

# SEGMENTAÇÃO DE DADOS 3D

João Rodrigues  
Área Eng. Elect.  
EST/UAlg

Eddy Loke e Hans du Buf  
Vision Laboratory  
UCEH/UAlg

## 1. INTRODUÇÃO

Este artigo apresenta um algoritmo para efectuar uma segmentação não supervisionada em 3D, mostrando a sua aplicação a diferentes volumes, tais como: volumes artificiais, acústica submarina e de ressonância magnética (NMR - *Nuclear Magnetic Resonance*). O trabalho tem por base o efectuado em 2D e já apresentado numa anterior Tecnovisão [1].

## 2. SEGMENTAÇÃO EM 3D

A segmentação consiste em três passos: (1) construção da pirâmide, (2) agrupamento e classificação (3) estimação da fronteira.

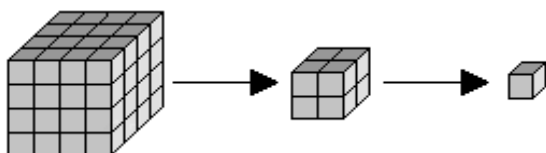
### 2.1. CONSTRUÇÃO DA PIRÂMIDE

A pirâmide em 3D usada é a octógona (fig. 1), a qual é construída a partir da média de blocos de 2x2x2 voxéis não sobrepostos do volume  $I(i,j,k)$ , com  $N_0^3$  o tamanho do volume.

Matematicamente para o volume  $I(i,j,k)$  definido por  $0 \leq i,j,k < N_0$ , a árvore octógona é dada por

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

$0 < l \leq l_{\max}$  e  $I(i,j,k;0) = I(i,j,k)$ . Na prática, não se constrói a pirâmide até ao nível mais alto, mas sobe-se apenas alguns níveis acima da base  $l=0$ . O número de níveis necessário depende da aplicação e só pode ser determinado efectuando diferentes experiências.



**Fig. 1:** Árvore octógona feita pela média de blocos de oito voxéis não sobrepostos.

### 2.2. AGRUPAMENTO E CLASSIFICAÇÃO

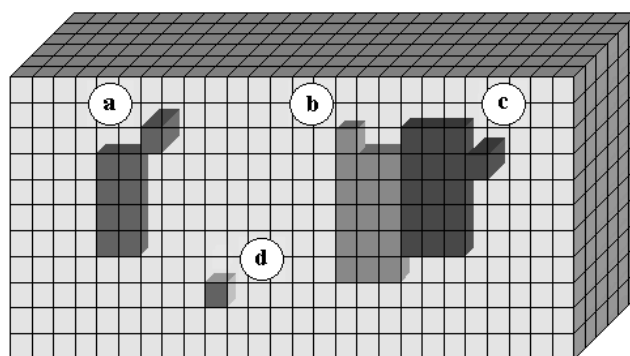
O segundo passo é a aplicação do algoritmo de aglomeração num determinado nível da pirâmide ( $l_{\max}$ ), recorrendo para tal ao algoritmo que faz uso de um espaço de atributos unidimensional - Agrupamento Centríoide Local [4]. Para a criação do volume das

etiquetas  $I_\gamma(i,j,k;l_{\max})$ , será aplicado o critério da distância mínima [1].

#### 2.2.1 RESTAURAÇÃO DA CONECTIVIDADE ESPACIAL

Devido à aplicação de um algoritmo de classificação sem restrições de conectividade espacial, existe a possibilidade de aparecerem voxéis isolados e/ou conjuntos de voxéis que formam regiões insignificantes. Um voxel  $(i,j,k)$  é considerado isolado (fig. 2d) se a sua etiqueta em  $I_\gamma(i,j,k;l_{\max})$  é diferente de todas as etiquetas na sua vizinhança 26 ( $N_{26}$ ). Este é atribuído a classe mais representativa das classes vizinhas determinadas em  $N_{26}$ . Caso existam classes com igual representatividade, o voxel é atribuído à classe com a distância mínima.

Uma região de  $I_\gamma(i,j,k;l_{\max})$  é considerada insignificante (fig. 2a, b, c), quando não contém um número suficiente de voxéis conectados, qualquer que seja a direcção analisada no volume. Os voxéis dentro dessas regiões são atribuídos por um processo determinístico às classes vizinhas e/ou entre elas próprias (fig. 2b, c), como se tratassem de voxéis isolados. Este método pode provocar um ganho de voxéis nas regiões insignificantes adjacentes, que pode tornar-se significativo, mas uma solução alternativa (método multi-passagem) sem este problema provocaria um gasto acentuado de tempo de CPU. Um método de multi-passagem deve ser considerado em aplicações onde não se pode aceitar um *bias* sistemático numa dada direcção, p. ex. de cima para baixo.



**Fig. 2:** Voxel isolado (d), região isolada (a) e duas regiões conectadas (b,c).

### 2.3. ESTIMAÇÃO DA FRONTEIRA

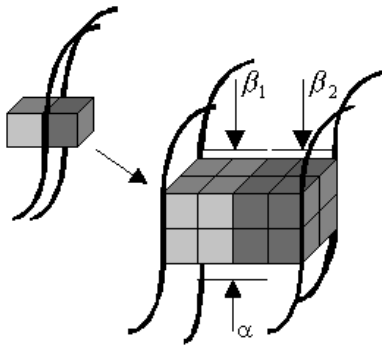
O último passo consiste em transformar o volume a partir do nível  $l_{\max}$  até ao nível 0.

Isto é conseguido pela projecção para baixo das etiquetas dentro das regiões para o nível  $l-1$ , em combinação com a análise dos voxels à volta das fronteiras na árvore original  $I$  no nível  $l$ .

Assumindo que estamos no nível  $l$  e vamos para o nível  $l-1$ , a estimação da fronteira é efectuada da seguinte forma:

- (A) Em  $I\gamma(i,j,k;l)$  os voxels da fronteira são determinados. Um voxel é considerado voxel da fronteira se existe pelo menos uma etiqueta diferente da sua em  $N_{26}$ .
- (B) O volume  $I\gamma(i,j,k;l-1)$  é obtida pela projecção para baixo da etiqueta de cada voxel pai não fronteiro para os seus respectivos oito voxels filhos.
- (C) Os filhos que pertencem à fronteira, definem a região  $\beta$  no nível  $l-1$  e vão ser reclassificados usando filtros borboleta em 3D.

A região fronteira  $\beta$  (fig. 3) está subdividida em duas:  $\beta_1$  que corresponde aos 8 voxels provenientes do voxel esquerdo no nível  $l$  e  $\beta_2$  proveniente do voxel direito. Definimos também uma região central  $\alpha$  que contém apenas aqueles voxels da região  $\beta$  que estabelecem o contacto entre as regiões  $\beta_1$  e  $\beta_2$ .



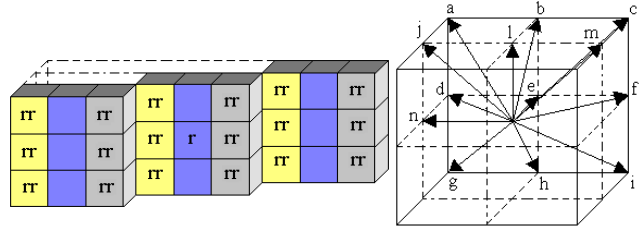
**Fig. 3:** Dois voxels fronteira no nível  $l$  com a definição das regiões fronteira no nível  $l-1$ .

De forma a determinar as classes dos voxels pertencentes à região  $\beta$  foi utilizado um conjunto de filtros, cujo formato e a distribuição dos pesos pela máscara  $3 \times 3 \times 3$  na orientação do eixo  $y$  está apresentado na fig. 4 (esquerda). Estes filtros são uma extensão dos filtros em 2D apresentado em [1] e [3]. Os restantes filtros cada um correspondendo a uma orientação são obtidos pela rotação e redistribuição dos pesos em função da orientação.

Foram considerados 13 orientações de  $a$  até  $n$  mostradas na fig. 4 (direita). Os pesos  $r$  e  $rr$  da máscara do filtro da na fig. 4 (esquerda) são dados por  $rr=(1-r)/n$  e  $r=0$  para  $d \leq 0$ ,  $r=(d-2)/6$  para  $2 < d < 8$  e  $r=1$  para  $d \geq 8$ , com a dissemelhança dada por  $d = |\mu_1 - \mu_2| / \sqrt{\sigma_1^2 + \sigma_2^2}$

com  $n$  o número de pesos diferentes na máscara de 0 e  $r$ , e onde  $\mu_i$  e  $\sigma_i^2$  são a média e a variância determinadas no volume  $I(i,j,k;l)$ , que correspondem às duas classes mais representativas (3 ou mais voxels com igual etiqueta) em  $N_{26}$  do voxel fronteira. Caso existam mais do que duas classes com igual representatividade considera-se as  $N_c$  classes.

No caso de não ter sido encontrada nenhuma classe que



**Fig. 4:** Coeficientes da máscara do filtro (esquerda) e orientação dos filtros de  $a$  até  $n$  (direita).

obedeça ao critério de representatividade ( $N_c=0$ ), o voxel é atribuído directamente ao valor do seu pai. Se existir apenas uma classe que obedeça ao critério ( $N_c=1$ ) o voxel é atribuído directamente a essa classe. Se o número de classes representativas for igual a dois ( $N_c=2$ ) aplica-se a cada voxel  $(i,j,k) \in \beta$  todos os 13 filtros simultaneamente no volume  $I(i,j,k;l)$ . No caso do número de classes representativas for maior que dois ( $N_c > 2$ ) um conjunto de filtros para cada direcção é constituído por um conjunto de máscaras baseadas nas dissemelhanças entre as classes, considerando todas as combinações de classes.

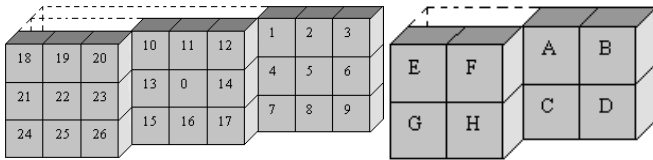
As metades esquerda e direita das máscaras (fig. 4 esquerda) são aplicadas separadamente, eliminando o problema de alisamento ao longo das fronteiras, obtendo-se os dois alvos  $S_1(i,j,k;l-1)$  e  $S_2(i,j,k;l-1)$  para cada filtro e sua máscara correspondente; isto resulta num conjunto de valores  $S_i$ , com  $i = 13N_c(N_c - 1)$ . Calcula-se então a distância Euclideana entre os diferentes  $S_i$  e os valores das médias  $\mu_n$  que correspondem às classes mais representativas, isto é,  $\|\mu_n - S_i\|$ .

Finalmente, cada voxel da fronteira em  $I\gamma(i,j,k;l-1)$  recebe uma etiqueta da classe que apresenta a distância mínima. Diminui-se o valor  $l$  um a um e repete-se novamente o procedimento de determinação da fronteira até alcançar a base da pirâmide.

#### 2.3.1. MELHORIA DO ALGORITMO

Um dos problemas que surge na atribuição dos voxels da fronteira é a grande quantidade de voxels que são necessários classificar (fig. 3). Por exemplo quando surge uma fronteira plana (2 voxels no nível  $l$ ) é

necessário classificar 16 voxels no nível  $l-1$ , 8 da região  $\beta_1$  e 8 da região  $\beta_2$ . Para diminuir o número de voxels a reclassificar foi considerado que só se reclassificaria os voxels da região  $\alpha$  (fig. 3), 4 da região  $\beta_1$  e 4 da região  $\beta_2$ . No caso de uma fronteira mais complexa considera-se todos os voxels de acordo com a posição da fronteira encontrada até ao máximo de 8 para cada região  $\beta_{1,2}$ . A todos os voxels que não são considerados no referido processo e que pertencem a  $\beta_{1,2}$ , será atribuída a etiqueta do seu pai.



**Fig. 5:** Posições dos voxels da fronteira em  $l$  (esquerda) e voxels de  $\alpha$  em  $l-1$  (direita).

Posição da fronteira em $l$	Voxels de $\alpha$ em $l-1$	Posição da fronteira em $l$	Voxels de $\alpha$ em $l-1$
1	A	14	B+D+F+H
2	A+B	15	C+G
3	B	16	C+D+G+H
4	A+C	17	D+F
5	A+B+C+D	18	E
6	B+D	19	E+F
7	C	20	F
8	C+D	21	E+G
9	D	22	E+F+G+H
10	A+E	23	F+H
11	A+B+E+F	24	G
12	B+F	25	G+H
13	A+C+E+F	26	H

**Tab. 1** – Voxel(s) a reclassificar da região  $\alpha$ .

A tabela 1 apresenta os voxels da região  $\alpha$  no nível  $l-1$  como função das posições onde foi determinada a fronteira no nível  $l$ . As respectivas posições são apresentadas na fig. 5. De forma a explicar a tabela 1 e a fig. 5, suponha que no nível  $l$  a fronteira é detectada nos voxels 3 e 12. Isto implica que no conjunto  $\alpha$  no nível  $l-1$  é necessário classificar B e B+F. Neste caso, o número de voxels de  $\beta_1$  diminui de 8 para 2, porque B conta só uma vez. Outra observação efectuada a partir da geometria das máscaras e das orientações dos 13 filtros é que nem todas as direcções definidas afectam todos os voxels fronteiros, porque as orientações que apontam para “longe” do voxel a reclassificar têm pouco ou nenhum efeito no voxel. De forma a diminuir ainda mais o tempo de CPU, considera-se apenas as direcções dos filtros que são adjacentes a esse voxel. Assim, são considerados 7 orientações para os filtros, levando a um  $S_i$ , com  $i = 7 N_c (N_c - 1)$ . Por exemplo, para o voxel A da região  $\alpha$  usa-se as orientações  $a, b, d, e, j, l, n$  como mostrado na fig. 4 (direita). Na tabela 2 estão apresentadas as orientações usadas em função do voxel determinado em  $\alpha$ .

### 3. RESULTADOS EXPERIMENTAIS

O algoritmo foi testado utilizando diferentes conjuntos

Voxel região $\alpha$	Orientações dos filtros						
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. 2** – Orientações dos filtros a aplicar em função do voxel da região  $\alpha$

de dados entre os quais um volume artificial no qual se analisou a influência dos parâmetros do algoritmo em função do ruído.

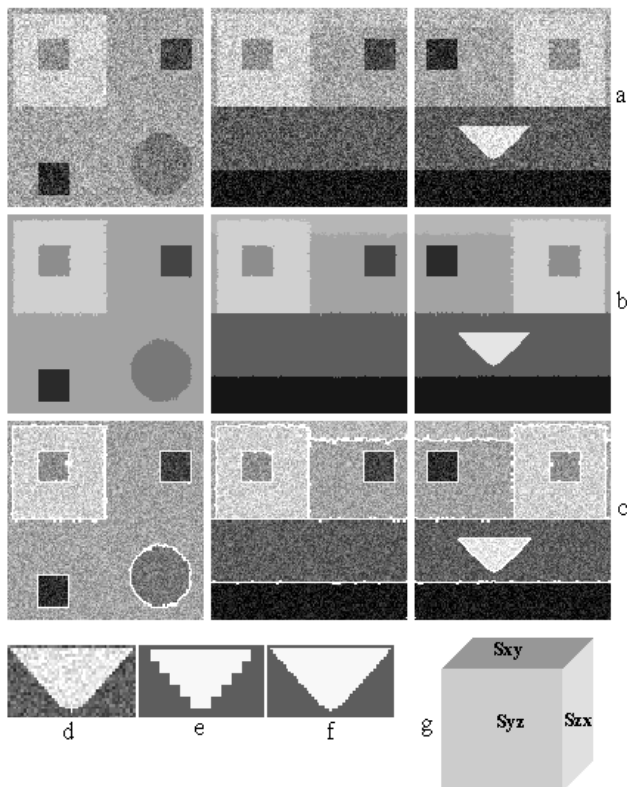
#### 3.1 ANÁLISE DO VOLUME ARTIFICIAL

O volume artificial é constituído por  $128 \times 128 \times 128$  voxels, com amplitudes que variam entre 0 e 255. O volume é composto por 10 regiões distintas constituídas por 4 paralelepípedos que dividem o volume em camadas, 4 cubos inseridos nas diferentes camadas mais uma esfera e um cone. A diferença entre as diferentes regiões é 25. A fig. 6 apresenta 3 secções ortogonais do volume: (a) com  $\sigma_{ruído} = 25$ , (b) segmentação final e (c) fronteiras da segmentação sobre o volume com ruído. A fig. 6 apresenta também uma secção central através do cone com ruído (d), a mais pequena resolução em  $l=2$  (e) e a resolução restaurada depois da segmentação (f). Como se pode ver, a qualidade da fronteira é bastante boa e apenas alguns voxels no topo do cone foram perdidos devido ao ruído.

Sobre este volume foi ainda implementada uma análise sobre os parâmetros ajustáveis baseada no somatório de voxels mal classificados (v.m.c.) resultantes da subtracção do volume original sem ruído pelo volume das etiquetas. Este ultimo foi obtido após a aplicação do algoritmo de segmentação ao volume original onde foi adicionado ruído Gaussiano com  $0 < \sigma_{ruído} < 40$ .

Os parâmetros ajustáveis analisados foram: O número de voxels a considerar na região  $\beta$ , o número de orientações do filtro para cada voxel fronteiro, possibilidade de considerar mais de 2 classes representativas na vizinhança da fronteira, alterar a quantidade de voxels nas regiões consideradas insignificativas e alterar o tamanho da máscara.

Os melhores resultados (menor número de v.m.c.) obteve-se quando consideramos apenas as 2 classes mais representativas na vizinhança da fronteira; utilizamos máscara  $3 \times 3 \times 3$ ; número de direcções da máscara de 7 (tabela 2); regiões insignificante de 27 voxels retiradas apenas no nível mais alto da pirâmide ( $l_{max}=2$ ) e voxels pertencentes à região fronteira serão determinados usando a tabela 1.



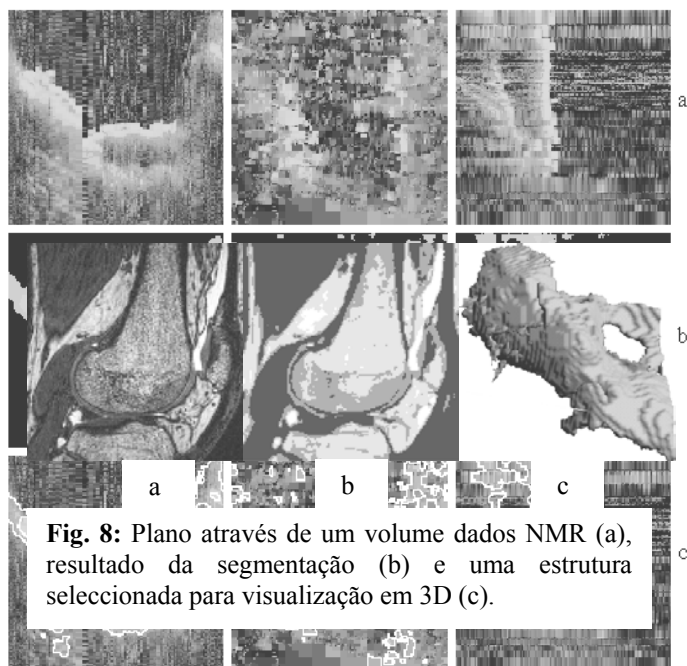
**Fig. 6:** Secções planas através do volume artificial: (a) dados com ruído, (b) depois da segmentação, (c) fronteiras da segmentação. As colunas da esquerda, centro e direita correspondem às secções Sxy, Syz e Szx respectivamente. Secções d, e, f apresentam uma secção central através do cone artificial.

### 3.2 APLICAÇÃO AO VOLUME DE ACÚSTICA SUBMARINA

Os dados acústicos aqui utilizados são capturados por um navio, com um sonar de penetração de fundo, onde são emitidos *pings* verticais e os seus ecos são guardados no navio enquanto este manobra. Esta manobra é bastante difícil, provocando a aquisição dos dados em linhas irregulares e consequentemente o aparecimento de lacunas.

O procedimento normal para retirar as lacunas é: (1) criar uma grelha em 2D dos *pings* usando coordenadas GPS; (2) interpolar em 3D (em longitude, latitude e profundidade). Este procedimento está descrito em [2]. Só depois da interpolação volumétrica a segmentação pode ser aplicada, encontrando-se os resultados da mesma na fig. 7. A linha (a) apresenta uma secção do volume original depois da aplicação da interpolação, na linha (b) as etiquetas depois da aplicação da segmentação e na linha (c) a secção do volume original com os contornos delimitados. Os dados na fig. 7 apresentam muito ruído e irregularidades, isto é devido a uma equalização do histograma que foi utilizada para efeitos de impressão. Os dados originais são um pouco irregulares devido à interpolação, mas apresentam muito menos ruído. A partir da fig. 7, pode-se concluir que os resultados da qualidade da segmentação no caso de dados reais são bastante bons.

### 3.3 APLICAÇÃO AO VOLUME DE DADOS NMR



**Fig. 8:** Plano através de um volume dados NMR (a), resultado da segmentação (b) e uma estrutura seleccionada para visualização em 3D (c).

**Fig. 7:** Secção planar sobre o volume de acústica. (a) dados originais, (b) depois da segmentação e (c) fronteiras no volume original. A coluna da esquerda, centro e direita correspondem às secções Sxy, Syz e Szx respectivamente.

Os resultados obtidos na aplicação de um volume NMR [5] estão apresentados na fig. 8. Onde a figura 8a mostra um plano do volume original que corresponde a um joelho e na figura 8b a segmentação resultante. A zona a branco à direita é chamada *suprapatellar bursa*, tendo sido este o objecto escolhido para visualizar-se em 3D com *OpenGL* usando *Gouraud shading* (fig. 8c).

### 4. CONCLUSÕES

Apresentou-se um algoritmo rápido para a segmentação não supervisionada que evita conhecimentos *à priori* das classes. No entanto, não é completamente não supervisionada devido à necessidade de seleccionar o nível  $l_{\max}$  da árvore octógona no qual o agrupamento é feito. Este nível depende das características do volume de dados disponíveis, e só pode ser determinado efectuando diferentes experiências. Com base nos dados considerados, pode-se concluir que quando aumentamos  $l_{\max}$  as fronteiras finais pioram, apesar de aparecerem menos regiões insignificantes e de ser mais fácil determinar as classes.

A estratégia de eliminar regiões insignificantes (27 voxels) apenas no nível da pirâmide  $l_{\max}$  mostrou-se óptima, devido ao desaparecimento dos voxels mal classificados nesse nível, bem como à propagação destes através da pirâmide. Os resultados do algoritmo 3D são bastante bons, uma vez que é possível identificar as diferentes regiões e suas fronteiras tal como um observador humano as delimitaria. Esta

afirmação não é só feita pela análise das secções (imagens) apresentadas, mas por uma análise conjunta de todas as secções adjacentes em qualquer das direcções. Finalmente mostrou-se uma extensão de Schroeter e Bigün [3] em 2D para 3D com pouco gasto de tempo de CPU até para grandes volumes de dados.

## 5. REFERÊNCIAS

[1] Rodrigues, J. e du Buf J.M.H. (1999) Segmentação de Imagens Acústicas, *Tecnovisão*, Nº 9, pp. 19-22.

[2] Loke, R. E. and du Buf, J.M.H. (1998) Volumetric processing of Topas underwater acoustic data, *Proc. IEEE/OES Conf. OCEANS'98*, pp. 742-746.

[3] Schroeter, P. and Bigün, J. (1995) Hierarchical image segmentation by multi-dimensional clustering and orientation-adaptive boundary refinement, *Pattern Recognition*, Vol 28, Nº 5, pp. 695-709.

[4] Spann, M. and Wilson, R. (1985) A quad-tree approach to image segmentation which combines statistical and spatial information, *Pattern Recognition*, Vol 18, Nº 3/4, pp. 257-269.

[5] Rodrigues, J., Loke, R.E. and du Buf, J.M.H. (2000) Fast Segmentation of 3D data using an octree, *RECPAD2000*, Porto, Portugal, May 11-12, pp 185-189.