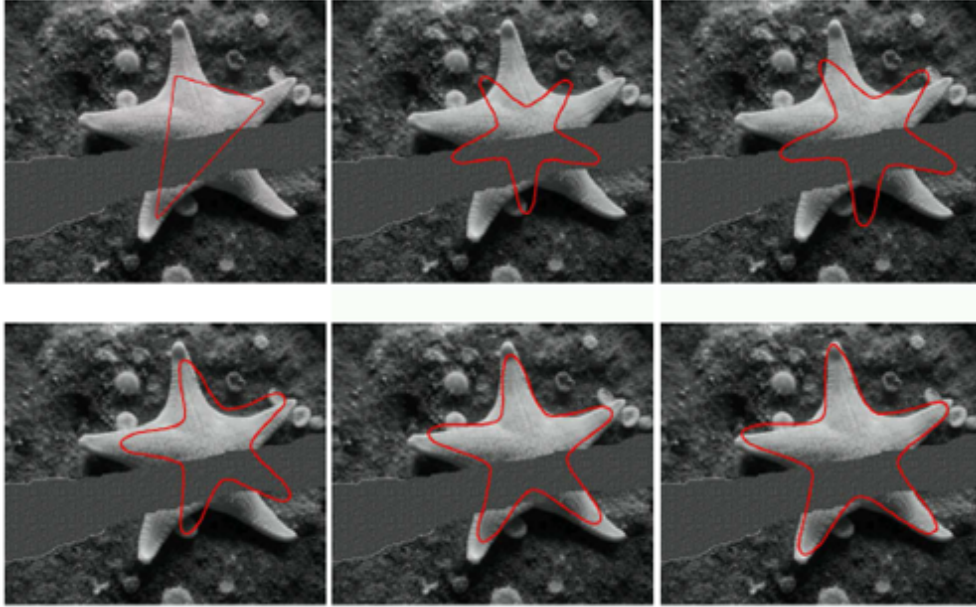


Figura 3.4: Esquerda sup: qualquer inicialização razoável para o critério histograma de intensidade regional. Uma parte da imagem foi apagada a fim de aumentar a dificuldade. Meio sup: A mudança automática para a média no mesmo local, com tamanho semelhante. Restante: algumas etapas do processo de segmentação com o conhecimento da forma média. Este critério rígido encontra a localização da estrela, mas não possui informações sobre como adaptar a forma final. Com efeito, para um menor peso dos anteriores. (Fonte: Charpiat *et al.*[1])

em que estamos trabalhando, isso significa, de fato, dizer que há apenas três opções para o deslocamento de um snaxel do contorno ativo numa iteração:

I) ele mantém sua posição.

II) ele passa a ocupar a posição de um vizinho, interior ao contorno na interação atual. Esse vizinho pertence ao mesmo octante que o vetor normal que aponta para dentro. Observamos que consideramos octantes centrados nas direções principais. Assim, o ângulo (α) que um vetor no octante k faz com o eixo horizontal atende a expressão

$$k\frac{\pi}{4} - \frac{\pi}{8} < \alpha \leq k\frac{\pi}{4} + \frac{\pi}{8} \quad (3.18)$$

III) situação idêntica anterior só que em relação a parte externa do contorno. A opção de movimento que ele irá tomar será a que otimiza uma função objetivo obtida a partir dos histogramas da parte interna e externa do contorno. Essa função objetivo, idealmente seria:

$$F = \sum_{i \in \{\text{atributos}\}} H_I(i) \cdot H_E(i) \quad (3.19)$$

Quando o objeto e o fundo têm cores constantes distintas, essa função objetivo será

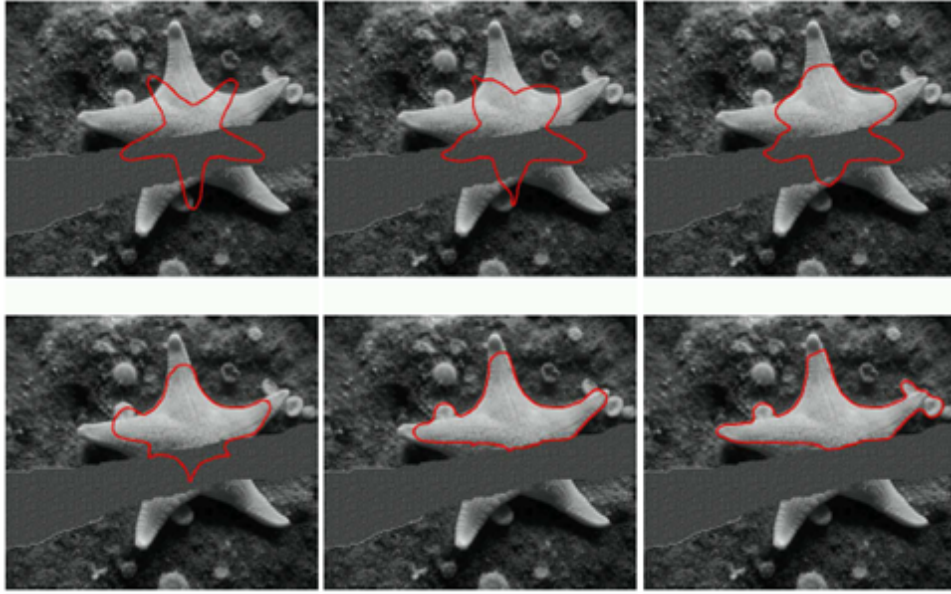


Figura 3.5: Linha superior: a mesma inicialização que a anterior. Restante: Alguns passos do processo de segmentação sem nenhum *shape prior*, para o mesmo critério de região. O resultado não tem a forma global da estrela e inclui as pequenas bolas brancas. (Fonte: Charpiat *et al.*[1])

nula apenas se o contorno ativo coincidir com a borda do objeto. A premissa para que ($F = 0$) caracterize a identificação da borda do objeto é assumir que os valores dos pixels dentro do objeto sejam completamente distintos dos valores dos pixels fora dele e que isso não aconteça para as regiões delimitadas por nenhuma curva fechada que não seja a sua borda. Observe que essa condição não é cumprida quando um objeto circular claro é envolvido por uma coroa circular escura, sendo o restante da imagem mais escura ainda. Nesse caso, tanto o contorno do objeto como o da coroa satisfariam $F = 0$ e dependendo da posição inicial do contorno ativo, o procedimento iterativo orientado para a minimização de F que descreveremos na próxima seção, convergiria para um ou outro caso, aquele que estivesse mais perto.

Esclarecemos que a proximidade nesse caso, precisaria ser definida com precisão, o que não faremos aqui. A grosso modo, mais perto significa aquela opção que puder ser atingida mudando o menor número de pixels de lado, do interior para o exterior da curva e vice-versa.

Mas a expectativa é que se partimos de uma projeção sobre o subespaço de modos principais da classe a que pertence o objeto, já devidamente ajustada em relação a escala, orientação e posição, então já devemos estar mais próximos da verdadeira borda do que de qualquer outra solução alternativa do problema de otimização acima. A premissa enunciada acima se torna: Para que ao atingir um mínimo local da função F , tenhamos chegado a borda do objeto, é preciso que não haja outro mais próximo da configuração inicial. Passando-se um filtro gaussiano nos histogra-

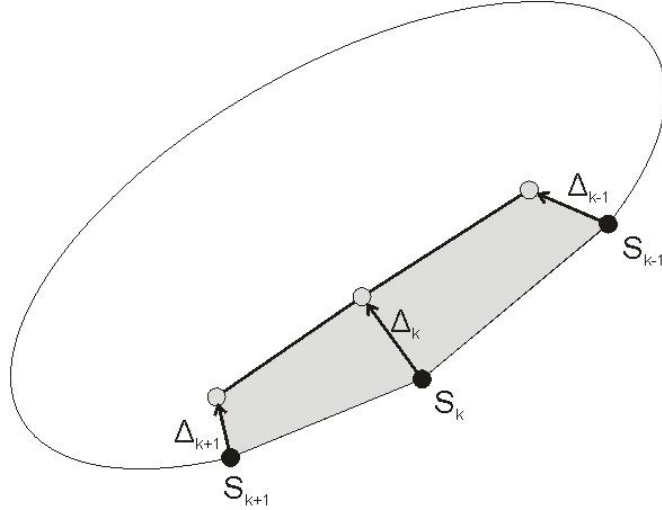


Figura 3.6: Região a ser considerada para se computar a variação do histograma de parte interna e externa da curva quando o snaxel S_k e seus vizinhos são movimentados numa iteração.

mas antes de multiplicá-los, valor a valor, essa última pressuposição se torna mais factível. Ela evita que artefatos relativamente homogêneos, normalmente pequenos pertencentes ao fundo sejam envolvidos pelo contorno final aparecendo, então como se fazendo parte do objeto. Imagine um objeto onde o atributo vale 20, cercado por um fundo onde ele vale 5 exceto em pixels esparsos uniformemente espalhados onde ele vale 6. Usando a F definida originalmente, se o contorno inicial envolver alguns desses pixels, eles poderão permanecer dentro do contorno, dado que esse fato não altera o valor de F . Já se a sua presença na parte de dentro fizer com que o valor do histograma dessa parte computado para 6, seja não nulo-o que aconteceria pela ação do filtro Gaussiano-, como ele certamente é não nulo para a parte externa, o valor da função objetivo vai aumentar. Assim, na solução ótima, esses pontos devem ficar na parte de fora do contorno. Além disso, onde as bordas são menos definidas, a filtragem do histograma torna os contornos obtidos mais suaves, em vista disso tudo, fizemos nossa função objetivo se tornar:

$$F' = \frac{1}{\#(\text{atributos})} \cdot \sum_{i \in \{\text{atributos}\}} \frac{(G_\sigma \circ H_I)(i)}{\#(I)} \cdot \frac{(G_\sigma \circ H_E)(i)}{\#(E)} \quad (3.20)$$

Na expressão acima G_σ , é a função Gaussiana de média 0 e variância σ^2 , já devidamente discretizada. Usamos discretizações para um raio de 16 ou 32 e um desvio padrão, usualmente da ordem de 2/3 do raio. Os termos $\#(I)$ e $\#(E)$ representam o número de pixels interiores e exteriores ao contorno, respectivamente. A divisão dos histogramas por eles, indica o percentual dos pixels de uma dada região que tem um determinado valor e ao referir esses valores ao tamanho da região evita que o

contorno tenda a se contrair demais. Além disso, junto com a divisão pelo cardinal do atributo, essas divisões promovem um processo de normalização que faz com que o valor de F' esteja entre 0 e 1

O exemplo da figura 3.7 serve para esclarecer o seguinte ponto. É claro que poderíamos remover os whiskers - trechos da curva em que ambos os lados são exteriores ou interiores - mediante o acréscimo à função F de um termo proporcional ao tamanho da curva gerada, que poderia, agora, ter mais de uma componente. Ocorre que não queremos nos livrar dos whiskers. Isso porque em situações de oclusão, a detecção deles pode indicar como componentes conexas do objeto encontradas na figura devem ser agregadas. Feita essa contextualização da abordagem escolhida, passamos a descrever como é realizada a evolução do contorno ativo em cada iteração:

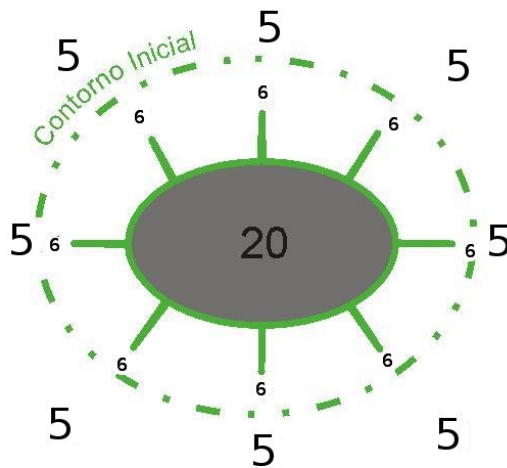


Figura 3.7: Resultado hipotético de uma evolução, cujo o contorno inicial considerou uma faixa de pixels (valor 6) não pertencentes ao objeto o que gerou whiskers.

1. Para cada pixel p cortado pelo contorno vigente computamos a normal- n_p a esse contorno. Obtemos essa normal, como a direção do gradiente no pixel da transformada da distância com sinal do contorno vigente. Para isso, computamos apenas uma versão “narrow-band” dessa transformada. Essa transformada deve ser sinal diferente nos pixels interiores e exteriores ao contorno.

2. Identificamos os octantes o_i e o_e cortados pela direção n_p . Sejam, então, p_i e p_e os pixels vizinhos a p associados a esses octantes. Optamos por considerar os pixels de borda como externos ao objeto, de forma que todos os vizinhos de p sejam interiores ou exteriores.

3. Suponha, agora, que p_i é interior. Nesse caso se o contorno se movimentar de p para p_i então, p_i que era interior passou a ser exterior o que vai ocasionar nos histogramas filtrados $H'_I = G_\sigma \bullet H_I$ e $H'_E = G_\sigma \bullet H_E$ relativos as partes interna e

externa, respectivamente as variações $d_1H'_I$ e $d_1H'_E$ dadas por:

$$d_1H'_I(n) = \begin{cases} -G_\sigma(n - n(p_i)), & n(p_i) - R \leq n \leq n(p_i) + R \\ 0, & \text{em caso contrario.} \end{cases} \quad (3.21)$$

$$d_1H'_E(n) = \begin{cases} +G_\sigma(n - n(p_i)), & n(p_i) - R \leq n \leq n(p_i) + R \\ 0, & \text{em caso contrario.} \end{cases} \quad (3.22)$$

onde, n é um valor qualquer do atributo, $n(p_i)$, o valor do pixel p_i e R o raio de definição da Gaussiana.

4. Da mesma forma, supondo que p_e é exterior então se assume que ao se mover o contorno de p para p_e , p passa a ser interior determinando nos histogramas as variações $d_2H'_I$ e $d_2H'_E$ expressas por:

$$d_2H'_I(n) = \begin{cases} +G_\sigma(n - n(p)), & n(p) - R \leq n \leq n(p) + R \\ 0, & \text{em caso contrario.} \end{cases} \quad (3.23)$$

$$d_2H'_E(n) = \begin{cases} -G_\sigma(n - n(p)), & n(p) - R \leq n \leq n(p) + R \\ 0, & \text{em caso contrario.} \end{cases} \quad (3.24)$$

onde $n(p)$, o valor do pixel p .

5. Essas variações, levando em conta a definição de F' , determinam as variações Δ_I e Δ_E indicadas pelas expressões abaixo. Nessas expressões W_{p_i} e W_p , representam as vizinhanças de raio R de n_{p_i} e n_p , respectivamente.

$$\Delta_I = \frac{1}{\#(\text{atributos})} \left\{ \sum_{n \notin W_{p_i}} \left(\frac{H'_I(n)}{\#(I) - 1} \cdot \frac{H'_E(n)}{\#(E) + 1} - \frac{H'_I(n) \cdot H'_E(n)}{\#(I) \cdot \#(E)} \right) + \sum_{n \in W_{p_i}} \left[\left(\frac{H'_I(n) - d_1H'_I(n)}{\#(I) - 1} \cdot \frac{H'_E(n) - d_1H'_E(n)}{\#(E) + 1} \right) - \frac{H'_I(n) \cdot H'_E(n)}{\#(I) \cdot \#(E)} \right] \right\} \quad (3.25)$$

$$\frac{\{(\#(E) - \#(I))F' + \frac{1}{\#atributo} \sum_{n \in W_{p_i}} [H'_I(n) \cdot d_1H'_E(n) - (H'_E(n) + d_1H'_E(n))d_1H'_I(n)]\}}{(\#(I) - 1)(\#(E) + 1)}$$

$$\Delta_E = \frac{1}{\#(atributos)} \left\{ \sum_{n \notin W_p} \left(\frac{H'_I(n)}{\#(I) + 1} \cdot \frac{H'_E(n)}{\#(E) - 1} - \frac{H'_I(n) \cdot H'_E(n)}{\#(I) \cdot \#(E)} \right) + \sum_{n \in W_p} \left[\left(\frac{H'_I(n) + d_2 H'_I(n)}{\#(I) + 1} \cdot \frac{H'_E(n) - d_2 H'_E(n)}{\#(E) - 1} \right) - \frac{H'_I(n) \cdot H'_E(n)}{\#(I) \cdot \#(E)} \right] \right\} = (3.26)$$

$$\frac{\left\{ (\#(I) - \#(E))F' + \frac{1}{\#atributo} \sum_{n \in W_p} [H'_E(n) \cdot d_2 H'_I(n) - (H'_I(n) + d_2 H'_I(n)) d_2 H'_E(n)] \right\}}{(\#(I) + 1)(\#(E) - 1)}$$

6. O deslocamento a ser aplicado ao pixel p é determinado da seguinte maneira:

i) Compute $\Delta_{min} = \min\{\Delta_I, \Delta_E\}$.

ii) Se $\Delta_{min} \geq 0$, p não se desloca.

iii) Caso contrário, se $\Delta_{min} = \Delta_I$, p se desloca para p_i , senão p se desloca para

p_E .

No que diz respeito ao procedimento descrito acima, cabem as seguintes considerações:

1) Ele se refere a uma situação padrão em que um dos vizinhos na direção da normal é interno e o outro externo. Se tal como acontece num whisker forem ambos internos ou externos teríamos o seguinte:

i) Se forem ambos externos, então a suposição de que um pixel de borda também é externo faz com que a variação deslocando o pixel de borda em qualquer dos dois sentidos seja nula. O pixel portanto não será deslocado. Assim, quando na evolução do algoritmo, um whisker externo se forma ele pode, na sequência do processo, crescer mais, mas nenhum trecho dele, já formado, varia de posição.

ii) Whiskers internos ocorrem principalmente devido:

a) A uma oclusão que faça a parte visível do objeto a ser segmentado deixar de ser simplesmente conexa.

b) Partes no interior do segmento têm textura consideravelmente distinta do resto. Por exemplo, o olho do peixe na figura 3.8 abaixo.

Não há razão para mantê-los exceto o fato de se manter a conectividade do contorno ativo. Por simplicidade, quando os dois vizinhos na direção da normal, forem interiores, mudamos nossa convenção passando a considerar os pixels de borda também como interiores. Nesse caso, qualquer que seja o deslocamento de p ele não trará mudanças à função objetivo F' , o que significa que usando o procedimento acima, p não se moverá. Nesse caso as considerações feitas acima em relação aos whiskers externos se aplicarão também aos internos.



Figura 3.8: Objeto com textura interna consideravelmente distinta do resto do objeto.

Fonte: <http://animais.blogmaneiro.com/fotos/peixes-ornamentais>

2) Situações em que p pertence à borda da imagem, em que apenas um vizinho na direção da normal precisa ser considerado, ou casos extremos em que $I = 1$ ou $E = 1$ não foram considerados no procedimento acima e precisam ser tratados adequadamente.

3) A complexidade do processo que determina o movimento de cada pixel de borda é $O(R)$, onde R é o alcance do filtro Gaussiano empregado. A complexidade de um passo da evolução é, portanto, $O(NR)$ onde N é o número de pixels da borda. Tempos aceitáveis para todo o processo foram obtidos em função do contorno inicial estar próximo da borda do segmento e empregando um valor de R baixo.

4) Uma vez movimentados os pixels da borda, o novo traçado dela é tratado. Esse tratamento visa:

i) Eliminar sequências de pixels repetidos e a existência de pequenos loops de tamanho menor que uma dada constante K (parâmetro) definida a priori.

ii) Quando dois pixels consecutivos - p_k e p_{k+1} - não forem adjacentes acrescentar entre eles os pixels que fazem parte da representação digital do segmento que liga p_k a p_{k+1} dada pelo algoritmo de Bresenham [40].

5) Além de procurar diminuir o valor F' , podemos agregar ao processo que determina o deslocamento de um pixel num passo, outros objetivos como a redução do comprimento total do contorno ativo. Considerando-se esse objetivo após se movimentar cada pixel p de um contorno C pelo processo acima, aplicaríamos ao pixel já transformado, um deslocamento $-D_C$ na direção normal a C computada em p no sentido do interior desse contorno. Mas, observe que, enquanto pelo procedimento acima, o contorno evoluía num contexto raster, passando de pixel para pixel, esse novo deslocamento se dá num contexto vetorial. Para compatibilizar os dois devemos associar a cada pixel da borda, coordenadas que identifiquem um ponto dentro da área que esse pixel ocupa. Quando um pixel da borda é criado essas coordenadas podem identificar o ponto central dessa área. Posteriormente o deslocamento D_C pode levá-lo a outra posição relativa dentro da área elementar em que se encontra.

6) Se o objetivo a ser agregado é a redução de alguma energia interna do contorno

então precisamos identificar entre os pixels do contorno inicial aqueles que serão empregados na definição dessa energia- isto é, aqueles que efetivamente farão o papel de snaxels. Esses pixels devem ser acompanhados durante toda a evolução do contorno, o que implica dizer que não devem ser eliminados pelo tratamento indicado no item 4

Encerramos esta seção com a apresentação de três exemplos de segmentação. Nesses exemplos, na figura 3.9, a curva em amarelo é a inicial do processo de evolução. Ela foi obtida a partir da projeção de uma curva qualquer que contenha o segmento sobre o subespaço dos modos principais da classe do objeto a ser segmentado, na figura 3.10 é mostrado o objeto na imagem original. No caso da figura 3.11, o objeto está submetido a oclusão em três partes. Observe que os whiskers externos identificam essas oclusões.

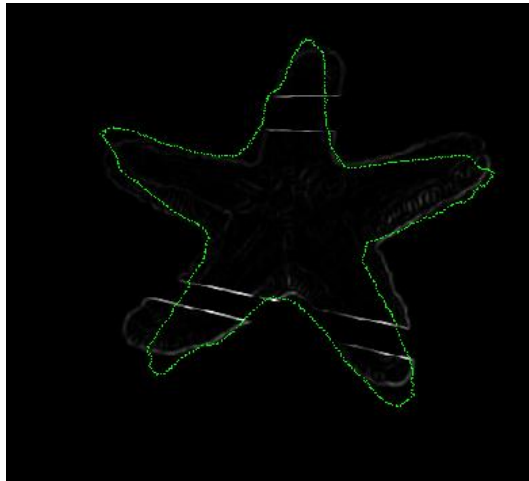


Figura 3.9: Curva inicial, projetada no subespaço dos modos principais da classe do objeto.

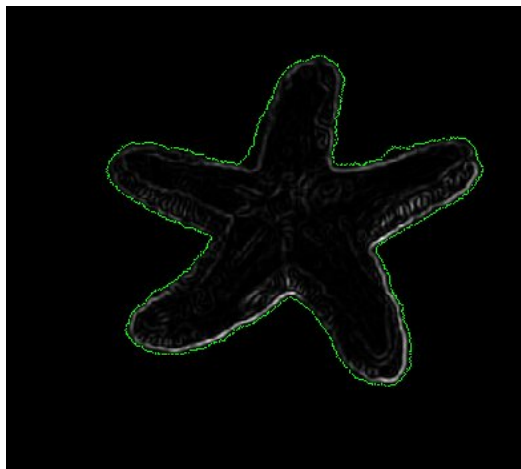


Figura 3.10: Objeto sem oclusão.

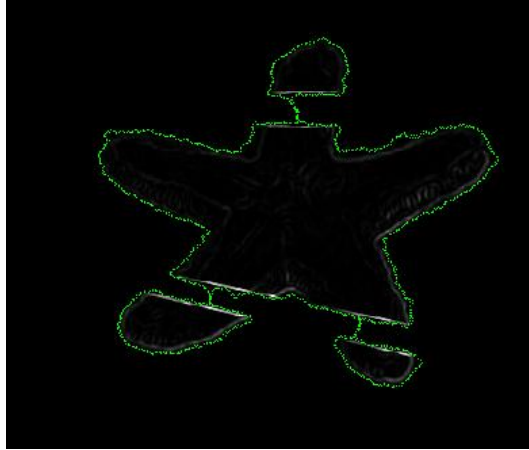


Figura 3.11: Objeto com oclusão em três partes.

3.8 Alternativas para o Tratamento da Oclusão

Uma questão em perspectiva quando se utiliza informação sobre a forma do objeto a ser segmentado é a possibilidade de tratar a oclusão, quando ela não encobrir substancialmente a imagem do objeto. Em vista disso, foi acrescentado ao sistema desenvolvido nesta tese um módulo para lidar com a oclusão do segmento. Acredita-se que ele possa funcionar satisfatoriamente se a condição(C1) dada a seguir for satisfeita.

“A textura predominante no objeto a ser segmentado é consideravelmente distinta tanto daquela do objeto que efetua a oclusão como a do fundo”. Propositamente na condição C1 os termos “predominante” e “consideravelmente distintos” não estão definidos de forma precisa. Isso porque o quanto deve ser essa predominância ou quão “diferentes” devem ser as texturas para que resultados satisfatórios sejam atingidos depende evidentemente do caso em questão e não consideramos pertinente, no atual estágio do trabalho, desenvolver estudo nesse sentido. Explicamos a condição C1 por que ela dá sentido ao rationale empregado na construção do módulo que trata oclusão. Esse tratamento é dividido em três etapas:

- I Detecção das partes oclusas.
- II Remoção dos trechos do contorno determinados pela oclusão.
- III Substituição desses trechos por correspondentes extraídos do modelo.

Entenda-se por modelo o contorno obtido efetuando-se a projeção de um traçado inicial dado pelo usuário sobre o espaço de modos principais da classe do objeto e ajustando-se a escala, orientação e posição. Esse ajuste é feito da seguinte maneira:

1. Transladando-se a projeção até que seu centro de massa coincida com o da figura dada inicialmente.

2. Escalando-se a projeção já transladada de forma que a distância média de seus pontos ao centro de gravidade seja a mesma da figura inicial.
3. Rodando-se a figura obtida em 2 até minimizar a distância entre ela e a figura inicial, nos moldes do que foi feito os elementos de uma classe em relação a curva média da classe.

Melhores resultados são obtidos se ao invés de usarmos a projeção da curva dada pelo usuário, aplicarmos o procedimento de segmentação descrito na seção anterior a ela e usarmos a projeção sobre os modos principais da curva resultante desse procedimento. O benefício de se aplicar além dessa, mais vezes o procedimento de segmentação é menor.

I - Detecção das partes oclusas

Na primeira etapa, além da detecção das partes oclusas propriamente dita, se pode estabelecer como as diferentes partes visíveis do objeto devem ser ligadas.

Vamos considerar dois casos. No primeiro, a parte que faz a oclusão tem textura similar ao fundo. Por exemplo, numa cena subaquática, uma corrente de água mais densa pode fazer com que partes do objeto apareçam pouco nítidas. No outro caso, o objeto que faz a oclusão é consideravelmente distinto do fundo.

No primeiro caso, uma vez evoluída a curva até se atingir uma dada regra de parada - uma alternativa é empregar como regra de parada, o fato de nenhum pixel ainda não atingido ser alcançado numa iteração - procuramos pela presença de whiskers externos ao contorno. A presença de estruturas desse tipo fará com que toda a parte ocludente fique do lado de fora sem que a curva seja feita desconexa.

Tendo em vista que estamos assumindo que objeto e fundo tem texturas distintas a presença desses whiskers será obrigatória se $F' = 0$. Não há necessidade de se montar uma estrutura específica para detectar os pixels que fazem parte dessas estruturas nos moldes do que é feito para detectar a auto-intersecção de uma T-snake.

Ocorre que para identificação da normal já construímos uma transformada da distância com sinal numa faixa ao longo do contorno. Um ponto de um whisker externo pode ser identificado pelo fato dessa transformada ter sinal positivo em ambos os lados da normal. Não há também necessidade de se determinarem pares de índices da sequência de pixels que define o contorno que se referem a um mesmo pixel. Os dois ramos do whisker podem ser tratados separadamente.

No segundo caso, em lugar dos whiskers o processo pode convergir para um trecho da borda entre o fundo e o objeto ocludente. Esses trechos podem ser determinados pelo fato do padrão do objeto alvo não aparecer na vizinhança. Então, se o esse

objeto tem textura homogêna reconstrói-se o histograma considerando essa textura e a do fundo como sendo uma só .

Procedendo assim, a cada borda do tipo fundo-objeto ocluser, onde esse último tenha textura diferente acabamos caindo no primeiro caso. Mas de fato não precisamos fazer isso. Se pudermos identificar esse tipo de borda podemos eliminar as sequências de pixels sucessivos que formam uma componente do contorno que seja desse tipo e ligar diretamente suas extremidades. Isso já deve ser suficiente para que as diversas partes visíveis possam ser ligadas adequadamente.

II - Remoção dos trechos do contorno determinados pela oclusão

Além dos whiskers externos cujo processo de identificação já foi descrito em I os trechos a serem removidos são aqueles que pertencem a borda de um objeto, ocludente ou não diferente daquele que queremos segmentar. Se esse tipo de borda puder ser identificado, ótimo, mas senão, como ocorre no caso em que as texturas do fundo e do objeto ocluser são pouco distintos empregamos duas heurísticas podem ser úteis em boa parte das situações.

1. Assumindo que o contorno inicial está próximo do objeto alvo, a presença do objeto ocludente faz com a evolução de boa parte dos pixels nas regiões oclusas tenha uma trajetória maior que a média. Assim, se para cada pixel de um contorno gerado durante a evolução associarmos um label que indica o pixel do contorno inicial que o gerou, podemos ao fim da evolução detectar a distância entre pixels do contorno final e seus correspondentes no contorno inicial e detectar aqueles para os quais essa distância é muito maior que a média.
2. Nas extremidades dos trechos a serem eliminados em geral há uma descontinuidade na orientação. Para tornar mais robusto o cômputo, eliminando oscilações locais, calculamos para cada pixel x_i o ângulo formado entre os segmentos $[x_{i-k}, x_i]$ e $[x_i, x_{i+k}]$. Valores de k experimentados: 5, 10, 15.

III - Substituição desses trechos por correspondentes extraídos do modelo

Para substituir um trecho de extremidades x_{i1} e x_{i2} que foi retirado do contorno final procedemos da seguinte forma.

1. Para cada $x_{ij}, j=1,2$ identificamos no contorno inicial da evolução os pixel y_{ij} a partir do qual ele foi gerado.
2. Seja então $d_{ij} = x_{ij} - y_{ij}, j=1,2$.

3. Para cada pixel y_k do contorno inicial com $i1 < k < i2$ determinamos as distâncias D_{k,i_j} , $j=1,2$ entre y_k e y_{i_j} .

4. Computamos:

$$z_k = y_k + \frac{D_{k,i2}d_{i1} + D_{k,i1}d_{i2}}{D_{k,i2} + D_{k,i1}} \quad (3.27)$$

5. Substituímos o trecho retirado da curva final pela sequência dos z_k , $i1 < k < i2$

Essa substituição do trecho removido por uma deformação do tipo Moving Least Squares do trecho (y_{i1}, y_{i2}) , da curva inicial revelou-se satisfatória considerando que os trechos oclusos são pequenos.

Capítulo 4

Implementação

Para a implementação da ferramenta a ser utilizada para o desenvolvimento do trabalho houve maior esforço em criar uma interface simples que permitisse a avaliação visual de quase todo o processo a ser realizado, bem como permitir que os modelos fossem gerados e gravados em disco.

4.1 Algoritmos

Para melhor compreensão vamos dividir o processo em dois grupos: Pré-processamento e Segmentação.

Pré-processamento, é a parte do sistema onde se processa o conjunto de poligonais de modo a se obter métricas representativas da dada classe. As principais operações realizadas nesse grupo são: Transformada da Distância, Derivadas Parciais, a direção normal e gradiente em cada ponto.

Para facilitar a compreensão essas operações foram reunidas em um único processo no fluxograma(processar), mostrado na figura 4.1. O resultado do pré-processamento é o conjunto de modos principais para o dado conjunto de modelos e uma figura média. Durante a execução do programa o conjunto de modos principais e a figura média ficam em memória, podendo os mesmos serem gravados em disco para posterior utilização sem necessidade de refazer o pré-processamento.

Segmentação, é a parte do sistema onde uma curva definida inicialmente pelo usuário (segundo os critérios da sessão 3.7) é utilizada para gerar os histogramas interno e externo. Também é a parte onde se realiza a evolução da curva inicial (projetada nos autovetores) seguindo a direção do gradiente projetado na direção normal à curva, para cada ponto, gradiente, esse, definido pelos histogramas das regiões, bem como a detecção da oclusão. Do mesmo modo, temos o fluxograma para a parte de segmentação também mostrado na figura 4.1.

Estrutura de dados

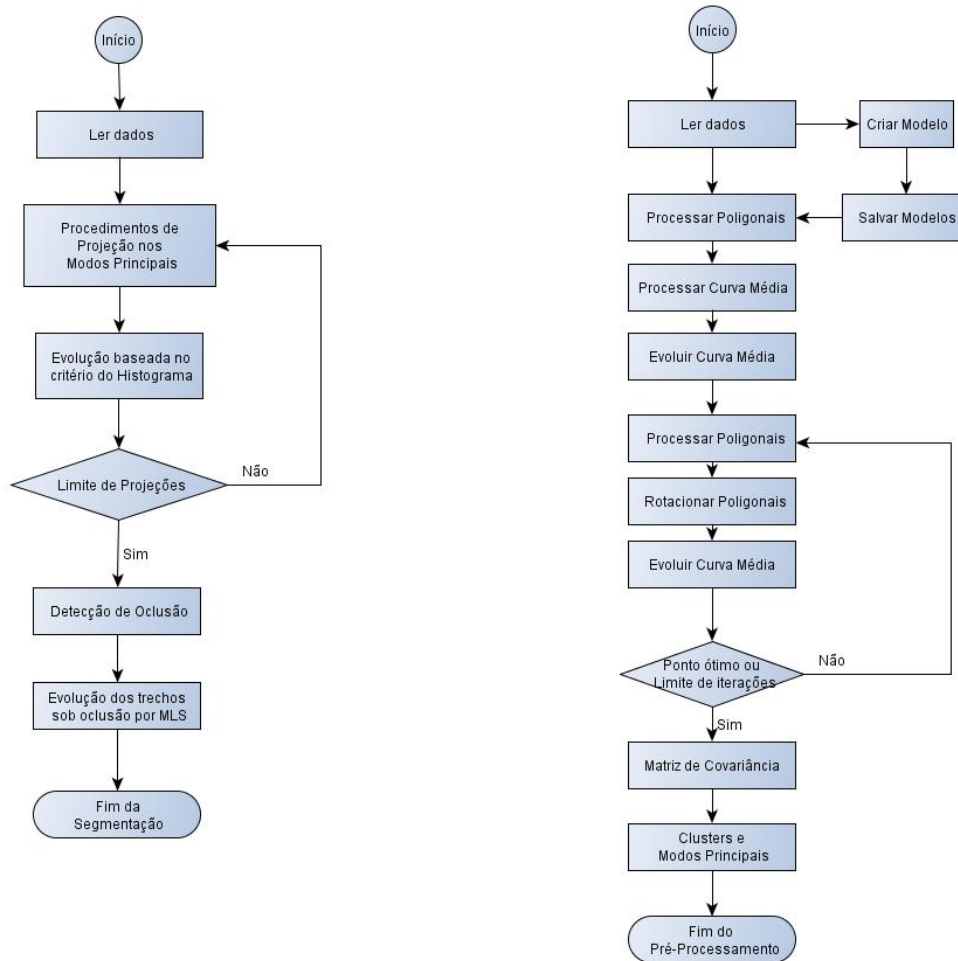


Figura 4.1: Fluxograma para a segmentação (direita) e Fluxograma do pré-processamento(esquerda)

Os dados foram estruturados seguindo uma modelagem orientada a objetos. Foi definida uma classe Poligonal que fosse capaz de representar uma poligonal com todas as características de interesse relacionadas à proposta desenvolvida, como por exemplo, a matriz com a transformada da distância, outra duas com as derivadas parciais em x e em y , pontos, gradientes, normais, entre outras. Os pontos da poligonal, como esta representa um modelo fechado e por se tratar de uma alocação dinâmica, foram colocados em uma lista duplamente encadeada e também colocados na classe. Para representação do conjunto de modelos foi criada uma classe Poligonais com a mesma proposta de representar um conjunto de poligonais com todas as características de interesse relacionadas à proposta desenvolvida, como por exemplo, a matriz com o somatório das transformadas das distâncias das poligonais, figura média, a matriz de covariância, os modos principais, entre outros. Por se tratar também de conjunto de dados alocados dinamicamente as poligonais também foram colocadas em uma lista duplamente encadeada.

4.1.1 Pré Processamento

Processo da evolução da Curva Média

Para que se possa compreender o processo aqui descrito para a evolução de um contorno ativo com o objetivo de obter a figura média, é necessário primeiramente entender que esse processo se divide em dois passos:

1. Minimização de μ

O primeiro passo foi definidos na equação 3.2, descrita novamente a seguir:

$$\partial_t \Gamma(t) = -\nabla_{\Gamma} \mu(\Gamma, R_i(\Gamma_i), \dots, R_N(\Gamma_N))$$

Essa primeira equação diferencial, tem o objetivo de minimizar μ fixando as transformações rígidas R_i em relação ao conjunto de modelos, e realizando um movimento em Γ na direção oposta ao gradiente para cada um de seus ponto.

Para se obter os gradientes da uma curva Γ_i foi utilizado o modelo discretizado mostrado na figura 4.2:

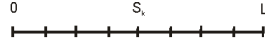


Figura 4.2: Representação dos pontos de Γ_i

A partir dessa representação, a curva Γ_i pode ser escrita na forma paramétrica $(x(s_1), y(s_1), x(s_2), y(s_2), \dots, x(s_n), y(s_n))$, logo o gradiente para cada ponto de Γ_i ($\nabla_{\Gamma} \mu$) pôde ser obtido da seguinte forma $\left(\frac{\partial \mu}{\partial x(s_1)}, \frac{\partial \mu}{\partial y(s_1)}, \frac{\partial \mu}{\partial x(s_2)}, \frac{\partial \mu}{\partial y(s_2)}, \dots, \frac{\partial \mu}{\partial x(s_n)}, \frac{\partial \mu}{\partial y(s_n)} \right)$.

2. Minimização do R_i

O segundo passo foi definido na equação 3.3, descrita novamente a seguir:

$$\partial_t R_i = -\nabla_{R_i} E^2(\Gamma, R_i(\Gamma_i))$$

Essa equação tem por objetivo, minimizar a distância entre o conjunto de modelos e Γ .

Uma vez obtido Γ mínimo no passo um, move-se (translação, escala, rotação) rigidamente todo o conjunto, de modo que, a distância de Γ_i para Γ seja mínima.

O R_i representa o movimento (translação, escala, rotação) que são aplicadas à Γ_i . Evidentemente a melhor translação pode ser obtida posicionando todos os modelos no centro da imagem, assim como, a melhor escala pode ser obtida obtendo-se o

raio médio do conjunto e escalando todos os modelos para que tenham o mesmo raio médio.

Como as operações de translação e escala podem ser obtidas de forma mais simples, todos os modelos foram inicializados no posicionamento ótimo com relação à esses dois parâmetros, logo, no passo dois, somente o parâmetros rotação deve ser encontrado.

Para encontrarmos o ângulo de rotação que minimize a equação 3.3 foi utilizado um sistema descrito pela figura 4.3:

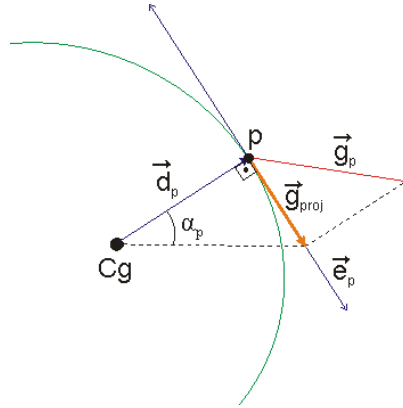


Figura 4.3: Sistema de vetores que descreve \vec{g}_{proj} , que é a projeção do gradiente \vec{g} no ponto p em um vetor \vec{e}_p , o qual é perpendicular à \vec{d}_p um vetor que parte do centro de gravidade Cg até o ponto p . Com isso, é possível a obtenção do ângulo de rotação α_p .

A melhor rotação para Γ_i foi obtida fazendo-se $\bar{\alpha} = \frac{1}{N} \sum_{p=0}^N \alpha_p$, onde N é o número de pontos em Γ_i .

Uma vez obtido o melhor R_i para cada Γ_i , torna-se a novamente otimizar Γ e o processo segue iterativamente alternando os passos um e dois até que os movimentos de Γ e de Γ_i não sejam mais significativos, ou, um dado limite de iterações seja alcançado.

Processo Matriz de Covariância e Modos Principais

Este processo é responsável pelo cálculo da matriz de covariância para o conjunto de amostras, a análise das componentes principais e definição dos modos principais de variação.

A matriz de covariância foi computada conforme descrito na sessão 3.3.2 e computado o PCA para a mesma, a partir daí, obtidos os modos principais de variação.

Os mesmos foram associados ao conjunto de amostras para a utilização nos processos de segmentação.

4.1.2 Processo de Segmentação

O processo de segmentação consiste em fazer uma projeção de um contorno inicial no espaço gerado pelos autovetores, evoluir esse novo contorno pelo método do Histograma, definido no capítulo anterior, refazer a projeção a fim de encontrar uma melhor região inicial, detectar trechos com oclusões, refazer a evolução desses trechos utilizando-se do mecanismo de MLS.

Procedimentos de Projeção

O Objetivo deste módulo, é fazer com que o contorno fique o mais próximo possível da solução final. Basicamente, efetua-se a projeção do traçado inicial dado pelo usuário sobre o espaço de modos principais da classe do objeto e ajustando-se a escala, orientação e posição. Esse ajuste é feito da seguinte maneira:

1. Transladando-se a projeção até que seu centro de massa coincida com o da figura dada inicialmente.
2. Escalando-se a projeção já transladada de forma que a distância média de seus pontos ao centro de gravidade seja a mesma da figura inicial.
3. Rodando-se a figura projetada até minimizar a distância entre ela e a figura inicial, nos moldes do que foi feito os elementos de uma classe em relação a curva média da classe.

Evolução baseada no Histograma

Esse módulo implementa os algoritmos descritos na sessão 3.7. Como resultados desse módulos, a seguir serão apresentadas uma série de imagens segmentadas na figura 4.4

Detector de Oclusão

Esse módulo implementa as três etapas descritas na sessão 3.8, a saber.

- I Detecção das partes oclusas.
- II Remoção dos trechos do contorno determinados pela oclusão.
- III Substituição desses trechos por correspondentes extraídos do modelo.

Para mostrar o resultado desse módulo, na figura 4.5 são mostrados as imagens das três etapas descritas acima:

Na figura 4.6 serão mostrados mais alguns exemplos do processo aplicado a imagens reais.

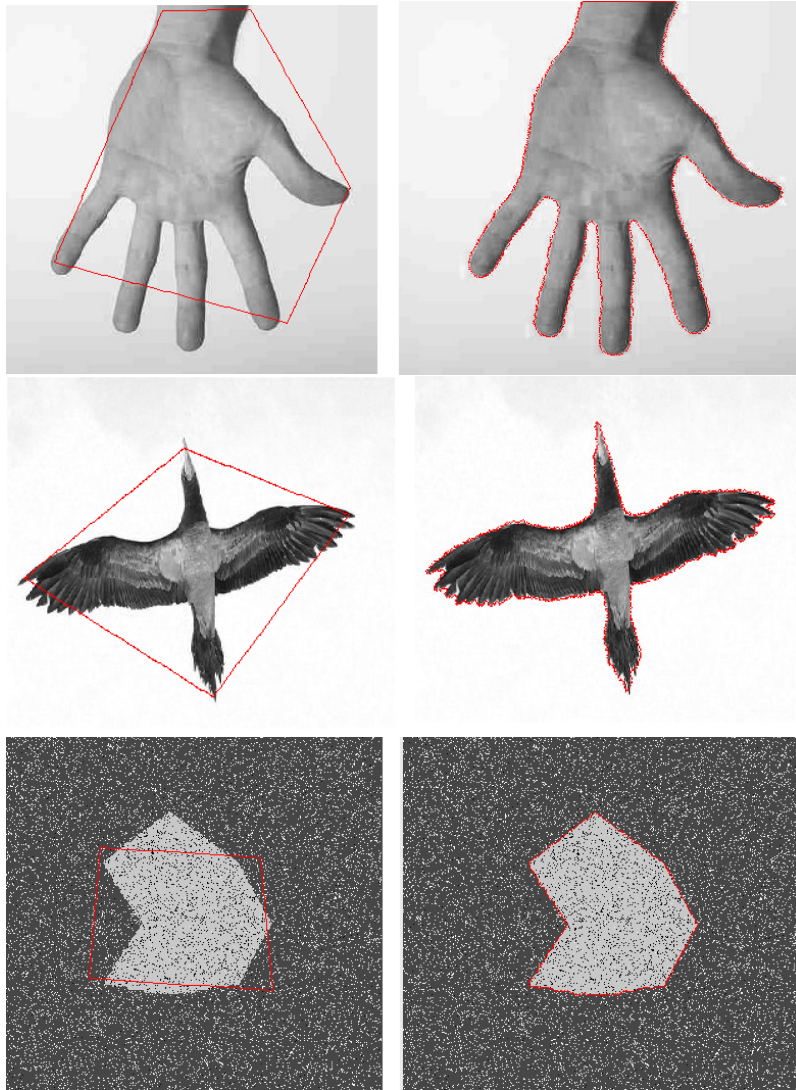


Figura 4.4: Alguns modelos segmentados pelo processo descrito.
 Fonte: http://apenas24horas.blogspot.com/2010_06_01_archive.html
 Fonte: <http://neo-ambiente.blogspot.com/2010/04/vida-aves.html>

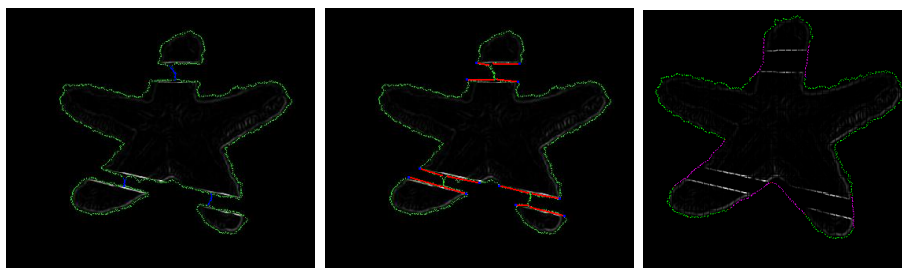


Figura 4.5: Resultados obtidos pelos detector de oclusão. Na primeira (direita), são detectados as partes oclusas, na segunda os trechos do contornos, e na terceira a substituição do trechos selecionados pelos correspondentes na figura de referência.

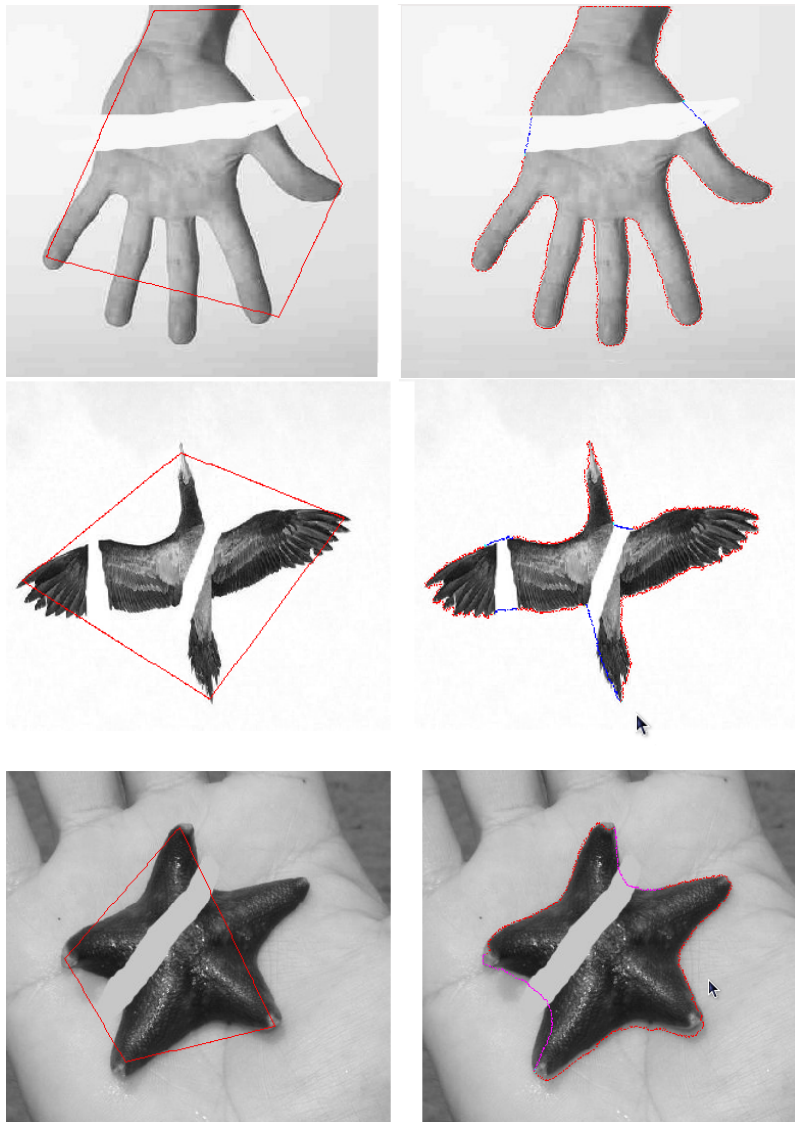


Figura 4.6: Um conjunto de imagens reais, no lado esquerdo o contorno inicial para a segmentação, do lado direito o contorno segmentado e a oclusão já tratada.

Fonte: http://apenas24horas.blogspot.com/2010_06_01_archive.html

Fonte: <http://neo-ambiente.blogspot.com/2010/04/vida-aves.html>

Fonte: <http://www.baixaki.com.br/papel-de-parede/25477-estrela-do-mar.htm>

4.1.3 Classificador

Tento em vista a utilização de um sistema de representação (curva média, modos principais dos gradientes de deformação) tentamos estabelecer um classificador, de modo, a identificar se um dado objeto pertence a uma dada classe. Entretanto, esse classificador não apresentou resultados esperados. Entretanto, achamos interessante apresentar alguns dos resultados obtidos com o mesmo.

Mecanismo de classificação

Foram utilizados os gradientes do processo de deformação elementar que leva a curva média da classe($\hat{\Gamma}$) a se tornar o mais próximo possível das curvas Γ_j da classe. Esse gradiente, conforme já foi visto, tem a direção da normal a curva média em cada um de seus pontos. Tomando a norma desses gradientes em cada ponto da curva média e para cada curva da família obtemos a matriz N_{ij} , onde i é o número de pontos em $\hat{\Gamma}$ e j o número de formas da classe. A partir dessa matriz computamos a matriz de covariância M , definimos um espaço característico do conjunto de formas da classe, que é o subespaço gerado pelos autovetores principais dessa matriz. A partir daí, cada elemento da classe passou a ser considerado como um ponto nesse espaço. Dentro desse contexto tentamos estabelecer um classificador de formas procedendo da seguinte maneira:

Dado um novo elemento, repetimos para ele todo o processo acima, encontrando um gradiente de deformação para ele nos pontos da curva média e projetando o vetor determinado pelas normas desses gradientes no espaço característico da classe. Se a diferença entre o vetor composto pelos gradientes de projeção da nova curva e sua projeção sobre o espaço característico da classe tivesse um valor reduzido, consideramos que a nova forma poderia pertencer à classe.

Entretanto, esse classificador não demonstrou ser efetivo em boa parte dos casos. Quando a variância dentro da classe era baixa, a nova forma, ainda que aparentemente similar, pode se afastar mais da curva média que essa variância. O reconhecimento, não é, então, efetuado o que pode ser considerado insatisfatório devido a similaridade visual da forma testada em relação às da família. Em casos em que a variância era maior foi possível obter vários exemplos de falsos positivos.

Para investigar o motivo dessa inefetividade, fizemos um conjunto de testes com uma classe grande, variando-se entre 100 e 200 elementos, e com uma variância controlada. Os elementos foram gerados por uma função de densidade de probabilidade. Empregando essa mesma função, geramos novos elementos para serem classificados e consideramos que o novo elemento gerado deveria pertencer à classe. Entretanto a partir de uma dada variância para a classe resultados negativos surgiram. Esse experimento se refere ao exemplo da figura 4.7 e 4.8 .

A idéia dessa função é receber um modelo inicial, onde, a distância entre um ponto e seus vizinhos é D . No nosso caso, fizemos um quadrado com oito vértices(cada lado tinha um vértices central) e essa variação foi aplicado apenas aos vértices centrais.

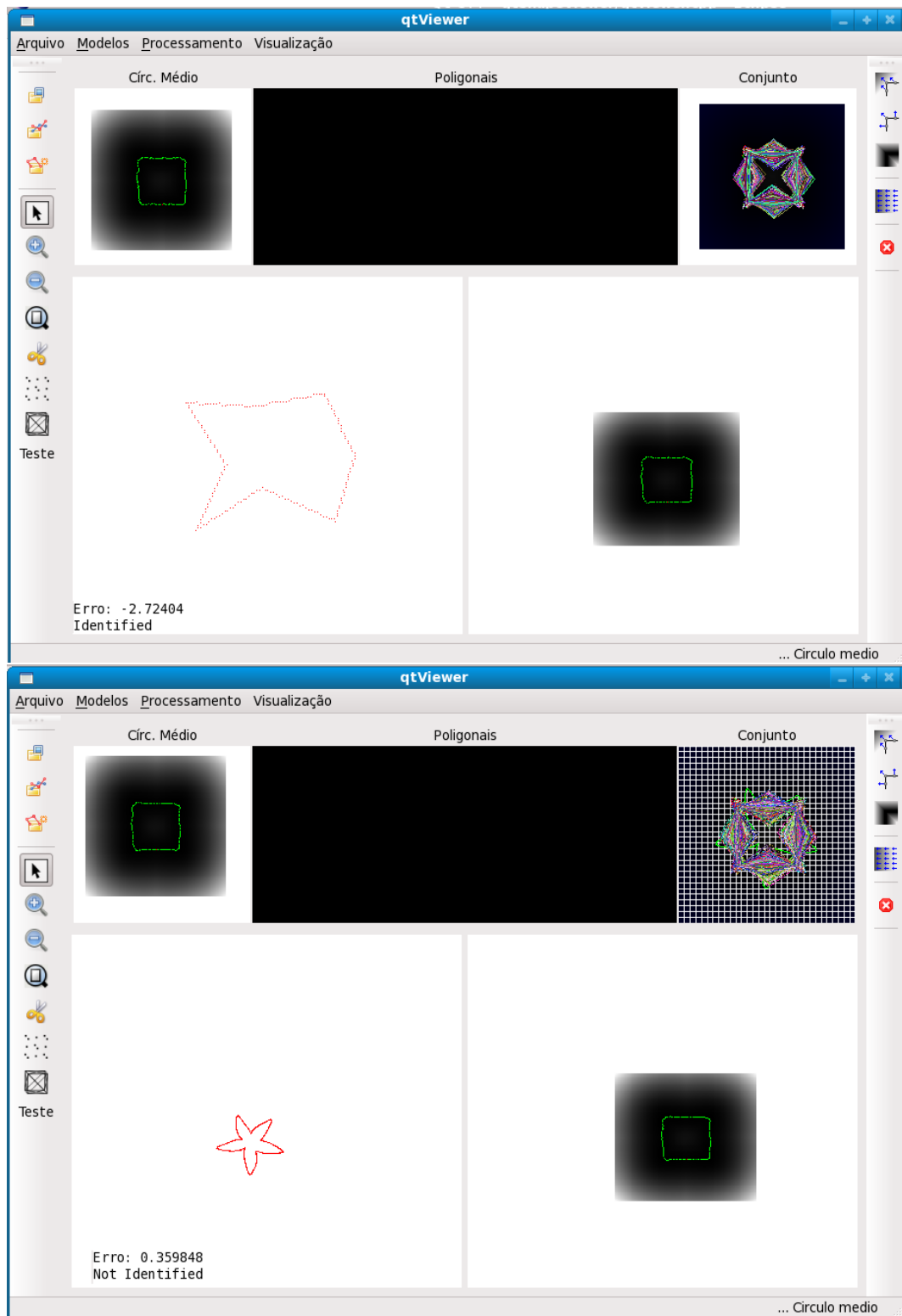


Figura 4.7: Tela principal de visualização da ferramenta com os exemplos utilizando um conjunto de com 150 elementos. Primeiro com um elemento gerado pela própria função e o outro com uma estrela.

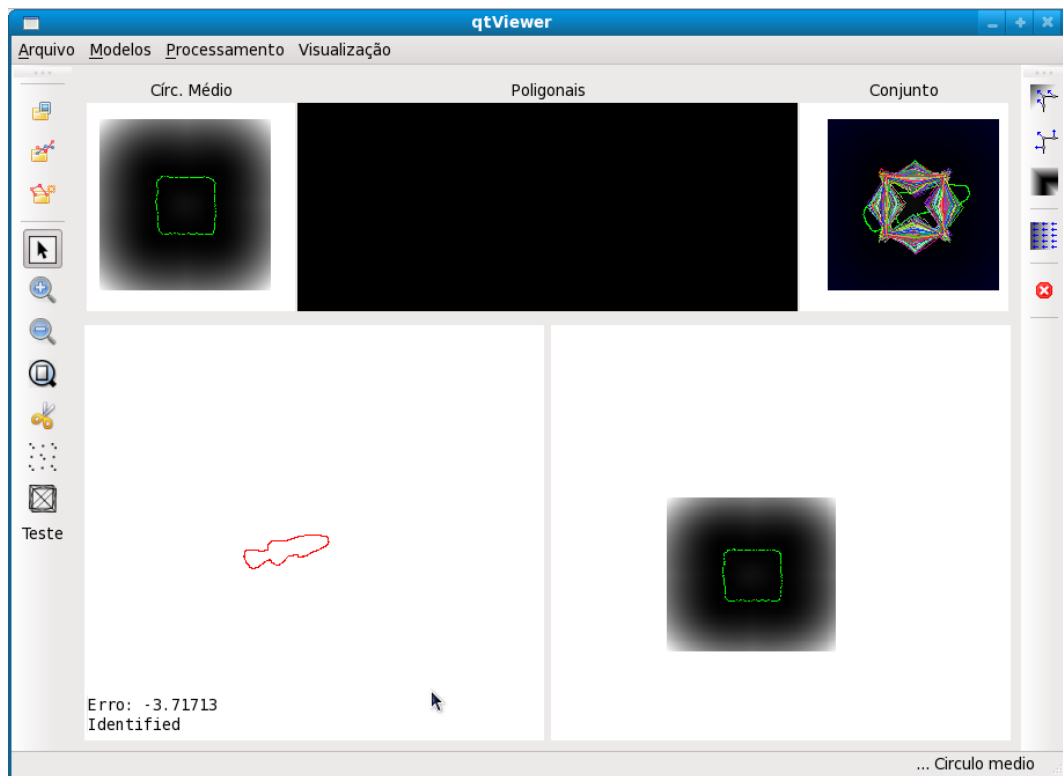


Figura 4.8: Tela principal de visualização da ferramenta com os exemplos utilizando um conjunto de com 150 elementos. Sendo um peixe como elemento a ser classificado.

4.2 Interface

Para a construção da ferramenta a ser utilizada para o desenvolvimento do trabalho houve maior esforço em criar uma interface simples que permitisse a avaliação visual de quase todo o processo a ser realizado, bem como permitir que os modelos fossem gerados e gravados em disco.

A tela principal da ferramenta consiste em 5 áreas de visualização e barras de ferramentas laterais, conforme mostra figura 4.9.

1 – Área destinada a visualização do processo de segmentação.

2 – Poligonais – Área destinada a visualização das poligonais que compõe o conjunto de treinamento, normalmente cada poligonal dentro do conjunto se apresenta por uma cor própria (atribuída aleatoriamente no momento em que se carregam as informações sobre os modelos (poligonais) conforme mostra a figura 4.9.

3 – Figura média – Área destinada a visualização da curva média, bem como sua evolução.

4 – Área destinada à visualização do conjunto de testes (poligonais) sobrepostos, conforme mostra a figura 4.9.

5 – Área destinada à exibição de imagens de interesse para cada estágio do processo.

A utilização da ferramenta inicia-se carregando-se o conjunto de poligonais aces-

sando o menu Processamento/processamento geral os modelos serão ajustados em suas dimensões, movidos para o centro de gravidade geral, rotacionados para minimizar a distância entre si, a curva média será calculada, será feita a evolução da mesma, baseado no somatório da transformadas das distancias de cada modelo, e realizado o PCA (Principal Component Analysis) para o conjunto de poligonais. O resultado desse processamento é mostrado na figura 4.9, onde é possível ver os modelos já ajustados quanto a dimensão, rotacionados, com o mesmo centro de gravidade a figura média já em seu estado final.

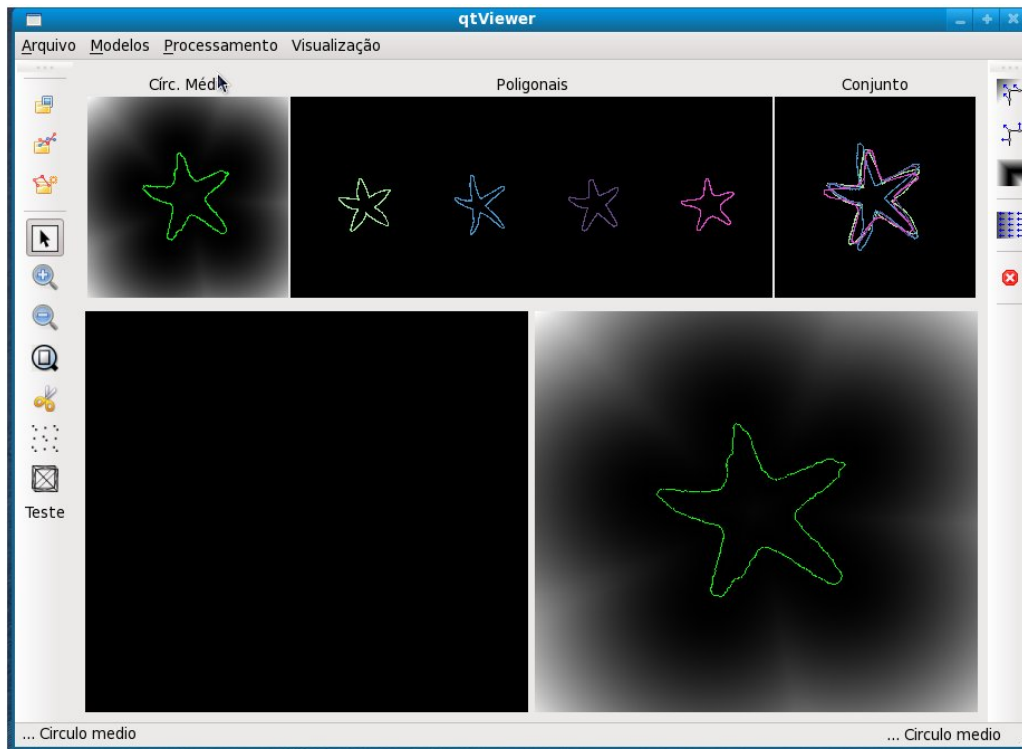


Figura 4.9: Tela após o processamento geral

A ferramenta possui também uma tela específica que possibilita a visualização das matrizes da Transformada da Distância (TD) de cada poligonal, o Somatório da Transformada da Distância (STD) para o caso da figura média, os gradientes da matriz que estiver sendo visualizada e os gradientes e normais calculados nos pontos das curvas. Esta tela pode ser acessada com um duplo click em qualquer poligonal que esteja visível em tela, conforme mostra figura 4.10.

Dependendo da poligonal escolhida a opção STD muda para melhor se adequar a situação, por exemplo, ao se selecionar a figura média é interessante poder visualizar a matriz de Somatório das Transformadas das Distancias que é a base para as métricas do mesmo, entretanto, ao se selecionar uma poligonal do conjunto de treinamento, é mais interessante que seja mostrada a Transformada da Distância da figura média que é a base para algumas métricas para a dada curva, conforme mostram figuras 4.11 e 4.12.

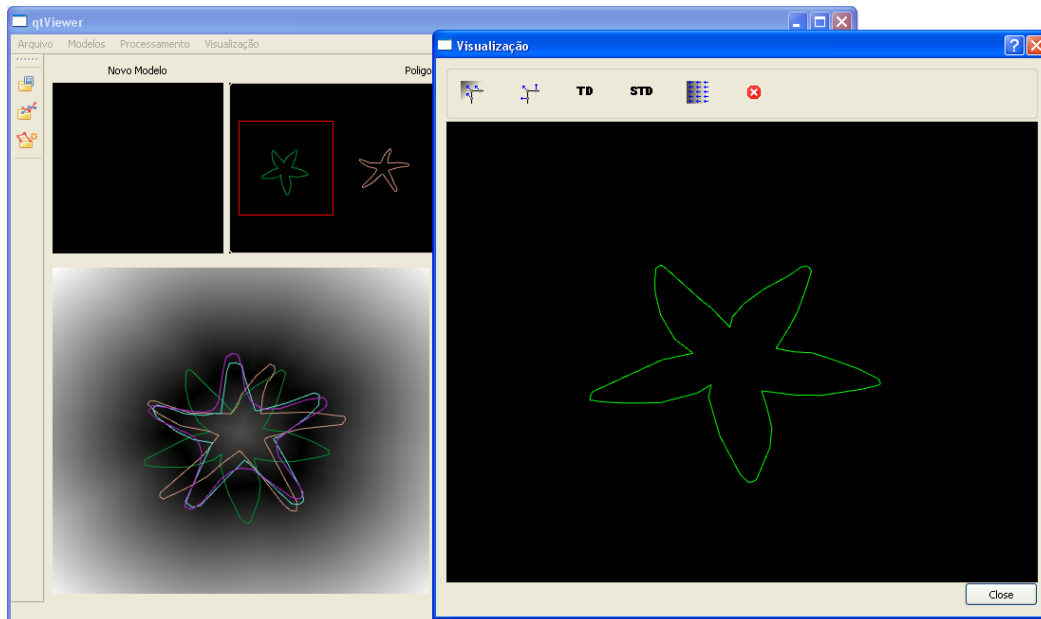


Figura 4.10: Tela de visualização detalhada da poligonal

4.3 Ferramentas de Desenvolvimento

A linguagem de desenvolvimento escolhida foi C++ devido ao fato de ser uma das mais robustas existentes hoje em dia, a eficiência demonstrada ao longo dos anos, a busca por uma maior portabilidade entre as plataformas existentes. Para a interface foi utilizado o pacote Qt pelo fato de permitir a fácil implementação de interfaces complexas e também multiplataforma não interferindo significativamente na portabilidade do programa.

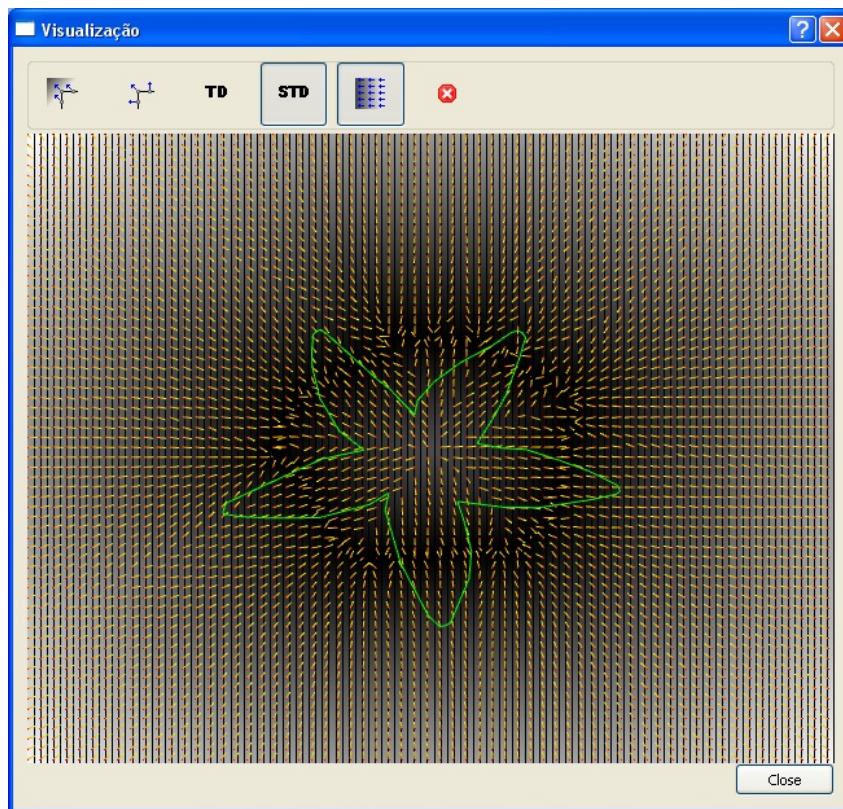


Figura 4.11: Foi utilizada uma poligonal a partir do conjunto de modelos, ao fundo a matriz de Transformada da Distancia do Circulo Médio (Opção selecionada STD) e os gradientes da dada matriz

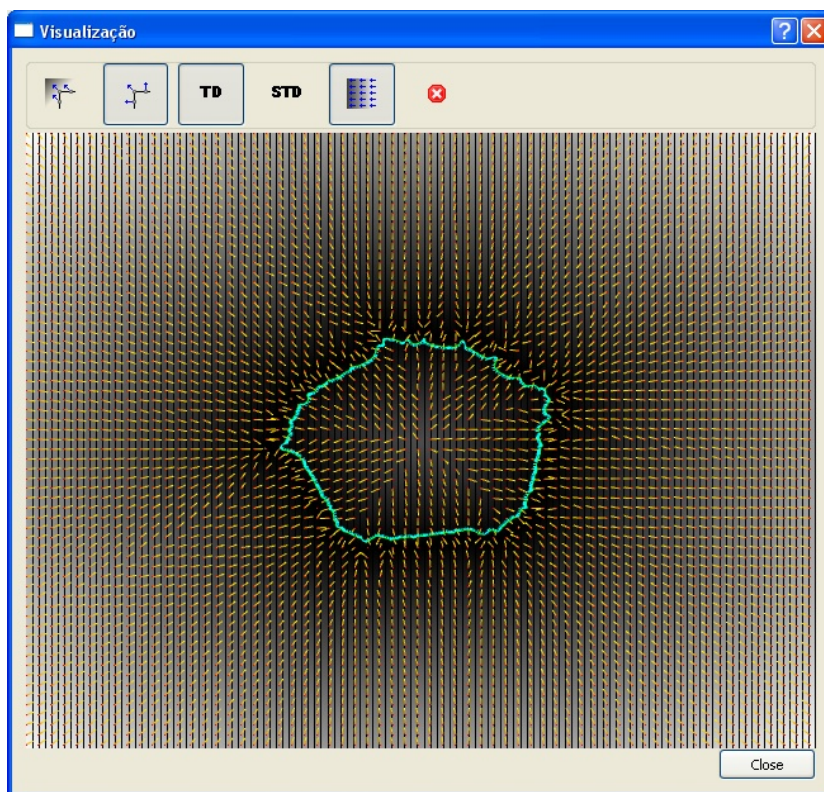


Figura 4.12: Foi utilizado a poligonal do círculo médio, ao fundo a matriz de Transformada da Distancia da poligonal selecionada (Opção selecionada TD) e os gradientes da dada matriz em exibição, e as normais em cada ponto da curva média também em exibição.

Capítulo 5

Conclusões e Trabalhos Futuros

5.1 Conclusões

Neste trabalho, abordamos o problema de se empregar informação a priori sobre a forma de um objeto no processo de segmentá-lo. Essa informação consiste simplesmente em se indicar a classe a que ele pertence para a qual já foi obtida uma dada representação, essa representação é a curva média de classe e os modos principais definidos como os autovetores principais da matriz de covariância dos gradientes em cada ponto da curva média, das deformações que levam a curva média em cada curva pertencente a classe.

A vantagem dessa metodologia é que um elemento da matriz de covariância passa a se referir a um par de pontos da curva média tomando a forma $\sum_{c_i \in \text{classe}} \alpha_{c_i}(i) \cdot \alpha_{c_i}(j)$ onde $\alpha_{c_i}(k)$ representa o gradiente de deformação da curva c_i no pixel k da curva média. Se não referíssemos tudo à curva média precisaríamos encontrar, em cada curva, os pontos associados à variável i e os associados à variável j , mas, é complicado estabelecer um conjunto de pontos correspondentes em cada curva da classe para serem associados à essas variáveis, ainda mais considerando que essas curvas tem tamanhos diferentes e podem estar sequenciadas a partir de pontos quaisquer de seus pontos.

A utilização dessa representação (curva média, modos principais dos gradientes de deformação) fará se construir um classificador, ou seja, um procedimento que decide se um dado objeto pertence à uma classe ou não, não redundou num processo sempre satisfatório. Situações em que os autovetores usados para obter a representação da classe são mais similares entre si do que uma instância, que ainda assim um observador indicaria pertencer à classe, produziram como resultado um falso negativo. Por outro lado, em classes com muitos elementos, consideravelmente diversificados, um objeto distinto de qualquer um deles, acabou prejudicando os resultados com falso positivos. Ambas as situações se revelaram mais comuns do que

inicialmente se pensou.

Como auxiliar a um processo de segmentação, entretanto, o uso dessa representação foi útil permitindo a obtenção de configurações iniciais próximas do contorno alvo. Em relação a isso, deve-se esclarecer que alguma informação posicional tem de ser passada pelo usuário pois não se tentou aqui, de questão de dados, uma imagem representando múltiplos objetos quais deles se enquadrariam numa dada classe.

Até que para diversificar a abordagem, fizemos o processo evolutivo que parte de um modelo obtido a partir de representação da classe para o contorno alvo, se orientado de forma a se reduzir a energia definida pelo produto escalar dos histogramas das partes interior e exterior do contorno vigente, assim toda a imagem e não apenas a vizinhança do contorno influi em sua evolução o que pode ser útil quando o segmento tem um padrão diferenciado do fundo e também porque se pode dispensar o procedimento de difusão usado para ampliar a influência de um contorno quando se aplica um método de snaxels clássico. Além disso, em condições normais, o processo permite a detecção de oclusões pela formação de whiskers externos ao segmento o que foi empregado para fazer aparecer as partes oclusas, obtendo para elas versões produzidas deformando por MLS trechos correspondentes do modelo.

5.2 Trabalhos Futuros

Entre os aprimoramentos do trabalho inicial nesta tese, o de maior prioridade é sem dúvida a inclusão de marcadores nos pontos mais significativos dos contornos.

A idéia é que os elementos de cada classe tenham o mesmo número de marcadores e que se estabeleça uma correspondência 1-1 entre eles o contorno inicial do processo de evolução já será gerado com esses marcadores e alguma energia interna baseada em suas posições poderá ser gerada. Os marcadores ajudarão também no processo de se orientar adequadamente o contorno inicial. Outra alternativa a ser experimentada é o emprego de uma “energia de modelo” para evitar que os pixels do contorno inicial evoluam segundo trajetórias mais longas tal como ocorre nos trechos que estão obstaculados.

Um caso que se pretende tratar especificamente é o de oclusões provocadas por objetos pequenos e de formas simples disseminados na imagem. Essa situação é usual no contexto de uma imagem subaquática.

Finalmente, há uma série de tarefas ainda a serem realizadas que dariam mais consistência ao trabalho desenvolvido aqui, entre elas, a de comparar os resultados obtidos pela metodologia descrita aqui com outros que usam formas diferentes de obter os modos principais. Além disso, a questão de se estudar o efeito da variabilidade de formas dentro de uma mesma família. Famílias em que ele for maior necessitam de mais modos principais para que a projeção sobre o subespaço gerado

por eles não seja de-correlacionado da forma que se quer segmentar.

Finalmente, considerando em particular que o processo de evolução empregado é “time-consuming” há uma série de procedimentos que ele efetua cuja implementação precisaria ser otimizada.

Referências Bibliográficas

- [1] CHARPIAT, G., FAUGERAS, O., KERIVEN, R. *Shape Statistics for Image Segmentation with Prior*. Research Report 07-29, CERTIS, ENPC, Marne la Vallee, jan. 2007.
- [2] COOTES, T. F., TAYLOR, C. J., COOPER, D. H., et al. “Active Shape Models - Their Training and Application”, *Computer Vision and Image understanding*, v. 61, n. 1, pp. 38–59, jan. 1995.
- [3] COOTES, T. F., TAYLOR, C. J. “Statistical Models of Appearance for Computer Vision.”, 1986. Disponível em: <http://www.isbe.man.ac.uk/~bim/Models/app_model.ps>. Acesso em: 14 maio 2010, 12:01:00.
- [4] MEMARSADEGHI, N., MOUNT, D. M., NETANYAHU, N. S., et al. “A fast implementation of the isodata clustering algorithm”, *International Journal of Computational Geometry and Applications*, v. 17, n. 1, pp. 71–103, fev. 2007.
- [5] SMITH, M., GOODCHILD, M., LONGLEY, P. *Geospatial Analysis. A Comprehensive Guide to Principles, Techniques and Software Tools*. 3 ed. De Montfort Mews, Matador, 2009.
- [6] JIA, X., RICHARDS, J. A. “Efficient Maximum Likelihood classification for imaging spectrometer data sets”, *IEEE Transaction on Geoscience and Remote Sensing*, v. 32, n. 2, pp. 274–28, mar. 1994.
- [7] RICHARDS, J. A., JIA, X. *Remote sensing digital image analysis: an introduction*. Springer-Verlag Berlin Heidelberg, 2006.
- [8] FIELD, D. “What is the goal of sensory coding?” *Neural Comput.*, v. 6, pp. 559–601, 1994.
- [9] LEWICKI, M. S., OLSHAUSEN, B. A. “A Probabilistic Framework for the Adaptation and Comparison of Image Codes”, *J. Opt. Soc. Am. A*, v. 16, pp. 1587–1601, 1999.

- [10] CANNY, J. “A Computational Approach to Edge Detection”. , 1986. Disponível em: <http://www.limsi.fr/Individu/vezien/PAPIERS_ACS/canny1986.pdf>. Acesso em: 18 de Abril de 2010, 16:30:00.
- [11] SIEBERT, A. “Dynamic Region Growing”. , 1986. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.40.7567&rep=rep1&type=pdf>>. Acesso em: 04 Jul. 2010, 11:35:30.
- [12] STATHIS, P., KAVALLIERATOU, E., PAPAMARKOS, N. “An Evaluation Technique for Binarization Algorithms”, *Journal of Universal Computer Science*, v. 14, n. 18, pp. 3011–3030, out. 2008.
- [13] WEINGARTNER, T., DILLMANN, R. “Split-and-merge segmentation using octrees”. In: *ACCV '95 Second Asian Conference on Computer Vision*, v. 2, pp. 57–61, Singapore, dez. 1995.
- [14] BEUCHER, S., C., L. “Use of watersheds in contour detection”. In: *Proceedings of International Workshop on Image Processing, Real-time Edge and Motion Detection/Estimation*, v. 2, pp. 2.1–2.12, Singapore, set. 1979.
- [15] ZENG, Y., SAMARAS, D., CHEN, W. AND PENG, Q. “Topology cuts: A novel min-cut/max-flow algorithm for topology preserving segmentation in N-D images”, *Comput. Vis. Image Underst.*, v. 112, n. 1, pp. 81–90, 2008.
- [16] MARFIL, R., MOLINA-TANCO, L., BANDERA, A., et al. “Pyramid segmentation algorithms revisited”, *Pattern Recogn.*, v. 39, n. 8, pp. 1430–1451, 2006.
- [17] FALCÃO, A. X., LOTUFO, R. A., ARAUJO, G. *The Image Foresting Transformation*. Relatório Técnico IC-00-12, Institute of Computing, University of Campinas, Campinas, jul. 2000.
- [18] KASS, M., WITKIN, A., TERZOPOULOS, D. “Snakes: Active contour models”, *International Journal of Computer Vision*, v. 1, n. 4, pp. 321–331, jan. 1988.
- [19] COHEN, L. D. “On Active Contours Models and Balloons”. In: *CVGIP: Image Understand*, n. 53, pp. 211–218, 1991.
- [20] XU, C., PRINCE, J. L. “Gradient Vector Flow: New External Force for Snakes”, *IEEE Proc. Conference on Computer Vision Pattern Recognition*, v. 7, n. 3, pp. 359–369, 1998.

- [21] AMINI, A. A., WEYMOUTH, T. E., JAIN, R. C. “Using dynamic programming for solving variational problems in vision”. In: *IEEE Trans. on Pattern Analysis and Machine Intel*, v. 12, pp. 855–867, 1990.
- [22] BULPITT, A. J., EFFORD, N. D. “An efficient 3D deformable model with self-optimising mesh”, *Image and Vision Computing*, v. 14, n. 8, pp. 573–580, ago. 1996.
- [23] MCINERNEY, T., TERZOPOULOS, D. “Topologically Adaptable Snakes”. In: *Medical Image Analysis*, pp. 840–845, jun. 1995.
- [24] CASELLES, V., KIMMEL, R., SAPIRO, G. “Geodesic Active Contours”, *International Journal of Computer Vision*, v. 1, n. 22, pp. 61–79, fev. 1997.
- [25] SETHIAN, J. A. “Numerical Methods for Propagating Fronts.” , 1987. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.46.2265&rep=rep1&type=pdf>>. Acesso em: 24 de abril de 2010, 15:00:00.
- [26] SETHIAN, J. A., VLADIMIRSKY, A. “Fast methods for the Eikonal and related Hamilton-Jacobi equations on unstructured meshes”. In: *Proceedings of the National Academy of Sciences of the United States of America*, v. 97, pp. 5699–5703, maio 2000.
- [27] MALLADI, R., SETHIAN, J. A., VEMURI, B. C. “Shape modeling with front propagation: A level set approach”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 17, n. 2, 1995.
- [28] HILL, A., TAYLOR, C. J. “Automatic landmark generation for Point Distribution Models”. In: *Proceedings of the conference on British machine vision*, v. 2, pp. 429–438, York, out. 1994.
- [29] CHAN, H., WEI, D., HELVIE, M. A., et al. “Computer-aided classification of mammographic masses and normal tissue: linear discriminant analysis in texture feature space”, *Physics in Medicine and Biology*, v. 40, pp. 857–876, maio 1995.
- [30] SENDA, S., MINOH, M., IKEDA, K. “A fast algorithm for the minimum distance classifier and its application to Kanji character recognition”. In: *Proceedings of the Third International Conference on Document Analysis and Recognition*, v. 1, pp. 283–286, Montreal, ago. 1995.
- [31] HOU, D. B., YU, D., ZHOU, Z. K. “Research on the Method of Neural Network Modeling Based on FCM Algorithm and Its Application on Vision-based

- Sensors”. In: *Progress In Electromagnetics Research Symposium*, v. 1, pp. 602–605, Hangzhou, ago. 2005.
- [32] MORA, A. M., FERNANDES, C. M., MERELO, J. J., et al. “KohonAnts: A Self-Organizing Ant Algorithm for Clustering and Pattern Classification”. In: Bullock, S., Noble, J., Watson, R., et al. (Eds.), *Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*, pp. 428–435, Southampton, jun. 2008.
- [33] RAVISHANKAR, S., JAIN, A., MITTAL, A. “Multi-stage contour based detection of deformable objects”. In: *Proceedings of the 10th European Conference on Computer Vision*, pp. 483–496, Marseille, out. 2008.
- [34] ZHU, Q., WANG, L. M., WU, Y., et al. “Contour context selection for object detection: A set-to-set contour matching approach”. In: *Proceedings of the 10th European Conference on Computer Vision*, pp. 774–787, Marseille, out. 2008.
- [35] CHARPIAT, G., FAUGERAS, O., KERIVEN, R. “Approximations of shape metrics”, *Foundations of Computational Mathematics*, v. 5, pp. 1–58, fev. 2005.
- [36] ROUSSON, M., PARAGIOS, N. “Shape Priors for Level Set Representations”. In: *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part II*, pp. 78–92, London, UK, 2002.
- [37] LEVENTON, M. E., GRIMSON, W. E. L., FAUGERAS, O. “Statistical shape influence in geodesic active contours”. In: *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, v. 1, pp. 316–323, ago. 2002.
- [38] BROX, T., ROUSSON, M., DERICHE, R., et al. “Unsupervised Segmentation Incorporating Colour, Texture and Motion”. In: *Computer Analysis of Images and Patterns, volume 2756 of Lecture Notes in Computer Science*, pp. 353–360. Springer, ago. 2003.
- [39] PARAGIOS, N., DERICHE, R. “Geodesic Active Regions and Level Set Methods for Supervised Texture Segmentation”, *International Journal of Computer Vision*, v. 46, pp. 223–247, 2002.
- [40] BRESENHAM, J. E. “Algorithm for computer control of a digital plotter”, *IBM Systems Journal*, v. 4, n. 1, pp. 25–30, jan. 1965.