

# Segmentação

# Bibliografia

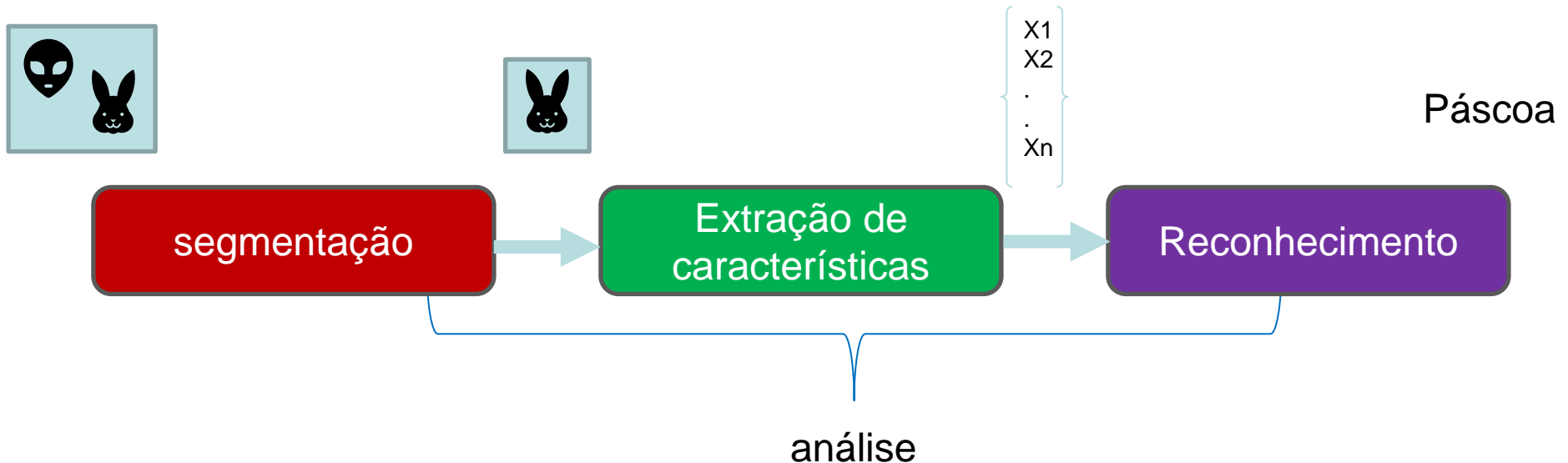


**Processamento digital de imagens, 3ª ed.**

Rafael C. Gonzalez e Richard E. Woods

# Segmentação

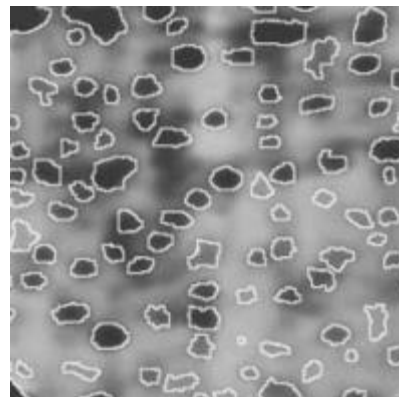
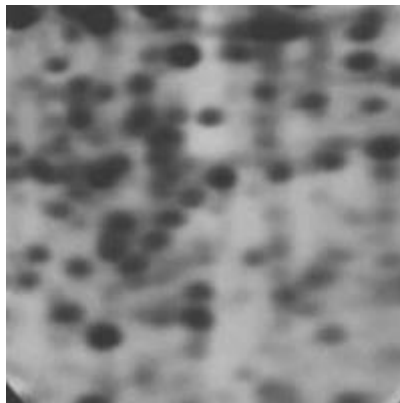
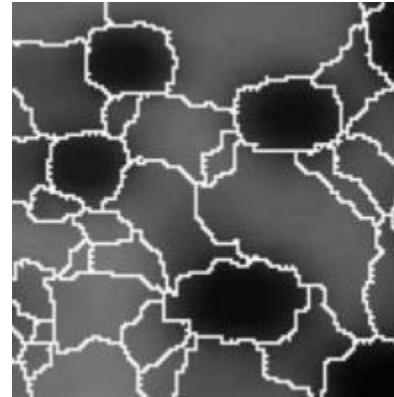
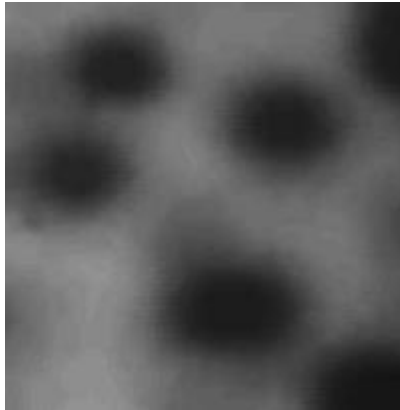
- Segmentação é uma tarefa básica no processo de análise de imagens



# Segmentação => Simplificação

- A simplificação da imagem é uma questão central na área de processamento de imagens e visão computacional, o que pode ser feito reduzindo-se as informações da imagem para regiões mais ou menos homogêneas
- O resultado é uma “caricatura” da realidade onde somente a parte importante está presente, sendo que os detalhes desnecessários e ruídos são extraídos.
- Aplicações:
  - Controle de qualidade
  - Inspeção automatizada de peças de fábricas
  - *Robótica*

# Problema de Segmentação



# Problema da segmentação

- É uma das tarefas mais difíceis em processamento de imagens
- Com o objetivo de aumentar a confiabilidade e o resultado da segmentação, deve-se fazer uso de todo e qualquer conhecimento prévio sobre o problema
- A segmentação determina o eventual sucesso ou fracasso da análise
- Os algoritmos de segmentação são geralmente baseados na busca pelas **descontinuidades** ou pelas **similaridades** dos níveis de cinza

# Algoritmos de segmentação

```
graph TD; A[Algoritmos de segmentação] -- "Mudanças bruscas nos níveis de cinza" --> B[Descontinuidades]; A -- "Semelhança nos níveis de cinza" --> C[Similaridades]; B -- "Detecção de:" --> D["▪ Pontos isolados<br/>▪ Linhas e<br/>▪ Bordas na imagem"]; C -- "Aplicação de:" --> E["▪ Limiarização<br/>▪ Crescimento de regiões"]
```

Mudanças bruscas  
nos níveis de  
cinza

## Descontinuidades

Detecção de:

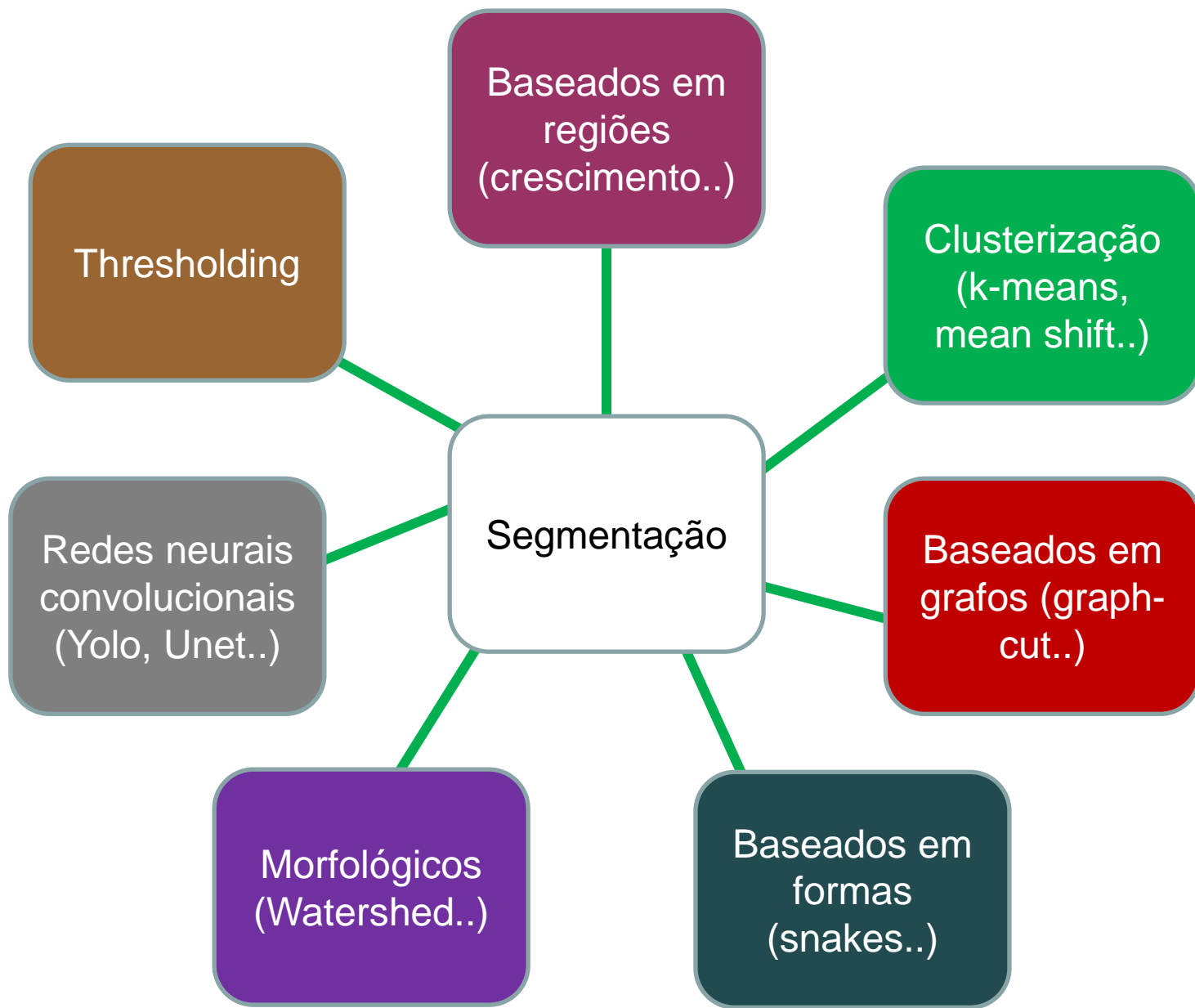
- Pontos isolados
- Linhas e
- Bordas na imagem

Semelhança nos  
níveis de cinza

## Similaridades

Aplicação de:

- Limiarização
- Crescimento de regiões





# Limiarização

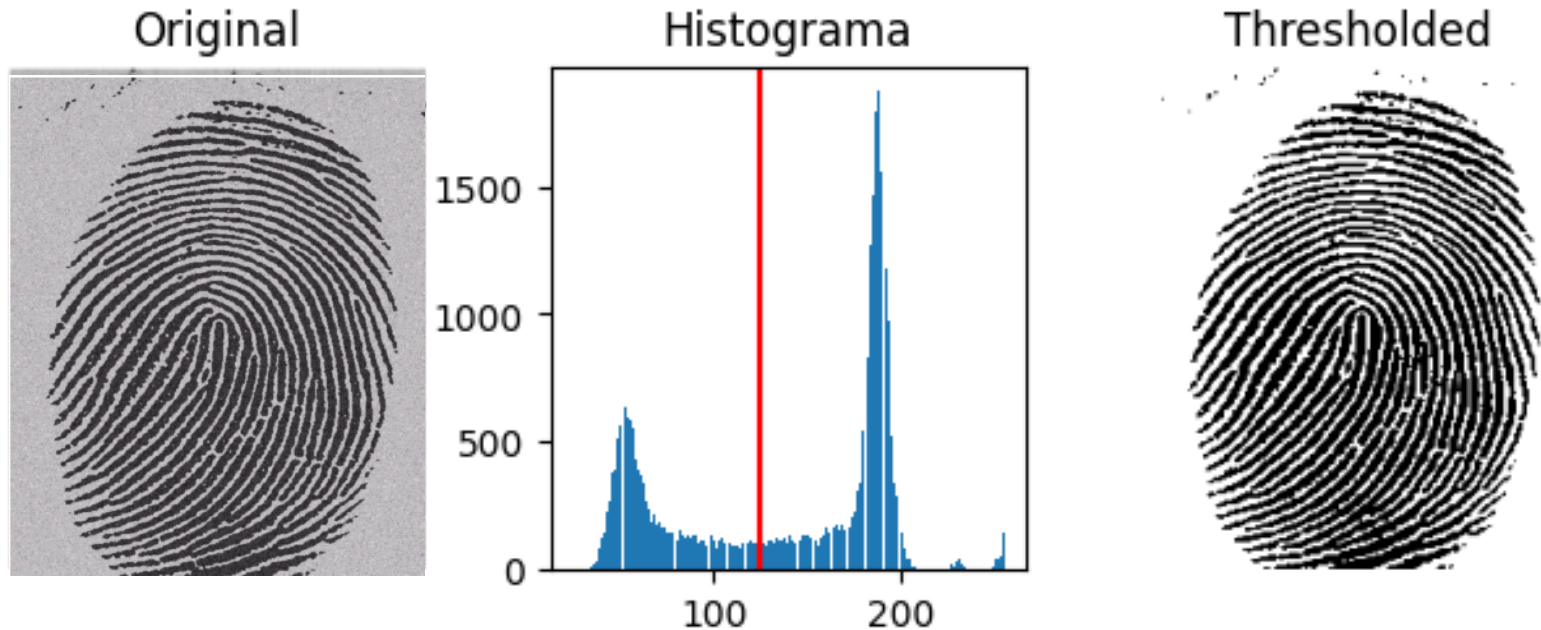
- A maneira óbvia de extrair objetos é selecionar um limiar  $T$  que separe os dois grupos.
- Então, para cada ponto  $(x,y)$  tal que  $f(x,y) > T$  é denominado ponto do objeto, caso contrario, fundo

$$g(x, y) = \begin{cases} 1 & \text{se } f(x, y) > T \\ 0 & \text{se } f(x, y) \leq T \end{cases}$$

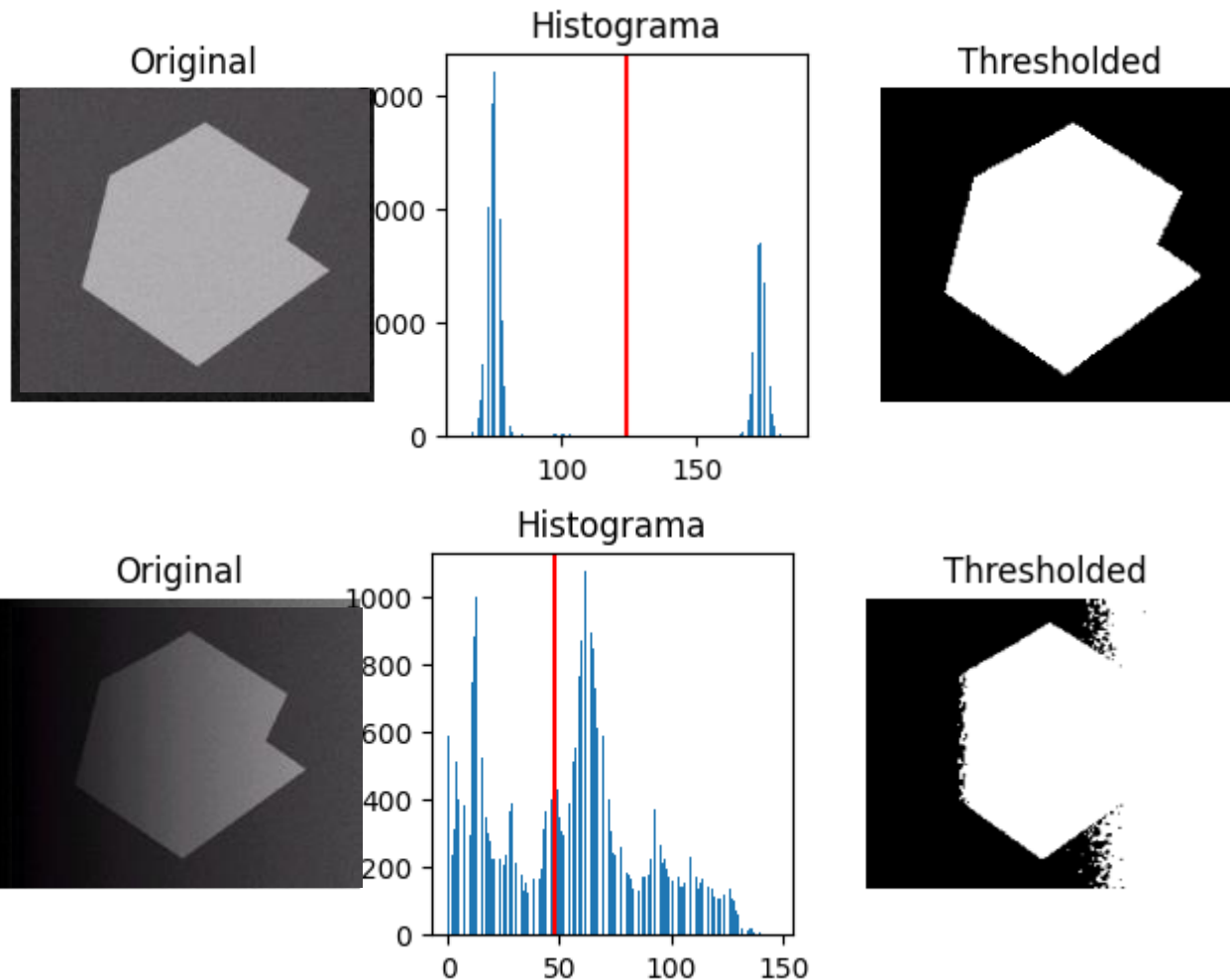
- Nesse caso,  $T$  depende apenas de  $f(x,y)$  e portanto é conhecida como limiarização global.

# Limiarização

- Consiste em separar regiões de uma imagem quando esta apresenta duas classes:
  - Fundo e objeto (background, foreground)



# Threshold único e iluminação



- Se tivermos objetos iluminados e/ou com efeito de degradê, técnicas de *threshold* com valor único não funcionam

# Segmentação

- Vimos métodos simples para segmentar a imagem, como:
  - Threshold
  - Histograma
  - Etc...

... Mas, esses métodos deixam resto de informações desnecessárias ou perdem informações

## Region-based segmentation

Let us first determine markers of the coins and the background. These markers are pixels that we can label unambiguously as either object or background. Here the markers are found at the two extreme parts of the histogram of grey values:

## Threshold-based segmentation

Let us first determine markers of the coins and the background. These markers are pixels that we can label unambiguously as either object or background. Here the markers are found at the two extreme parts of the histogram of grey values:

# Segmentação por Regiões

- Para superar essas limitações de métodos no domínio do valor, foram desenvolvidos métodos que operam no domínio do espaço e utilizam informações sobre:
    - Vizinhança de pixel
    - Variação de valor de pixel em uma vizinhança
    - Estatísticas sobre agrupamentos de pixel
- ... Para gerar uma imagem resultante dividida em regiões

# Segmentação por Regiões

- A partir de um pixel inicial, chamado de semente, agrega-se outros pixels de acordo com um critério pré-estabelecido.
- Entre as características utilizadas nos critérios de agregação estão: **nível de cinza, cor, textura ou estatística da população de pixels**

# Segmentação por Regiões

## Premissas:

1. A região deve ser homogênea considerando algum fator de tolerância e algumas características predefinidas
2. Só serão consideradas “regiões fechadas” aquelas delimitadas por fronteiras contínuas que separam seu interior dos outros segmentos
3. As características devem ser adequadas para indicar univocamente a região; e
4. O conjunto de todas as regiões deve formar a imagem

# Segmentação por Regiões

- Seja  $R$  a região completa de uma imagem. Pode se imaginar a segmentação como um processo de dividir  $R$  em  $n$  regiões  $R_1, R_2, \dots, R_n$ , tal que:

$$\bigcup_{i=1}^n R_i = R$$

$$R_i \cap R_j = \emptyset$$

$$P(R_i) = V$$

$$P(R_i \cup R_j) = F$$

- Em que  $P(R_i)$  é um predicado lógico (ex: todos os pixels possuem a mesma intensidade) sobre os pontos do conjunto  $R_i$  e  $\emptyset$  é o conjunto vazio.



# Segmentação por Regiões

- Agregação de pixels
  - Agrupa pixels ou sub-regiões em regiões maiores.
  - Começa com um conjunto de pontos, chamados sementes, e a partir deles, cresce as regiões anexando a cada ponto semente aqueles pixels que possuam propriedades similares
    - Nível de cinza, textura, cor.

# Segmentação por Regiões

Sementes

0	0	5	6	7
1	1	5	8	7
0	1	6	7	7
2	0	7	6	6
0	1	5	6	5

Propriedade P: Diferença absoluta entre os níveis de cinza daqueles entre o pixel e semente deve ser menor que T

T=3

0	0	5	6	7
1	1	5	8	7
0	1	6	7	7
2	0	7	6	6
0	1	5	6	5

Note que qualquer semente levaria ao mesmo resultado.

## Desvantagens:

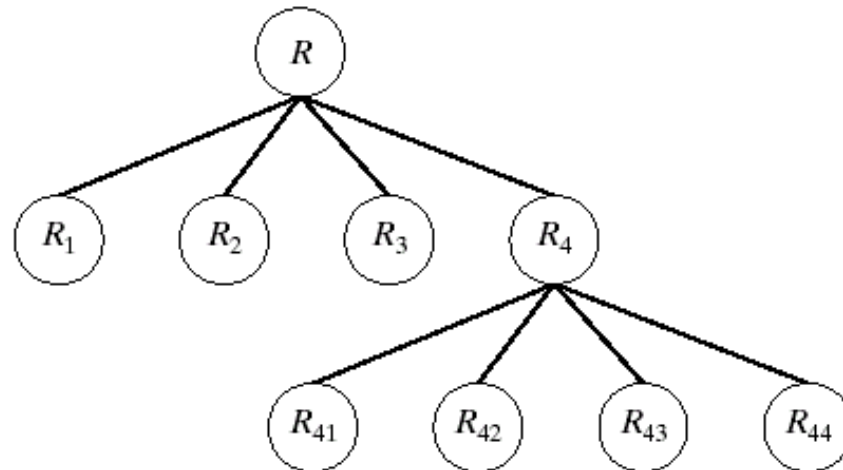
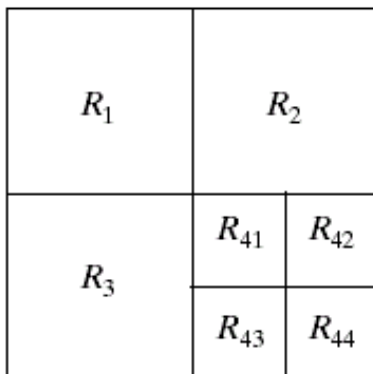
- Seleção das sementes
- Seleção da propriedade P

# Segmentação por Regiões

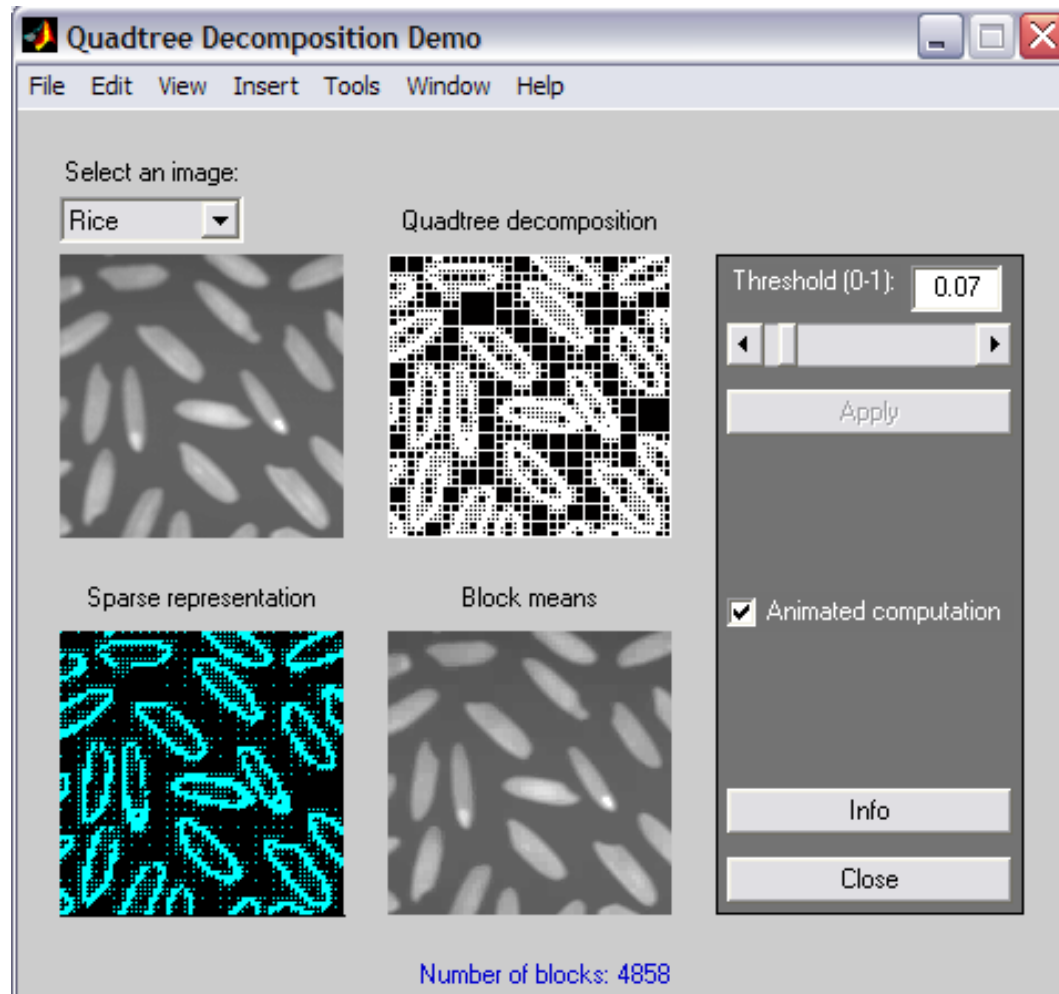
- Divisão e fusão
  - Seja  $R$  a imagem completa e seja  $P$  um predicado.
  - No caso de uma imagem quadrada, uma abordagem para segmentação de  $R$  consiste em subdividi-la sucessivamente em quadrantes cada vez menores de modo que  $P(R_i) = V$ .
  - Ou seja, se  $P(R)$  for falso para qualquer quadrante, o quadrado deve ser subdividido em sub-quadrantes.
  - Essa técnica possui uma representação conveniente chamada *quadtree*

# Segmentação por Regiões

- As fusões são limitadas inicialmente a grupos de quatro blocos que sejam descendentes na representação *quadtree* e que satisfaçam o predicado P.
- Quando fusões desse tipo não forem mais possíveis, o procedimento é terminado por uma fusão final.



# Segmentação por Regiões



# Clustering

- Existem basicamente dois tipos de clustering:
  - Divisive
    - A imagem é vista como um cluster, e então são feitas divisões sucessivas.
      - Segmentação por regiões.
  - Agglomerative
    - Cada pixel é visto como um cluster, e clusters são unidos recursivamente até formarem um bom cluster.

# Clustering

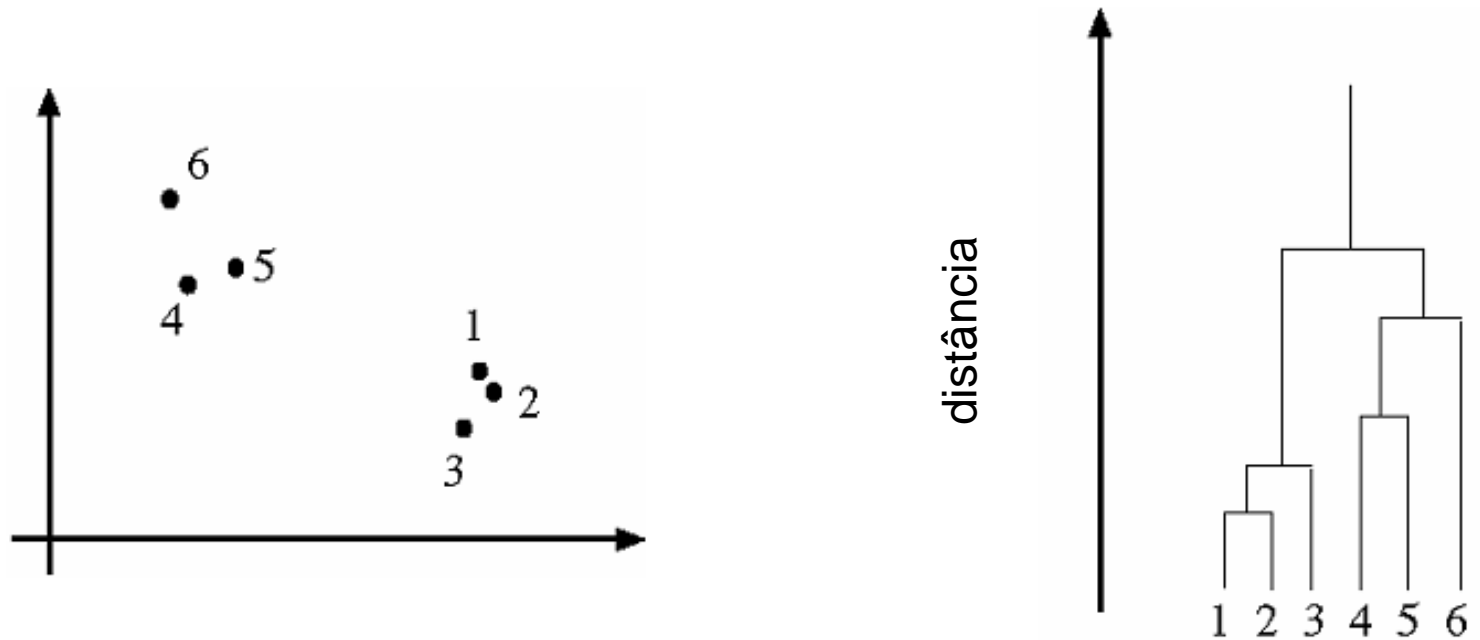
- Parâmetro a definir
  - Número de clusters.
  - Geralmente não se conhece a priori o número de clusters que existem na imagem.

# Clustering

- Uma coleção de objetos que são similares entre si, e diferentes dos objetos pertencentes a outros clusters.
- Isso requer uma medida de similaridade.
- No exemplo anterior, a similaridade utilizada foi a *distância*.
  - *Distance-based Clustering*



# Dendrograma Clustering



# K-Means

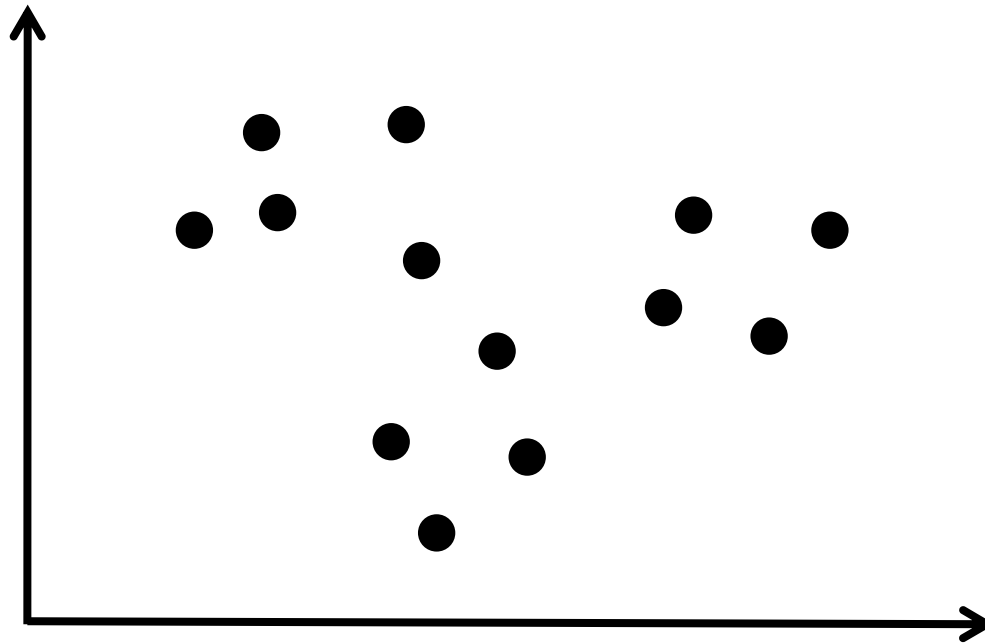
- É a técnica **mais simples** de aprendizagem não supervisionada.
- Consiste em fixar **k centróides** (de maneira aleatória), um para cada grupo (clusters).
- Associar cada indivíduo ao seu **centróide mais próximo**.
- Recalcular os centróides com base nos indivíduos classificados.

# Algoritmo K-Means

- (1) Selecione  $k$  centróides iniciais.
- (2) Forme  $k$  clusters associando cada exemplo ao seu centróide mais próximo.
- (3) Recalcule a posição dos centróides com base no centro de gravidade do cluster.
- (4) Repita os passos 2 e 3 até que os centróides não sejam mais movimentados.

# Algoritmo K-Means

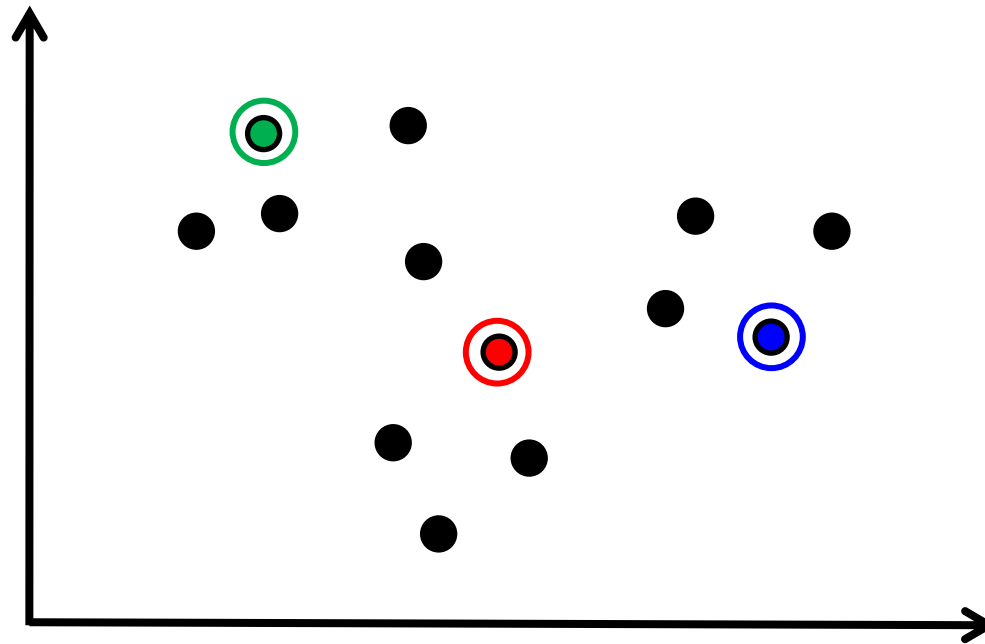
- Exemplo:



# Algoritmo K-Means

- Exemplo:

$k = 3$

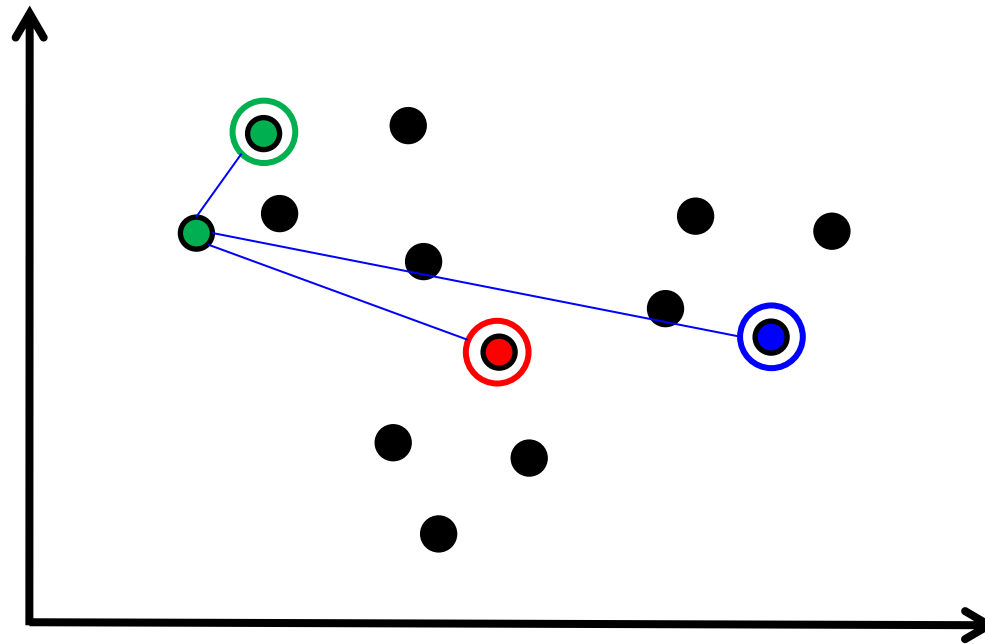


Seleciona-se  $k$  centróides iniciais.

# Algoritmo K-Means

- Exemplo:

$k = 3$

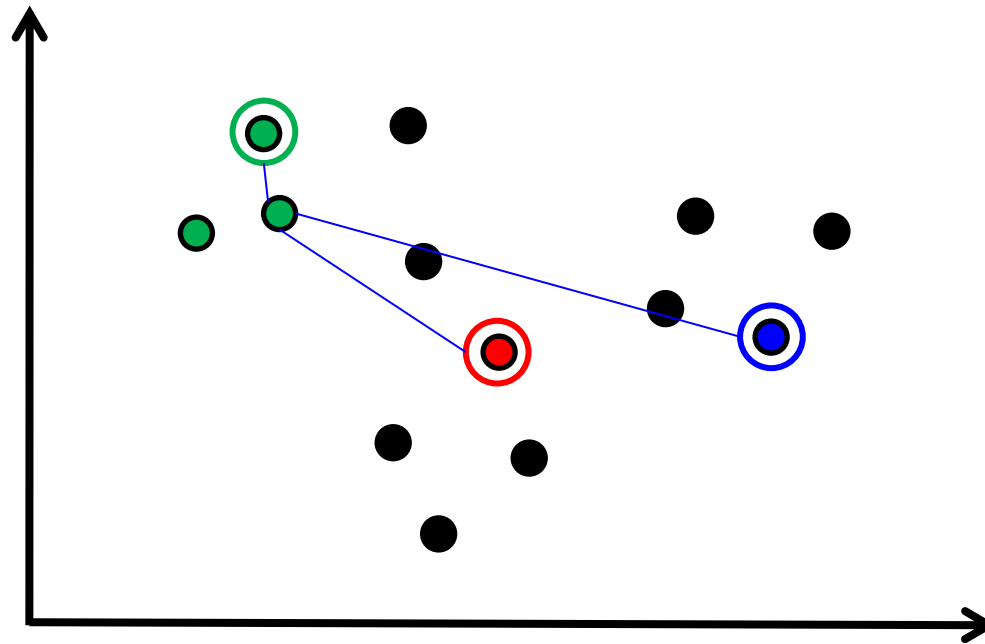


1ª iteração

# Algoritmo K-Means

- Exemplo:

$k = 3$

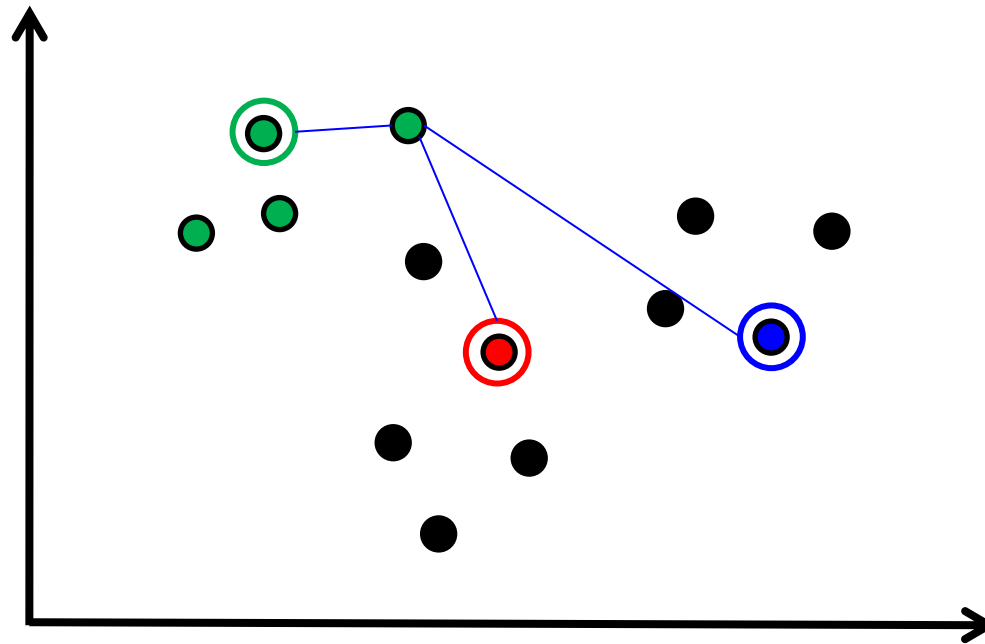


2ª iteração

# Algoritmo K-Means

- Exemplo:

$k = 3$



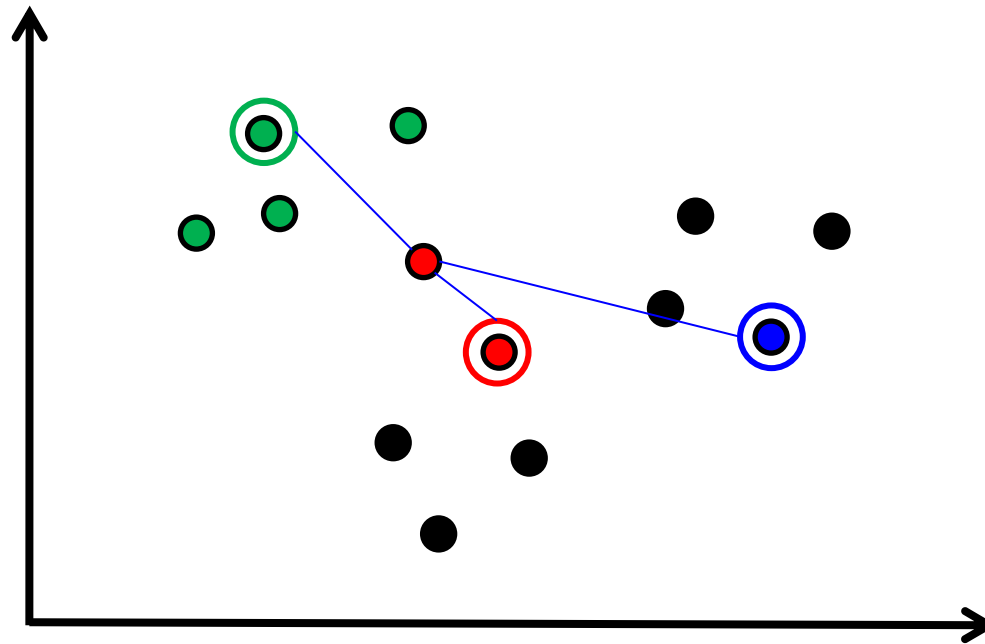
3ª iteração



# Algoritmo K-Means

- Exemplo:

$k = 3$

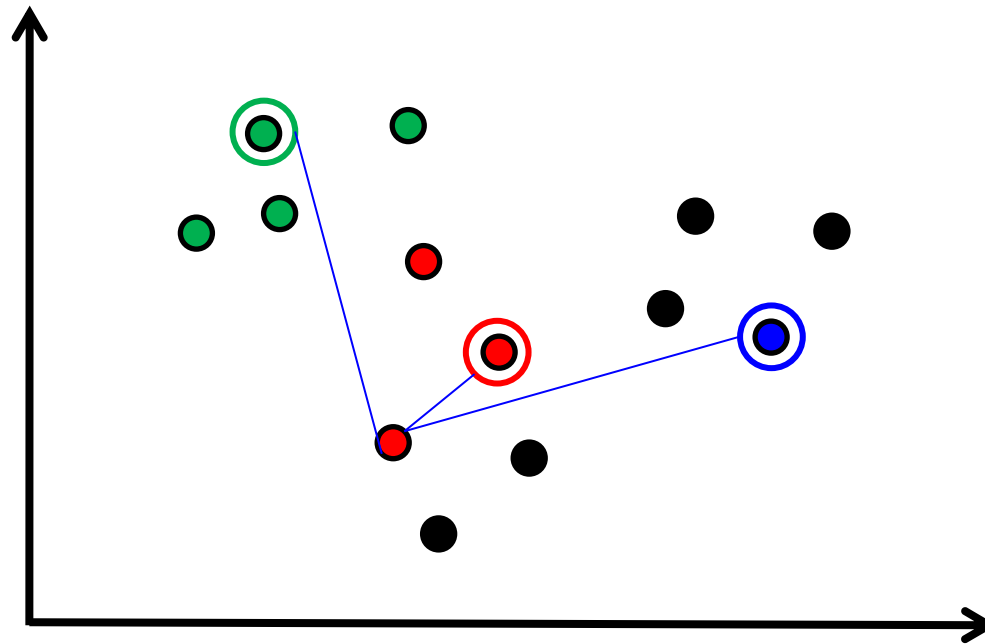


4ª iteração

# Algoritmo K-Means

- Exemplo:

$k = 3$

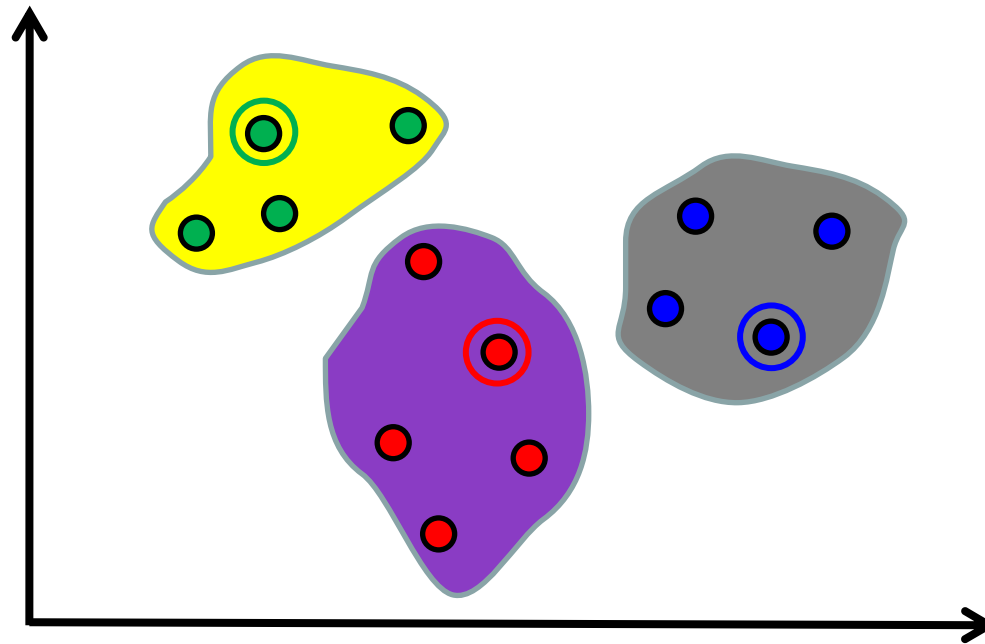


5ª iteração

# Algoritmo K-Means

- Exemplo:

$k = 3$

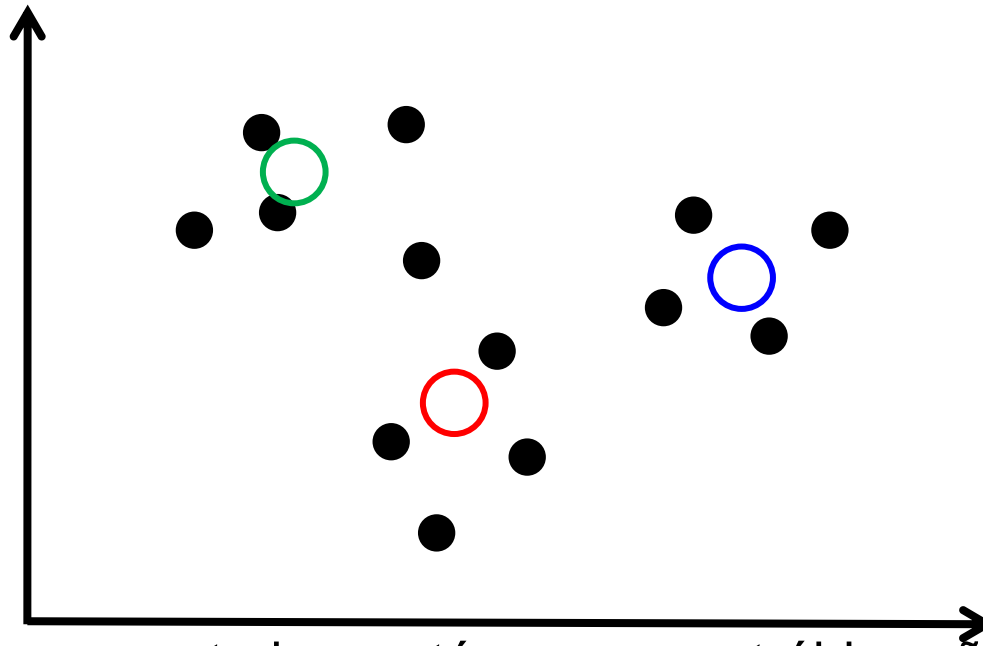


$n^{\text{a}}$  iteração

# Algoritmo K-Means

- Exemplo:

$k = 3$

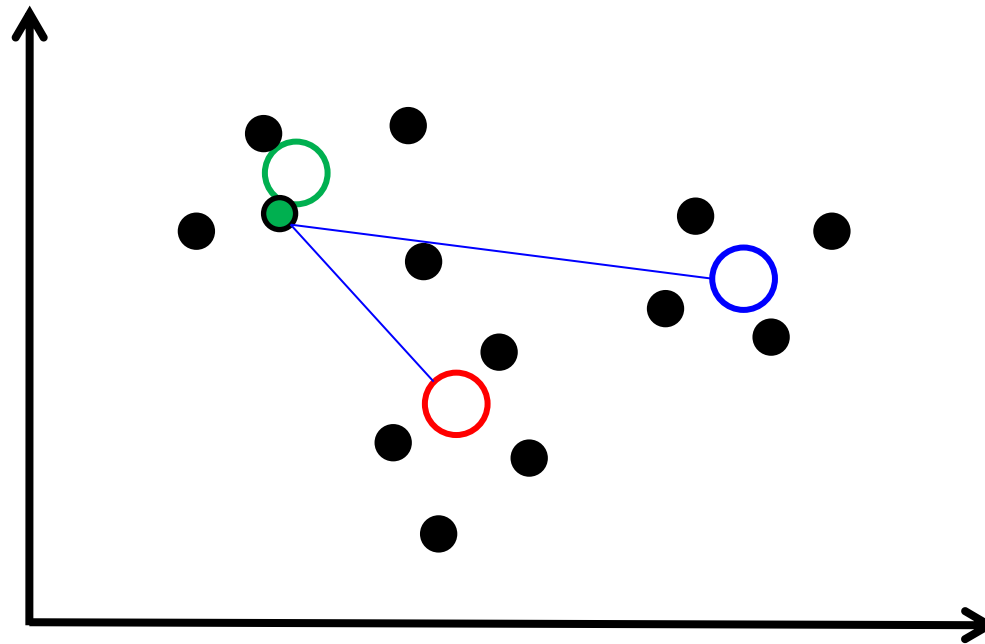


Repete-se os passos anteriores até que os centróides não se movam mais.

# Algoritmo K-Means

- Exemplo:

$k = 3$

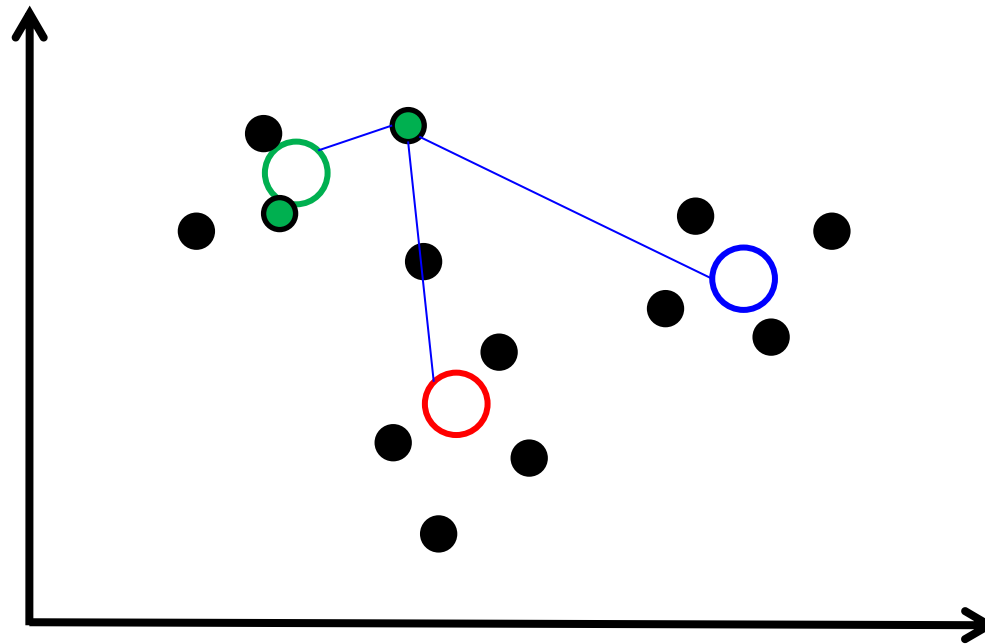


1ª iteração

# Algoritmo K-Means

- Exemplo:

$k = 3$

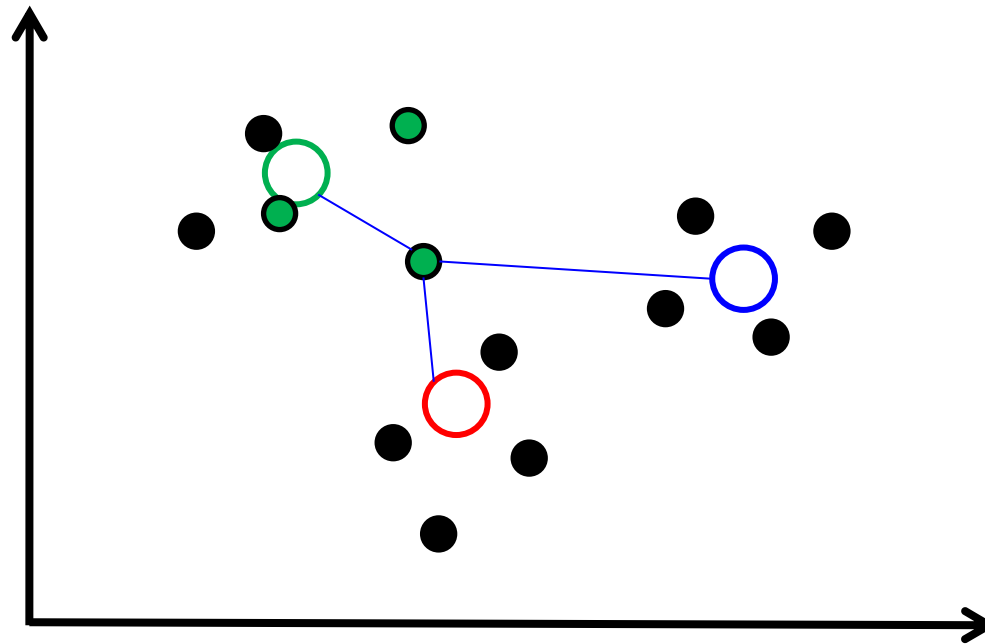


2ª iteração

# Algoritmo K-Means

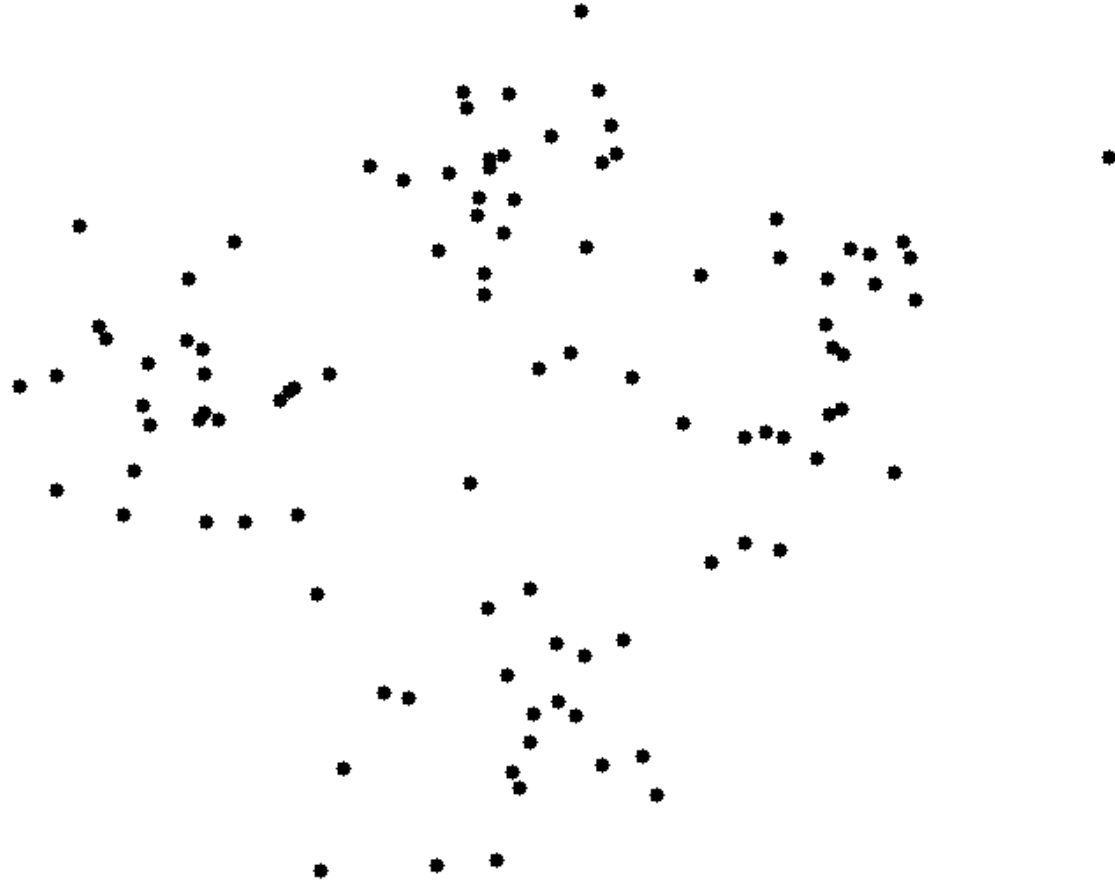
- Exemplo:

$k = 3$



3ª iteração

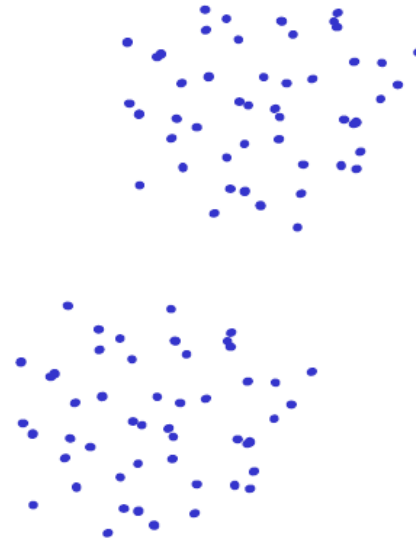
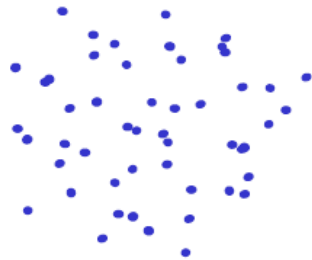
# Algoritmo K-Means





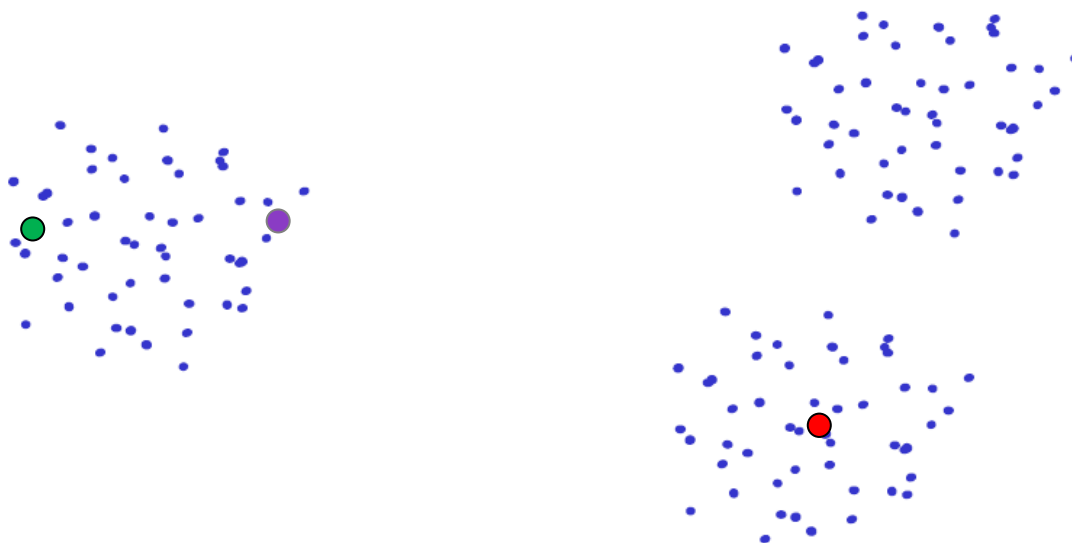
# Problemas do K-Means

- O principal problema do K-Means é a dependência de uma **boa inicialização**.



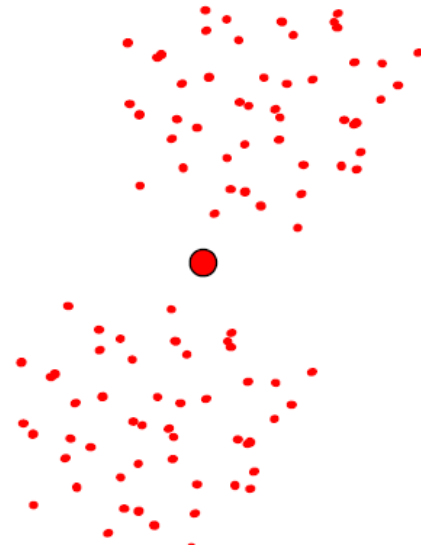
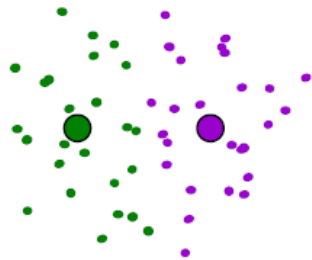
# Problemas do K-Means

- O principal problema do K-Means é a dependência de uma **boa inicialização**.

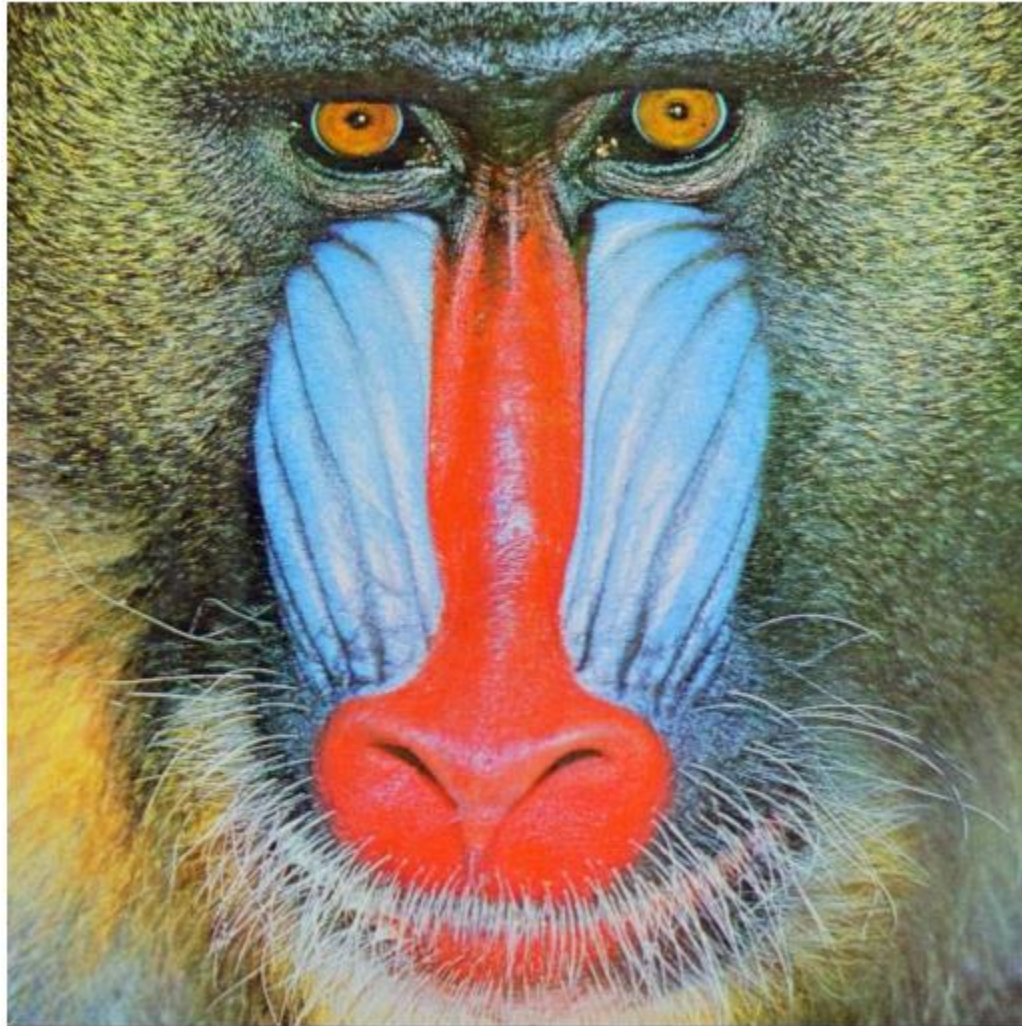


# Problemas do K-Means

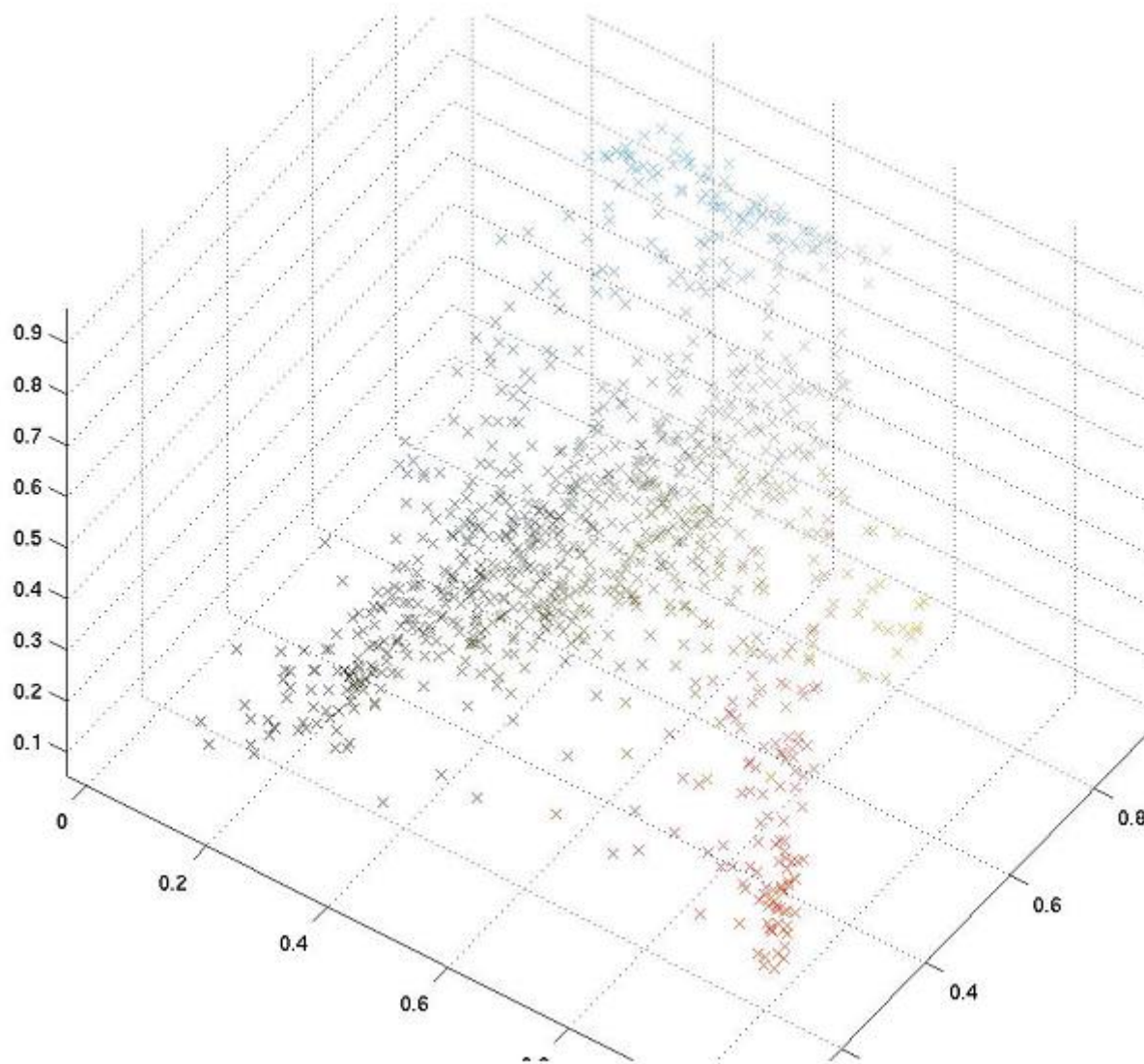
- O principal problema do K-Means é a dependência de uma **boa inicialização**.



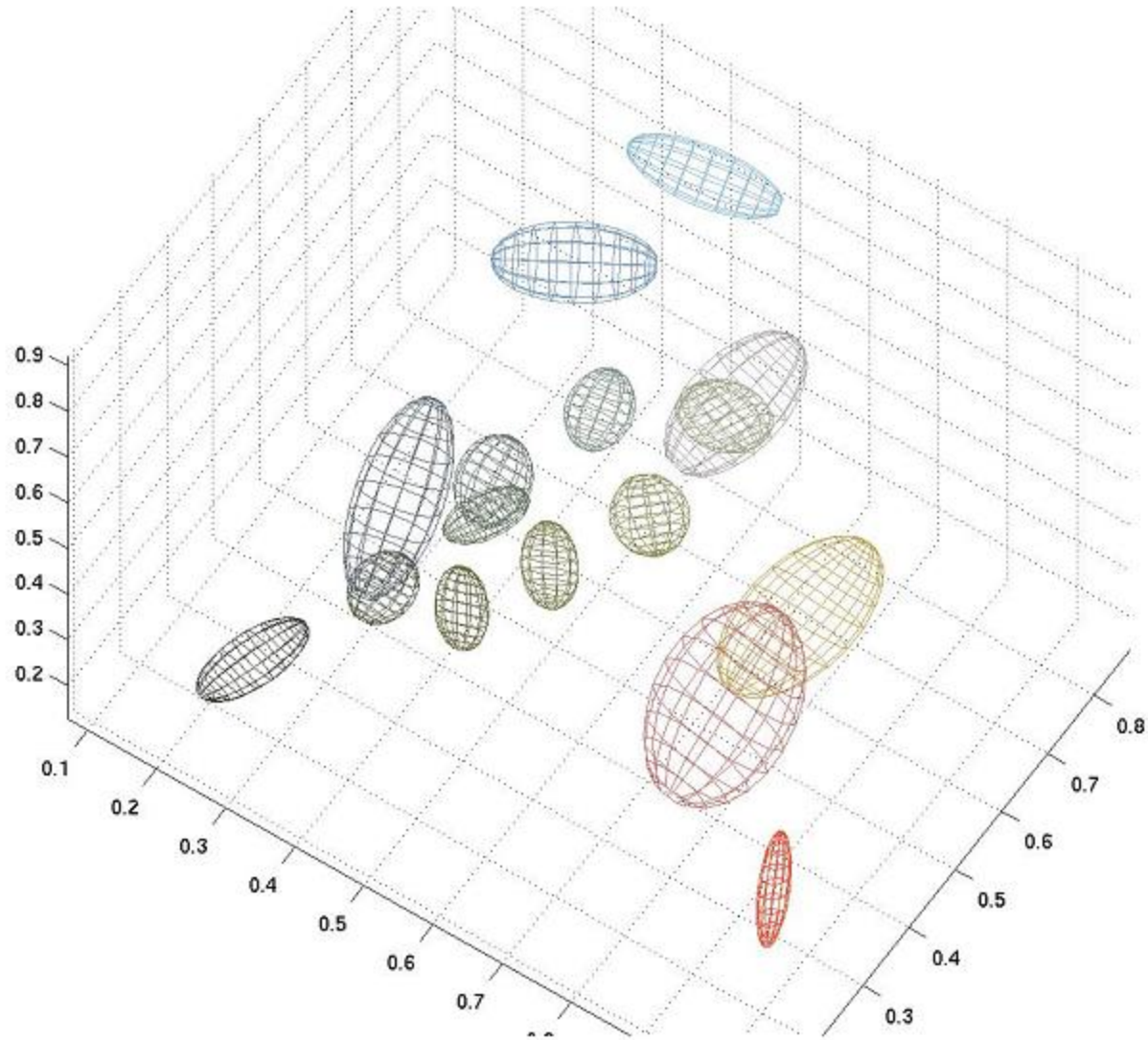
# k-Means – Um Exemplo



# k-Means – Um Exemplo

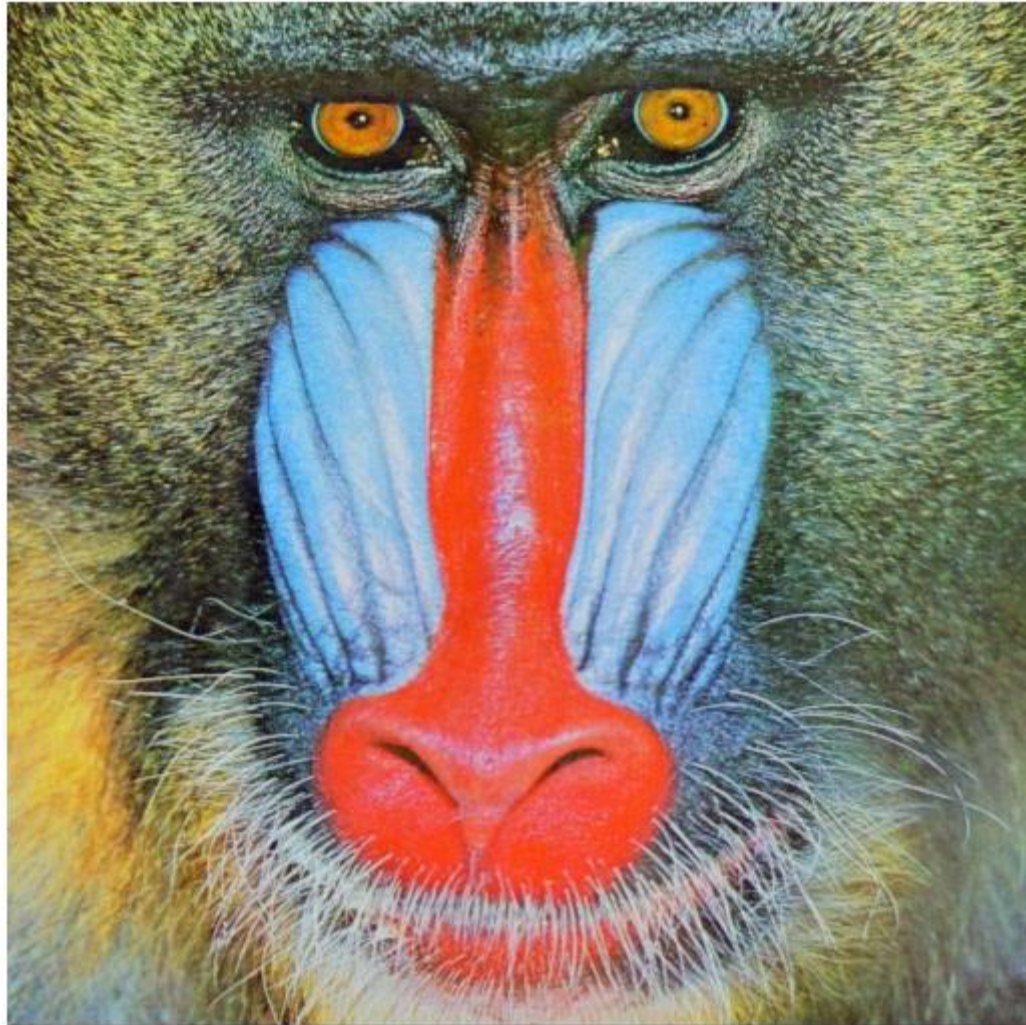


# k-Means – Um Exemplo

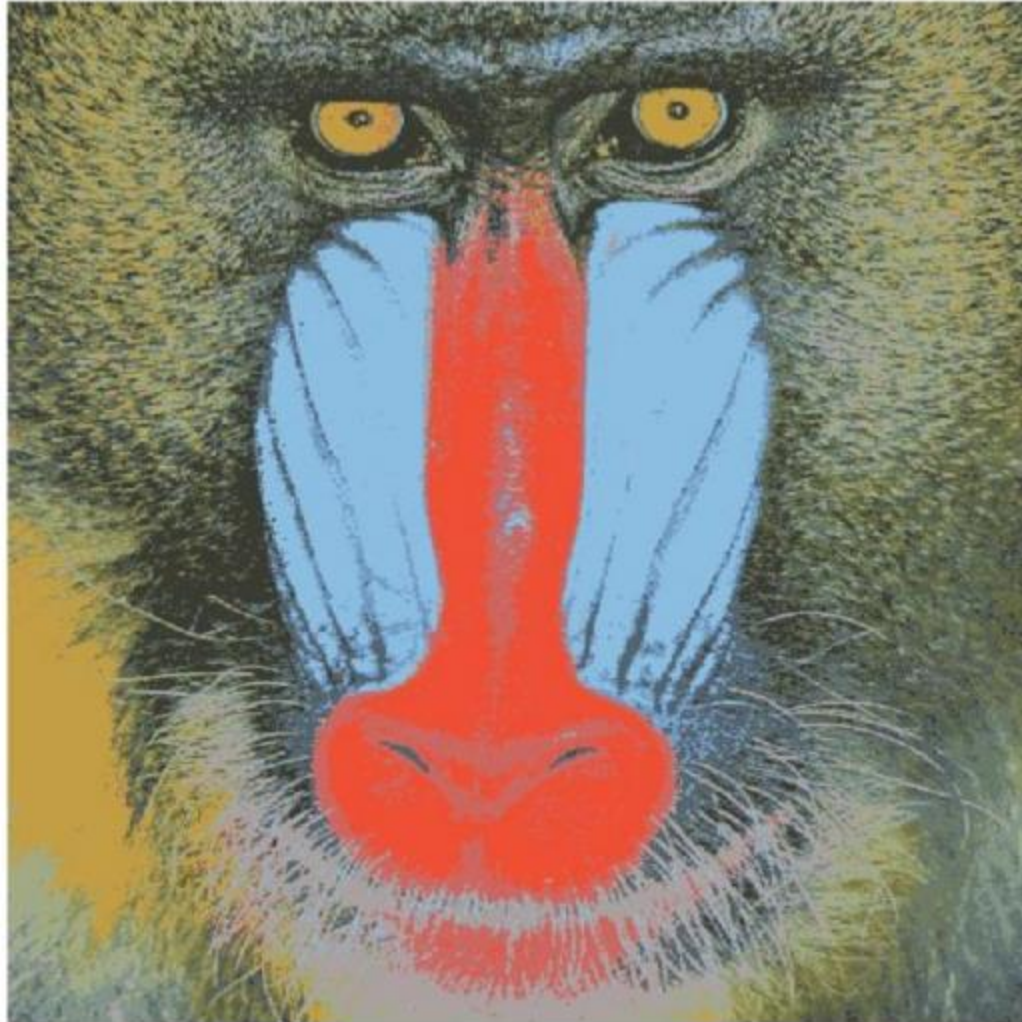




# k-Means – Um Exemplo



# k-Means – Um Exemplo





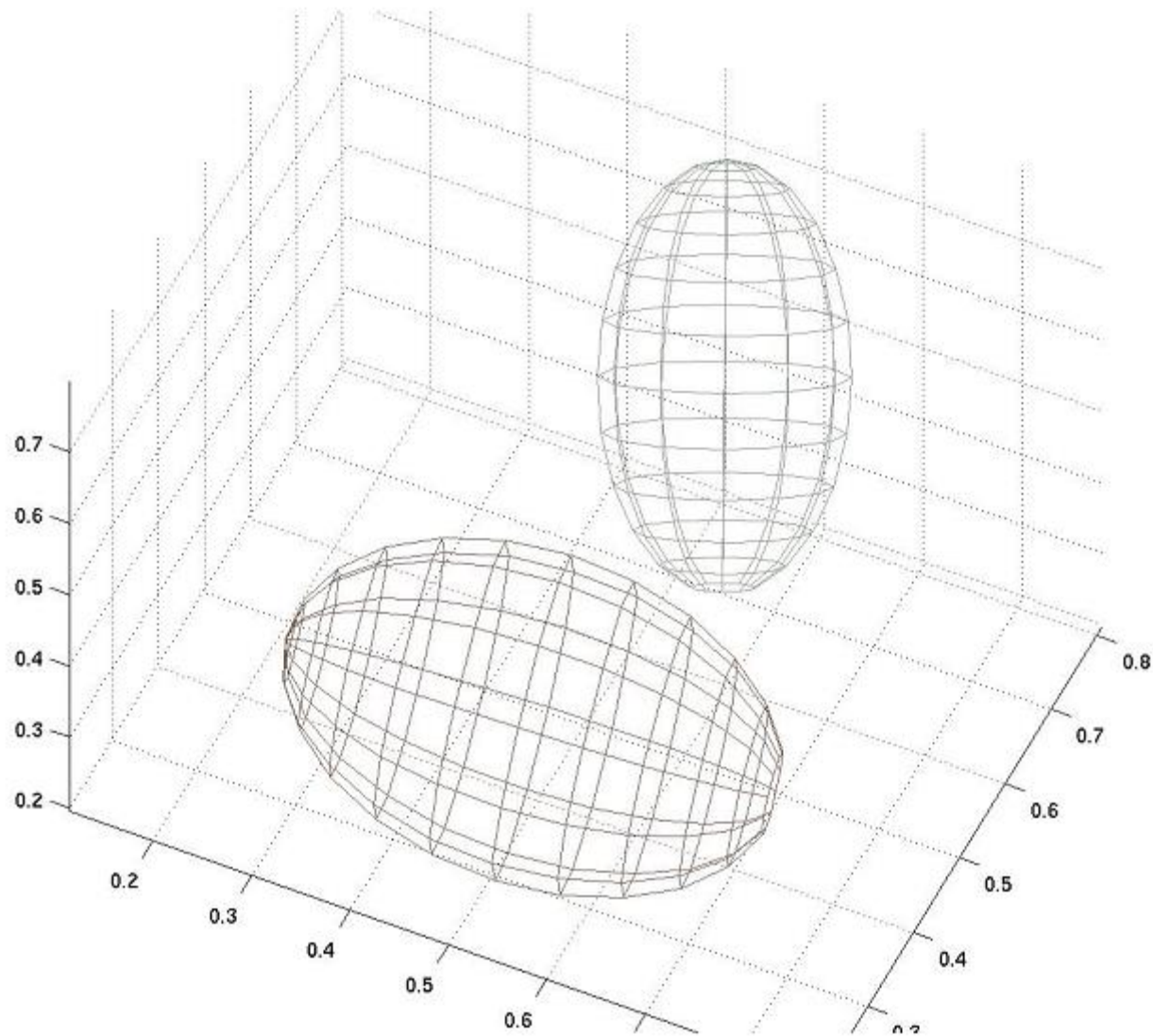
# k-Means – Um Exemplo



# k-Means – Um Exemplo



# k-Means – Um Exemplo



# k-Means – Um Exemplo

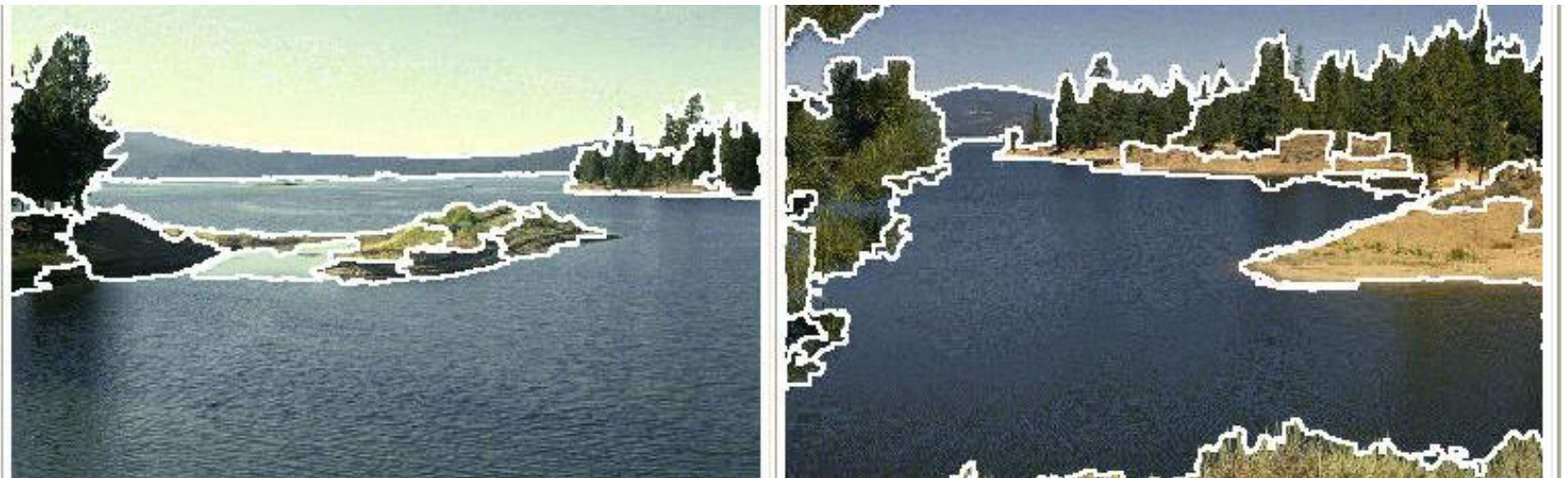




# Mean shift

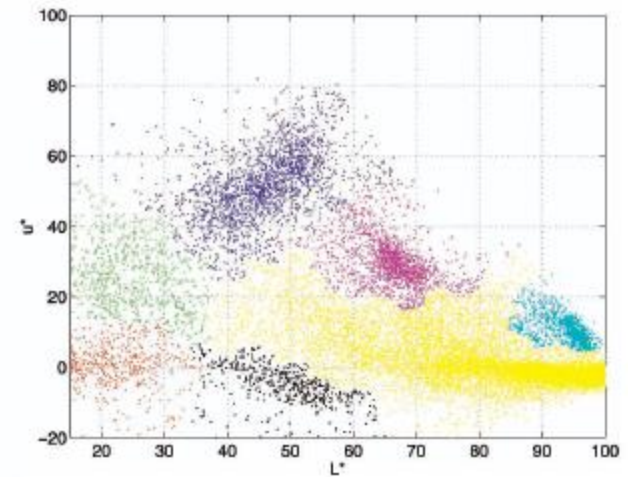
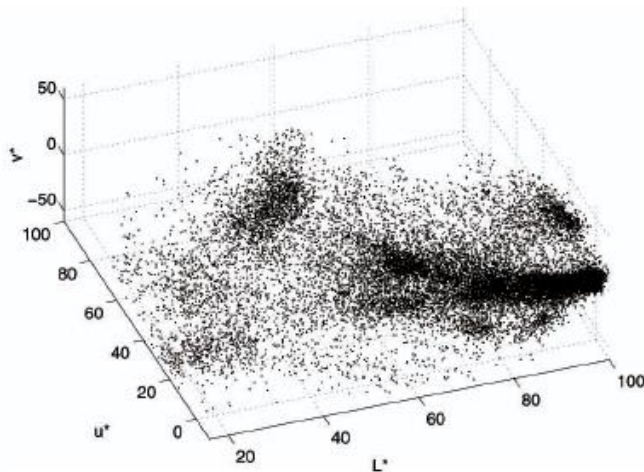
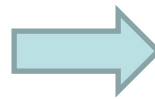
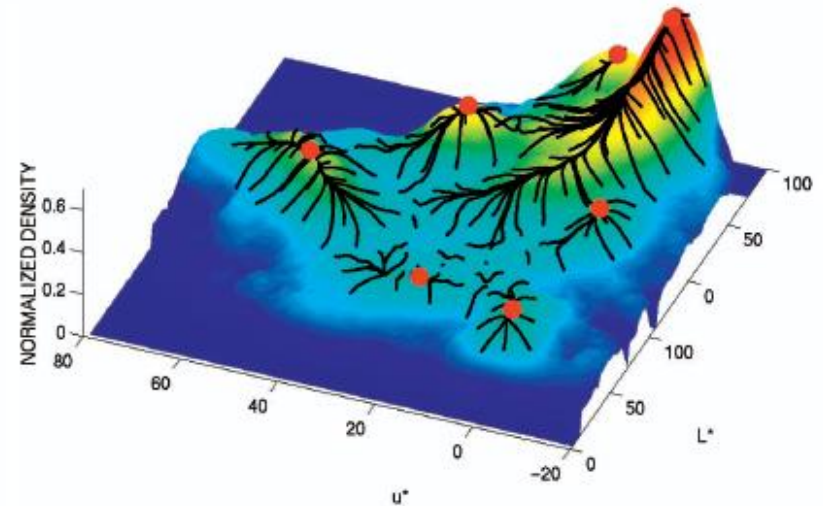
D. Comaniciu and P. Meer, Mean Shift: A Robust Approach toward Feature Space Analysis, PAMI 2002.

Técnica versátil para segmentação baseada em agrupamento

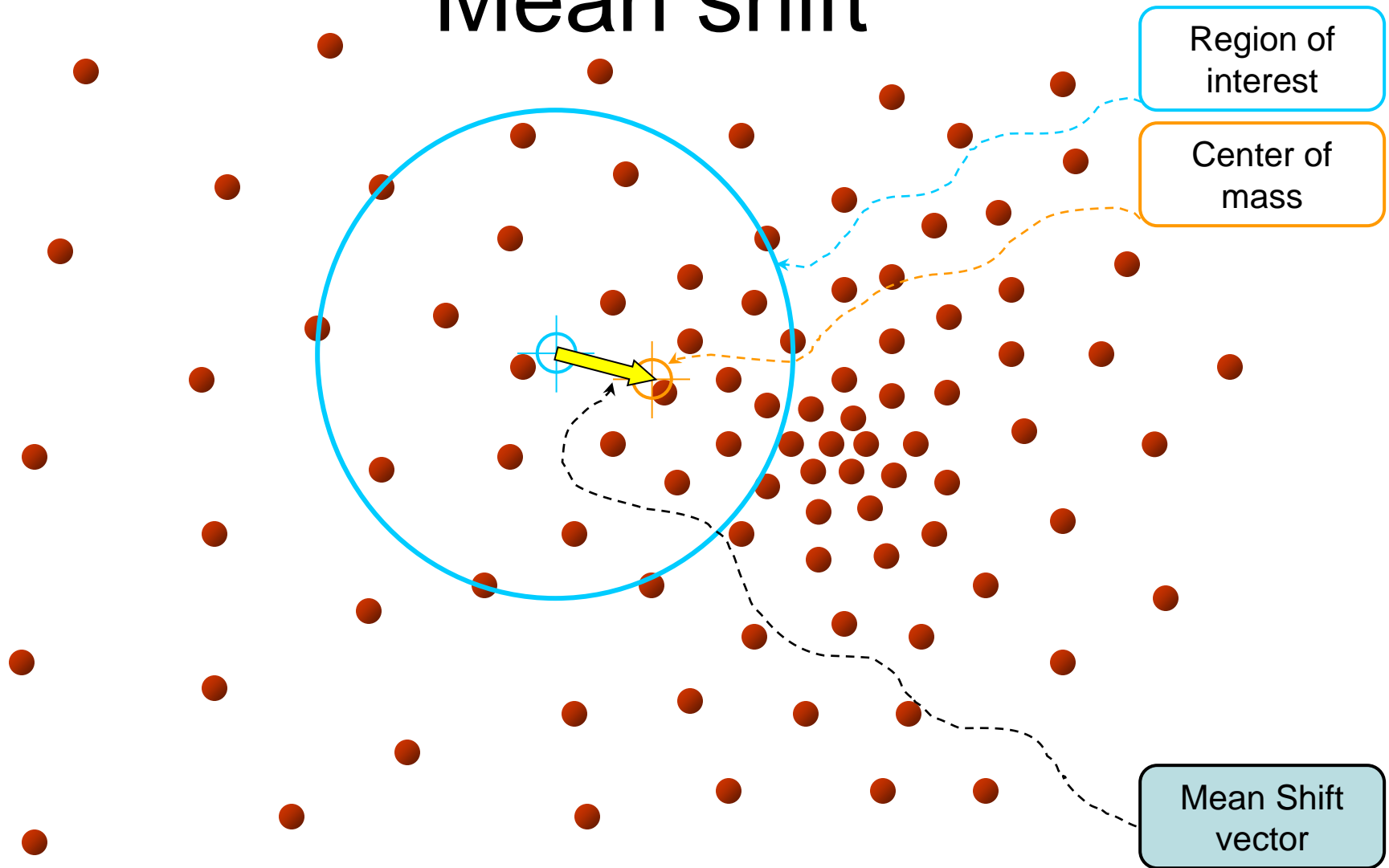


# Mean shift

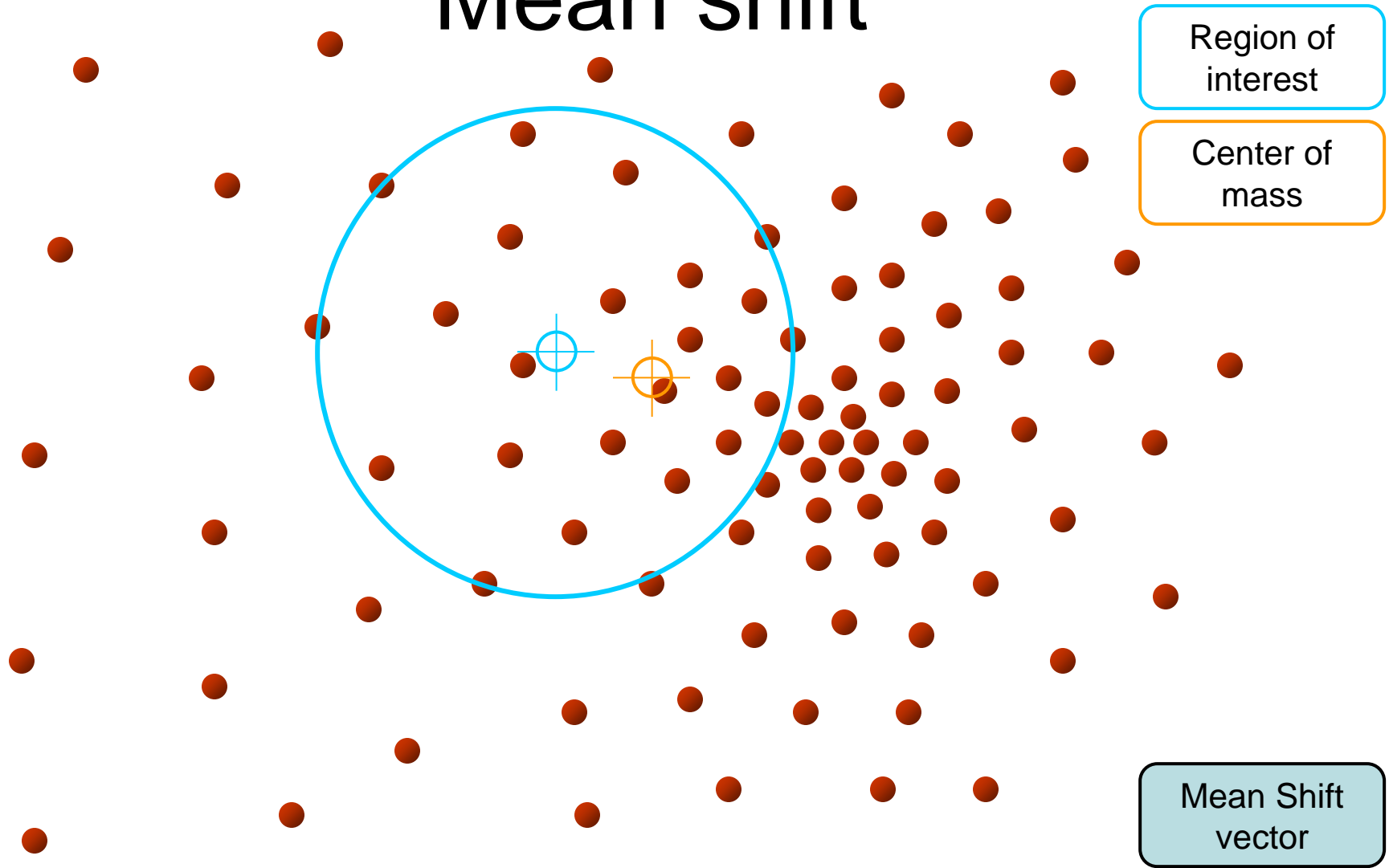
- Tente encontrar cluster não-paramétrico



# Mean shift



# Mean shift



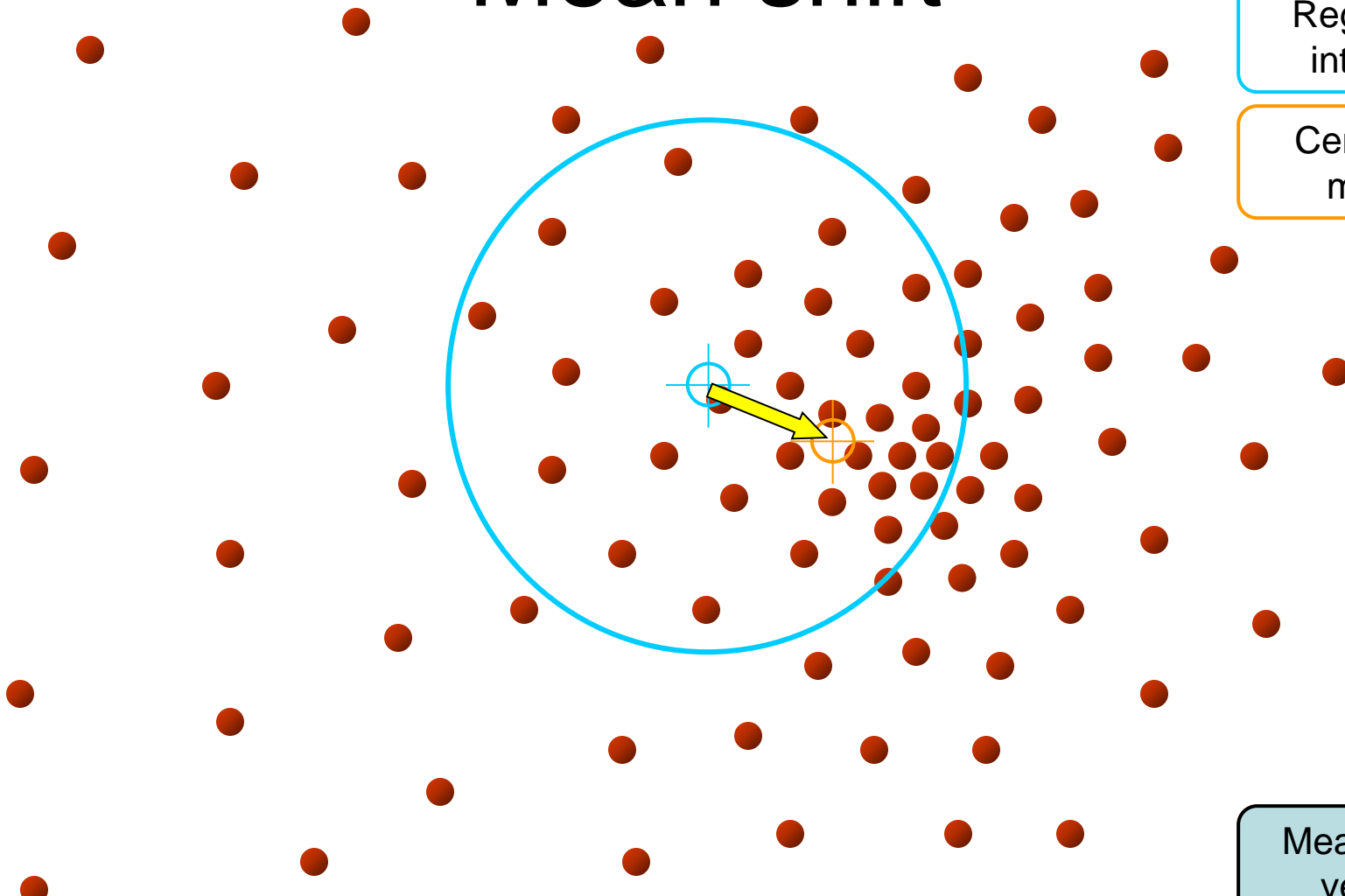


# Mean shift

Region of  
interest

Center of  
mass

Mean Shift  
vector

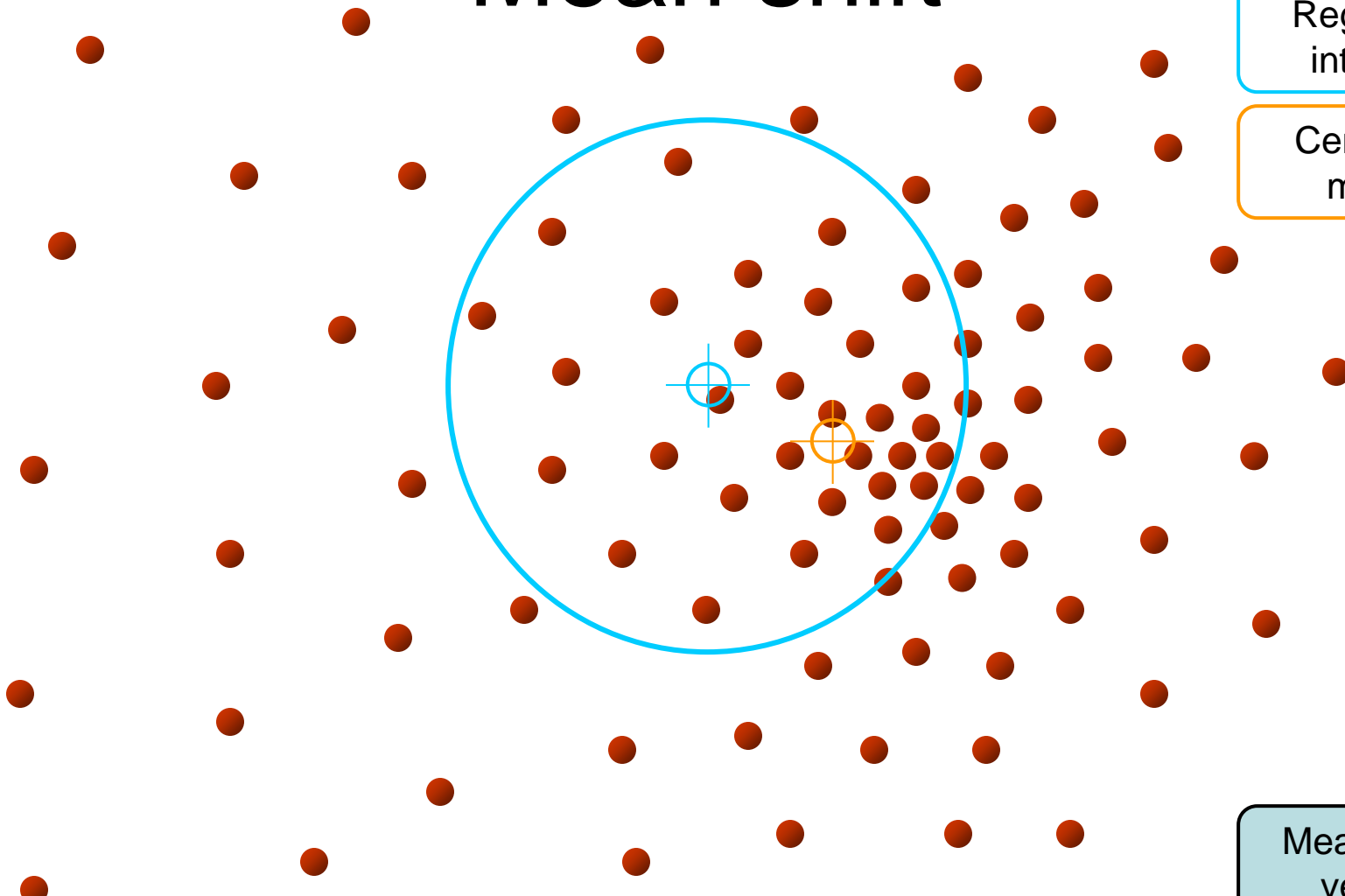


# Mean shift

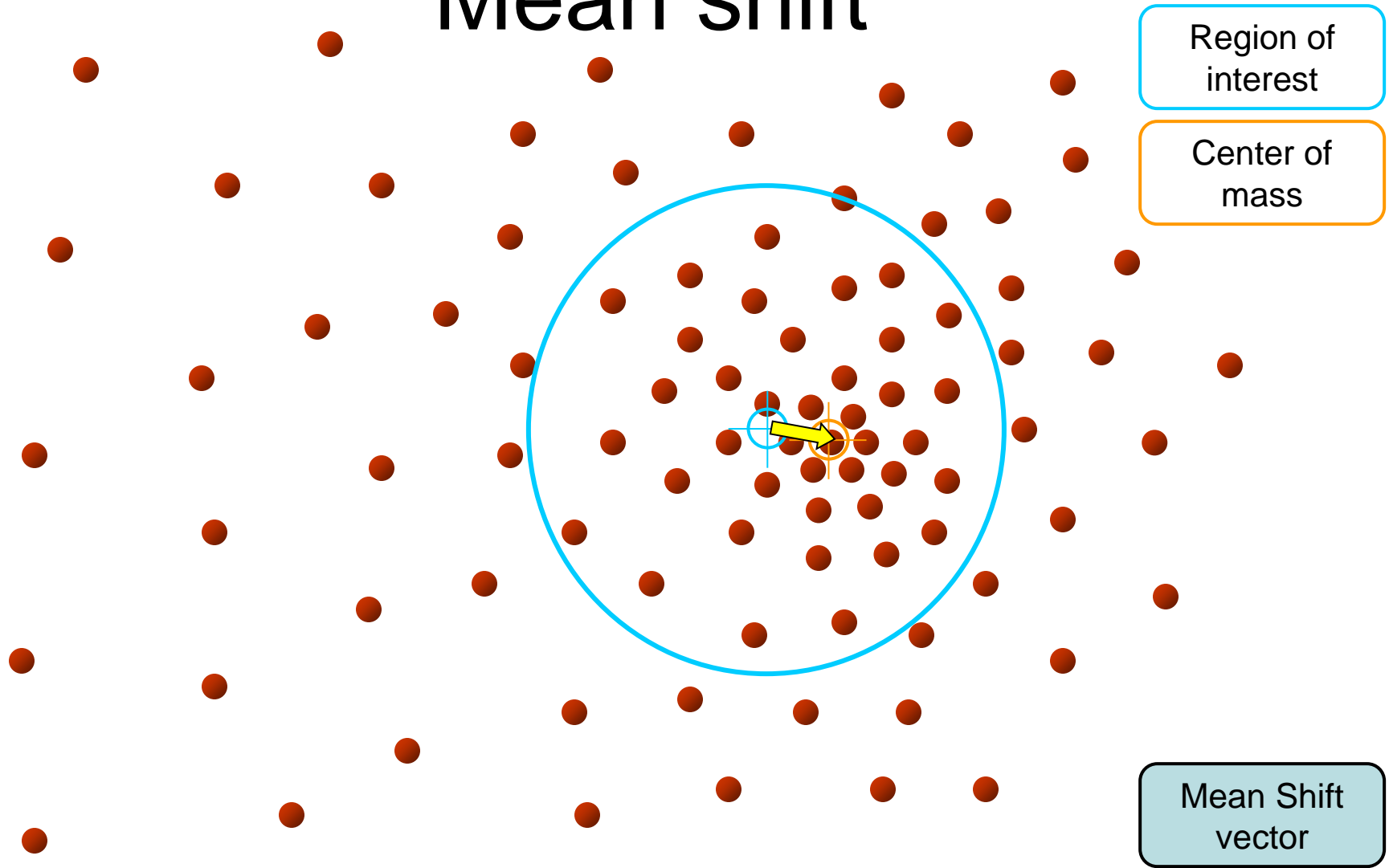
Region of  
interest

Center of  
mass

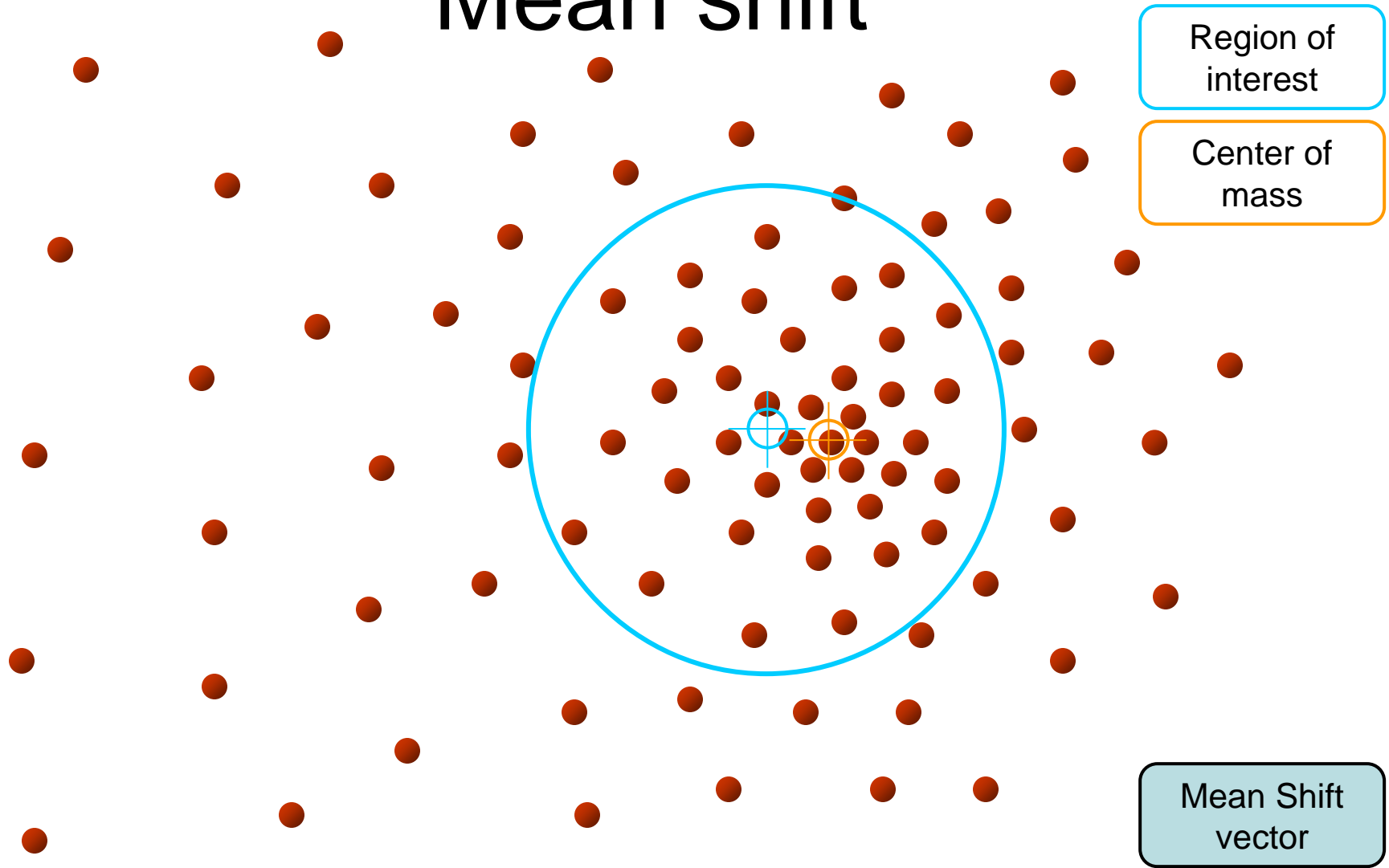
Mean Shift  
vector



# Mean shift



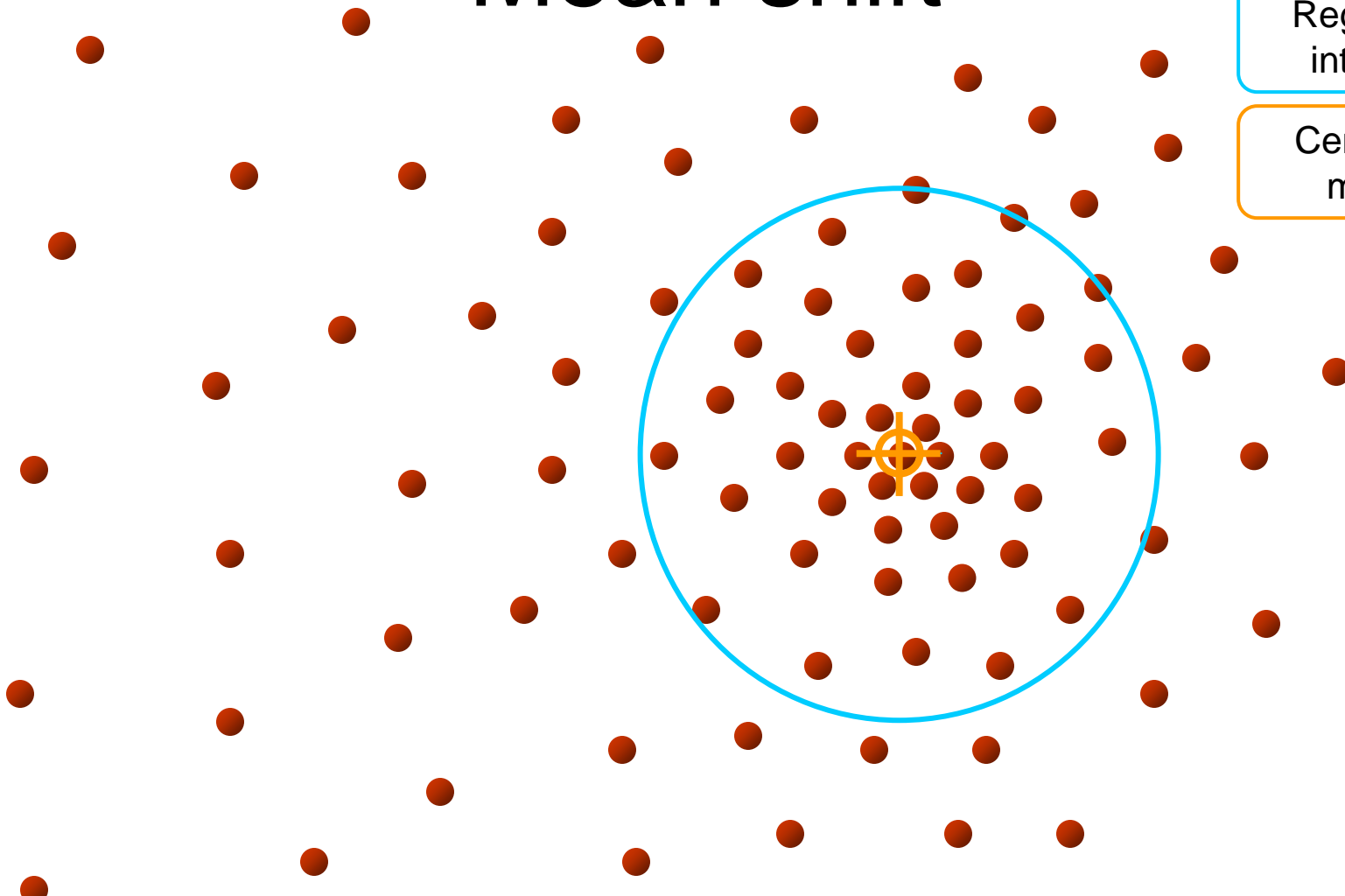
# Mean shift



# Mean shift

Region of  
interest

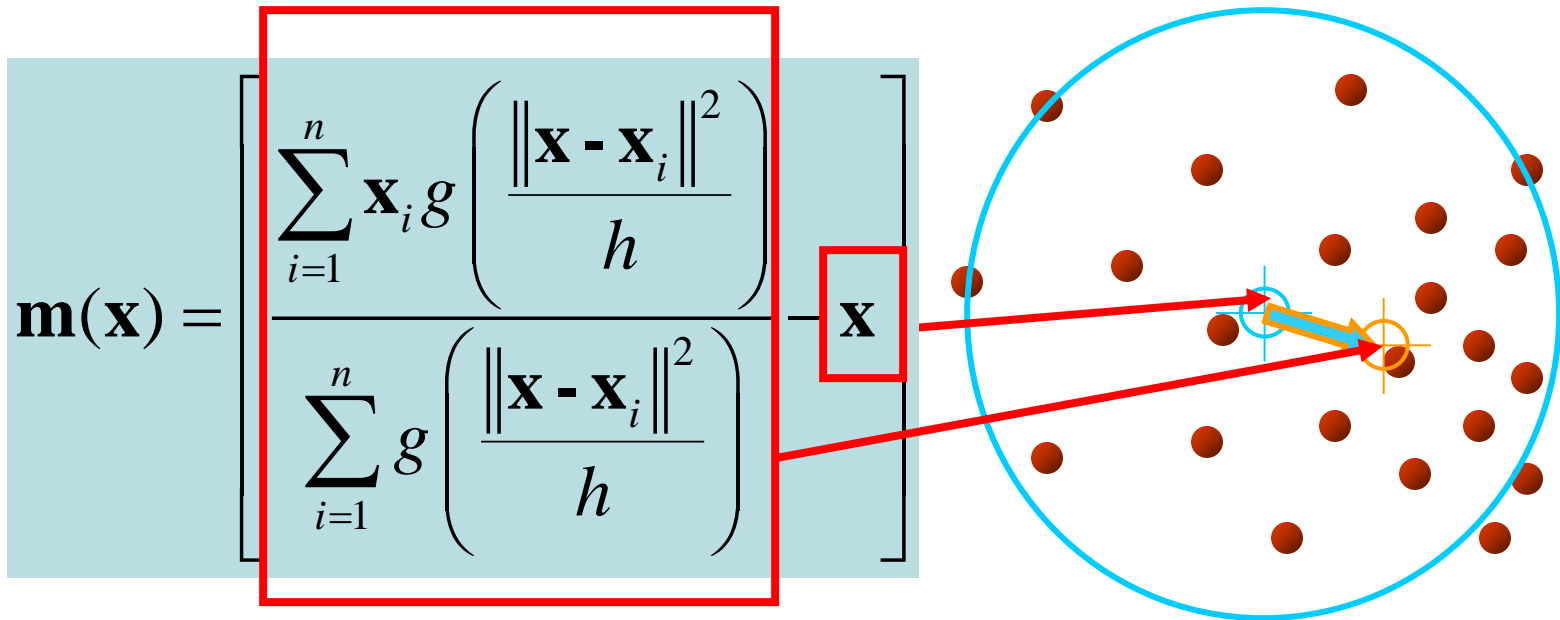
Center of  
mass



# Calcolando o Mean Shift

Simple Mean Shift procedure:

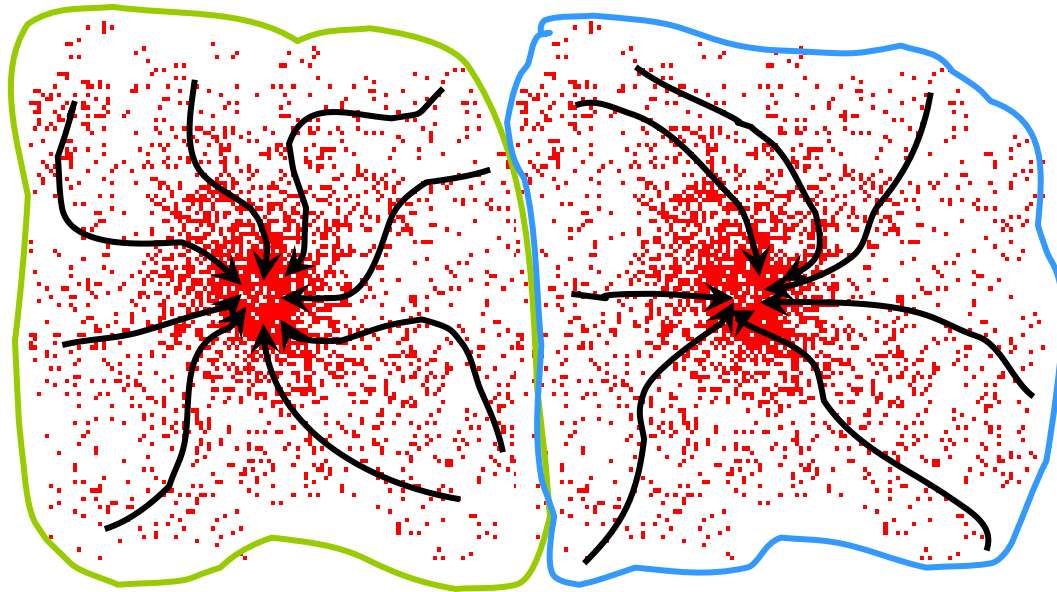
- Compute mean shift vector
- Translate the Kernel window by  $\mathbf{m}(\mathbf{x})$



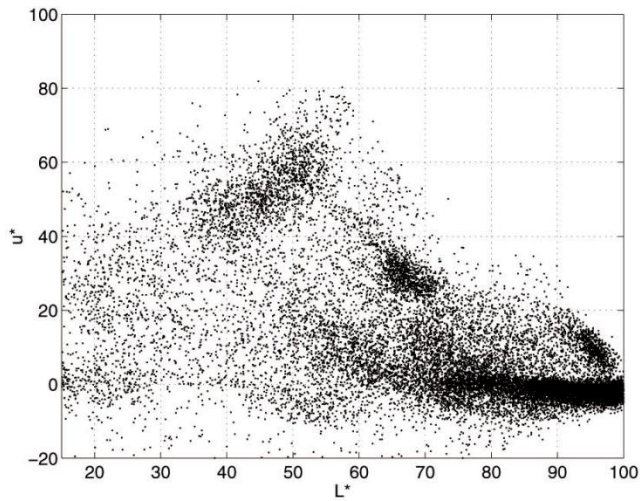
# Bacia de atração

**Bacia de atração:** a região para a qual todas as trajetórias levam ao mesmo modo

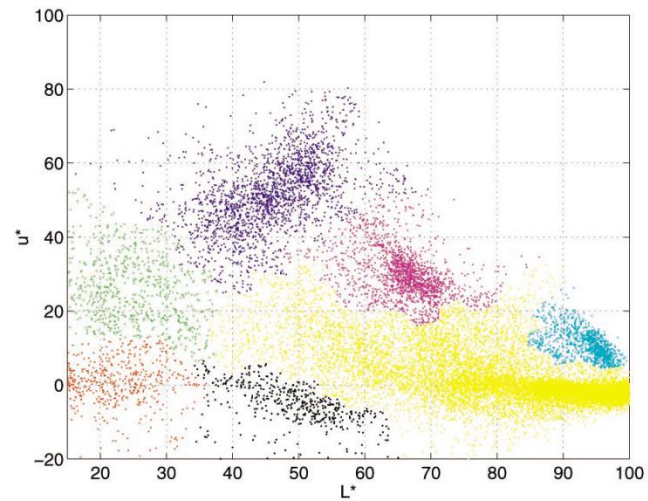
**Cluster:** todos os pontos de dados na bacia de atração de um modo



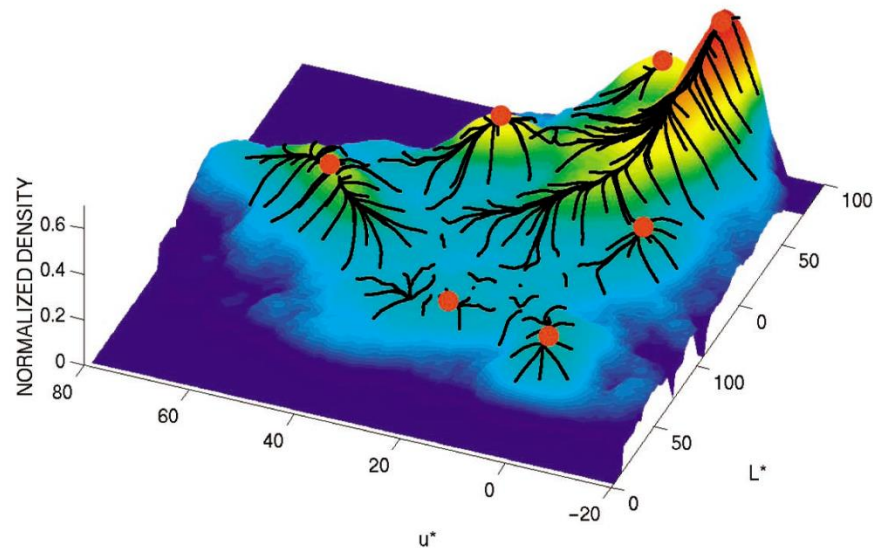
# Bacia de atração



(a)



(b)

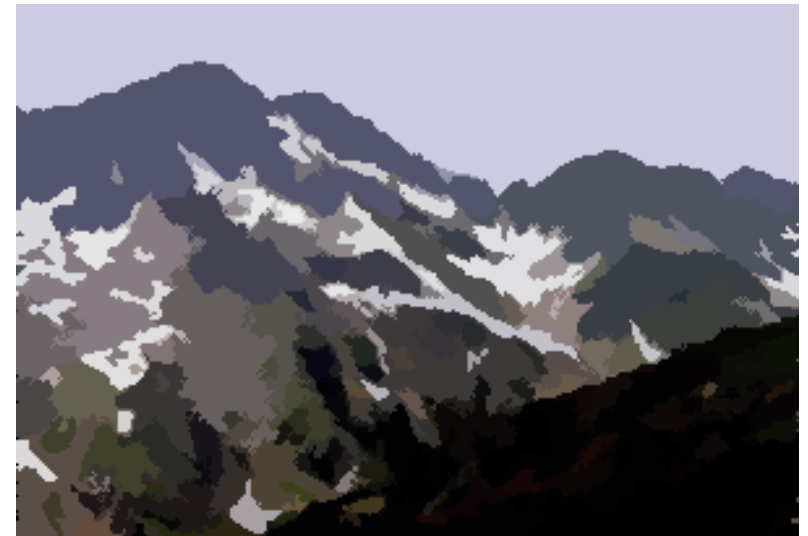




# Mean shift clustering

- The mean shift algorithm seeks *modes* of the given set of points
  1. Choose kernel and bandwidth
  2. For each point:
    - a) Center a window on that point
    - b) Compute the mean of the data in the search window
    - c) Center the search window at the new mean location
    - d) Repeat (b,c) until convergence
  3. Assign points that lead to nearby modes to the same cluster

# Mean shift



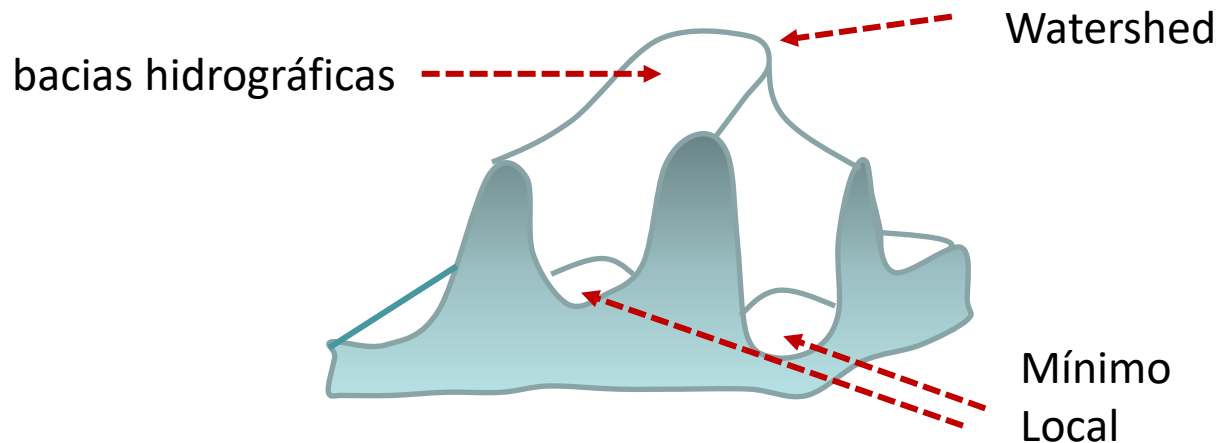
# Mean shift



# Watershed

Se uma imagem é vista como um relevo 3D

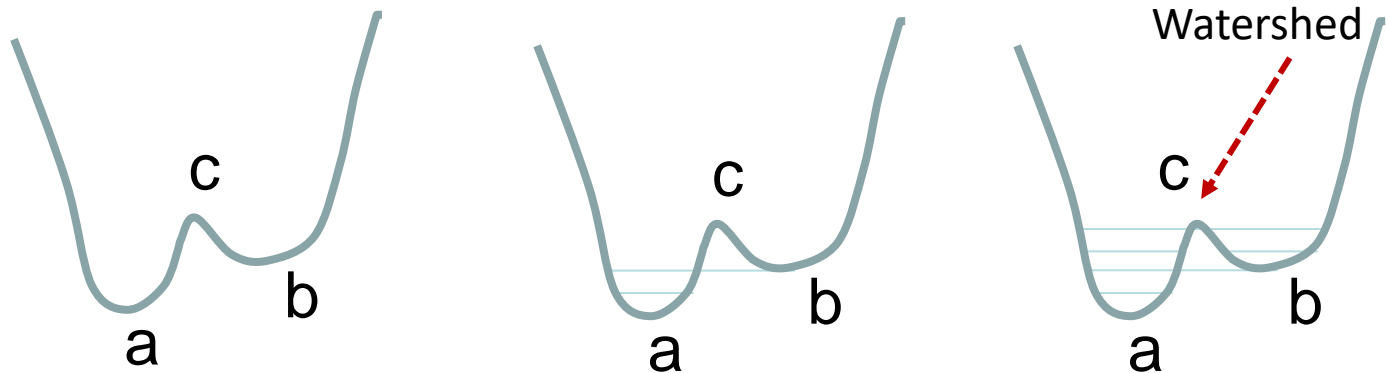
- Watershed consiste da linha dividindo duas bacias
- Uma gota caindo em uma posição do relevo encontra um único mínimo exceto se ele está em um ponto de watershed onde existem dois caminhos levando a dois mínimos distintos



# Watershed

Simulação da nascente de água a partir de um mínimo local da imagem

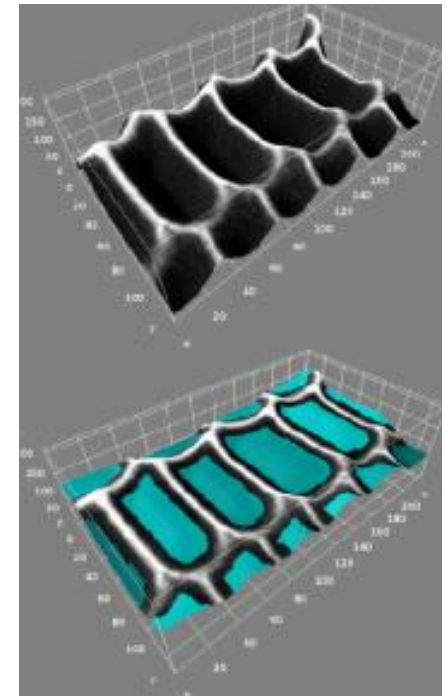
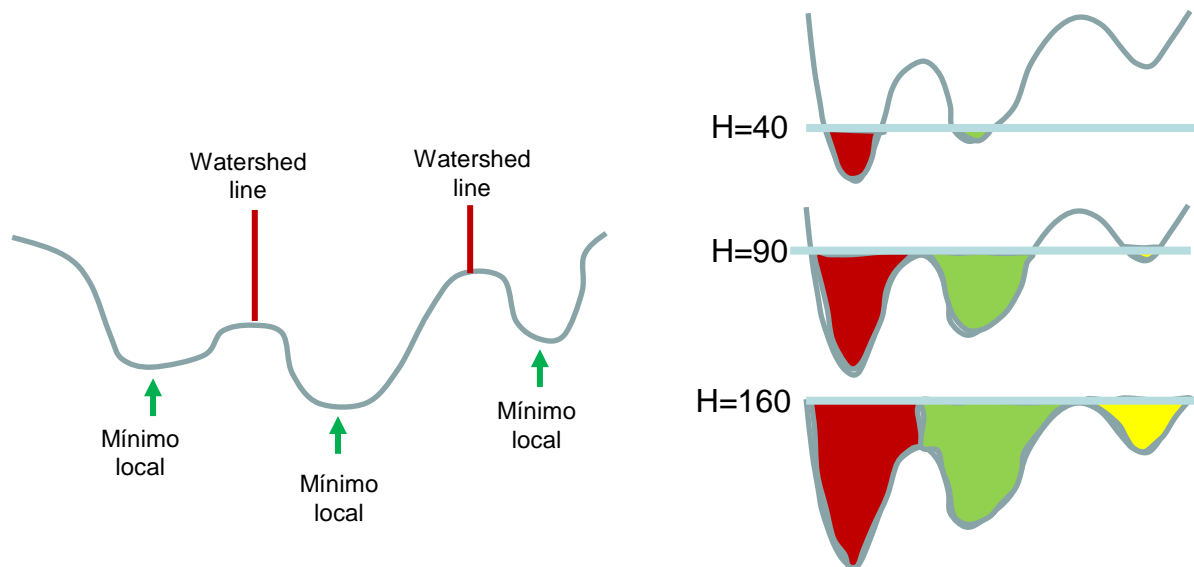
- O mínimo  $p$  é preenchido com uma velocidade uniforme
- Um ponto de watershed aparece quando **duas regiões distintas se encontram**
- O watershed é progressivamente construído em cada sequência de níveis da imagem



# Watershed

Use uma analogia topográfica. Considere os níveis de cinza como altitudes

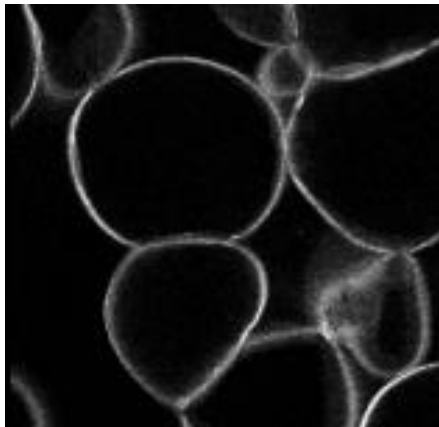
- Identificar mínimos locais
- Bacias de inundação a partir de mínimos
- Separe as bacias por um “barragem” → a água



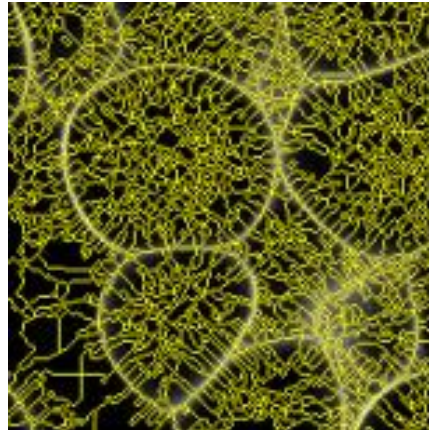


# Watershed limitações

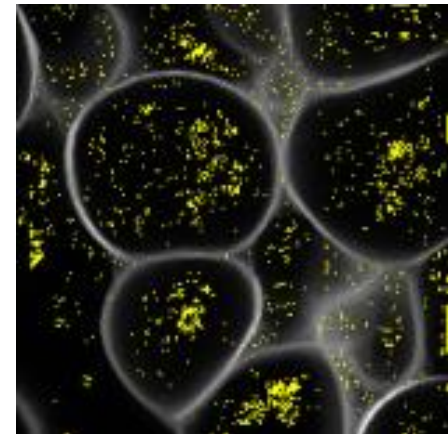
- Excesso de segmentação (muitas regiões)
  - devido à presença de muitos mínimos locais



Original



Watershed

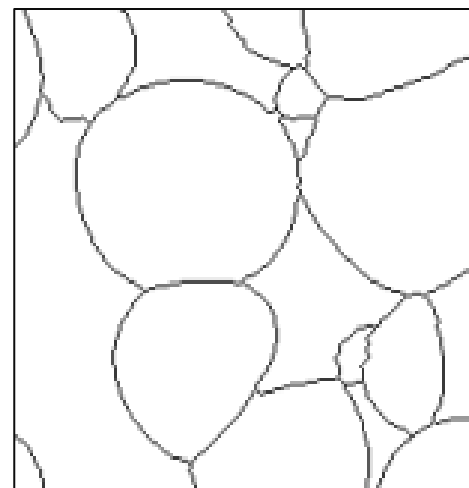
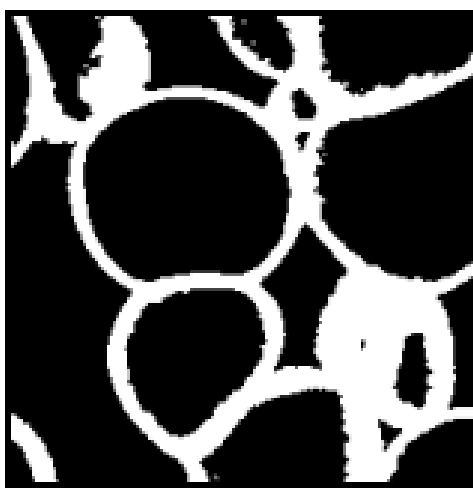
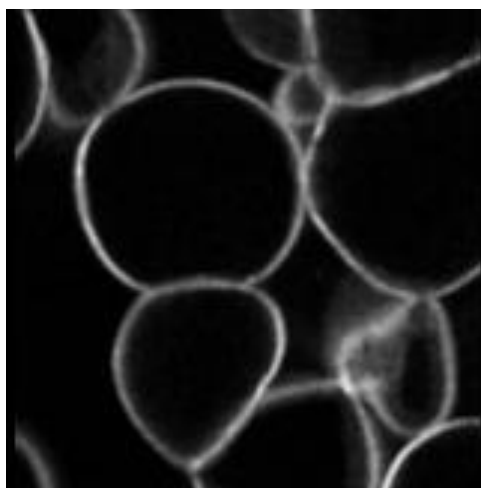


Mínimo local

# Watershed soluções

**Ideia:** remover mínimos indesejados

- Filtragem da imagem de entrada (Gaussiana, mediana...)
- Detectar mínimos automaticamente





# Técnicas modernas

