



UNIVERSIDADE DO ALGARVE

Unidade de Ciências Exactas e Humanas

**SEGMENTAÇÃO DE IMAGEM
EM TRÊS DIMENSÕES**

João Miguel Fernandes Rodrigues
Licenciado em Engenharia Electrotécnica

**Dissertação para a obtenção do grau de Mestre
em Engenharia de Sistemas e Computação**

Faro, 1998

Dissertação realizada sob a orientação do
Prof. Doutor Johannes Martinus Hubertina du Buf
Professor Associado Convidado da
Unidade de Ciências Exactas e Humanas da
Universidade do Algarve

Declaro sob compromisso de honra que a presente dissertação é original.

Aos meus pais.

AGRADECIMENTOS

A realização de uma tese só é possível se o ambiente que nos rodeia é caracterizado por factores humanos, culturais e materiais favoráveis. O desenvolvimento da presente tese foi efectuado com as características especificadas, e por esse motivo quero expressar os meus agradecimentos:

Ao meu Orientador, Professor Doutor Hans du Buf, por ter tornado possível a realização desta tese, pela colaboração, críticas, sugestões e apoio sempre demonstrado.

Ao Dr. Stavri Nikolov e Dr. Eddy Loke, pela disponibilização das imagens pertencentes à base de dados ISACS.

À minha namorada, Mestre Célia Ramos, pela ajuda e sugestões na fase de leitura e revisão do texto.

À Escola Superior de Tecnologia, principalmente à Área Departamental de Engenharia Electrotécnica, pela oportunidade e facilidades que me concedeu para a realização da tese.

À Universidade do Algarve - Unidade de Ciências Exactas e Humanas, Área Departamental de Engenharia Electrónica e Computação, pelas facilidades que me concedeu para a realização da tese.

À minha família, amigos e colegas, pela paciência, apoio e compreensão sempre demonstrada.

Nota: Esta Dissertação foi elaborada dentro do programa de formação avançada para o ensino superior - PRODEP - Acção 5.2

ABSTRACT

This thesis presents a segmentation algorithm in three dimensions (3D). The algorithm uses an octree pyramid in which the noise in the feature space that causes class-overlap is reduced at the expense of the image resolution. Hence, at a coarse representation level the separation between classes is increased whereas the within-class variance is reduced. At this level a clustering algorithm is applied without any connectivity constraints, yielding the prototypes of the classes and creating the image of labels by means of a classification. Isolated voxels and/or insignificant classes are removed by assigning them to their neighbours, assuring the spatial connectivity. The spatial resolution is then increased by the down-projection of the labels, level by level. At each level the uncertainty of the classification of the boundary voxels is minimised by using a dynamic re-allocation of these as a function of the boundary orientation. The classification of the others boundary voxels to the nearest class prototypes is done by means of an orientated-adaptive 3D Butterfly filter in several directions. The algorithm is tested using different data sets, amongst others seafloor sonar images.

KEYWORDS

Image Segmentation in 3D; Segmentation of seafloor images in 3D; Image Segmentation in 2D; Segmentation of seafloor images in 2D; Boundary Refinement; Quadtree Pyramids; Octree Pyramids.

SUMÁRIO

Esta dissertação apresenta um algoritmo de segmentação de imagem em três dimensões (3D). O algoritmo faz uso de uma pirâmide baseada numa árvore octógona onde a presença de ruído no espaço de atributos, que causa sobreposição de classes, vai ser reduzida à custa da resolução da imagem. Num nível alto da pirâmide é obtida uma maior separação entre as diferentes classes, e a variância dentro destas é reduzida. Neste nível é aplicado um algoritmo de agrupamento sem restrições de conectividade, obtendo-se os protótipos das classes e a respectiva imagem de etiquetas. No entanto, podem surgir alguns voxéis mal classificados (isolados) e/ou classes insignificantes. Estes voxéis e classes são atribuídos às classes vizinhas por um processo determinístico de uma única passagem, o qual garante a conectividade espacial. A resolução é melhorada pela projecção para baixo das etiquetas. Para cada nível (que se desce) surge a incerteza na classificação dos voxéis fronteira. Esta é minimizada por uma selecção dinâmica e por uma re-classificação dos voxéis pertencentes à fronteira em função da direcção desta, estes são atribuídos aos protótipos das classes mais próximos fazendo para tal uso de filtros Borboleta 3D em diferentes direcções. O algoritmo é testado para uma grande variedade de imagens onde se salientam as imagens submarinas obtidas através de sonar.

PALAVRAS CHAVE

Segmentação de Imagem em 3D; Segmentação de imagens submarinas em 3D; Segmentação de Imagem em 2D; Segmentação de imagens submarinas em 2D; Refinamento da fronteira; Árvore Quaternária; Árvore Octógona; Pirâmide.

ÍNDICE GERAL

DEDICATÓRIA.....	I
AGRADECIMENTOS	II
ABSTRACT.....	III
KEYWORDS.....	III
SUMÁRIO	IV
PALAVRAS CHAVE.....	IV
ÍNDICE GERAL	V
LISTA DE ABREVIATURAS	VIII
CAPÍTULO 1 - INTRODUÇÃO.....	1
1.1- ENQUADRAMENTO DO TRABALHO	1
1.2 – OBJECTIVOS	2
1.2.1 – <i>Ligação ao Projecto ISACS</i>	3
1.3 - ORGANIZAÇÃO DOS CAPÍTULOS	4
CAPÍTULO 2 - UMA APROXIMAÇÃO À SEGMENTAÇÃO DE IMAGEM ...	6
2.1 - INTRODUÇÃO	6
2.2 - SEGMENTAÇÃO DE IMAGEM USANDO ÁRVORE QUATERNÁRIA QUE COMBINA INFORMAÇÃO ESTATÍSTICA E ESPACIAL.	9
2.2.1 – <i>Árvore Quaternária</i>	10
2.2.2 - <i>Classificação</i>	12
2.2.2.1- <i>Agrupamento Centróide Local</i>	12
2.2.3 – <i>Estimação da Fronteira</i>	16
2.3 - SEGMENTAÇÃO HIERARQUIZADA PELO USO DE AGRUPAMENTO MULTI- ATRIBUTOS E REFINAMENTO DE FRONTEIRA COM ORIENTAÇÃO ADAPTATIVA.....	17
2.3.1 – <i>Restauração do Espaço de Continuidade</i>	19
2.3.2 – <i>Estimação da Fronteira</i>	22
2.3.2.1 - <i>Sumário do Algoritmo da Estimação da Fronteira</i>	24

2.4 – DISCUSSÃO DOS ALGORITMOS PROPOSTOS	25
CAPÍTULO 3 - IMPLEMENTAÇÃO DA SEGMENTAÇÃO EM DUAS DIMENSÕES.....	29
3.1 - INTRODUÇÃO	29
3.2 - IMPLEMENTAÇÃO DO ALGORITMO DE SEGMENTAÇÃO EM 2D.....	29
3.2.1 - <i>Construção da Pirâmide</i>	29
3.2.2 - <i>Algoritmo de Aglomeração</i>	30
3.2.3 - <i>Espaço de Continuidade</i>	32
3.2.4 - <i>Estimação das Fronteiras</i>	34
3.3 – CRÍTICA AOS RESULTADOS EM 2D.....	38
3.4 – CONCLUSÕES SOBRE A SEGMENTAÇÃO EM 2D	40
CAPÍTULO 4 - SEGMENTAÇÃO DE IMAGEM EM TRÊS DIMENSÕES	42
4.1 – INTRODUÇÃO	42
4.2 – DESCRIÇÃO DO ALGORITMO EM 3D.....	42
4.2.1 - <i>Árvore Octógona</i>	43
4.2.2 - <i>Classificação</i>	44
4.2.2.1 - <i>Aglomeração das Classes</i>	44
4.2.2.2 - <i>Restauração do Espaço de Continuidade</i>	44
4.2.3 - <i>Estimação da Fronteira</i>	47
4.2.3.1 - <i>Restrição dos Voxeis Fronteira</i>	52
4.2.3.2 - <i>Restrição nas Direcções do Filtro Borboleta 3D</i>	54
4.3 – DESCRIÇÃO SUMÁRIA DO ALGORITMO EM 3D.....	55
4.4 – TESTES E RESULTADOS EM 3D	57
4.4.1 - <i>Análise de uma Imagem Sintética</i>	57
4.4.2 – <i>Resultados Obtidos com o Volume de Dados do Projecto ISACS</i>	65
CAPÍTULO 5 - CONCLUSÕES.....	71
5.1 – ANÁLISE CRÍTICA DO TRABALHO.....	71
5.2 – CONCLUSÕES	74
5.3 – TRABALHO FUTURO	76

APÊNDICES

APÊNDICE A – RESULTADOS DA SEGMENTAÇÃO EM DUAS DIMENSÕES.	77
APÊNDICE B – GRÁFICOS DE TESTES EM TRÊS DIMENSÕES.	79
APÊNDICE C – RESULTADOS DA SEGMENTAÇÃO SOBRE O VOLUME DE DADOS DO PROJECTO ISACS	83
ANEXOS	
ANEXO A–FILTRO ADAPTATIVO PONDERADO GAUSSIANO	87
BIBLIOGRAFIA.....	88

LISTA DE ABREVIATURAS

1D - Uma dimensão.

2D - Duas dimensões.

3D - Três dimensões.

AGWF - *Adaptive Gaussian Weighted Filter* (Filtro Adaptativo Ponderado Gaussiano).

CPU - *Central Processing Unit* (Unidade Central de Processamento).

d.p. - Desvio padrão.

GPS - *Global Positioning System* (Sistema de Posicionamento Global).

ISACS - *Integrated System for Analysis and Characterization of the Seafloor* (Sistema Integrado para a Análise e Caracterização de Fundos Marinhos).

MAST - *Marine Science and Technology Programme* (Programa Marinho da Ciência e Tecnologia).

PC – *Personal Computer* (Computador Pessoal).

v.m.c - Voxeis mal classificados.

VRML – *Virtual Reality Modeling Language* (Linguagem de Modelação de Realidade Virtual).

CAPÍTULO 1

INTRODUÇÃO

1.1- ENQUADRAMENTO DO TRABALHO

A observação de fundos oceânicos utilizando sonar, em conjunto com a computação, torna possível a visualização, a análise e a extracção de regiões e/ou objectos na forma tridimensional. A obtenção de um conhecimento correcto dos fundos oceânicos pode trazer inúmeras vantagens, entre elas, o estudo dos relevos de forma a explorar as riquezas minerais, a detecção de objectos afundados (por exemplo: embarcações afundadas, cabos, etc.) e a construção de mapas para operações táticas. Estas vantagens são ainda mais interessantes se pensarmos em Portugal, país com um grande espelho de água, como a zona ideal para desenvolver e testar este tipo de ferramenta tecnicamente avançada, onde o ser humano não interage directamente com o meio, possibilitando a optimização dos recursos marinhos e humanos.

Este trabalho está inserida na área da computação e no domínio do processamento de imagem. O processamento de imagem é um termo colectivo para todas as operações que começam com uma imagem como entrada e resultam noutra imagem como saída. As ferramentas de processamento de imagem são utilizadas na compressão, realce, restauração, modelação, reconstrução e ainda na análise de imagem e visão computacional [Jain 89].

A análise de imagem é utilizada para descrever funções e operações em imagens que retornam um resultado numérico relativo a uma propriedade ou uma parte específica da imagem (objecto). Uma componente importante da análise de imagem é a automatização de análises de rotina em laboratório, bem como a celeridade na execução das mesmas. Outra, é a possibilidade de efectuar medidas quantitativas dos objectos, ajudando-nos a melhorar as nossas observações e a interpretar melhor as imagens abrindo novos caminhos, testando novas hipóteses.

Dentro da análise de imagem surge a segmentação como uma ferramenta em que a entrada é uma imagem mas a saída é uma "descrição" da imagem ou da cena [Kak 82]. A segmentação de imagem em três dimensões permite a obtenção desta descrição e possibilita a interpretação e visualização dos resultados em três dimensões, facilitando ao ser humano a sua interpretação.

1.2 – OBJECTIVOS

O objectivo deste trabalho é implementar um **algoritmo de segmentação de imagem em três dimensões**, que seja robusto e onde os custos em termos de tempo gasto de CPU sejam mínimos, mas que permita o tratamento de qualquer tipo de imagem em três dimensões, com principal ênfase para os volumes de dados submarinos obtidos a partir do projecto ISACS.

Para levar a cabo o objectivo acima referido foram definidos os seguintes sub-objectivos:

- Implementar uma estratégia de segmentação de imagem em duas dimensões já existente na literatura, que apresente bom resultados e que permita a sua extrapolação para três dimensões.

- Testar o algoritmo em duas dimensões com imagens subaquáticas obtidas do projecto ISACS.
- Extrapolar o algoritmo de segmentação em duas dimensões para três dimensões.
- Simular objectos dentro de um cubo e aplicar o algoritmo.
- Tentar otimizar o algoritmo em três dimensões relativamente à precisão da fronteira e ao tempo de cálculo.
- Testar o algoritmo em três dimensões com os dados de acústica submarina provenientes do projecto ISACS.

1.2.1 – Ligação ao Projecto ISACS

O projecto ISACS “*Integrated System for Analysis and Characterization of the Seafloor*” é financiado pela Comunidade Europeia no âmbito do programa MAST III, cujo objectivo é a análise e a classificação do fundo do mar pela exploração e correcta integração dos dados adquiridos a partir de diferentes tipos de sonares [Berntsen 97].

O trabalho apresentado nesta dissertação está inserido no projecto que está a ser realizado pelo cooperante Português *Vision Laboratory* – Universidade do Algarve, que consiste na implementação de um sistema para visualização interactiva de dados de sonar em três dimensões, baseado numa nova aproximação a volumes em três dimensões (a partir dos dados obtidos por sonar), onde a segmentação e a reconstrução estão presentes [Nikolov 97a].

A partir dos dados obtidos via sonar serão criados volumes (imagens em três dimensões), nos quais são aplicados algoritmos de segmentação, a partir dos dados gerados da segmentação são criadas funções VRML, com o objectivo de rapidamente e iterativamente visualizar e interpretar um volume de dados (obtidos utilizando

sonares). Este esquema permite ainda obter diferentes simulações, feitas a partir de dados provenientes de diferentes algoritmos (de segmentação em três dimensões), e consequente comparação entre estes, com o objectivo de determinar a melhor relação entre qualidade e tempo de cálculo (CPU). De salientar que minimizar o tempo de cálculo é muito importante, uma vez que os dados de acústica submarina em 2D/3D facilmente atingem a ordem dos Megabytes.

1.3 - ORGANIZAÇÃO DOS CAPÍTULOS

A presente dissertação, sem considerar a introdução, é composta por três partes: fundamentos teóricos, algoritmos implementados e trabalho experimental, conclusões; e por fim são apresentados: apêndices, anexo e bibliografia.

A primeira parte, **Fundamentos Teóricos**, é constituída por um capítulo:

- O capítulo 2 - **Uma aproximação à Segmentação de Imagem**, refere-se a conceitos e definições genéricas necessários para a compreensão do trabalho desenvolvido. Apresenta resumidamente o trabalho desenvolvido pelos autores Spann e Wilson, e por Schroeter e Bigün na segmentação de imagem em duas dimensões que serviu de base para o trabalho desenvolvido nesta dissertação.

A Segunda parte, **Algoritmos implementados e Trabalho Experimental**, é constituída por dois capítulos:

- O capítulo 3 – **Implementação da Segmentação em Duas Dimensões**, consiste na descrição da implementação do algoritmo de segmentação em duas dimensões proposto pelos autores anteriores, e nas alterações efectuadas com o objectivo de o tornar mais rápido. Será implementada a nova proposta do algoritmo e testada, em

conjunto com a anterior, sobre imagens subaquáticas obtidas da base de dados ISACS. Termina com a crítica dos resultados obtidos.

- O capítulo 4 - **Segmentação de Imagem em Três Dimensões**, consiste na extrapolação do algoritmo realizado em duas dimensões para as três dimensões, referindo alterações efectuadas e os testes realizados sobre imagens sintéticas com o objectivo de otimizar resultados. Finalmente, apresentam-se os resultados e críticas dos resultados obtidos pela aplicação do algoritmo ao volume de dados proveniente da base de dados do projecto ISACS.

A terceira parte, **Conclusões**, é constituída por um capítulo:

- O capítulo 5 - **Conclusões**, apresenta a discussão do trabalho executado pelo autor, as conclusões extraídas pelo mesmo e sugestões para o trabalho futuro.

Os **Apêndices** incluem imagens resultantes dos testes executados em duas e três dimensões, bem como alguns gráficos ilustrativos do comportamento do algoritmo de segmentação em três dimensões.

O **Anexo** é constituído por um algoritmo de alisamento de ruído em três dimensões que serviu de base para a implementação de um algoritmo de segmentação de imagem, que por sua vez serviu como termo de comparação com o algoritmo implementado nesta dissertação.

Finalmente a **Bibliografia**, engloba os elementos bibliográficos específicos que foram utilizada nesta dissertação.

UMA APROXIMAÇÃO À SEGMENTAÇÃO DE IMAGEM

2.1 - INTRODUÇÃO

Ao observar a figura 2.1, pode-se dividir a imagem em duas grandes regiões: uma correspondente a uma montanha (região mais clara) e outra ao fundo da imagem (região mais escura). A partir desta observação, pode-se definir a segmentação de imagem de uma maneira simples, como o processo pelo qual a computação decompõe a imagem nas suas componentes, ou seja, traduz a descrição da imagem original numa descrição de duas ou mais regiões com as suas propriedades representativas e com um conjunto de posições correspondentes aos elementos de cada região.



Fig. 2.1 – Imagem “topas”, original do projecto ISACS.

Antes de aprofundar o assunto da segmentação de imagem, serão apresentados alguns conceitos essenciais para a compreensão do tema:

O primeiro é o de **imagem digital** $I(x,y)$, como sendo uma imagem que está discretizada em coordenadas espaciais e brilho. Podemos considerar uma imagem digital como uma matriz onde a intersecção entre cada linha e cada coluna indica as coordenadas de cada ponto da imagem, o elemento da matriz identifica a cor do ponto (no exemplo corresponde a um nível de cinzento). Cada elemento individual (ponto) da matriz chamamos **pixel**.

Outro dos conceitos utilizado ao longo deste trabalho é o de **histograma**, que representa a medida de distribuição dos níveis de cinzento dos pixels numa imagem. No caso de imagens de 8 bits, os valores variam entre 0 e 255. Assim, o histograma é o somatório de todos os pixels na imagem com a mesmo nível de cinzento associado (feito para todos os níveis independentemente).

Outro conceito é o de **agrupamento**, que pode ser definido como um conjunto de valores de **I**, para os quais a gama de valores que ocorrem relativamente muitas vezes, está separada da gama de valores que aparecem poucas vezes. Intuitivamente estes agrupamentos correspondem à população natural de pixels, isto é, às **classes** naturais. No caso de um único **atributo**¹ (por exemplo: o nível de cinzento), os agrupamentos devem aparecer no histograma como picos de valores de **I**. Os picos devem estar afastados a uma distância mínima, ou então obrigar que existam vales profundos entre eles para garantir que apresentam populações distintas (classes).

Para duas imagens de atributos, por exemplo **z1** e **z2**, os agrupamentos devem aparecer como regiões densas, onde os pontos distribuídos **z1 versus z2** devem ser separados pela partição do espaço (z_1, z_2) em bocados apropriados, como pode ser

¹ Os atributos podem ser cores, propriedades das texturas, etc.

observado na figura 2.2. Este conceito pode ser generalizado para um número qualquer de imagens de atributos.

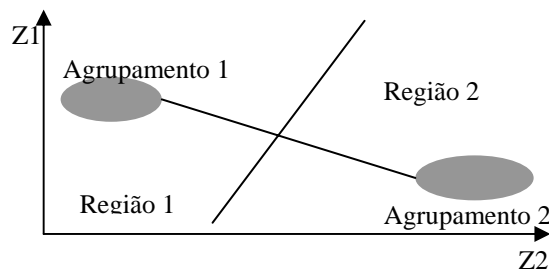


Fig 2.2 – Representação do espaço de atributos bidimensional

Ao observar de novo a figura 2.1, e ao constatar que a maioria das imagens naturais podem ser consideradas como um conjunto de regiões, com propriedades mais ou menos homogêneas, com presença de ruído em maior ou menor grau, verifica-se que dentro de cada uma das duas grandes regiões aparecem zonas em que o seu valor local é diferente do fundo e não corresponde exactamente ao valor da montanha. Ainda podem aparecer esporadicamente pontos ou pequenos grupos de pontos, mais claros ou mais escuros que as regiões onde se encontram inseridos, sugerindo a possibilidade da existência de mais do que duas regiões, ou a sobreposição de regiões. Considerando esta possibilidade e não sendo dada à priori nenhuma informação do que representa a imagem, nem da quantidade de regiões que a constitui e não querendo fazer nenhum treino do classificador para cada imagem ou grupos de imagens que serão analisadas, vai ser utilizada a **segmentação não supervisionada**.

Na segmentação não supervisionada, tanto os protótipos das regiões como as fronteiras das regiões têm de ser determinadas, estes dois pontos estão relacionados entre si da seguinte forma: protótipos de regiões bem definidas são necessários para determinar fronteiras correctas, e fronteiras correctas são necessárias para determinar

protótipos bem definidos. Os protótipos e as fronteiras não podem ser simultaneamente determinados, o que nos leva a considerá-los independentes em termos de cálculo, o que permite dizer que não faz diferença o que vai ser calculado em primeiro lugar, o protótipo ou a fronteira.

Após a apresentação destes conceitos e sabendo que não nos é dada a relação entre o sinal e o ruído ou outra informação relativamente às imagens, é necessário um algoritmo que combine uma operação de alisamento do ruído com uma segmentação não supervisionada. Um algoritmo que combina estes atributos é o proposto por Spann e Wilson [1985].

2.2 - SEGMENTAÇÃO DE IMAGEM USANDO ÁRVORE QUATERNÁRIA QUE COMBINA INFORMAÇÃO ESTATÍSTICA E ESPACIAL.

A aproximação ao problema da segmentação de imagem dada por Spann e Wilson combina um classificador não paramétrico, baseado no algoritmo de agrupamento - Agrupamento Centróide Local (que não requer qualquer tipo de informação à priori), com o alisamento do ruído feito por uma representação de imagem em árvore quaternária, seguido de uma estimação da fronteira que é feita à medida que descemos a pirâmide², a qual tem por base os segmentos obtidos nos níveis mais altos da árvore.

Este esquema é simples de implementar e pode ser resumindo em três componentes [Spann 85; Wilson 88]: **Alisamento por Árvore Quaternária, Classificação, e Estimação de Fronteira.**

² A pirâmide é construída pela sobreposição das imagens que resultam da árvore quaternária.

2.2.1 – Árvore Quaternária

O primeiro passo do algoritmo - Alisamento por árvore quaternária, consiste na concepção de uma pirâmide na forma de uma árvore quaternária. Este método baseia-se em blocos 2x2 de média dos pixels, onde o nível da árvore quaternária acima da base (de dimensões $\frac{N}{2} \times \frac{N}{2}$) consiste no conjunto de médias de blocos 2x2 de pixels não sobrepostos da imagem original, obtendo-se uma imagem com as dimensões $2^{m-1} \times 2^{m-1}$, com $m = \log_2(N)$. Este processo pode ser repetido tantas vezes até que o nível principal seja alcançado e em que o pico da pirâmide tenha um só pixel cujo valor é o valor médio da imagem inteira.

Para especificar melhor esta fase considere-se uma imagem³ (NxN) definida por $I(i,j)$ para $0 \leq i, j < N$, onde $N=2^m$. A árvore quaternária desta imagem é definida como

$$q(i, j; l) = \frac{1}{4} \sum_{a=0}^1 \sum_{b=0}^1 q(2i+a, 2j+b; l-1), \quad (2.1)$$

onde $0 < l \leq m$; $0 \leq i, j < 2^{m-l}$ e $q(i, j; 0) = I(i, j)$.

As árvores quaternárias são um método rápido de alisamento da imagem, produzem uma troca conveniente entre a redução da quantidade do ruído e a influência da junção de dados de regiões distintas para o posterior cálculo da fronteira. A truncagem no nível $m' < m$ possibilita obter um ganho máximo de alisamento, enquanto se mantém uma resolução suficiente para garantir uma segmentação correcta para todas as regiões cujo raio é $r \geq 2^{m'+1}$.

³ Não existe a obrigatoriedade da imagem ser quadrada.

Para exemplificar esta afirmação, considere-se duas regiões com médias μ_1 e μ_2 com ruído branco Gaussiano de variância σ^2 . Para simplificar, assume-se que uma região é um objecto convexo e que tem um raio $r = 2^n$, com $n > m$. Seja p_l a probabilidade de que o nó⁴ (pixel) no nível l tenha filhos com média μ_1 , então o valor esperado m_l e a variância σ_l^2 no nível l podem ser escritos a partir de (2.1) como

$$m_l = E[q(i, j; l)] = p_l(\mu_1 - \mu_2) + \mu_2, \quad l > 0, \quad (2.2)$$

$$\sigma_l^2 = 2^{-2l} \sigma^2. \quad (2.3)$$

A relação sinal ruído ρ_l no nível l é definido por

$$\rho_l = \frac{(m_l - \mu_2)}{\sigma_l}, \quad (2.4)$$

$$\rho_0 = \rho = \frac{\mu_1 - \mu_2}{\sigma}. \quad (2.5)$$

Substituindo agora (2.2), (2.3) e (2.5) em (2.4) obtemos

$$\rho_l = 2^l p_l \rho. \quad (2.6)$$

Para objectos convexos existe uma aproximação para p_l , que é válida nas condições referidas, pela verificação de que nesses objectos (círculos, quadrados, etc.), a razão entre o perímetro e a área é dada por $2/r$ (onde r é o raio do objecto), uma vez que os nós do perímetro contêm uma média de 50% dos pixels dentro do objecto, segue-se que p_l pode ser aproximado por:

$$p_l = 1 - \frac{2^l}{r} = 1 - 2^{l-n}. \quad (2.7)$$

⁴ A cada nó no nível l correspondem 4 filhos no nível $l-1$.

Para encontrar o nível apropriado de alisamento é necessário encontrar m' , para tal, partimos de (2.7) e (2.6) de forma que $\rho_{m'+1} < \rho_{m'}$, e vamos obter para m' o valor de $n-1$, que dará um ganho de alisamento $\rho_{m'} = 2^{m'-1} \rho$. Assim, verifica-se que o nível óptimo de alisamento depende apenas do tamanho da região, e não da razão entre o sinal e o ruído, sendo esta uma das premissas para a realização do presente trabalho.

2.2.2 - Classificação

O segundo passo do algoritmo - Classificação, resolve o problema da segmentação não supervisionada. A imagem proveniente do nível m da árvore quaternária vai ser classificada utilizando um algoritmo não paramétrico que tem por base as estatísticas dos níveis de cinzento. O algoritmo escolhido é o Agrupamento Centróide Local.

2.2.2.1- Agrupamento Centróide Local

Para definir o centróide local [Wilson 88], considere a densidade de probabilidade $p(x)$. O centróide local é definido em cada ponto x por:

$$\mu(x) = x + \frac{\int_{-w}^w x' p(x+x') dx'}{\int_{-w}^w p(x+x') dx'} . \quad (2.8)$$

Esta equação mostra que o centróide local no ponto x é precisamente o centro de massa da distribuição de probabilidade calculado sobre a janela de tamanho $2w$ centrada em x .

A equação anterior pode ser estendida para a forma discreta pela troca de integrais por somatórios. Em situações práticas, apenas estão disponíveis as estimações da densidade de probabilidade na forma de histograma.

No caso de densidades de probabilidade (globais) que consistem na soma de um conjunto de densidades (locais não sobrepostas), pode ser mostrado que a definição de centróide local preserva a média (da classe local). Para esse fim, considere-se a densidade

$$p(x) = \sum_i p_i(x), \quad (2.9)$$

e define-se a média da classe local μ_i como

$$\mu_i = \frac{\int_{\mu_i-w}^{\mu_i+w} xp_i(x)dx}{\int_{\mu_i-w}^{\mu_i+w} p_i(x)dx}. \quad (2.10)$$

Neste caso a densidade $p_i(x)$ é definida por

$$p_i(x) = 0 \quad \mu_i - \frac{w}{2} < x < \mu_i + \frac{w}{2}, \quad (2.11)$$

e $|\mu_i - \mu_j| > 2w$ para todos os $i \neq j$. A partir da equação (2.8) o centróide local no ponto x_0 para uma janela de tamanho $2w$ é dado por:

$$\mu(x_0) = x_0 + \frac{\int_{-w}^w x' p(x_0 + x') dx'}{\int_{-w}^w p(x_0 + x') dx'}, \quad (2.12)$$

onde $|x_0 - \mu_i| < w$ para alguns i . Definindo a nova variável x'' como

$$x'' = x' - (\mu_i - x_0), \quad (2.13)$$

a equação torna-se

$$\mu(x_0) = \mu_i + \frac{\int_{-w-\mu_i+x_0}^{w-\mu_i+x_0} x'' p(x''+\mu_i) dx''}{\int_{-w-\mu_i+x_0}^{w-\mu_i+x_0} p(x''+\mu_i) dx''} = \mu_i. \quad (2.14)$$

Assim, ao mover cada probabilidade (de massa) $p(x)$ para o seu centróide local na classe espacial, é produzida uma nova densidade:

$$p'(x) = \sum_{i=1}^N n_i \delta(x - \mu_i), \quad (2.15)$$

onde n_i é a área sob cada densidade local $p_i(x)$ e $\delta(x - \mu_i)$ é a função Dirac. A nova classificação é uma em que cada ponto é atribuído a classe definida pela média da classe da distribuição sobre a qual está localizado.

Em geral, a distribuição de probabilidade não pode ser escrita como a soma de um conjunto de distribuições locais não sobrepostas. No entanto, usando a ideia atrás referida, podemos construir um esquema iterativo que resulta na distribuição dada na equação (2.15).

Consideremos então o seguinte esquema iterativo, onde $h(x)$ é o valor do histograma na posição x da classe espacial. O algoritmo funciona pela contínua actualização do histograma, movendo as massas de probabilidade para a posição do seu centróide local até à inexistência de alterações no histograma (critério de paragem). Seja $h^n(x)$ o histograma actualizado na iteração n , o algoritmo é estruturado da seguinte forma [Wilson 88]:

1. $n:=1$

2. $h^0(x) := h(x)$

3. $h^n(x) := \sum_{y \in B(x)} h^{n-1}(y)$, com $y \in B(x)$ sse $\mu^n(y) = x$,

$$\text{onde } \mu^n(y) = y + \frac{\sum_{y'=-m}^m y' h^{n-1}(y + y')}{\sum_{y'=-m}^m h^{n-1}(y + y')}$$

4. $h^n(x) = h^{n-1}(x) ? \quad \forall x$

Não: $n=n+1$ voltar para 3)

Sim: actualizar o tamanho da janela; voltar para 1)

Neste caso, o centróide local é calculado dentro de uma janela de tamanho $2m+1$. O histograma final é dado por:

$$h^n(x) = \sum_{j=1}^{N_i} n_j \delta_{xx_j}, \quad (2.16)$$

com $\min_{i,j} |\mu_i - \mu_j| > 2m+1$, terminando esta distribuição, o algoritmo mantém-se inalterado quando se efectua mais iterações.

Na equação anterior, N_i é o número de classes encontradas e os valores de x_j são dados aproximadamente pela expressão (2.17), onde $x \in C_j$ se x for classificado na classe j .

$$x_j \approx \frac{\sum_{x \in C_j} x h(x)}{\sum_{x \in C_j} h(x)} \quad (2.17)$$

A convergência deste algoritmo é difícil de determinar, mas na prática foi observado que converge sempre com um pequeno número de iterações (10-20), sendo o número de classes encontradas dependente do tamanho da janela e dos picos do histograma

original. No entanto, esta dependência é retirada pela passagem do algoritmo por um conjunto de janelas que são aumentadas de cinco pontos em cada passagem, o resultado só é aceite quando sucessivas passagens produzem resultados coerentes.

De modo a classificar os dados no fim do algoritmo, os autores Spann e Wilson recomendam que numa tabela seja registado o movimento ao longo das iterações. Se x for um ponto no histograma inicial $h(x)$, então a classificação final é dada por $l^n(x)$, onde $l^0(x)=x$ e $l^j(x)=l^{j-1}(y)$, para qualquer que seja o y tal que $x=\mu^j(y)$, $j=1, n$.

No princípio da iteração p , a tabela $l^p(x)$ representa a nova posição das probabilidades (de massa) originalmente na posição x da classe espacial, assim a classificação final é obtida pela indexação dentro da tabela recorrendo ao valor mais alto da árvore quaternária $q(i, j; l)$.

2.2.3 – Estimação da Fronteira

Como foi mencionado anteriormente, o alisamento pela árvore quaternária é uma forma de troca de resolução da classe com resolução espacial. Cada ponto da fronteira no nível mais alto da árvore, depois do procedimento de agrupamento e classificação, define blocos de pixels de $L \times L$ no nível mais baixo da árvore (maior resolução espacial), com $L=2^l$ e sendo l a altura da árvore quaternária. Neste ponto, o problema passa a ser como obter de novo a resolução espacial inicial.

A solução só pode ser encontrada fazendo uma consideração adicional, isto é que a classificação introduzida no nível mais alto da pirâmide é válida nos níveis mais baixos. Uma vez definida a região fronteira, os nós não pertencentes à região fronteira

são atribuídos as etiquetas do seu pai, os nós na região fronteira são classificados de tal forma que a largura da região fronteira é reduzida por um factor de 2 em cada passo para baixo da árvore quaternária. O resultado é a fronteira entre regiões no nível mais baixo da pirâmide e assim é restaurada a resolução espacial. Uma descrição mais precisa da estimação da fronteira pode ser encontrada em [Wilson 88].

Uma inovação e melhoria para este algoritmo foi realizada principalmente ao nível da estimação das fronteiras por Schroeter e Bigün [1995] e irá ser analisada na próxima secção.

2.3 SEGMENTAÇÃO HIERARQUIZADA PELO USO DE AGRUPAMENTO MULTI-ATRIBUTOS E REFINAMENTO DE FRONTEIRA COM ORIENTAÇÃO ADAPTATIVA

O trabalho executado por Schroeter e Bigün [1995] apresenta uma segmentação de imagem multi-atributo baseada na estrutura: **1) árvore quaternária**, **2) classificação** e **3) estimação da fronteira**, como executada por Spann e Wilson, mas onde a estimação da fronteira é efectuada por um processo orientativo, embebido na estrutura da pirâmide e onde é permitido o uso de diferentes algoritmos de agrupamento sem restrições de conectividade espacial.

O algoritmo pode ser descrito como a construção, para todos os atributos (textura, cores, etc.), de uma árvore quaternária até um nível predefinido, em que cada nível superior é obtido pelo alisamento do nível antecedente. O próximo passo é a partição do espaço de atributos em classes C . Esta operação pode ser vista como encontrar C subconjuntos do vector de atributos à volta dos respectivos protótipos (ou centros de classe; centróides) sobre o espaço de atributos alisados. Foi mostrado pelos autores que a escolha do algoritmo de agrupamento não é crítica e que resultados semelhantes

da segmentação são obtidos por diferentes algoritmos (como *fuzzy c-means*, *sum of squares*, *centroid* e *complete link*), e ainda que o algoritmo funciona correctamente num espaço de atributos unidimensional. A seguir, classes pequenas bem como pixels isolados são distribuídas para a classe mais próxima, usando uma vizinhança (de oito) que garante a conectividade espacial – **restauração do espaço de continuidade**.

O último passo é o procedimento de **estimação da fronteira** que gradualmente melhora a resolução espacial. Primeiro, a fronteira é identificada no nível onde o algoritmo de agrupamento é executado e os filhos dos nós da região fronteira definem a região fronteira no nível de resolução acima. Aos nós não pertencentes à fronteira no nível dos filhos é dada a mesma etiqueta e propriedades que o nó do seu pai.

A incerteza das classes dentro da região fronteira é grande e tem de ser reduzida antes de reatribuir os elementos da fronteira, os autores propõem-se fazer isso pelo uso de filtros adaptativos orientados. Para cada direcção local dominante calculada, um filtro Borboleta (*Butterfly*) é definido. Esta é a forma utilizada para reduzir a influência nos atributos ao longo da fronteira. Para cada pixel da fronteira, os dois alvos (asas da Borboleta) do filtro correspondente à orientação local são aplicados separadamente, produzindo duas respostas distintas entre cada alvo e as duas classes associadas que definem a fronteira. O valor do pixel da fronteira é atribuído à classe mais próxima. A seguir será apresentada a descrição detalhada do processo de restauração do espaço de continuidade e da estimação da fronteira.

2.3.1 – Restauração do espaço de continuidade

Após a conclusão do processo de agrupamento, cada vector pertencente à mesma classe recebe uma etiqueta γ_i ($1 \leq i \leq C$), assim no nível mais baixo de resolução l_{\max} podemos definir uma imagem de etiquetas $I\gamma(i, j; l_{\max})$.

Neste nível podem ser observadas fronteiras grosseiras entre diferentes classes, deve também ser notado que alguns pixels estão espacialmente mal classificados devido a que nenhuma connectividade espacial ter sido utilizada nos algoritmos de agrupamento e classificação.

Seja $N_8(i, j)$ a vizinhança de um pixel na localização (i, j) composta pelos seus 8 vizinhos mais próximos, $N_8(i, j) = \{(i + u, j + v)\}$, $-1 \leq u, v \leq 1$ e $(u, v) \neq (0,0)$. Um pixel (i, j) é considerado como espacialmente mal classificado se $I\gamma(i, j; l_{\max})$ é diferente de todas as etiquetas em $N_8(i, j)$, neste caso é atribuído à classe mais representativa em $N_8(i, j)$.

Classes pequenas ou perdidas, designadas por **classes insignificantes**, são atribuídas aos seus vizinhos. Uma classe no espaço de atributos é distribuída a uma ou mais sub-regiões em $I\gamma(i, j; l_{\max})$ e é considerada **significativa** apenas quando a sua maior sub-região contém um número suficiente de pixels conectados, assim a preferência é dada às classes que estão espacialmente distribuídas às regiões maiores e mais compactas.

Existe ainda o conceito de **região insignificante** que está relacionado com o da classe. A classe pode ser considerada significativa, mas ainda pode existir na imagem

regiões insignificativas dessa mesma classe. Essas regiões são determinadas (insignificantes) e atribuídas como se de classes (insignificantes) se tratassem. Vai-se referir ao longo deste trabalho estes dois conceitos como se de um só se tratasse, e vai-se utilizar o termo classe/região no texto que mais facilite a leitura e compreensão do mesmo.

Por outro lado é claro que o significado de região insignificante está directamente relacionado com a altura da pirâmide, ou seja, para um valor alto de l_{\max} o tamanho da região insignificante devera ser mais pequeno do que para um valor mais baixo de

l_{\max} .

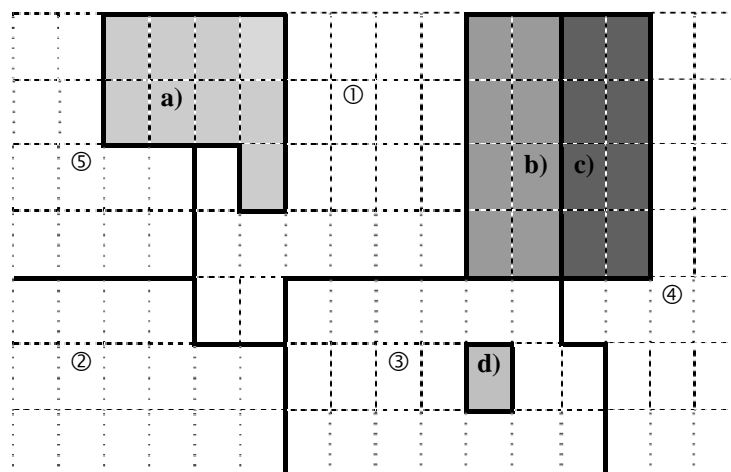


Fig 2.3 – Classes a serem realocadas em 2D.

Na figura 2.3, está representado um excerto de uma imagem depois de aplicado o algoritmo de agrupamento. Foram determinadas cinco regiões distintas (numeradas de um a cinco), no entanto, surgiu a classe (pixel) d) e as classes a), b), c) (zonas sombreadas) que falharam o teste de serem significativas, tendo de ser atribuídas a uma das regiões vizinhas e no caso b) e c) entre elas próprias. No presente caso, o procedimento de atribuição é feito por um processo determinístico de multi-passagens.

Define-se **classe isolada** como uma classe insignificativa que não partilha fronteiras com outras classes insignificativas. Estas classes podem ser atribuídas num só passo, embora múltiplos passos são necessários se as classes são adjacentes uma às outras.

Em primeiro lugar, serão atribuídas as classes isoladas, como C_a constituída por n_a pixels da região a), ao conjunto das classes candidatas entre as classes vizinhas de C_a e determinadas para cada pixel $(i, j) \in C_a$. Se a etiqueta $I\gamma(i+u, j+v; l_{\max}) \in N_8(i, j)$ é diferente de $I\gamma(i, j; l_{\max})$ então uma nova classe candidata é obtida.

Caso contrário, a vizinhança é estendida e $I\gamma(i, j; l_{\max})$ é comparada com a próxima etiqueta $I\gamma(i+2u, j+2v; l_{\max})$ na mesma direcção. Estas operações são repetidas para as oito direcções possíveis, $-1 \leq u, v \leq 1$. Para cada pixel $(i, j) \in C_a$, a distância Euclidiana entre o valor correspondente (i, j) e o das diferentes classes concorrentes é calculado, a cada pixel será atribuído a classe mais próxima.

Classes **não isoladas** são atribuídas por multi-passagem, este processo pode ser ilustrado considerando as classes b) e c). No primeiro passo, estas classes são atribuídas como se fossem classes isoladas. O mesmo estado inicial de $I\gamma(i, j; l_{\max})$ é considerado para ambas classes. O máximo número de pixels $n_{i,\max}$ atribuído à mesma classe candidata é determinado para $i = b), c)$. Então a razão r_i entre $n_{i,\max}$ e a população respectiva n_i das classes é calculada, $r_i = n_{i,\max} / n_i, i = b), c)$. Se a razão $r_i = 1$ significa que a classe i tende a ser atribuída a uma única classe candidata. A atribuição da região com a maior razão r_i é então validada. Supondo que b) é atribuída primeiro antes de atribuir c), um segundo passo é ainda necessário se c) é considerada uma classe insignificativa, pois pode acontecer o caso dos pixels de b)

serem atribuídos a c) fazendo com que c) seja uma classe significativa. Se não for o caso, c) é atribuído usando o procedimento descrito para classes isoladas. Podemos generalizar o procedimento anterior para mais de duas classes não isoladas, ocorrendo mais passagens até que todas as classes insignificantes sejam atribuídas.

2.3.2 – Estimação da Fronteira

O último passo no algoritmo é a melhoria das fronteiras, para melhorar a resolução espacial das imagens etiquetadas $I\gamma(i, j; 0 \leq l \leq l_{\max})$. Como referido anteriormente, protótipos correctos podem ser obtidos apenas à custa de resolução espacial, assim no nível mais alto a incerteza que afecta as fronteiras é grande. Propõe-se diminuir essa incerteza usando para tal filtros adaptativos orientados.

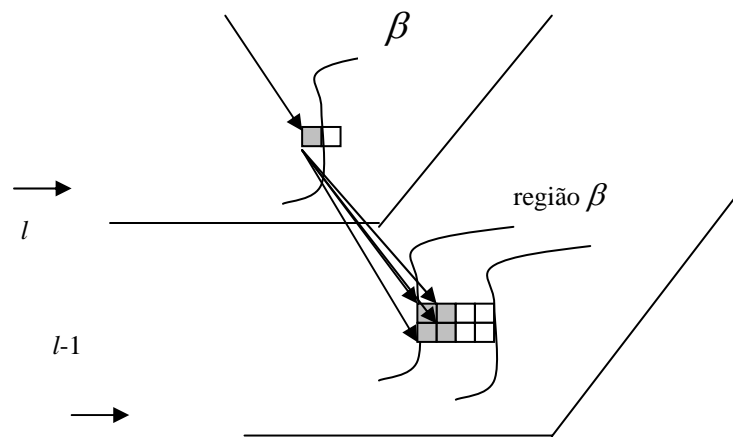


Fig 2.4 - Região fronteira β em 2D.

Primeiro no nível mais grosseiro l_{\max} os pixels da fronteira são determinados, cada pixel (i, j) é considerado como um pixel da fronteira se pelo menos uma etiqueta em $N_8(i, j)$ é diferente de $I\gamma(i, j; l_{\max})$. Depois $I\gamma(i, j; l_{\max}-1)$ é obtida pela projecção em sentido decrescente da etiqueta de cada nó pai não-fronteiro para os seus respectivos 4 nós filhos, $I(i, j; l) = I(i/2, j/2; l+1)$, sendo $/$ o inteiro da divisão. Os

filhos na fronteira definem a região β na qual vai ser efectuado o alisamento orientado (fig. 2.4).

Para alisar, é necessário estimar a orientação da fronteira local e esta é determinada por um número complexo

$$z_1 = (\nabla I_\gamma)^2 * m, \tag{2.18}$$

com ∇I_γ o gradiente da imagem das etiquetas complexa $\left(\frac{\partial I_\gamma}{\partial x}\right) + i\left(\frac{\partial I_\gamma}{\partial y}\right)$ e $*m$ a convolução com o filtro de média. A partir do argumento de z_1 é obtida a orientação dominante [Schroeter 95]. No presente caso, o módulo do gradiente é 1 na transição entre duas classes e 0 dentro das classes. A função de alisamento m (eq. 2.19) é Gaussiana de tamanho 7×7 (sxs), onde $-s/2 \leq x, y \leq s/2$ e σ é determinado pela relação $(s/2 + 1)^2 = 6\sigma^2$.

$$m(x, y) = \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \tag{2.19}$$

Para cada orientação local dominante é definido um filtro Borboleta (fig. 2.5), onde r é função da dissemelhança entre as duas classes que definem a fronteira e $rr=(1-r)/n$, com n o número de pesos da máscara diferentes de 0 e r .

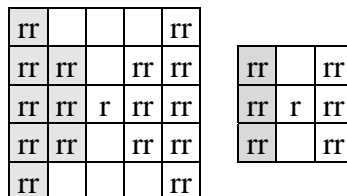


Fig. 2.5 – Filtro Borboleta em 2D (orientação horizontal, máscara 5x5 e 3x3)

A função dissemelhança é dada por

$$d = \frac{|\mu_1 - \mu_2|}{\sqrt{\sigma_1^2 + \sigma_2^2}}, \quad (2.20)$$

onde μ e σ^2 são a média e a variância de duas classes em ambos os lados da fronteira. Com $r = \text{fct}(d)$ retirado de uma função obtida experimentalmente inicialmente pelos autores Spann e Wilson (fig. 2.6).

É necessário ainda definir o filtro nas diferentes direcções Θ , isto é feito pela rotação da máscara da figura 2.6 ($\Theta=0$) e por redistribuição dos pesos de forma a corresponderem à grelha da imagem [Schroeter 95].

Os filtros são calculados para um número fixo de orientações e guardados numa tabela. Foi referido por Schroeter e Bigün [1995] que tipicamente são obtidos resultados semelhantes de segmentação para diferentes números de orientações (6 a 8) partindo do mesmo nível l_{\max} .

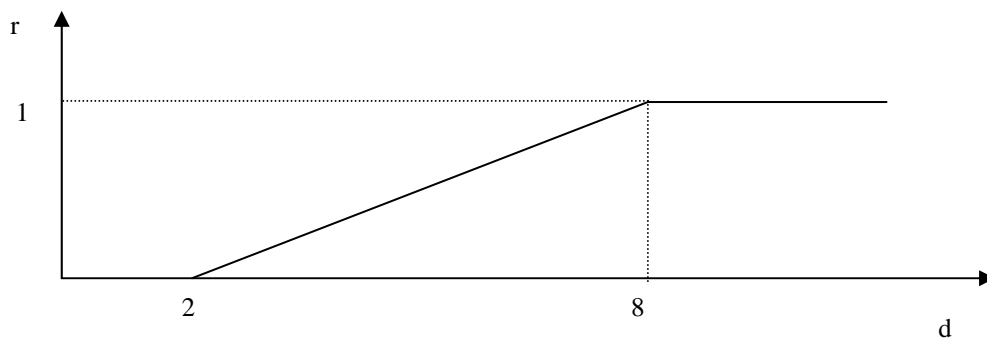


Fig. 2.6 – Função para o cálculo de r

2.3.2.1 Sumário do Algoritmo da Estimação da Fronteira

a) Projectar para baixo as etiquetas, $I(i, j; l-1) = I(i/2, j/2; l)$ e definir a região de fronteira β no nível $l-1$. Calcular a média μ_i e a variância σ_i^2 para todas as classes $2 \leq i \leq C$ no nível l .

b) Para cada pixel fronteiro no nível l calcular a orientação dominante usando as equações: $z_1 = (\nabla I_\gamma)^2 * m$, determinar as duas classes C_1 e C_2 em ambos os lados da

fronteira, calcular os pesos dos filtros usando $d = \frac{|\mu_1 - \mu_2|}{\sqrt{\sigma_1^2 + \sigma_2^2}}$; $r = \text{fct}(d)$ e $rr = (1-r)/n$ e

depois propagar as orientações para os pixels correspondentes no nível $l-1$.

c) Se um dado pixel $(i, j) \notin \beta$ (v.d. fig. 2.4), significa que já pertence a uma classe, atribuir-se o seu valor à sua classe correspondente. Para cada pixel $(i, j) \in \beta$, aplicar na imagem de atributos, no nível $l-1$, o filtro Borboleta correspondente à orientação local dominante. As metades esquerda e direita do filtro (alvos) são aplicadas separadamente, eliminando o problema de alisamento sobre as fronteiras. São obtidas duas respostas, os alvos $r_1(i, j, l-1)$ e $r_2(i, j, l-1)$.

d) Para cada pixel $(i, j) \in \beta$, calcular as 4 distâncias entre as respostas de dois filtros $r_u(i, j, l-1)$ e o protótipo μ_v correspondente à classe C_1 e C_2 , isto é, $\|\mu_v - r_u(i, j, l-1)\|$, $1 \leq u, v \leq 2$. Cada pixel da fronteira recebe uma etiqueta de classe que dá a distância mínima.

e) Diminuir o valor l de um e repetir o passo a) até que o fim da pirâmide seja alcançado.

2.4 – DISCUSSÃO DOS ALGORITMOS PROPOSTOS

O trabalho de Spann e Wilson é uma resolução do problema da segmentação de uma forma que evita o problema do conhecimento à priori do tipo de imagem a tratar. Especificamente, as classes são obtidas pelo uso de um algoritmo de aglomeração unidimensional baseado nos níveis de cinzento que não requer qualquer informação à

priori do número das classes. Isto é feito à custa da resolução espacial e de acordo com o princípio de que quando se definirem os protótipos das classes é necessário encontrar fronteiras concretas e que quando se encontram fronteiras é necessário calcular os protótipos.

A resolução espacial é restaurada seguindo o princípio de que as propriedades espaciais das regiões não são variantes sobre a escala. É devido a esta consideração que é justificada a propagação dos membros das classes através da árvore quaternária.

A melhoria óbvia deste algoritmo é o uso de um processo de estimação da fronteira adaptativa orientada. Os autores Schroeter e Bigün implementam esta melhoria, propondo um processo de estimação orientada adaptativa, embebida numa estrutura de multi-resolução piramidal que permite o uso de diferentes algoritmos de aglomeração multi-atributos, sem conectividade espacial obrigatória.

Apesar do problema do conhecimento à priori do número exacto de classes não ficar resolvido, este foi reduzido pela estimação do número inicial de classes e pela re-atribuição das classes insignificantes.

Experiências efectuadas pelos autores Schroeter e Bigün demonstraram que classificações similares são obtidas com diferentes números iniciais de classes C , e pequenas diferenças na distribuição espacial das etiquetas são eliminadas pelo processo de refinamento da fronteira. O processo de selecção do algoritmo de aglomeração é sempre um problema porque a forma dos aglomerados é desconhecida, no entanto, o algoritmo é relativamente insensível a este problema. As experiências

efectuadas pelos autores demonstraram que resultados muito semelhantes podem ser obtidos com diferentes algoritmos de aglomeração.

Um algoritmo de multi-passagem é proposto para atribuir classes pequenas e isoladas aos seus vizinhos por um processo que garante a conectividade espacial. A resolução é depois aumentada pela projecção para baixo das etiquetas das classes, em cada nível as fronteiras são melhoradas pela distribuição dos pixeis da região fronteira para as classes mais próximas. No entanto, a incerteza na classe tem de ser primeiro reduzida, isto é feito utilizando um filtro adaptativo do tipo Borboleta na orientação dominante. Os autores referem que resultados semelhantes de detecção de fronteira são obtidos com diferentes algoritmos de aglomeração e considerando diferentes números de orientações para os filtros.

Uma desvantagem do algoritmo, devido à sua estrutura simples em árvore quaternária, é a incapacidade de fazer o seguimento de pequenas regiões durante a construção da pirâmide. Um acordo entre o tamanho da mais pequena região detectável e a diminuição do ruído tem de ser encontrado. Os autores referem que a escolha do nível da pirâmide tem um certo grau de liberdade onde mais do que um nível pode produzir bons resultados de segmentação.

Outra desvantagem que surge neste tipo de algoritmo é a ocorrência de erros na classificação de um pixel no nível mais alto, que por consequência vai causar erros nas fronteiras nos níveis mais baixos, consequentemente erros no resultado final (fronteira). Estes podem ser minimizados pela aplicação do algoritmo que restaura o espaço de continuidade.

A apresentação de bons resultados para um grande conjunto de imagens experimentais e para uma grande gama de relações sinal ruído, indica um método eficiente e ao mesmo tempo de grande potencial de desenvolvimento. Por este motivo foi seleccionado para servir de base para a implementação do algoritmo em três dimensões. Esta implementação será executada em duas fases:

- Numa primeira fase será implementado o algoritmo de segmentação em duas dimensões tendo por base uma pirâmide construída sobre uma árvore quaternária, onde no nível mais alto desta será aplicado o agrupamento centróide local num espaço de atributos unidimensional, seguido da restauração da resolução espacial usando para a estimativa dos pixels fronteira filtros adaptativos orientados. Sobre este algoritmo efectuarão-se testes para verificar como se comporta com imagens práticas provenientes do projecto ISACS (capítulo 3).
- Na segunda fase consistirá na extrapolação deste algoritmo para três dimensões e aplicação em volumes de dados provenientes do projecto ISACS.

IMPLEMENTAÇÃO DA SEGMENTAÇÃO EM DUAS DIMENSÕES

3.1 - INTRODUÇÃO

Neste capítulo será explicada e ilustrada a implementação prática do algoritmo sumariado no anterior, fazendo para tal uso de uma das imagens existentes na base de dados ISACS (designada por: *topas*). Serão também referidas as alterações pontuais efectuadas no algoritmo, com a finalidade de melhorar o seu tempo de execução.

3.2 - IMPLEMENTAÇÃO DO ALGORITMO DE SEGMENTAÇÃO EM 2D

O algoritmo referido no capítulo II pode ser sumariado em quatro passos:

1) construção da Pirâmide; 2) algoritmo de aglomeração; 3) retirar regiões insignificantes e pixels isolados e 4) determinar as fronteiras.

3.2.1 - Construção da Pirâmide

O primeiro passo, a construção da pirâmide, tem por base a equação 2.1. Na figura 3.1 são visualizadas as imagens resultantes na forma de pirâmide, desde as dimensões 256x256 até 8x8 (nível *l* até 5). Tal como foi referido no capítulo anterior e por observação das imagens à medida que subimos na pirâmide, pode verificar-se que o ruído diminui, no entanto, a resolução de imagem piora e as regiões pequenas têm tendência a desaparecer ou a serem aglutinadas nas regiões maiores.

Em termos de apresentação do trabalho executado em duas dimensões, será ilustrado passo a passo o comportamento do algoritmo, considerando para tal a imagem “topas” e o nível $l=2$ como o nível mais alto da pirâmide (correspondente a uma imagem de 64×64). No fim do capítulo podem ser observados os resultados obtidos para o nível $l=1$ e $l=3$.

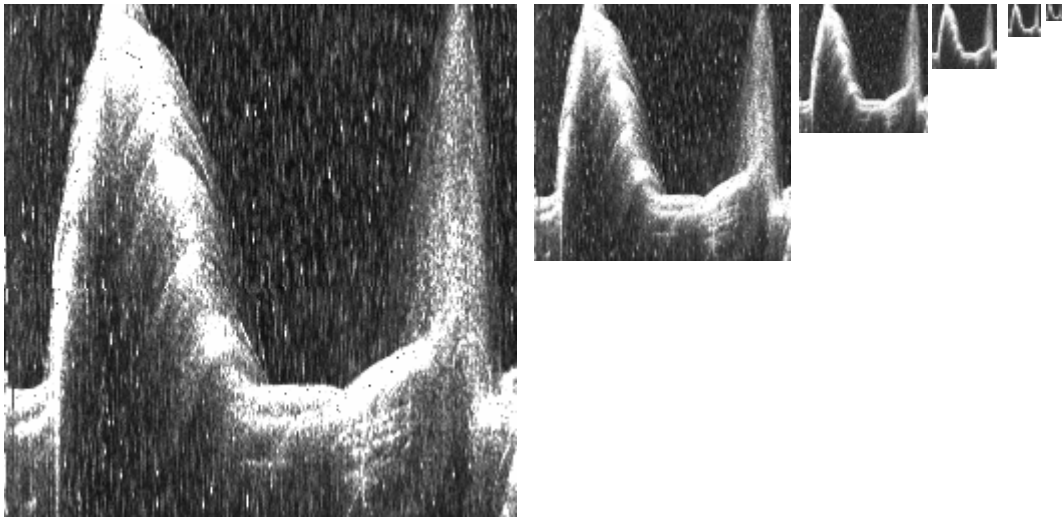


Fig. 3.1 - Construção da pirâmide com a imagem “topas” em 2D

3.2.2 - Algoritmo de Aglomeração

O segundo passo é a aplicação do algoritmo de aglomeração no nível mais alto da pirâmide, recorrendo para tal ao algoritmo que faz uso de um espaço de atributos unidimensional (nível de cinzento) - agrupamento centróide local, como referido no capítulo anterior. Este algoritmo permite encontrar os centros das classes existentes na imagem no nível em que vai ser aplicado. Para a atribuição dos pixels da imagem às etiquetas correspondentes (centro da classe), em vez de se seguir o critério proposto pelos autores Spann e Wilson referido no capítulo anterior (seguir cada pixel para o seu centro de classe respectiva), será aplicado o critério da distância mínima.

O **critério da distância mínima** consiste no cálculo do valor médio entre centros de classes consecutivas, e após a determinação deste, atribuir a etiqueta da classe interior a todos os pixels da imagem $q(i, j; l)$ compreendidos entre duas distâncias mínimas consecutivas, criando a imagem das etiquetas $I_\gamma(i, j; l)$. Este método é muito simples e rápido na atribuição dos pixels às classes correspondentes, podendo no entanto esporadicamente classificar incorrectamente alguns pixels, mas estes serão corrigidos pela restauração do espaço de continuidade (passo 3) e à medida que melhoramos a resolução espacial (passo 4).

Na figura 3.2a, pode ser observado o histograma da imagem original, no qual apenas podem ser "distinguidas" duas grandes regiões: a região da montanha (linha à direita) e a região correspondente ao fundo com ruído (toda a região mais à esquerda). À medida que vamos subindo na pirâmide, o histograma começa a aparecer com mais picos sendo cada vez mais fácil determinar a existência de outras regiões e conseqüentemente das classes (figura 3.2b correspondente ao histograma da imagem $q(i, j; 2)$), neste nível é aplicado o algoritmo de agrupamento onde são determinados os centros das classes. Na figura 3.2c, encontra-se representado o histograma correspondente à imagem das etiquetas ($I_\gamma(i, j; 2)$), após a aplicação do algoritmo de agrupamento, de classificação e após terem sido retiradas as regiões insignificantes e os pixels isolados (ver secção 3.2.3). A imagem que apresenta já todos os pixels atribuídos aos seus centros de classes correspondentes e que corresponde ao histograma da figura 3.2c pode ser observada na figura 3.3c. Na imagem em causa foram encontradas cinco classes distintas.

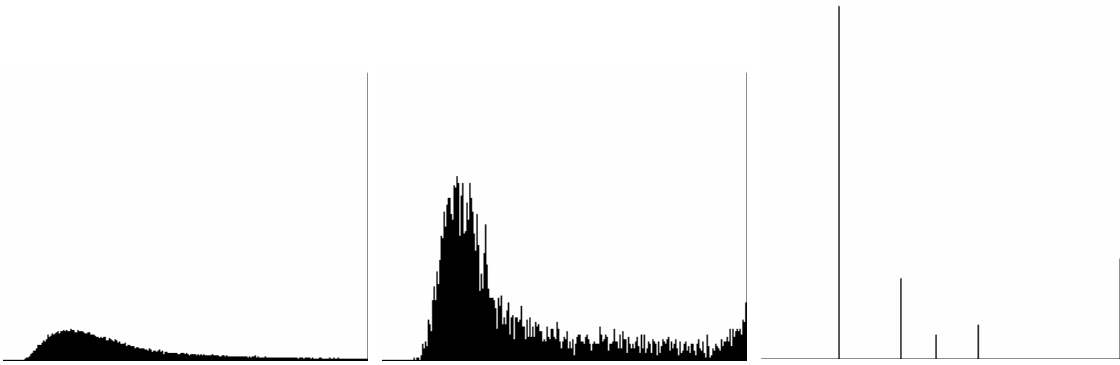


Fig. 3.2 –Histogramas da imagem “topas” em 2D. Da esquerda para a direita a), b), c).

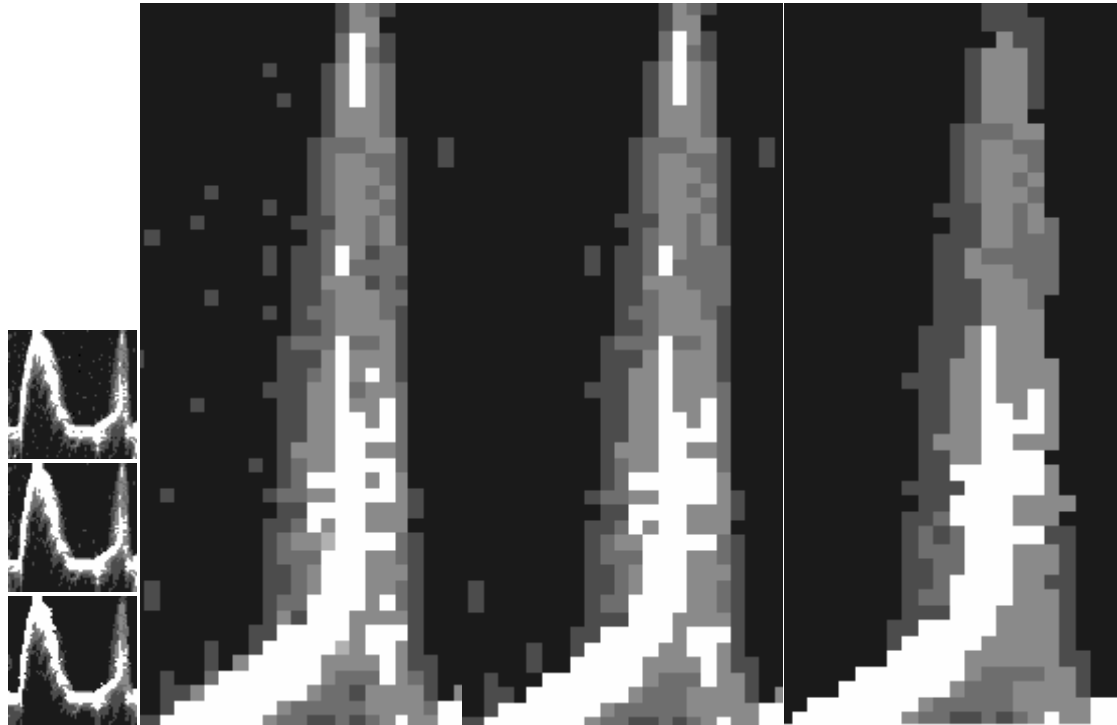
3.2.3 - Espaço de Continuidade

No terceiro passo, **retirar regiões insignificantes** (pequenas) e **pixels isolados**, foi utilizado um algoritmo determinístico com conectividade espacial que numa única passagem pela imagem faz a atribuição das regiões consideradas insignificantes bem como dos pixels isolados. Este algoritmo é diferente do proposto por Schroeter e Bigün, mas é utilizado com o objectivo de processar a imagem mais rapidamente e com o mínimo de perda de qualidade.

A cada **pixel isolado** (considera-se aquele que na localização (i, j) tem a etiqueta diferente de todos os seus vizinhos de oito), é atribuída a etiqueta correspondente à classe mais representativa. Por **classe mais representativa** é considerada aquela em que as etiquetas se repetem mais vezes em $N_8(i, j)$, no caso em que existam duas ou mais classes com igual representatividade é atribuída a etiqueta que mais se aproxima do pixel isolado.

As regiões são consideradas **insignificantes** quando o número de pixels conectado em qualquer direcção é menor ou igual a 9. São atribuídas numa única passagem em que cada pixel pertencente à região é atribuído um a um, como se fosse um pixel isolado.

O resultado da imagem das etiquetas, no nível $l=2$ e após ter sido executado o algoritmo de aglomeração e classificação e terem sido retirados os pixels isolados e as



regiões insignificantes, pode ser observado na figura 3.3c.

Fig. 3.3 – Imagem das etiquetas ($l=2$), na esquerda de cima para baixo: a) após o agrupamento e a classificação; b) depois de retirar pixels isolados; c) depois de retirar regiões insignificantes. Da esquerda para a direita, o canto superior direito das imagens atrás referidas: d),e),f).

Na figura 3.3d, observa-se uma ampliação do canto superior direito da imagem das etiquetas no nível $l=2$ (fig. 3.3a). Podemos observar uma elevada quantidade de pixels isolados, bem como algumas regiões pequenas que não têm significado em comparação com as adjacentes. Na figura 3.3e, observa-se a ampliação da mesma região mas agora com os pixels isolados já reclassificados. Por último, na figura 3.3f pode-se observar a imagem com os pixels e regiões insignificantes retiradas. Desta imagem pode-se concluir que o algoritmo proposto para retirar as regiões insignificantes funciona como esperado.

3.2.4 - Estimação das Fronteiras

O quarto e último ponto, determinar as fronteiras, consiste em transformar a imagem a partir do nível mais alto da pirâmide até ao nível com a resolução original.

Para a passagem da imagem das etiquetas $I_\gamma(i, j; 2)$ para a imagem um nível mais abaixo $I_\gamma(i, j; 1)$, foram executadas as seguintes operações:

No nível mais alto ($l=2$) foram determinados os pixels fronteira. Para tal, foi verificado quais os pixels que continham pelo menos uma etiqueta em $N_8(i, j)$ diferente do pixel (i, j) . Para todos os pixels não pertencentes à fronteira, foram atribuídas as etiquetas dos pais aos quatro filhos correspondentes (ver fig. 2.4).

Foram predefinidas quatro máscaras do tipo Borboleta (fig. 2.5) em quatro direcções distintas. Estas direcções foram determinadas olhando a partir do pixel central (0) para os pixels adjacentes, surgindo assim oito sentidos distintos, o que sugere oito máscaras distintas, mas por observação da figura 3.4b e pela simetria da máscara concluí-se que quatro dos sentidos são redundantes.

A decisão de implementar apenas estas quatro direcções deve-se ao facto de ter sido constatado que após a realização de testes com as imagens disponíveis e com um número de direcções diferentes (entre quatro e oito), não se verificou alterações significativas em termos de resultados finais para uma máscara Borboleta 3x3. Isto reforça a observação efectuada pelos autores Schroeter e Bigün que afirmaram que resultados semelhantes se obtêm para um número diferente de máscaras. Esta constatação permitiu determinar as quatro máscaras nas direcções a), b), c), d) da

figura 3.4b e predefini-las como as utilizáveis, evitando assim o tempo de cálculo despendido na rotação da máscara da posição horizontal para a direcção estimada.

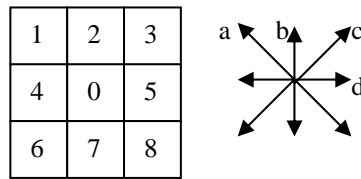


Fig. 3.4 – Direcção e sentidos de aplicação do filtro Borboleta em 2D.

Da esquerda para a direita a), b).

Para os pixels fronteiros, os coeficientes da máscara do filtro Borboleta 3x3 foram determinados independentemente para cada pixel da região de fronteira β (fig. 2.4).

Os pesos das máscaras são calculados tendo em consideração os critérios que os

autores Schroeter e Bigün aconselham: $d = \frac{|\mu_1 - \mu_2|}{\sqrt{\sigma_1^2 + \sigma_2^2}}$; $r = \text{fct}(d)$ e $rr = (1-r)/n$,

com $\text{fct}(\cdot)$ dada pela fig. 2.6), e n o número de pesos da máscara diferentes de 0 e r .

Para a atribuição da etiqueta ao pixel $(i, j) \in \beta$ foram implementados dois métodos que têm por base o mesmo princípio:

No **primeiro método**, seguindo o que os autores Schroeter e Bigün propõem, para cada pixel $(i,j) \in \beta$ estima-se a orientação da fronteira utilizando a equação 2.28, determinando as duas médias das classes uma de cada lado da fronteira μ_1 e μ_2 . A partir desta estimativa utiliza-se o filtro predefinido com a direcção que mais se aproxima à da direcção predominante. Calcula-se de seguida as quatro distâncias entre as respostas dos dois alvos do filtro $r_u(i, j, l-1)$ e os protótipos μ_v correspondentes às duas classes, $\|\mu_v - r_u(i, j, l-1)\|$, $1 \leq u, v \leq 2$. Cada pixel da fronteira recebe a etiqueta da classe para a qual a distância é mínima.

No **segundo método**, para cada pixel $(i, j) \in \beta$ são determinadas as duas classes mais representativas, existentes em $N_8(i, j)$ da imagem $I_\gamma(i, j; 2)$, para a determinação dessas classes foram apenas consideradas representativas aquelas que eram compostas por dois ou mais pixels. Quando existiam duas ou mais classes com o mesmo grau de representatividade, considera-se as C_n classes (com $1 \leq n \leq 4$) para o cálculo da dissemelhança e conseqüentemente dos pesos das máscaras. Foram efectuadas todas as combinações de duas a duas entre diferentes classes existentes à volta do pixel fronteiro, de seguida e em separado, é determinado para cada máscara os dois alvos do filtro $r_1(i, j, k, l)$ e $r_2(i, j, k, l)$ (ver fig. 2.5). É depois calculada a distância euclidiana entre os alvos e as classes determinadas anteriormente $\|\mu_v - r_u(i, j, l-1)\|$,

$1 \leq u \leq \frac{2 \cdot n_filtro \cdot (C_n!)}{2!(C_n - 2)!}$, com "n_filtro" o número de filtros considerados (quatro no

presente caso), e $1 \leq v \leq \frac{C_n!}{2!(C_n - 2)!}$. Sendo o pixel atribuído a etiqueta que corresponde

à distância mínima entre $\|\mu_v - r_u(i, j, 1)\|$.

Com este procedimento simplificou-se o processo de reclassificação dos pixels pertencentes à fronteira, bem como diminuiu-se o tempo despendido no cálculo da estimativa da direcção da fronteira e da rotação da máscara para a direcção estimada, isto por troca do cálculo dos diferentes r_u , que no caso da C_n ser dois (é verificado que raramente o valor é maior do que dois) temos oito valores distintos de r_u .

Observou-se que os dois métodos apresentam resultados bons e aproximados, no entanto, o segundo método para o mesmo número de máscaras (quatro) e para máscaras 3x3 apresenta um resultado ligeiramente melhor, e o tempo de cálculo é

ligeiramente mais rápido (o segundo método é 1.25 vezes mais rápido que o primeiro, isto para a imagem da figura 2.1 com as dimensões 256x256).

A imagem das etiquetas resultante para o nível $l=1$ pode ser observada na figura 3.5a para o primeiro método e na 3.5b para o segundo método. Foi referido anteriormente que o segundo método apresenta resultados um pouco melhores que o primeiro, isto justifica-se pelo facto de um observador humano usualmente delimitar as regiões com contornos “suaves”. Esse tipo de contorno está mais presente na imagem 3.5b (2º método) do que na imagem 3.5a (1º método). Pode-se verificar isso perfeitamente quando se observa o contorno da montanha mais à esquerda na imagem.



Fig. 3.5 – Imagem das etiquetas ($l=1$). Da esquerda para a direita
a) primeiro método; b) segundo método.

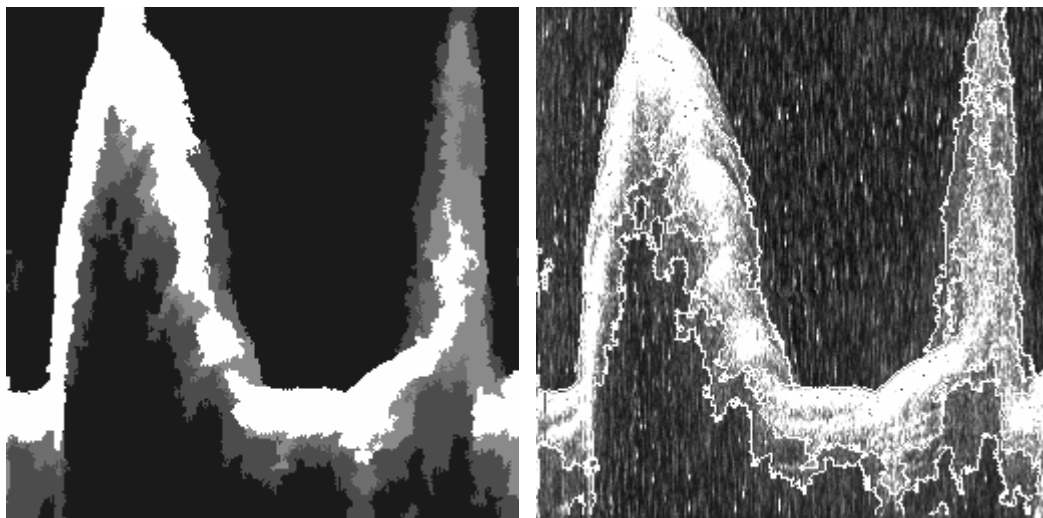


Fig. 3.6 – Imagem resultado para ($l_{\max} = 2$). Da esquerda para a direita; a) Imagem das etiquetas; b) imagem com as fronteiras marcadas.

O quarto passo - Estimação da Fronteira é repetido tantas vezes quanto as necessárias para alcançar o nível mais baixo da pirâmide.

O nível mais baixo da pirâmide ($l=0$) pode ser observado na figura 3.6a, para a imagem das etiqueta e na figura 3.6b para a imagem original com as fronteiras sobrepostas (que estão representadas a branco), isto fazendo a estimação da fronteira utilizando o segundo método.

3.3 – CRÍTICA AOS RESULTADOS EM 2D

Para algumas imagens reais obtidas a partir da base de imagens ISACS, foram realizados mais alguns testes (Apêndice A) considerando o nível mais alto da pirâmide o nível $l=\{1, 2, 3\}$. Para a imagem *topas* os resultados obtidos podem ser observados na figura 3.7a) e b) e figura 3.7c) e d) respectivamente para $l=1$ e $l=3$.

Da observação das imagens resultantes foram constatados três factos:

- Primeiro é a independência, relativamente ao nível ($l=\{1, 2, 3\}$), na determinação dos centros das classes, isto é, em qualquer dos níveis testados os centros das classes determinados são praticamente coincidentes. Nota-se uma pequena excepção para o nível $l=1$, onde aparece uma pequena classe que pode ser visualizada na figura 3.7a, b). Esta desaparece por completo quando no nível mais alto da pirâmide considera-se que as classes insignificantes são todas aquelas que têm menos de 10 pixels conectados (em vez dos 9 que se esta à considerar).
- Segundo é que quanto mais se subir na pirâmide, pior é a resolução das fronteiras e menor é a quantidade de regiões insignificantes, no entanto, torna-se mais fácil a determinação dos centros das classes e conseqüentemente a determinação das grandes regiões representativas da imagem (v.d. fig. 3.7c e d)).

- Terceiro é que quanto menos se subir na pirâmide melhor é a resolução da fronteira e maior é a quantidade de regiões pequenas ou parasitas que aparecem na imagem (fig. 3.7a e b)). No entanto, estas podem ser retiradas totalmente ou parcialmente por aplicação do algoritmo do passo 3 (aumentando o número de pixels para o qual consideramos a região insignificante), a desvantagem deste procedimento é que o algoritmo em termos globais torna-se mais lento, sendo mais rápido descer um nível da pirâmide do que recorrer ao algoritmo do passo 3).

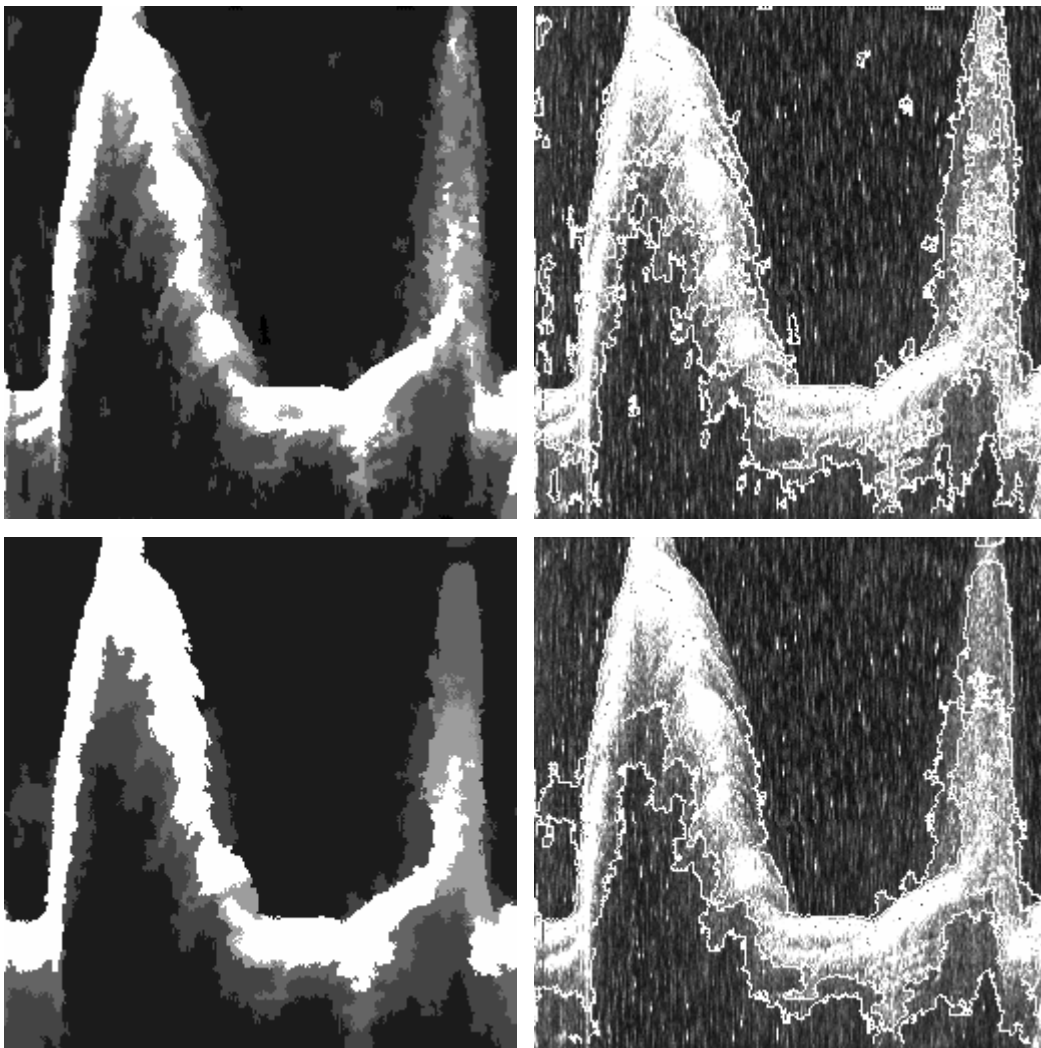


Fig. 3.7 – À esquerda imagens das etiquetas, à direita imagens com as regiões delimitadas. Em cima ($l_{\max} = 1$) a),b), em baixo ($l_{\max} = 3$) c),d).

3.4 – CONCLUSÕES SOBRE A SEGMENTAÇÃO EM 2D

Os resultados do algoritmo de duas dimensões são considerados muito bons, uma vez que foi possível determinar as diferentes regiões da imagem e delimitá-las, tal como o observador humano as delimitaria, que era um dos nossos objectivos.

Pode-se também concluir que o algoritmo é "imune" ao tipo de ruído que as imagens contêm, sendo suficiente para tal subir até ao nível $l=3$ da pirâmide, pois este apresenta um bom compromisso entre a quantidade de pequenas regiões, e a qualidade dos contornos que se observam na imagem resultado (ver figura 3.7).

É um algoritmo fiável, rápido e que apresenta boas potencialidades para ser extrapolado para três dimensões. Isto utilizando como comparação o algoritmo de partida proposto por Schroeter e Bigün e pela análise dos resultados obtidos para todas as imagens aplicadas. Pode-se visualizar alguns desses resultados no Apêndice A, para $l=\{1, 2, 3\}$.

No entanto, após extraídas as conclusões acima apresentadas, surgem algumas perguntas tais como: Qual o nível óptimo da pirâmide para este tipo de imagem? Qual foi o melhor resultado obtido? Qual o resultado que nos dá mais informação sobre as imagens?

A resposta a estas questões não está directamente relacionada com o algoritmo implementado, mas sim com as informações que os peritos de acústica submarina conseguem retirar da análise das diferentes imagens obtidas a partir dos diferentes níveis da pirâmide, e da transmissão por estes, do que é informação importante e do que é informação redundante.

Para a observação e análise final das imagens resultantes, e conseqüentemente das fronteiras, propõe-se um processo de retroalimentação (a ser executado pelos peritos em acústica submarina), que pode ser dividido em dois métodos:

- O primeiro método consiste em visualizar as imagens resultantes (no nível $l=0$) efectuando-se a segmentação partindo de diferentes níveis da pirâmide ou seja, visualizar a imagem final partindo por exemplo do nível $l_{\max}=3$ (menor quantidade de informação), depois decidir, se for necessário, utilizar como nível de partida o nível $l_{\max}=2$ ou $l_{\max}=1$. Caso seja necessário, pode-se ainda subir mais na pirâmide e/ou pode-se apenas segmentar partes da imagem utilizando diferentes níveis.
- O segundo método consiste em visualizar directamente ao longo dos níveis da árvore os resultados obtidos, retirando do nível pretendido a informação tida como relevante.

Outra hipótese é aplicar o algoritmo de retirar regiões insignificantes no nível mais baixo da pirâmide ($l=0$), conseguindo assim retirar as regiões pouco significativas e manter uma elevada resolução das fronteiras, mas isto à custa de um aumento do tempo (CPU) total do algoritmo. Não se aconselha este método devido ao ultimo ponto referido, elevado tempo de cálculo.

SEGMENTAÇÃO DE IMAGEM EM TRÊS DIMENSÕES

4.1 – INTRODUÇÃO

O algoritmo implementado no capítulo anterior apresenta bons resultados para imagens em duas dimensões (incluindo as imagens acústica submarina provenientes do projecto ISACS), apresentando-se assim, como um bom algoritmo para ser extrapolado para as três dimensões. Será explicado neste capítulo o algoritmo a propôr para três dimensões, será também realizado um conjunto de testes para verificar a sua fiabilidade e a sua potencialidade.

Em três dimensões surge um novo conceito - **voxel**, elemento básico de uma imagem digital a três dimensões $I(x, y, z)$, que corresponde à intersecção entre a linha, a coluna e a "profundidade", ou seja é o equivalente a unidade básica de uma imagem a duas dimensões - pixel.

4.2 – DESCRIÇÃO DO ALGORITMO EM 3D

A segmentação em três dimensões, será concretizada em três passos: O primeiro passo - **Alisamento**, será construída uma pirâmide fazendo uso de uma árvore octógona. No segundo passo - **Classificação**, será efectuado o agrupamento no espaço de atributos unidimensional, utilizando - Agrupamento Centróide Local, seguidamente os voxéis isolados serão realocados bem como as classes insignificantes. No último passo - **Estimação da fronteira**, a resolução espacial será

melhorada gradualmente através da projecção para baixo das etiquetas e por redefinição dos voxéis fronteiros utilizando para tal filtros Borboleta 3D em diferentes direcções.

4.2.1- Árvore Octógona

A construção da pirâmide a três dimensões é baseada na árvore octógona (fig. 4.1), a qual é construída a partir da média de blocos de $2 \times 2 \times 2$ voxéis não sobrepostos, que resulta numa imagem de tamanho $2^{m-1} \times 2^{m-1} \times 2^{m-1}$ (no nível acima da base). Este processo pode ser repetido até atingir o nível mais alto, isto é, até ao nível cujo valor é o valor médio da imagem inteira.

Considere-se uma imagem $N \times N \times N^1$, $I(i, j, k)$ definida para $0 \leq i, j, k < N$, onde $N = 2^m$. A árvore octógona é definida por:

$$q(i, j, k; l) = \frac{1}{8} \sum_{a=0}^1 \sum_{b=0}^1 \sum_{c=0}^1 q(2i+a, 2j+b, 2k+c; l-1) \quad (4.1)$$

onde $0 < l \leq m$; $0 \leq i, j, k < 2^{m-l}$ e $q(i, j, k; 0) = I(i, j, k)$

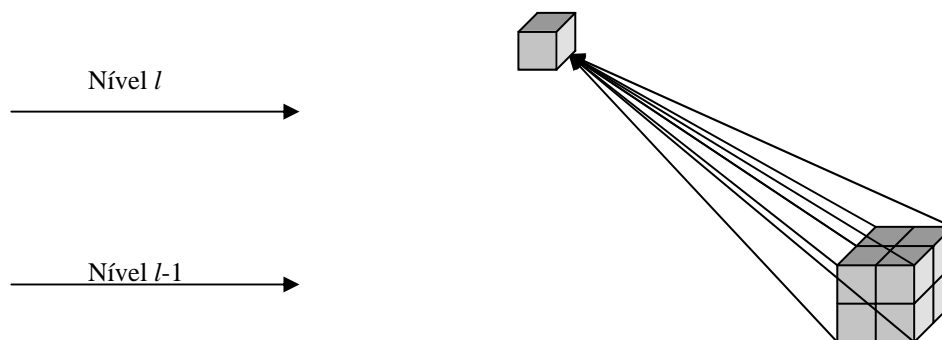


Fig. 4.1 –Árvore octógona.

Tal como no caso da árvore quaternária, a árvore octógona é um método rápido e eficaz para reduzir o ruído, e a truncagem no nível $l_{\max} < m$ possibilita a obtenção de

um ganho de alisamento, enquanto se mantém a resolução suficiente para garantir uma segmentação correcta.

4.2.2 - Classificação

4.2.2.1 - Aglomeração das Classes

A procura das classes é executada pelo algoritmo de agrupamento, utilizado em duas dimensões, uma vez que este é independente da dimensão da imagem. O algoritmo considera um espaço de atributos unidimensional baseado nos níveis de cinzento dos voxéis - agrupamento centroide local (ver Cap. II - 2.2.2-Classificação). A aplicação do algoritmo é implementada no nível mais alto da pirâmide (l_{\max}). A atribuição das etiquetas aos voxéis é efectuada através do critério da distância mínima (ver Cap. III - 3.2.2 - Algoritmo Aglomeração).

No fim do processo de aglomeração (agrupamento), cada voxel pertencente à mesma classe recebe uma etiqueta γ_i ($1 \leq i \leq C$), sendo C o número de classes determinadas.

No nível mais baixo de resolução l_{\max} , obtemos a imagem das etiquetas $I\gamma(i, j, k; l_{\max})$.

4.2.2.2 - Restauração do Espaço de Continuidade

No nível mais baixo de resolução l_{\max} , podem ser observadas fronteiras grosseiras entre diferentes classes, em que alguns dos voxéis e/ou conjuntos de voxéis estão espacialmente mal classificados. Isto é causado pela aplicação de um algoritmo de aglomeração sem restrições de conectividade espacial.

¹ O mesmo raciocínio pode ser utilizado para volume diferente de um cubo.

Um voxel (i, j, k) é considerado **voxel isolado** (fig. 4.2d) ou espacialmente mal classificado se em $I\gamma(i, j, k; l_{\max})$ a sua etiqueta é diferente de todas as etiquetas em $N_{26}(i, j, k)$, com $N_{26}(i, j, k)$ os vinte e seis vizinhos mais próximos de um voxel na localização (i, j, k) , $N_{26}(i, j, k) = \{(i + u, j + v, k + t)\}$, $-1 \leq u, v, t \leq 1$ e $(u, v, t) \neq (0, 0, 0)$.

Para reclassificar um voxel isolado, determina-se as classes candidatas. A selecção das classes candidatas é baseada nas etiquetas $I\gamma(i + u, j + v, k + t; l_{\max}) \in N_{26}(i, j, k)$ diferentes de $I\gamma(i, j, k; l_{\max})$. Só são consideradas **classes candidatas** aquelas que o número de voxéis seja maior ou igual a três. Caso não exista nenhuma classe com as condições necessárias para ser candidata, o voxel é atribuído à primeira classe não candidata determinada.

A cada voxel isolado (i, j, k) é atribuído a classe mais representativa das classes candidatas em $N_{26}(i, j, k)$, sendo esta a que apresenta a maior quantidade de voxéis pertencentes na vizinhança N_{26} . No caso de existirem classes com igual representatividade, será calculada a diferença absoluta entre o voxel e as diferentes classes com igual representatividade. Ao voxel (i, j, k) é atribuído a classe para a qual a diferença é mínima.

Regiões insignificantes (ou regiões pequenas) são atribuídas às classes vizinhas. Uma região pertencente a $I\gamma(i, j, k; l_{\max})$, é considerada insignificante quando não contém um número suficiente de voxéis conectados, qualquer que seja a direcção analisada na imagem. No entanto, o significado de regiões insignificantes está directamente relacionado com a altura da pirâmide, ou seja, para um valor grande de

l_{\max} a quantidade de voxels para a qual é considerado classe insignificante deverá ser mais pequena do que para um valor mais baixo de l_{\max} .

Consideremos as regiões sombreadas da figura 4.2. Estas regiões falham o teste de serem significativas (≤ 27 voxels conectados), então têm de ser atribuídas a uma região vizinha e no caso b) e c) entre elas próprias. O procedimento de atribuição é feito por um processo determinístico numa única passagem.

O processo de determinação/atribuição da região insignificante inicia-se na coordenada inicial do volume (0, 0, 0), canto superior esquerdo (fig. 4.2), vai passar por todos os voxels, terminando na coordenada final do volume (N_x, N_y, N_z), canto inferior direito (fig. 4.2). A região logo que é determinada (região insignificante) como por exemplo C_a constituída por n_a voxels, é reclassificada. Para tal é criada no processo de determinação uma lista das posições (no volume) dos voxels pertencentes à região (C_a). Esses voxels são atribuídos individualmente as suas classes vizinhas em N_{26} seguindo a ordem da lista como se de voxels isolados se tratassem, com a restrição que nenhum deles pode ser atribuído a etiqueta que antes tinha. A etiqueta de um voxel ($\in C_a$) já reclassificado é utilizada na determinação das classes candidatas do novo voxel ($\in C_a$) que esta a ser classificado (caso pertença a N_{26} desse mesmo voxel).

Para o caso de duas classes insignificantes adjacentes b) e c), a primeira região insignificante encontrada por exemplo b) (neste momento ainda não se sabe da existência da região c)), é processada da mesma forma que foi a região a), havendo a possibilidade dos voxels serem atribuídos às etiquetas correspondente à classe c)

desde que estes verificassem as condições de atribuição para voxéis isolados. Após o término da reclassificação de todos os voxéis da classe b), se a classe c) for insignificante, então é tratada da mesma forma que a classe a) ou b). Podemos generalizar este procedimento para mais de duas classes insignificantes adjacentes.

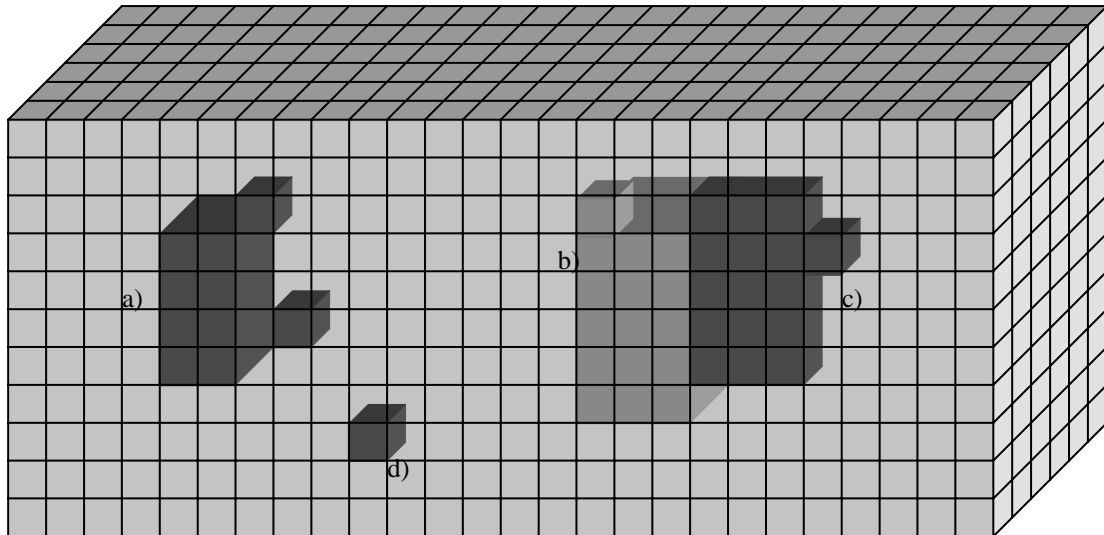


Fig 4.2 – Classes que têm de ser realocadas em 3D

Sabe-se no entanto que este método pode provocar um ganho de voxéis nas regiões insignificantes como c), que só poderia ser minimizado por um método multi-passagem. Aceita-se no entanto este erro por troca da diminuição do tempo despendido na reclassificação das regiões insignificantes.

4.2.3 - Estimação da Fronteira

O último passo do algoritmo é a melhoria das fronteiras, com o objectivo de melhorar a resolução espacial da imagem etiquetada $I\gamma(i, j, k; l)$, utilizando-se para tal filtros com a forma Borboleta 3D com o objectivo de reduzir a influência da incerteza da classificação dos voxéis ao longo da fronteira

A resolução espacial é gradualmente restaurada pela projecção para baixo das classes etiquetadas e pela análise dos voxéis à volta das fronteiras. Estes são atribuídos à

classe vizinha mais próxima. Assumiu-se que os protótipos das classes têm valores constantes ao longo dos diferentes níveis da pirâmide.

No nível com menor resolução l , os voxels da fronteira são determinados. Cada voxel(i, j, k) é considerado como um **voxel da fronteira** se pelo menos uma etiqueta em $N_{26}(i, j, k)$ é diferente de $I\gamma(i, j, k; l)$. A imagem $I\gamma(i, j, k; l-1)$ é obtida pela projecção para baixo da etiqueta de cada voxel pai não fronteiro para os seus respectivos oito voxels filhos, $I\gamma(i, j, k; l-1) = I\gamma(i/2, j/2, k/2; l)$, com / o inteiro da divisão.

Assim, os filhos que não pertencem à fronteira recebem a etiqueta do seu pai, e os filhos que pertencem à fronteira, definem a região β no nível $l-1$ (fig. 4.3) na qual os protótipos das classes vão ser determinados fazendo uso dos filtros Borboleta 3D. A região fronteira β está subdividida em duas: β_1 que corresponde aos oito voxels do nível $l-1$ provenientes do voxel direito da fronteira no nível l , e β_2 que corresponde aos oito voxels do nível $l-1$ provenientes do voxel esquerdo. Designamos por região fronteira α , os voxels da região β_1 que estão em contacto com os da região β_2 , e os de β_2 que estão em contacto com β_1 . O significado da região α será explicado mais tarde na secção 4.2.3.1 - Restrição dos Voxels Fronteira.

Para determinar os protótipos das classes dos voxels pertencentes à região β foi definida uma extensão do filtro Borboleta em duas dimensões para três dimensões designado por Borboleta 3D. O formato e a distribuição dos pesos pela máscara na orientação do eixo y é apresentado na figura 4.4 para uma máscara $3 \times 3 \times 3$ e $5 \times 5 \times 5$. O

cálculo do filtro para outras direcções é efectuado pela rotação deste para a direcção pretendida.

Tal como a duas dimensões, foi pré-definido um conjunto de filtros em 3D, que deu origem à definição de treze filtros. O critério considerado como ponto de partida para a escolha dos filtros e as suas direcções, foi olhar a partir do pixel central (0) de uma máscara 3x3x3, para os seus adjacentes (fig. 4.5a). Está apresentado na figura 4.5b todas as direcções e sentidos possíveis, que correspondem às treze direcções e aos vinte e seis sentidos distintos.

Por observação da simetria do filtro (figura 4.4 e 4.5b) verifica-se que é suficiente o cálculo das treze direcções para se obter todos os sentidos do filtro, pois os restantes são redundantes. Os filtros foram definidos nas orientações correspondentes a a), b), c), d), e), f), g), h), i), j), l), m), n) apresentadas na figura 4.5c

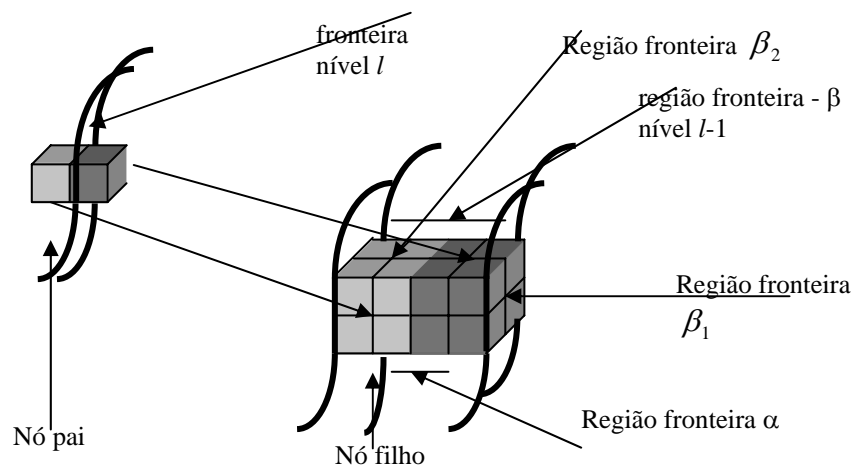


Fig 4.3 – Regiões fronteira β e α em 3D.

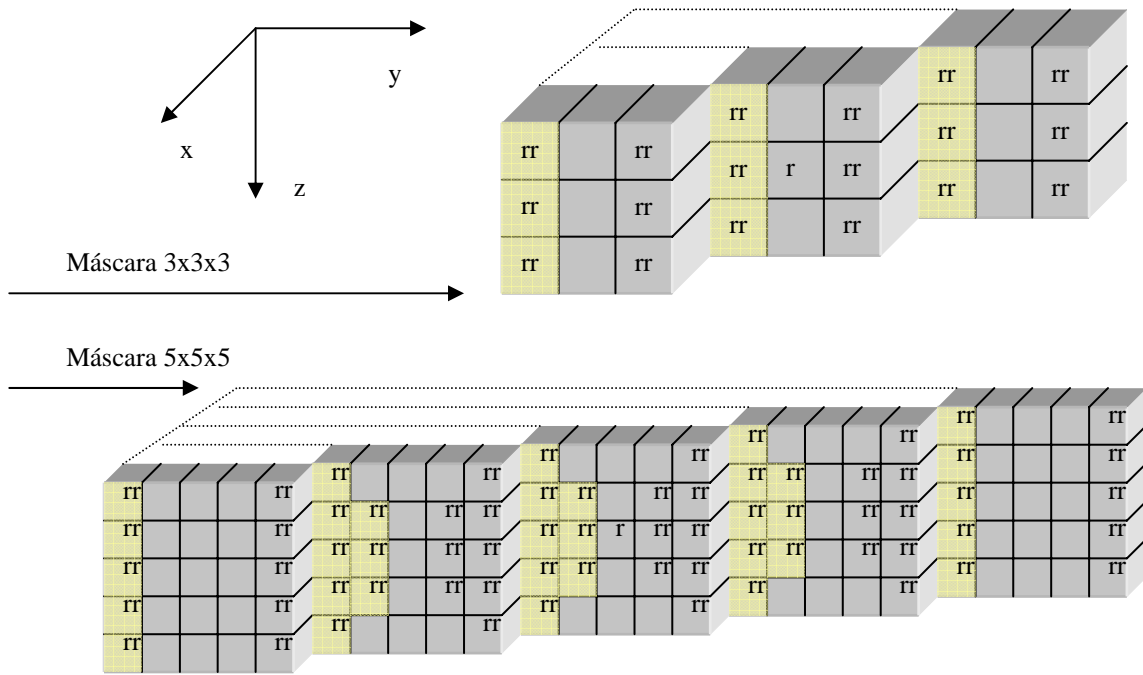
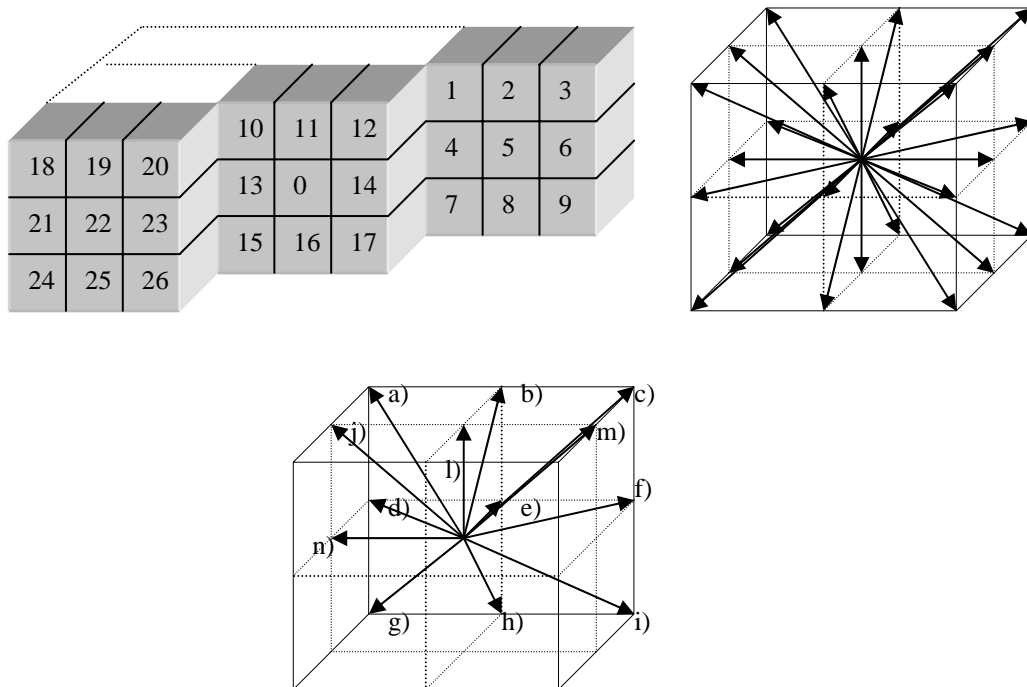


Fig. 4.4 – Filtro Borboleta em 3D (orientação do eixo y).



.Fig 4.5 – Direções e sentidos de aplicação dos filtros Borboleta em 3D.

Da esquerda para direita: a) ,b). Em baixo c).

Os pesos r e rr das máscaras dos filtros, são dados por $r = \text{fct}(d)$ e $rr = (1 - r)/n$, com

$$d = \frac{|\mu_1 - \mu_2|}{\sqrt{\sigma_1^2 + \sigma_2^2}}$$

e onde $\text{fct}(\cdot)$ é dada pela figura 2.6) e com n o número de pesos

diferentes na máscara de 0 e r . O μ_i e σ_i^2 são a média e a variância determinadas na imagem $q(i, j, k; l)$ (imagem nível l da pirâmide, ver eq. 4.1), que correspondem às classes mais representativas do voxel fronteira, determinado na imagem $I_\gamma(i, j, k; l)$.

Define-se **classe representativa** como a classe que é constituída por três ou mais voxéis com etiqueta igual. Considera-se as duas classes mais representativas aquelas que contêm maior quantidade de voxéis com a mesma etiqueta. Caso existam mais de duas classes com igual representatividade considera-se C_n classes.

A cada voxel $(i, j, k) \in \beta$ são aplicadas todas as direcções dos filtros pré-definidos (ver fig. 4.5c) na imagem $q(i, j, k; l)$. Se o número de classes representativas for igual a dois ($n=2$) então cada filtro é constituído por uma única máscara (ver fig. 4.4). Caso contrário, se o número de classes representativas (C_n) for maior que dois, o filtro para cada direcção pré-definida é constituído por um conjunto de máscaras obtidas pela combinação de todos os pares das classes (C_n). Para o caso de três classes com igual representatividade $n=3$ cada filtro em cada direcção é constituído por três máscaras em que os coeficientes de cada máscara (r e rr) vão variar em função da dissemelhança, que por sua vez depende de C_n . Para cada filtro os coeficientes da primeira máscara são determinados utilizando C_1 e C_2 (μ_1 e μ_2), para a segunda C_2 e C_3 (μ_2 e μ_3) e para a terceira máscara C_1 e C_3 (μ_1 e μ_3).

Se o número de classes representativas for igual a zero, isto é, se não for encontrada nenhuma classe que obedeça ao critério de representatividade então o voxel é atribuído directamente ao valor do seu pai sem qualquer filtragem. Foi constatado que o número de classes representativas raramente é maior que dois, os valores típicos são um ou dois.

As metades esquerda e direita dos filtros são aplicadas separadamente, eliminando o problema de alisamento sobre as fronteiras, sendo obtido os dois alvos $r_1(i, j, k; l-1)$ e $r_2(i, j, k; l-1)$ para cada filtro e sua correspondente máscara.

Para cada voxel $(i, j, k) \in \beta$ é calculado todas as distâncias entre as respostas obtidas $r_u(i, j, k; l-1)$ e os protótipos das classes μ_v (correspondentes às classes representativas C_n), isto é, $\|\mu_v - r_u(i, j, k; l-1)\|$, $1 \leq u \leq \frac{2 \cdot n_filtro \cdot (C_n!)}{2!(C_n - 2)!}$, e

$1 \leq v \leq \frac{C_n!}{2!(C_n - 2)!}$, com "n_filtro" o número de filtros considerados (treze no presente caso).

Cada voxel da fronteira recebe uma etiqueta da classe que apresenta a distância mínima de $\|\mu_v - r_u(i, j, k; l-1)\|$. Diminui-se o valor l de um e repete-se novamente o procedimento de determinação da fronteira até alcançar o fim da pirâmide.

4.2.3.1 - Restrição dos Voxeis Fronteira

Um dos problemas que surge na atribuição dos voxeis da fronteira é a grande quantidade de voxeis pertencentes à região fronteira β , que são necessários classificar (figura 4.3). Sempre que surge uma fronteira (dois voxeis no nível l) é necessário

reclassificar dezasseis voxéis (no nível $l-1$), oito correspondendo à região β_1 e oito à região β_2 . Para diminuir o número de voxéis a reclassificar foi considerado o seguinte critério: só são calculados os voxéis da região β na direcção onde foi determinada a fronteira, na figura 4.3 corresponderá a reclassificar só os voxéis pertencentes à região α , ou seja quatro da região β_1 e quatro da região β_2 . No caso em que a fronteira está definida em mais do que uma direcção então serão determinados todos os voxéis até ao máximo de oito para cada região β_n , com $n=1$ ou 2. A todos os voxéis que não são afectados pelo processo anterior e que pertencem à região β_n , será atribuída a etiqueta do seu pai.

A partir da figura 4.6, e recorrendo à tabela 4.1, podemos identificar o voxel ou os voxéis que vão ser considerados na região β_n mediante a direcção e sentido onde a fronteira é determinada.

A título demonstrativo da aplicação da tabela 4.1, suponha-se que no nível l a fronteira é detectada nos voxéis 3 e 12, que corresponde na região fronteira β_2 ao reclassificação respectivamente do voxel B e B+F. Como pode ser observado neste caso, a reclassificação do número de voxéis diminuiu de oito para três, com o mesmo voxel (B) reclassificado duas vezes. Para evitar o duplo cálculo, quando um voxel da região β_2 é reclassificado uma vez fica marcado evitando o seu cálculo de novo. Então o número de voxéis reclassificados passou de oito para dois, diminuindo assim bastante o tempo de cálculo da fronteira. O mesmo processo vai ser aplicado a todos os voxéis necessários pertencentes à região fronteira β_1 .

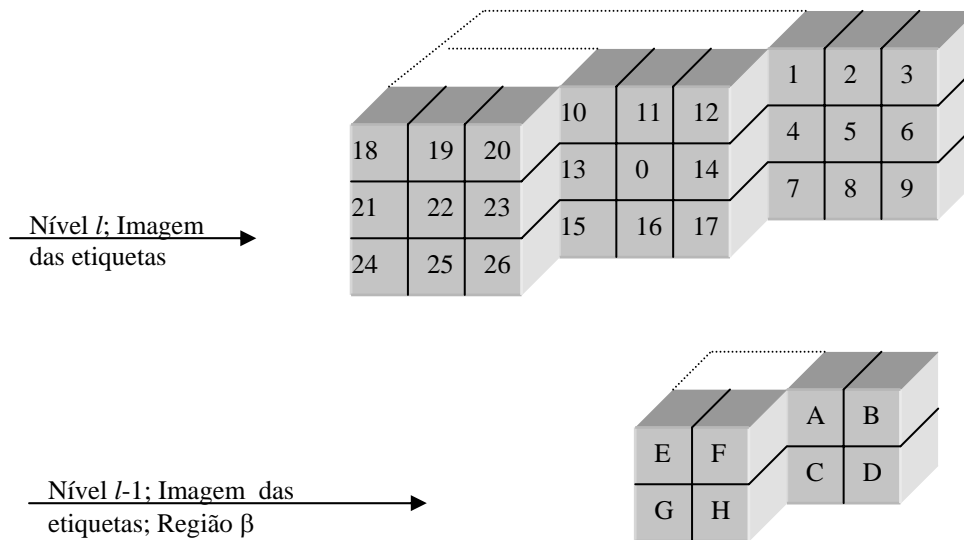


Fig. 4.6 – Etiquetas dos voxéis na região β_n .

Fronteira em:	Voxel(s) região β_n	Fronteira em:	Voxel(s) região β_n	Fronteira em:	Voxel(s) região β_n	Fronteira em:	Voxel(s) região β_n
1	A	8	C+D	15	C+G	22	E+F+G+H
2	A+B	9	D	16	C+D+G+H	23	F+H
3	B	10	A+E	17	D+F	24	G
4	A+C	11	A+B+E+F	18	E	25	G+H
5	A+B+C+D	12	B+F	19	E+F	26	H
6	B+D	13	A+C+E+F	20	F		
7	C	14	B+D+F+H	21	E+G		

Tab. 4.1 – Voxéis a considerar na região β_n .

4.2.3.2 - Restrição nas Direcções dos Filtros Borboleta 3D

Outra observação efectuada a partir das figuras 4.5c, 4.6 e tabela 4.1 é que aparentemente nem todas as direcções dos filtros actuam directamente os voxéis fronteiros. Assim, com o objectivo de diminuir o tempo de cálculo dos treze filtros, foi considerado que os filtros que actuam directamente cada voxel da região β_n são aquelas cuja direcção está mais próxima da direcção do voxel que é a origem da fronteira no nível l .

Foram escolhidos os sete filtros com as direcções adjacentes à do voxel que provocou a fronteira no nível l . É visualizado na figura 4.7 as direcções consideradas para o voxel A da região β_1 , e na tabela 4.2 as direcções dos filtros que foram consideradas as que afectam cada voxel da região β_n .

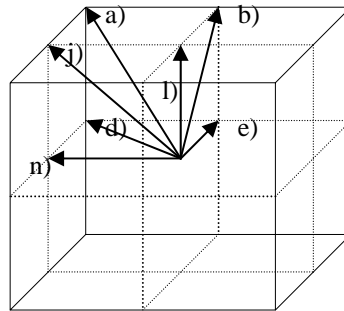


Fig. 4.7 – Direcções e sentidos dos filtros para o voxel A da região β_1 .

Voxel região β_n	Máscaras Borboleta 3D aplicadas						
A	a)	b)	d)	e)	j)	l)	n)
B	b)	c)	e)	f)	l)	m)	n)
C	d)	e)	g)	h)	l)	m)	n)
D	e)	f)	h)	i)	j)	l)	n)
E	e)	f)	h)	i)	j)	l)	n)
F	d)	e)	g)	h)	l)	m)	n)
G	b)	c)	e)	f)	l)	m)	n)
H	a)	b)	d)	e)	j)	l)	n)

Tab. 4.2 – Restrição do número de filtros a usar para cada voxel da região β_n .

4.3 – DESCRIÇÃO SUMÁRIA DO ALGORITMO EM 3D

Terminada a explicação pormenorizada da estimação da fronteira pode-se descrever o algoritmo completo de segmentação em três dimensões da seguinte forma:

1. **Construir a pirâmide** até ao nível pretendido utilizando a árvore octógona;

2. Classificação - Fazer o agrupamento no espaço de atributos unidimensional (nível de cinzento dos voxéis), recorrendo a um algoritmo sem restrições de conectividade, realocar os voxéis isolados bem como as classes pequenas ou perdidas;

2.1. Calcular o Agrupamento Centróide Local no nível mais alto da pirâmide;

2.1.1. Calcular todas as distâncias mínimas entre os centros das classes;

2.1.2. Criar a imagem das etiquetas em 3D, $I_\gamma(i, j, k; l_{\max})$;

2.2. Retirar as regiões insignificantes e os voxéis isolados (numa única passagem);

2.2.1. Determinar se um voxel é isolado dentro da sua vizinhança $3 \times 3 \times 3$ em $I_\gamma(i, j, k; l_{\max})$;

2.2.1.1. Se for voxel isolado será atribuído à classe adjacente mais representativa;

2.2.1.2. Caso existam duas classes ou mais com igual representatividade, o voxel será atribuído a classe mais próxima da sua etiqueta;

2.2.2. Determinar as regiões insignificantes em $I_\gamma(i, j, k; l_{\max})$;

2.2.2.1. Caso seja região insignificante, atribuir os seus voxéis como se tratassem de voxéis isolados - 2.2.1.1) e 2.2.1.2).

3. Estimação da fronteira - Melhorar gradualmente a resolução espacial pela projecção para baixo das etiquetas, e redefinir os voxéis fronteiros utilizando filtros do tipo Borboleta 3D;

3.1. Calcular a média e a variância das regiões da imagem 3D em $q(i, j, k; l)$;

3.2. Determinar se o voxel pertence à fronteira em $I_\gamma(i, j, k; l)$. Descer um nível na pirâmide;

3.2.1. Se o voxel não pertencer à fronteira será atribuída a etiqueta do seu pai em $I_\gamma(i, j, k; l-1)$;

3.2.2. Se o voxel pertence à região fronteira:

3.2.2.1. Determinar as duas classes mais representativas adjacentes ao voxel em $I_\gamma(i, j, k; l)$;

3.2.2.1.1 Caso exista mais de duas classes com igual representatividade considerar C_n classes representativas.

3.2.2.2. No nível $l-1$ reclassificar todos os voxéis filhos com as etiquetas dos pais em $I\gamma(i, j, k; l-1)$;

3.2.2.3. Determinar os voxéis pertencentes à região β_n utilizando a tabela 4.1 e deixando os restantes com a etiqueta do seu pai;

3.2.2.3. Calcular a dissemelhança, calculando todas as combinações duas a duas das classes representativas (C_n);

3.2.2.4. Determinar os pesos da(s) máscara(s) dos filtros Borboleta 3D (r e rr);

3.2.2.5. Aplicar os filtros com as máscaras respectivas em todas as direcções pré-definidas;

3.2.2.5.1. Se $r = 1$, os dois alvos do filtro ($r_2(\cdot) = r_1(\cdot)$) ficam com o valor do voxel central da máscara;

3.2.2.5.2. Caso contrário, determinar todos os alvos correspondentes a todas as máscaras existentes r_u ;

3.2.2.5.3. Calcular a diferença absoluta entre todos $\|C_n - r_u(\cdot)\|$;

3.2.2.5.4. Atribuir o voxel da região fronteira em $I\gamma(i, j, k; l-1)$ ao C_n , correspondente ao valor mínimo $\|C_n - r_u(\cdot)\|$;

3.3. Voltar para o passo 3.1) até chegar ao nível inicial ($l=0$) da pirâmide.

4.4 – TESTES E RESULTADOS EM 3D

4.4.1 - Análise de uma Imagem Sintética

Foi realizado um elevado número de testes e experiências sobre diferentes tipos de imagens sintéticas em três dimensões.

Para sintetizar os resultados e testes foi criada uma imagem, que tenta englobar o maior leque possível de situações. Esta imagem tem como dimensões 128x128x128, em que cada voxel é representado por um número em vírgula flutuante. Para visualizar a imagem é feita a truncagem para um inteiro entre 0 e 255.

A escolha destas dimensões para a imagem foi motivada pela simplificação da visualização dos resultados, porque na mesma linha permite ver três secções xy_n , yz_n e xz_n , com n a posição da respectiva secção relativamente à origem do eixo de coordenadas do cubo, figura 4.8 e 4.9a,b,c .

A imagem tem como características: dez regiões distintas, os valores das regiões estão compreendidos entre 0.0 (preto) e 255.0 (branco). É constituída por quatro rectângulos que dividem a imagem em camadas, quatro cubos inseridos nas diferentes camadas, uma esfera e um cone. As regiões têm um incremento do nível de cinzento de 25.0. O resultado da segmentação para a imagem sem ruído, considerando o nível mais alto da pirâmide $l=2$, (imagem $32 \times 32 \times 32$), pode ser observado na figura 4.9d,e,f.

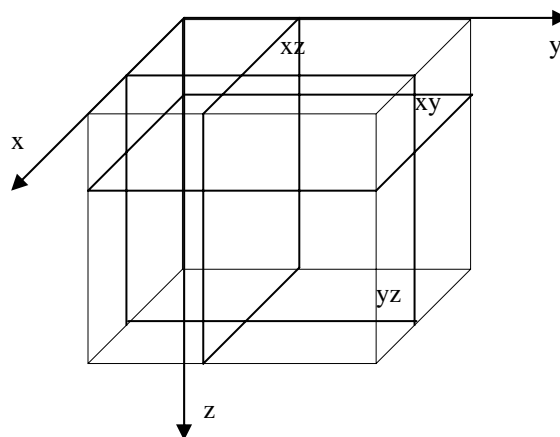


Fig. 4.8 - Representação das três secções apresentadas na imagem 3D.

Podemos verificar que a imagem sem ruído foi bem segmentada, e que a resolução espacial foi restaurada. Pode-se ainda observar na figura 4.10 em pormenor esta conclusão, para tal foi aplicado ruído aditivo Gaussiano com desvio padrão de 25, e foi realizada uma ampliação da zona do cone na fig 4.10a. De seguida construiu-se a pirâmide até ao nível $l=2$ (fig. 4.10b), observou-se que o cone perdeu resolução

(observe-se que os voxéis dos vértices desapareceram), no entanto esta foi restaurada de novo quando alcançamos o nível mais baixo da pirâmide (fig. 4.10c).

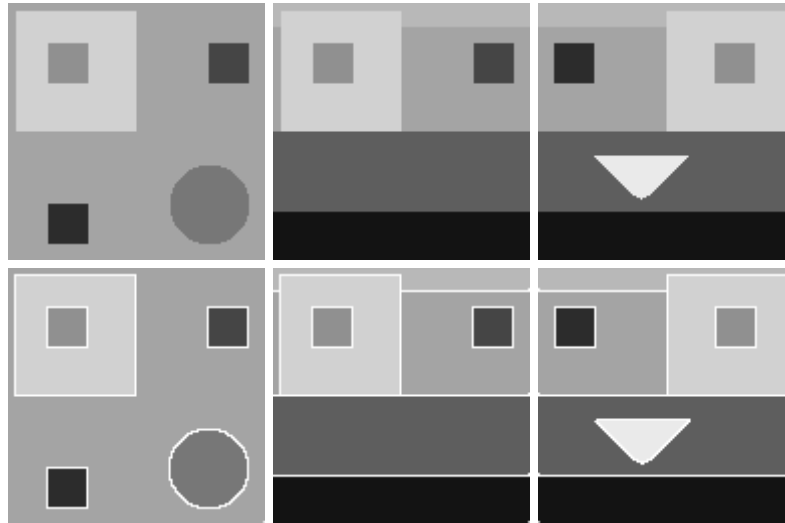


Fig. 4.9 - A imagem “Sintética” 3D. Da esquerda para a direita a) Secção xy_{25} ; b) Secção yz_{25} ; c) Secção zx_{25} ; em baixo imagem delimitada d) e) f).



Fig. 4.10 – Ampliação do cone ($\sigma=25$). Da esquerda para a direita a) imagem original; b) imagem etiquetas $l=2$; c) imagem etiquetas $l=0$.

São as imagens com ruído que normalmente temos de processar e é este tipo de imagem com que nos deparamos normalmente após a aquisição de dados (imagens) do meio ambiente. A seguir vai ser estudado o comportamento do algoritmo perante este tipo de imagens (fig. 4.11).

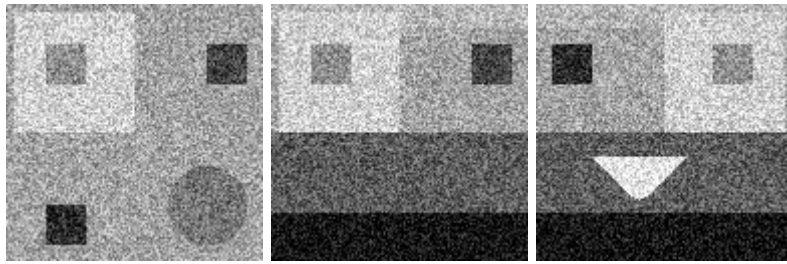


Fig. 4.11 - Imagem “Sintética” ($\sigma=25$). Da esquerda para a direita a), b), c).

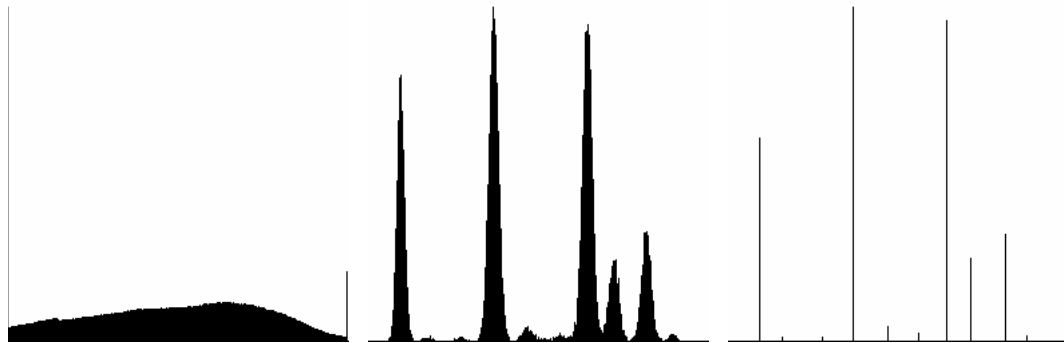


Fig. 4.12 – Histogramas da imagem “Sintética”. Da esquerda para a direita a) histograma original; b) histograma $l=2$; c) histograma da imagem das etiquetas.

Tal como já foi referido o primeiro passo foi a construção da pirâmide até ao nível $l=2$, a relação sinal ruído para a imagem original ($l=0$) é dada pela eq. 2.5), e varia entre $1 \leq \rho \leq 2.6$, sendo o menor valor obtido entre as duas camadas superiores (fig. 4.11b), em que a diferença da média entre as regiões é de 25 e o desvio padrão é de 25. O maior valor entre a segunda camada e o cubo nela inserido (fig. 4.11c), com a diferença da média das regiões de 65. No nível $l=2$ será aplicado o algoritmo de aglomeração e a relação sinal ruído varia entre $6 \leq \rho \leq 16$. Na figura 4.12 é visualizado o histograma da imagem original (fig. 4.11a), onde não se consegue observar as regiões onde vão surgir os centros das classes. No histograma no nível $l=2$ (fig. 4.12b) já é possível visualizar as regiões de onde vão surgir os centros das classes. O histograma da imagem das etiquetas resultantes da aplicação do algoritmo de aglomeração no nível $l=2$, e após a reclassificação dos voxéis mal classificados bem como dos voxéis isolados é apresentado na figura 4.12c.

Os resultados obtidos estão apresentados para a imagem das etiquetas na figura 4.13a), b), c), e para a imagem com os contornos delimitados na figura 4.13d), e), f).

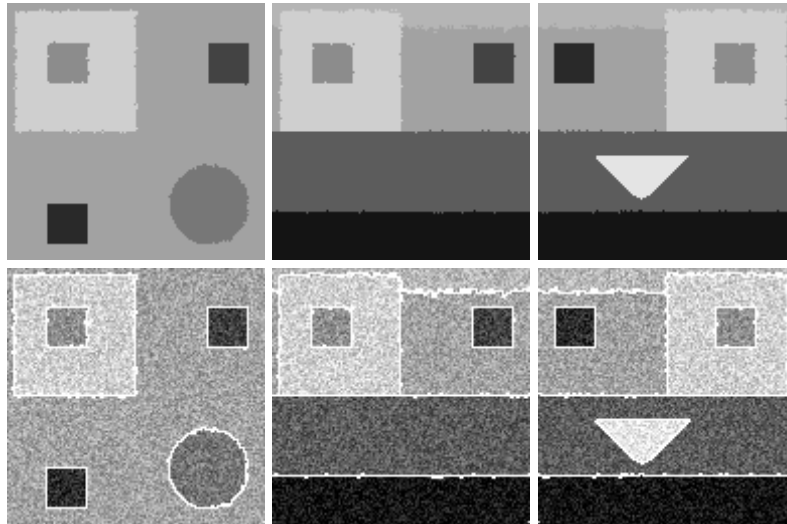


Fig. 4.13 – Resultados para a imagem “Sintética” ($\sigma=25$), em cima a), b), c), imagem das etiquetas, em baixo d), e), f), imagem com as regiões delimitadas a branco.

Pode ser observado que os resultados apresentam algumas falhas em alguns pontos, principalmente no contorno da esfera e onde o desvio padrão é maior ou igual à diferença mínima existente entre as diferentes regiões (≈ 25), mas no entanto todas as regiões têm os contornos completamente delimitados.

A seguir vai ser analisado o que acontece quando são alteradas algumas variáveis do algoritmo em função do ruído (desvio padrão (**d.p.**) da imagem original), o critério de comparação utilizado foi o número de voxéis mal classificados (**v.m.c.**), o processo de cálculo e os resultados apresentados na forma gráfica que podem ser observados no Apêndice B.

Diferente número de voxéis a considerar para a região fronteira β : verificou-se que a estratégia considerada de calcular automaticamente os voxéis pertencentes à

região β_n , $n=1,2$ no que concerne ao cálculo de todos os voxéis, foi uma opção que revelou sucesso. Esta melhoria pode ser notada em dois aspectos: nos resultados obtidos em função do d.p. e no tempo cálculo do algoritmo (Tab. 1 - apêndice B). Os resultados comparativos em função de d.p. podem ser observados no gráfico 1 - "Número de voxéis pertencentes à região fronteira" do apêndice B.

Quantidade diferente de direcções do filtro a considerar para cada voxel $\in \beta$: ao ser considerado sete em vez das treze filtros e ao serem consideradas apenas as direcções da tabela 4.2, os resultados finais são muito semelhantes, sendo a maior diferença obtida quando o d.p. é 40, e apresentando nesta situação apenas uma diferença de 80 v.m.c., o que não é significativo dentro do total de voxéis de fronteira a classificar (33680), no entanto, o algoritmo tornou-se mais rápido (Tab. 1 - apêndice B). Os resultados comparativos das duas estratégias em função do ruído podem ser observados no gráfico 2 - "Número de direcções do filtro" do apêndice B.

Alteração da quantidade de classes representativas (C_n): como era de prever à medida que aumentamos o número de classes consideradas representativas para o cálculo da dissemelhança o resultado piora e o tempo de cálculo aumenta (pode-se observar a comparação entre o tempo de cálculo considerando C_n 2 e 3 na Tab.1 do apêndice B). Isto é devido à coincidência entre os valores intermédios provenientes das máscaras que não estão correctamente direccionadas com os valores de C_n . Os resultados comparativos em função do ruído podem ser observados no gráfico 3 - "N.º de classes significativas para o cálculo do voxel fronteiro" do apêndice B.

Alteração do número de regiões consideradas insignificativas, nas diferentes camadas da imagem: foram testadas várias estratégias, encontrando-se as mais representativas apresentadas no gráfico 4 - "Região insignificante" do apêndice B. Foi verificado que os melhores resultados foram obtidos quando em todos os níveis (no nível da classificação mas também nos níveis inferiores $l=2,1,0$), da pirâmide retirávamos regiões insignificantes com trinta e seis voxels adjacentes, e o pior resultado quando não considerávamos a existência de regiões insignificantes em nenhum nível. Foi ainda verificado que entre estas duas estratégias a pior diferença era de 200 v.m.c.. Esta diferença não é muito significativa no total de voxels fronteira a classificar, mas a diferença de tempo global na execução do algoritmo era significativa aproximadamente: 4.5 vezes mais (ver Tab. 1 - apêndice B). Desta forma, foi decidido utilizar como estratégia no nível mais alto da pirâmide retirar as regiões insignificantes com um máximo de 27 voxels, e nas restantes regiões não foi considerado a existência de voxels mal classificados nem de regiões insignificantes. Utilizando esta estratégia, na pior situação (d.p. de 40), foi obtido uma diferença 40 v.m.c. em relação ao melhor resultado referido anteriormente, isto com um acréscimo mínimo no tempo despendido na execução do algoritmo (1.02 vezes mais), como podemos observar na tabela 1 do apêndice B.

Alteração do tamanho da máscara: foram testadas as máscaras $3 \times 3 \times 3$ e $5 \times 5 \times 5$. Tal como previsto, os resultados para a máscara de maior dimensão foram melhores do que para a de menor dimensão. No entanto, a diferença de resultados não foi significativa, sendo para o d.p. de 40 obtida uma diferença de 200 v.m.c. Os resultados comparativos em função do ruído, podem ser observados no gráfico 5 -

"Dimensão da Máscara" do apêndice B e a comparação de tempos na tabela 1 do mesmo apêndice.

Tempo de execução: o tempo de execução do algoritmo está directamente relacionado com a qualidade. Assim para a qualidade das imagens da figura 4.13 e para todas as que vão ser apresentadas a seguir, foram consideradas as seguintes características: filtros de dimensão 3x3x3, número de direcções da máscara sete, considerar para o cálculo da região insignificante 27 voxels apenas no nível mais alto da pirâmide, quantidade de voxels de 3 para considerar a região significativa, os voxels pertencentes à região fronteira serão determinados automaticamente (ver Tab. 4.1).

Em termos gerais não é possível definir exactamente o tempo que o algoritmo demora a executar, devido a três factores:

O primeiro é que, quando a função de dissimilaridade tem valores muito elevados os voxels da fronteira são directamente atribuídos, este procedimento é extremamente rápido, pois evita a aplicação das máscaras.

O segundo é o tipo de fronteiras existentes na imagem. Quanto menos voxels pertencentes à região fronteira (β_n) forem tratados, menor é o tempo despendido no cálculo.

O terceiro é a quantidade de regiões adjacentes aos voxels fronteira consideradas significativas. Existem três situações: A primeira, a mais lenta será aquela onde aparece n (de C_n) com um valor maior que dois (sendo esta também a mais improvável de acontecer). A segunda, quando n (de C_n) for igual a dois que é o caso mais normal de acontecer. A terceira, a mais rápida quando a segunda região não

atingir o critério de ser considerada significativa (≥ 3 voxels) porque neste caso o voxel pertencente a β_n é atribuído automaticamente com a etiqueta da primeira.

Para demonstrar a qualidade em termos de velocidade e qualidade foi implementado um outro algoritmo bastante simples:

1. Alisamento do ruído da imagem por um algoritmo que preserva as fronteiras-AGWF para três dimensões aplicado 10 vezes (ver Anexo A);
2. Aplicação do Algoritmo de Agrupamento – Agrupamento Centróide Local;
3. Retirar as regiões insignificantes (< 27 voxels conectados).

A escolha desta estratégia e método de alisamento do ruído da imagem como referência de comparação, é porque o algoritmo AGWF está largamente documentado e testado em diferentes literaturas [Du Buf 90], tanto em termos de tempo (CPU) como de aplicação e características. O segundo e terceiro passo foram os mesmos considerados no algoritmo proposto no capítulo 4.3.

Os resultados em termos de v.m.c. (v.d. Gráfico 6 - "Comparação de algoritmos" do apêndice B), excedem muito mais do dobro em qualquer dos níveis de ruído (d.p. entre 0 e 20), sendo as fronteiras completamente mal classificadas para níveis de ruído com d.p. acima de 20, de salientar ainda que estes resultados pioram muito caso se exclua o ponto 3.

Uma única passagem do algoritmo AGWF é 2.5 vezes mais lenta (ver tab. 1 do apêndice B) que toda a execução do algoritmo referido no capítulo 4.3 (para as imagens cujas secções estão apresentadas na figura 4.11).

4.4.2 – Resultados obtidos com o volume de dados do Projecto ISACS

Antes de apresentar os resultados obtidos com o volume de dados do projecto ISACS, será descrito muito sucintamente o processo de criação do volume, e o porquê das diferentes resoluções das secções do volume (figura 4.15).

O volume é constituído por dados obtidos em 1D. Estes dados (*pings*) são adquiridos utilizando sonar e têm algum processamento como a compensação do movimento do barco, redução da influência do ruído não correlado, etc.[Nikolov 97b].

Um *ping* corresponde a uma única coluna de dados com a dimensão n . A criação de uma imagem 2D ($n \times m$), é conseguida pela aquisição sequencial de *pings* (P_k a P_{k+m}). Uma imagem (2D) obtida por este processo pode ser observada na figura 3.1 e 4.15c.

Cada *ping* está localizado no mapa de *ping* (*pingmap*) em relativamente à sua longitude e latitude (figura 4.14a). Esse mapa é criado com a ajuda de GPS². A razão deste mapa não formar uma malha deve-se ao facto dos dados serem adquiridos com a ajuda de um barco, e da dificuldade de manter esse barco num percurso linear. Assim na altura da conversão do mapa de *pings* para o espaço tridimensional (volume) surgem lacunas, estas podem ser observadas na figura 4.14c, figura esta que corresponde à secção do mapa de *pings* representada na figura 4.14b.

Estas lacunas têm de ser preenchidas antes da aplicação do algoritmo de segmentação. Na altura da entrega desta dissertação está a decorrer o estudo e testes relativamente ao melhor processo para preencher as lacunas.

Uma vez que este estudo não faz parte dos objectivos da dissertação, essas lacunas foram preenchidas com o algoritmo mais simples de interpolação - interpolação linear (1D).

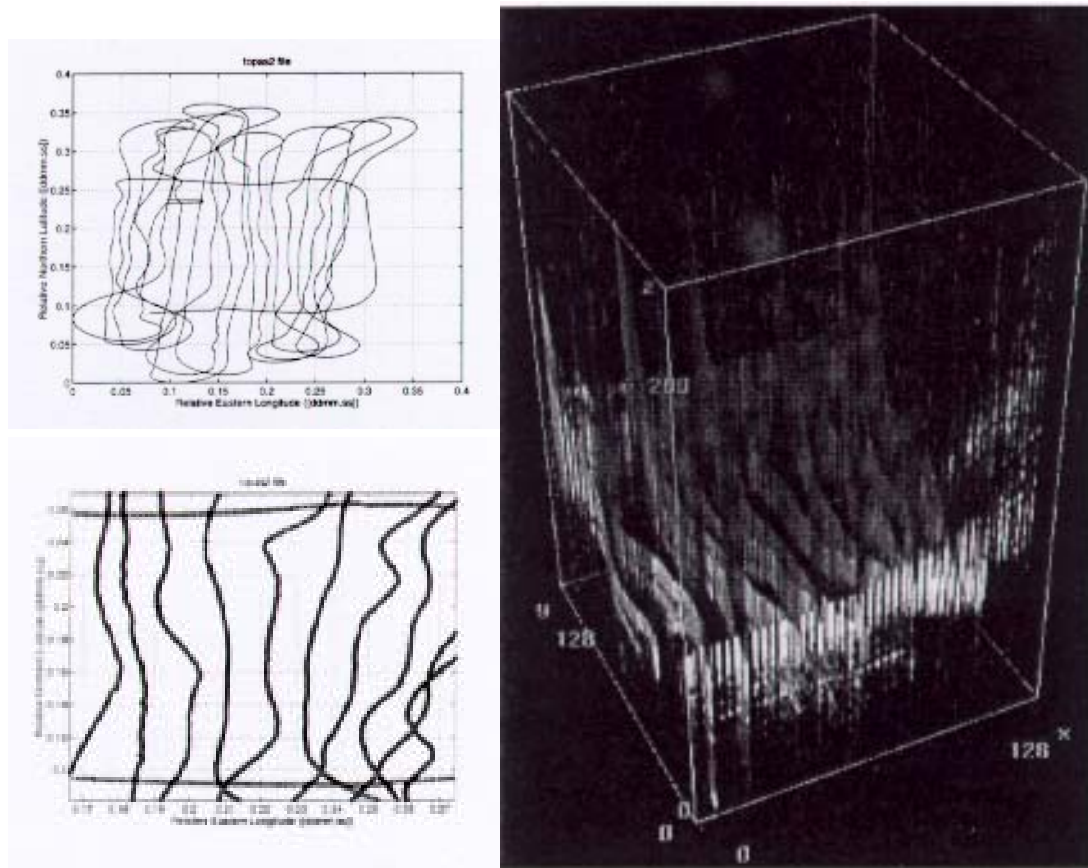


Fig. 4.14 - Da esquerda para a direita: Em cima: a) mapa de *pings*; Em baixo: b) secção do mapa de *pings*; c) volume de dados correspondente ao mapa de *ping*.

Figuras extraídas de [Nikolov 97b].

Com esta decisão obteve-se um volume de dados com as características genéricas do volume final, onde foi possível aplicar o algoritmo de segmentação e tirar conclusões. As diferentes resoluções apresentadas na figura 4.15 reflectem este facto, onde a secção xz (figura 4.15c) apresenta os dados obtidos dos *pings* e a secção xy e yz

² GPS é um receptor de rádio especializado e desenhado para receber sinais de rádio transmitidos por satélites e calcular a sua posição baseado-se nessa informação.

(figura 4.15a,b) apresentam imagens os dados resultantes da aplicação da interpolação.

Apresenta-se agora os resultados obtidos sobre o volume de dados do projecto ISACS -"topas3D_simul". O algoritmo aplicado foi o proposto na secção 4.3 “Descrição Sumária do Algoritmo em 3D”, para um $l_{\max} = 2$. As dimensões da imagem analisada é de 256x256x256.

Apresenta-se na figura 4.15 três secções efectuadas sobre a imagem *topas3D_simul*, e na figura 4.16 os histogramas: da imagem original, da imagem no nível $l=2$ e histograma das etiquetas para $l=2$. Pode-se observar pela análise do histograma 4.16b, que neste nível ainda se torna difícil determinar as diferentes regiões da imagem, e conseqüentemente o algoritmo de agrupamento demora mais tempo a convergir. Isto leva a concluir que este não é o nível ideal para aplicar o algoritmo de agrupamento, havendo a necessidade de subir mais na pirâmide para melhor obter os centros das classes. Quanto a qualidade dos resultados obtidos pelo algoritmo, estes só podem correctamente interpretados por um perito em acústica submarina. Pode-se observar mais resultados no Apêndice C.

Pode-se observar na figura 4.17a, b, c) os cortes respeitante à imagem das etiquetas no nível $l=2$ e 4.17d, e, f) ao nível $l=1$.

Pode-se observar na figura 4.18 os resultados finais obtidos; em cima a imagem das etiquetas e em baixo a imagem original com os contornos delimitados para um $l_{\max} = 2$, e no Apêndice C, os resultados para $l_{\max} = \{1, 2, 3\}$. Foram descobertas nesta imagem (3D) cinco classes (regiões) que foram delimitadas com sucesso.

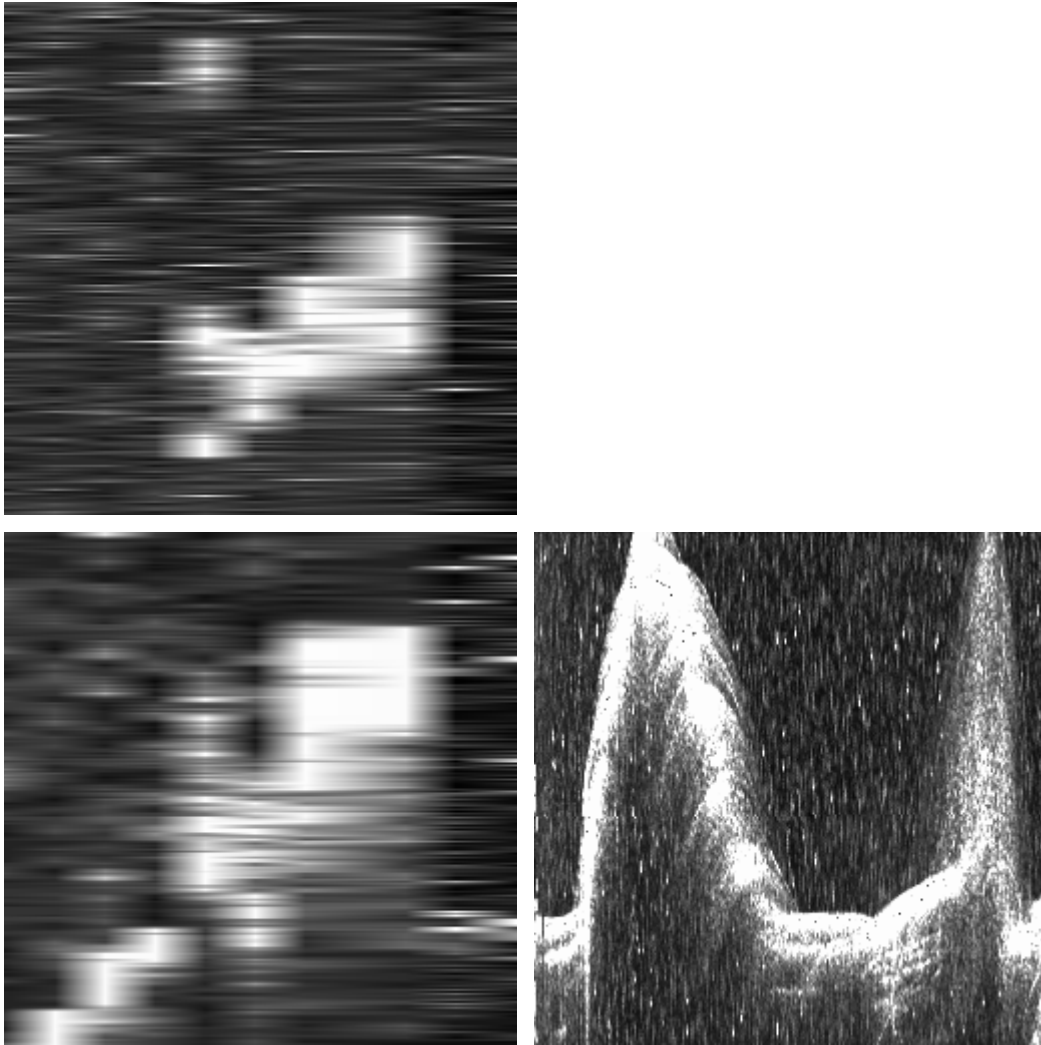


Fig. 4.15 – Secção sobre a Imagem “Topas3D_simul” em 3D. Em cima a) secção xy_{100} ; Em baixo esquerda b) corte yz_{100} ; direita c) corte xz_{100} .

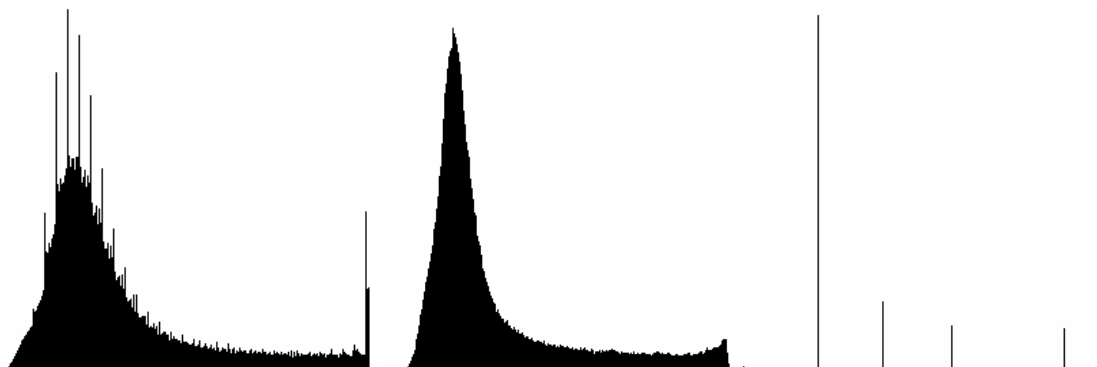


Fig. 4.16 –Histograma da imagem “Topas3D_simul” em 3D. Esquerda para direita a) imagem original; b) original para $l=2$; c) etiquetas $l=2$.

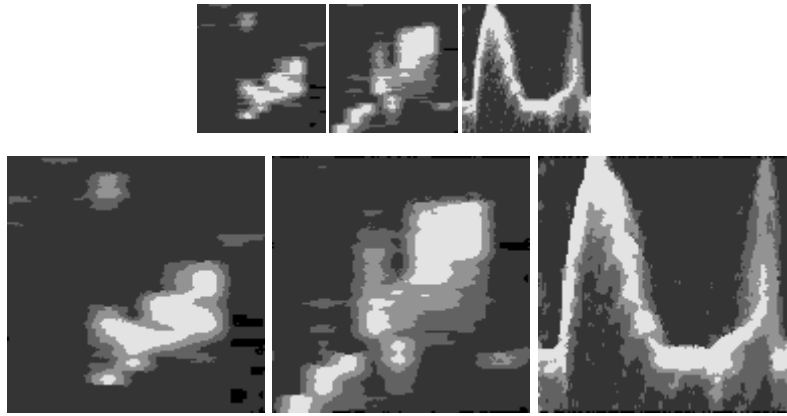


Fig. 4.17 – Secção sobre a Imagem “Topas3D_simul” em 3D. Em cima da esquerda para a direita a), b), c) imagens das etiquetas para $l=2$. Em baixo d), e), f) imagens das etiquetas para $l=1$.

Torna-se complicado para um observador não treinado criticar os resultados deste tipo de imagens. Pois é difícil concluir se o número de regiões segmentadas for inferior ou superior ao esperado, e se a imagem resultante apresenta regiões pouco representativas, ou se o nível máximo para a pirâmide (neste caso $l=2$) nos dá toda a informação relevante sobre as diferentes regiões da imagem. No entanto, mesmo um observador não treinado em imagens de acústica submarina pode concluir que os resultados são muito bons e coerentes, isto não só pela análise das secções do volume apresentado, mas por uma análise conjunta de todas as secções adjacentes as apresentadas em qualquer dos eixos. Pode-se concluir que foram determinadas as regiões visualmente mais representativas e fazendo uso de uma ampliação da imagem das etiquetas em comparação com a imagem original pode-se concluir que os contornos das regiões (fig. 4.18 d), e), f)) foram determinados com muita precisão.

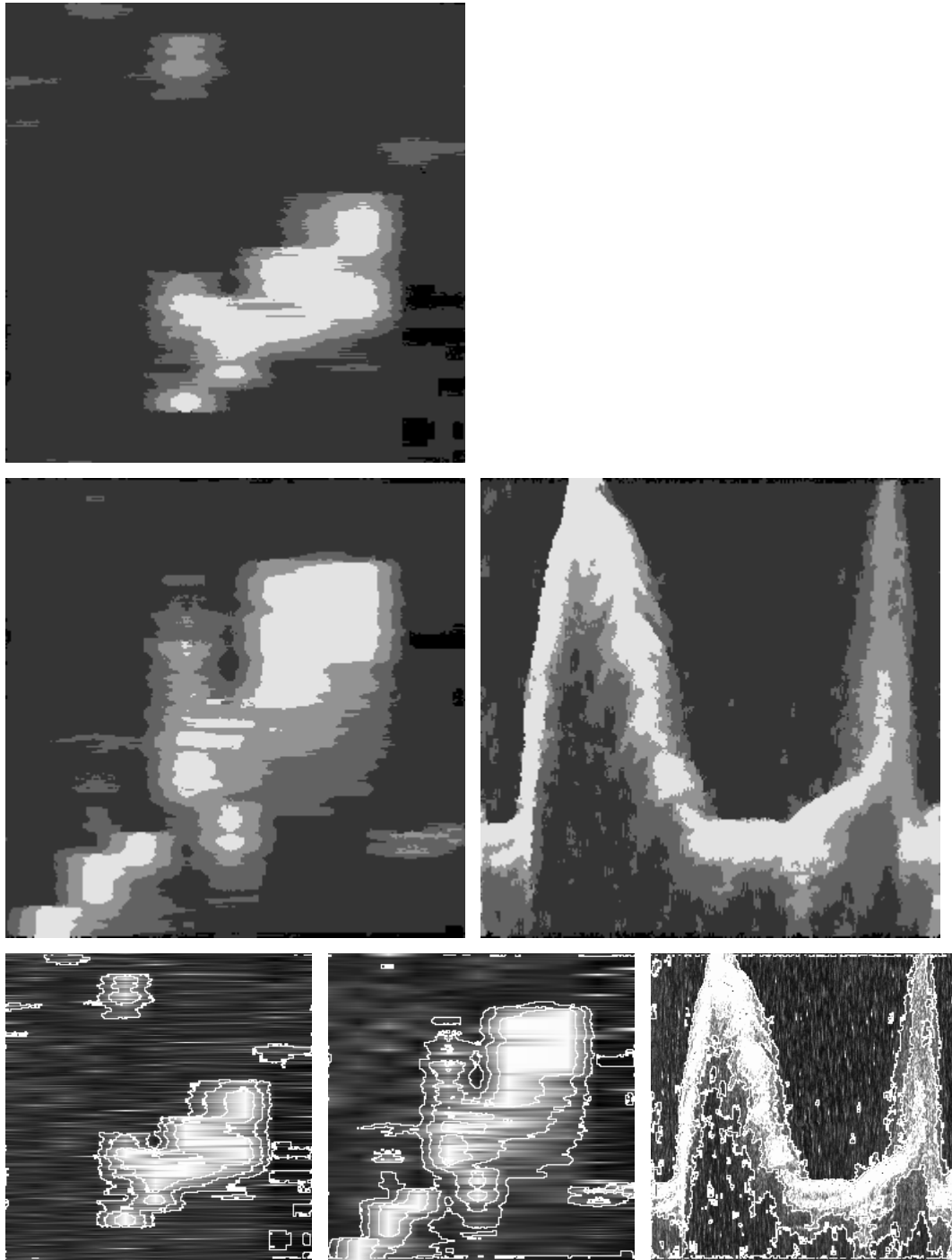


Fig. 4.18– Secção sobre a Imagem de resultado em 3D para $l=2$. Em cima imagem das etiquetas a),b),c). Em baixo imagem original com os contornos delimitados d),e),f).

CONCLUSÕES

5.1 – ANÁLISE CRÍTICA DO TRABALHO

A análise crítica deste trabalho vai concentrar-se em cinco pontos: método utilizado para efectuar o agrupamento, processo pelo qual foi estimada a fronteira, erro na estimação da fronteira, volume de dados e análise dos resultados obtidos.

Relativamente ao **método de agrupamento** utilizado, os autores Schroeter e Bigün apresentam em [Schroeter 95] a comparação de resultados utilizando diferentes algoritmos de agrupamento, onde afirmam que obtêm resultados muito semelhantes, apenas com uma pequena variação do número das classes. Pode-se assim concluir que a escolha do algoritmo de agrupamento não é muito crítica. A escolha recaiu sobre o Agrupamento Centróide Local [Spann 85; Wilson 88] num espaço de atributos unidimensional, justificando-se pelo atrás referido e por se pretender implementar um algoritmo robusto e rápido que tenha por base os níveis de cinzento da imagem. Experiências confirmaram os resultados de Spann e Wilson, pois verifica-se que com um histograma "difícil" em l_{\max} foi possível obter um bom agrupamento e uma rápida convergência. Os resultados também confirmam os de Schroeter e Bigün pois foi conseguida também uma boa segmentação, ficando apenas a faltar discutir os resultados com os peritos em acústica submarina.

Quanto ao processo usado para **estimar a fronteira**, a decisão de não se seguir em completo a filosofia proposta por Schroeter e Bigün, deve-se ao facto que em duas dimensões o método explicado nesta dissertação ser mais rápido que o proposto pelos autores atrás referidos, apresentando resultados finais semelhantes, e assim se esperar que em três dimensões o mesmo deva acontecer. Põe-se no entanto algumas restrições quanto a uma conclusão definitiva sobre qual dos métodos será mais rápido em 3D. Esta ressalva é feita devido a dois factores: O primeiro devido à complexidade de formas que podem surgir em três dimensões. O segundo quando existe a necessidade da obtenção de uma ainda melhor precisão na definição das fronteiras, recorrendo para tal a um número maior de direcções dos filtros e/ou de máscaras de dimensões maiores.

Uma análise sistemática do **erro na estimação da fronteira**, tendo em consideração o algoritmo aqui proposto, é uma tarefa quase "impossível" devido à dependência da forma da vizinhança e da dificuldade da análise da propagação dos erros na pirâmide. Uma análise pode ser efectuada no entanto para o caso de imagens sintéticas, onde para um ruído Gaussiano com um desvio padrão máximo testado de 40 (e com uma relação de sinal ruído que varia entre as diferentes camadas da imagem de $0.6 \leq \rho \leq 1.6$), o número de voxéis fronteira mal classificados tem um valor percentual de 9%, para a imagem sintética que tem por base a apresentada em 4.4.1 (com um desvio padrão de 25) obteve-se um erro de 0.9% de voxéis mal classificados. Este erro pode ser aceite como um bom resultado para um primeiro estudo efectuado sobre o assunto - Segmentação 3D e como base para um trabalho futuro.

Relativamente ao **volume de dados** utilizado para testes (topas3D_simul), ser uma aproximação “grosseira” de um volume de acústica submarina em 3D, espera-se que

este não vai afectar em nada os resultados que se vai obter a quando da disponibilização dos volumes finais do projecto ISACS, isto porque o algoritmo não está dependente da resolução do volume, funcionando correctamente para qualquer volume aplicado. Estima-se que os volumes finais resultantes do projecto ISACS em termos genéricos tenham as mesmas características, mas com uma melhor resolução depois da interpolação final aplicada.

Outro ponto a analisar, e talvez o mais relevante, é o significado dos **resultados obtidos**. Tal como foi referido no final dos capítulos III e IV, idealmente o significado dos resultados só deverá ser analisado por uma pessoa especializada em imagens de acústica submarina, por esta estar sensibilizada para este tipo de imagem, sendo-lhe mais fácil a tarefa de conseguir ou não extrair informação relevante das imagens finais. No entanto relativamente à qualidade dos resultados, as regiões foram delimitadas tal como o observador humano as delimitaria, podendo-se assim concluir que os resultados são muito bons, podendo no entanto relativamente a quantidade de informação que apresentam ter alguma redundância. Para melhorar a quantidade de informação transmitida pelos resultados propõe-se ao perito em acústica submarina um esquema iterativo onde se subirá até um nível consideravelmente alto da pirâmide (quanto mais alto se subir na pirâmide mais genérico será o resultado das regiões obtidas), nesse ponto será seleccionada a zona da imagem onde se pretende obter maior informação, e apenas nessa área será aplicado o algoritmo para um l_{\max} menor (quanto menos se subir na pirâmide maior será a resolução das regiões).

5.2 – CONCLUSÕES

O algoritmo apresentado resolve o problema da segmentação de imagem não supervisionada em 3D como era o objectivo proposto. Para evitar o problema do

conhecimento à priori das classes, estas são obtidas à custa da resolução espacial, por um algoritmo de agrupamento aplicado numa pirâmide multi-resolução, baseada numa árvore octógona. Uma vez a fronteira localizada no nível mais alto da pirâmide, a procura será limitada à fronteira do nível resolução seguinte (nível mais baixo). O esquema para a sua obtenção é adaptativo. Em cada nível da pirâmide faz-se uma selecção e uma estimação aos voxels fronteiros, baseando-se na comparação entre as regiões fronteiras e os alvos de um filtro Borboleta em diferentes direcções.

Relativamente a alguns pontos específicos do trabalho são apresentadas algumas conclusões:

Quanto ao **nível da pirâmide** podemos concluir que:

- Existe independência dos centros das classes relativamente ao nível da pirâmide ($l=\{1, 2, 3\}$), isto é, em qualquer dos níveis testados os centros das classes determinados são praticamente coincidentes e a quantidade de classes é igual.
- Quanto mais se subir na pirâmide menor é a resolução das fronteiras obtidas na imagem final e mais genérica é a informação obtida sobre as regiões.
- Quanto mais se subir na pirâmide menor é a quantidade de regiões insignificantes que surgem no resultado final, e mais fácil se torna a determinação dos centros das classes.
- Quanto menos se subir na pirâmide melhor é a resolução da fronteira, e maior é a quantidade de regiões pequenas ou parasitas que aparecem na imagem final.
- Fica mais lento retirar as regiões consideradas pequenas ou insignificantes do que a subir mais um nível na pirâmide.

Relativamente aos **parâmetros do algoritmo** que influenciam o resultado final:

- Verificou-se que as fronteiras apresentam melhor definição ao considerar o cálculo automático dos voxels pertencentes à região fronteira β (restringindo o número).
- Ao considerar apenas sete das direções do filtro para cada voxel fronteiro apresenta resultados finais muito semelhantes as considerarmos todas as direções do filtro.
- Ao aumentar o número de classes representativas (C_n) para o cálculo da dissemelhança, piora muito o resultado em termos da qualidade da fronteira.
- Os melhores resultados foram obtidos quando em todos os níveis da pirâmide retirávamos regiões insignificantes, e os piores resultados foram obtidos quando não considerávamos a existência de regiões insignificantes em qualquer nível.
- A estratégia de retirar as regiões insignificantes com um máximo de 27 voxels no nível mais alto da pirâmide, verificou-se a ideal, porque fazia desaparecer os voxels mal classificados nesse nível bem como a sua propagação para os restantes níveis.
- Para máscaras de dimensões maiores, as fronteiras obtidas apresentam melhor definição (precisão).

Quanto aos **resultados finais obtidos**, pode-se concluir que:

- Os resultados do algoritmo em três dimensões são considerados muito bons, uma vez que foi possível determinar as diferentes regiões da imagem e delimitá-las tal como o observador humano as delimitaria, isto não só pela análise das secções (imagens) apresentadas, mas por uma análise conjunta de todas as secções adjacentes em qualquer das direções dos eixos.

- Pode-se também concluir que o algoritmo é "imune" ao tipo de ruído que as imagens subaquáticas contêm, sendo suficiente para tal subir até ao nível $l_{\max} = 3$ da pirâmide.

5.3 – TRABALHO FUTURO

- Aplicação de uma melhor interpolação aos dados provenientes do projecto ISACS e consequente obtenção de melhores imagens de testes.
- Discutir com um perito em acústica submarina, a escolha do nível ideal de segmentação, bem como do processo de retro-alimentação (previsto para Maio de 1998).
- Melhorar o sistema: Extração - Segmentação - Visualização 3D iterativa, a partir da observação de dados reais.
- Implementar o algoritmo de agrupamento num espaço multi-atributos, comparando resultados.

**APÊNDICE
A**

***RESULTADOS DA SEGMENTAÇÃO
EM DUAS DIMENSÕES***

Na página seguinte podem ser observados os resultados obtidos após aplicação do modelo de Segmentação em 2D, foram utilizadas imagens de acústica submarina obtidas a partir da base de dados ISACS.

Na linha superior encontram-se as imagens de originais de dimensões 256x256. Para cada imagem, em cada coluna encontra-se a imagem das etiquetas e por baixo desta a respectiva imagem original com as fronteiras sobrepostas. Nas colunas, de cima para baixo, a partir da segunda linha, o nível mais alto da pirâmide vai aumentando, sendo respectivamente $l_{\max} = \{1, 2, 3\}$.

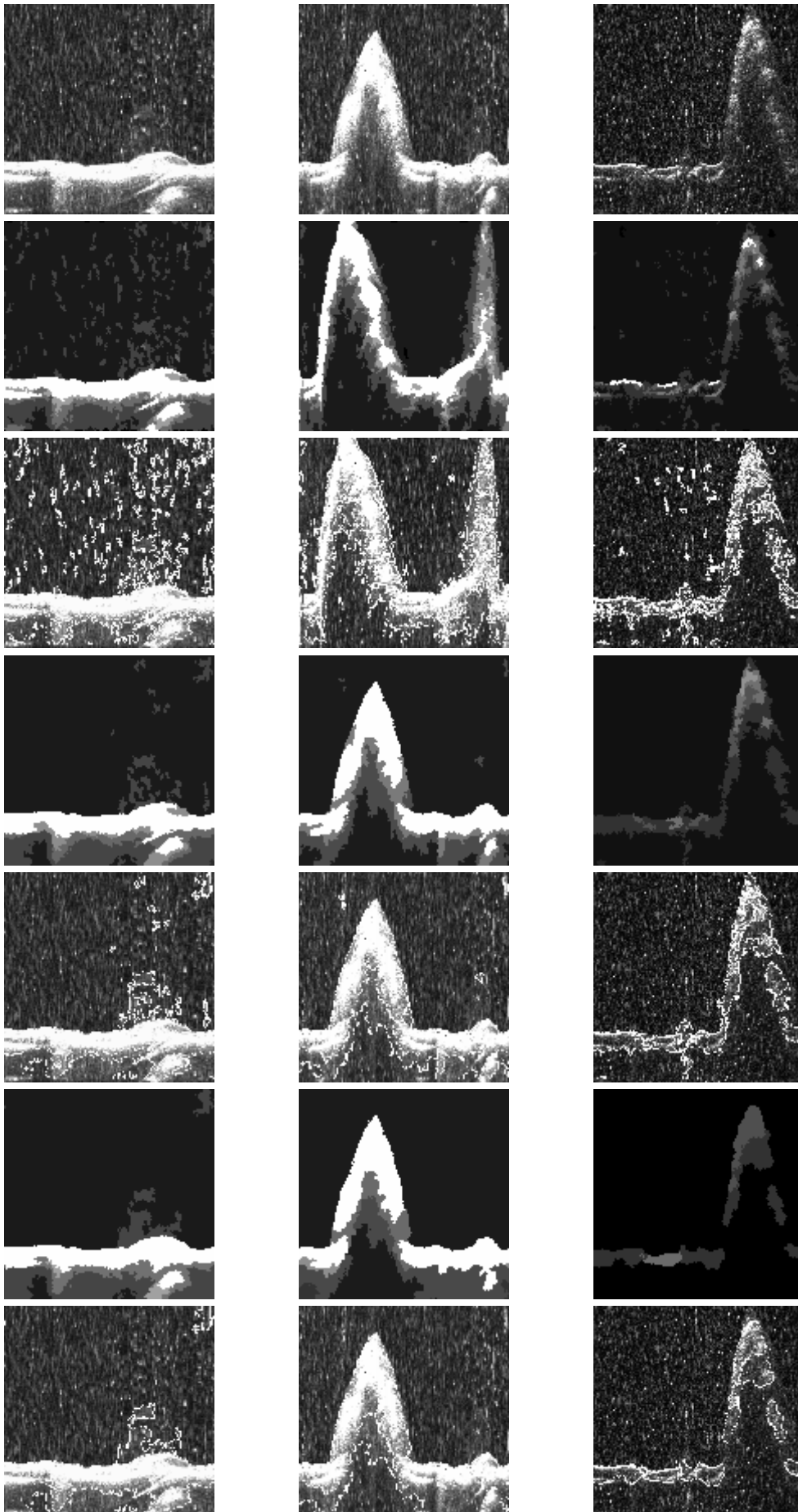


Fig. A.1 –Resultados para 2D.

**APÊNDICE
B**

**GRÁFICOS DE TESTES
EM TRÊS DIMENSÕES**

Foram realizados um conjunto de testes com a finalidade de salientar e demonstrar alguns pormenores do algoritmo implementado, os resultados estão representados sobre a forma de gráficos. Os teste foram executados sobre a imagem sintética em três dimensões referida no capítulo 4.4.1 e cujas secções estão representadas na figura 4.10a,b,c - Imagem “Sintética” 3D.

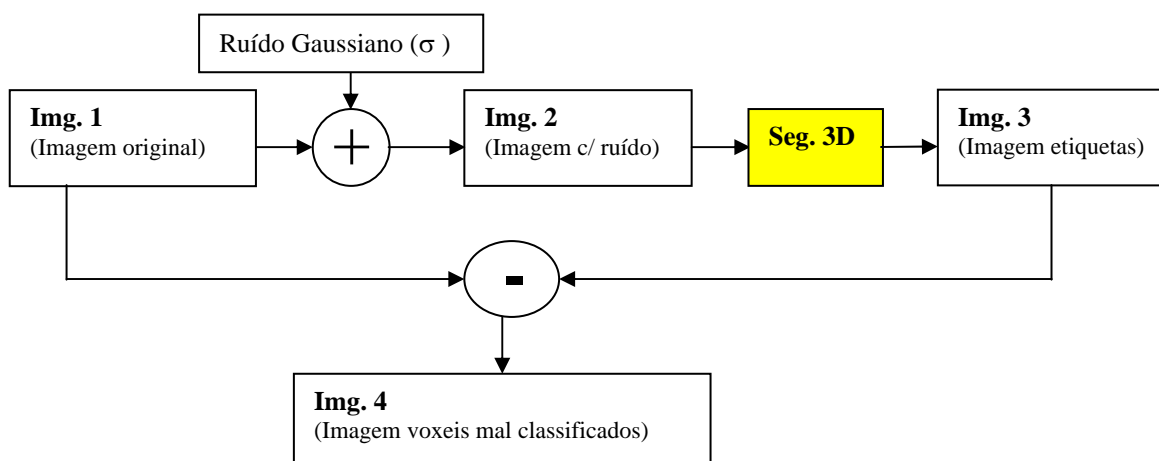
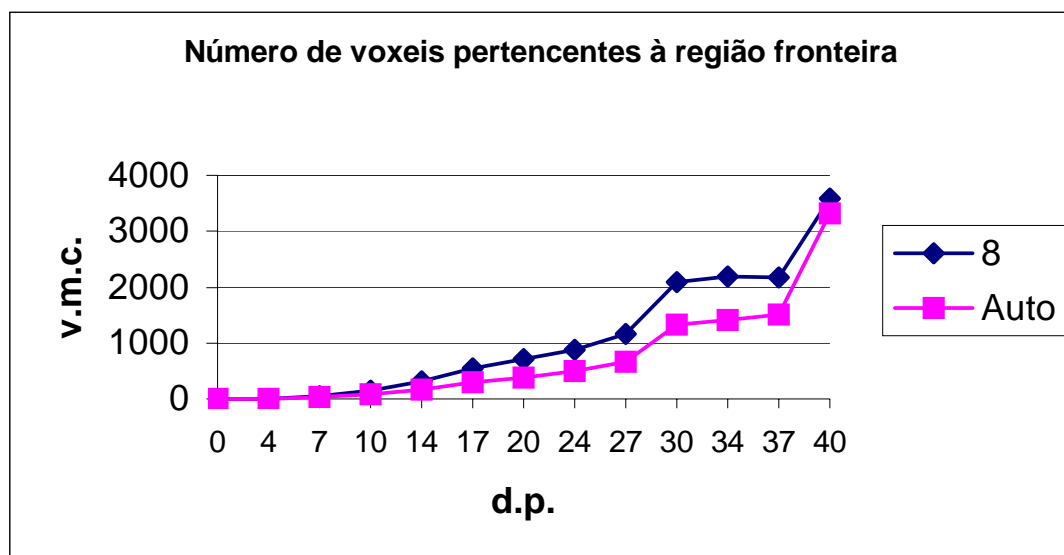


Fig. B.1 - Esquema para determinar **v.m.c.**

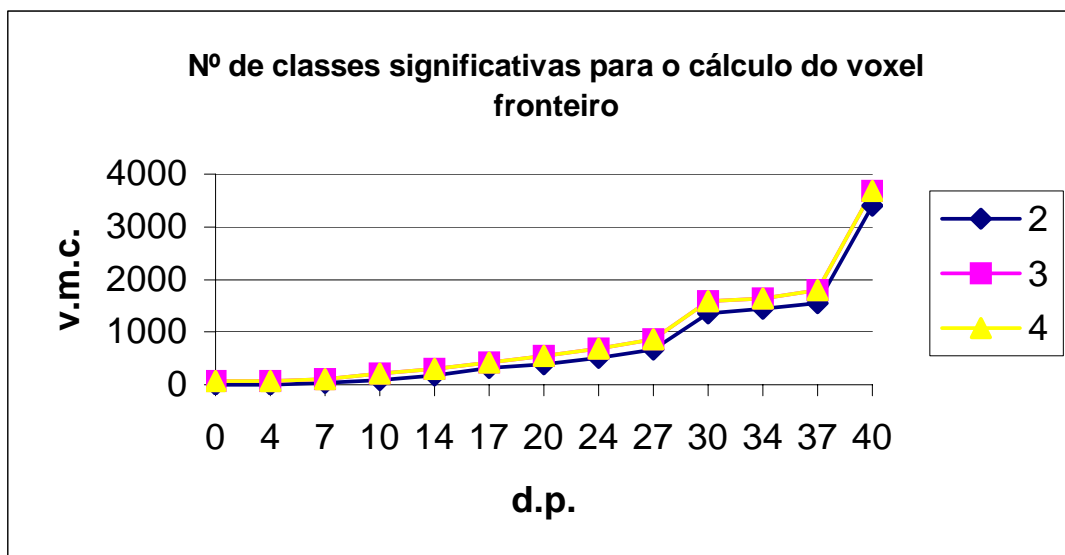
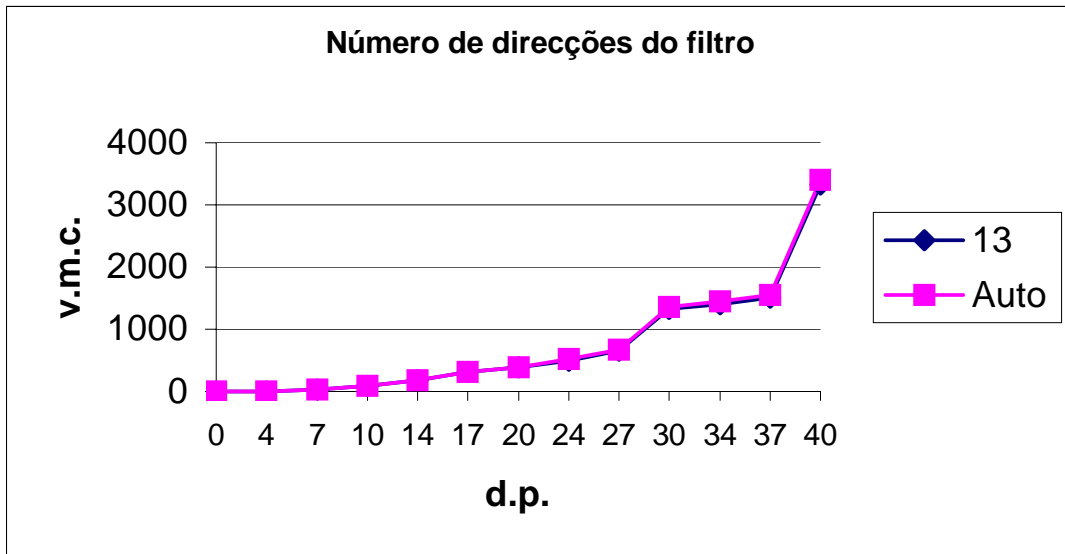
Os teste consistiram na comparação da imagem original sem ruído (Img. 1) com a imagem das etiquetas. A imagem das etiquetas foi obtida após a aplicação do algoritmo de segmentação em 3D (Img. 3). Às imagens de entrada para o referido algoritmo (Img. 2), foi adicionado ruído Gaussiano, com diferentes valores de desvio padrão (**d.p.**). O termo de comparação entre os diferentes pontos do algoritmo foi o somatório de voxéis mal classificados (**v.m.c.**) resultantes da subtracção da Img. 1

pela Img. 3. O esquema que ilustra a realização de cada um dos testes pode ser visualizado na figura B.1. Os tempos que cada teste demorou a ser executado num computador pessoal – *Pentium* 166Mhz estão descritos na tabela 1 no final deste apêndice.

Todos os valores obtidos obedecem às seguintes condições, excepto se for referido ao contrário, tamanho da máscara Borboleta 3D: 3x3x3; n.º de direcções do filtro é 7 (Auto), seguindo o proposto na Tab. 4.2; região insignificante (só no nível mais alto da pirâmide com o n.º de voxels < 27); imagem de partida 128x128x128 com $l_{\max} = 2$; n.º de classes significativas adjacentes ao voxel fronteiro é 2; n.º de voxels fronteira pertencentes à região β_n são determinados automaticamente.



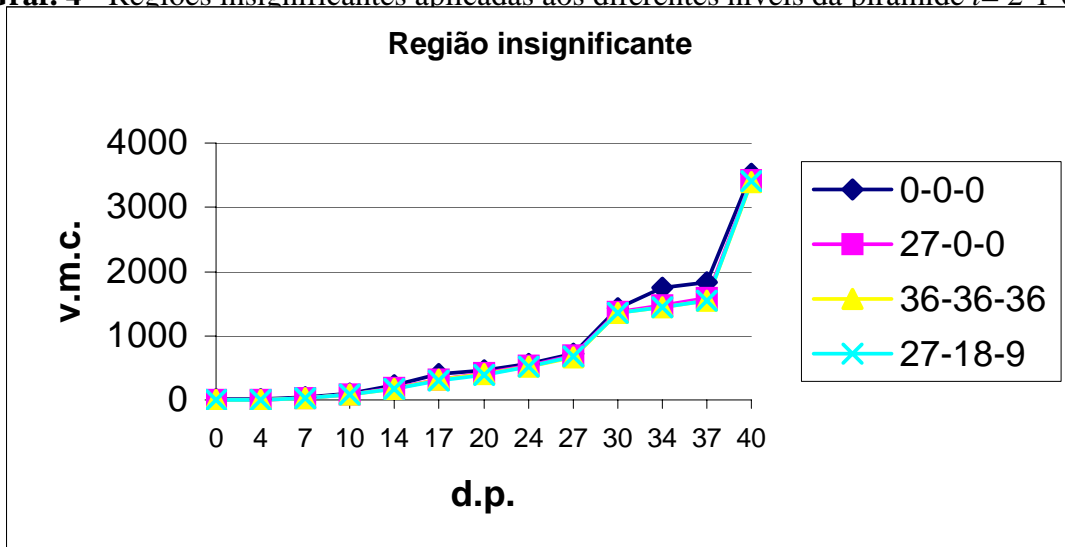
Graf. 1- Alteração na determinação dos voxels pertencentes à fronteira β_n . Cálculo de todos os voxels (8), ou cálculo dinâmico (Auto).

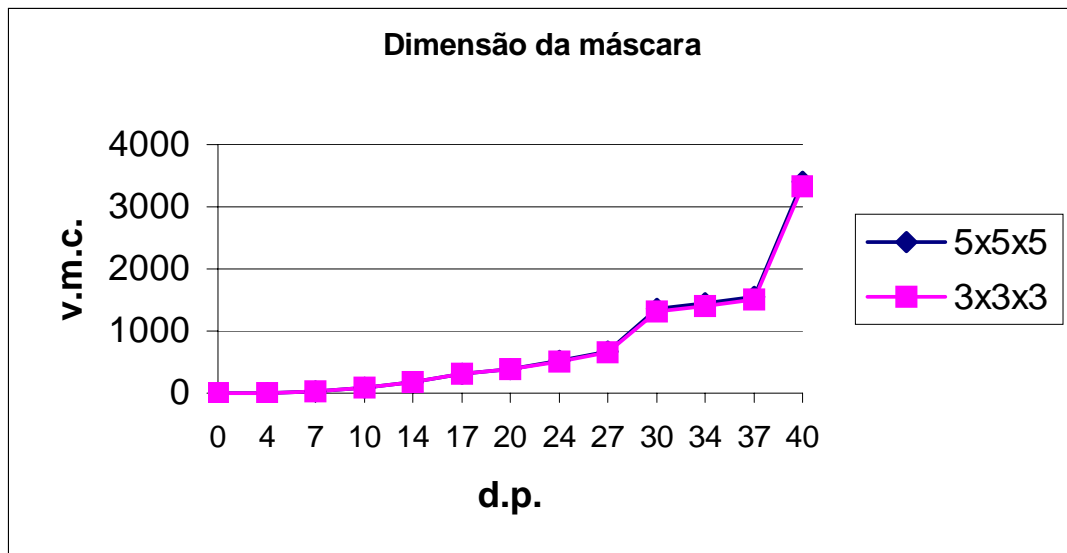


Graf. 2 - Alteração do número de direcções para as quais aplicamos o filtro.

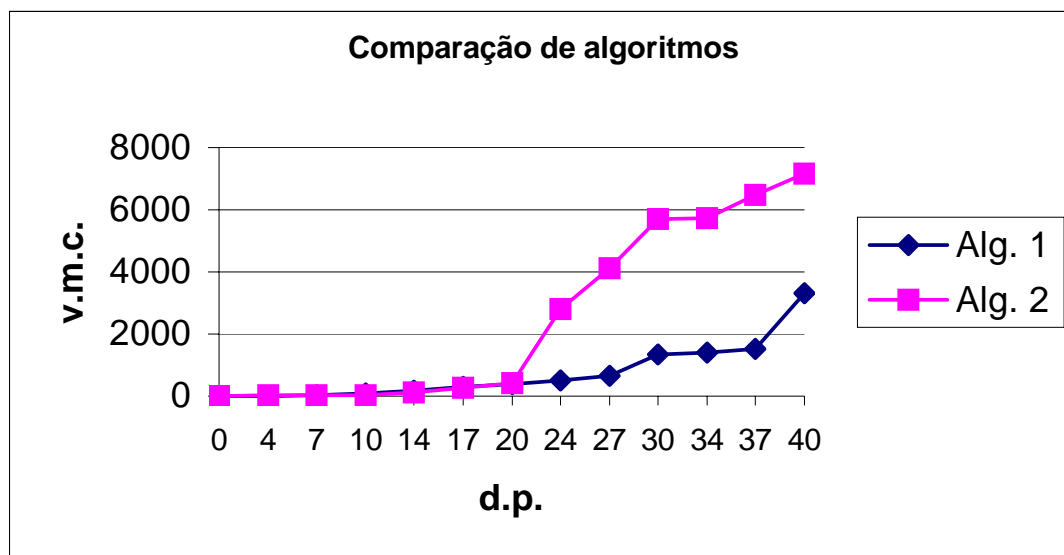
Graf. 3 - Número de classes significativas adjacentes para o cálculo do voxel $\in \beta$.

Graf. 4 - Regiões insignificantes aplicadas aos diferentes níveis da pirâmide $l=2-1-0$.





Graf. 5 - Dimensões da máscara Borboleta 3D.



Graf. 6 - Comparação dos Algoritmos. Algoritmo 1 – apresentado no ponto 4.4.1; Algoritmo 2 – apresentado no ponto 4.3.

Cálculo dos tempos de execução das diferentes opções dos gráficos	Tempo
Segmentação 3D - Condições referidas na introdução do apêndice. (Graf. 6 - Alg. 1)	55s
Calcular todos os voxels (8) da fronteira β_n (Graf. 1)	82s
Considerar todas as direcções (13) do filtro para cálculo do voxel fronteira (Graf. 2)	76s
Alterar a quantidade de classes representativas para 3 (Graf. 3)	104s
Alterar classes insignificantes nos três níveis da pirâmide para 0-0-0 (Graf. 4)	54s
Alterar classes insignificantes nos três níveis da pirâmide para 36-36-36 (Graf. 4)	256s
Utilizando um tamanho de máscara 5x5x5 (Graf. 5)	75s
Tempo de execução de uma única passagem AGWF	141s
Todo o algoritmo referido no ponto 4.3 (Graf. 6 – Alg. 2)	2074s

Tab. 1 - Relação de tempos obtidos para um PC - *Pentium* 166Mhz

APÊNDICE C

RESULTADOS DA SEGMENTAÇÃO SOBRE O VOLUME DE DADOS DO PROJECTO ISACS

Nas páginas seguintes podem ser observados os resultados obtidos após aplicação do modelo de Segmentação em 3D. Foram utilizados volumes de dados obtidos a partir da base de dados ISACS.

A descrição do processo de aquisição e composição dos volumes utilizados para os testes encontra-se descrito resumidamente no capítulo IV na secção 4.4.1-"Resultados obtidos com imagens do projecto ISACS". O volume sobre o qual os testes vão ser executados é uma aproximação do volume de dados final. Esta aproximação é obtida por uma interpolação linear (1D) na direcção do eixo dos x sobre os dados existentes, apenas com o objectivo de provisoriamente preencher as lacunas existentes no volume de dados.

Na primeira linha de imagens encontra-se as secções da imagem original de dimensões 256x256x256 e respectivamente as secção xy_n , yz_n e xz_n da esquerda para a direita, com n a posição da respectiva secção relativamente à origem do eixo de coordenadas. De cima para baixo: imagens das etiquetas e delimitadas pelos contornos para respectivamente $l_{\max} = \{1, 2, 3\}$.

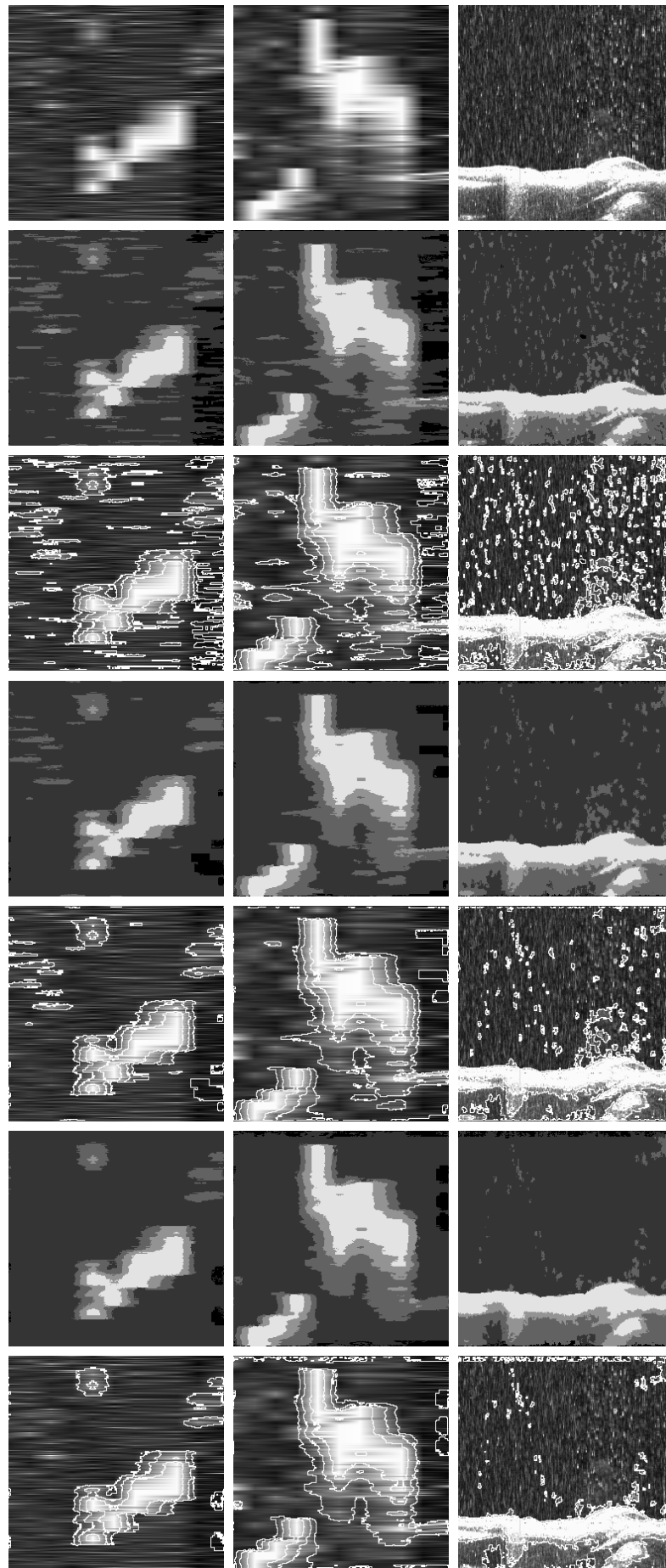


Fig. C.1 – Representação da secção xy_{75} , yz_{75} e xz_{75} da imagem “topas3D_simul”.

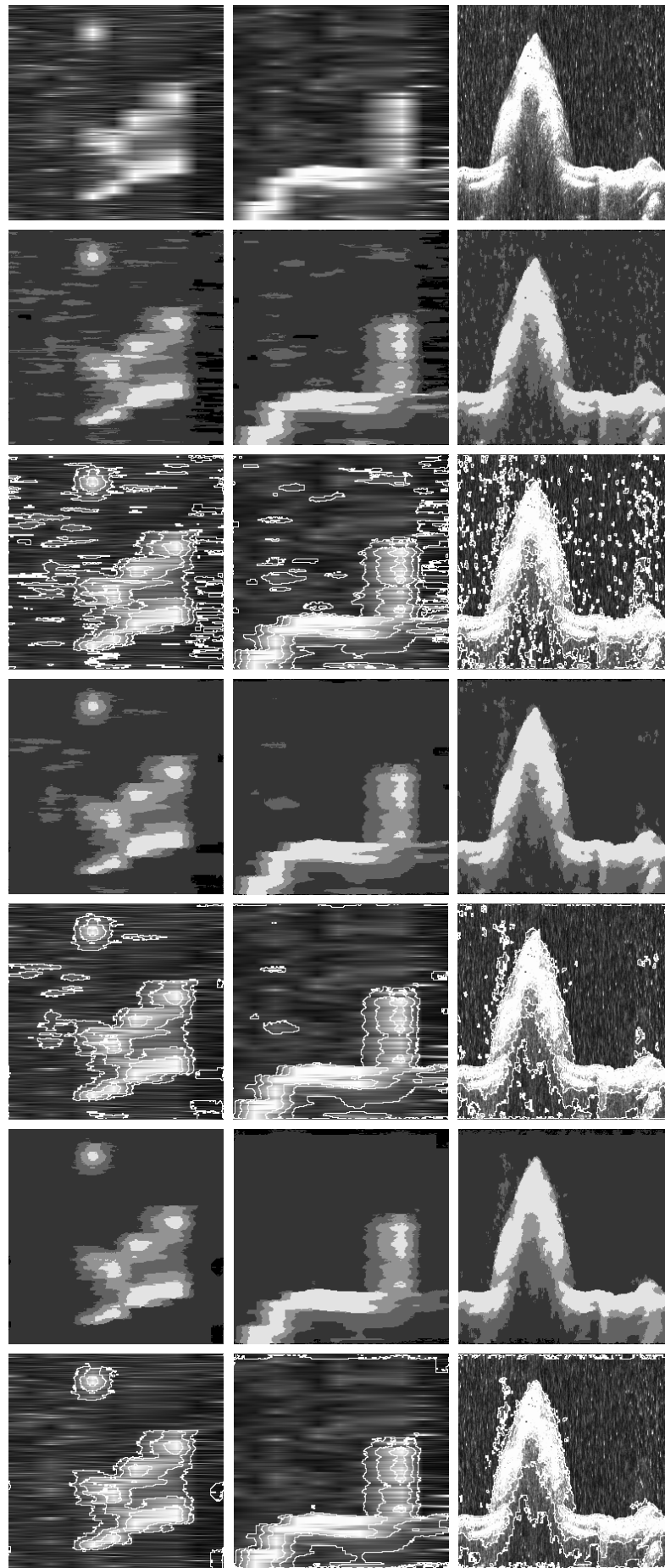


Fig. C.2 – Representação da secção xy_{150} , yz_{150} e xz_{150} da imagem “topas3D_simul”.

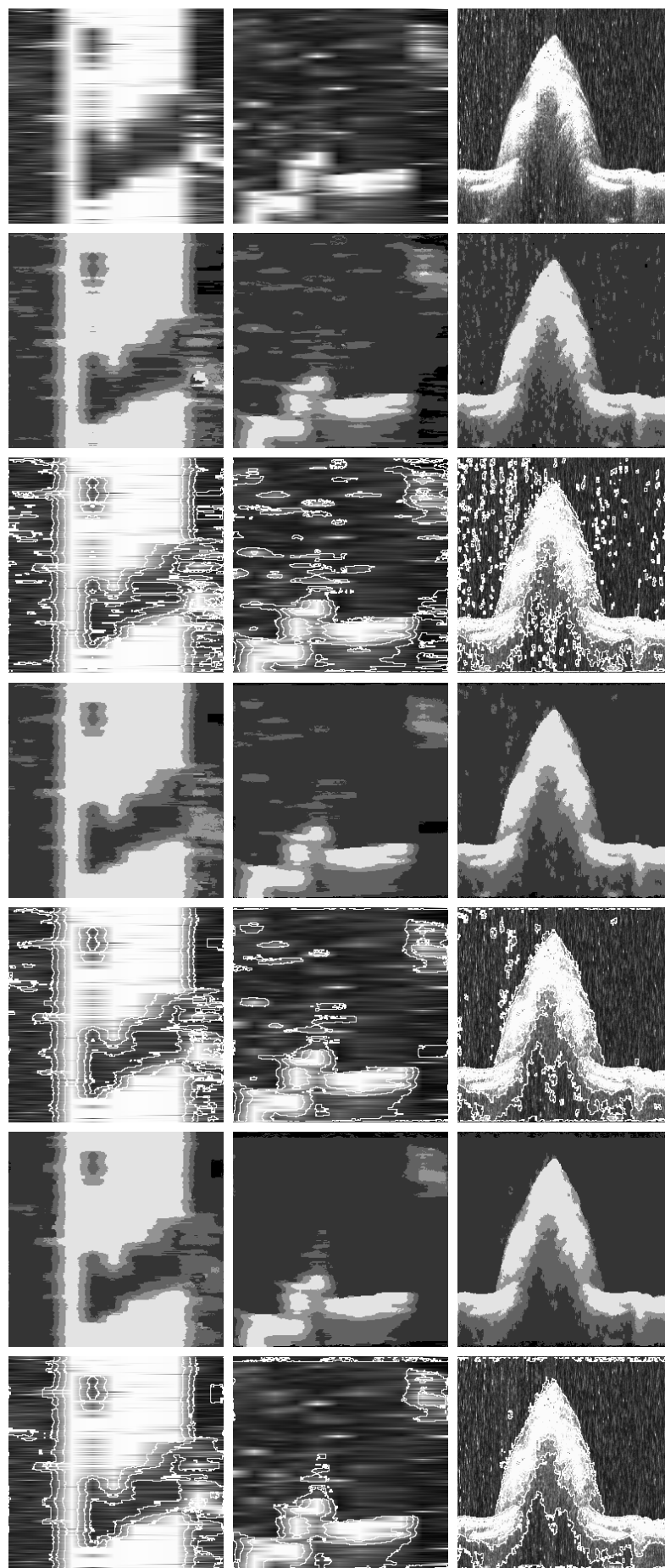


Fig. C.3 – Representação da secção xy_{200} , yz_{200} e xz_{200} da imagem “topas3D_simul”.

**ANEXO
A**

**FILTRO ADAPTATIVO
PONDERADO GAUSSIANO**

Este filtro é caracterizado pela redução do ruído enquanto preserva as fronteiras. Tem como desvantagem o tempo que demora a convergir, sendo esse tempo relativamente proporcional à quantidade de ruído que se pretende reduzir (através de mais ou menos iterações) [Du Buf 90].

A definição do filtro para as três dimensões é a seguinte:

Seja d a diferença entre o voxel central e os outros voxels na vizinhança $3 \times 3 \times 3$

$$d(i, j, k; l, m, n) = |x(i, j, k) - x(i+l, j+m, k+n)|; l, m, n \in \{-1, 0, 1\}. \quad (\text{B.1})$$

Os coeficientes do filtro são baseados no cálculo da função exponencial de acordo com B.2, sendo o peso do voxel central do filtro colocado a zero. A variância é calculada a partir da vizinhança $3 \times 3 \times 3$ do voxel central, sendo este excluído do cálculo.

$$h(i, j, k; l, m, n) = \exp\left(\frac{-d(\cdot)^2}{\sigma^2}\right) \quad (\text{B.2})$$

A filtragem é finalmente obtida por uma simples convolução (eq. B.3):

$$y(i, j, k) = \frac{\sum_{l=-1}^1 \sum_{m=-1}^1 \sum_{n=-1}^1 h(i, j, k; l, m, n) x(i, j, k; l, m, n)}{\sum_{l=-1}^1 \sum_{m=-1}^1 \sum_{n=-1}^1 h(i, j, k; l, m, n)}. \quad (\text{B.3})$$

BIBLIOGRAFIA

- [Berntsen 97] B. Berntsen, "An introduction to ISACS", Proceedings of the 20th Scandinavian Symposium on Physical Acoustics, edited by H.Hobæk, Ustaoset, April 1997.
- [Du Buf 90] J. M. H. Du Buf and T. G. Campbell, "A Quantitative comparison of Edge-Preserving Smoothing Techniques", *Signal Processing* Vol 21, pp. 289-301, 1990.
- [Jain 89] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall International Edition, 1989.
- [Kak 82] A. C. Kak, *Digital Picture Processing*, Vol. 1 e Vol. 2, Academic Press Inc, 1982.
- [Nikolov 97a] S.G. Nikolov, E. Loke, J.M.H. du Buf, "Interactive 3D visualization of sonar data using VRML", Presented at the Int. Workshop on Synthetic-Natural Hybrid Coding and Three Dimensional Imaging, Rhodes, Greece, Sept 1997.
- [Nikolov 97b] S.G. Nikolov and J. Dybedal, *Volumetric Reconstruction, Visualization and Fusion of TOPAS Sub-Bottom Profiler Data*, Technical Report, UALG-ISACS-TR04, 1997.
- [Schroeter 95] P. Schroeter and J. Bigün, "Hierarchical Image Segmentation by Multi-Dimensional Clustering and Orientation-Adaptive Boundary Refinement", *Pattern Recognition* Vol 28, N° 5, pp. 695-709, 1995.
- [Spann 85] M. Spann and R. Wilson, "A Quad-Tree Approach to Image

Segmentation Which Combines Statistical and Spatial Information",
Pattern Recognition Vol 18, N° 3/4, pp. 257-269, 1985.

[Wilson 88] R. Wilson and M. Spann, *Image Segmentation and Uncertainty*,
Research Studies Press Ltd., 1988.