

# Segmentação de Imagens

## ✓ Motivação

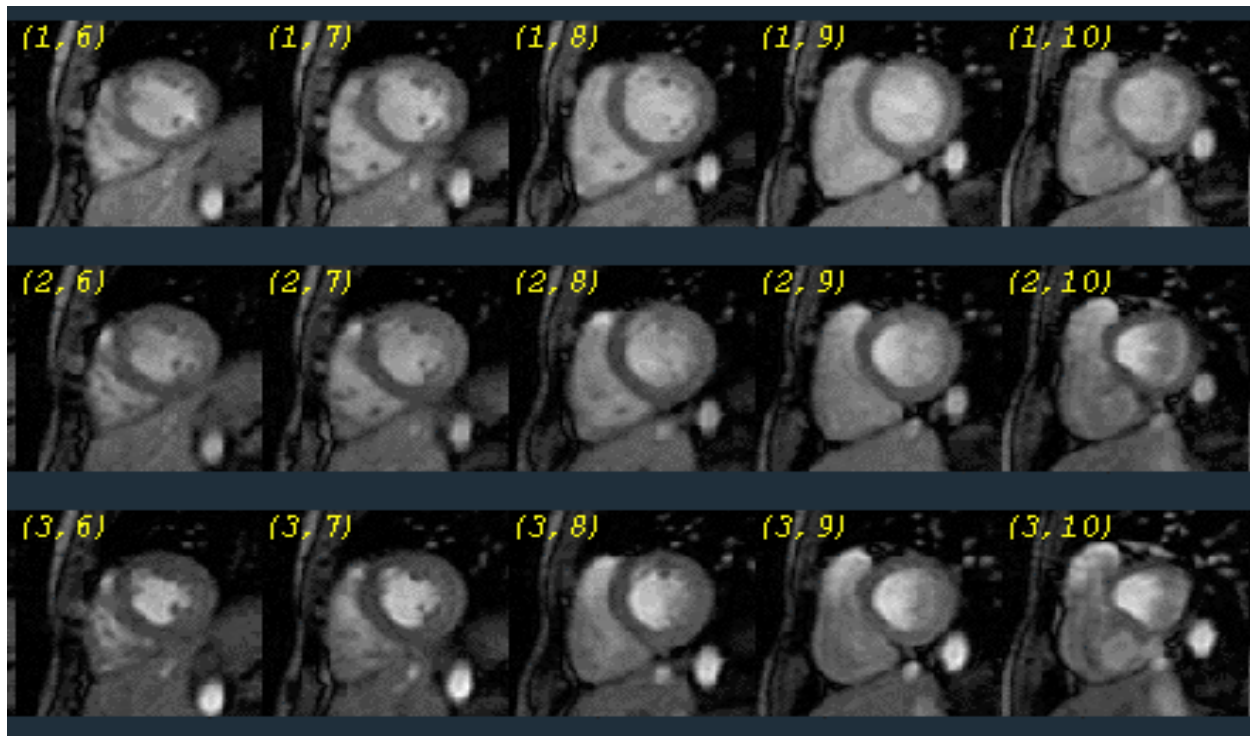
- Identificação de objetos
- Quantização: contagens, área, perímetro, volume
- Visualização 2D, 3D
- Reconhecimento de padrões
- Classificação



- ◆ Normal
- ◆ Patolog.
  - ◆ Congen.
  - ◆ Adquir.
  - ◆ ...

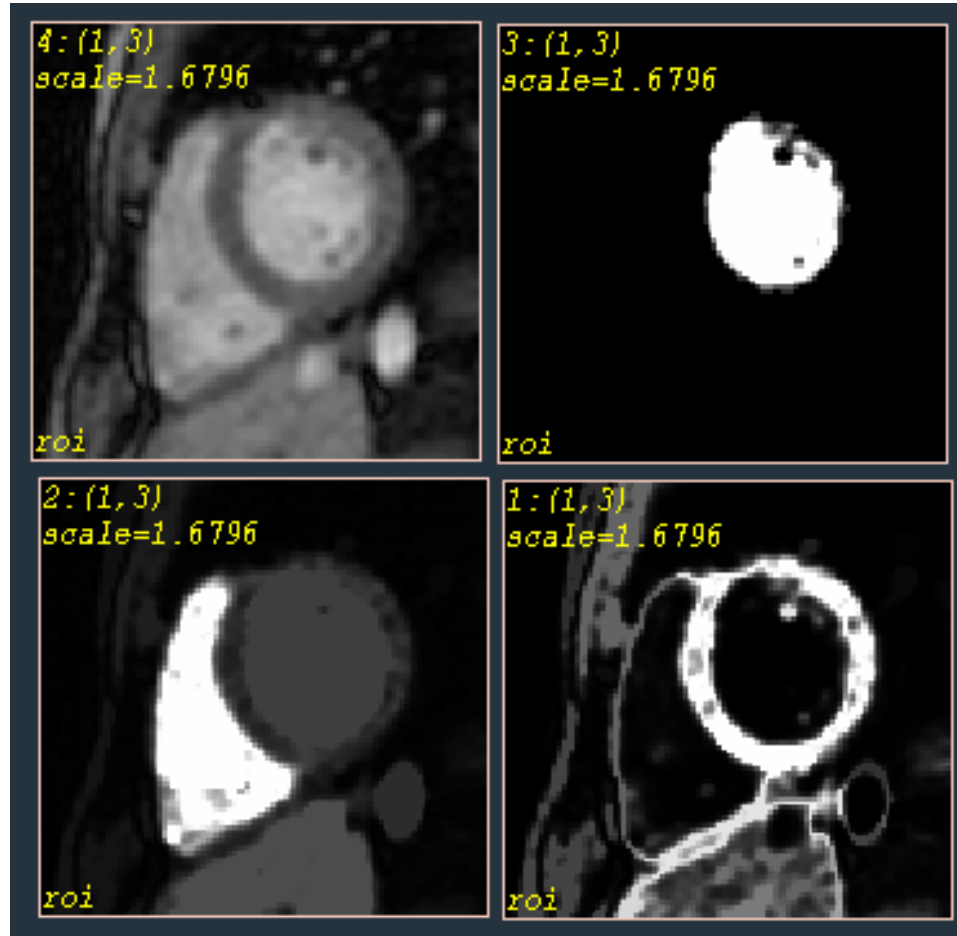
# Motivação

- ✓ Processos convencionais (manual e semi-automático) : demorados e cansativos
  - gated MRI : 16 volumes, 12 cortes => 192 img
  - gated SPECT: idem



# Resultados em RM

Transv.  
RM

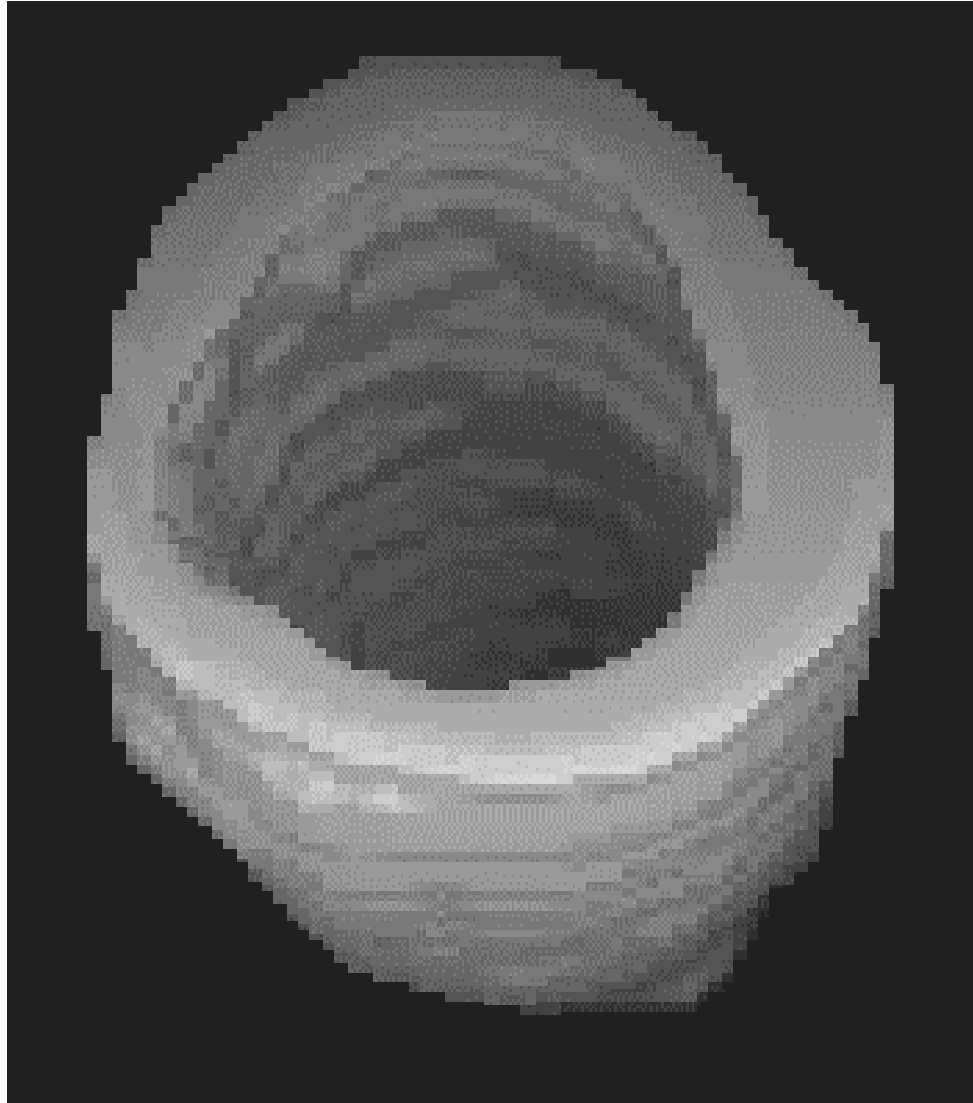


VE

VD

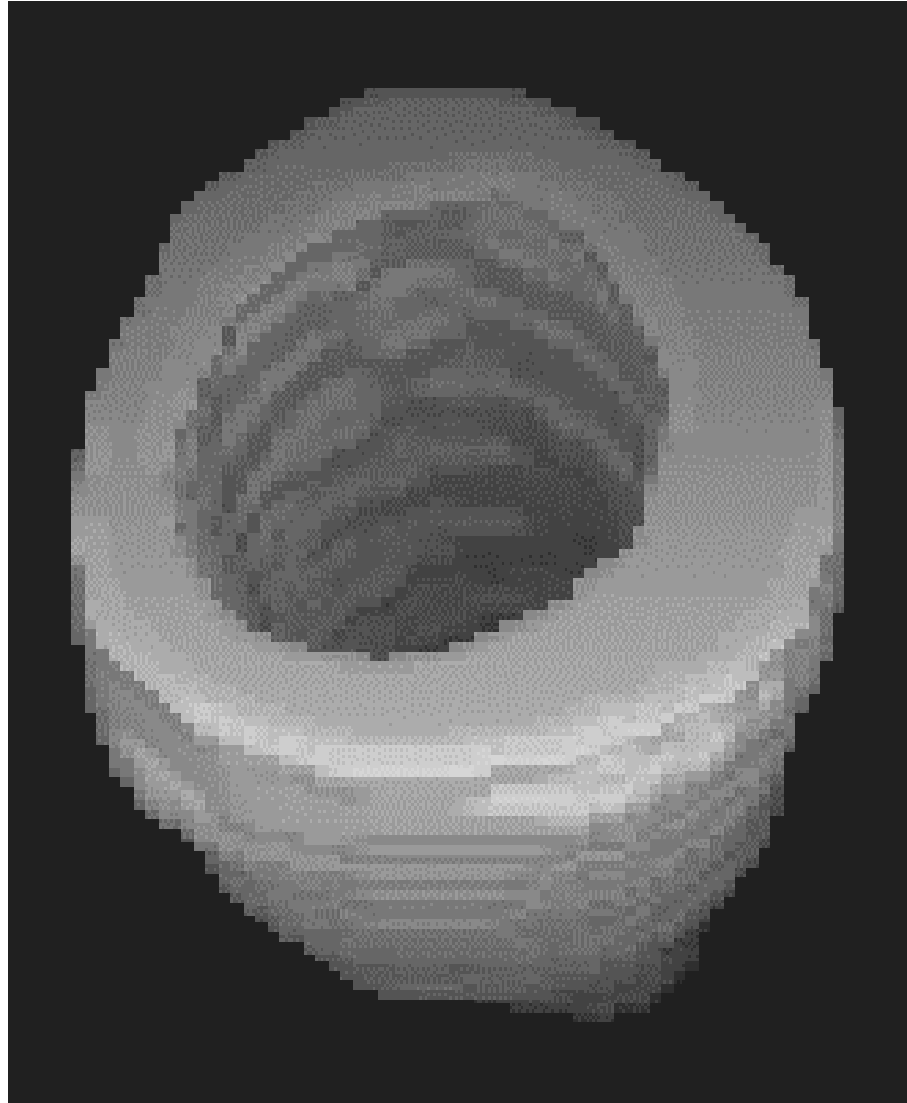
Mioc.

# VE: pre-Ventriculotomia



Diástole

# VE: pre-ventriculotomia



Sístole

# Visão Global

---

- ✓ Operadores
- ✓ Representação dos resultados da segmentação
- ✓ Técnicas de segmentação
  - thresholding
  - snakes
  - region-growing
  - split-merge
  - fuzzy connectedness
  - redes neurais
  - métodos estatísticos
- ✓ Interpretação

# Segmentação de Imagens

## ✓ Por descontinuidade

- operadores (detetores)

- ponto

- linha

- borda (gradiente, laplaciano e LoG)

- contornos

- manual

- semi-automático

- automático (conexão de bordas, Transf. de Hough)

## ✓ Por similariedade

- thresholding

- region-growing

- split and merge

# Gradiente

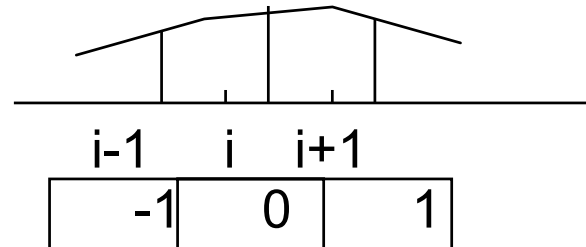
$$\nabla f(x, y) = \frac{\partial f(x, y)}{\partial x} u_x + \frac{\partial f(x, y)}{\partial y} u_y$$

$$\left. \frac{\partial f(x, y)}{\partial x} \right|_{x=i} = f\left(i + \frac{1}{2}\right) - f\left(i - \frac{1}{2}\right)$$

$$f\left(i + \frac{1}{2}\right) = \frac{f(i) + f(i+1)}{2}$$

$$f\left(i - \frac{1}{2}\right) = \frac{f(i-1) + f(i)}{2}$$

$$\left. \frac{\partial f(x, y)}{\partial x} \right|_{x=i} = \frac{f(i+1) - f(i-1)}{2}$$



$$\frac{\partial f(x, y)}{\partial x}$$

-1	0	1
-1	0	1
-1	0	1

Sobel

-1	0	1
-2	0	2
-1	0	1

$$\frac{\partial f(x, y)}{\partial y}$$

-1	-1	-1
0	0	0
1	1	1

-1	-2	-1
0	0	0
1	2	1



## Algoritmo p/ Laplaciano em x?

$$\left. \frac{\partial f(x, y)}{\partial x} \right|_{x=i} = f\left(i + \frac{1}{2}\right) - f\left(i - \frac{1}{2}\right)$$

$$f\left(i + \frac{1}{2}\right) = \frac{f(i) + f(i+1)}{2}$$

$$f\left(i - \frac{1}{2}\right) = \frac{f(i-1) + f(i)}{2}$$

$$\left. \frac{\partial f(x, y)}{\partial x} \right|_{x=i} = \frac{f(i+1) - f(i-1)}{2}$$

$$\nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2}$$

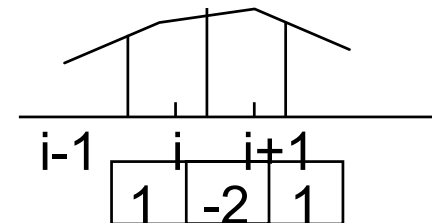
# Sobel, Laplace,...

$$\text{Sobel } (f(x, y)) = \sqrt{\left(\frac{\partial f(x, y)}{\partial x}\right)^2 + \left(\frac{\partial f(x, y)}{\partial y}\right)^2}$$

$$\text{Laplace } \nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2}$$

$$\begin{aligned} \left. \frac{\partial^2 f(x, y)}{\partial x^2} \right|_i &= \frac{\partial \left( \frac{\partial f}{\partial x} \right)}{\partial x} = \left( \frac{\partial f}{\partial x} \right)_{i+\frac{1}{2}} - \left( \frac{\partial f}{\partial x} \right)_{i-\frac{1}{2}} \\ &= (f(i+1) - f(i)) - (f(i) - f(i-1)) \\ &= f(i-1) - 2f(i) + f(i+1) \end{aligned}$$

0	1	0
1	-4	1
0	1	1



$$H(z) = (z^{-1} - 2 + z)$$

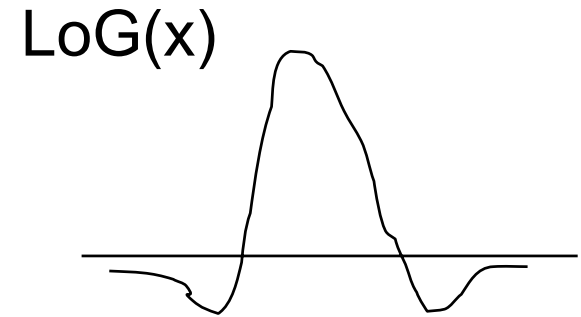
$$H(w) = -2(1 - \cos(wT))$$

# Laplaciano da Gaussiana (LoG)

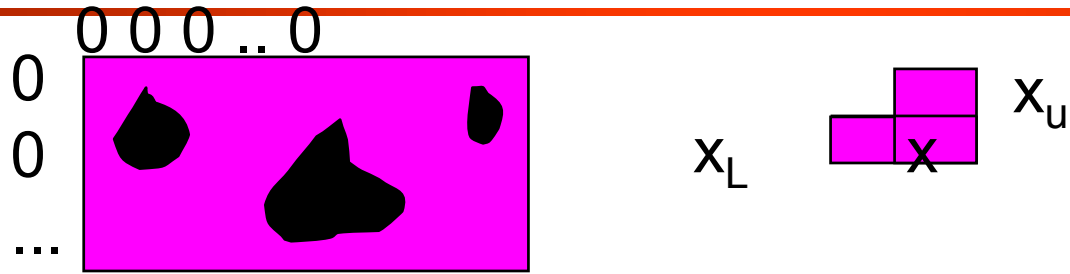
- ✓ Edge detector
  - Gauss=>Smooth
  - Laplace=>Zero crossing

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

$$g(x, y) = \nabla^2 Gauss(x, y) * f(x, y)$$



# Label (blob)



0  $k=1$  (classe do background/borda 0)

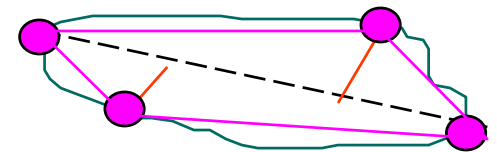
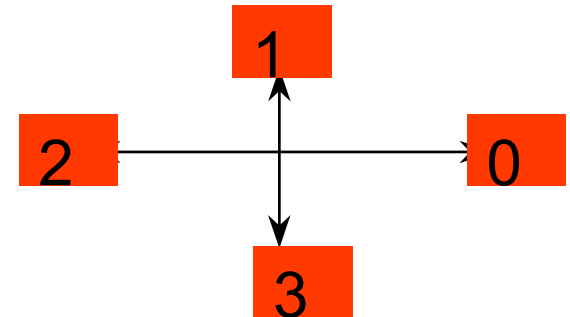
scan image  $x$

- if  $f(x) == 0$  continue
- if  $f(x_u) == 1$  and  $f(x_L) == 0 \Rightarrow c(x) = c(x_u)$
- if  $f(x_u) == 0$  and  $f(x_L) == 1 \Rightarrow c(x) = c(x_L)$
- if  $f(x_u) == 1$  and  $f(x_L) == 1 \Rightarrow$   
 $c(x) = c(x_L),$   
 $c(x_u) \text{ equiv. } c(x_L)$
- if  $f(x_u) == 0$  and  $f(x_L) == 0 \Rightarrow c(x) = k++$

Expansão p/ 3D

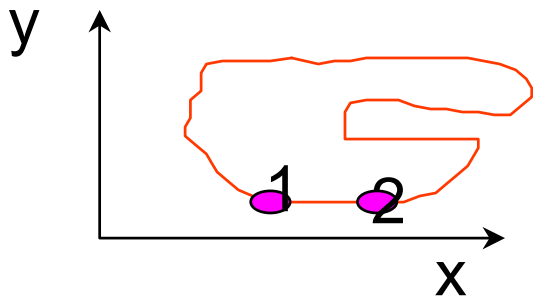
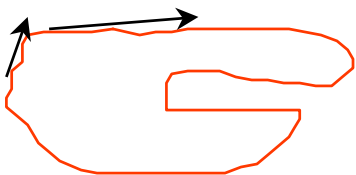
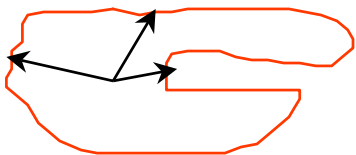
# Representação dos objetos

- ✓ formato, texturas
- ✓ características desejadas:
  - invariância à Translação, Rotação e Escala (e Ponto de início p/ contornos)
- ✓ Contornos de objetos
  - encadeamento: 013102...
    - direção
    - variação de direção
    - Invar. T, R, (E se usar escala adequada), (I se normalizar)
  - polígonos
    - max. distância ao contorno
  - convex hull
    - menor conjunto convexo que contém o objeto.

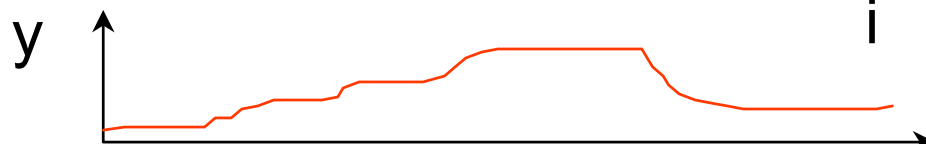
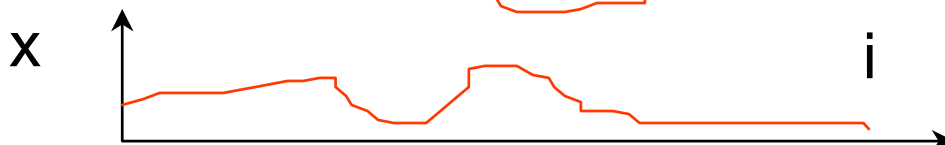
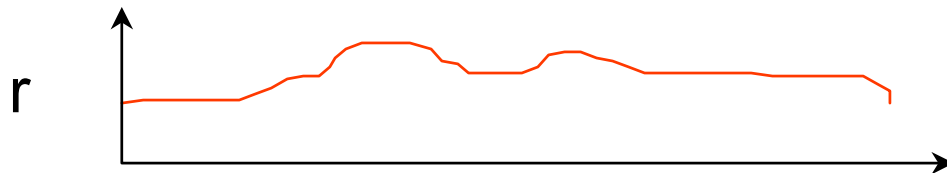


# Representação de contornos

- esqueleto
- assinaturas



$r(\theta)$ : inv. T, R (se mesmo I),  
E (se normalizado pela var.)  
 $\theta(i)$ :  
 $x(i) + jy(i)$



# Segmentação por região

## ✓ Região: Region-growing

- conectividade, afinidade, tamanho, forma, possibilidade
  - semente
  - para cada vizinho, agregar o mesmo se similar. Se agregado, considerar os vizinhos deste.

## ✓ Região: Split and Merge

- quadtree, octree
  - testar homogeneidade de cada quadrante
    - se não homogêneo, subdividir e continuar até último quadrante
  - merge de quadrantes vizinhos com homogeneidades similares.

---

✓ Movimento:

- diferença acumulativa: histograma
- domínio da freq.: projeção das imagens. velocidade média



# Técnicas: Thresholding

- ✓  $T = T[ f(x,y), x, y, p(x,y) ]$ 
  - de imagem  $f(x,y)$
  - $p(x,y)$ : propriedade local
- ✓ Global
- ✓ Ótimo:
- ✓ local : baseado na região das bordas
- ✓ baseado em características: grad. e laplaciano
- ✓ multi-banda

$$\min_T \text{Erro}(T)$$

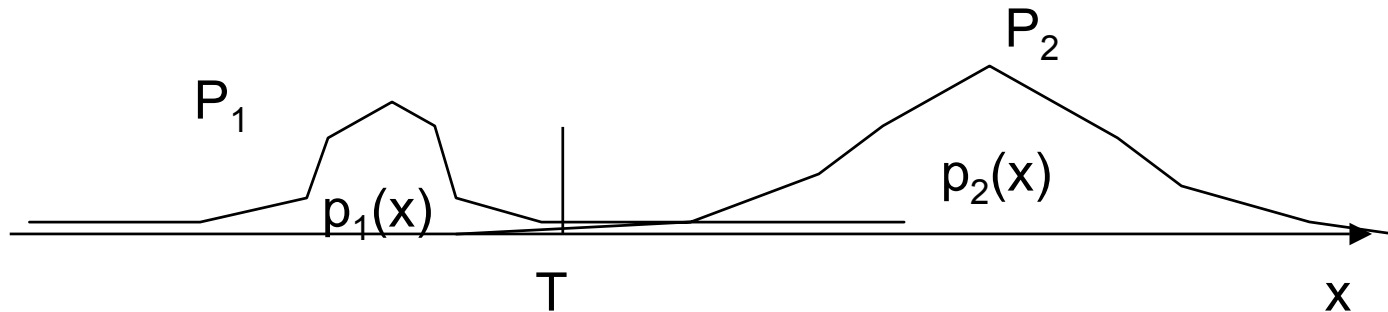
$$\text{Erro}(T) = P_2 \cdot \int_{-\infty}^T p_2(x) dx + P_1 \cdot \int_T^{\infty} p_1(x) dx$$

$$\frac{\partial \text{Erro}(T)}{\partial T} = 0 \Rightarrow P_1 \cdot p_1(T) = P_2 \cdot p_2(T)$$

$p_1(x), p_2(x)$ : se Gaussian

$$T = \frac{\mu_1 + \mu_2}{2} + \frac{\sigma^2}{\mu_1 - \mu_2} \ln\left(\frac{P_2}{P_1}\right)$$

# Optimal thresholding



$$p(x) = P_1 \cdot p_1(x) + P_2 \cdot p_2(x)$$

$$p_i(x) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{(x - \mu_i)^2}{2\sigma_i^2}\right)$$

$$P_1 + P_2 = 1$$

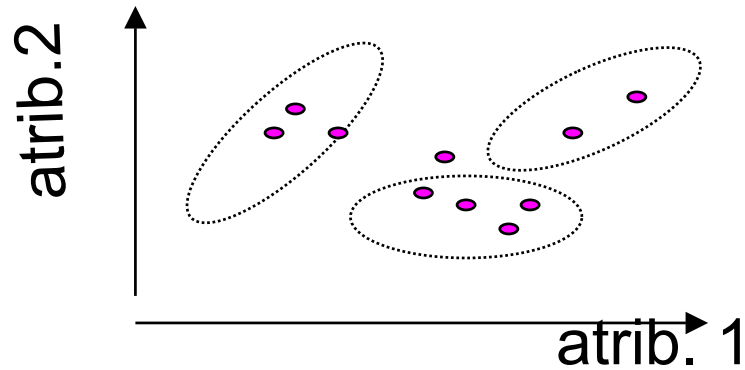
$$\text{Erro} = P_2 \cdot \int_{-\infty}^T p_2(x) \cdot dx + P_1 \cdot \int_T^{\infty} p_1(x) \cdot dx$$

$$\min_T \text{Erro} \Rightarrow \frac{\partial \text{Erro}}{\partial T} = 0$$

$$\Rightarrow P_1 \cdot p_1(T) = P_2 \cdot p_2(T)$$

$$T = \frac{\mu_1 + \mu_2}{2} + \frac{\sigma^2}{\mu_1 - \mu_2} \ln\left(\frac{P_2}{P_1}\right)$$

# Isodata (k-means)



- 0) K classes com centro em  $c_i$
- 1) Inicializar  $c_j^{(0)}$
- 2) Para cada  $x_i \Rightarrow$  atribuir  $x_i$  p / classe  $j$  com menor distancia
- 3) Recalcular  $c_j$
- 4) Repetir 2) e 3) ate nao haver mais alter.

# Segmentação de estruturas 3D dinâmicas

## ✓ Motivação

- medida e visualização de estruturas do mundo real
- estruturas dinâmicas
- coração, VE, miocárdio, ...
- parâmetros: FE, volume, área, ...

## ✓ Dificuldades

- elevado volume de dados:  
16 vol. X 12 slices X 256 X 256  
=> 192 slices
- estruturas complexas : forma, superposição, diversas intersecções com um plano, não-homogeneidade, especificidade, ...

# PROBLEMA e PROPOSTA

## ✓ PROBLEMA

- Segmentação de estruturas 3D, 4D
- Estruturas complexas

## ✓ PROPOSTA

- conectividade fuzzy considerando múltiplas imagens de um objeto

## ✓ RAZÕES INTUITIVAS

- estruturas de interesse tem voxels conectados (fuzzy) pelo menos em alguma dimensão
- conectividade baseado em afinidade entre voxels
- fácil expansão p/ 3D, 4D

# Fuzzy Connectedness entre 2 vox.

- ✓ Considera adjacencia e afinidade entre dois voxels
- ✓ Dados 2 voxels “i” e “j” com valores (vetorial)  $v(i)$ ,  $v(j)$

$$fc(i,j)=\max\{f(i,j)\}$$

$$f(i,j)=g(\text{adj}(i,j), v(i), v(j), i, j)$$

adj : adjacencia (fuzzy)

$$f(i,j)=\text{adj}(i,j) \cdot \text{affin}(v(i), v(j), i, j)$$

affin: afinidade (fuzzy)

# Fuzzyness

- ✓ fuzzy sets, membership

Domínio  $X$

Fuzzy Set =  $\{ (x, f(x)) \mid x \in X \}$

$f : X \Rightarrow [0,1]$  (membership)

- ✓ fuzzy relation

$i \in X, j \in X$

relation =  $\{(i,j), r(i,j)\}$

$r : X \times X \Rightarrow [0,1]$

## ✓ fuzzy relation

- connectedness
- adjacency
- affinity

## ✓ Proposta

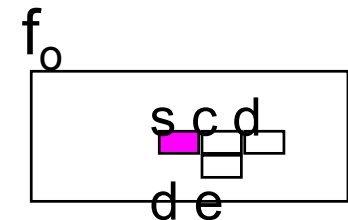
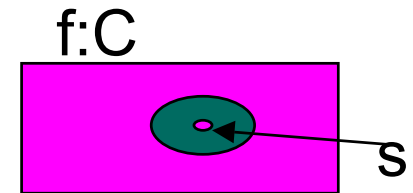
- $\text{adj}(i,j)=1$  (se 1-adjacent)  
=0
- $\text{affin}(i,j)=\exp(-0.5d^{**2})$   
 $d^{**2}$  : dist. estatística
- programação dinâmica  
queue



# Conectividade Fuzzy

## ✓ Algorit: Fuzzy Object Extraction

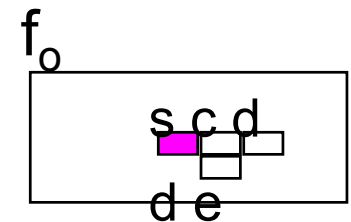
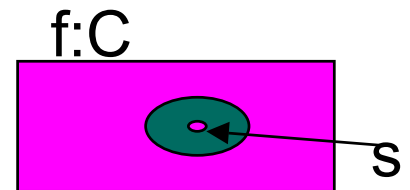
- Input: imagem  $(C, \mathbf{f})$  e seed  $\mathbf{s}$
- Output: imagem  $(C_o, \mathbf{f}_o)$
- Auxiliares: queue  $Q$ 
  0. Set  $f_o(i)=0$ ,  $i \neq s$  e  $f_o(s)=1$
  1. Push all spels  $\mathbf{c}$  of  $C_o$  |  $\text{affin}(s, \mathbf{c}) > 0$  to  $Q$
  2. While  $Q$  is not empty do
    - 2.1. Remove a spel  $\mathbf{c}$  from  $Q$
    - 2.2. find  $f_{\max} = \max_d [\min(\mathbf{f}_o(d), \text{affin}(c, d))]$
    - 2.3. if  $f_{\max} > \mathbf{f}_o(c)$  then
      - set  $\mathbf{f}_o(c) = f_{\max}$
      - push all spels  $\mathbf{e}$  |  $\text{affin}(c, \mathbf{e}) > 0$  to  $Q$
      - endif
    - 2.4. endwhile



# Conectividade Fuzzy

## ✓ Algoritmo: Fuzzy Object Extraction

- Input: imagem  $(C, f)$  e seed  $s$
- Output: imagem  $(C_o, f_o)$
- Auxiliares: queue  $Q$ 
  0. Set  $f_o(i)=0$ ,  $i \neq s$  e  $f_o(s)=1$
  1. Push all spels  $c$  of  $C_o$  |  $\text{affin}(s,c) > 0$  to  $Q$
  2. While  $Q$  is not empty do
    - 2.1. Remove a spel  $c$  from  $Q$
    - 2.2. find  $f_{\max} = \max_d [\min(f_o(d), \text{affin}(c,d))]$
    - 2.3. if  $f_{\max} > f_o(c)$  then
      - set  $f_o(c) = f_{\max}$
      - push all spels  $e$  |  $\text{affin}(c,e) > 0$  to  $Q$
    - 2.4. endwhile



---

# Segmentação de Imagens Médicas 3D baseado em Vetor de Atributos e Conectividade Competitiva

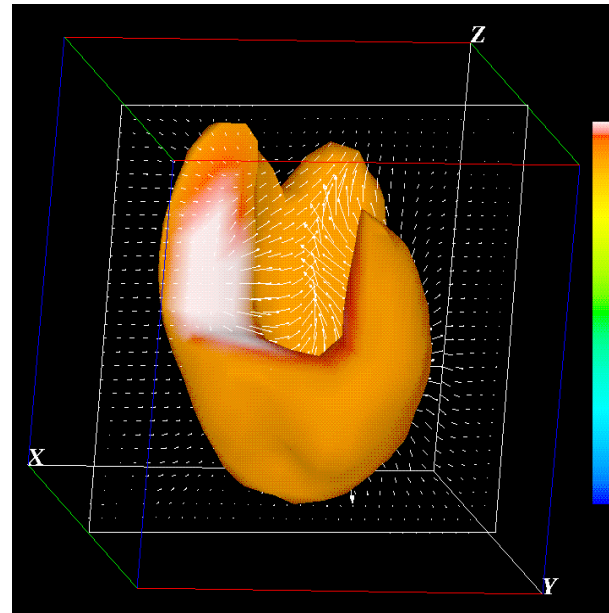
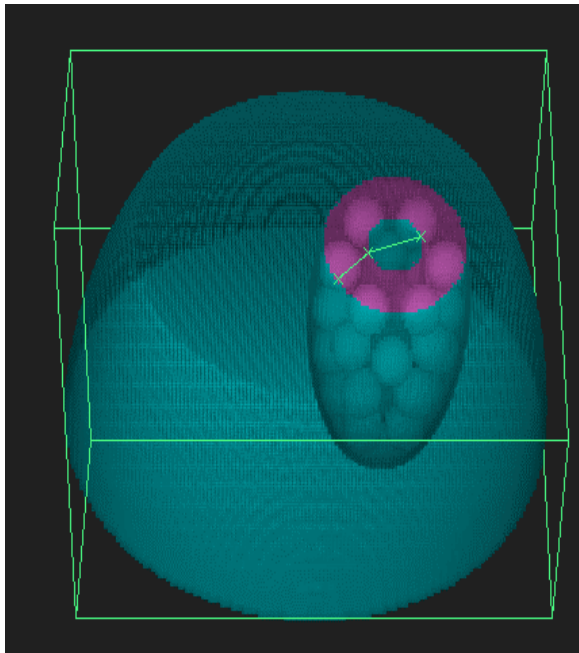
S.S. Furuie<sup>1</sup>, M.F.S. Rebelo<sup>1</sup>, M.A.  
Gutierrez<sup>1</sup>, J.K. Udupa<sup>2</sup>

<sup>1</sup>Divisão de Informática  
Instituto do Coração - HC.FMUSP

<sup>2</sup>Medical Image Processing Group  
University of Pennsylvania

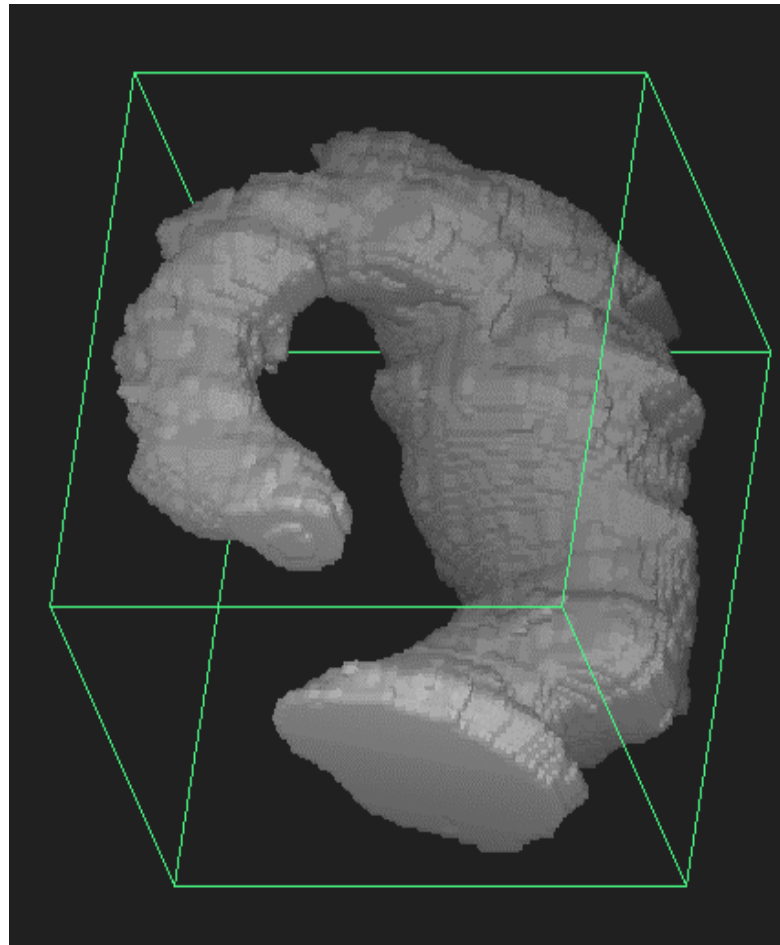
# Motivação

- ✓ Segmentação de objetos é crítica para quantificação e visualização 3D



# Motivação

- Evitar a extenuante tarefa de segmentar centenas de imagens manualmente

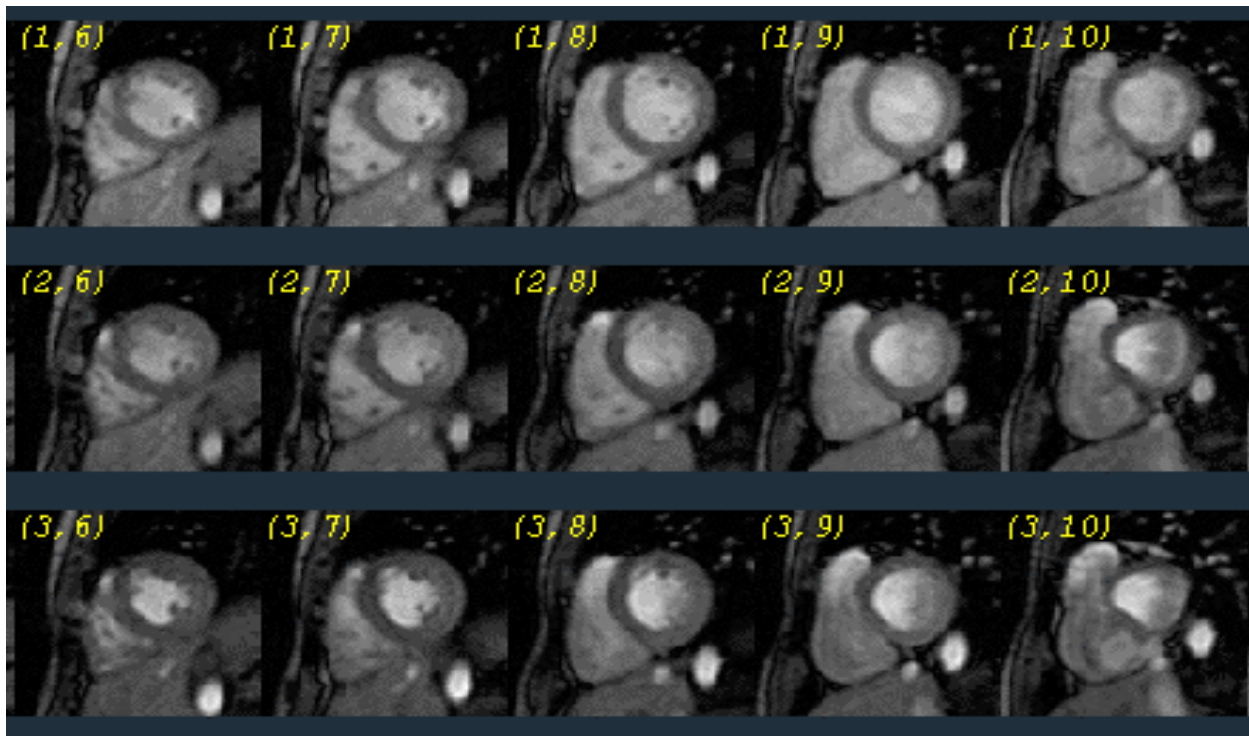


**Segmentação  
manual  
(RM)**

**Aorta com patologia**

# Motivação

- ✓ Processos convencionais (manual e semi-automático) : demorados e cansativos
  - gated MRI : 16 volumes, 12 cortes => 192 img
  - gated SPECT: idem

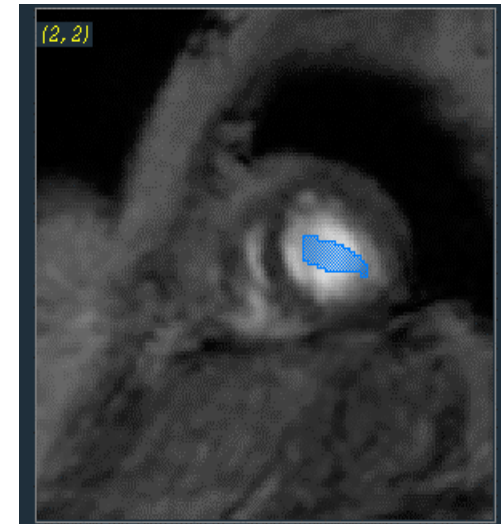


# Objetivos

- ✓ Segmentação de estruturas Médicas 3D
  - Com pequena intervenção humana
  - Independente da modalidade e órgão
  - Aplicável a 3D dinâmicas
  - Inclusão de múltiplos atributos
  - Permita inclusão de informações “a priori”
  - Múltiplos objetos (VE, VD, Miocárdio, ...)

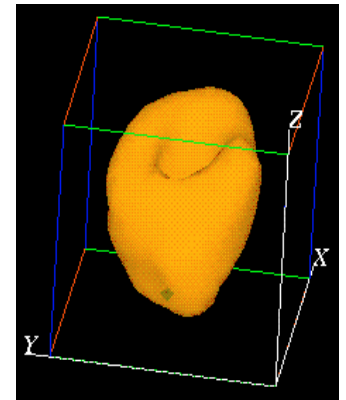
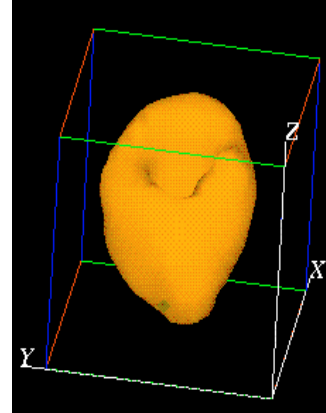
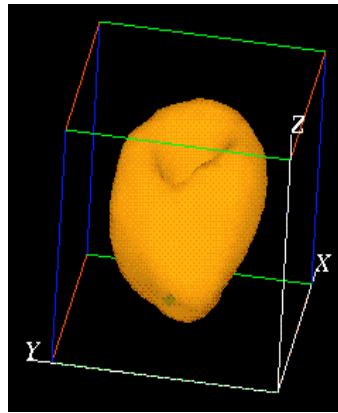
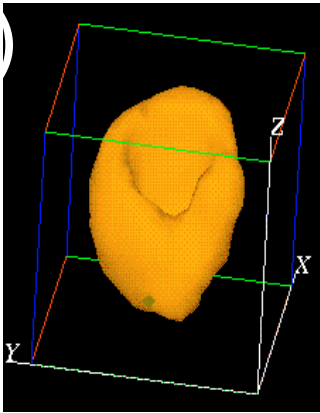
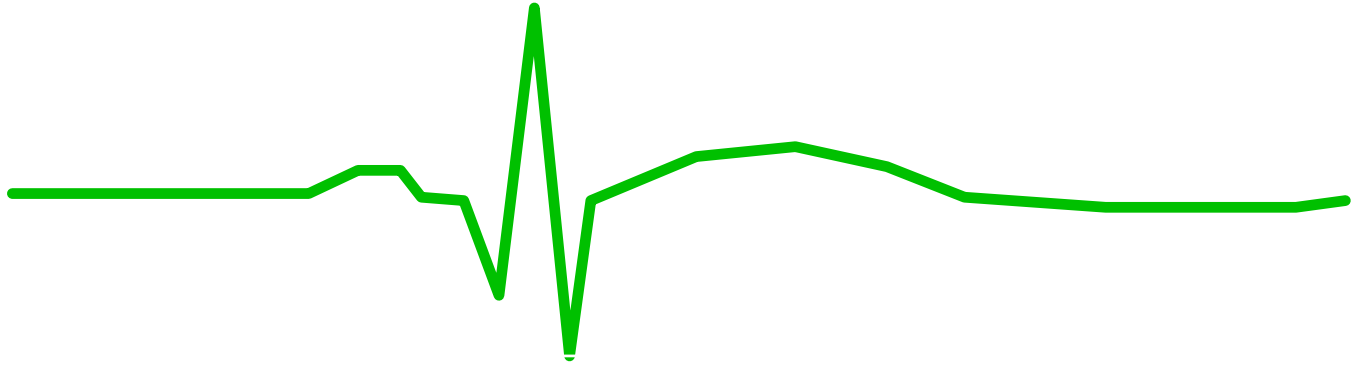
# Protocolo

- ✓ Imagens 3D dinâmicas de Ressonância Magnética do Coração
- ✓ 16 volumes com 12 cortes (256 x 256, 16 bits)
- ✓ 1,64 x 1,64 x 10,0 mm
- ✓ O usuário fornece uma região do objeto de interesse em um dos cortes.



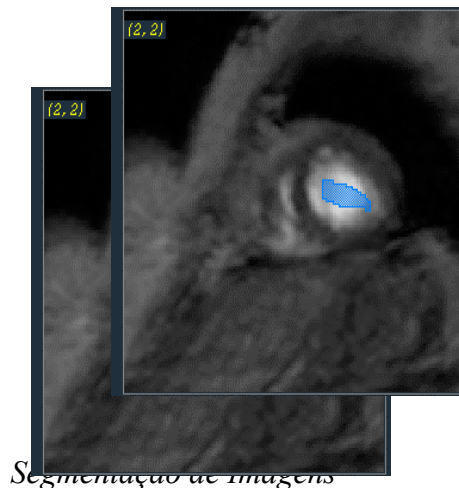


# Gated 3D



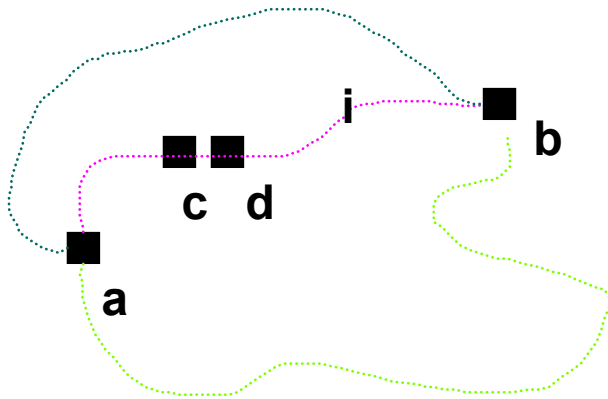
# Hipóteses

- ✓ Possibilidade de treinar o classificador através de uma amostra do objeto com características aproximadamente homogêneas.
- ✓ Os parâmetros estimados, através de um corte (treinamento), são válidos para todo o objeto;



# Metodologia

- ✓ Cálculo do *vínculo* de cada voxel a cada objeto de interesse
  - Vínculo entre dois voxels depende da afinidade e da conectividade espacial em todos os caminhos possíveis



$$V(a,b) = \max_i \{ v(a,b | i) \}$$

$$v(a,b | i) = \min \{ \text{afin}(c,d) \}$$

# Metodologia

- ✓ Afinidade é baseado no vetor de atributos e adjacência

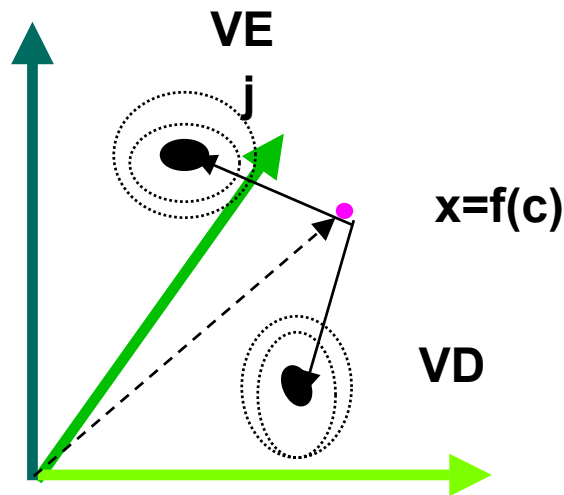
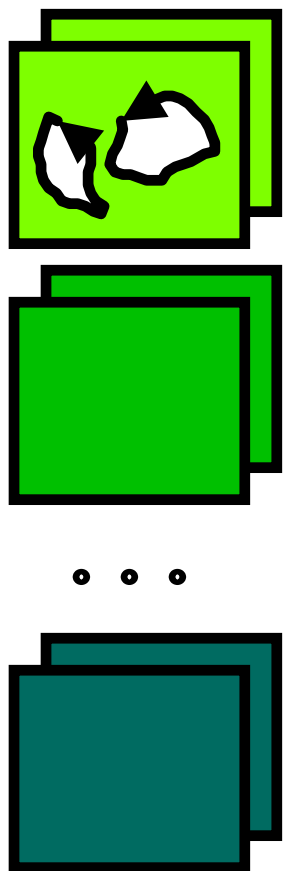
$$\text{afin}(c,d) = h[ \text{adj}(c,d), f(c), f(d), c, d ]$$

$f(x)$ : vetor de atributos sobre o voxel  $x$

- ✓  $\text{afin}(c,d) = \text{adj}(c,d) \cdot h[ f(c), f(d) ]$
- ✓  $h( ) =$  grau de similaridade dos atributos ao objeto
- ✓ programação dinâmica

# associação 'fuzzy'

função de associação : verossimilhança



$$h_j: R^M \rightarrow [0,1]$$

$$h_j(x) = \exp\left(-\frac{1}{2} \cdot d_j^2(x)\right)$$

$$d_j^2(x) = (x - \bar{x}_j)^T \cdot S_j^{-1} \cdot (x - \bar{x}_j)$$

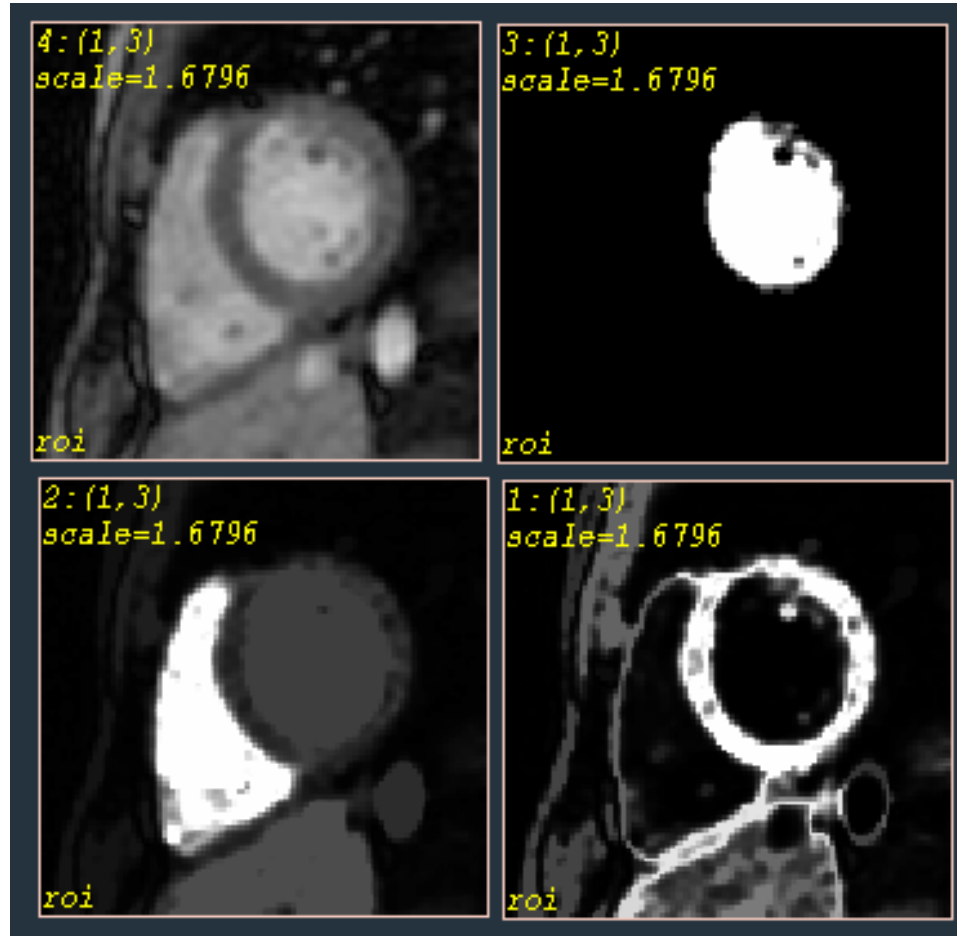
# Algoritmo

O algoritmo, para um dado objeto de interesse, consiste portanto em:

- Estimar os parâmetros estatísticos (média e covariância) do cluster correspondente ao objeto através dos voxels de uma região fornecida pelo usuário;
- Calcular a matriz de covariância inversa;
- Obter o vínculo de cada voxel a cada objeto
  - Para cada voxel, calcular a associação ao objeto
  - Calcular afinidade
  - Obter o vínculo (Programação dinâmica)

# Resultados em RM

Transv.  
RM



VE

VD

Mioc.

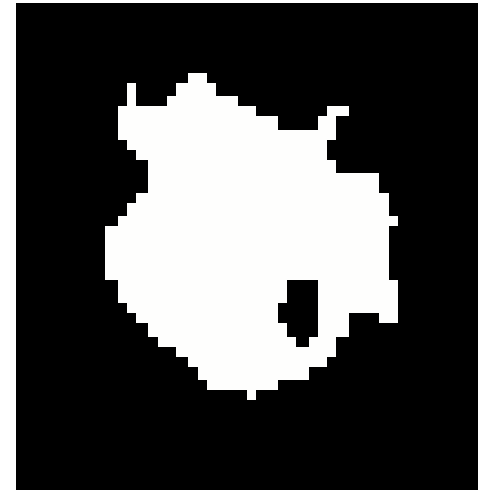
# Competição

Imagem com a  
associação aos  
objetos VE e Mioc.

Músculos



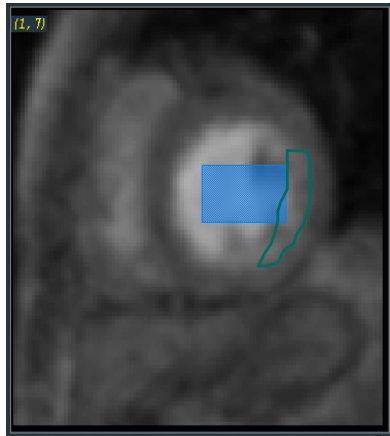
Extração do  
objeto VE



```
if (  $f_{VE}(x) > f_{MIOC}(x)$  )  
then  $f_{VE}(x)=1$   
else  $f_{VE}(x)=0$ 
```

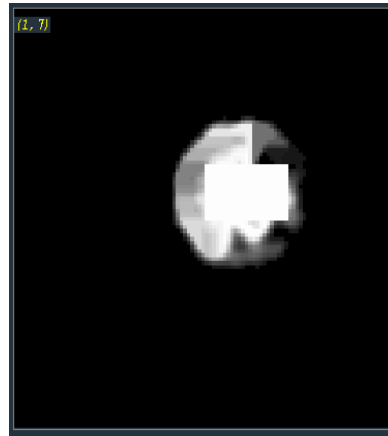


# Cavity Segmentation: results



a

Original



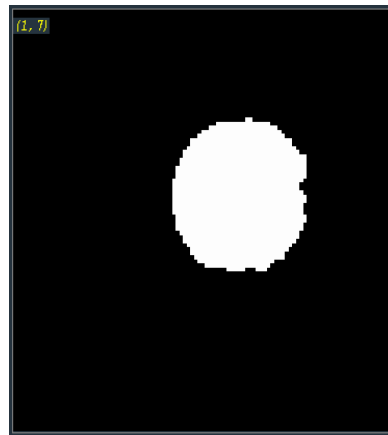
b

CAVI memb.



c

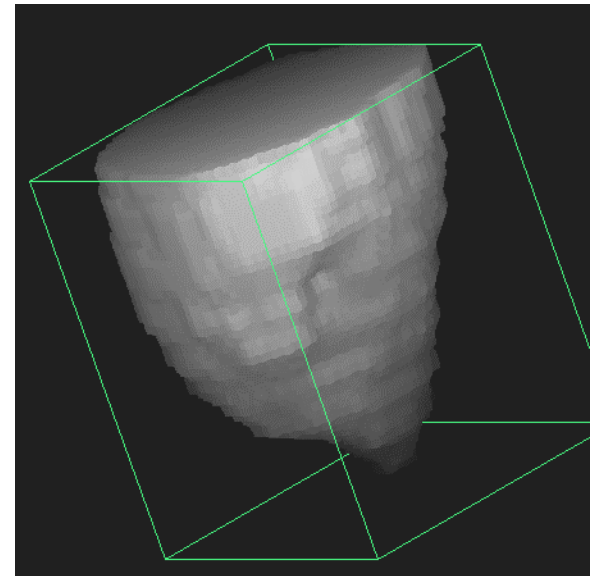
MYOC memb.



d

Segm. Cavity

Proc. time (1 vol.)=21 s  
Ultra Sparc 1/140



# Segmentation of LV myocardium

---

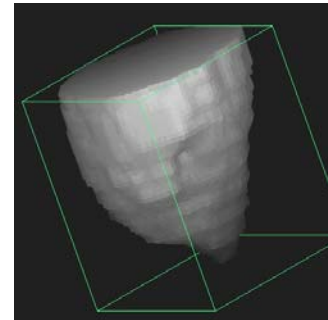
- ✓ From the endocardium surface: incorporation of connected voxels with low cost
- ✓ Cost: gradient, local texture, smoothness and shape

# Cost function

$$\text{cost}(i) = \max\{a.\text{grad}^2(i), b.\text{texture}(i), c.\text{shape}(i)\}$$

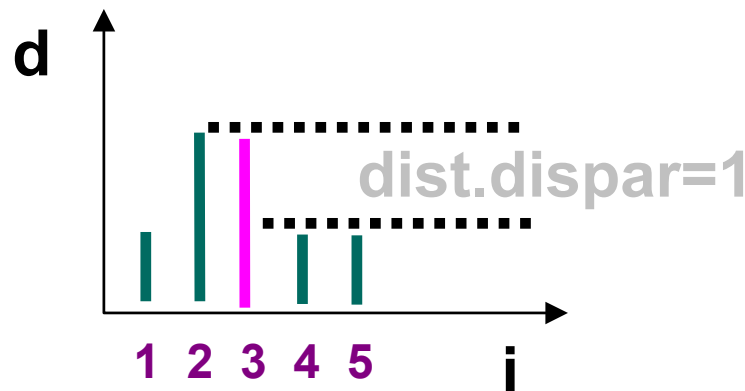
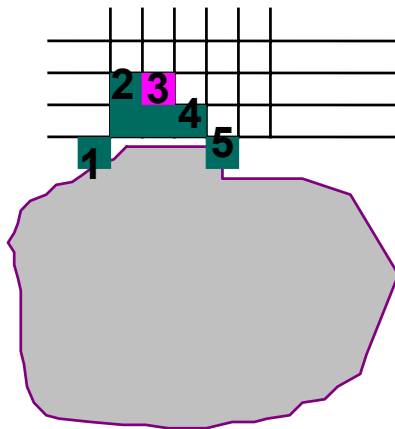
- $\text{grad}(i)$  : local 3D gradiente magnitude  
 $a = 1/\text{std}(\text{grad}^2)$
- $\text{texture}(i)$  : difference of local mean and expected  
 $\text{texture}(i) = (v_i - m)^2$   
 $b = 1/\text{std}(\text{texture})$
- $\text{shape}(i)$ : distance disparity in relation to endocardium  
 $c = 1/2$  (two voxels of distance disparity)

$$\text{Threshold} = \max\{\text{avg}(\text{grad}^2), \text{avg}(\text{texture})\} + 1.0$$



# Smoothness and shape

- ✓  $\text{shape}(i) = \max_{m,n} \{d_m - d_n \mid m,n \text{ in } N(i)\}$   
 $d_k$  : the smallest distance of voxel  $k$  to endocardium surface  
 $N(i)$  : neighbor surface voxels of  $i$ , including  $i$ . The window size used for this work was  $5 \times 5 \times 5$ .
- ✓ Based on distance map

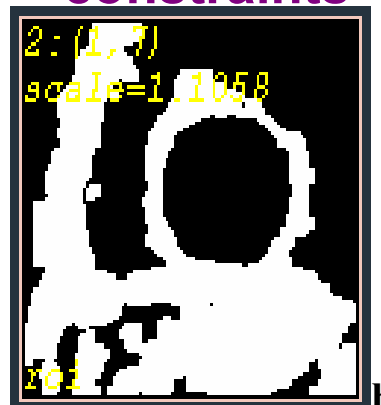


# Results: LV Myocardium

Original



w/o shape constraints

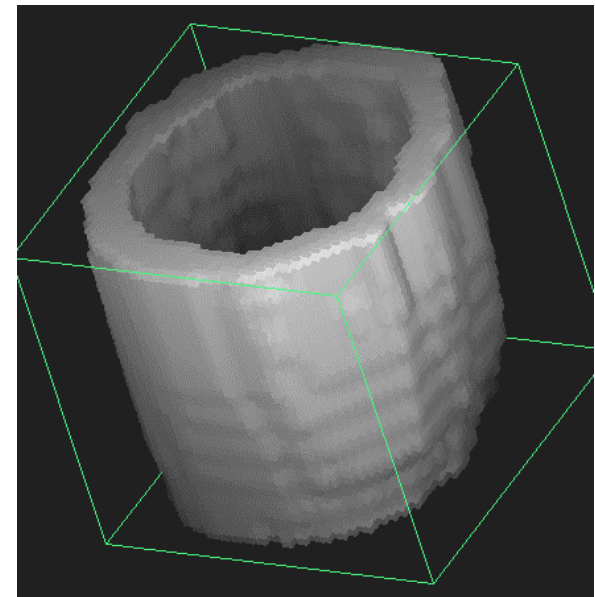


Proposed

Manual

*Segmentação de Imagens*

Proc. time (1 vol.): 4 s  
Ultra Sparc 1/140



# Conclusions

---

- ✓ Two-step technique: cavity and myocardium
- ✓ Semi-automatic volumetric approach
  - Assures connectivity
  - Inclusion of priors:
    - position
    - smoothness and shape
- ✓ Independent of modality, patient, calibration,...
- ✓ Further investigation:
  - objective comparison with manual approach
  - alternative shape cost functions

# Vantagens/Desvantagens

- + treinamento baseado na imagem
  - Independência da escala, equipamento,..
- + método inerentemente 3D, 4D
- + incorpora conectividade espacial e afinidade considerando múltiplos atributos
  
- lento
- assume distribuição normal multivariada para os atributos

# Conclusão

- ✓ Abordagem geral
  - no domínio espacial: imagens 2D, 3D, ...
  - no domínio dos atributos: imagens múltiplas
- ✓ Assegura conectividade
- ✓ Aprende as características dos objetos, considerando as circunstâncias da aquisição (treinamento mínimo)
- ✓ Resultados no sentido 'fuzzy'
- ➔ Pesquisas futuras:
  - agilização do algoritmo
  - texturas específicas que ressaltem objetos
  - medidas de afinidade



# Resultados

- ✓ Phantoms 3D
  - bons resultados
- ✓ MRI do coração (4D)
  - aquisição degradada
  - classificação autom.
  - vários falsos negativos
- ✓ A investigar
  - outra aquisição
  - definir novos affinity functions
  - refinamento (simulated annealing, semi-automático, ...)

---

# Abordagem usando 'Level set' para segmentação de imagens

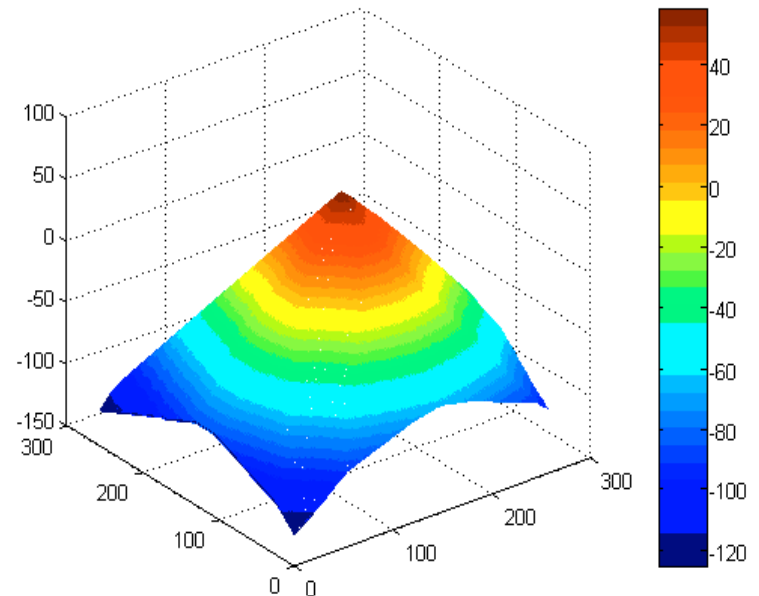
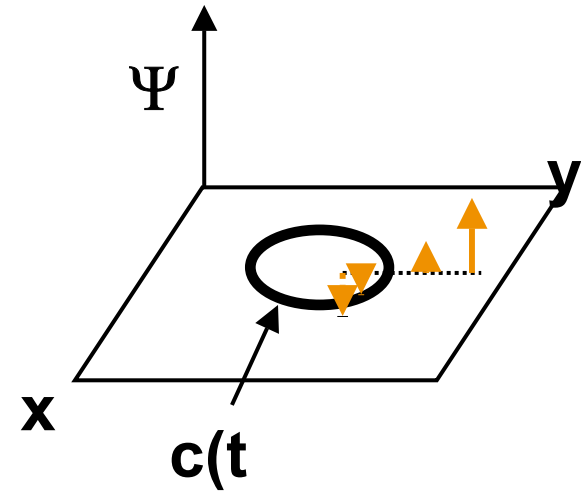
**Unidade de Pesquisa e Desenvolvimento**  
**Serviço de Informática**  
**InCor**

# 4 LEVEL SET FUNCTIONS

$$\Psi : \mathbb{R}^N \rightarrow \mathbb{R}$$

$$\Psi(c(t), t) = 0$$

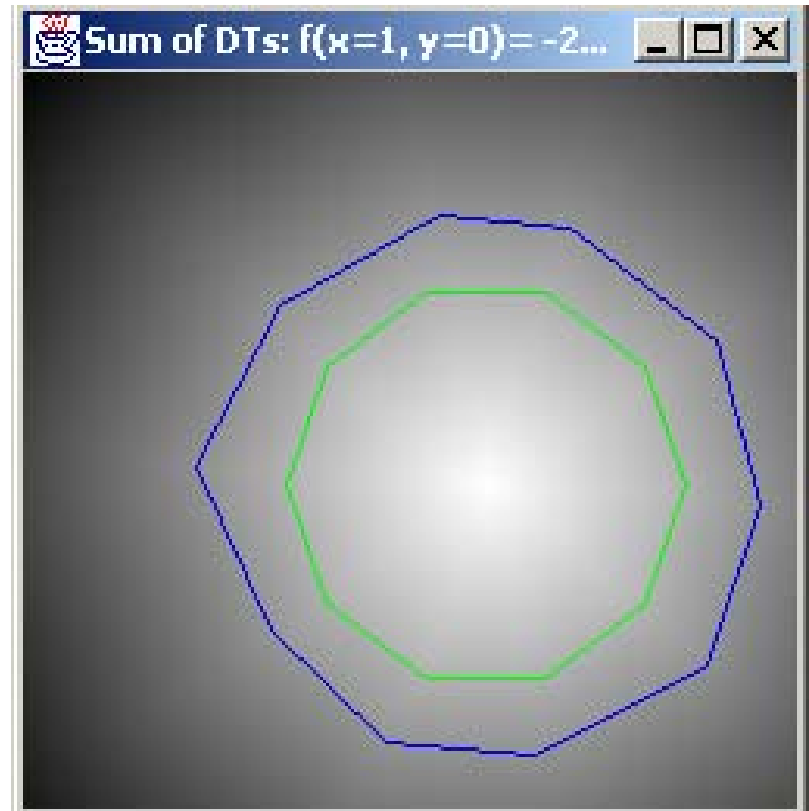
Signed distance transform



## 6 Cost function: similarity

- ✓ Similarity
  - pointwise curvatures of level set  $\Psi_K$
  - Gaussian curvature

$$\Psi_K(\mathbf{r}) = \Psi_S(\mathbf{r}) + \Psi_D(\mathbf{r})$$



Level set  $\Psi_K$  used for curvature estimation.

# Motivação

- ✓ Metodologia para segmentação de imagens
  - n-dimensional
  - contornos complexos
- ✓ Descrição dos objetos identificados
- ✓ Análise e visualização



# Snakes

---

- ✓ Não possibilita intersecção de contornos
- ✓ 2D
- ✓ Dependente da modelagem das forças/velocidades
- ✓ Dependente de pesos dos termos da função

# Descrição

- ✓ Baseado em modelos físicos de propagação de interfaces e modelos deformáveis
  - fluído
  - chamas
  - crescimento de cristais
- ✓ Solução de PDE do tipo Hamilton-Jacobi

## PARA MAIS INFORMAÇÕES...

Shape modeling with front propagation: a level set approach

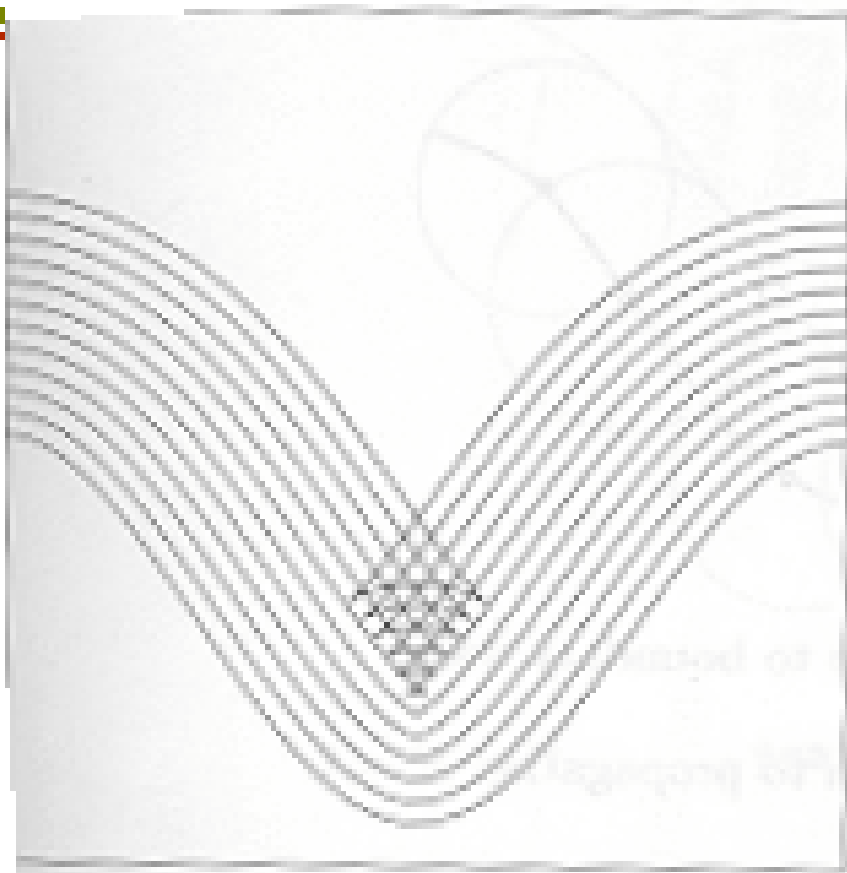
R Malladi, JA Sethian, BC Venturi. IEEE PAMI 17(2):158-175, feb1995

# Vantagens

---

- ✓ Sem suposições a priori de topologias
- ✓ Suporta formas complexas e nDimensionais
- ✓ Pontos fracos
  - modelagem da velocidade de propagação
  - condições de parada



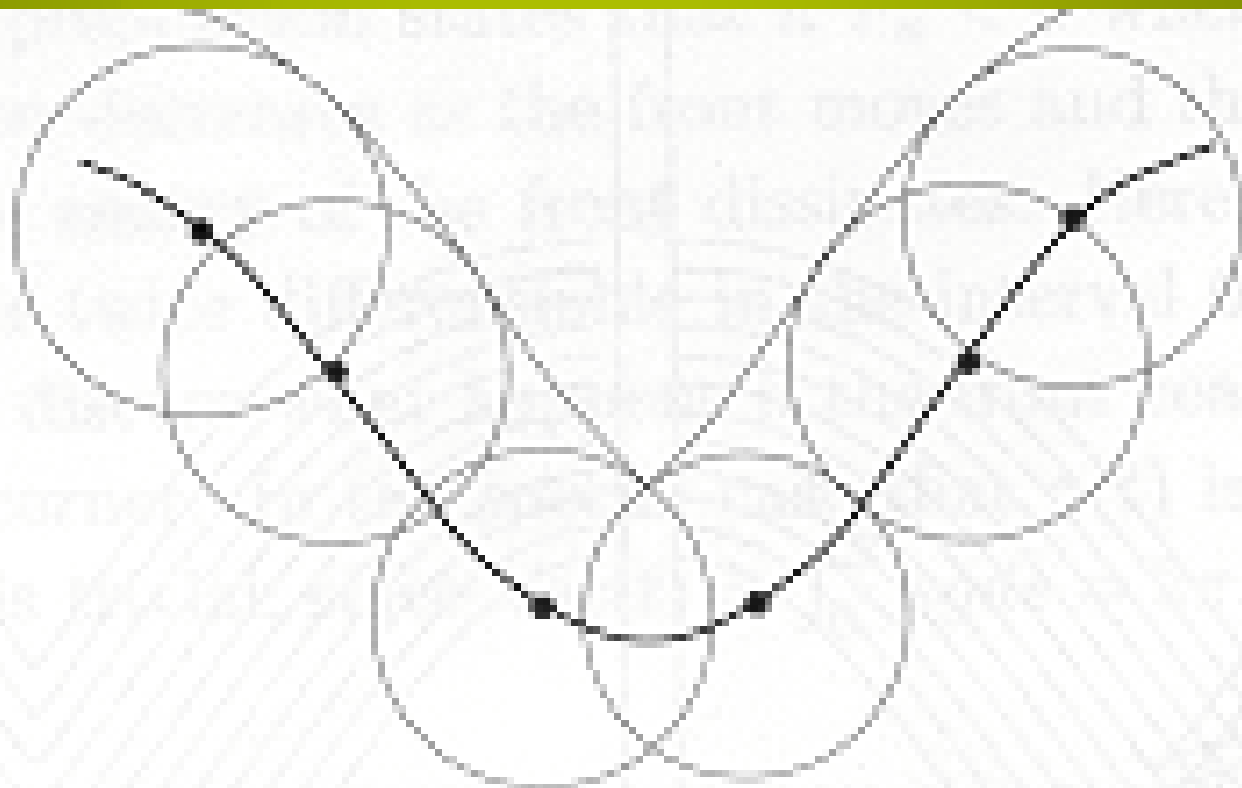


(a) Swallowtail ( $F = 1.0$ )



(b) Entropy solution ( $F = 1.0$ )

Fig. 2.3. Cosine curve propagating with unit speed.



Globally closest points to boundary data

Fig. 2.4. Huygens' solution to propagation  $F = 1$

# Metodologia

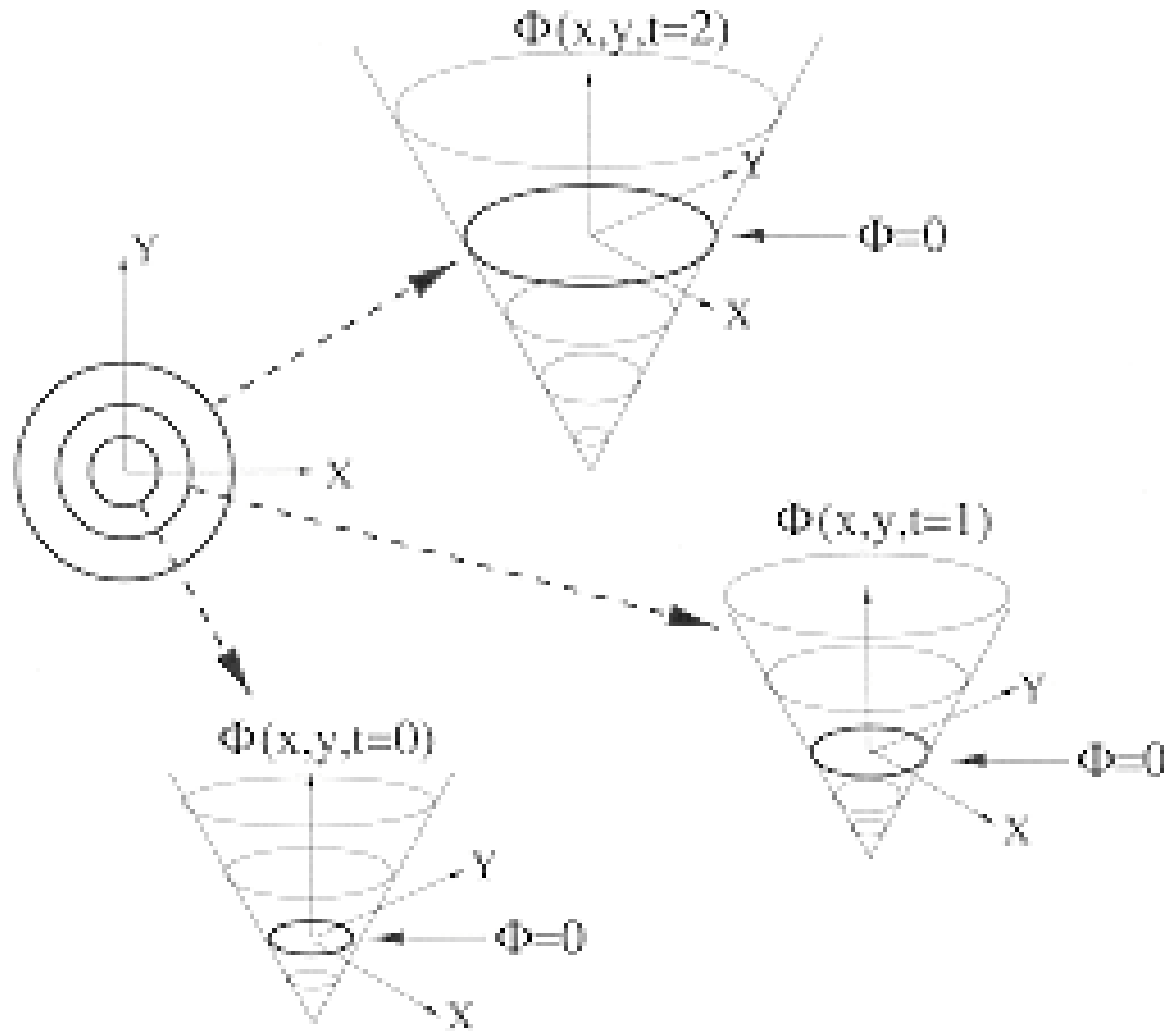
- ✓ Dado um contorno
- ✓ Gerar uma hipersuperfície
- ✓ No qual o contorno é solução

$$\gamma(t) : [0, \infty) \rightarrow \mathbb{R}^N$$

$$\Psi(\vec{x}, t) : \mathbb{R}^N \rightarrow \mathbb{R}$$

$$\gamma(t) = (\vec{x} \mid \Psi(\vec{x}, t) = 0)$$

# Exemplo de hipersuperfície



# Metodologia: level set

$$\gamma(t) = (\vec{x} \mid \Psi(\vec{x}, t) = 0)$$

$$\Psi(\vec{x}, t) = 0$$

$$\frac{\partial \Psi}{\partial t} + \frac{\partial \Psi}{\partial x_1} \cdot \frac{dx_1}{dt} + \frac{\partial \Psi}{\partial x_2} \cdot \frac{dx_2}{dt} + \dots = 0$$

$$\frac{\partial \Psi}{\partial t} + \vec{\nabla} \Psi \cdot \frac{d\vec{x}}{dt} = 0$$

$$\frac{\partial \Psi}{\partial t} + |\vec{\nabla} \Psi| \cdot \left( \vec{n} \cdot \frac{d\vec{x}}{dt} \right) = 0$$

- ✓ Diferenciando...
- ✓ Notando que gradiente é normal ao contorno
- ✓ PDE do tipo Hamilton-Jacobi
- ✓ Condição inicial

# Modelagem da velocidade

$$\frac{\partial \Psi}{\partial t} + |\vec{\nabla} \Psi| \cdot \left( \vec{n} \cdot \frac{d\vec{x}}{dt} \right) = 0$$

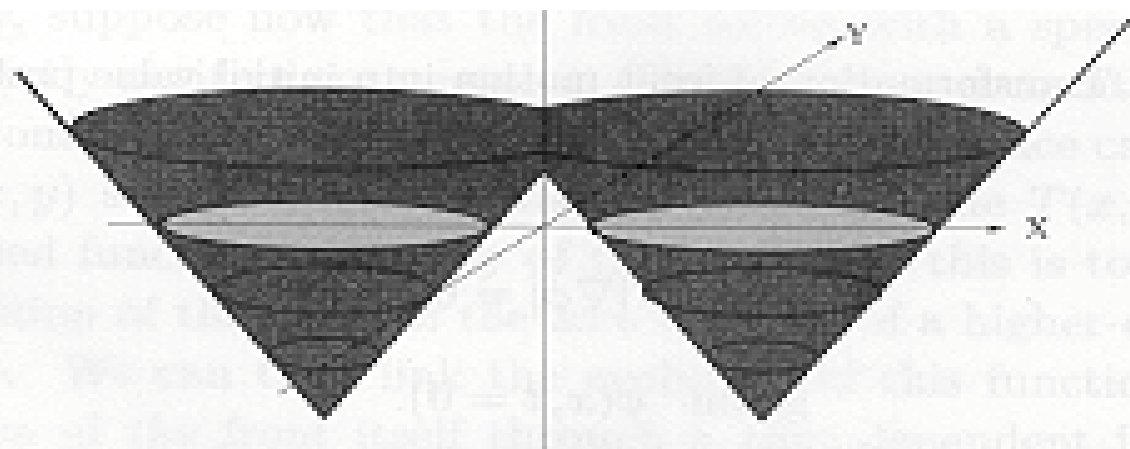
$$\frac{\partial \Psi}{\partial t} + |\vec{\nabla} \Psi| \cdot F = 0$$

$$\vec{n} = \frac{\vec{\nabla} \Psi}{|\vec{\nabla} \Psi|}$$

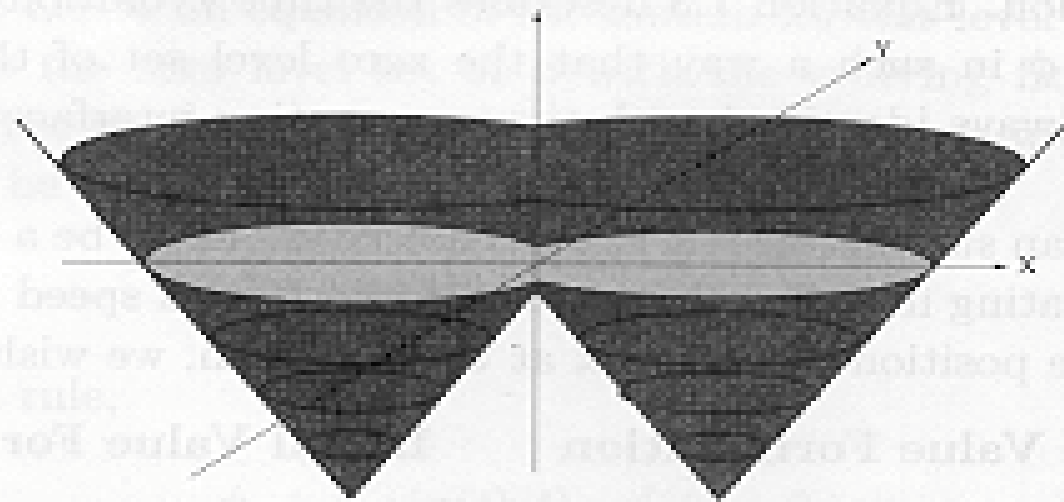
$$K = \nabla \cdot \vec{n}$$

$$F = g \cdot (F_0 + F(K))$$

$$g = \frac{1}{1 + |\nabla G_\sigma * I(x, y)|}$$



The level set surface  $\phi$  (dark gray):  
Two separate initial fronts (in light gray).



Later in time: the interface topology has changed,  
yielding a single curve as the zero level set.

Fig. 1.6. Topological change.

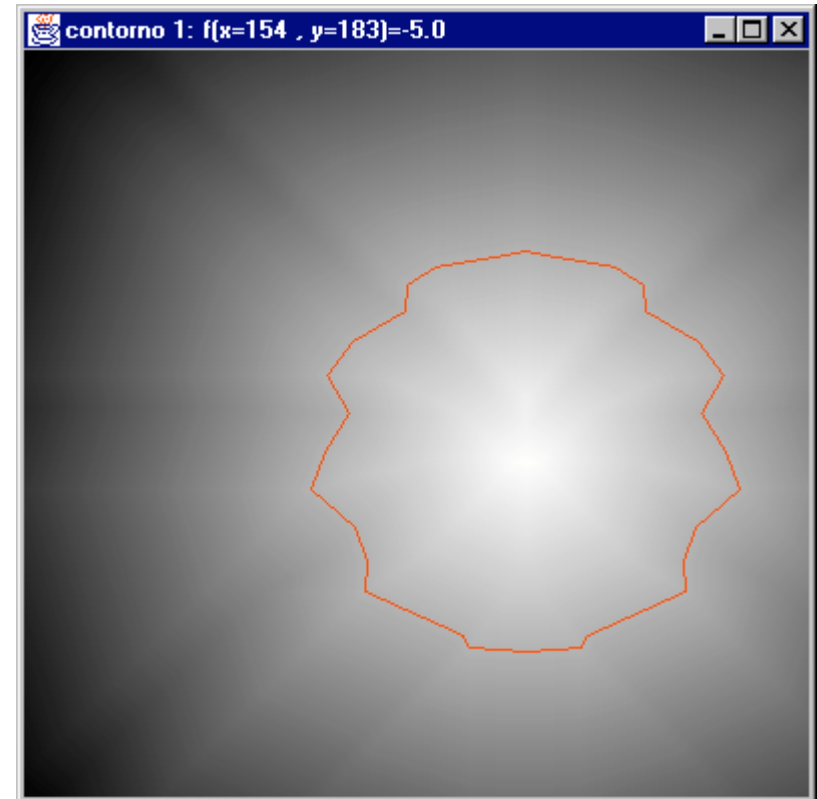
# Discretização

$$\Psi_{i,j}^{m+1} = \Psi_{i,j}^m - \Delta T \cdot \left| \vec{\nabla} \Psi_{i,j}^m \right| \cdot F_{i,j}$$

$$(x(s), y(s))$$

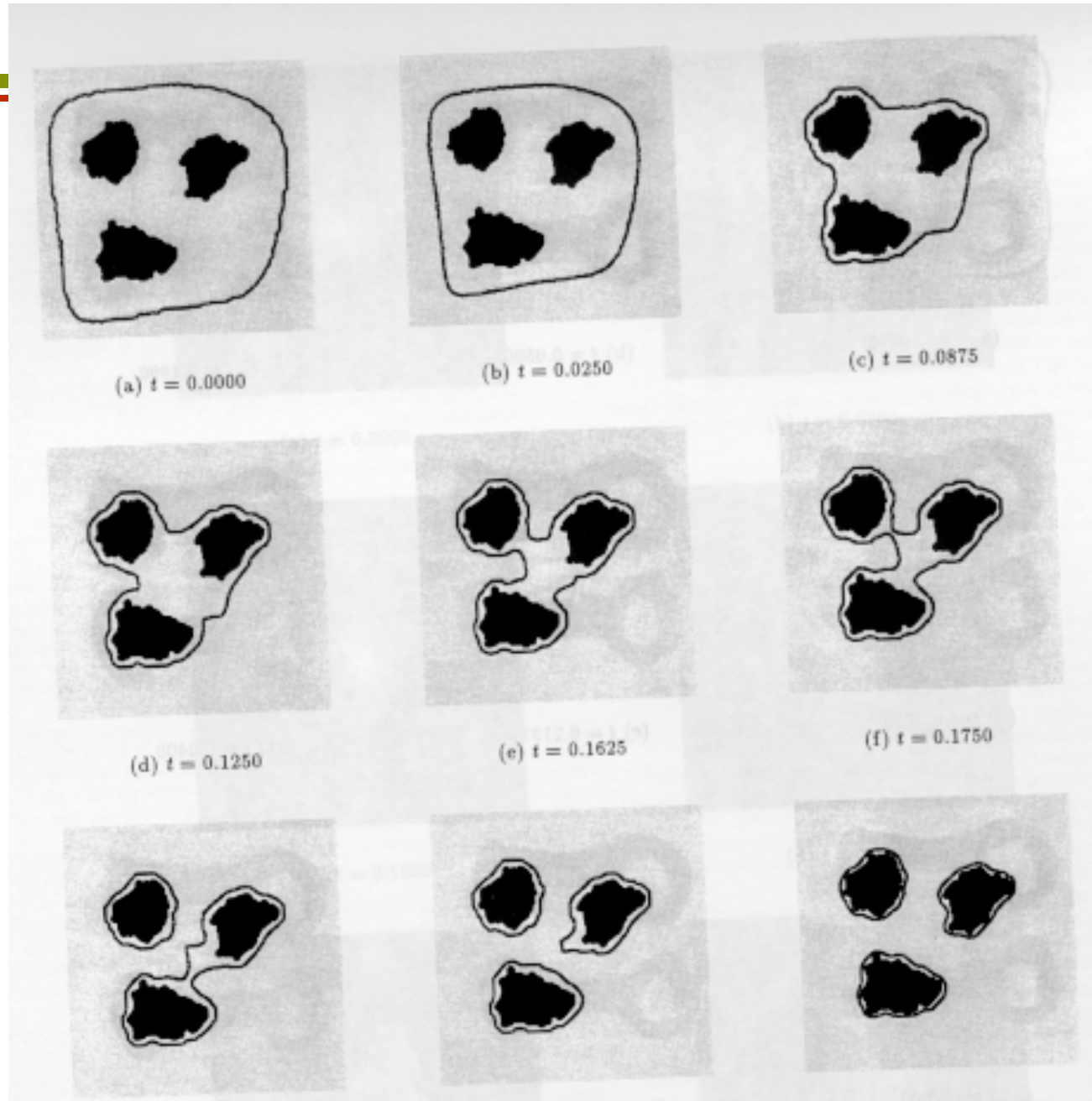
$$\vec{n} = (y_s, -x_s) / \sqrt{(x_s^2 + y_s^2)}$$

$$K = \frac{y_{ss} \cdot x_s - x_{ss} \cdot y_s}{\sqrt{(x_s^2 + y_s^2)}^3}$$

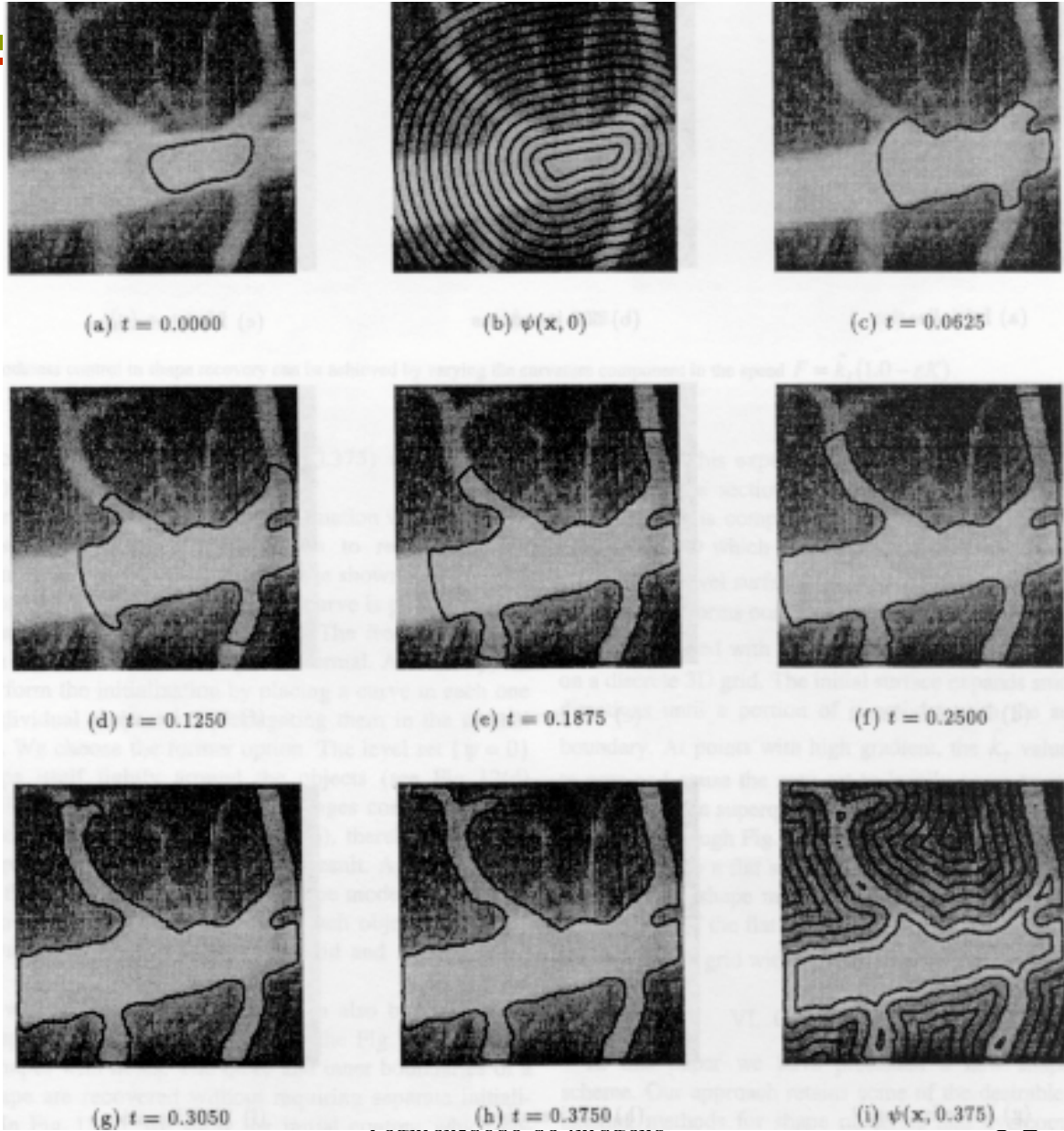


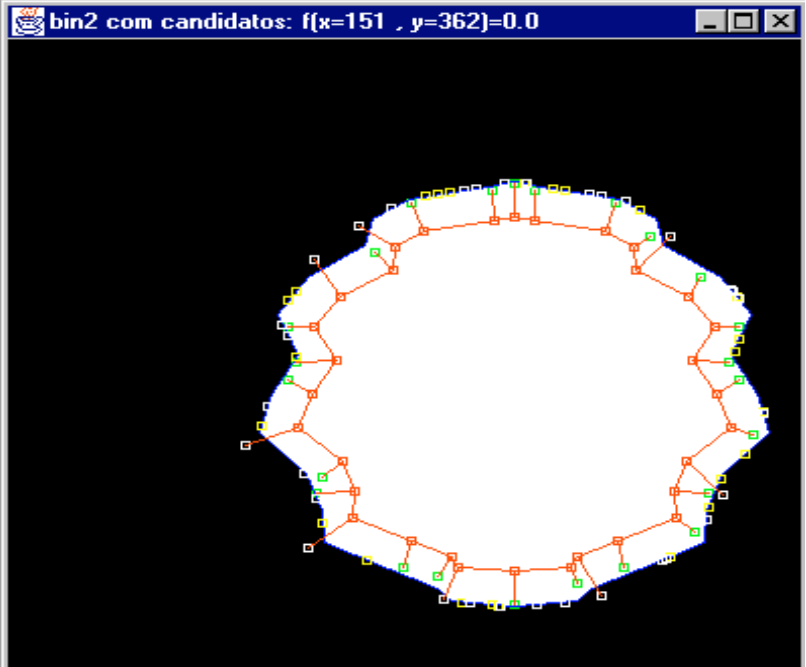
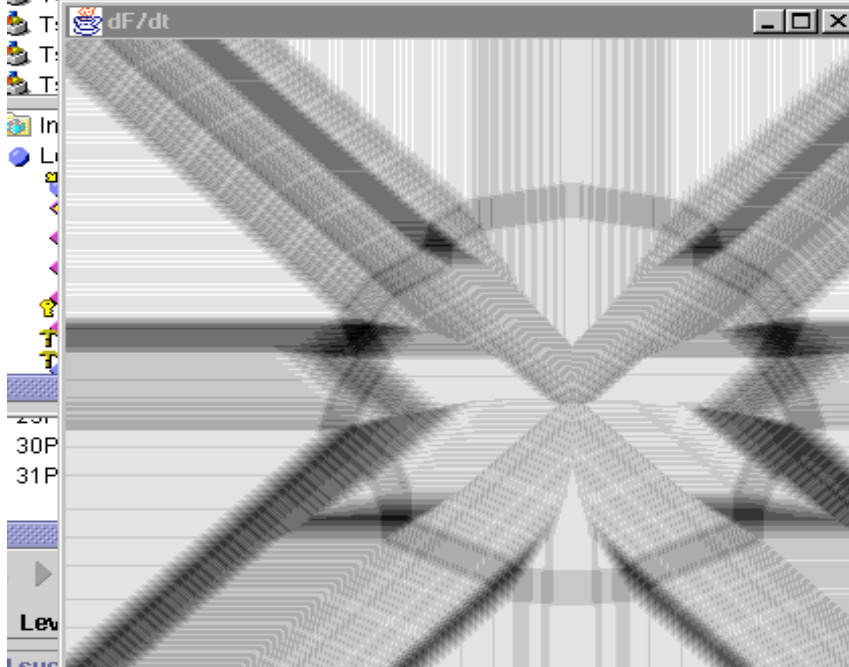
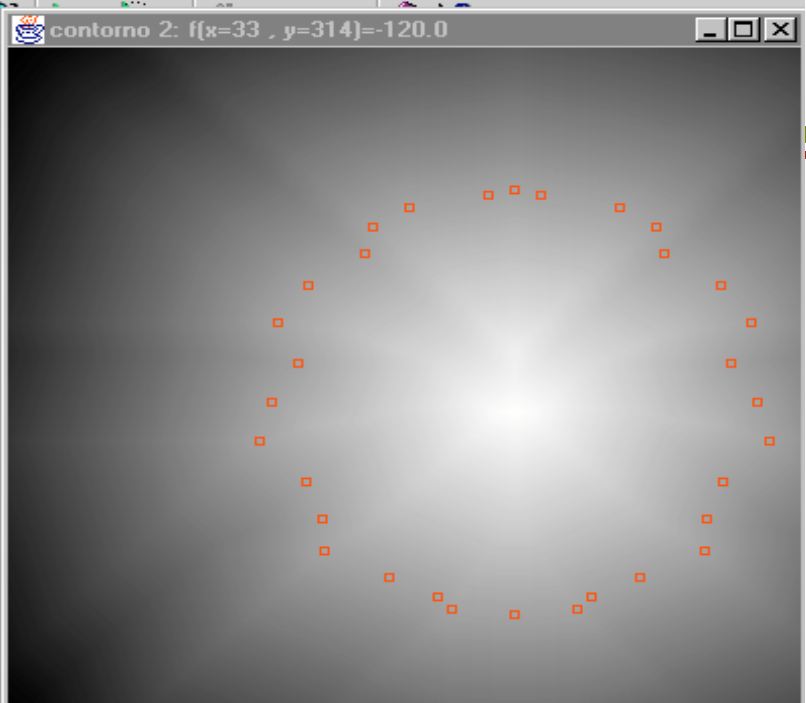
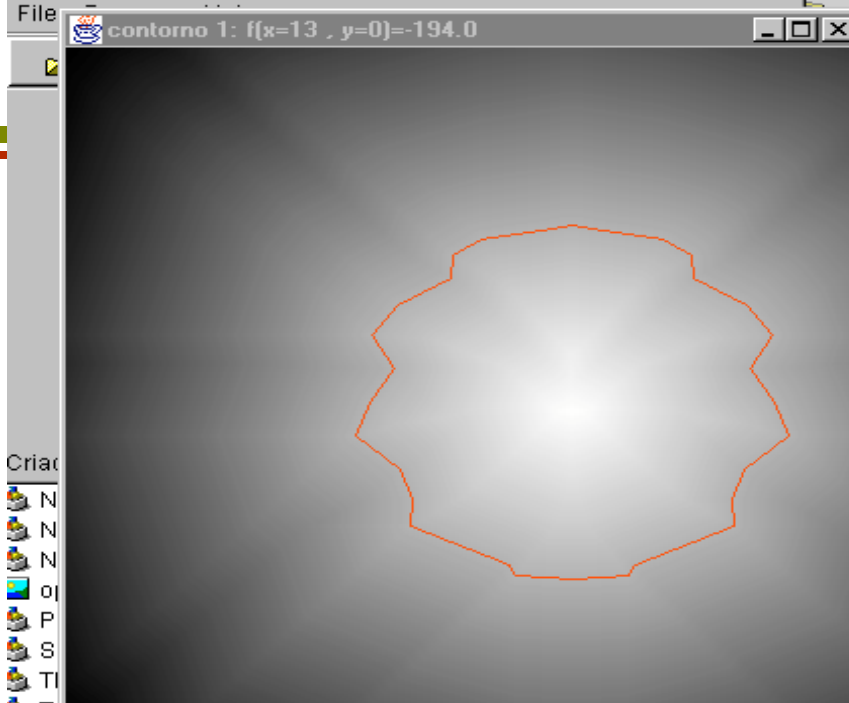


# split



# ramificação





- Criar
- N
- N
- N
- oi
- P
- S
- TI
- T:
- T:
- T:
- In
- L
- 20P
- 30P
- 31P
- Lev
- Lev

# Outras Aplicações

---

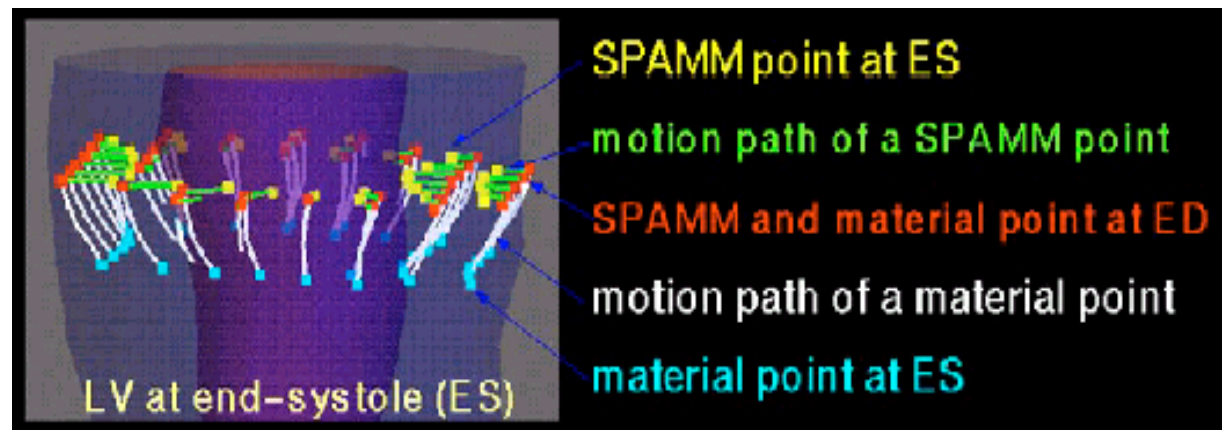
- ✓ Restauração de imagens
- ✓ Evolução de contornos
- ✓ Propagação de combustão, fluídos, sismologia, crescimento de cristais, ...
- ✓ Difusão, deposição de materiais, ...

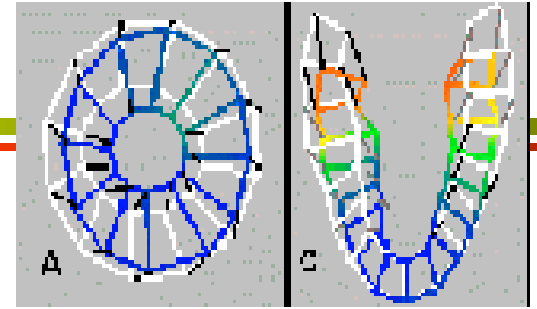
---

# Segmentação de estruturas médicas baseado em contornos ativos e elementos finitos

# Tema: Modelos deformáveis

- ✓ Importância
  - motivação
  - aplicações
  
- ✓ Revisão de métodos 2D/3D
  - fundamentos/metodologia
  - vantagens/desvantagens
  
- ✓ Conclusões

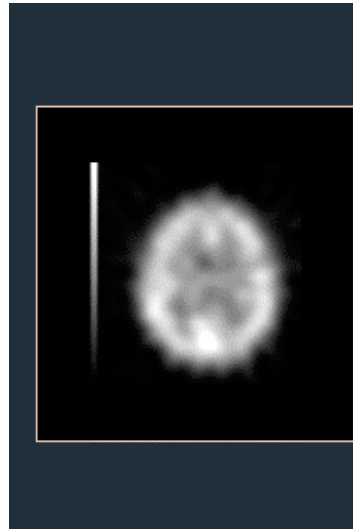
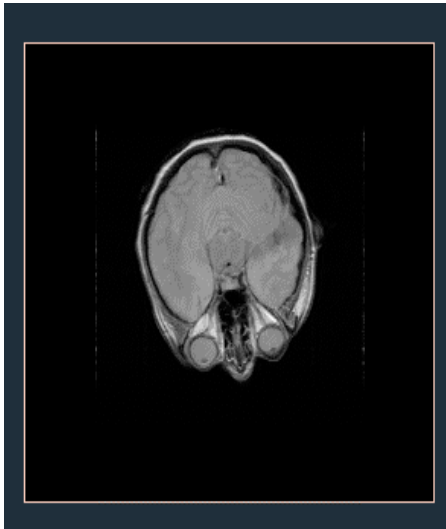




- ✓ Avaliação funcional dos músculos cardíacos
  - Global
    - variação do volume/área/energia cinética
  - Regional
    - pela movimentação regional do endocárdio
    - pelo encurtamento da parede do miocárdio
      - espessura do miocárdio
  - Local
    - pela trajetória de pontos do músculo (tracking)
    - pela velocidade dos pontos do músculo

# Aplicações

- Quantificação da avaliação cardíaca
- Visualização da deformação (fisiologia) local
- Alinhamento e fusão de imagens
- Interpolação



*Segmentação de Imagens*



# Segmentação de estr. deform.

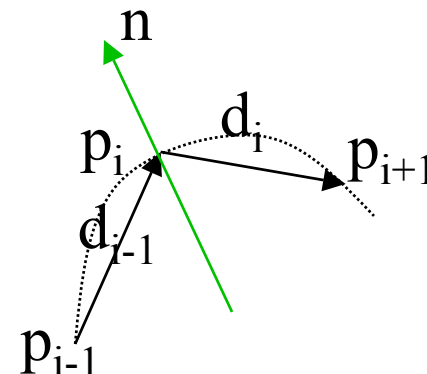
- ✓ Baseados na intensidade do pixel/voxel
  - optical flow, region-growing, conectividade,..
  - ad hocs
- ✓ Baseados em contornos/superfícies
  - Snakes
  - Balloon
  - Topological snakes
- ✓ Baseados em marcas fiduciais (int. e ext.)
  - MR tagging

# Snakes

- Objetivo: segmentação (obtenção de contornos e superfícies)
- Dado: imagem (intensidades) e solução inicial
  - obter uma função de energia (interna + externa) em função parametrizada
  - minimizar em relação ao contorno
    - analiticamente
    - numericamente: steepest descent, CG, ...

# Snakes: equilíbrio de forças

- 1) P/ cada vértice do contorno inicial, obter:
  - forças internas (curvatura,..)
  - forças externas (gradiente ..)
- 2) Determinação iterativa da velocidade e posição



$$\vec{a}_i^{(k+1)} = \frac{1}{m_i} \cdot \vec{f}_i^{(k+1)}$$

$$\vec{v}_i^{(k+1)} = \vec{v}_i^{(k)} + \vec{a}_i^{(k+1)} \cdot \Delta t$$

$$\vec{p}_i^{(k+1)} = \vec{p}_i^{(k)} + \vec{v}_i^{(k+1)} \cdot \Delta t$$

$$curv = \frac{\vec{d}_i - \vec{d}_{i-1}}{\|\vec{d}_i - \vec{d}_{i-1}\|}$$

$$f_{int} = (curv \otimes filtro)_{radial}$$

$$f_{ext} = (-\nabla I)_{radial}$$

# Snakes: fundamento

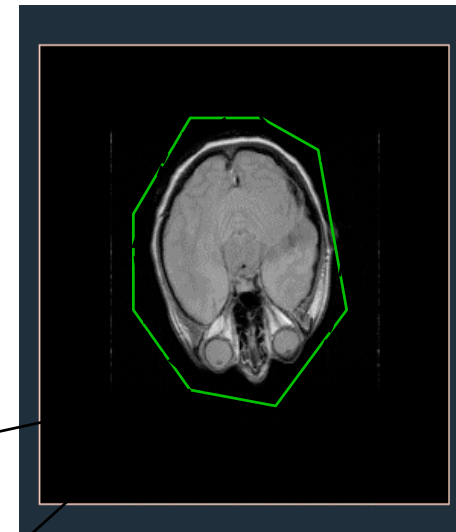
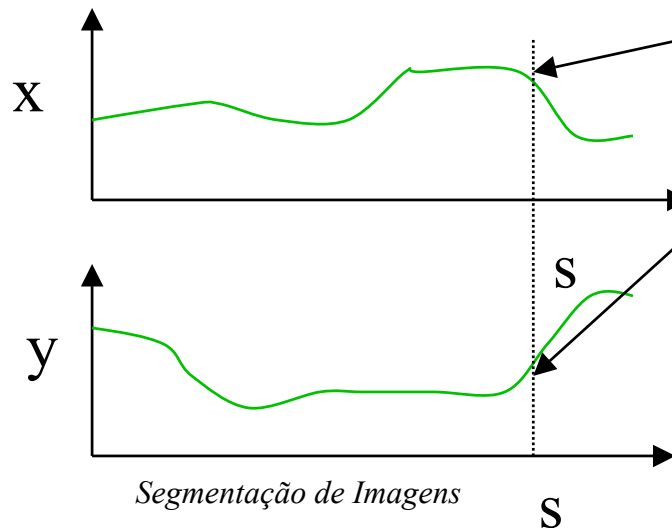
$c(s) : \text{config}$

$$E(c) = E_I(c) + E_E(c)$$

$$E(c) = \int_c \left[ w_1 \cdot \left| \frac{\partial c}{\partial s} \right|^2 + w_2 \cdot \left| \frac{\partial^2 c}{\partial s^2} \right|^2 \right] ds + \int_c w_3 \cdot P(c) \cdot ds$$

$$\min_c E(c)$$

$$c(s) = (x(s), y(s))$$



$$\min_c E(c)$$

$$E(c) = \int_c F(s; c, c', c'') \cdot ds$$

$$c(s), c', c''$$

$$F(s; c(s), c'(s), c''(s))$$

$$F(s; c + \varepsilon \cdot v(s), c' + \varepsilon \cdot v', c'' + \varepsilon \cdot v'') = F(s; c, c', c'') + \varepsilon \cdot v \cdot \frac{\partial F}{\partial c} + \varepsilon \cdot v' \cdot \frac{\partial F}{\partial c'} + \varepsilon \cdot v'' \cdot \frac{\partial F}{\partial c''} + \dots$$

$$\min_{\varepsilon} E \Rightarrow \frac{\partial E}{\partial \varepsilon} = 0 \Rightarrow \text{Euler-Lagrangian}$$

$$\frac{\partial F}{\partial c} - \frac{d}{ds} \left( \frac{\partial F}{\partial c'} \right) + \frac{d^2}{ds^2} \left( \frac{\partial F}{\partial c''} \right) = 0$$

# Definição adequada da Energia

- ✓ Elasticidade (suavidade do contorno)
- ✓ Rigidez (curvaturas )
- ✓ Informações de borda

$c(s) : \text{config}$

$$E(c) = E_I(c) + E_E(c)$$

$$E(c) = \int_c [w_1 \cdot \left| \frac{\partial c}{\partial s} \right|^2 + w_2 \left| \frac{\partial^2 c}{\partial s^2} \right|^2] ds + \int_c w_3 \cdot P(c) \cdot ds$$

$$P(c(s)) = -\left\| \nabla (G_\rho * \text{Im}) \right\|$$

# Solução da minimização

Equilíbrio

$$E(c) = \int_c [w_1 \cdot |c'(s)|^2 + w_2 |c''(s)|^2] ds + \int_c w_3 \cdot P(c) \cdot ds$$

*Euler – Lagrange* =>

$$-\frac{d}{ds} \left( w_1 \cdot \frac{dc}{ds} \right) + \frac{d^2}{ds^2} \left( w_2 \cdot \frac{d^2 c}{ds^2} \right) + \nabla P(c) = 0$$

Dinâmica

$$\frac{dc}{dt} - \frac{d}{ds} \left( w_1 \cdot \frac{dc}{ds} \right) + \frac{d^2}{ds^2} \left( w_2 \cdot \frac{d^2 c}{ds^2} \right) + \nabla P(c) = 0$$

# Implementação: difer.finitas

$$-w_1.c'' + w_2.c'''' = -\nabla P(c)$$

$$c(i) = \begin{bmatrix} x(i) \\ y(i) \end{bmatrix}$$

$$x'(i) = \frac{x(i+1) - x(i)}{h}$$

$$x''(i) = \frac{x(i-1) - 2x(i) + x(i+1)}{h^2}$$

$$x''''(i) = (x''(i))''$$

...

$$-w_1 \cdot \frac{x(i-1) - 2x(i) + x(i+1)}{h^2} + w_2 \cdot [a_1 \cdot x(i-2) + \dots + a_5 \cdot x(i+2)] = -\nabla P(i)$$

Para cada ponto  $i$  do contorno em equilíbrio



# Diferenças finitas

$$0 + 0 + \dots + b_1 \cdot x(i-2) + b_2 \cdot x(i-1) + \dots + b_5 \cdot x(i+2) = -\nabla P(i)$$

$$\begin{bmatrix} b_1 & b_2 & \dots & 0 \\ 0 & b_1 & \dots & \\ 0 & & b_2 & \\ 0 & & b_1 & b_2 \\ 0 & & & b_1 \end{bmatrix} \begin{bmatrix} x(1) \\ x(2) \\ \dots \\ x(n) \end{bmatrix} = \begin{bmatrix} -\nabla P(1) \\ -\nabla P(2) \\ \dots \\ -\nabla P(n) \end{bmatrix}$$

$$A \cdot \vec{X} = \vec{f}(\vec{X})$$

Sistema de equações com dependência implícita

- Métodos numéricos iterativos

# Modo dinâmico

$$\frac{dc}{dt} - \frac{d}{ds} \left( w_1 \cdot \frac{dc}{ds} \right) + \frac{d^2}{ds^2} \left( w_2 \cdot \frac{d^2 c}{ds^2} \right) + \nabla P(c) = 0$$

$$\frac{d\vec{X}}{dt} + A \cdot \vec{X} = \vec{f}(\vec{X})$$

$$\frac{X^t - X^{t-1}}{\Delta t} + A \cdot X^{t-1} = f(X^{t-1})$$

$$X^t = (I - \Delta t \cdot A) \cdot X^{t-1} + \Delta t \cdot f(X^{t-1})$$

Solução iterativa até atingir o equilíbrio

# Snakes: Vantagens/desvantagens

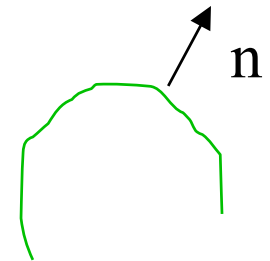
---

- + Emula modelo físico. Imagens são apenas representações de modelos físicos.
- Processos reais são complexos p/ modelar
- Constantes desconhecidas
- Pesos relativos ?
- Apenas segmentação e alinhamento. Tracking exige modelagem mais precisa.

# Balloon

- ✓ Simulated annealing
- ✓ Region-growing com função custo
- ✓ Modelo físico de expansão
  - Pode superar bordas espúrias

$$f = w_1 \cdot \vec{n}(s) - k \cdot \frac{\nabla P(s)}{\|\nabla P\|}$$

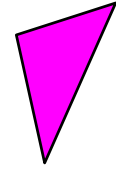
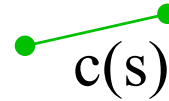


# Elementos finitos (FEM)

- ✓ O elemento como um todo é modelado e segue uma lei.
  - Valores internos ao elemento podem ser estimados a partir dos extremos
- ✓ Minimiza-se o resíduo total considerando as funções de interpolação



Modelo  $-w_1.c''+w_2.c'''' = -\nabla P(c)$



Resíduo

$$-w_1.c''+w_2.c''''+\nabla P(c) = 0$$

$$c(s) = c_1.N_1(s) + c_2.N_2(s)$$

$$residuo_i = \int_{c1}^{c2} [-w_1.c''+w_2.c''''+\nabla P(c)].N_i(s)ds = 0$$

$$A.c = f$$

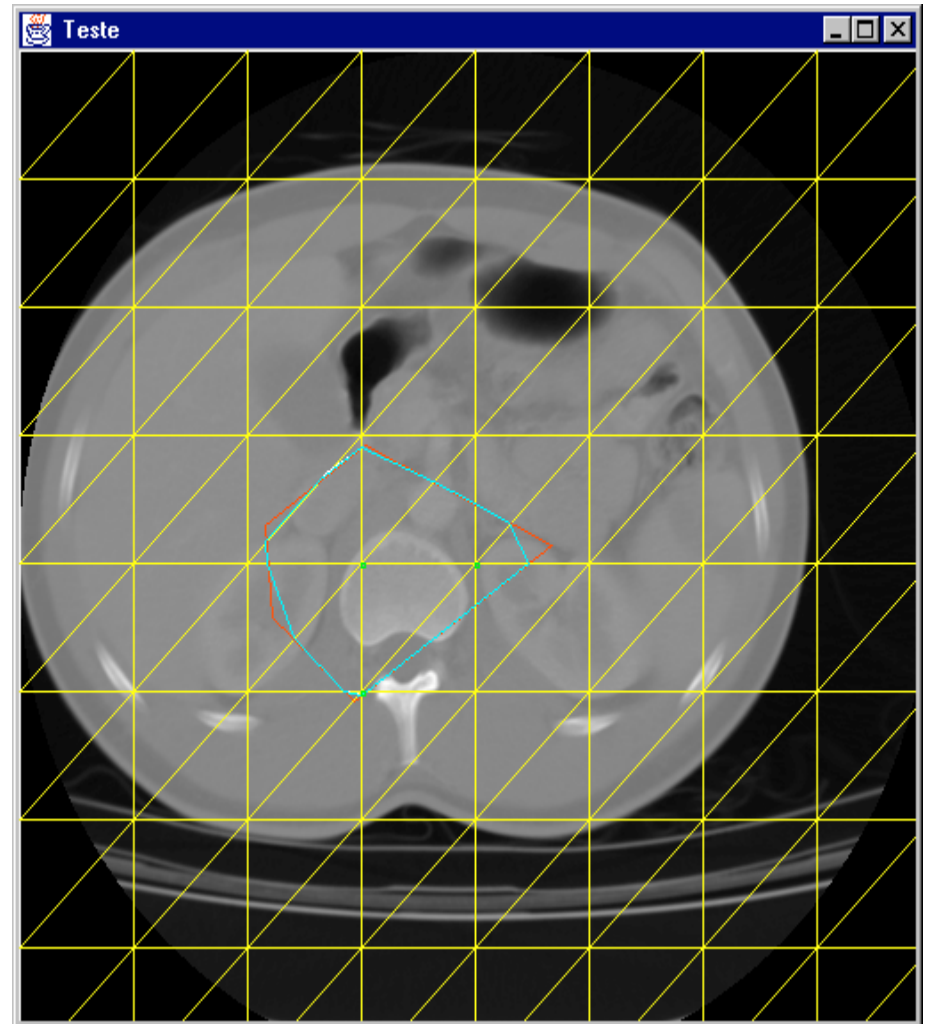
$$\frac{dX}{dt} + A.X = \vec{f}(X)$$

$$\frac{X^t - X^{t-1}}{\Delta t} + A.X^{t-1} = f(X^{t-1})$$

$$X^t = (I - \Delta t.A).X^{t-1} + \Delta t.f(X^{t-1})$$

# Topological Snakes

- ✓ Snakes sobre um grid do domínio
  - + simples que o FEM
  - + eficiente que o snakes
  - + incorpora o balloon



# T-snake

$$\frac{dc}{dt} - \frac{d}{ds} \left( w_1 \cdot \frac{dc}{ds} \right) + \frac{d^2}{ds^2} \left( w_2 \cdot \frac{d^2 c}{ds^2} \right) + \nabla P(c) = 0$$

$$k \cdot \frac{dc}{dt} - a \cdot f_{elástica} + b \cdot f_{flexao} = f_{balloon} + f_{externa}$$

$$c_i^{t+1} = c_i^t + (a \cdot f_{elástica} - b \cdot f_{flexao} + f_{balloon} + f_{externa}) \cdot \Delta t / k$$

$$f_{elástica} = \text{Laplaciano} (c_i^t)$$

$$f_{flexao} = \text{Laplaciano} (\text{Laplaciano} (c_i^t))$$



# resultados

