

# Interactive Segmentation of Multiple 3D Objects in Medical Images by Optimum Graph Cuts

Nikolas Moya  
Alexandre Xavier Falcão

Universidade de Campinas

12 de Março de 2015

- ▶ CAPES (2013)
- ▶ FAPESP (2013/17991-0)



# Agenda

Introdução

Contexto

Motivação

Objetivos

Metodologia

Image Foresting Transform

Inconsistência na DIFT

Relaxação

Inconsistência na relaxação

Experimentos

Configuração

Resultados

Considerações finais

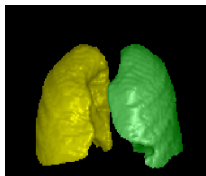
# Introdução

# Segmentação de imagens

- ▶ Segmentação é o processo de definir objetos relevantes em uma imagem: reconhecimento vs delineamento.
- ▶ Reconhecimento: tarefa de determinar a localização aproximada do objeto (humano  $\gg$  máquina).
- ▶ Delineamento: Definir a extensão espacial do objeto (máquina  $\gg$  humano).



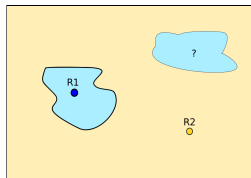
(a) Reconhecimento é feito pelo usuário



(b) Delineamento é feito pela máquina

# Segmentação de imagens

- ▶ Vasta literatura com métodos interativos, automáticos e híbridos.
- ▶ Estamos interessados em métodos interativos que propagam rótulos de objetos a partir de marcadores.
- ▶ Os rótulos são propagados através de voxels vizinhos de modo que ao final da propagação deve existir um caminho conexo entre cada elemento e o marcador correspondente.
- ▶ A segmentação é dita **inconsistente** quando esta conexidade não ocorre.



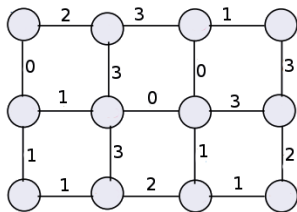
A região no canto superior direito está rotulada com azul, mas não há um caminho conexo de mesmo rótulo até seu marcador.

# Segmentação baseada em grafos

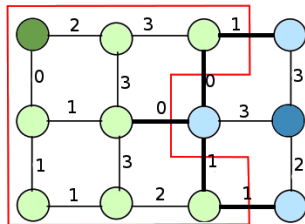
- ▶ Os métodos mais recentes usam de maneira implícita ou explícita o conceito de imagem como um grafo: os voxels são os nós e uma relação de adjacência define as arestas.
- ▶ A segmentação é um corte no grafo que separa marcadores internos e externos, minimizando um funcional de energia.

$$E_{sum}(\mathcal{C}) = \sum_{\forall(s,t) \in \mathcal{C}} \bar{w}^{\alpha}(s, t) \quad (1)$$

$$E_{max}(\mathcal{C}) = \lim_{\alpha \rightarrow +\infty} E^{\alpha}(\mathcal{C}) = \max_{\forall(s,t) \in \mathcal{C}} \bar{w}(s, t) \quad (2)$$



(a) Imagem como um grafo ponderado



(b) Corte ótimo no grafo

# Segmentação baseada em grafos

- ▶ A função de energia depende do algoritmo:  
**Min-cut/Max-flow** <sup>1</sup> minimiza  $E_{sum}$  e **Image Foresting Transform** (IFT) <sup>2</sup> minimiza  $E_{max}$ .
- ▶ Quais são as diferenças entre Min-cut/Max-flow e IFT?

	IFT	Min-cut/Max-flow
Tipo de energia:	$E_{max}$	$E_{sum}$
Múltiplo objetos:	Sim	Não
Complexidade de tempo:	$O( D  +  \mathcal{A} )$	$O( D  \mathcal{A} )$
Esforço do usuário:	Menos marcadores	Mais marcadores
Fronteiras:	Irregulares	Suaves

---

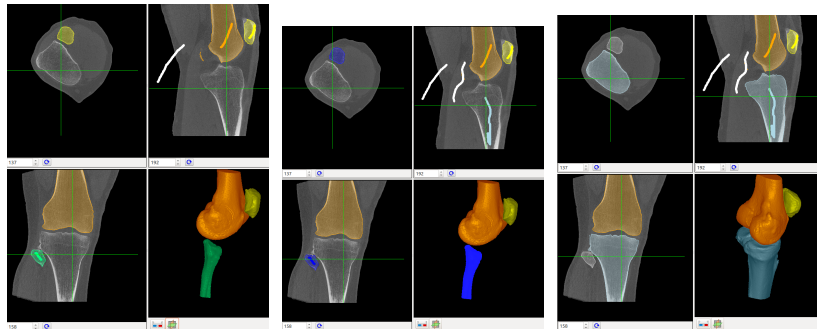
<sup>1</sup>(Ford & Fulkerson, 1962)

<sup>2</sup>(Falcão, Stolfi, & de Alencar Lotufo, 2004)



# Segmentação interativa

- ▶ Os métodos assumem que a segmentação pode ser resolvida com um único conjunto de sementes.
- ▶ O usuário não sabe a priori a posição ideal dos marcadores.
- ▶ A segmentação precisa ser re-executada diversas vezes para diferentes conjuntos de marcadores.



Mesmo que esta execução seja em tempo linear, a resposta não é interativa para imagens atuais cuja resolução é cada vez maior.

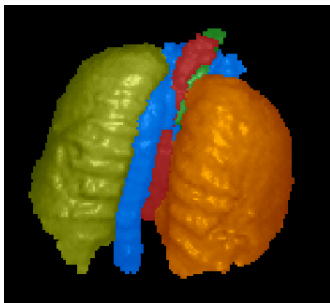
- ▶ Segmentar os objetos separadamente não é prático (Problema de sobreposição).
- ▶ O esforço do usuário durante o processo de segmentação deve ser reduzido.
- ▶ O tempo de resposta às intervenções do usuário deve ser interativo.
- ▶ O algoritmo **Differential Image Foresting Transform**<sup>3</sup> (DIFT) atende a estes requisitos.
- ▶ Porém, ele gera bordas irregulares que precisam ser suavizadas sem gerar inconsistência na segmentação.

---

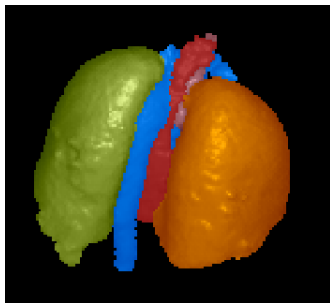
<sup>3</sup>(Falcão & Bergo, 2004)

# Motivação

- Visto que a energia do corte depende de uma única aresta (a de maior valor), vazamentos são mais comuns em  $E_{max}$  do que em  $E_{sum}$ .



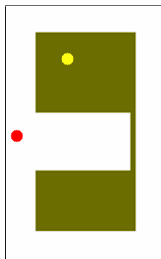
(a) Segmentação com DIFT sem suavização



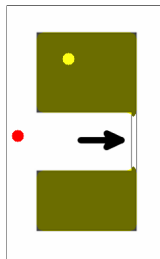
(b) Segmentação com DIFT com suavização

# Motivação

- Um filtro de difusão no mapa de rótulos (relaxação), pode produzir uma segmentação inconsistente.



(a)  
Segmentação  
sem filtro de  
difusão



(b)  
Segmentação  
relaxada

A relaxação pode gerar inconsistências na segmentação.

- ▶ Propor uma técnica de segmentação de múltiplos objetos baseados na DIFT com relaxação de borda sem gerar inconsistências.

# Metodologia

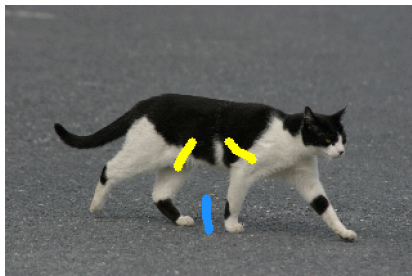
# Image Foresting Transform

- ▶ Uma ferramenta para desenvolver operadores em imagens baseado em conectividade ótima.
- ▶ Escolhemos a Transformada Watershed implementada com o algoritmo da DIFT.
- ▶ O grafo é direcionado e deve ter peso alto nas bordas e baixo nas demais regiões.
- ▶ O custo de um caminho é o maior valor ao longo do caminho.
- ▶ Um caminho com término  $t$  é ótimo se seu custo for menor do que todos os outros possíveis caminhos que terminam em  $t$ .
- ▶ Isto equivale a maximizar  $E_{min}$ .

$$E_{min}(\mathcal{C}) = \min_{\forall (s,t) \in \mathcal{C}} w(s, t) \quad (3)$$

# Propagação de sementes

- ▶ A ordem de propagação é monotonicamente crescente em relação ao valor da conectividade.

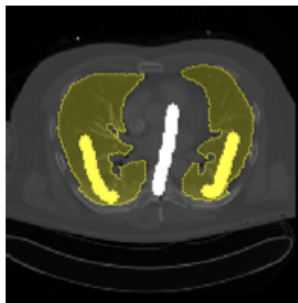


A propagação parte das sementes, visto que sementes são caminhos triviais de custo 0. [Mostrar GIF]



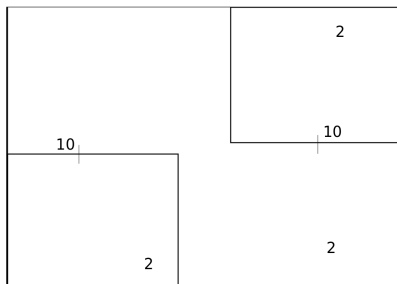
# Definição de objeto

- ▶ Um objeto pode ser definido por múltiplos componentes conexos representados por árvores de caminhos ótimos enraizados em sementes internas.
- ▶ Na atualização da floresta, é propagado o rótulo dos marcadores (id), para que a remoção de um marcador não destrua todos os componentes conexos de mesmo rótulo.

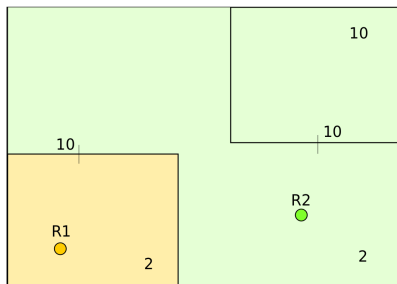


Exemplo de um único objeto formado por dois componentes conexos.

# Image Foresting Transform

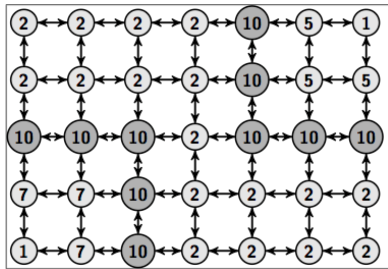


(a) Imagem de gradiente

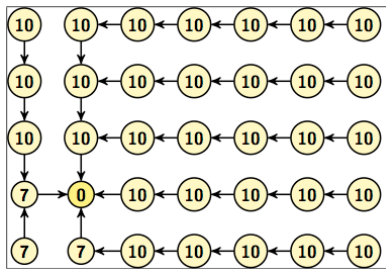


(b) Mapa de custos e rótulos após a execução

# Differential IFT

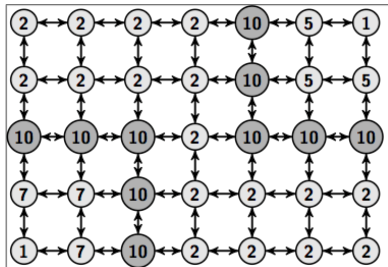


(a) Imagem de gradiente

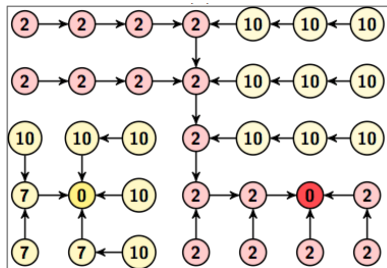


(b) Floresta após a execução com um único marcador

# Differential IFT



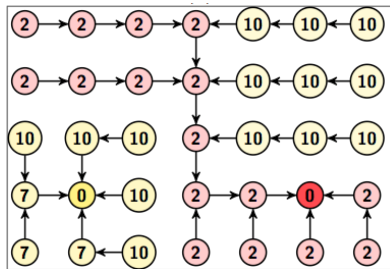
(a) Imagem de gradiente



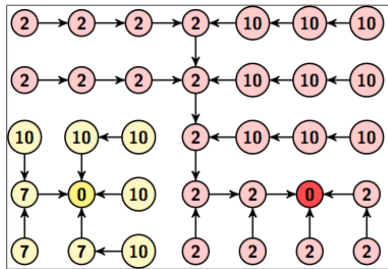
(b) Floresta incorreta após a adição do segundo marcador.

A região no canto superior direito é uma *tie-zone*.

# Differential IFT



(a) Floresta incorreta após a adição do segundo marcador

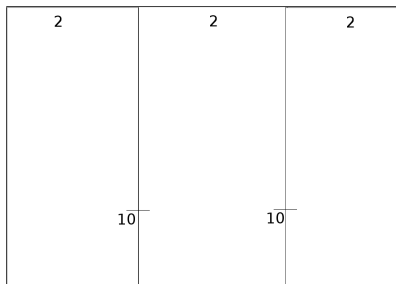


(b) Floresta esperada devido ao teste do predecessor

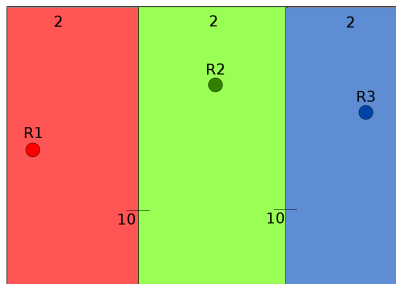
Este problema foi considerado na formulação original da DIFT. Visto que os custos da floresta estão corretos, um teste do predecessor foi adicionado para corrigir inconsistências de rótulos.

# Differential IFT

- Como funciona a remoção de marcadores?

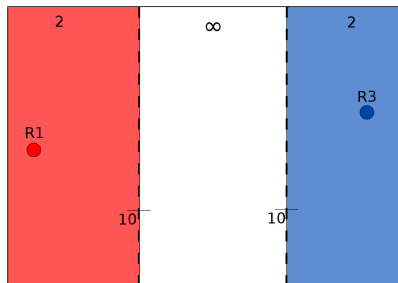


(a) Imagem de gradiente



(b) Segmentação inicial

# Differential IFT



(a) Remoção do marcador verde



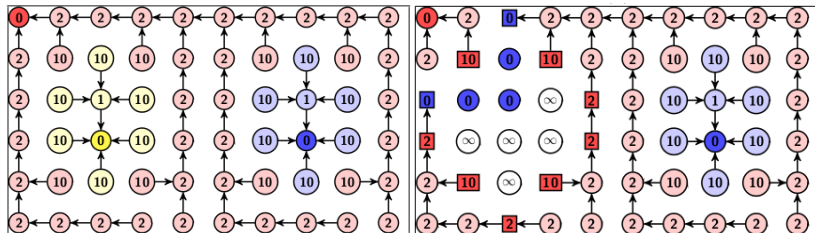
(b) Resultado após a propagação das sementes de fronteira

Quando um marcador é removido, a zona de influência daquele marcador é atribuída custo infinito e os pixels adjacentes a esta região competem entre si.

# Inconsistência na DIFT



# Inconsistência na DIFT

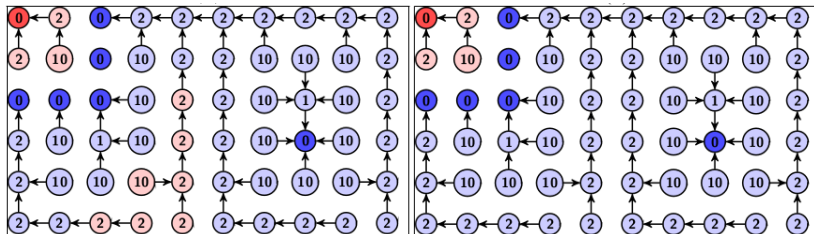


(a) Segmentação inicial de múltiplos objetos

(b) Remoção do marcador amarelo e adição do marcador azul

Exemplo de inconsistência na formulação original da DIFT com adição e remoção de marcadores simultaneamente.

# Inconsistência na DIFT



(a) Inconsistência de nós rotulados como vermelho

(b) Resultado obtido após a nossa correção.

Os nós vermelhos não foram conquistados pelos nós azuis pois eram sementes de fronteira. Um teste impedia que nós já retirados da fila fossem reinseridos.

# Relaxação

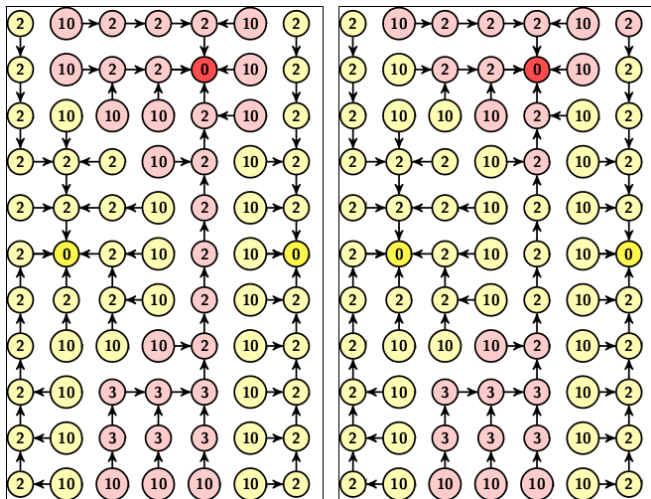
- ▶ Processo iterativo de filtragem por difusão <sup>4</sup>.
- ▶ É executado apenas na fronteira do mapa de rótulos.
- ▶ A partir da imagem de gradiente normalizada, é gerada uma imagem de pesos e em seguida uma imagem de coeficientes.
- ▶ Para cada pixel da fronteira, é calculado um grau de pertinência ponderado entre seus adjacentes.
- ▶ O rótulo com maior pertinência é o novo rótulo do pixel.
- ▶ Considera apenas o mapa de rótulos, portanto este processo pode causar inconsistências na floresta.

---

<sup>4</sup>(Malmberg, Nyström, Mehnert, Engstrom, & Bengtsson, 2010)

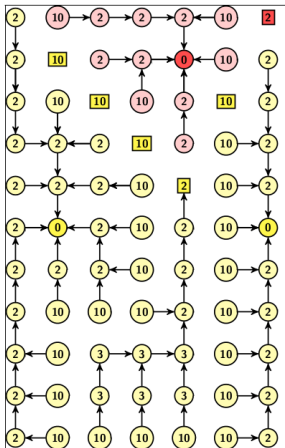
# Inconsistência na relaxação

# Inconsistência na relaxação

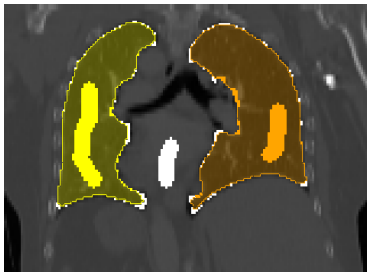


Uma relaxação no mapa de rótulos pode fazer com que um objeto fino desconecte-se.

# Corrigindo a inconsistência



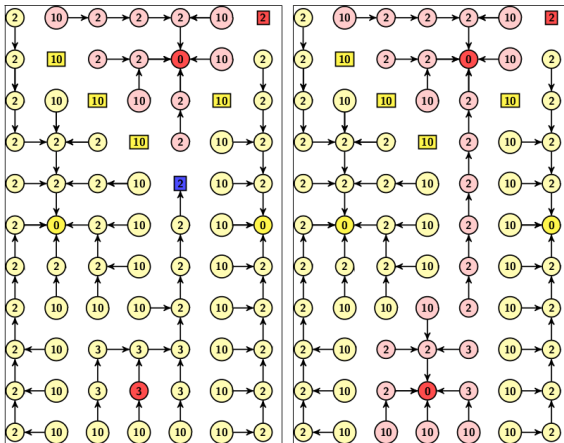
(a) Segmentação sem relaxação



(b) Segmentação com relaxação

Sementes de relaxação foram adicionadas na floresta para garantir conexidade sem alterar a otimalidade da floresta.

# Desfazendo a relaxação



Marcadores do usuário tem maior prioridade do que sementes de relaxação



# Experimentos

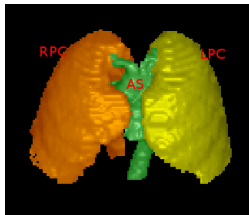
- ▶ **DIFT**: Differential Image Foresting Transform
- ▶ **DIFT-TR**: Differential Image Foresting Transform with Terminal Relaxation.
- ▶ **DRIFT**: Differential and Relaxed Image Foresting Transform
- ▶ **MODGC**: Multiple Objects Dynamic Graph Cut
- ▶ Em MODGC, executamos o algoritmo Dynamic Graph Cut <sup>5</sup> nos objetos separadamente e mesclamos em um mapa de rótulo único.
- ▶ Desempate: o rótulo de um voxel é o maior rótulo entre as imagens de rótulos de cada objeto.

---

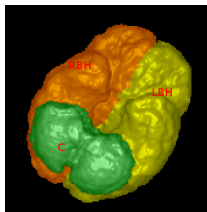
<sup>5</sup>(Kohli & Torr, 2007)

# Base da dados

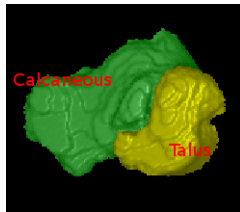
- ▶ As bases foram avaliadas nos quatro algoritmos: DIFT, DIFT-TR, DRIFT, and MODGC.
- ▶ Três bases de dados de múltiplos objetos.
- ▶ Acurácia e eficiência.



(a) Base tórax com 36 imagens



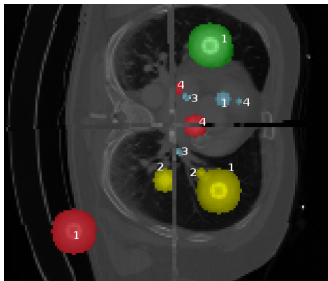
(b) Base cérebro com 40 imagens



(c) Base pé com 20 imagens

# Robô geodésico

- ▶ Insere marcadores no centro geodésico dos componentes de erro.
- ▶ Evita custos e vícios de usuários especialistas.
- ▶ Diversos parametros para configurar seu comportamento.



Uma possível ordem de inserção de marcadores utilizando o robô geodésico.

# Average Symmetric Surface Distance

- ▶ Distância entre duas superfícies (segmentação e *ground truth*)
- ▶ A distância entre as superfícies multiplicado pela dimensão do voxel converte o erro em milímetros.
- ▶ Uma segmentação perfeita tem distância 0.

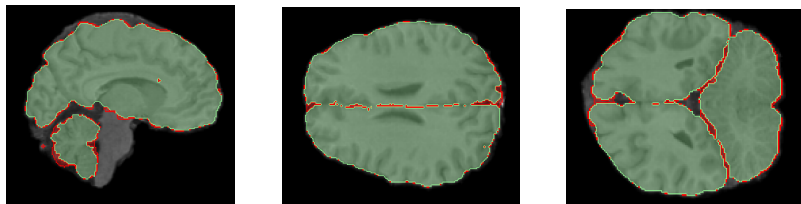


Imagem do cérebro com ASSD de 0.65. Voxels rotulados corretamente são apresentados em verde e os incorretos em vermelho.

# Objetos individuais vs Múltiplos objetos

- ▶ Experimento comparando a segmentação dos objetos separadamente e simultaneamente usando o algoritmo da DIFT na base do tórax.
- ▶ Segmentar simultaneamente os objetos é mais preciso e rápido, visto que a complexidade da DIFT é independente do número de objetos.

DIFT Objeto (mm)				DIFT Múltiplo Objetos (mm)
AS	RPC	LPC	Média	Média
8.73	1.57	1.94	4.08	1.29 $\pm$ 0.20

Comparação entre segmentação de objetos individuais e múltiplos objetos na base tórax.

## Base tórax - Acurácia e Eficiência

	Média	AS	RPC	LPC
DIFT	0.67 $\pm$ 0.06	1.09 $\pm$ 0.1	0.46 $\pm$ 0.08	0.46 $\pm$ 0.06
DIFT-TR	<b>0.60 <math>\pm</math> 0.05</b>	0.93 $\pm$ 0.1	0.44 $\pm$ 0.07	0.43 $\pm$ 0.05
DRIFT	<b>0.63 <math>\pm</math> 0.13</b>	1.08 $\pm$ 0.3	0.41 $\pm$ 0.06	0.40 $\pm$ 0.03
MODGC	0.80 $\pm$ 0.05	1.19 $\pm$ 0.1	0.62 $\pm$ 0.06	0.59 $\pm$ 0.04

Erro ASSD em milímetros para a base tórax.

	Num marcadores	Num execuções	Tempo de Resposta (s)
DIFT	70 $\pm$ 12	16 $\pm$ 4	<b>0.40 <math>\pm</math> 0.08</b>
DIFT-TR	70 $\pm$ 12	16 $\pm$ 4	<b>0.49 <math>\pm</math> 0.10</b>
DRIFT	<b>57 <math>\pm</math> 12</b>	17 $\pm$ 4	<b>0.51 <math>\pm</math> 0.11</b>
MODGC	88 $\pm$ 16	14 $\pm$ 3	7.08 $\pm$ 1.69

Número médio de marcadores, correções e tempo médio de espera entre cada correção do usuário na base do tórax.

## Base cérebro - Acurácia e Eficiência

	Média	C	LBH	RBH
DIFT	$0.73 \pm 0.05$	$0.91 \pm 0.1$	$0.64 \pm 0.07$	$0.64 \pm 0.06$
DIFT-TR	<b><math>0.63 \pm 0.04</math></b>	$0.83 \pm 0.1$	$0.53 \pm 0.07$	$0.53 \pm 0.05$
DRIFT	$0.73 \pm 0.06$	$0.90 \pm 0.1$	$0.64 \pm 0.07$	$0.64 \pm 0.06$
MODGC	$0.85 \pm 0.07$	$0.91 \pm 0.1$	$0.75 \pm 0.18$	$0.89 \pm 0.08$

Erro ASSD em milímetros para a base cérebro.

	Num marcadores	Num execuções	Tempo de Resposta (s)
DIFT	$335 \pm 79$	$26 \pm 8$	<b><math>0.17 \pm 0.04</math></b>
DIFT-TR	$335 \pm 79$	$26 \pm 8$	<b><math>0.22 \pm 1.25</math></b>
DRIFT	<b><math>296 \pm 84</math></b>	$24 \pm 7$	$3.08 \pm 2.03$
MODGC	$332 \pm 47$	$28 \pm 6$	$15.91 \pm 7.29$

Número médio de marcadores, correções e tempo médio de espera entre cada correção do usuário na base do cérebro.



## Base pé - Acurácia e Eficiência

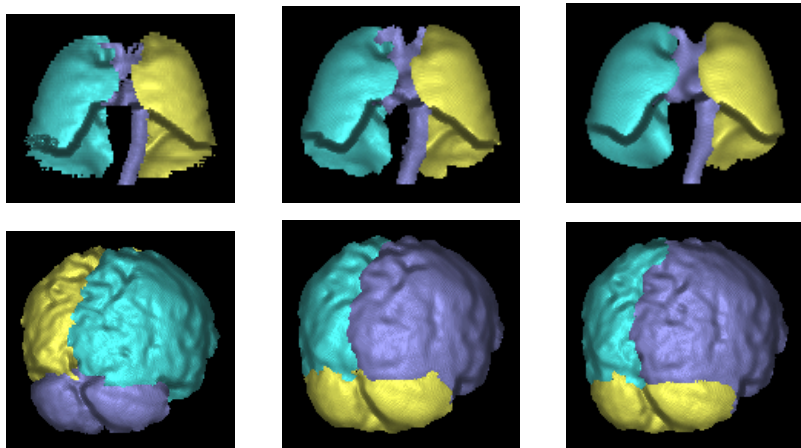
	Média	Calcaneous	Talus
DIFT	1.30 $\pm$ 0.10	1.44 $\pm$ 0.12	1.17 $\pm$ 0.11
DIFT-TR	<b>0.90 <math>\pm</math> 0.06</b>	0.96 $\pm$ 0.08	0.85 $\pm$ 0.06
DRIFT	1.05 $\pm$ 0.09	1.17 $\pm$ 0.12	0.93 $\pm$ 0.10
MODGC	1.52 $\pm$ 0.20	1.99 $\pm$ 0.41	1.05 $\pm$ 0.06

Erro ASSD em milímetros para a base pé.

	Num marcadores	Num execuções	Tempo de Resposta (s)
DIFT	393 $\pm$ 144	45 $\pm$ 4	<b>0.5 <math>\pm</math> 0.1</b>
DIFT-TR	393 $\pm$ 144	45 $\pm$ 4	1.8 $\pm$ 0.2
DRIFT	<b>243 <math>\pm</math> 116</b>	47 $\pm$ 3	2.4 $\pm$ 0.4
MODGC	436 $\pm$ 17	55 $\pm$ 4	19.5 $\pm$ 6.6

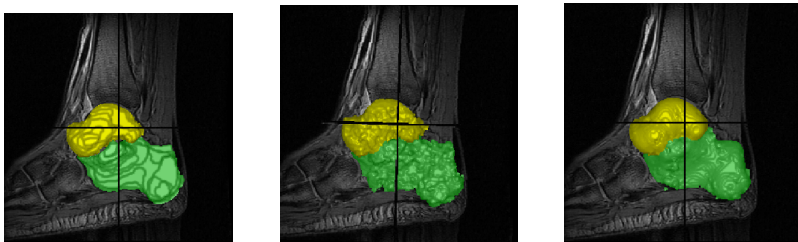
Número médio de marcadores, correções e tempo médio de espera entre cada correção do usuário na base do pé.

# Performance Visual



Comparação entre GT, DIFT e DRIFT em imagens do tórax e cérebro.

# Performance Visual



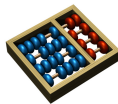
Comparação entre GT, DIFT e DIFT-TR em uma imagem da base do pé.

- ▶ Segmentação de múltiplos objetos é mais vantajosa do que segmentar os objetos separadamente.
- ▶ O algoritmo DRIFT requer menos esforço do usuário sendo levemente menos acurado.
- ▶ DIFT-TR foi mais acurado ou equivalente em relação ao DRIFT.
- ▶ Resultado visual muito bom em imagens ruidosas, como por exemplo a base do pé com o algoritmo DIFT-TR.

# Contribuições

- ▶ DRIFT é um novo algoritmo de segmentação de múltiplos objetos baseados na DIFT com relaxação do mapa de rótulos sem gerar inconsistências.
- ▶ Correção da inconsistência da DIFT na adição e remoção de marcadores simultâneos.
- ▶ Correção de inconsistências causados no mapa de rótulos durante a relaxação.
- ▶ Experimentos com usuários reais foi deixado para trabalhos futuros.
- ▶ Outro trabalho futuro pode usar o DRIFT para suavizar ou corrigir segmentações a partir do mapa de rótulos, desde que este mapa de rótulo seja transformado em uma floresta de caminhos ótimos.

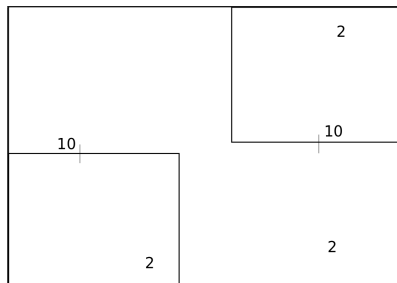
# Perguntas?



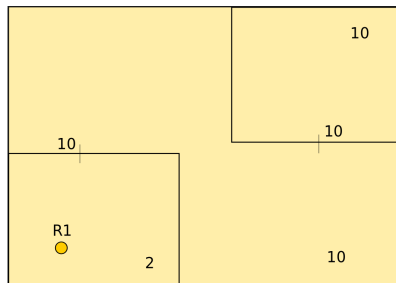
# References

- Falcão, A., & Bergo, F. (2004). Interactive volume segmentation with differential image foresting transforms. *IEEE Transactions on Medical Imaging*, 23(9), 1100-1108.
- Falcão, A., Stolfi, J., & de Alencar Lotufo, R. (2004). The image foresting transform: theory, algorithms, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1), 19-29.
- Ford, L., & Fulkerson, D. (1962). Flows in networks.
- Kohli, P., & Torr, P. H. S. (2007). Dynamic graph cuts for efficient inference in markov random fields. *IEEE Transactions Pattern Analysis Machine Intelligence*, 29(12), 2079-2088.
- Malmberg, F., Nyström, I., Mehnert, A., Engstrom, C., & Bengtsson, E. (2010). Relaxed image foresting transforms for interactive volume image segmentation. In *Proceedings of spie on medical imaging* (Vol. 7623).

# Differential IFT



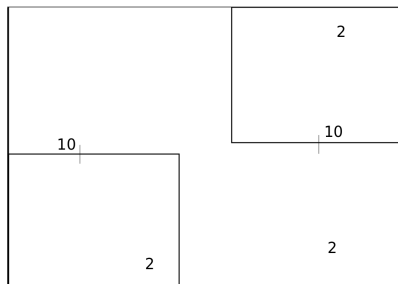
(a)



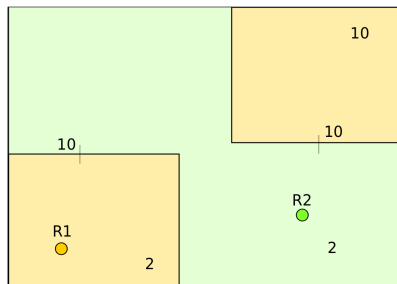
(b)



# Differential IFT



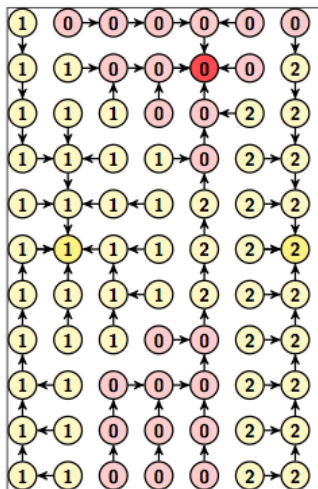
(c)



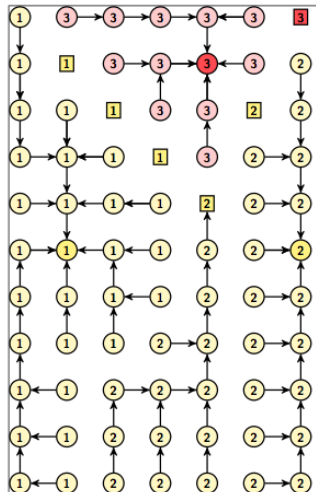
(d)

The upper right region is called a tie-zone, since both seeds offer the same path cost.

# Differential IFT



(a)



(b)

The upper right region is called a tie-zone, since both seeds offer the

## The IFT-watershed algorithm

Algorithm 2 presents a Watershed Transform from Labeled Markers using the IFT algorithm. This formulation was first presented in [63].

### Algorithm 2 – IFT-watershed

Input: Gradient image  $\hat{G} = (D, G)$ , adjacency relation  $\mathcal{A}$ , seed set  $S \subset D$  labeled by  $\lambda$ .  
 Output: Optimum path forest  $P$ , connectivity  $C$ , root  $R$ , and object label  $L_O$  maps.  
 Auxiliary: Priority queue  $Q$  and a variable  $tmp$ .

```

1. For each  $t \in D$ , do
2.    $P(t) \leftarrow nil, R(t) \leftarrow t, C(t) \leftarrow +\infty$ 
3.   If  $t \in S$ , then
4.      $C(t) \leftarrow 0, L_O(t) \leftarrow \lambda(t)$ 
5.    $Insert\ t\ in\ Q$ 
6. While  $Q \neq \emptyset$ , do
7.   Remove  $s$  from  $Q$  such that  $C(s)$  is minimum
8.   For each  $t \in \mathcal{A}(s)$ , such that  $C(t) > C(s)$ , do
9.      $tmp \leftarrow \max\{C(s), G(t)\}$ .
10.    If  $tmp < C(t)$ , then
11.      If  $C(t) \neq +\infty$ , then remove  $t$  from  $Q$ 
12.       $Set\ P(t) \leftarrow s, R(t) \leftarrow R(s), C(t) \leftarrow tmp, L_O(t) \leftarrow L_O(s)$ 
13.     $Insert\ t\ in\ Q$ 
14. Return  $(P, C, R, L_O)$ 
  
```

# Relaxação do mapa de rótulos

```
1. For each  $s \in \mathcal{P}$ , do
2.   For each  $t \in \mathcal{A}(s)$ , do
3.     If  $\hat{F}.LO(s) \neq \hat{F}.LO(t)$ 
4.        $\mathcal{B}_p \leftarrow \mathcal{B}_p \cup \{s, t\}$ 
5.    $L_{O_p} \leftarrow \hat{F}.LO, L_{O_n} \leftarrow \hat{F}.LO, L_{M_p} \leftarrow \hat{F}.LM, L_{M_n} \leftarrow \hat{F}.LM$ 
6.   For each relaxation iteration from 1 to  $\Omega.T$ , do
7.      $\mathcal{B}_n \leftarrow \emptyset$ 
8.     For each  $s$  in  $\mathcal{B}_p$ , do
9.        $\mathcal{B}_n \leftarrow \mathcal{B}_n \cup \{s\}$ 
10.      For each  $l \in \{0, 1, \dots, n\}$ , do
11.         $mk(l) \leftarrow -1$  and  $\mu(l) \leftarrow 0$ 
12.      For each  $t \in \mathcal{A}(s)$ , do
13.         $Set\ l \leftarrow L_{O_p}(t), \mu(l) \leftarrow \mu(l) + \Omega.W(t)M_p(t)$ 
14.         $mk(l) \leftarrow L_{M_p}(t)$ , and  $\mathcal{B}_n \leftarrow \mathcal{B}_n \cup \{t\}$ 
15.      For each  $l \in \{0, 1, \dots, n\}$ , do
16.         $\mu(l) \leftarrow \mu(l) / \Omega.N_f(s)$ 
17.       $Compute\ l_{\max}\ such\ that\ \mu(l_{\max}) = \max_{l=0,1,\dots,n} \{\mu(l)\}.$ 
18.       $L_{O_n}(s) \leftarrow l_{\max}, M_n(s) \leftarrow \mu(l_{\max}),\ and\ L_{M_n}(s) \leftarrow mk(l_{\max})$ 
19.      While  $\mathcal{B}_p \neq \emptyset$ , do
20.         $Remove\ s\ from\ \mathcal{B}_p$ 
21.         $Set\ L_{O_p}(s) \leftarrow L_{O_n}(s), L_{M_p}(s) \leftarrow L_{M_n}(s),\ and\ M_p(s) \leftarrow M_n(s).$ 
22.       $\mathcal{B}_p \leftarrow \mathcal{B}_n$ 
23.    $L'_O \leftarrow L_{O_n}, L'_M \leftarrow L_{M_n}, \mathcal{B}_d \leftarrow \mathcal{B}_n$ 
24.   Return  $(L'_O, L'_M, \mathcal{B}_d)$ 
```

## Algoritmo

– LABELCORRECTION

1. **Enquanto**  $\mathcal{B}_d \neq \emptyset$
2.     Remove  $t$  from  $\mathcal{B}_d$
3.     **Se**  $L'_O(t) \neq \hat{F}.L_O(t)$  **Então**
4.          $color \leftarrow white$
5.          $(r, color) \leftarrow SearchRelaxationRoot(\hat{F}, L'_O, t, color)$
6.         **Se**  $color = black$  **Então**
7.              $PropagateRootAttributes(\hat{F}, \mathcal{A}, L'_O, L'_M, r)$
8.      $\hat{F}.P(r) \leftarrow nil$

# Encontrar a raiz de relaxação

## Algoritmo

– SEARCHRELAXATIONROOT

1. **Se**  $\hat{F}.P(t) = nil$ , **Então**
2.      $color \leftarrow gray$
3.     **Se**  $L'_O(t) \neq \hat{F}.L_O(t)$ , **Então**
4.          $r \leftarrow t$ ,  $color \leftarrow black$
5. **Senão**
6.      $(t, color) \leftarrow SearchRelaxationRoot(\hat{F}, L'_O, \hat{F}.P(t), color)$
7.     **Se**  $color = gray$ , **Então**
8.         **Se**  $L'_O(t) \neq L'_O(\hat{F}.P(t))$  **Então**
9.              $r \leftarrow t$ ,  $color \leftarrow black$
10. **Retorne**  $(r, color)$

## Algoritmo

– PROPAGATEROOTATTRIBUTES

1.  $\mathcal{T} \leftarrow \{r\}$
2. **Enquanto**  $\mathcal{T} \neq \emptyset$
3.     Remove  $s$  from  $\mathcal{T}$
4.      $\hat{F}.L_O(s) \leftarrow L'_O(r), L'_O(s) \leftarrow L'_O(r), \hat{F}.L_M(s) \leftarrow L'_M(r)$
5.      $L'_M(s) \leftarrow L'_M(r), \hat{F}.R(s) \leftarrow r$
6.     **Para Cada**  $t \in \mathcal{A}(s)$ , **Faça**
7.         **Se**  $P(t) = s$  **Então**
8.              $\mathcal{T} \leftarrow \mathcal{T} \cup \{t\}$