

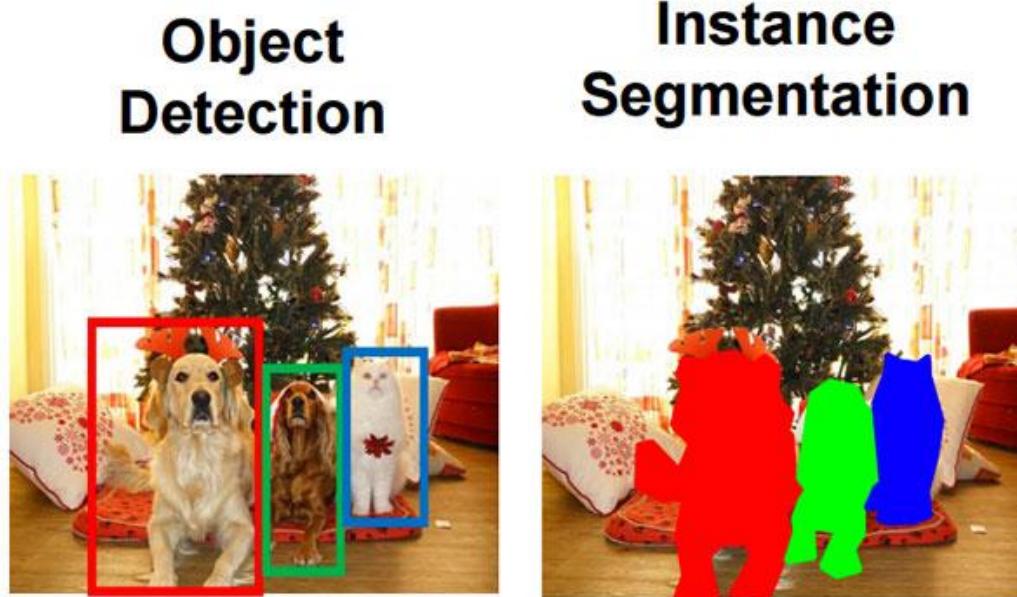
Segmentação de imagem

Componentes Ligados
KMeans

Transformada de Watershed
Gradient Path Labelling

Deteção vs Segmentação

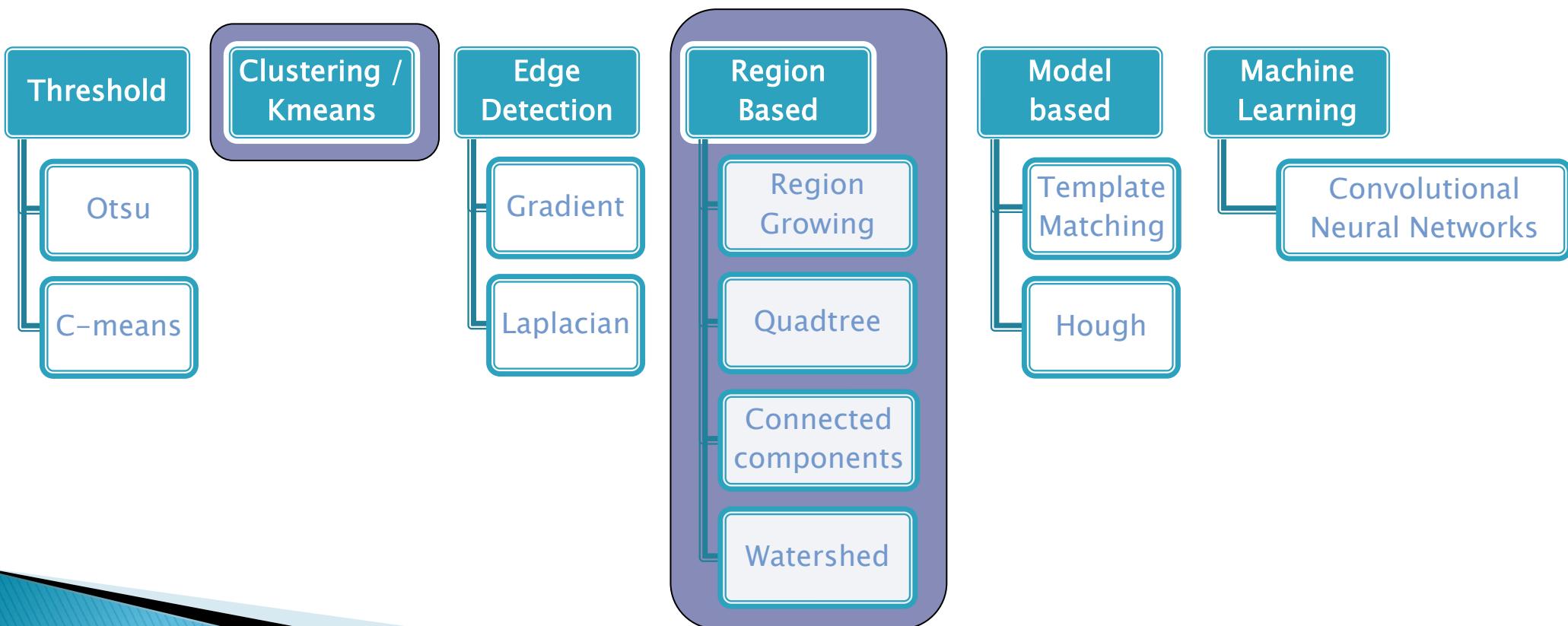
- ▶ A deteção dá uma caixa de delimitação contendo o objeto, enquanto a segmentação dá uma máscara dos pixels de cada objeto



Source : cs231n.stanford.edu

Segmentação

- ▶ É o processo de dividir a imagem em múltiplos segmentos com significado e mais fáceis de analisar

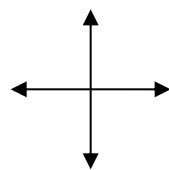


COMPONENTES LIGADOS

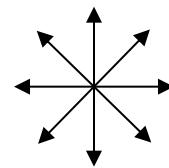


Algoritmos de componentes ligados

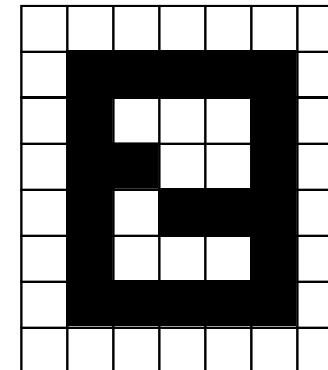
- ▶ São muito utilizados em imagens de objectos sobre fundos contrastantes.
- ▶ Todos os algoritmos etiquetam o primeiro píxel da imagem e tentam propagar a etiqueta pelos píxeis à sua direita ou abaixo dele.
- ▶ Podem ser usadas adjacências a 4 ou a 8 pixels:



Adjacência a
4 pixels

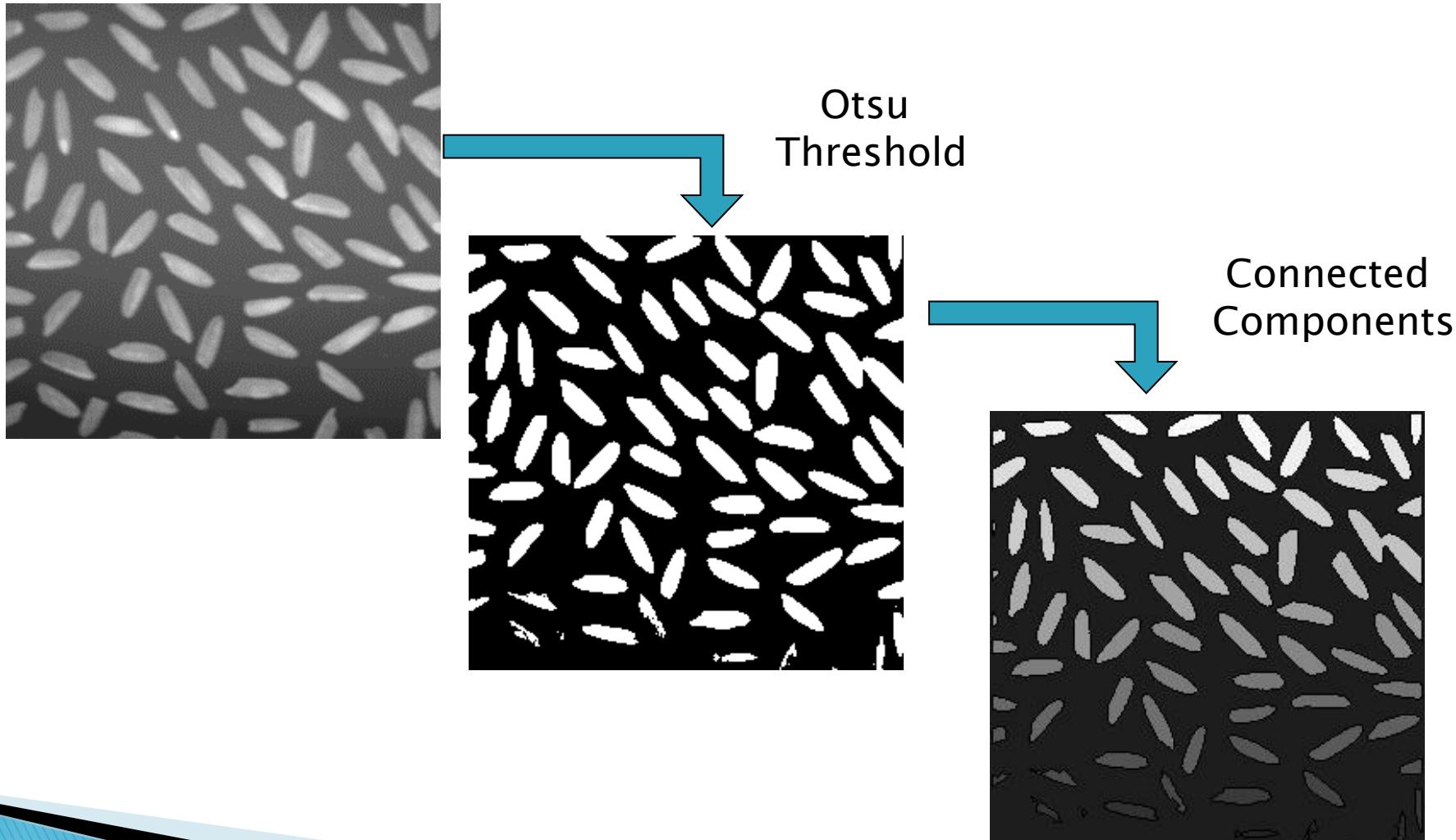


Adjacência a
8 pixels

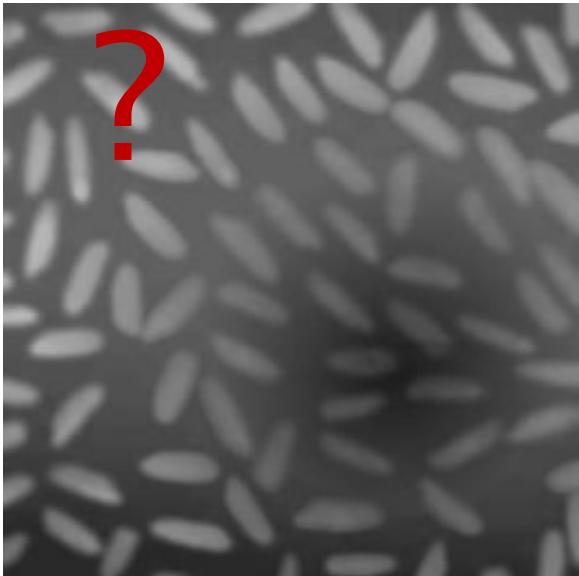


Quantos objectos?
Quantos buracos?

Componentes Ligados - II

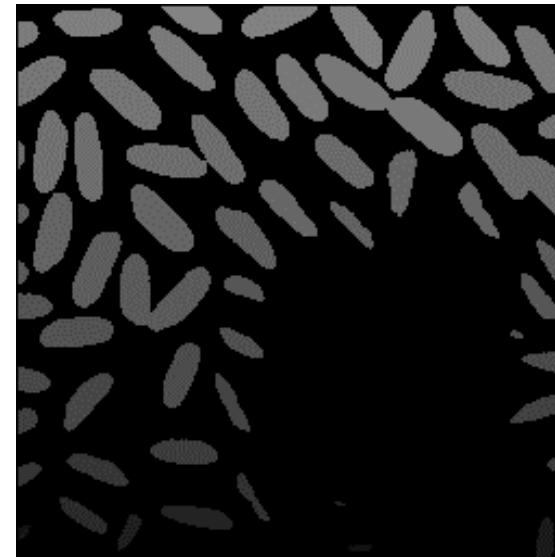


Componentes Ligados - III



e com iluminação não uniforme?

Otsu
Threshold Connected
Components



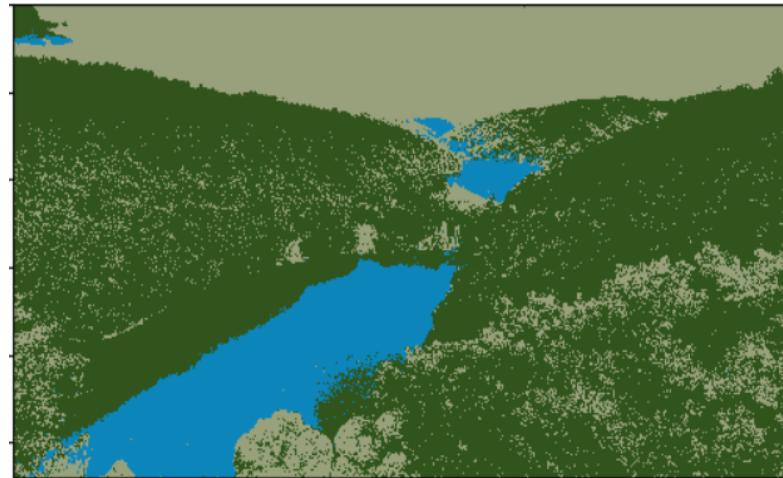
K Means Clustering

K-Means Clustering

- ▶ **Objetivo:** Reduzir o número de cores para um máximo de K
- ▶ Os pixels são atribuídos a um dos clusters K com base na sua semelhança de cor (distância)
- ▶ Processo iterativo que termina quando uma configuração estável é alcançada



Original



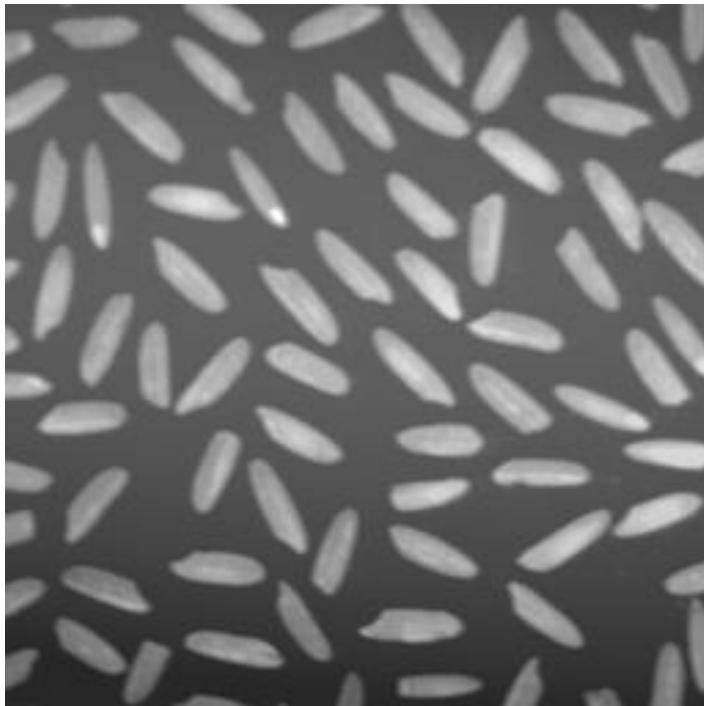
k-means with $k = 3$

Algoritmo

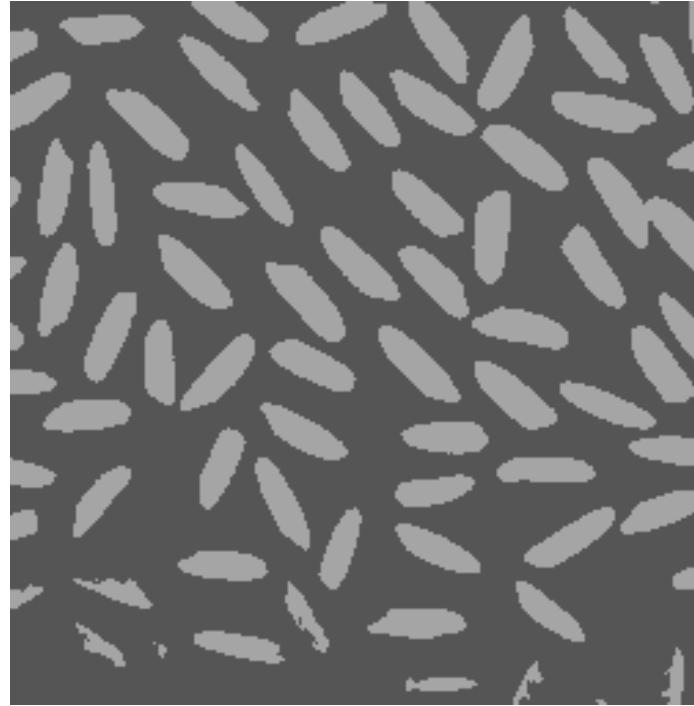
1. Escolha K centros de cluster (cores), aleatoriamente ou baseado em algum método heurístico;
2. Atribua cada pixel ao cluster que minimiza a distância entre o pixel e o centro do cluster – semelhança de cor
3. Re-cálculo dos centros de cluster, com uma média de todos os pixels no cluster
4. Repita os passos 2 e 3 até que a convergência seja alcançada (isto é, nenhum pixel mude de cluster)

Source:wikipedia

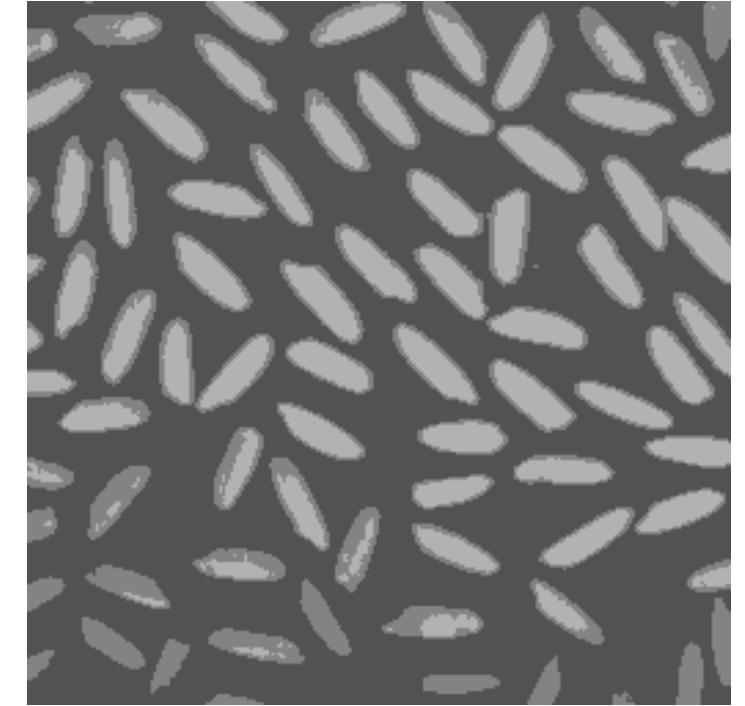
Exemplos



Original

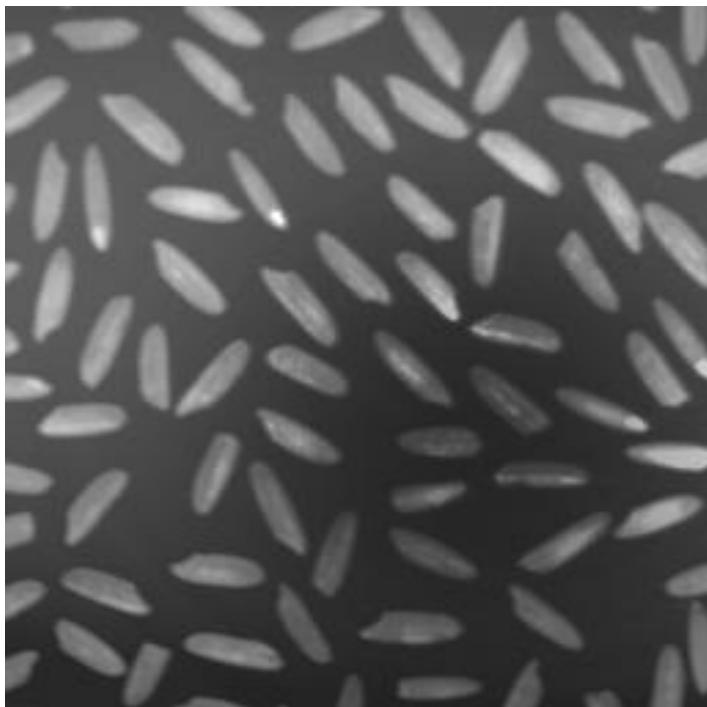


k -means com $k = 2$

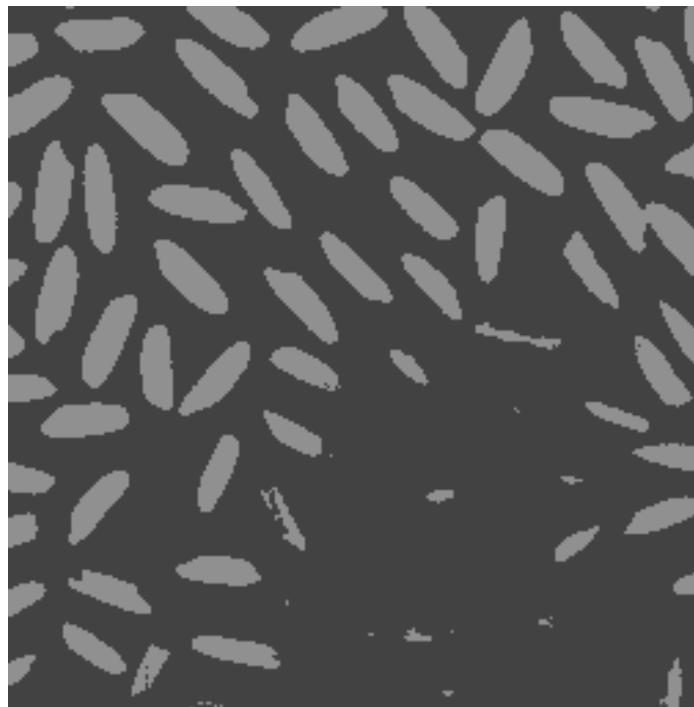


k -means com $k = 3$

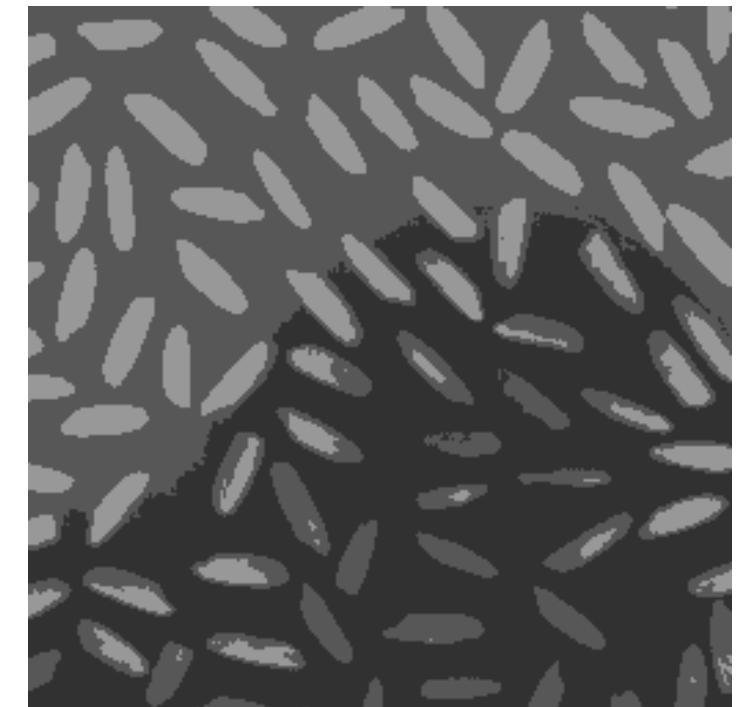
Exemplos



Original



k -means com $k = 2$

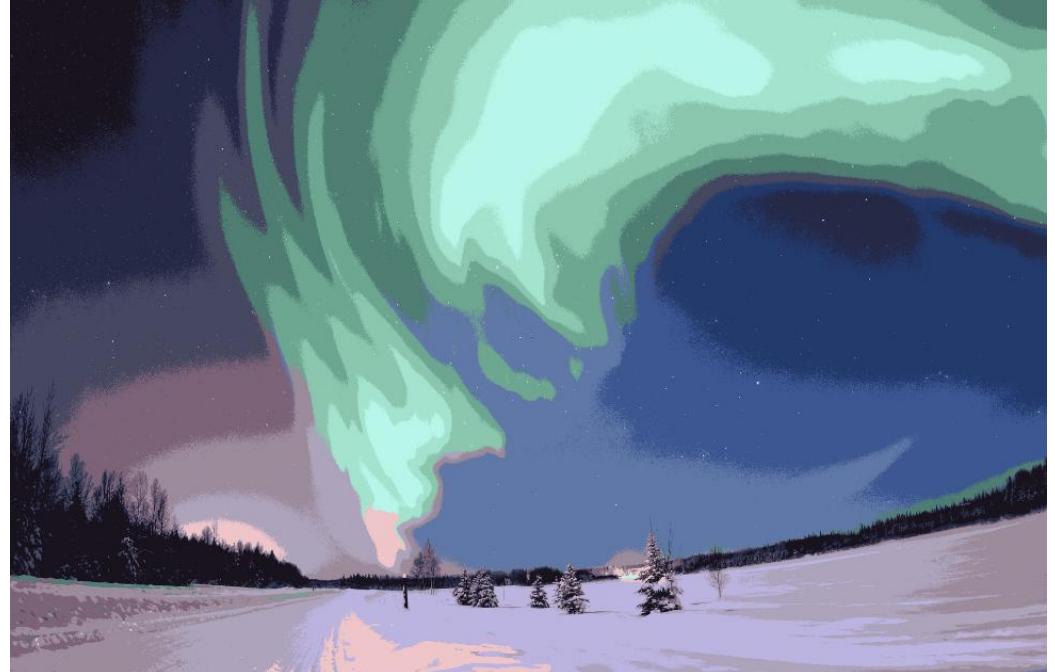


k -means com $k = 3$

Exemplos



Original



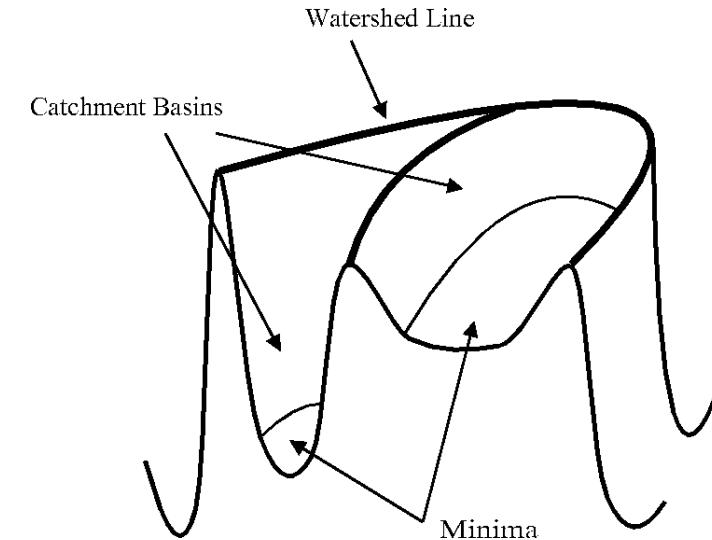
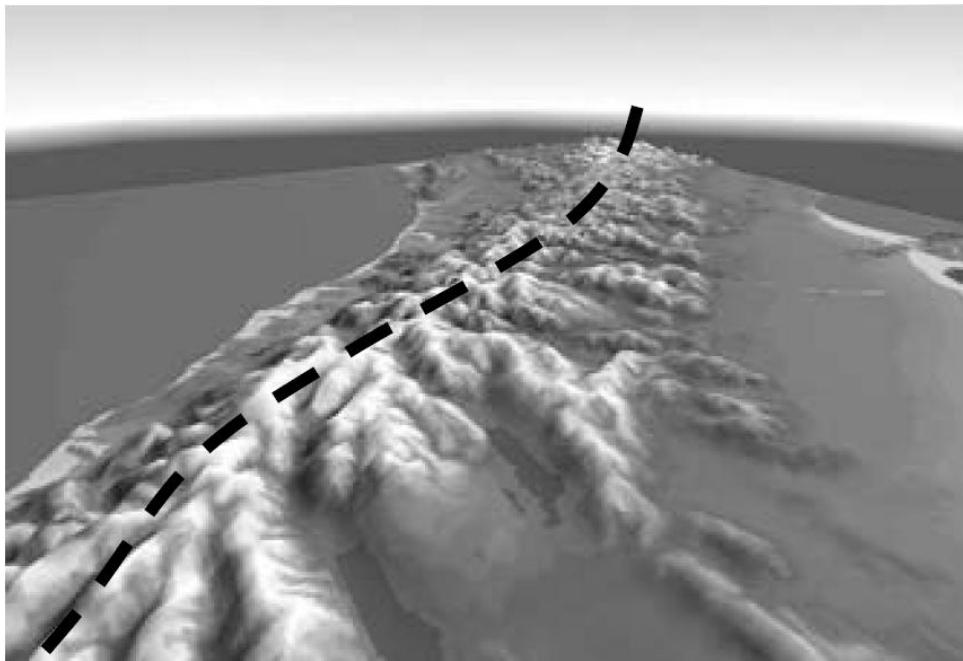
k -means com $k = 16$

TRANSFORMADA DE WATERSHED



Watershed – a ideia base

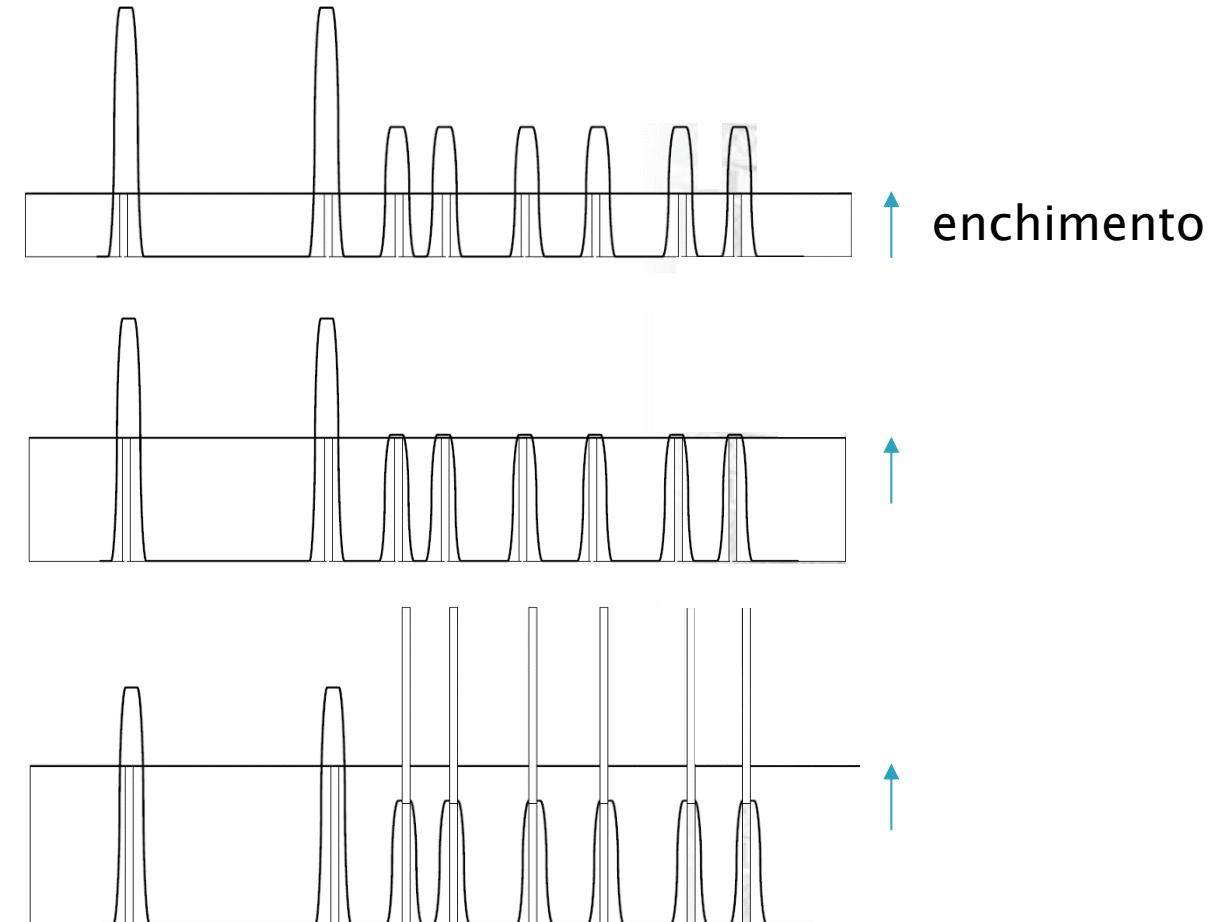
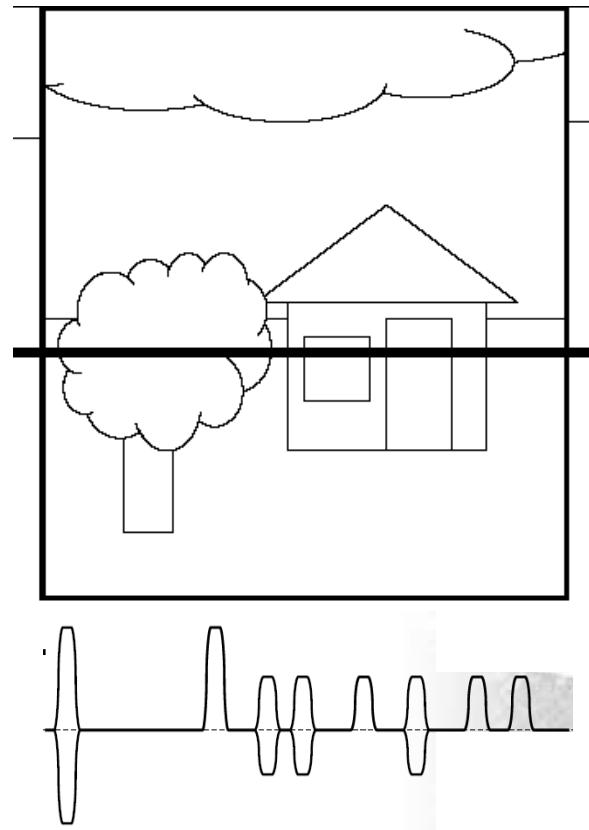
- ▶ Watershed = Bacia Hidrográfica



Segmentação baseada em arestas

- ▶ **Objectivo:**
 - Encontrar áreas de padrão uniforme
 - A fronteira entre áreas deve corresponder a alterações nas propriedades das imagens
- ▶ Os conceitos de base do algoritmo são:
 - Arestas (**watersheds**) – são zonas de grande gradiente que representam descontinuidades na intensidade da imagem
 - Regiões (**catchment basins**) – são áreas sem arestas, limitadas por arestas

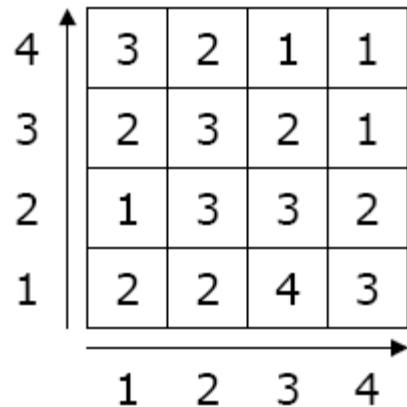
Como funciona...



Algoritmo básico da watershed

- ▶ Algoritmo:
 1. Ordenar os pixéis de acordo com a sua amplitude e por ordem Top-left -> bottom-right
 2. Para cada pixel:
 - Se os seus vizinhos não tiverem etiqueta dar-lhe uma nova etiqueta
 - Se tiver vizinhos com uma única etiqueta recebe essa etiqueta
 - Se tiver vizinhos com várias etiquetas é uma watershed
- ▶ Esta versão básica tem alguns problemas
 - É muito sensível ao ruído podendo gerar demasiadas regiões muito pequenas (sobre-segmentação)
 - É computacional exigente: requer uma lista de pixéis ordenada

Exemplo (1/16)



Algoritmo:

Ordenar os pixels de acordo com a sua amplitude

Para cada pixel:

Se os seus vizinhos não tiverem etiqueta dar-lhe uma nova etiqueta

Se tiver vizinhos com uma única etiqueta recebe essa etiqueta

Se tiver vizinhos com várias etiquetas é uma watershed

$(3,4) = 1$

$(4,4) = 1$

$(4,3) = 1$

$(1,2) = 1$

$(2,4) = 2$

$(1,3) = 2$

$(3,3) = 2$

$(4,2) = 2$

$(1,1) = 2$

$(2,1) = 2$

$(1,4) = 3$

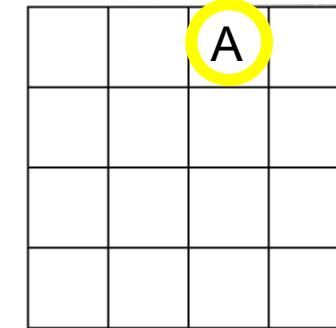
$(2,3) = 3$

$(2,2) = 3$

$(3,2) = 3$

$(4,1) = 3$

$(3,1) = 4$



Etiquetas

Ordenação

Exemplo (2/16)

	4		
	3	2	1
3	2	3	2
2	1	3	3
1	2	2	4
	1	2	3

$(3,4) = 1$
 $(4,4) = 1$
 $(4,3) = 1$
 $(1,2) = 1$
 $(2,4) = 2$
 $(1,3) = 2$
 $(3,3) = 2$
 $(4,2) = 2$
 $(1,1) = 2$
 $(2,1) = 2$
 $(1,4) = 3$
 $(2,3) = 3$
 $(2,2) = 3$
 $(3,2) = 3$
 $(4,1) = 3$
 $(3,1) = 4$

Ordenação

Algoritmo:

Ordenar os pixels de acordo com a sua amplitude

Para cada pixel:

Se os seus vizinhos não tiverem etiqueta dar-lhe uma nova etiqueta

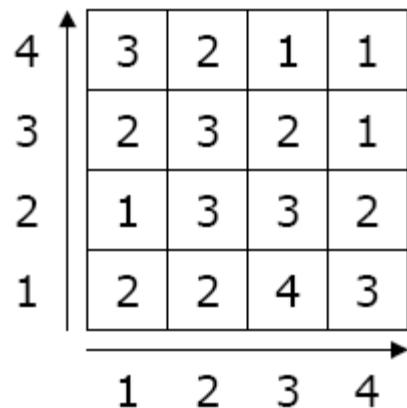
Se tiver vizinhos com uma única etiqueta recebe essa etiqueta

Se tiver vizinhos com várias etiquetas é uma watershed

		A	A

Etiquetas

Exemplo (3/16)



$(3,4) = 1$
 $(4,4) = 1$
 $(4,3) = 1$
 $(1,2) = 1$
 $(2,4) = 2$
 $(1,3) = 2$
 $(3,3) = 2$
 $(4,2) = 2$
 $(1,1) = 2$
 $(2,1) = 2$
 $(1,4) = 3$
 $(2,3) = 3$
 $(2,2) = 3$
 $(3,2) = 3$
 $(4,1) = 3$
 $(3,1) = 4$

Ordenação

Algoritmo:

Ordenar os pixels de acordo com a sua amplitude

Para cada pixel:

Se os seus vizinhos não tiverem etiqueta dar-lhe uma nova etiqueta

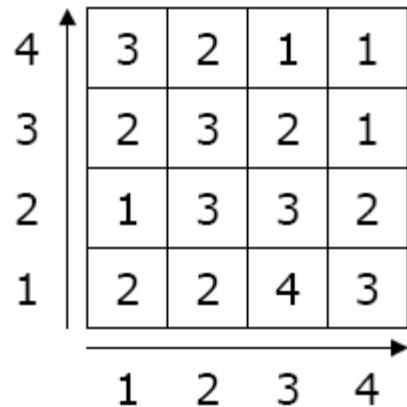
Se tiver vizinhos com uma única etiqueta recebe essa etiqueta

Se tiver vizinhos com várias etiquetas é uma watershed

		A	A
			A

Etiquetas

Exemplo (4/16)



Algoritmo:
Ordenar os pixels de acordo com a sua amplitude
Para cada pixel:
Se os seus vizinhos não tiverem etiqueta dar-lhe uma nova etiqueta
Se tiver vizinhos com uma única etiqueta recebe essa etiqueta
Se tiver vizinhos com várias etiquetas é uma watershed

$(3,4) = 1$
 $(4,4) = 1$
 $(4,3) = 1$
 $(1,2) = 1$
 $(2,4) = 2$
 $(1,3) = 2$
 $(3,3) = 2$
 $(4,2) = 2$
 $(1,1) = 2$
 $(2,1) = 2$
 $(1,4) = 3$
 $(2,3) = 3$
 $(2,2) = 3$
 $(3,2) = 3$
 $(4,1) = 3$
 $(3,1) = 4$

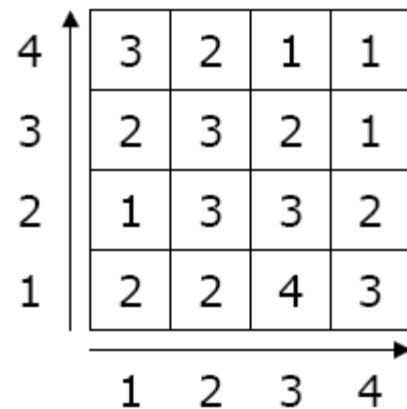
Ordenação

A 4x4 grid showing the resulting watershed regions. The regions are labeled as follows:

		A	A
			A
B			

Etiquetas

Exemplo (5/16)



Algoritmo:
Ordenar os pixels de acordo com a sua amplitude
Para cada pixel:
Se os seus vizinhos não tiverem etiqueta dar-lhe uma nova etiqueta
Se tiver vizinhos com uma única etiqueta recebe essa etiqueta
Se tiver vizinhos com várias etiquetas é uma watershed

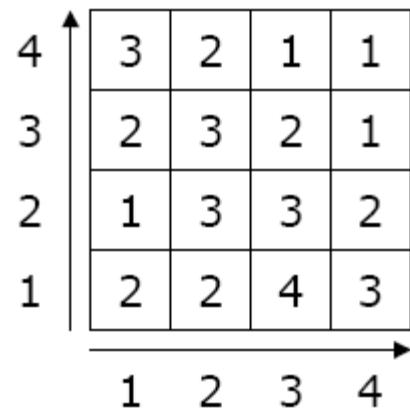
$(3,4) = 1$
 $(4,4) = 1$
 $(4,3) = 1$
 $(1,2) = 1$
 $(2,4) = 2$
 $(1,3) = 2$
 $(3,3) = 2$
 $(4,2) = 2$
 $(1,1) = 2$
 $(2,1) = 2$
 $(1,4) = 3$
 $(2,3) = 3$
 $(2,2) = 3$
 $(3,2) = 3$
 $(4,1) = 3$
 $(3,1) = 4$

Ordenação

	A	A	A
			A
B			

Etiquetas

Exemplo (6/16)



Ordenação

$(3,4) = 1$
 $(4,4) = 1$
 $(4,3) = 1$
 $(1,2) = 1$
 $(2,4) = 2$
 $(1,3) = 2$ (highlighted)
 $(3,3) = 2$
 $(4,2) = 2$
 $(1,1) = 2$
 $(2,1) = 2$
 $(1,4) = 3$
 $(2,3) = 3$
 $(2,2) = 3$
 $(3,2) = 3$
 $(4,1) = 3$
 $(3,1) = 4$

A 4x4 grid illustrating the watershed regions. The regions are labeled as follows:

	A	A	A
B			A

The region 'B' is highlighted with a yellow circle.

Etiquetas

Exemplo (7/16)

	4			
	3	2	1	1
	2	3	2	1
	1	3	3	2
	2	2	4	3
		1	2	3

$(3,4) = 1$
 $(4,4) = 1$
 $(4,3) = 1$
 $(1,2) = 1$
 $(2,4) = 2$
 $(1,3) = 2$
 $(3,3) = 2$
 $(4,2) = 2$
 $(1,1) = 2$
 $(2,1) = 2$
 $(1,4) = 3$
 $(2,3) = 3$
 $(2,2) = 3$
 $(3,2) = 3$
 $(4,1) = 3$
 $(3,1) = 4$

Ordenação

Algoritmo:

Ordenar os pixels de acordo com a sua amplitude

Para cada pixel:

Se os seus vizinhos não tiverem etiqueta dar-lhe uma nova etiqueta

Se tiver vizinhos com uma única etiqueta recebe essa etiqueta

Se tiver vizinhos com várias etiquetas é uma watershed

	A	A	A
B		A	A
B			

Etiquetas

Exemplo (8/16)

	4		
	3		
	2		
	1		
4	3	2	1
3	2	3	2
2	1	3	3
1	2	2	4
	1	2	3

$(3,4) = 1$
 $(4,4) = 1$
 $(4,3) = 1$
 $(1,2) = 1$
 $(2,4) = 2$
 $(1,3) = 2$
 $(3,3) = 2$
 $(4,2) = 2$
 $(1,1) = 2$
 $(2,1) = 2$
 $(1,4) = 3$
 $(2,3) = 3$
 $(2,2) = 3$
 $(3,2) = 3$
 $(4,1) = 3$
 $(3,1) = 4$

Ordenação

Algoritmo:

Ordenar os pixels de acordo com a sua amplitude

Para cada pixel:

Se os seus vizinhos não tiverem etiqueta dar-lhe uma nova etiqueta

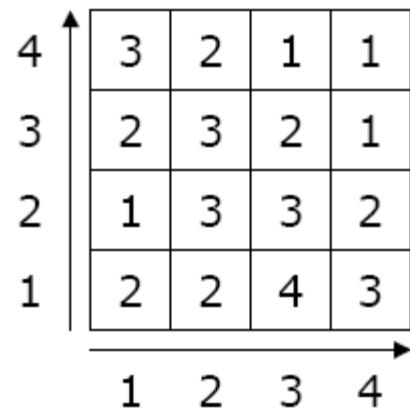
Se tiver vizinhos com uma única etiqueta recebe essa etiqueta

Se tiver vizinhos com várias etiquetas é uma watershed

	A	A	A
B		A	A
B			A

Etiquetas

Exemplo (9/16)



Algoritmo:
Ordenar os pixels de acordo com a sua amplitude
Para cada pixel:
Se os seus vizinhos não tiverem etiqueta dar-lhe uma nova etiqueta
Se tiver vizinhos com uma única etiqueta recebe essa etiqueta
Se tiver vizinhos com várias etiquetas é uma watershed

$(3,4) = 1$
 $(4,4) = 1$
 $(4,3) = 1$
 $(1,2) = 1$
 $(2,4) = 2$
 $(1,3) = 2$
 $(3,3) = 2$
 $(4,2) = 2$
 $(1,1) = 2$
 $(2,1) = 2$
 $(1,4) = 3$
 $(2,3) = 3$
 $(2,2) = 3$
 $(3,2) = 3$
 $(4,1) = 3$
 $(3,1) = 4$

Ordenação

	A	A	A
B		A	A
B			A
B			

Etiquetas

Exemplo (10/16)

	4		
	3		
	2		
	1		
4	3	2	1
3	2	3	2
2	1	3	3
1	2	2	4
	1	2	3

Algoritmo:
Ordenar os pixels de acordo com a sua amplitude
Para cada pixel:
Se os seus vizinhos não tiverem etiqueta dar-lhe uma nova etiqueta
Se tiver vizinhos com uma única etiqueta recebe essa etiqueta
Se tiver vizinhos com várias etiquetas é uma watershed

$(3,4) = 1$
 $(4,4) = 1$
 $(4,3) = 1$
 $(1,2) = 1$
 $(2,4) = 2$
 $(1,3) = 2$
 $(3,3) = 2$
 $(4,2) = 2$
 $(1,1) = 2$
 $(2,1) = 2$
 $(1,4) = 3$
 $(2,3) = 3$
 $(2,2) = 3$
 $(3,2) = 3$
 $(4,1) = 3$
 $(3,1) = 4$

Ordenação

	A	A	A
B		A	A
B			A
B	B		

Etiquetas

Exemplo (11/16)

	4		
	3		
	2		
	1		
4	3	2	1
3	2	3	2
2	1	3	3
1	2	2	4
	1	2	3

$(3,4) = 1$
 $(4,4) = 1$
 $(4,3) = 1$
 $(1,2) = 1$
 $(2,4) = 2$
 $(1,3) = 2$
 $(3,3) = 2$
 $(4,2) = 2$
 $(1,1) = 2$
 $(2,1) = 2$
 $(1,4) = 3$
 $(2,3) = 3$
 $(2,2) = 3$
 $(3,2) = 3$
 $(4,1) = 3$
 $(3,1) = 4$

Ordenação

Algoritmo:

Ordenar os pixels de acordo com a sua amplitude

Para cada pixel:

Se os seus vizinhos não tiverem etiqueta dar-lhe uma nova etiqueta

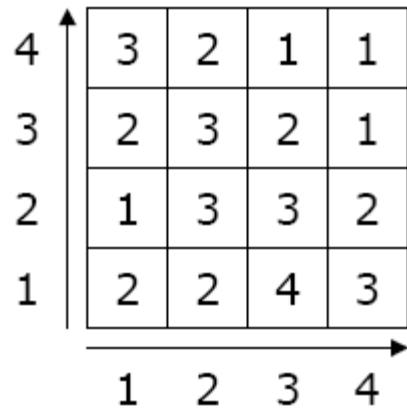
Se tiver vizinhos com uma única etiqueta recebe essa etiqueta

Se tiver vizinhos com várias etiquetas é uma watershed

*	A	A	A
B		A	A
B			A
B	B		

Etiquetas

Exemplo (13/16)



$(3,4) = 1$
 $(4,4) = 1$
 $(4,3) = 1$
 $(1,2) = 1$
 $(2,4) = 2$
 $(1,3) = 2$
 $(3,3) = 2$
 $(4,2) = 2$
 $(1,1) = 2$
 $(2,1) = 2$
 $(1,4) = 3$
 $(2,3) = 3$
 $(2,2) = 3$
 $(3,2) = 3$
 $(4,1) = 3$
 $(3,1) = 4$

Ordenação

Algoritmo:

Ordenar os pixels de acordo com a sua amplitude

Para cada pixel:

Se os seus vizinhos não tiverem etiqueta dar-lhe uma nova etiqueta

Se tiver vizinhos com uma única etiqueta recebe essa etiqueta

Se tiver vizinhos com várias etiquetas é uma watershed

*	A	A	A
B	*	A	A
B			A
B	B		

Etiquetas

Exemplo (12/16)

	4		
	3		
	2		
	1		
4	3	2	1
3	2	3	2
2	1	3	3
1	2	2	4
	1	2	3

$(3,4) = 1$
 $(4,4) = 1$
 $(4,3) = 1$
 $(1,2) = 1$
 $(2,4) = 2$
 $(1,3) = 2$
 $(3,3) = 2$
 $(4,2) = 2$
 $(1,1) = 2$
 $(2,1) = 2$
 $(1,4) = 3$
 $(2,3) = 3$
 $(2,2) = 3$
 $(3,2) = 3$
 $(4,1) = 3$
 $(3,1) = 4$

Ordenação

Algoritmo:

Ordenar os pixels de acordo com a sua amplitude

Para cada pixel:

Se os seus vizinhos não tiverem etiqueta dar-lhe uma nova etiqueta

Se tiver vizinhos com uma única etiqueta recebe essa etiqueta

Se tiver vizinhos com várias etiquetas é uma watershed

*	A	A	A
B	*	A	A
B	B		A
B	B		

Etiquetas

Exemplo (14/16)

	4		
	3		
	2		
	1		
4	3	2	1
3	2	3	2
2	1	3	3
1	2	2	4
	1	2	3

$(3,4) = 1$
 $(4,4) = 1$
 $(4,3) = 1$
 $(1,2) = 1$
 $(2,4) = 2$
 $(1,3) = 2$
 $(3,3) = 2$
 $(4,2) = 2$
 $(1,1) = 2$
 $(2,1) = 2$
 $(1,4) = 3$
 $(2,3) = 3$
 $(2,2) = 3$
 $(3,2) = 3$
 $(4,1) = 3$
 $(3,1) = 4$

Ordenação

Algoritmo:

Ordenar os pixels de acordo com a sua amplitude

Para cada pixel:

Se os seus vizinhos não tiverem etiqueta dar-lhe uma nova etiqueta

Se tiver vizinhos com uma única etiqueta recebe essa etiqueta

Se tiver vizinhos com várias etiquetas é uma watershed

*	A	A	A
B	*	A	A
B	B	*	A
B	B		

Etiquetas

Exemplo (15/16)

	4		
	3		
	2		
	1		
4	3	2	1
3	2	3	2
2	1	3	3
1	2	2	4
	1	2	3

$(3,4) = 1$
 $(4,4) = 1$
 $(4,3) = 1$
 $(1,2) = 1$
 $(2,4) = 2$
 $(1,3) = 2$
 $(3,3) = 2$
 $(4,2) = 2$
 $(1,1) = 2$
 $(2,1) = 2$
 $(1,4) = 3$
 $(2,3) = 3$
 $(2,2) = 3$
 $(3,2) = 3$
 $(4,1) = 3$
 $(3,1) = 4$

Ordenação

Algoritmo:

Ordenar os pixels de acordo com a sua amplitude

Para cada pixel:

Se os seus vizinhos não tiverem etiqueta dar-lhe uma nova etiqueta

Se tiver vizinhos com uma única etiqueta recebe essa etiqueta

Se tiver vizinhos com várias etiquetas é uma watershed

*	A	A	A
B	*	A	A
B	B	*	A
B	B		A

Etiquetas

Exemplo (16/16)

	4		
	3		
	2		
	1		
4	3	2	1
3	2	3	2
2	1	3	3
1	2	2	4
	1	2	3

$(3,4) = 1$
 $(4,4) = 1$
 $(4,3) = 1$
 $(1,2) = 1$
 $(2,4) = 2$
 $(1,3) = 2$
 $(3,3) = 2$
 $(4,2) = 2$
 $(1,1) = 2$
 $(2,1) = 2$
 $(1,4) = 3$
 $(2,3) = 3$
 $(2,2) = 3$
 $(3,2) = 3$
 $(4,1) = 3$
 $(3,1) = 4$

Ordenação

*	A	A	A
B	*	A	A
B	B	*	A
B	B	*	A

Etiquetas

Algoritmo:
Ordenar os pixels de acordo com a sua amplitude
Para cada pixel:
Se os seus vizinhos não tiverem etiqueta dar-lhe uma nova etiqueta
Se tiver vizinhos com uma única etiqueta recebe essa etiqueta
Se tiver vizinhos com várias etiquetas é uma watershed

Exemplo de watershed

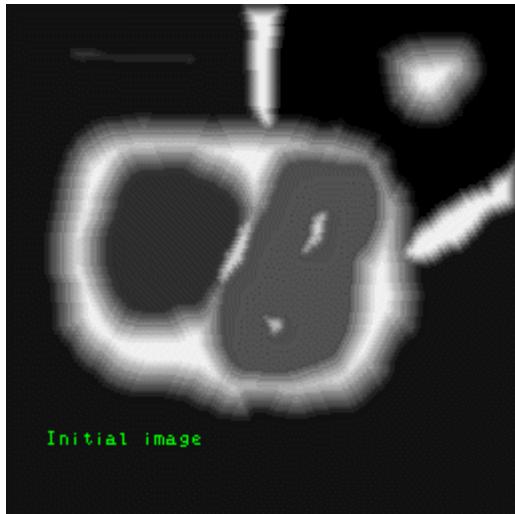
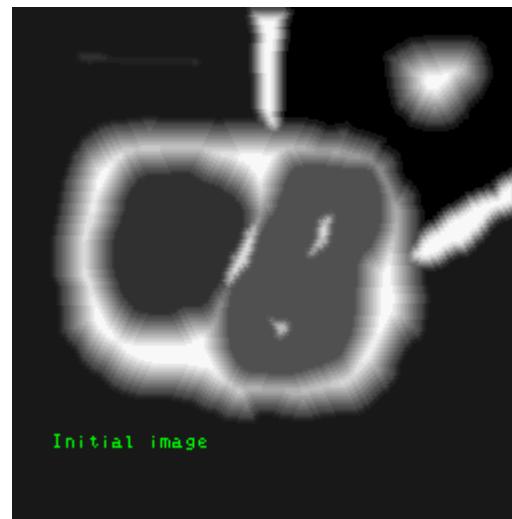


Imagen inicial

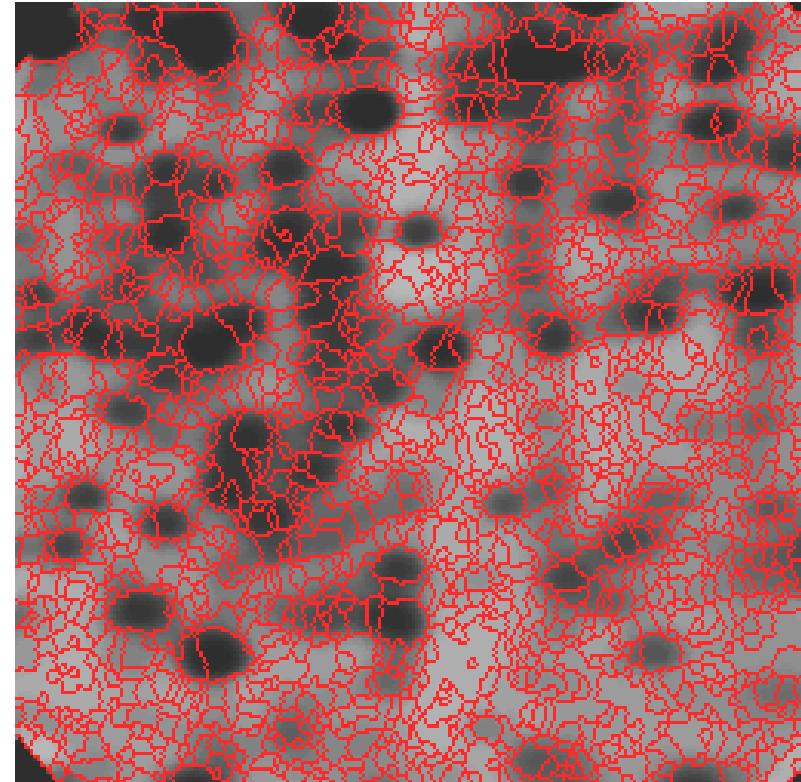
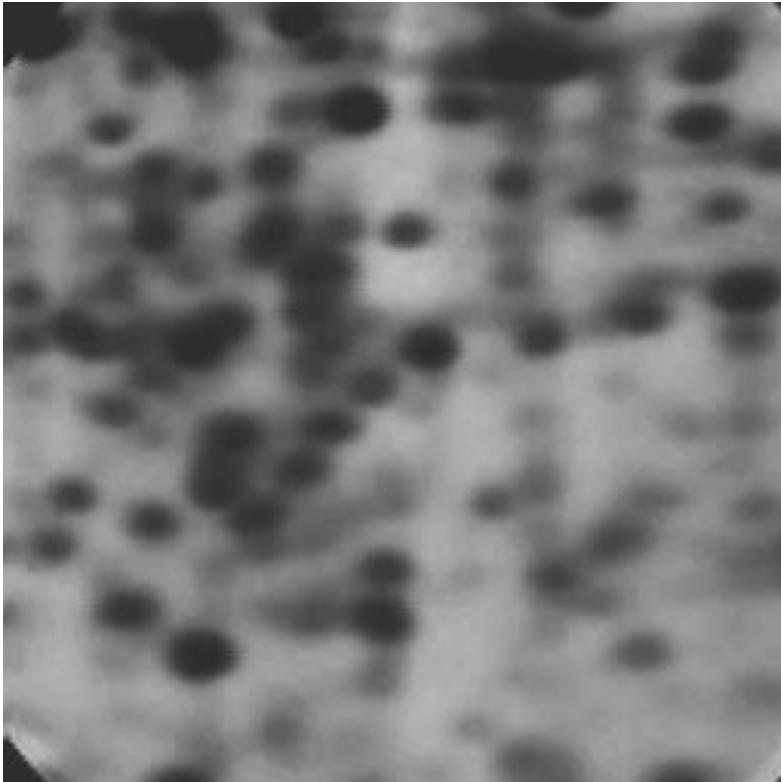


Processo

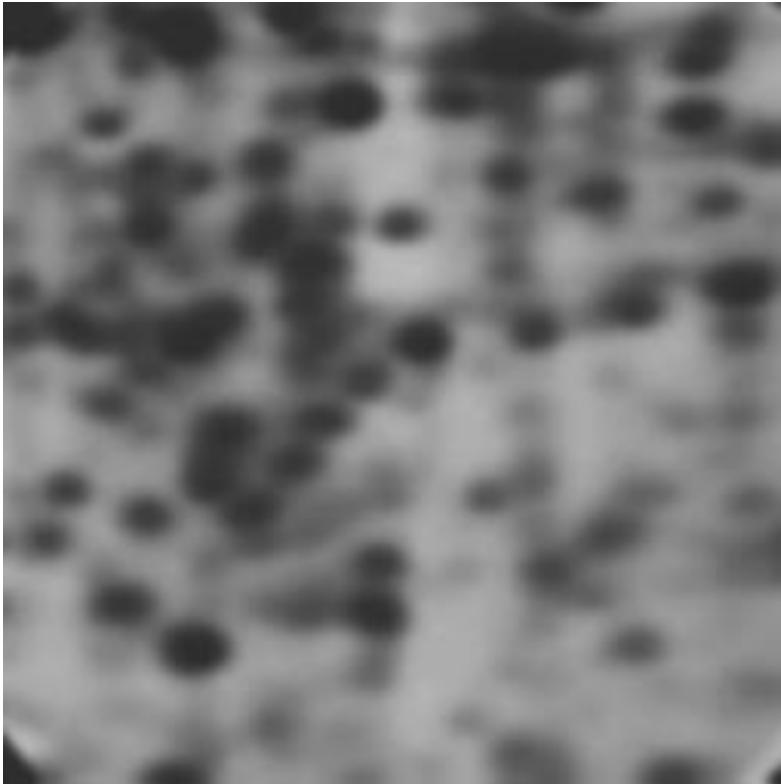


Resultado final

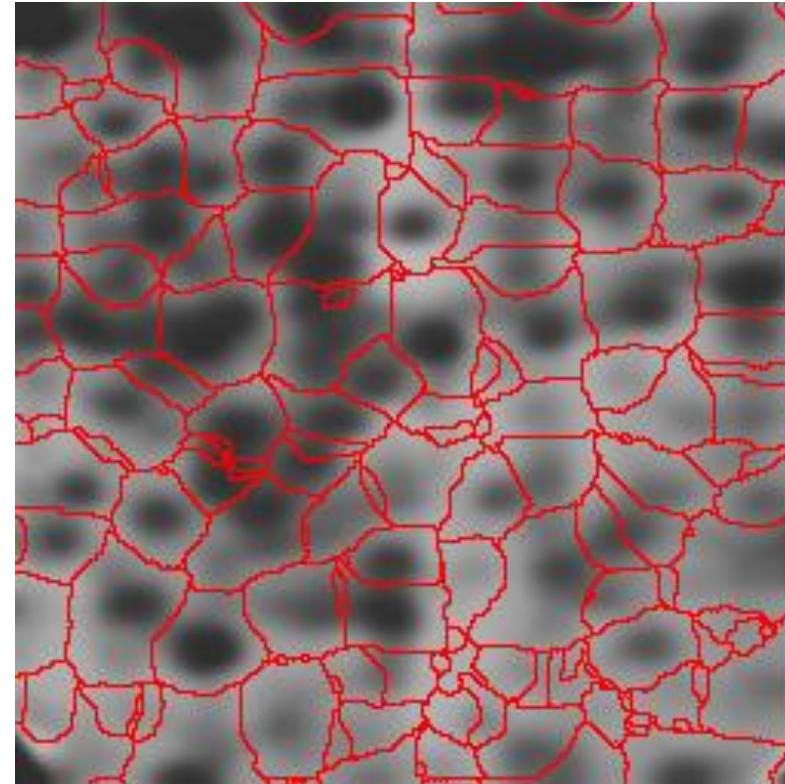
Exemplo de sobre-segmentação



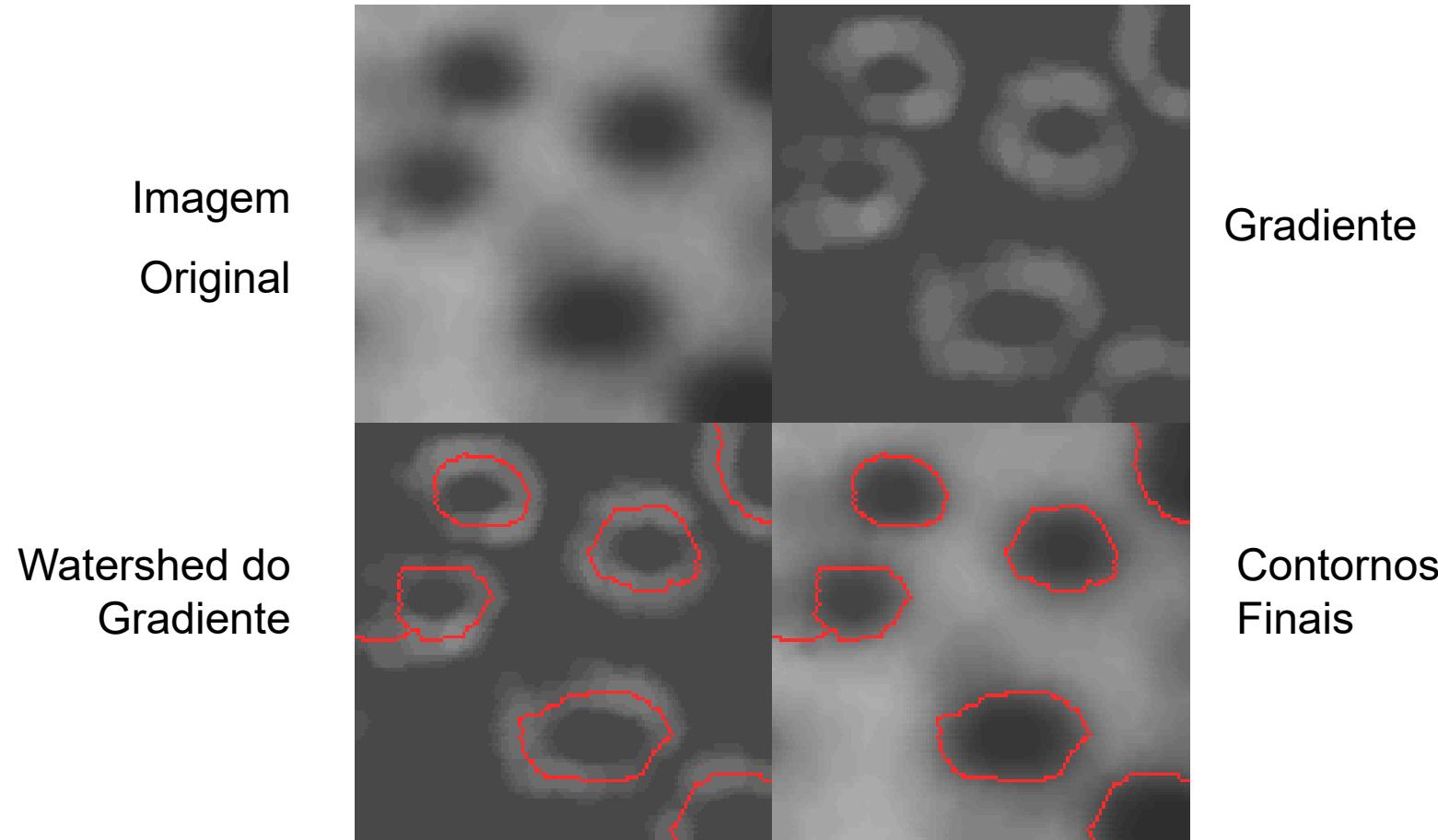
Exemplo de sobre-segmentação



Blur 5x5 Image



Outro exemplo de Watershed



Watershed com marcas

- ▶ Inundação começa a partir de marcas predefinidas



Desvantagem: tem de haver conhecimento à priori de quantas regiões e onde estão localizadas

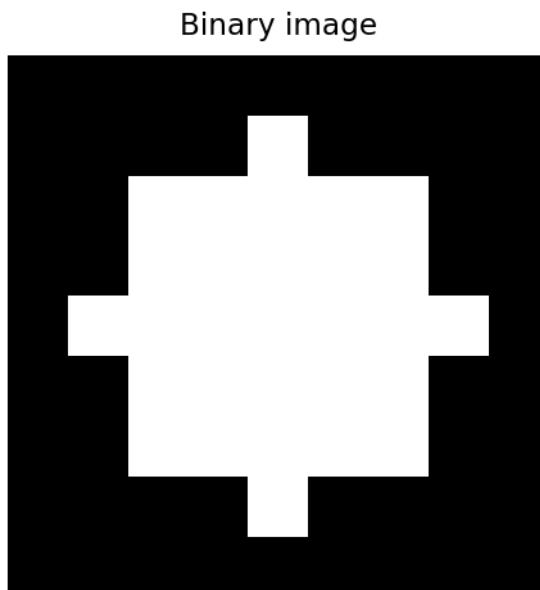


Distance Transform

Distance Transform

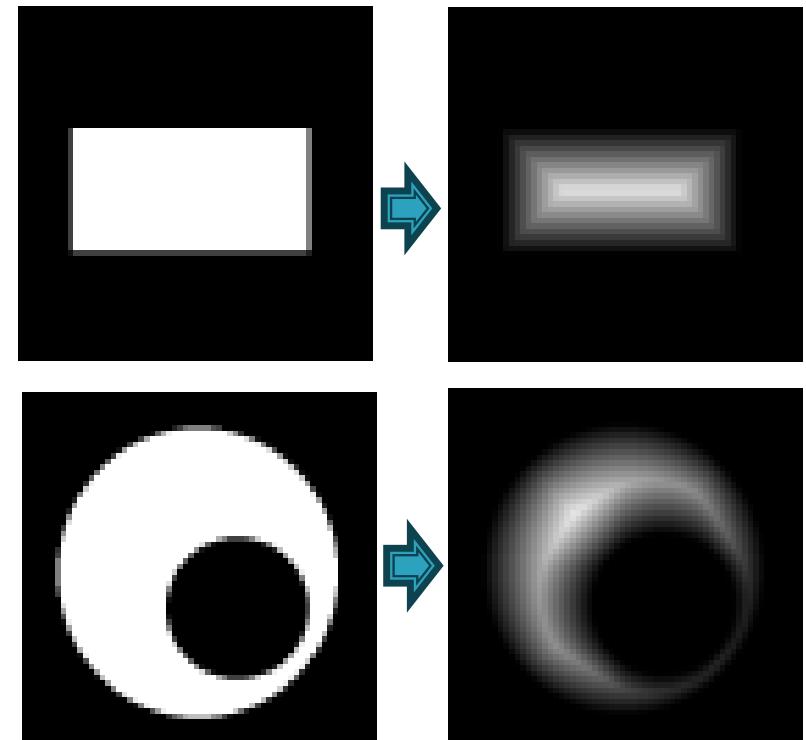
► Usada para:

- criar imagens sem planaltos (útil para a Transformada de Watershed)
- Fazer erosões ou dilatações com muitas iterações, aplicando um threshold ao resultado da distance transform



Distance transform

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	1.0	1.0	1.41	1.0	1.0	0.0	0.0	0.0
0.0	0.0	1.0	2.0	2.24	2.0	1.0	0.0	0.0	0.0
0.0	1.0	1.41	2.24	3.16	2.24	1.41	1.0	0.0	0.0
0.0	0.0	1.0	2.0	2.24	2.0	1.0	0.0	0.0	0.0
0.0	0.0	1.0	1.0	1.41	1.0	1.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0



<https://bioimagebook.github.io/chapters/2-processing/6-transforms/transforms.html>

Algoritmo

1. O processo começa com uma imagem binária, que contém pixels de primeiro plano (objetos) e pixels de fundo.
2. O algoritmo calcula a distância de cada pixel de primeiro plano até o pixel de fundo mais próximo.
3. O resultado é uma imagem de "mapa de distância".
 1. Os pixels de fundo recebem uma distância de zero.
 2. Os pixels de primeiro plano recebem valores que aumentam à medida que sua distância do pixel de fundo mais próximo aumenta.

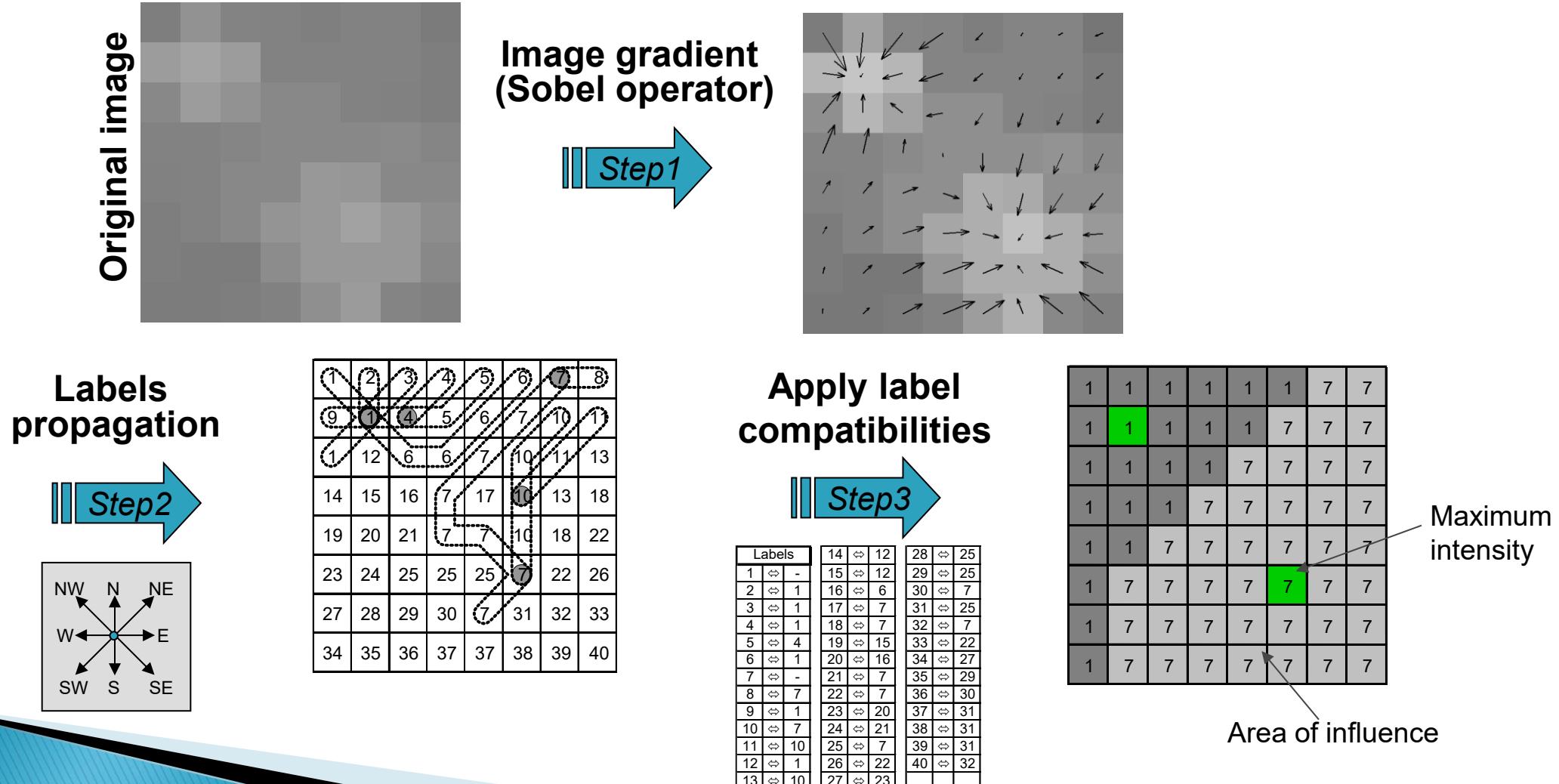


GPL – GRADIENT PATH LABELLING (CAMINHO DO MAIOR GRADIENTE)

Gradient Path Labelling

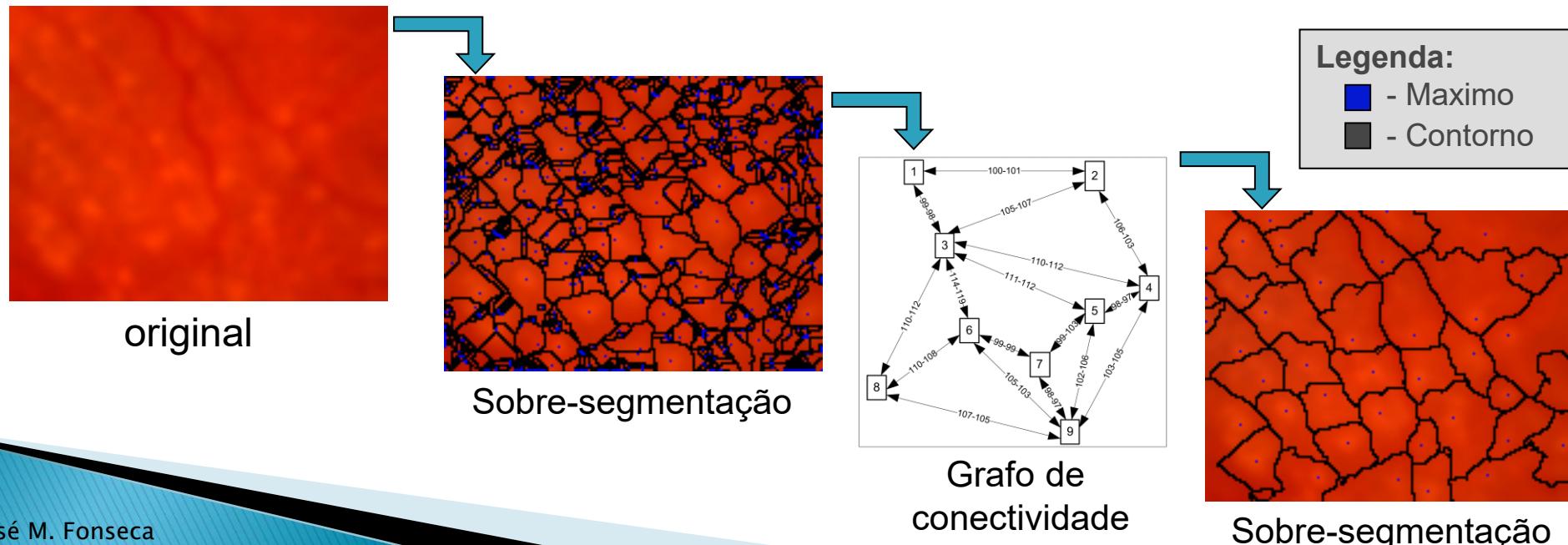
- ▶ Criado por Mora e Fonseca em 2005
- ▶ Pretende usar o **gradiente** da imagem como base da segmentação
- ▶ É maioritariamente usado para **deteção de zonas de maior ou menor intensidade** (negativo)
- ▶ É um algoritmo de etiquetagem como os Componentes Ligados, mas com um resultado idêntico à Transformada de Watershed
- ▶ **Vantagens:**
 - melhor segmentação e menor complexidade
 - Parametrizável

O algoritmo GPL (Gradient Path Labelling)



GPL- eliminação da sobresegmentação

- ▶ Nos planaltos nem todos os caminhos de gradiente terminam no mesmo máximo de intensidade gerando uma sobre-segmentação.
- ▶ Solução:
 - Fazer a **compatibilização** – juntar os máximos vizinhos que possam ser ligados através de um caminho que não desça mais do que um valor pré-definido.



Exemplo do GPL vs Watershed

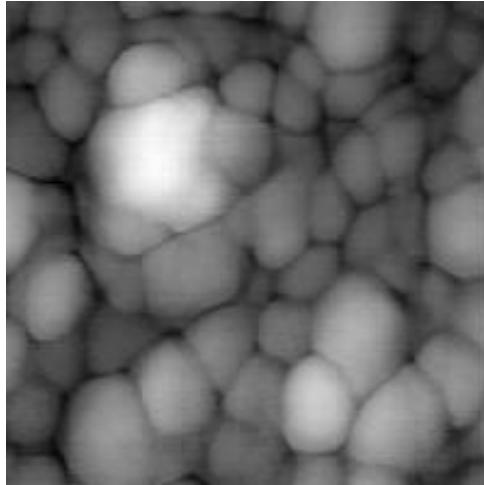
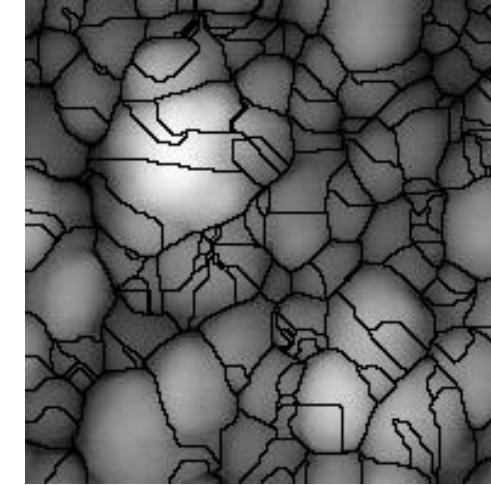


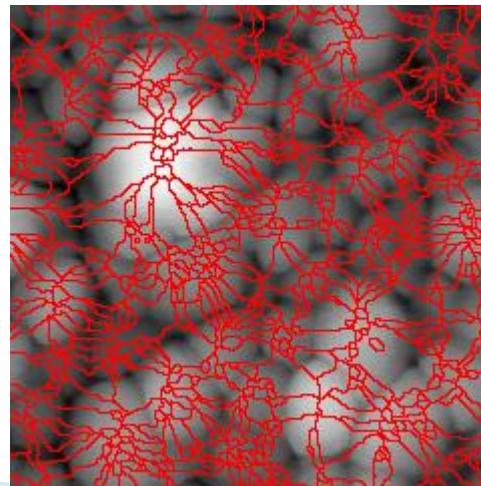
Imagen
original



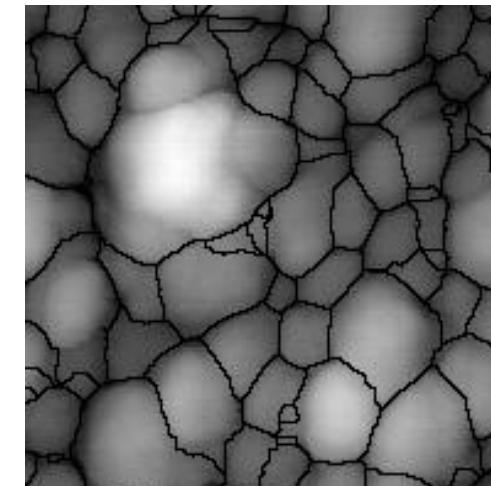
Watershed
Matlab



GPL
sem merge



Watershed
Immersion



GPL
com merge

Exemplo do GPL vs Watershed

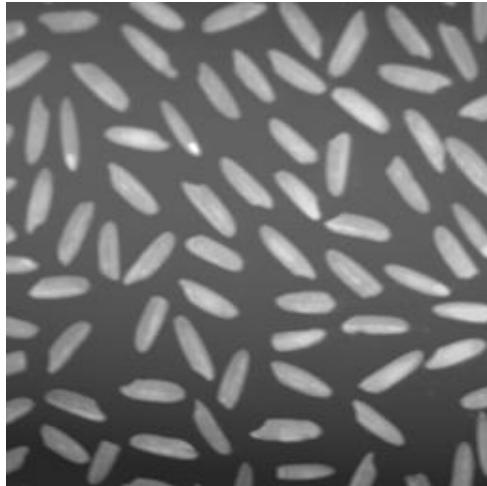
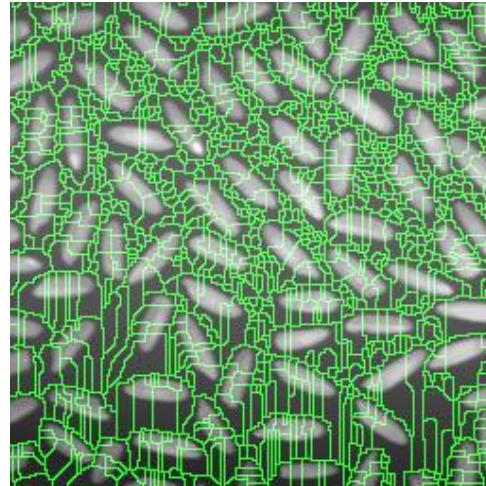
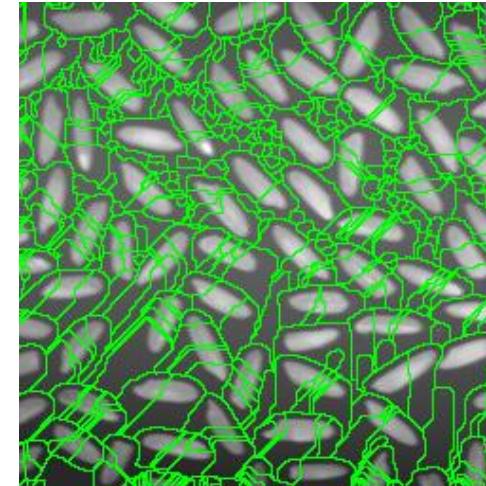


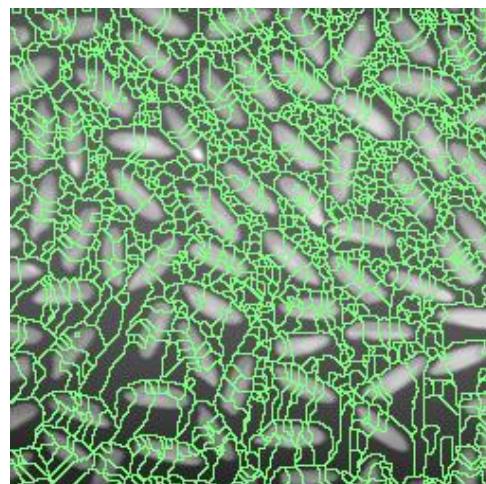
Imagen
original



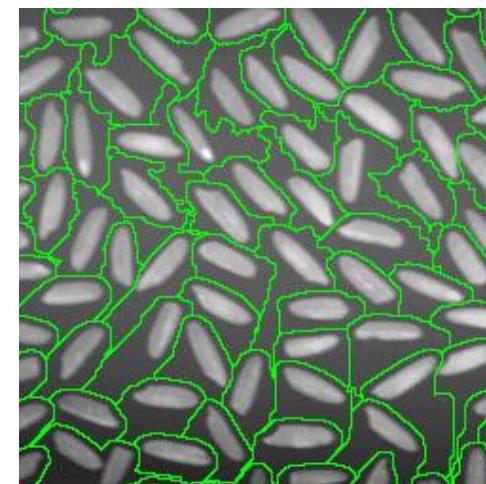
Watershed
Matlab



GPL
sem merge



Watershed
Immersion



GPL
com merge

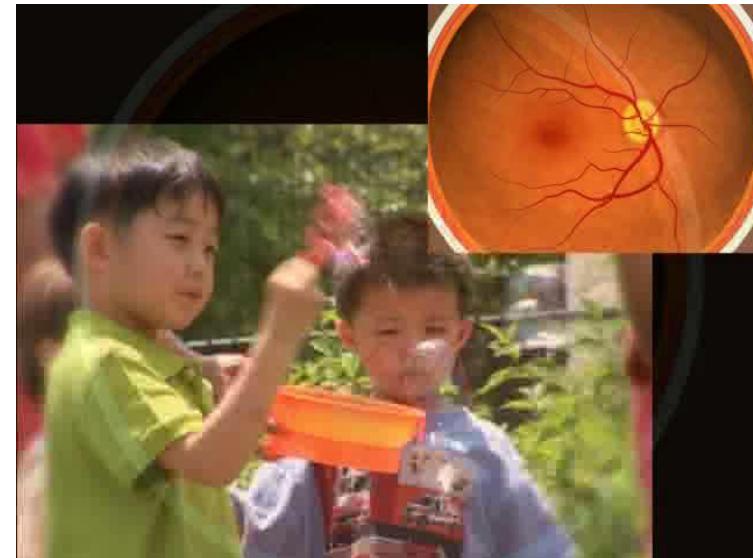
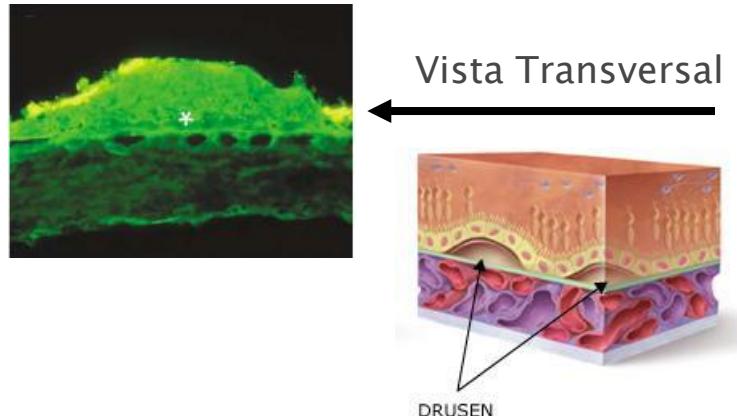
GPL – Aplicações



Aplicação a imagens da retina

Deteção de Drusas

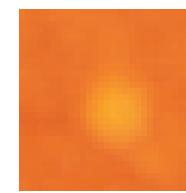
- ▶ O que são Drusas?
 - São acumulações de materiais extra-celulares por baixo da superfície da retina que formam pequenas elevações
 - São visíveis em imagens de fundoscopia como manchas de cor clara e amarelada
- ▶ Objetivo do trabalho:
 - Quantificar a área afetada



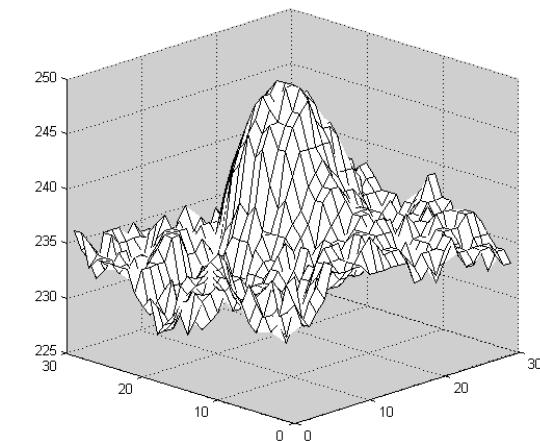
© National Eye Institute

A nossa abordagem

- ▶ Suposição: as imagens de retina permitem estimar a forma 3D
 - Numa imagem 2D em escala cinza o valor da intensidade pode representar a 3^a coordenada (profundidade)



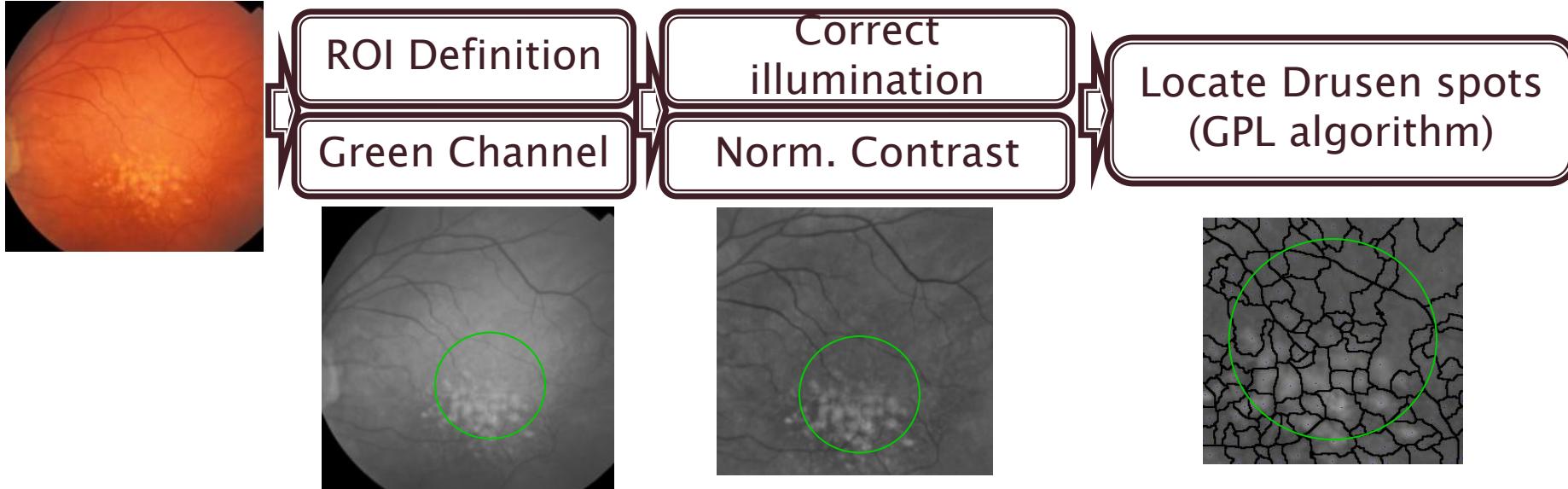
2D view



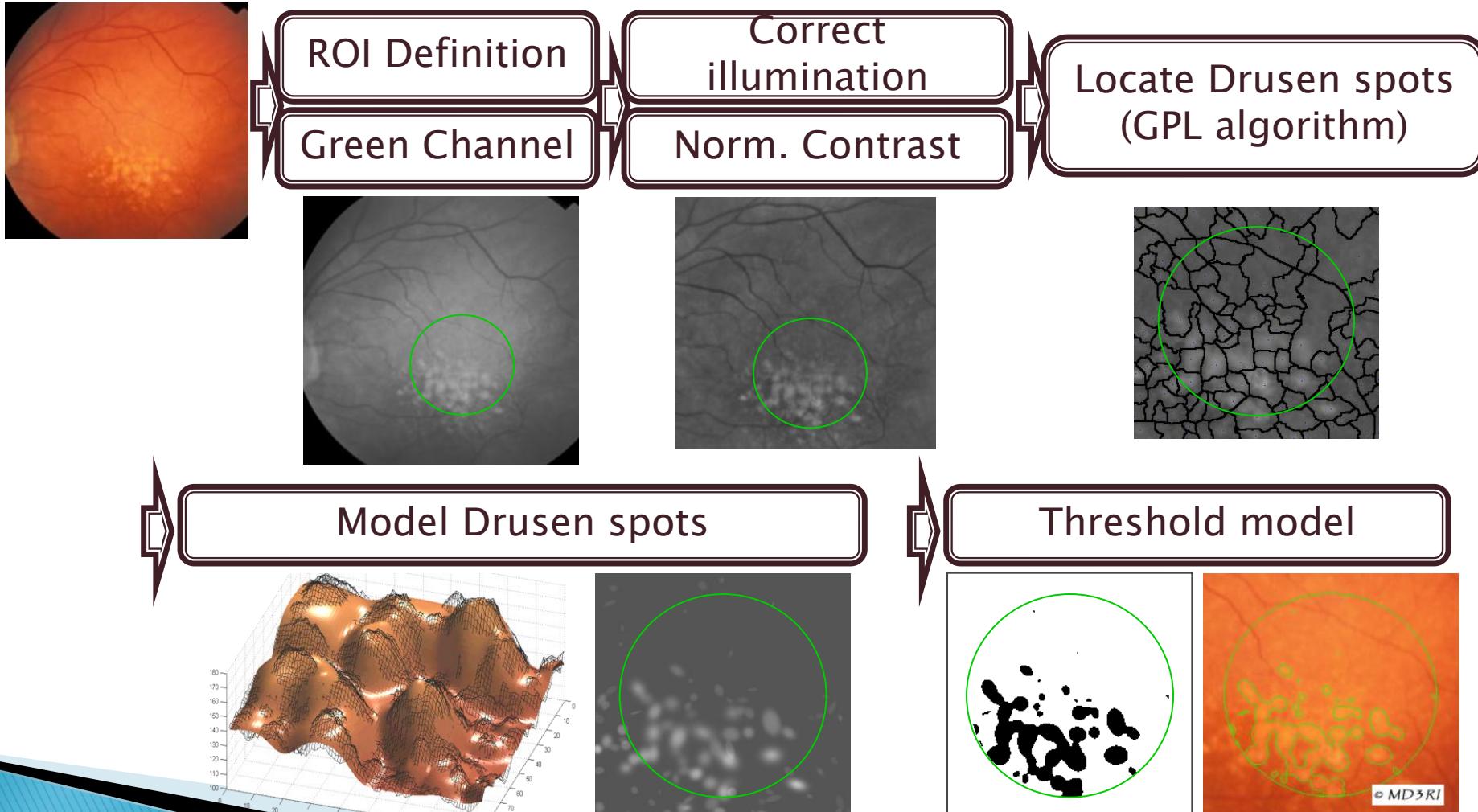
Estimated 3D view

- ▶ Algoritmo proposto:
 1. Correção da iluminação
 2. deteção de pontos de retina que podem estar relacionados com Drusen
 3. modelação de cada ponto de retina utilizando uma função 3D

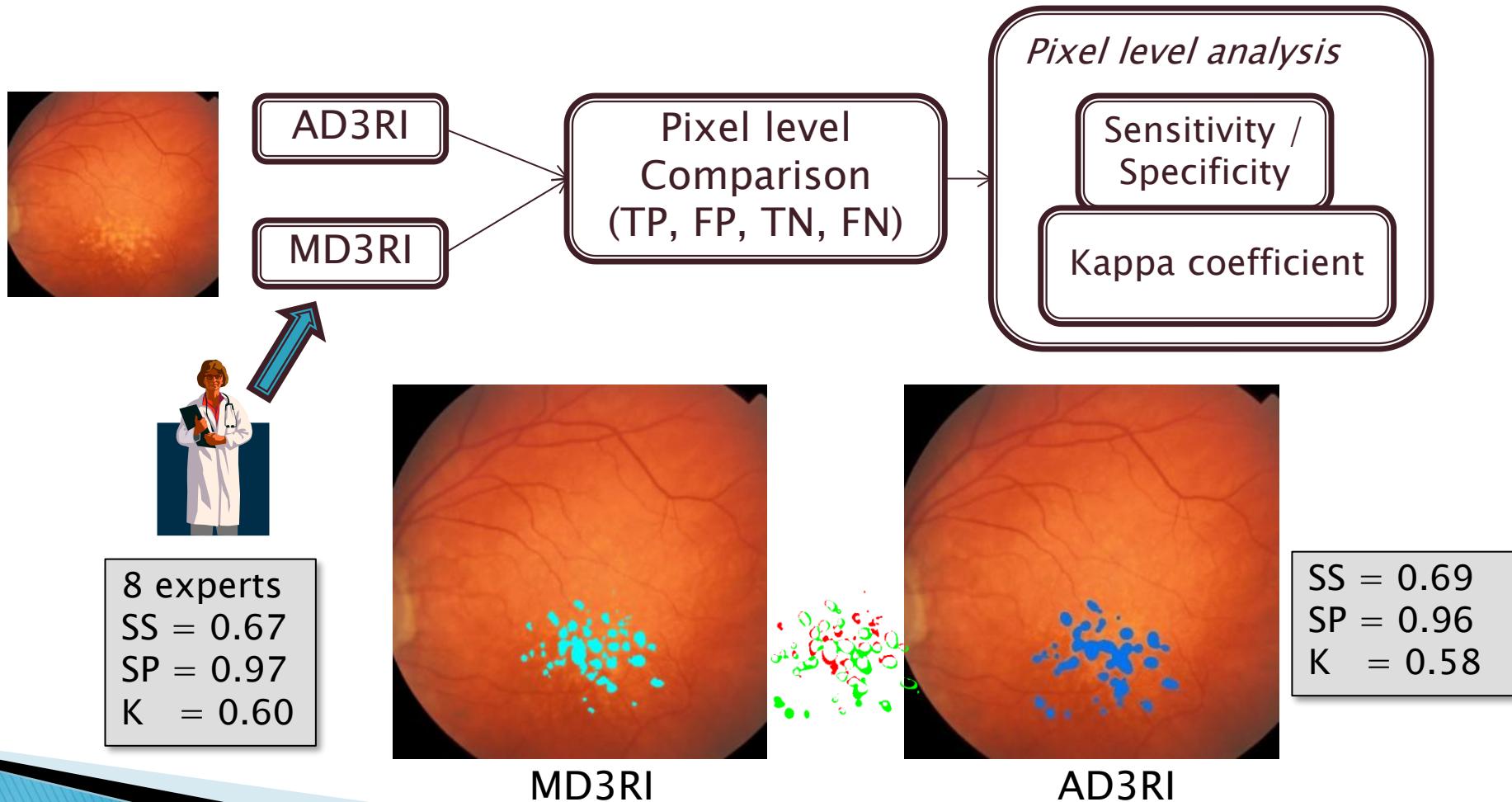
AD3RI Metodologia



AD3RI Metodologia

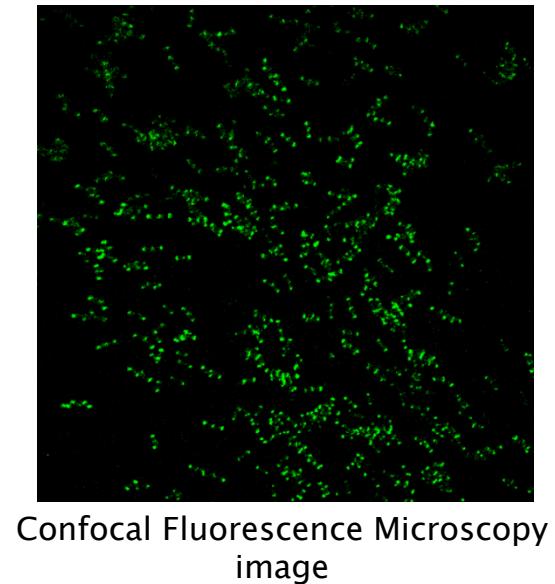
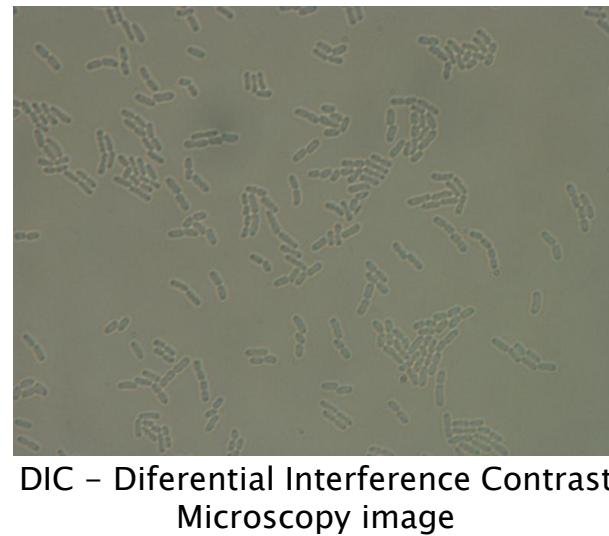
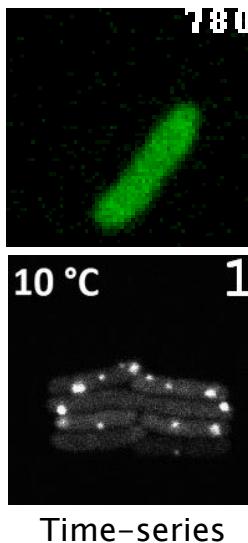


Evaluation tests - pixel level

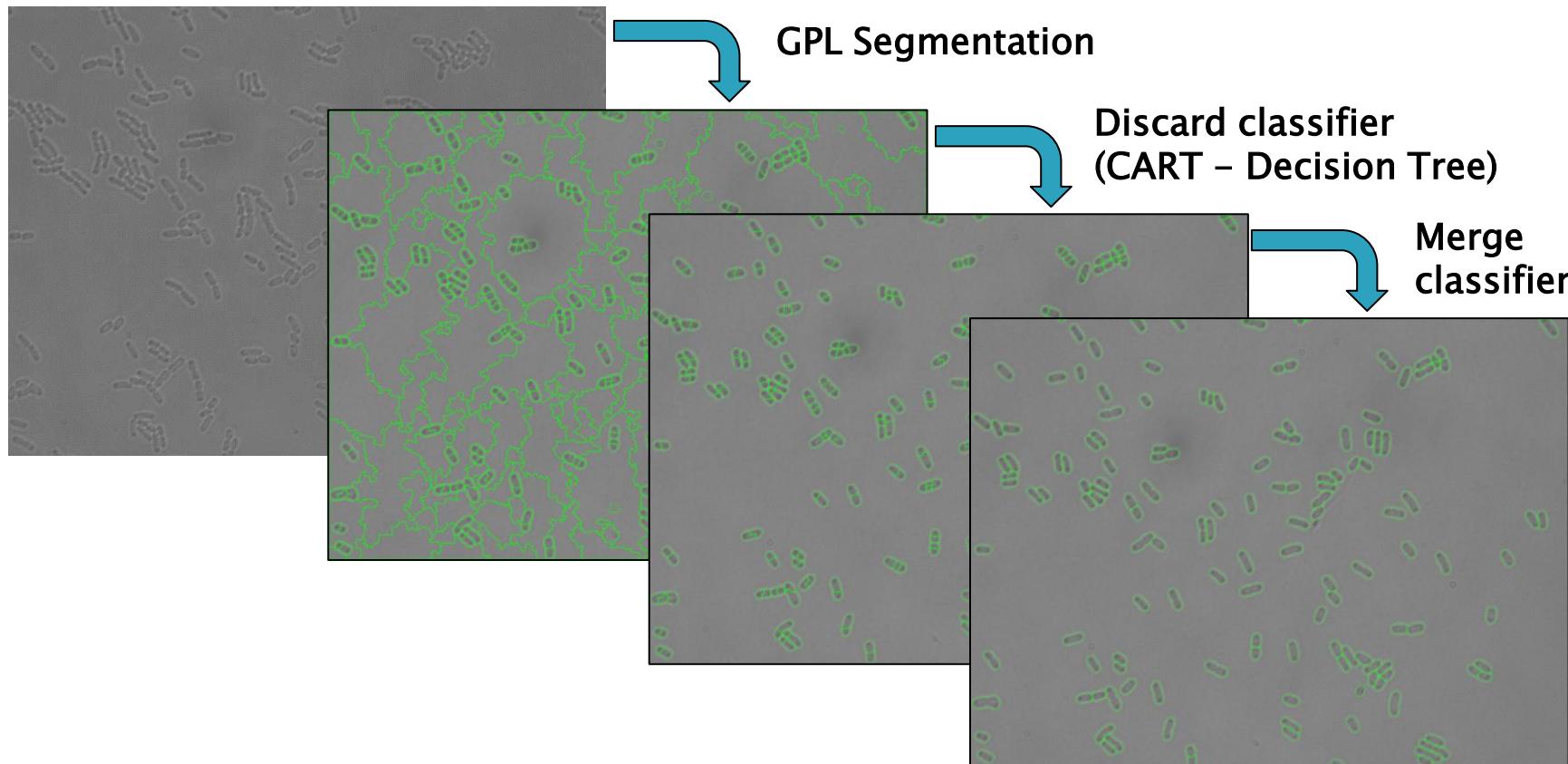


Bacteria segmentation and tracking

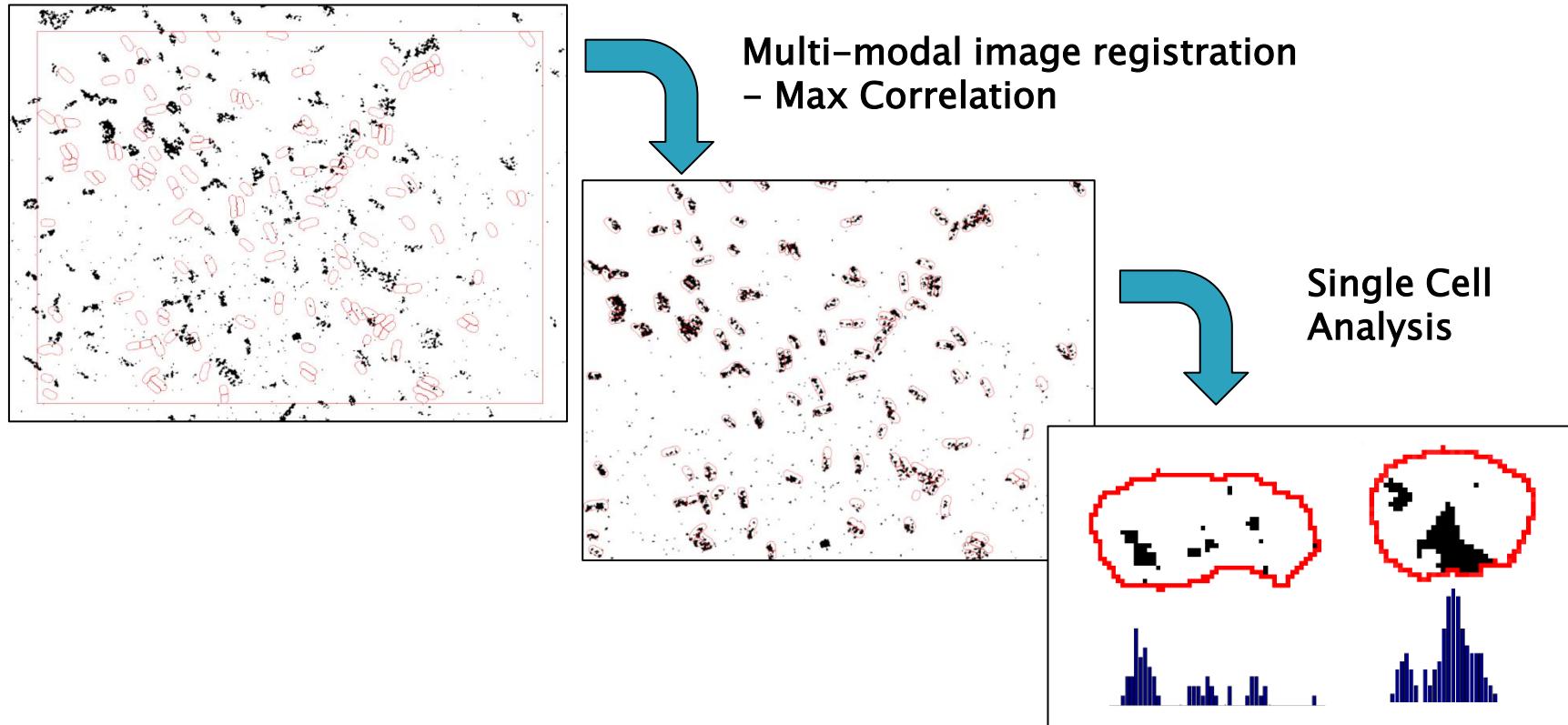
- ▶ Study single cell dynamics from microscopy time-series images for statistically significant studies
- ▶ Extraction from multi-modality images, individual cells information belonging to bacteria populations with large number of elements.



Segmentation / Classification Procedure



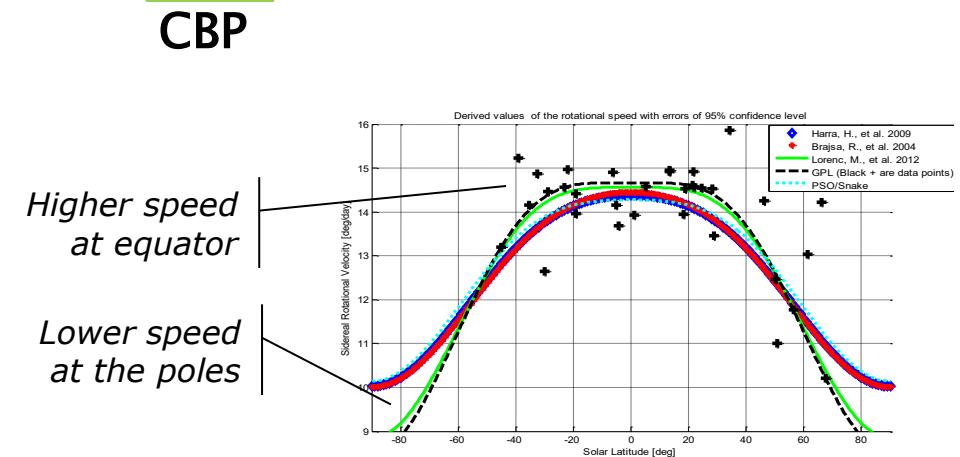
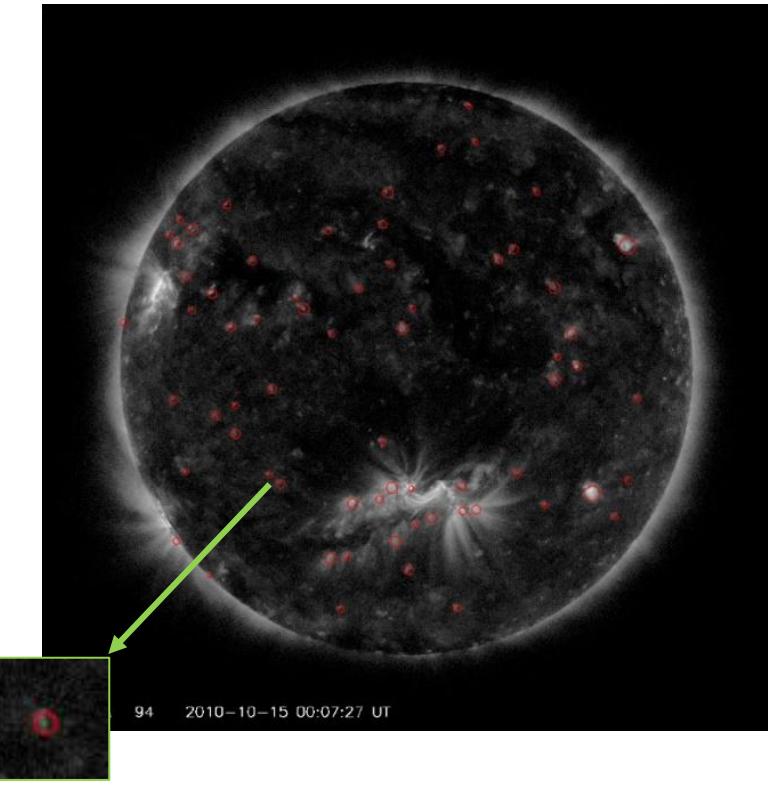
Multi-modal alignment and cell analysis



Solar Rotation Profile

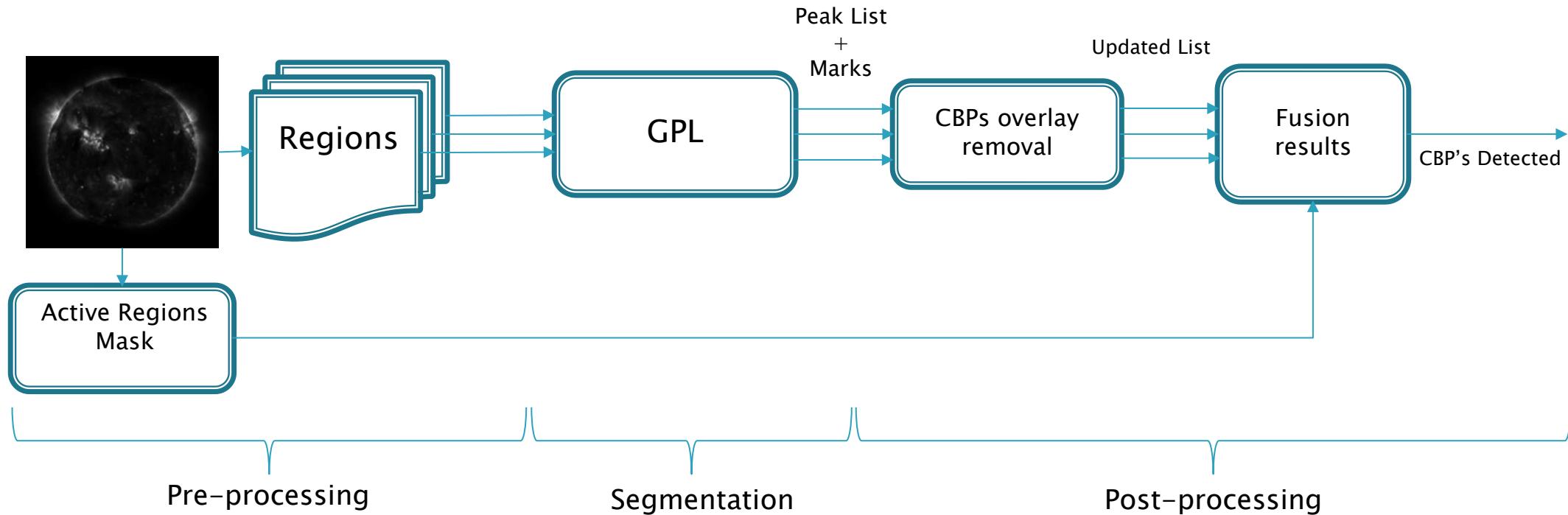
- ▶ Detection of Coronal Bright Points (CBPs) in solar time-series and evaluate the solar rotation profile
- ▶ SDO – Solar Dynamics Observatory (2010) produces 16 Mpixel images every 10 min.

- ▶ Method:
 - GPL segmentation,
 - Filter the outputs by size and location
 - Tracked CBPs location

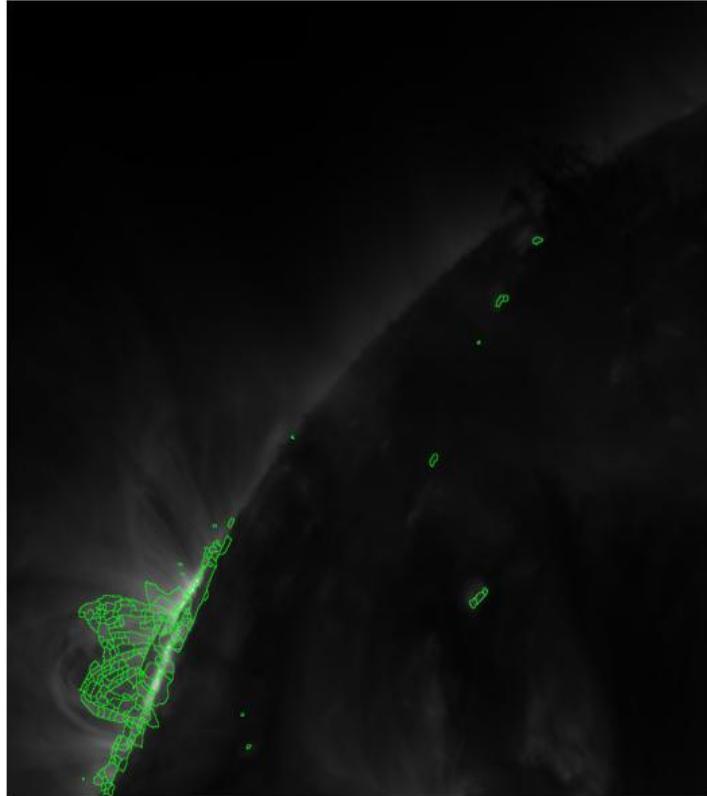


Dorotovič, I., Coelho, A., Rybák, J., Mora, A., & Ribeiro, R. (2018). Gradient Path Labelling method and tracking method for calculation of solar differential rotation using coronal bright points. *Astronomy and Computing*, 25, 168–175. doi: 10.1016/j.ascom.2018.09.008

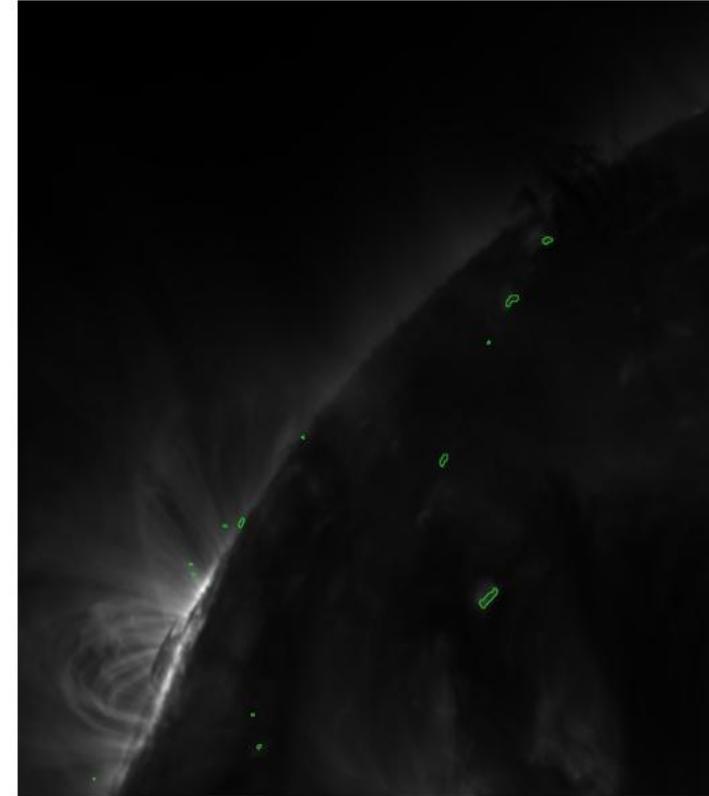
Solar Rotation Profile – II



Solar Rotation Profile – III



GPL
+ minimum size filtering



After remerge +
minimum size filtering