

# SEGMENTAÇÃO DE IMAGENS

**PROF. VALMIR MACÁRIO FILHO**



# SEGMENTAÇÃO

- Segmentação é o agrupamento de Pixels de uma imagem em regiões com as mesmas propriedades com respeito a uma aplicação particular
- O nível até o qual essa subdivisão deve ser realizada depende do problema sendo resolvido
- Geralmente a segmentação de imagens é uma parte vital de uma série de processos que tem como objetivo o “entendimento” da imagem (Visão Computacional)

# SEGMENTAÇÃO





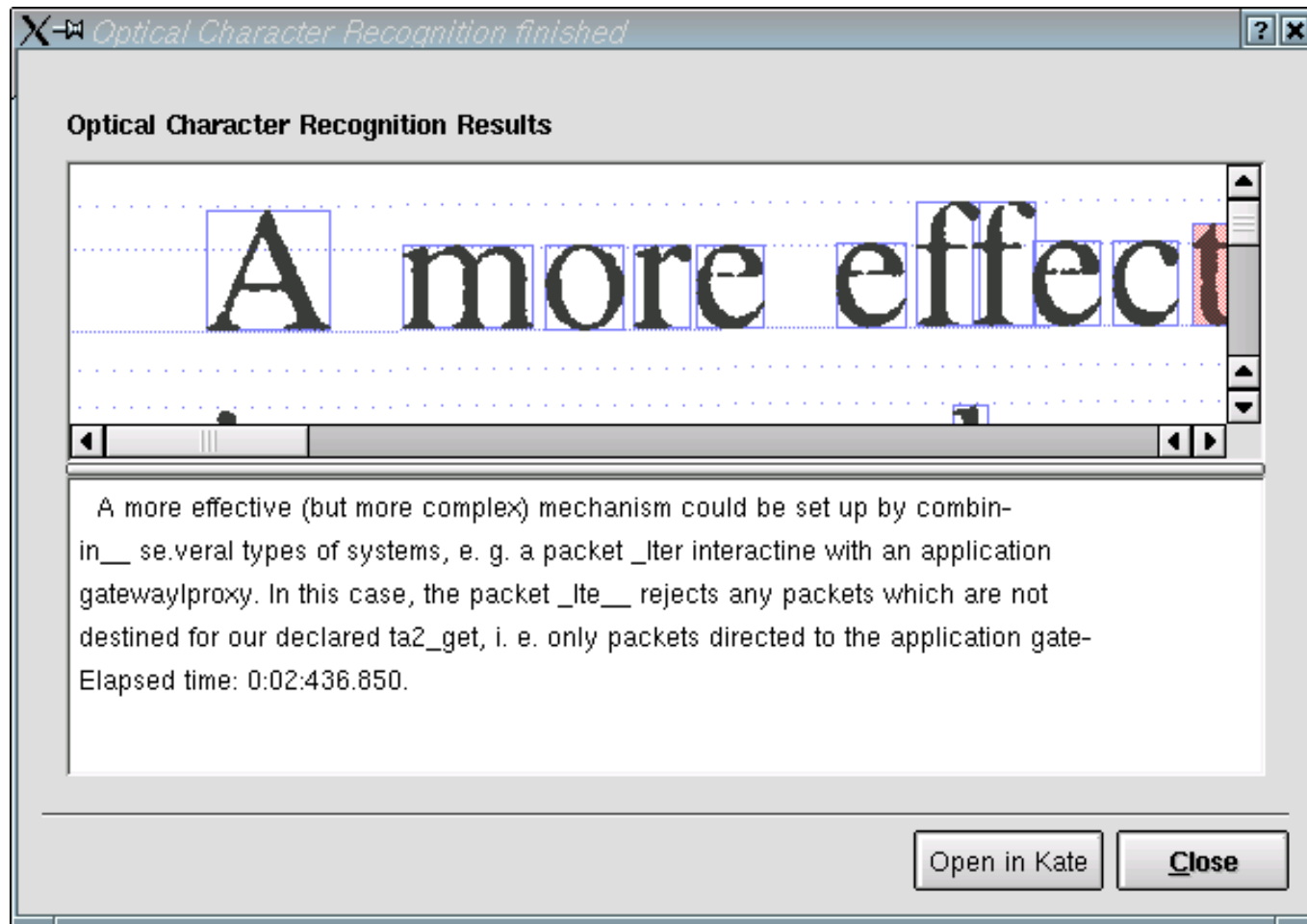
# SEGMENTAÇÃO



# SEGMENTAÇÃO PARA RECONHECIMENTO DE OBJETOS

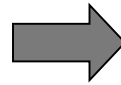


# SEGMENTAÇÃO PRA RECONHECIMENTO DE CARACTERES

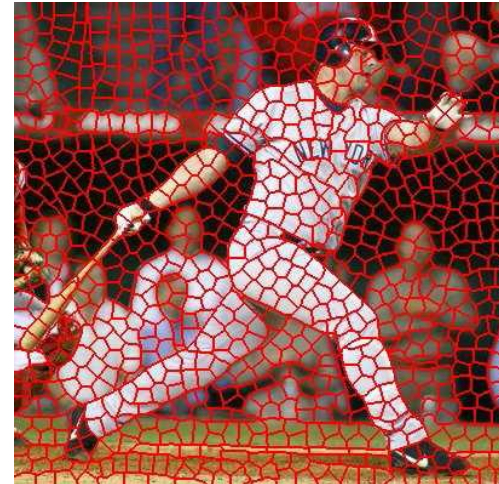
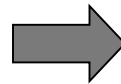




# SEGMENTAÇÃO POR EFICIÊNCIA



[Felzenszwalb and Huttenlocher 2004]



[Shi and Malik 2001]

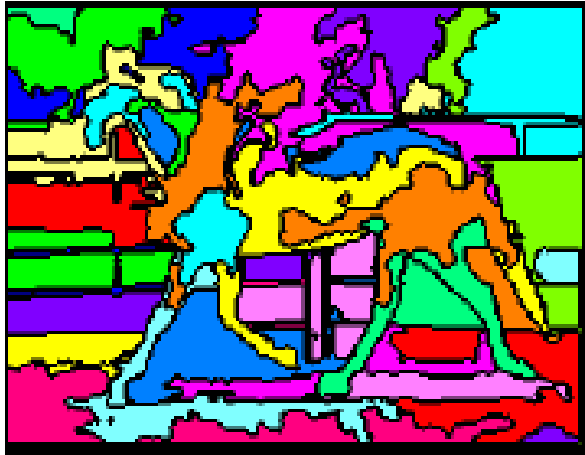
[Hoiem et al. 2005, Mori 2005]

# SEGMENTAÇÃO POR RESULTADO

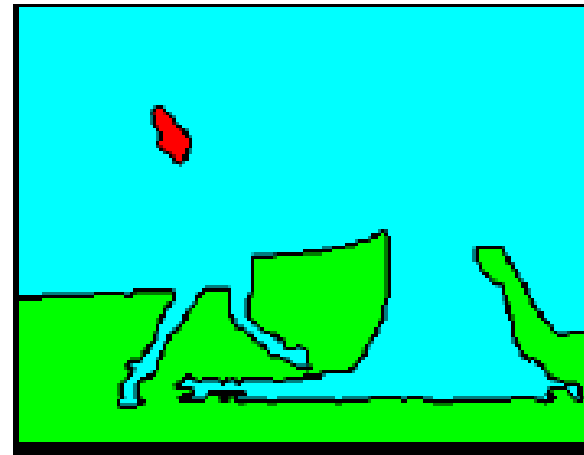




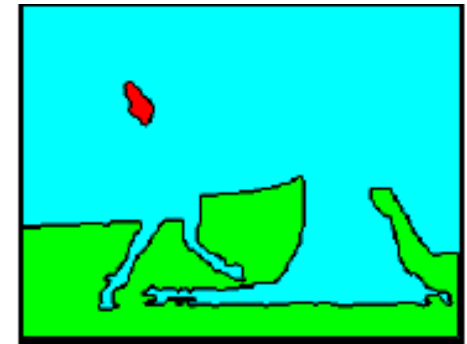
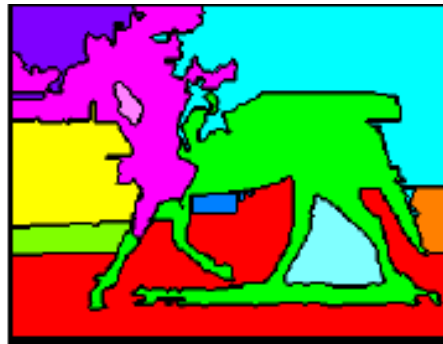
# TIPOS DE SEGMENTAÇÃO



Super Segmentação



Sub Segmentação



Segmentação Múltipla

# SEGMENTAÇÃO

- **Pixels possuem propriedades em comum**
  - **Nível de cinza**
  - **Textura**
  - **Cor**
  - **Profundidade**
  - **Movimento**
  - **etc...**
- **Região homogênea**
- **Existe uma fronteira (subconjunto) cercada por bordas ou segmento**

# OBJETIVOS

- **Expor alguma informação contida na imagem:**
  - Partes da um foto
  - Tamanho ou forma de um objeto
  - Superfícies
  - Letras
  - Padrões
  - Silhuetas
  - Movimento em vídeos
  - Objetos com profundidades diferentes
  - Etc..



# **APLICAÇÕES IMPORTANTES**

- **Localização de locais em imagens de satélite**
- **Reconhecimento de caracteres**
- **Reconhecimento de faces**
- **Realidade Aumentada**
- **Imagens Médicas**
- **Leitura de impressões digitais**
- **Buscas em bancos de dados de imagem**

# **DIFICULDADES INERENTES**

- **Segmentação de forma autônoma.**
- **Controle de luminosidade (existência de sombras próximas à borda).**
- **Regiões não homogêneas (níveis de cinza e textura).**
- **Bordas das regiões irregulares imprecisas.**
- **Precisão depende da qualidade da distinção entre os diferentes elementos de imagem.**
- **Determinação da melhor estratégia**

# TÉCNICAS DE SEGMENTAÇÃO

- **Baseada em histograma**
    - Limiarização (Simples ou Multinível)
    - Limiarização Automática
  - **Descontinuidade (Detecção de bordas)**
    - Mudanças bruscas nos níveis de cinza
    - Linhas e bordas na imagem
  - **Similaridade**
    - Baseada em regiões
    - Subdivisão das imagens e reunião das regiões homogêneas
- ❖ Já vistos
- ❖ Interesse dessa aula



# TÉCNICAS DE SEGMENTAÇÃO

- **Transformação para outro espaço vetorial**
  - imagem  $\rightarrow$  espaço Euclidiano
  - executa uma transformação linear
  - Exemplos:
    - Histograma
    - Limiarização
- **Processa a imagem**
  - Executa a transformação inversa
  - Utilização da imagem no próprio domínio
  - Diretamente nos pixels da 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 REGIÃO

- **Crescimento de regiões**
  - Conexão simples ou híbrida
  - Divisão e Fusão (Quadtree)
  - Transformada Watershed
- **Clustering**
  - Mean shift
  - K-means
  - Fuzzy C-Means
- **Baseado em Grafos**
  - Graph cut



# CRESCIMENTO DE REGIÕES

## Agregação de pixels

- Agrupa pixels ou sub-regiões em regiões maiores
- Os pixels a serem agrupados devem ter propriedades similares. (nível de cinza, textura, cor, etc...)
- 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

# CRESCIMENTO DE 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

# **CRESCIMENTO DE REGIÕES**

**Problemas com a Técnica:**

**1) Seleção das sementes: depende da natureza do problema.**

**Ex: em aplicações militares com imagens com infravermelho, os pontos mais quentes, logo, mais brilhantes, são de interesse**

**2) Seleção das Propriedades que estabeleçam os critérios de similaridade: depende do tipo de dados disponíveis.**

**Ex: as imagens de satélite usam a informação de cor.**

# **CRESCIMENTO DE REGIÕES**

## **3) Utilização de conectividade e adjacência:**

**Ex: uma imagem formada por um arranjo aleatório de 3 intensidades diferentes. Se a conexão entre pixels não for levada em conta, o resultado da segmentação não terá nenhum significado.**

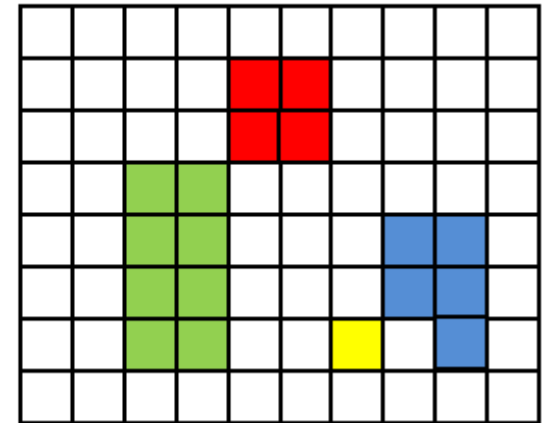
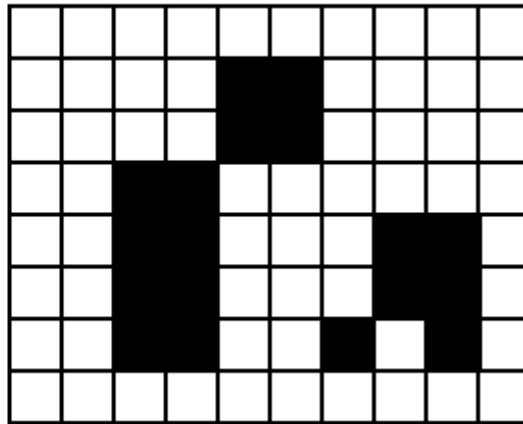
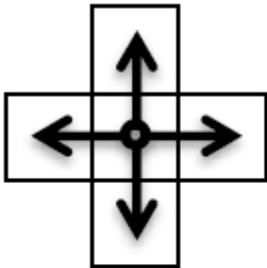
**4) Formulação de uma regra de parada: utilização de critérios de tamanho, semelhança entre um pixel candidato e os pixels da Região e Formato de uma dada Região.**

# COMPONENTES CONEXOS

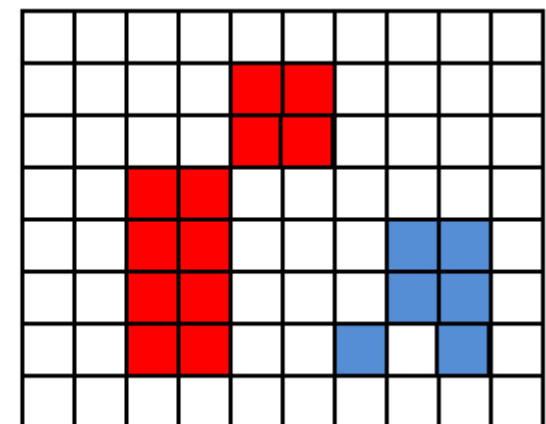
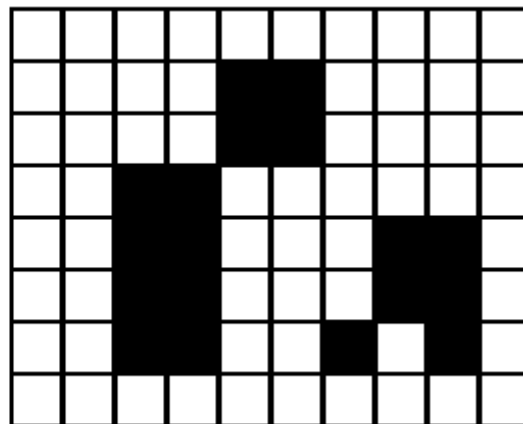
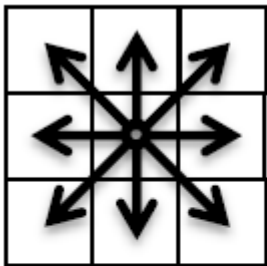
- **Algoritmo:** Dada uma imagem binária, retorna uma imagem com etiquetas numéricas, de maneira que todos os pixels que pertencem a uma mesma região estejam com a mesma etiqueta.
- **Dois passos: passar por cada pixel da imagem da esquerda para a direita e de cima para baixo**
  - Passo 1: etiquetar cada pixel da imagem de acordo com sua conectividade e guardar possíveis equivalências.
  - Passo 2: resolver todas as equivalências detectadas e escolher um rótulo para cada equivalência, no segundo passo, resolver e atribuir o rótulo selecionado.

# CONECTIVIDADE

vizinhança 4-conectada



vizinhança 8-conectada





# ALGORITMO DE ETIQUETAGEM

- **Caso 1: (Nenhum vizinho igual a 1)**



- **Caso 2: (Apenas um vizinho igual a 1)**



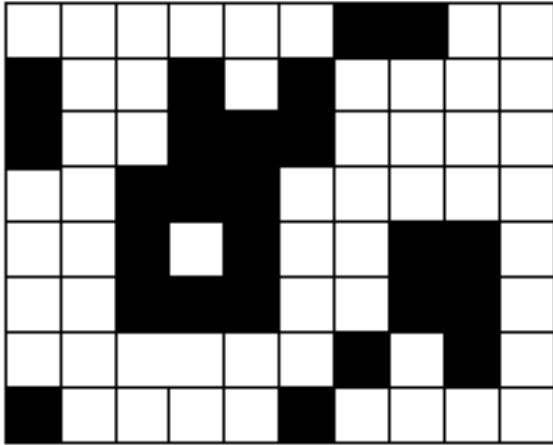
- **Caso 3: (Mais de um vizinho igual a 1)**



Guardar E1=E2

# APLICAÇÃO DO ALGORITMO VIZINHANÇA 4-CONECTADA

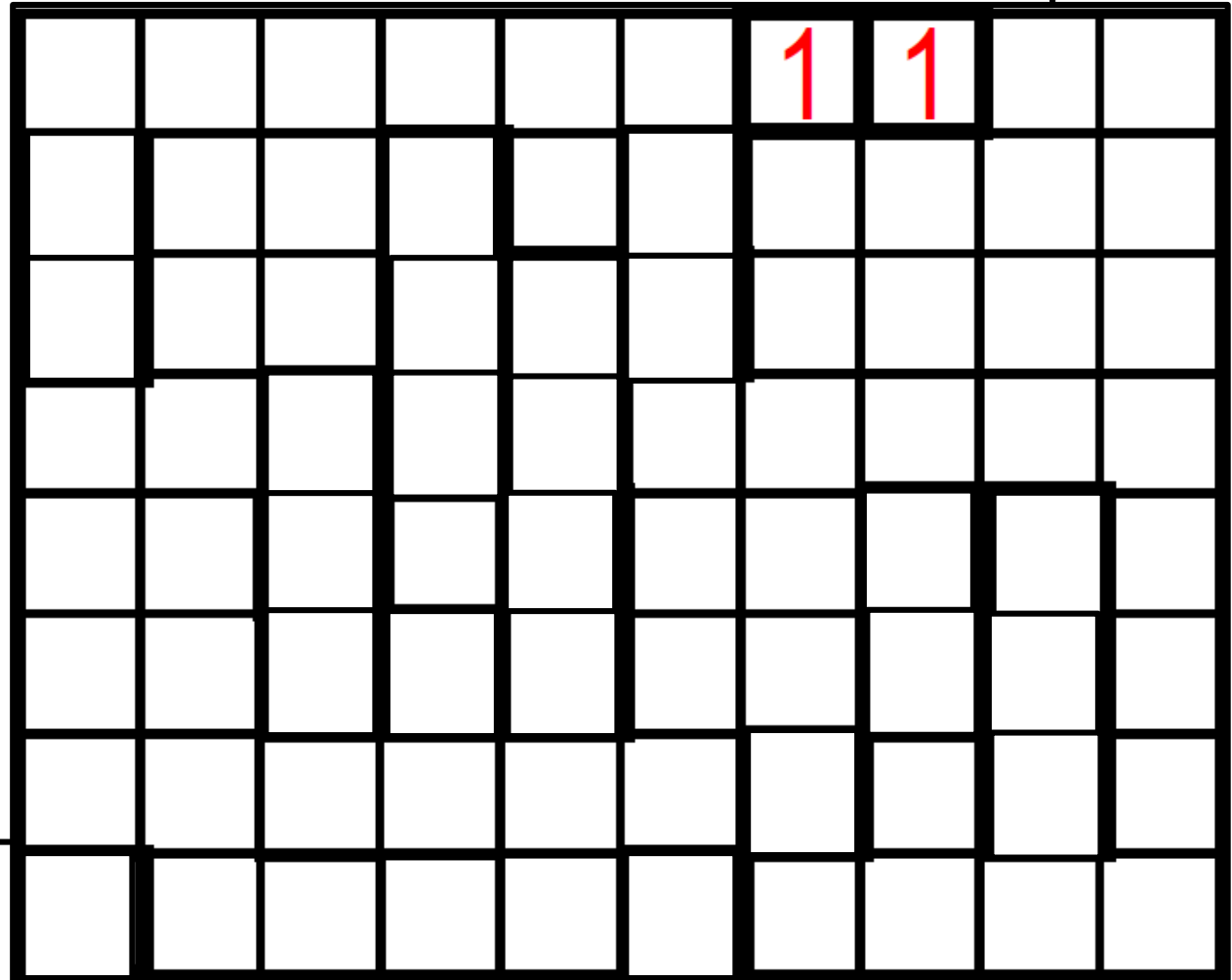
Imagem



Equiv

3=4

3=5



# PASSO 2

## VIZINHANÇA 4-CONECTADA

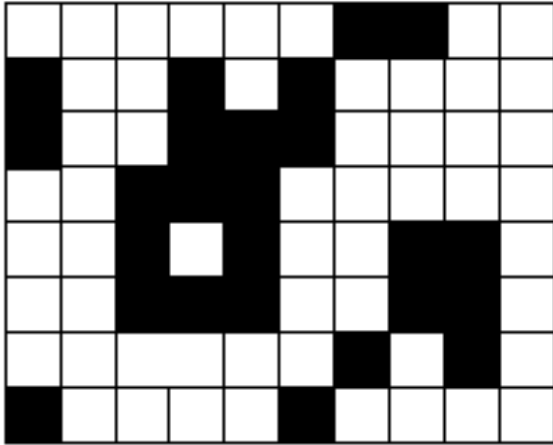
Equiv

3=4=5 Etiqueta 3

						1	1		
2			3		4				
2			3	3	3				
		5	5	5					
		5		5			6	6	
		5	5	5			6	6	
						7		6	
8					9				

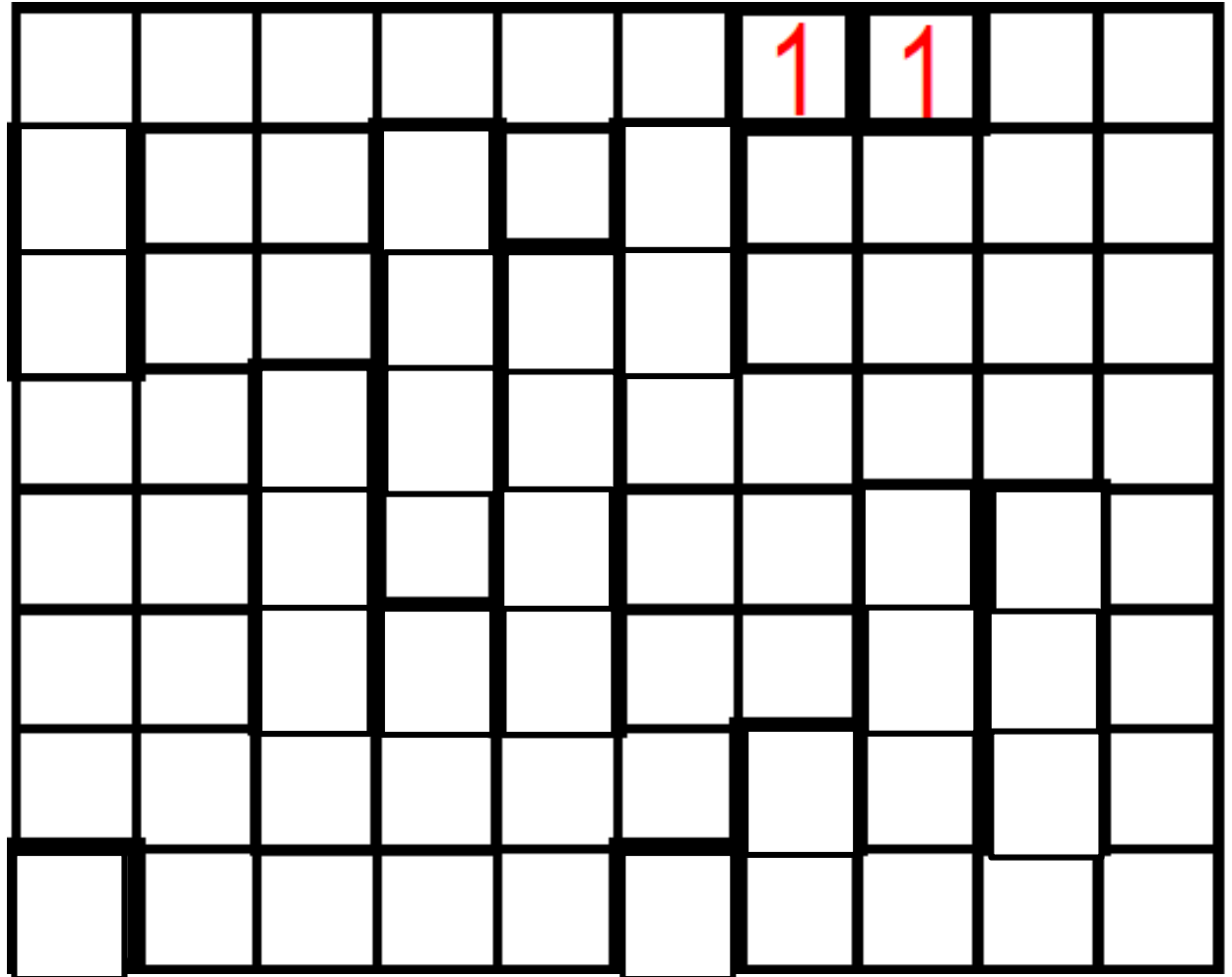
# APLICAÇÃO DO ALGORITMO VIZINHANÇA 8-CONECTADA

Imagem



Equiv

1=3



# PASSO 2

## VIZINHANÇA 8-CONECTADA

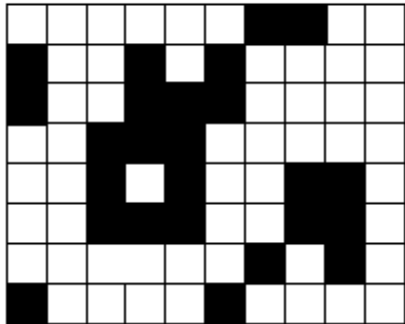
Equiv

1=3 Etiqueta 1

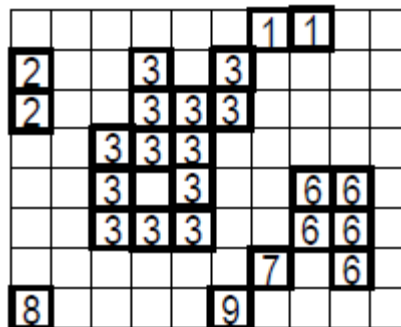
						1	1		
2			3		1				
2			3	3	3				
		3	3	3					
		3		3			4	4	
		3	3	3			4	4	
						4		4	
5					4				

# APLICAÇÃO DO ALGORITMO CORES

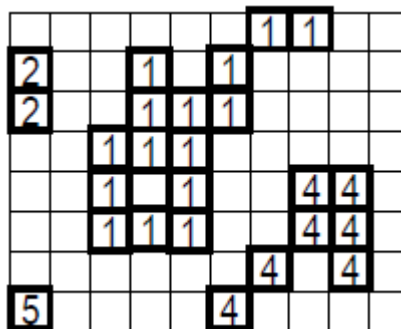
Imagen



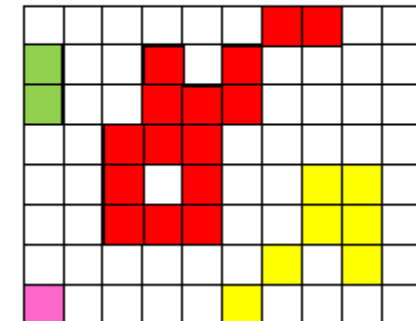
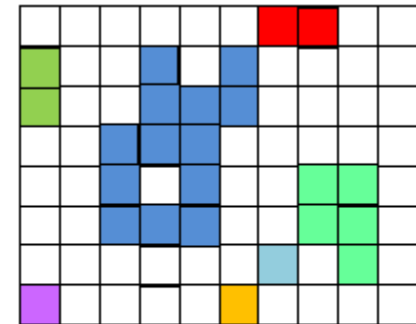
4-Conectada



8-Conectada



Etiquetas:





# LOCALIZAÇÃO DE JANELAS

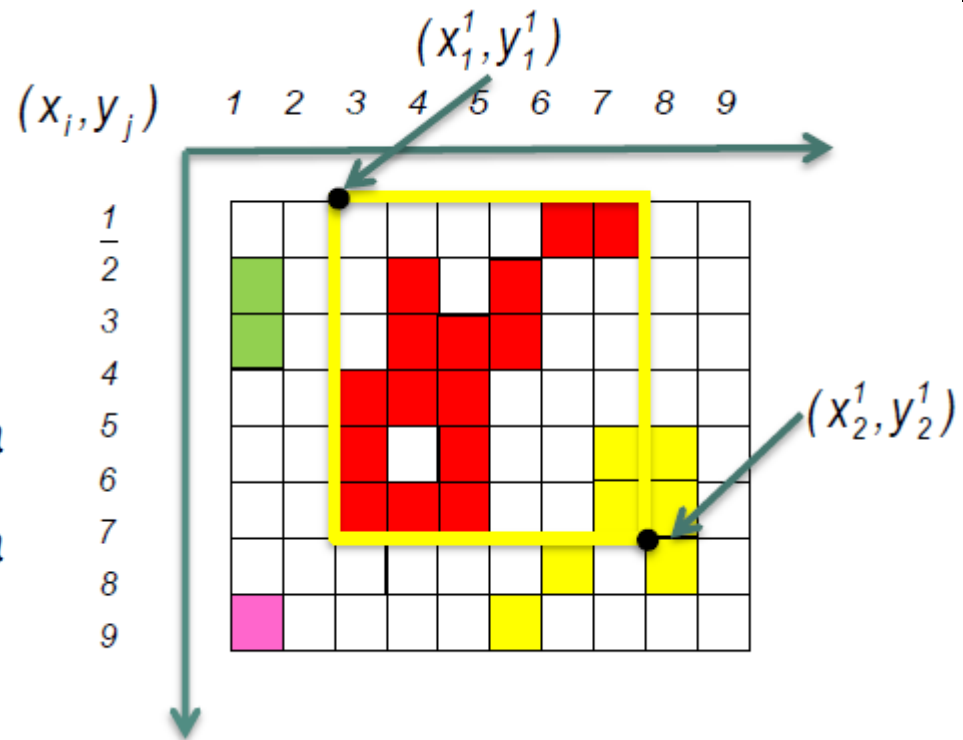
Para cada etiqueta:  $E_k$

$$x_1^k = \min_{x_i} \{(x_i, y_j) : etiqueta(x_i, y_j) = E_k\}$$

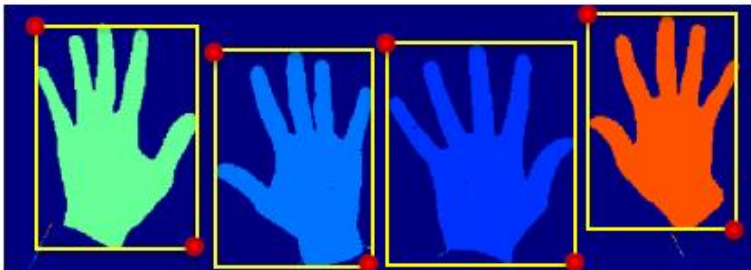
$$y_1^k = \min_{y_j} \{(x_i, y_j) : etiqueta(x_i, y_j) = E_k\}$$

$$x_2^k = \max_{x_i} \{(x_i, y_j) : etiqueta(x_i, y_j) = E_k\}$$

$$y_2^k = \max_{y_j} \{(x_i, y_j) : etiqueta(x_i, y_j) = E_k\}$$



# EXEMPLO



Etiquetas:

- |   |   |
|---|---|
| 1 | ■ |
| 2 | ■ |
| 3 | ■ |
| 4 | ■ |
| 5 | ■ |

# DIVISÃO E FUSÃO DE 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*

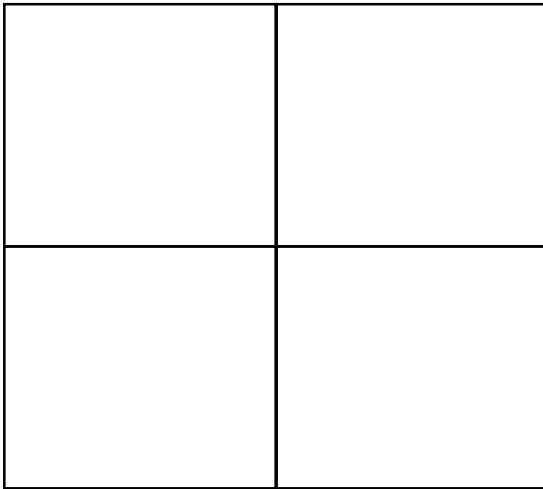
# QUADTREE

## Segmentação como seleção de regiões semelhantes em uma imagem

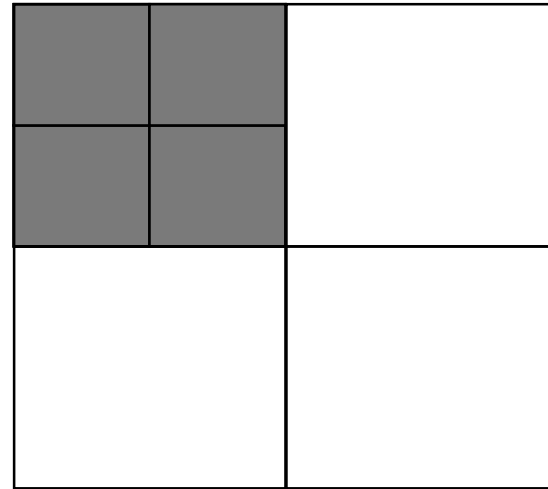
- Algoritmo de decomposição Quadtree
  - O algoritmo divide a imagem em blocos
  - Cada bloco é então testado para um dado critério de homogeneidade
    - Se o bloco atende o critério, ele não é mais dividido
    - Se não atende, ele é subdividido e cada parte é testada sob o mesmo critério

# QUADTREE

## Decomposição Quadtree



- 1ª Divisão da imagem
- Cada bloco é analisado



- Os blocos que não atendem o critério são divididos de novo

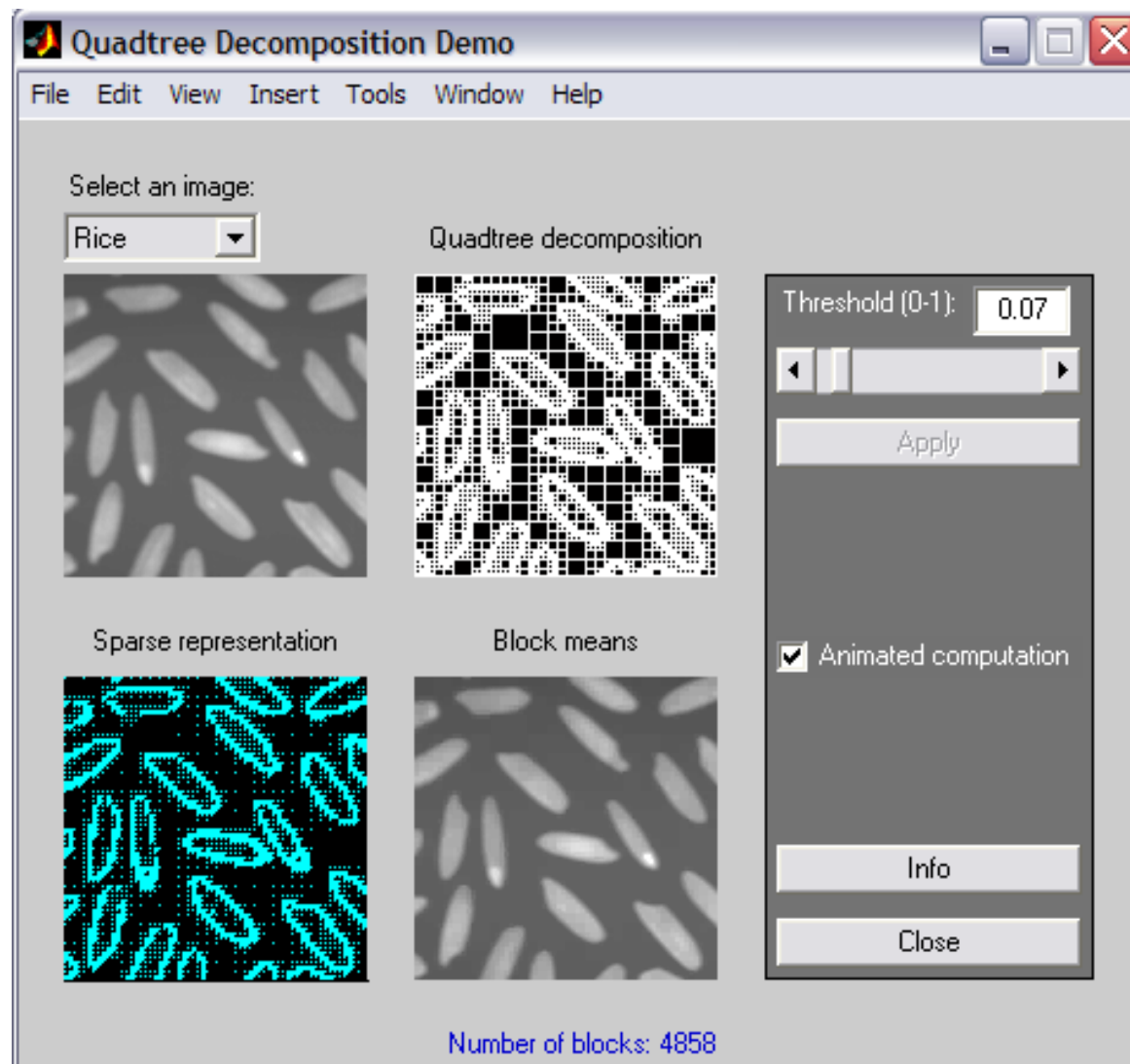
# QUADTREE

## Decomposição Quadtree

- Exemplo de critério:
  - (a maior cor do bloco – a menor cor do bloco)  $\leq 20$
- No MatLab
  - Função qtdecomp
  - Divide a imagem em quadrados de tamanhos: 512, 256, 128, 64, 32, 16, 8, 4, 2, e 1
  - Critério como estabelecido acima
  - Restrição: tamanho da imagem deve ser proporcional às dimensões possíveis
  - Exemplo: qtdecomp (imagem, 0.2)
    - 0.2 é o ponto de corte normalizado por 255



# QUADTREE



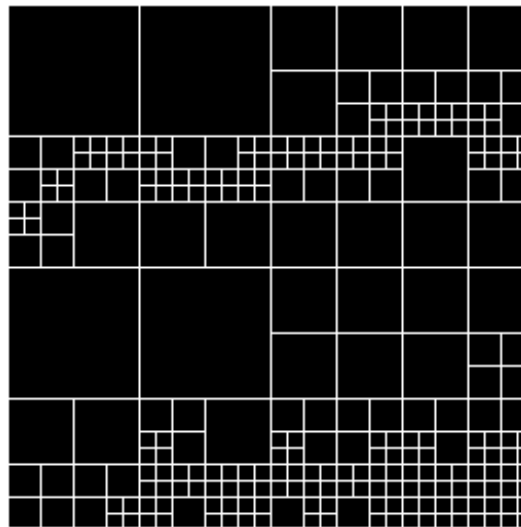
# SEGMENTAÇÃO POR REGIÕES

## Decomposição Quadtree

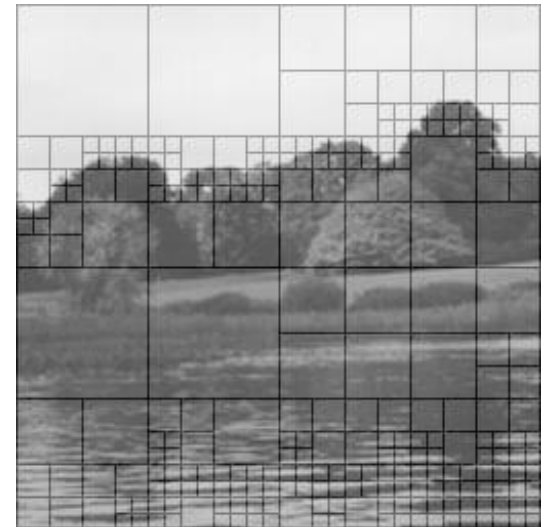
- Exemplo:



Imagem original



Decomposição



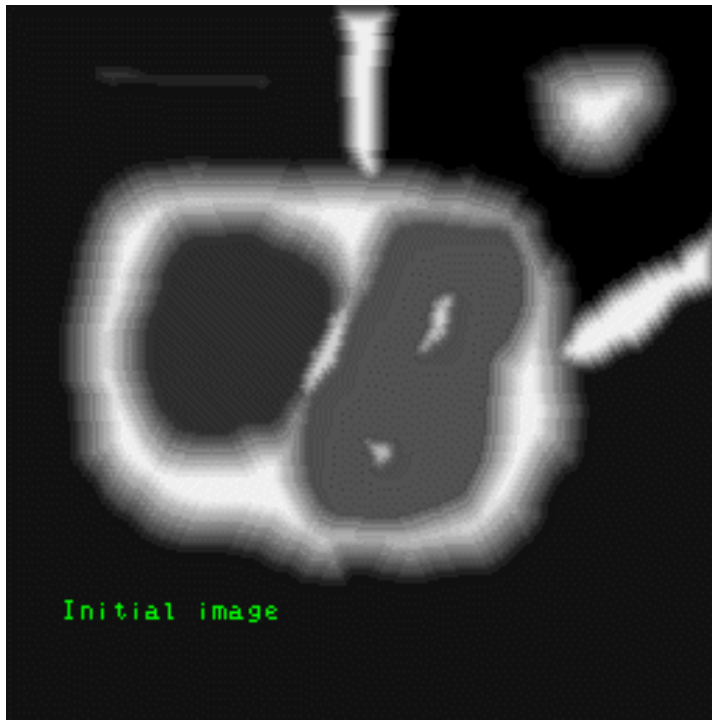
Visualização das regiões

# WATERSHED

- A transformação *Watershed* é similar ao *Edge Detection*
- Utiliza a técnica do Gradiente para detecção de bordas
- Primeiramente considera-se os pixels com maior magnitude de gradiente da como sendo bordas
- Assim, fazendo uma analogia com o cair de gotas em um superfície topológica, as gotas escorrerão um mínimo de intensidade local
- Os pixels que escorrem para o mínimo local formam uma poça, que pode ser considerado um grupo na segmentação

# WATERSHED

Watershed, em geografia, são as saliências que dividem as áreas inundadas por diferentes rios (Bacias Hidrográficas)



# WATERSHED

- Se aplicarmos conceitos de geografia àquela imagem topográfica acima que é como imaginamos a nossa imagem de gradientes, podemos então imaginar um método de segmentação que se baseia neste conceito geográfico e simula o que acontece em um conjunto de bacias hidrográficas sempre que chove muito: enchente
- Basta então que façamos chover nesta topografia de forma a inundar as regiões profundas (de gradiente nulo ou muito baixo). Regiões que formam bacias de contenção isoladas por gradientes fechados formarão regiões ou segmentos separados

# WATERSHED

Procede-se da seguinte forma:

## 1. Define-se uma altura mínima do nível de água

- Este valor é um parâmetro e determinará a sensibilidade do algoritmo
- Todos os gradientes com valor abaixo deste mínimo serão cobertos pela água na inicialização do algoritmo e ignorados
- Inicializa-se as regiões com os pixels que contém valores de gradiente abaixo deste limiar (pode-se limiarizar a imagem e passar para 0 tudo abaixo da altura mínima de água)

## 2. Conecta-se os pixels através do algoritmo de componentes conexas

- Tudo que está abaixo do limiar inicial de altura de água é rotulado e agrupado em segmentos conexos e o que estiver acima é inicialmente ignorado.
- A grande maioria dos segmentos gerados não vai se tocar, porque existem muitos gradientes que a esta altura do campeonato ainda estão "fora da água"



# WATERSHED

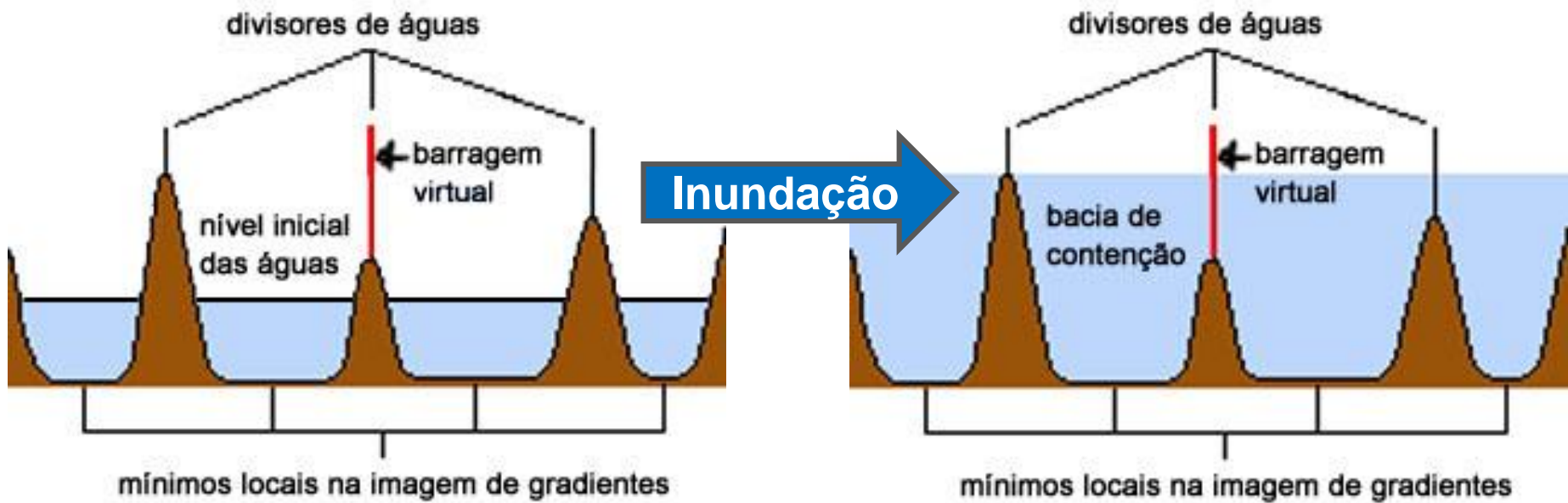
## 3. Os outros gradientes estarão fora da água

- Como possuem alturas diferentes, independentemente de sua altura, possuirão acima de seus pontos mais altos uma barragem virtual de largura 0 e altura infinita que impede que diferentes bacias de contenção se misturem
- Os pontos mais altos serão determinados naturalmente com o aumento do nível de inundação que se seguirá

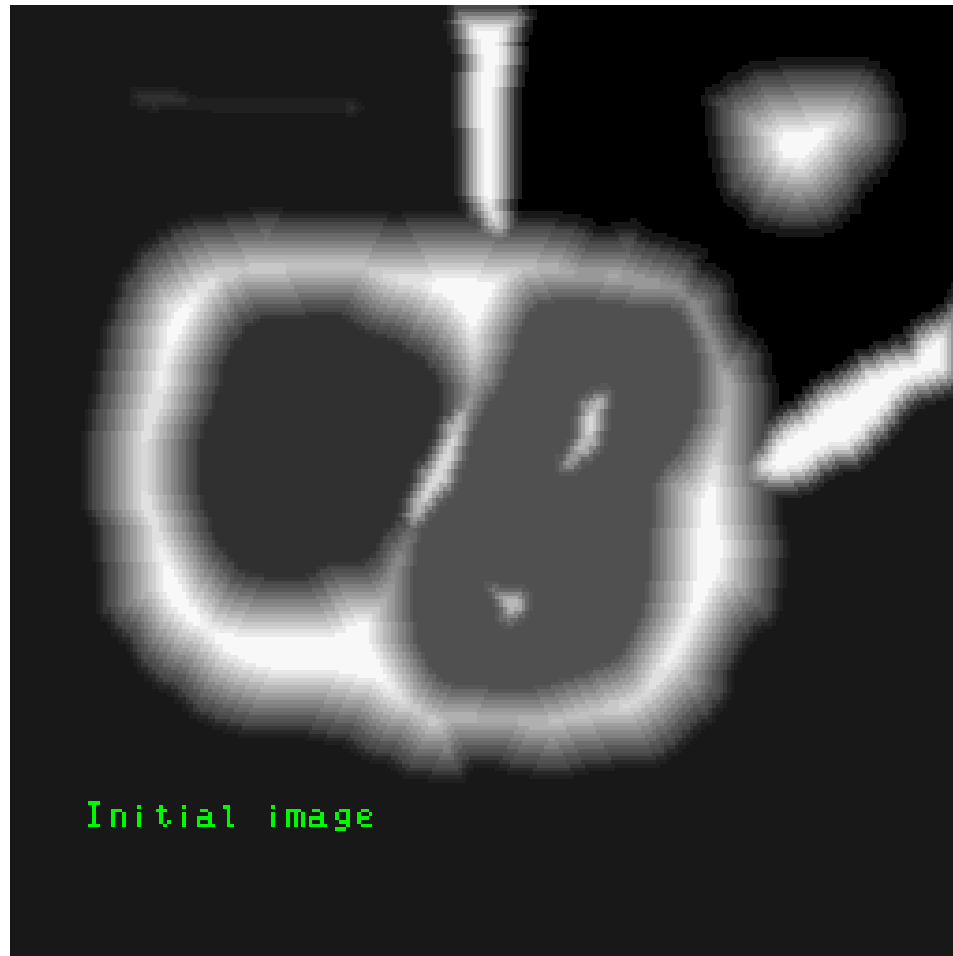
## 4. Faz-se chover incrementando iterativamente o nível de água

- A cada iteração, rotula-se todos os pixel ainda não rotulados que foram cobertos pela água, conectando-os à região mais próxima, dando a eles o rótulo dessa região
- Quando se atinge a barragem virtual e há ambiguidade na rotulação, deve-se atribuir o rótulo da região de gradiente descendente. Se o pixel se encontrar em um platô, sorteie
- O processo termina quando não há mais pixels sem rótulo e as águas cobriram tudo

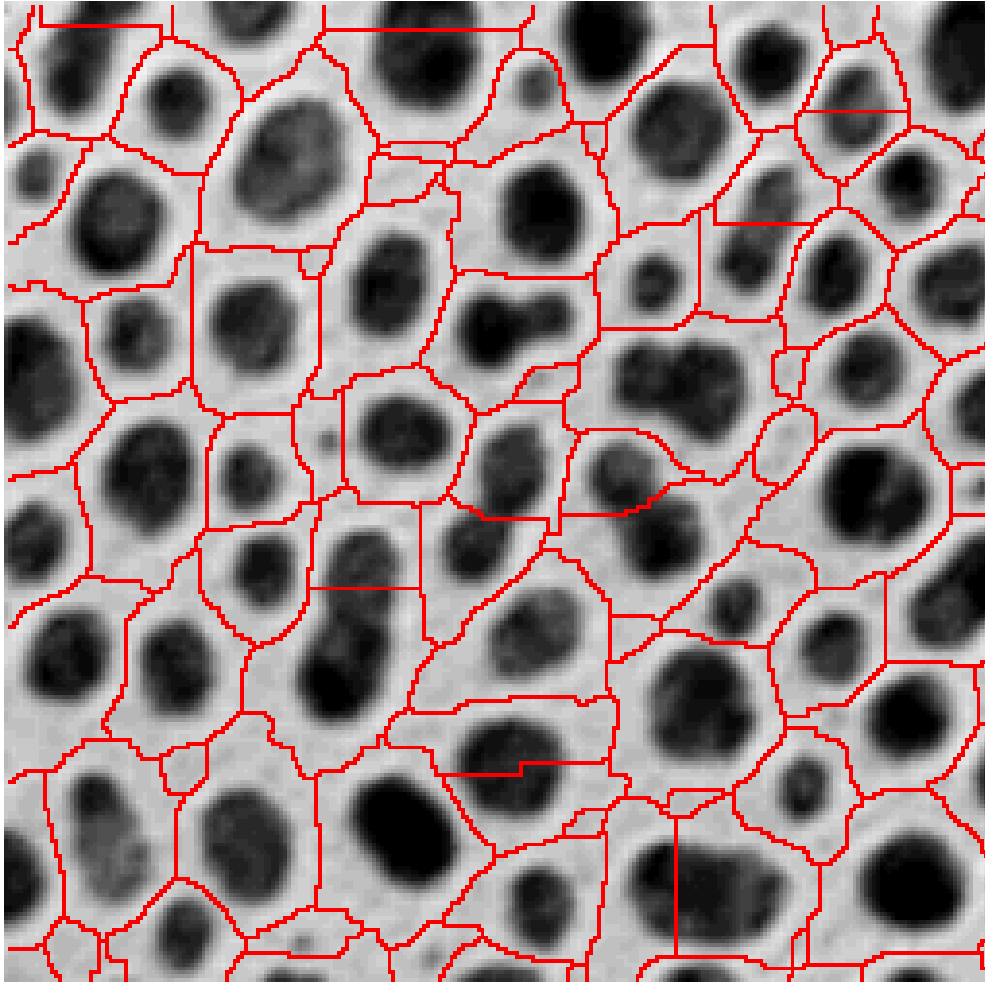
# WATERSHED



# WATERSHED



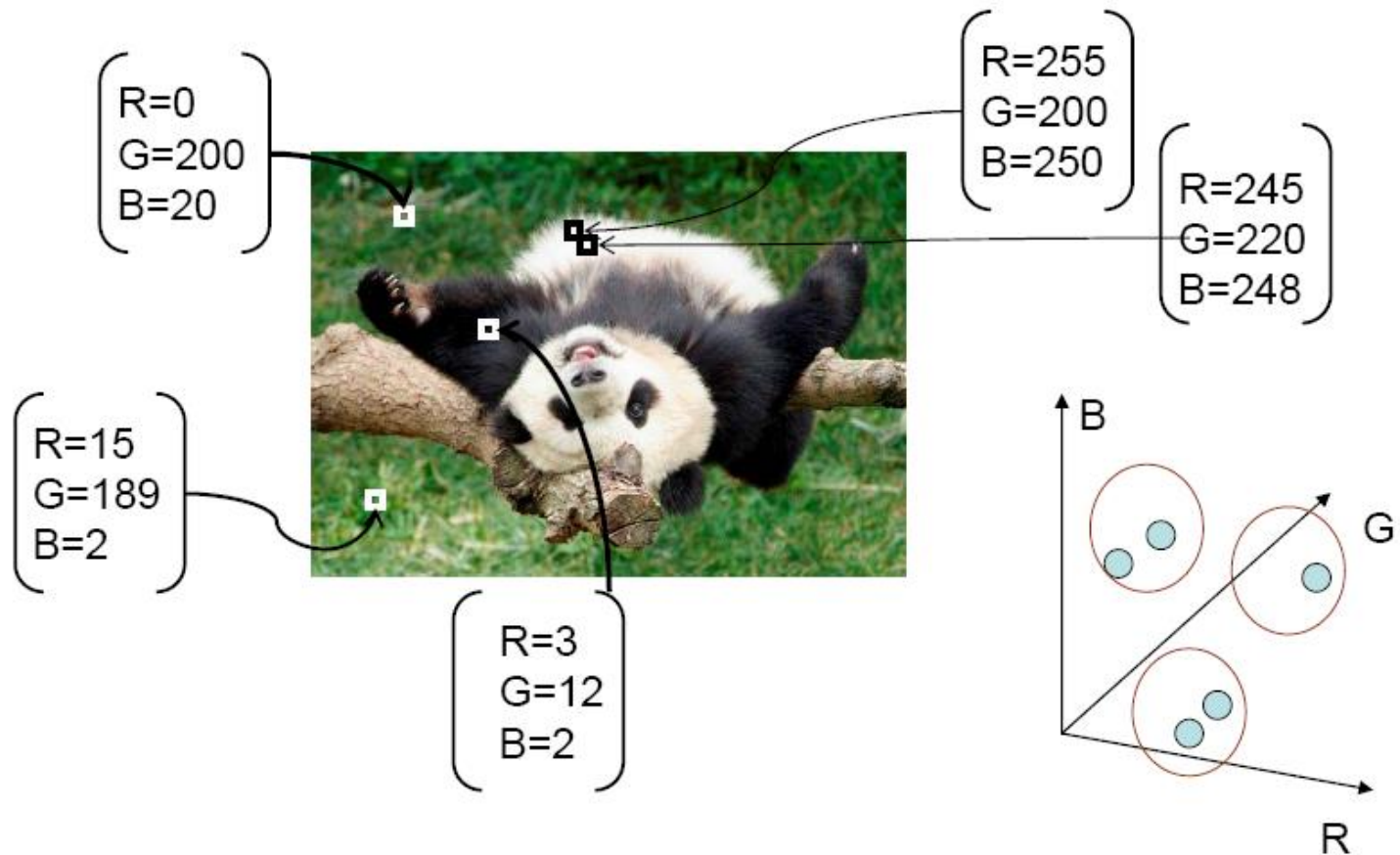
# WATERSHED



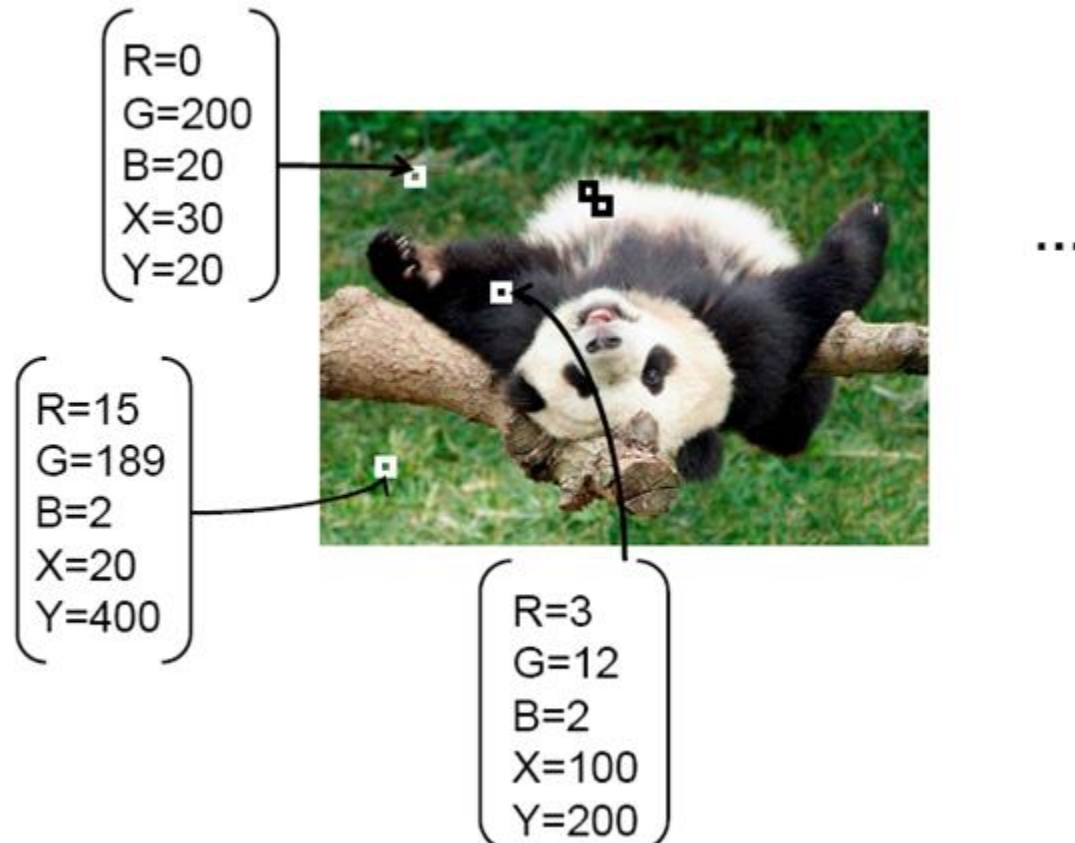
# CLUSTERING

- **Clustering: coloca pixels semelhantes no mesmo grupo e**
- **Principais Desafios:**
  - 1) O que faz com que dois pontos/imagens/pixels sejam semelhante?
  - 2) Como podemos calcular um agrupamento geral de similaridades entre pares?

# ESPAÇO DE CARACTERÍSTICAS



# ESPAÇO DE CARACTERÍSTICAS



# Agrupamento utilizando dados de intensidade e de cor

Imagem



Clusters em intensidade



Clusters em cores





# CLUSTERING

- **K-Means**
- **Mean-Shift (melhorar)**

# ALGORITMO K-MEANS

Algoritmo particional baseado em Otimização do Erro Quadrado

$$J(X) = \sum_{i=1}^n \sum_{k=1}^K d(x_i, c_k)$$

Diagram illustrating the components of the K-Means objective function  $J(X)$ :

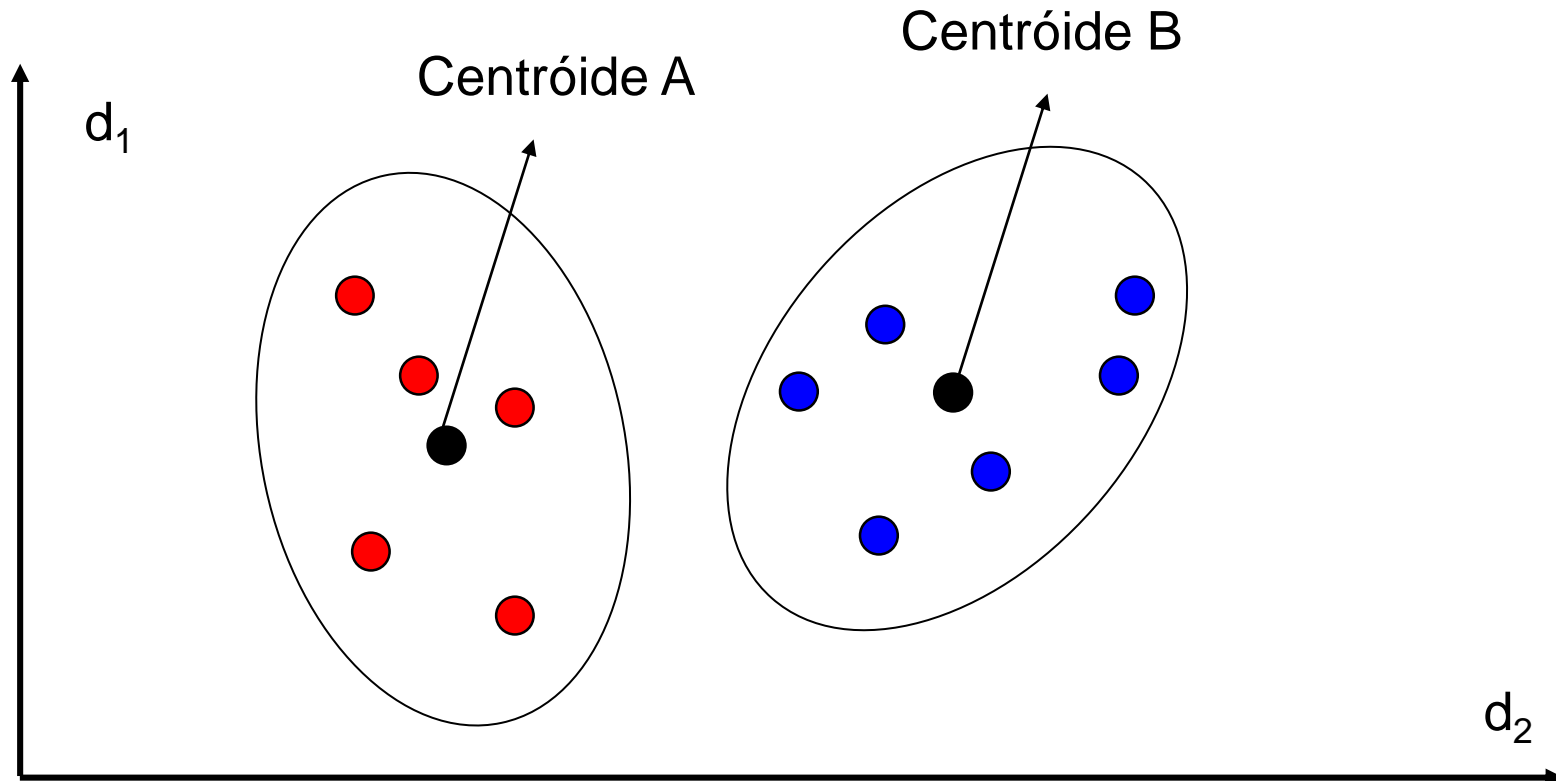
- $J(X)$ : Conjunto de Objetos (Set of Objects)
- $x_i$ : i-ésimo objeto (i-th object)
- $c_k$ : centróide do cluster k (centroid of cluster k)

Distância Euclidiana  $d(x_i, c_k) = \sqrt{(|x_{i1} - c_{k1}|^2 + |x_{i2} - c_{k2}|^2 + \dots + |x_{ip} - c_{kp}|^2)}$

$x_i = (x_{i1}, x_{i2}, \dots, x_{ip})$  e  $c_k = (c_{k1}, c_{k2}, \dots, c_{kp})$  são dois vetores  $p$ -dimensionais

# ALGORITMO K-MEANS

Encontra de forma iterativa os  
*centróides* dos grupos **G**



# ALGORITMO K-MEANS

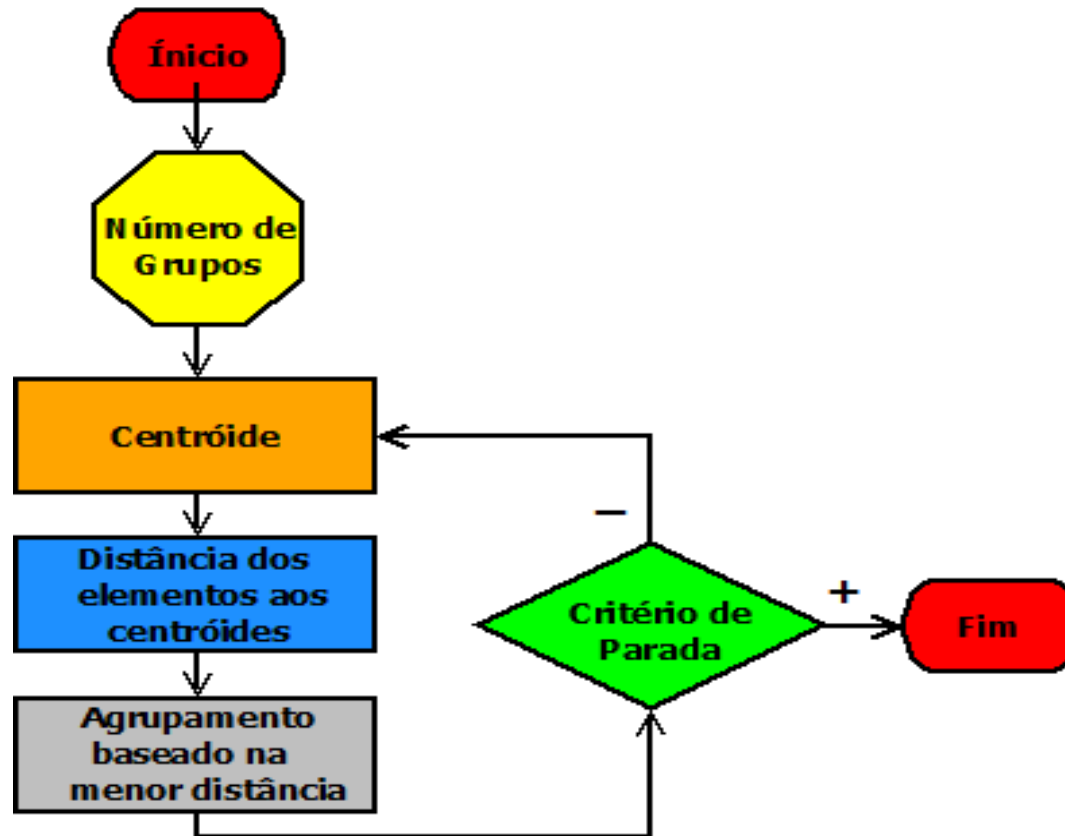
Clusters definidos com base nos **centróides** (**centro de gravidade**, ou o ponto médio dos cluster:

$$c_k = \frac{\sum_{x_i \in G_k} x_i}{K}, \quad k = 1, \dots, K.$$

Alocação dos objetos  $x_i$  nos clusters  $G_k$  feita com base na similaridade com o centróide até critério de parada

# AGRUPAMENTO

## K-Means



# ALGORITMO K-MEANS

**Passo 1: Defina  $k$  centróides iniciais, escolhendo  $k$  objetos aleatórios;**

**Passo 2: Atribua cada objeto para o cluster correspondente ao centróide mais similar;**

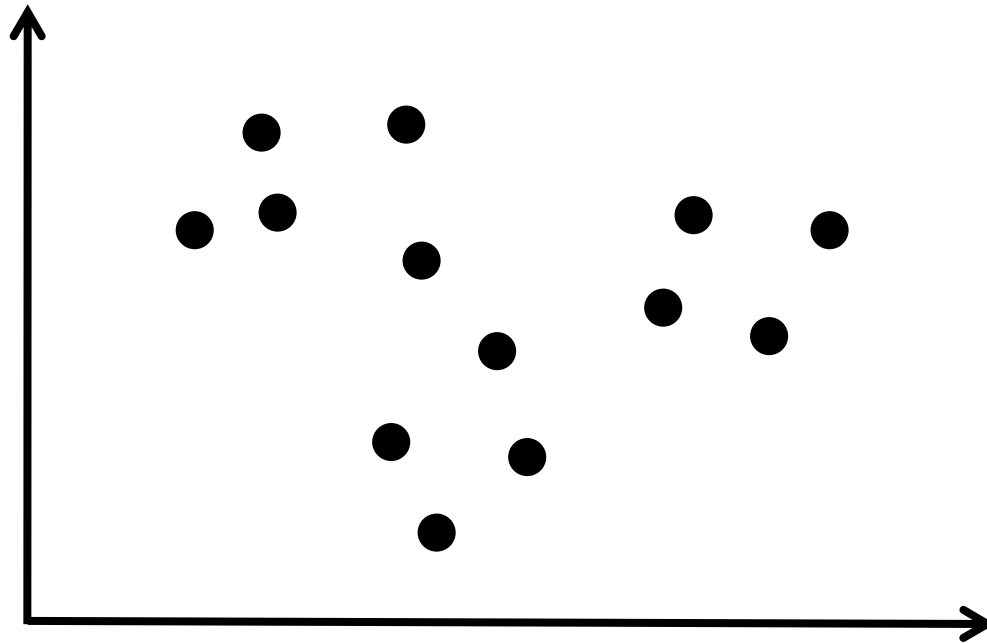
**Passo 3: Recalcule os centróides dos clusters.**

**Passo 4: Repita passo 2 e 3 até atingir um critério de parada**

- e.g. até um número máximo de iterações ou;
- até não ocorrer alterações nos centróides (i.e. convergência para um mínimo local da função de erro quadrado)

# ALGORITMO K-MEANS

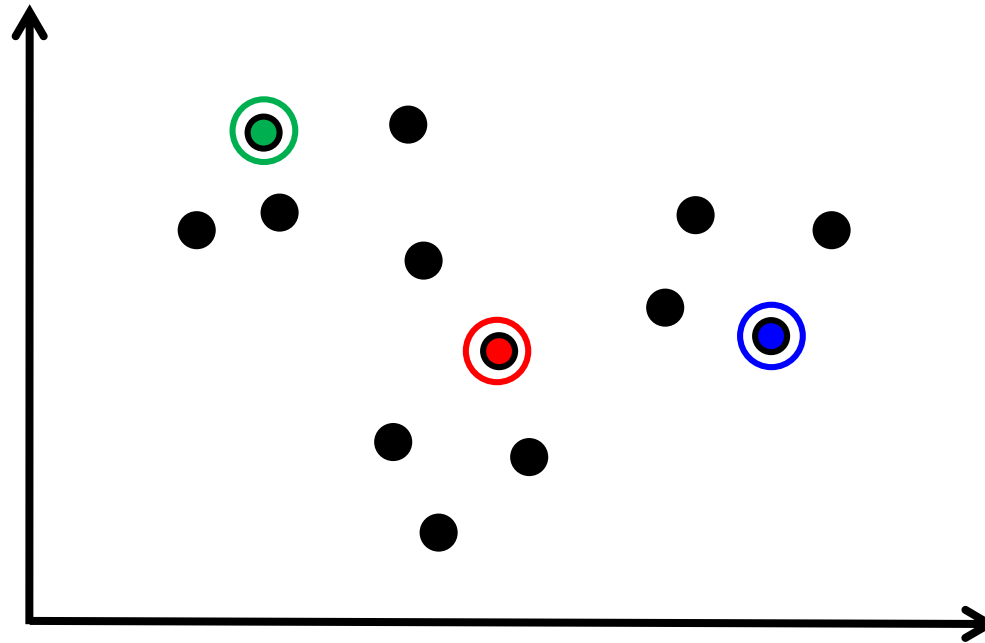
💡Exemplo:



# ALGORITMO K-MEANS

💡 Exemplo:

$k = 3$



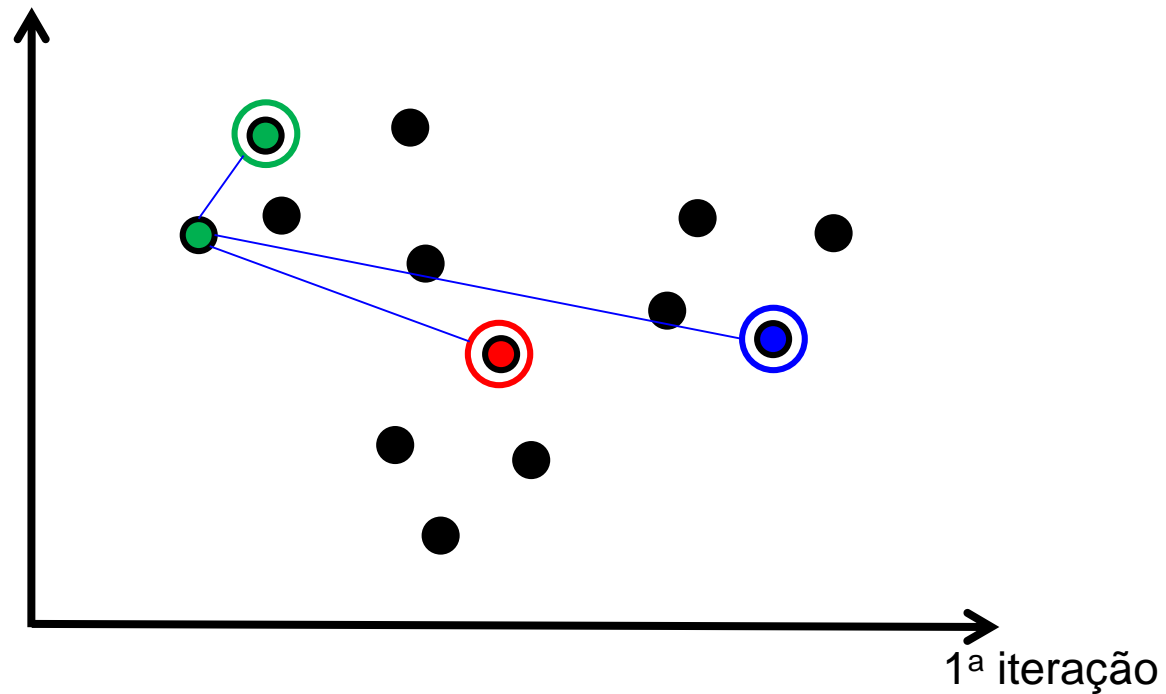
Seleciona-se  $k$  centróides iniciais.



# ALGORITMO K-MEANS

💡 Exemplo:

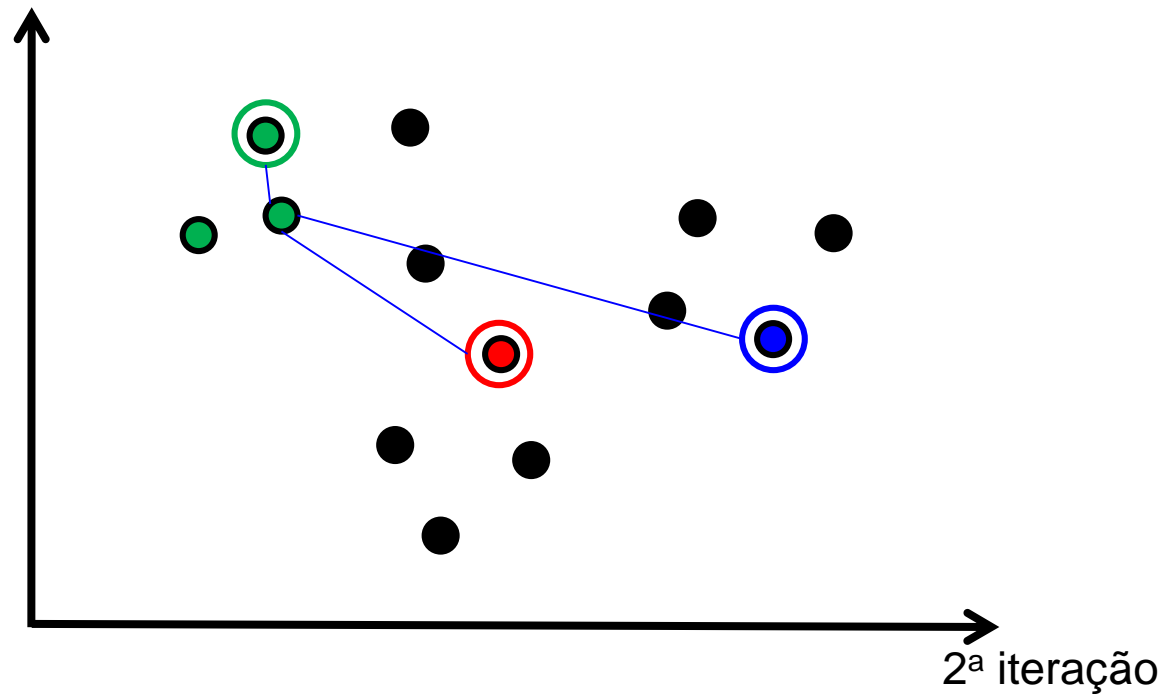
$k = 3$



# ALGORITMO K-MEANS

💡 Exemplo:

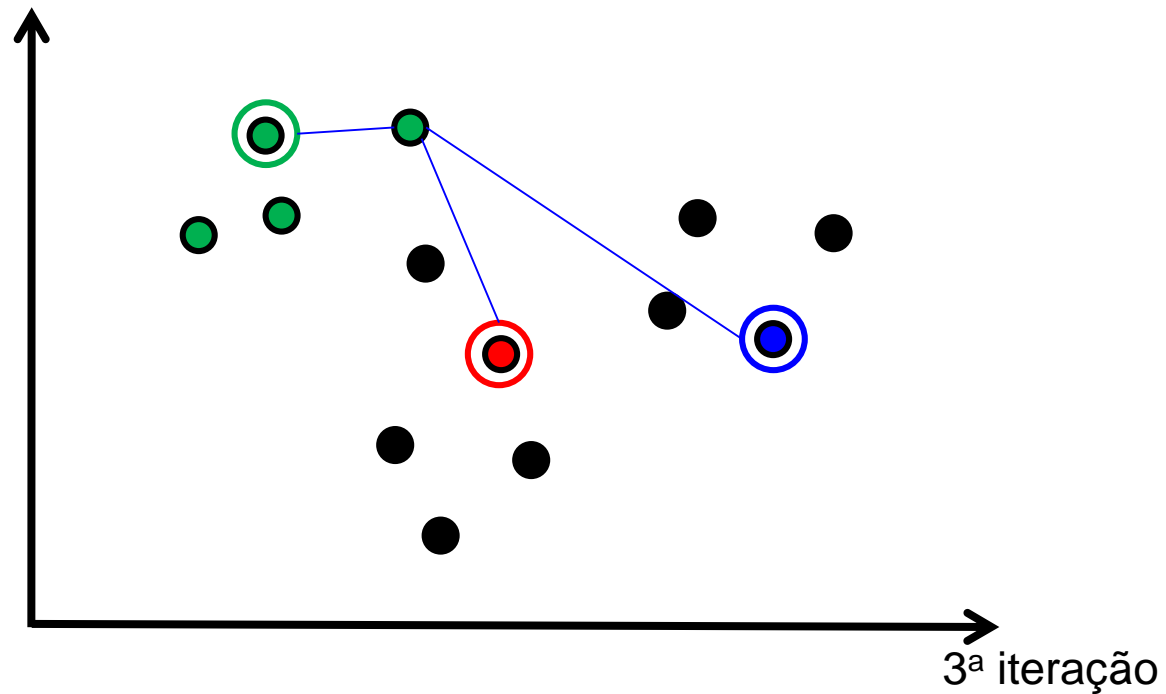
$k = 3$



# ALGORITMO K-MEANS

💡 Exemplo:

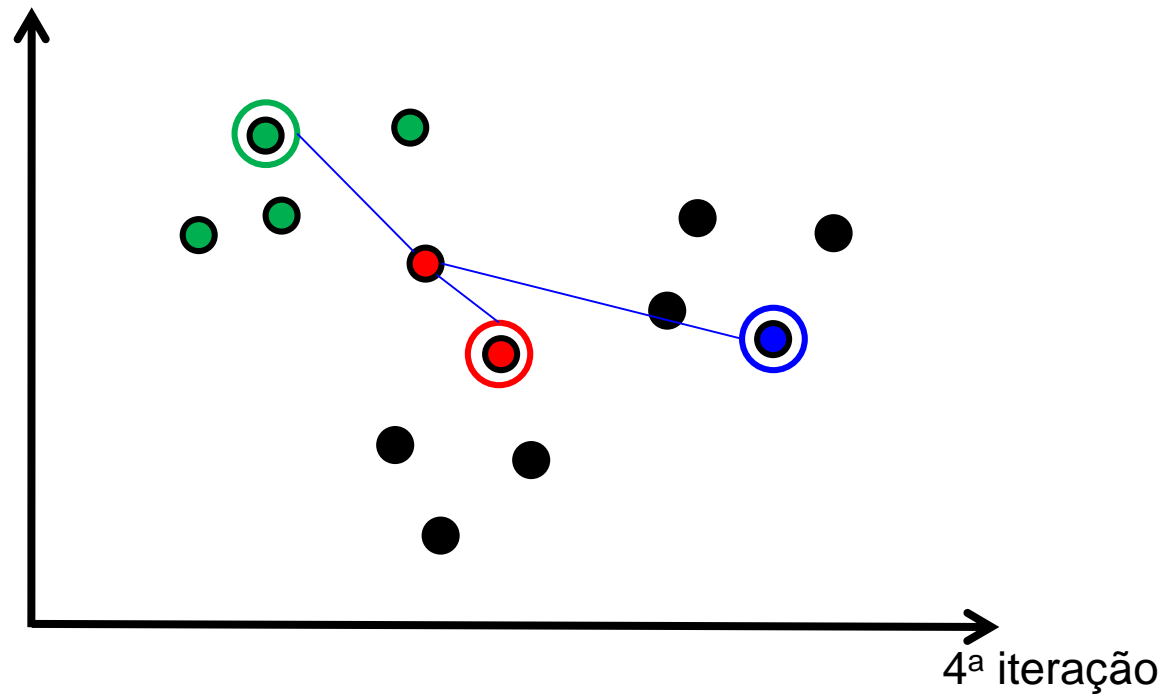
$k = 3$



# ALGORITMO K-MEANS

💡 Exemplo:

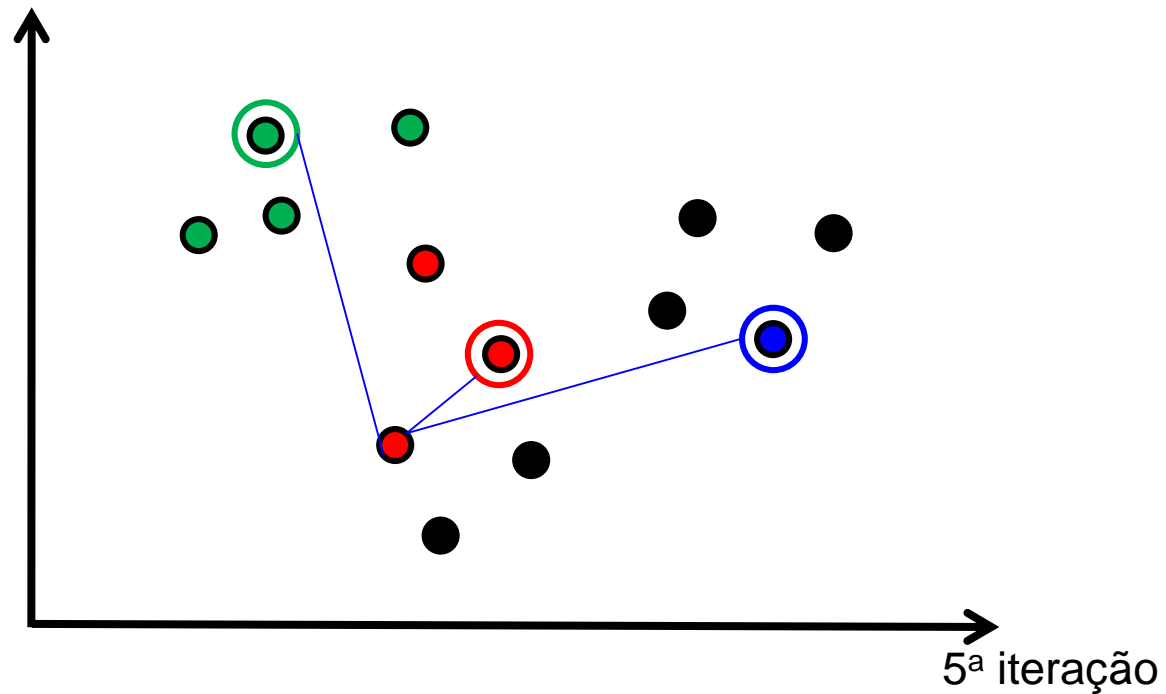
$k = 3$



# ALGORITMO K-MEANS

💡 Exemplo:

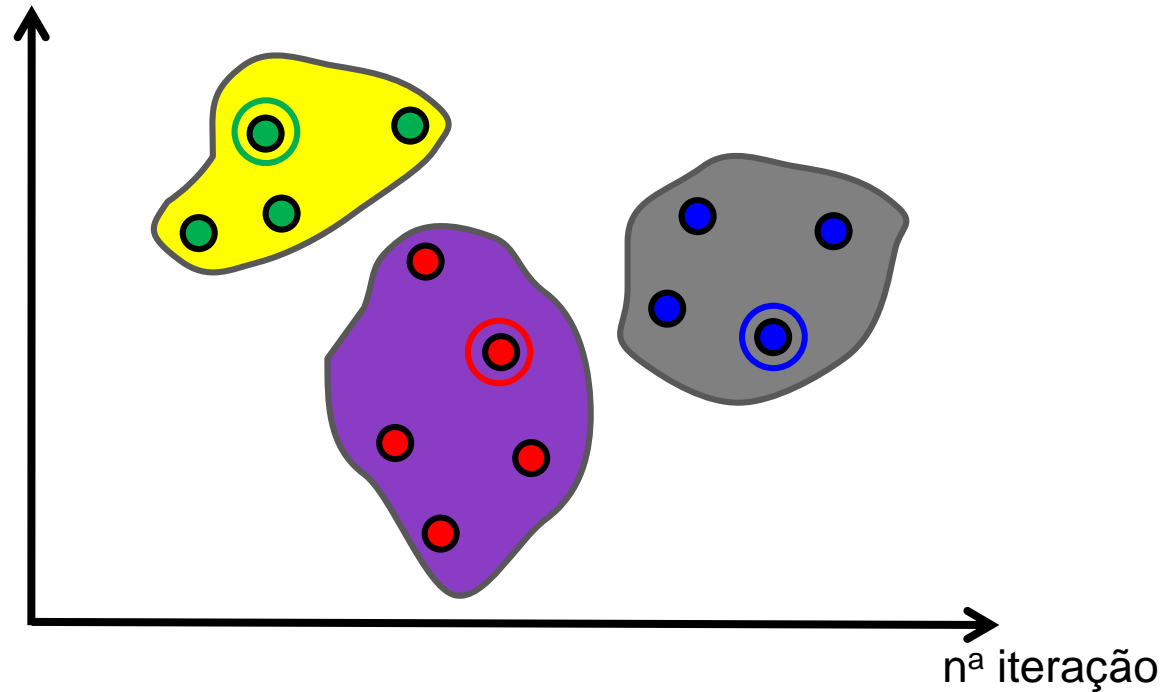
$k = 3$



# ALGORITMO K-MEANS

💡 Exemplo:

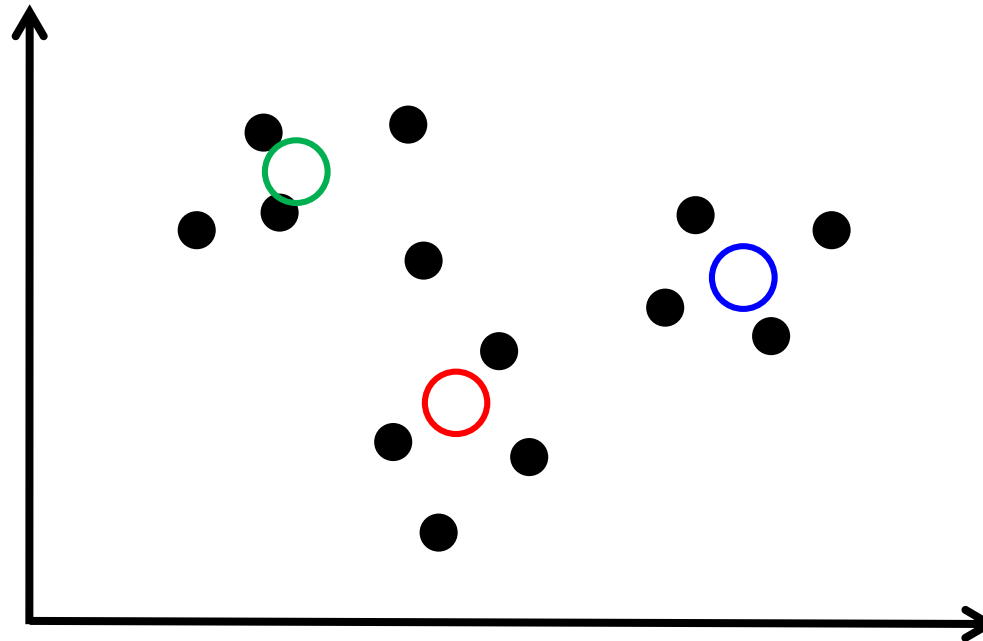
$k = 3$



# ALGORITMO K-MEANS

💡 Exemplo:

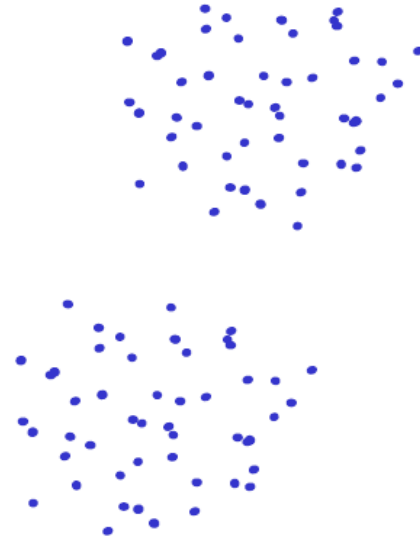
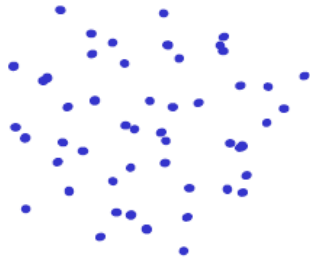
$k = 3$



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

# PROBLEMAS DO K-MEANS

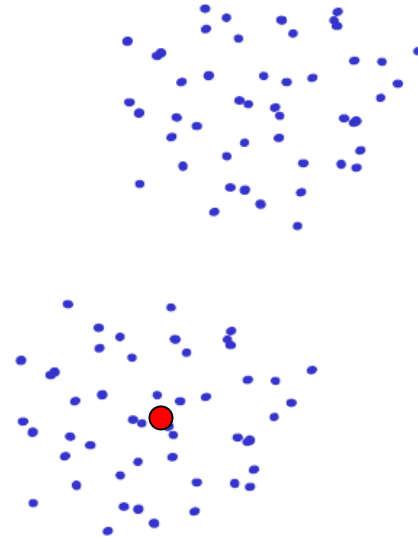
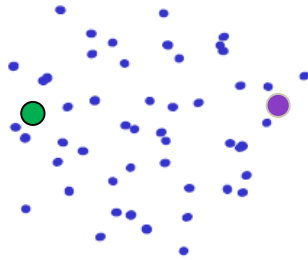
💡 O principal problema do K-Means é a dependência de uma boa inicialização.





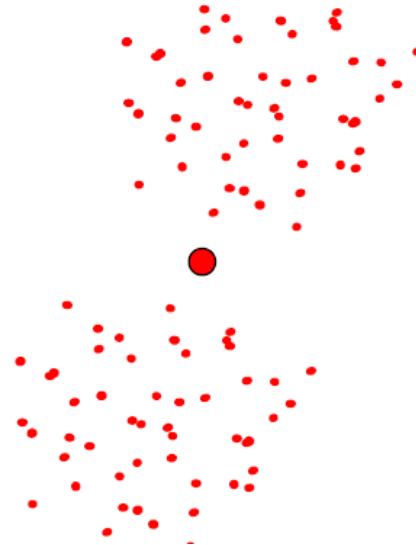
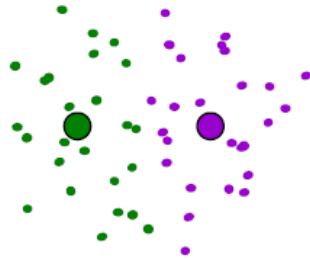
# PROBLEMAS DO K-MEANS

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



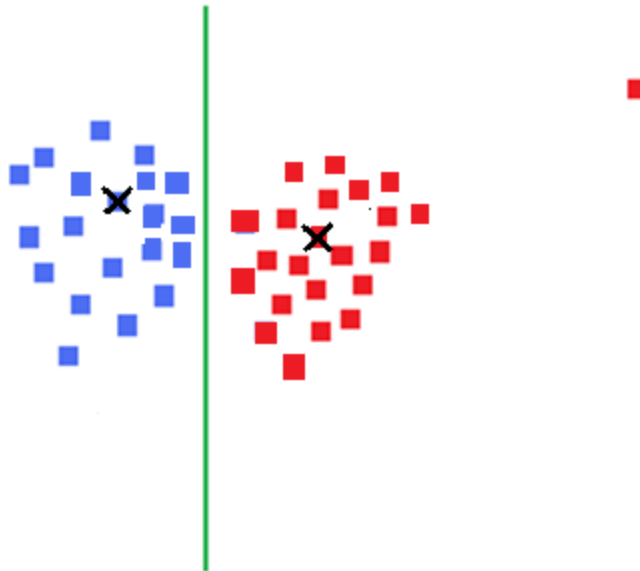
# PROBLEMAS DO K-MEANS

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



# PROBLEMAS DO K-MEANS

Sensitivo a outliers



# K-MEANS EXEMPLO 1

1. Select an image:  2. Select a processor:  3. Click

Options:  
Init Method



640\*480 (590,68): RGB(158,206,229) Process done !

# K-MEANS EXEMPLO 2

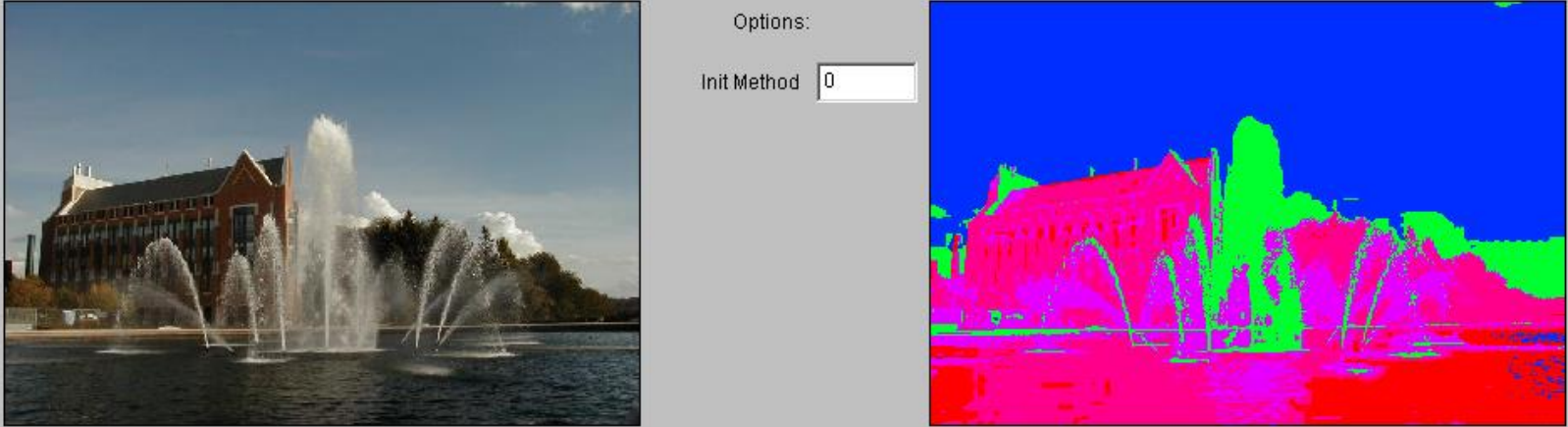
1. Select an image:  2. Select a processor:  3. Click

Options:  
Init Method

640\*480 (636,95): RGB(102,130,151)

Process done !

(590,209): RGB(0,46,255)



# K-MEANS EXEMPLO 3

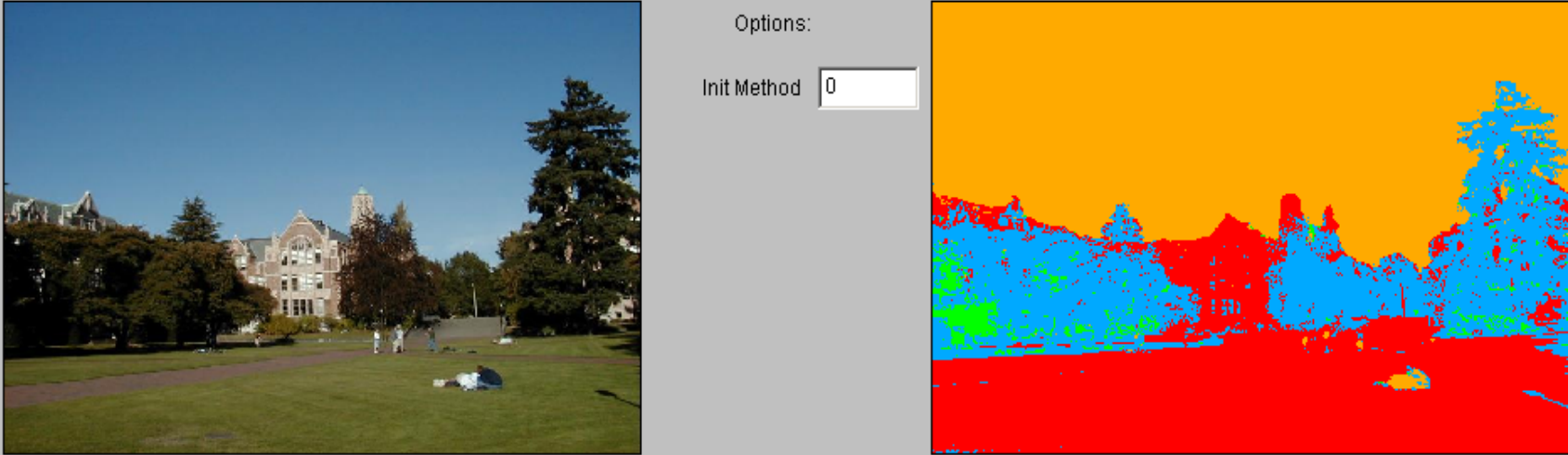
1. Select an image:  2. Select a processor:  3. Click

Options:  
Init Method

640\*480 (607,118): RGB(20,22,1)

Process done !

(228,26): RGB(255,170,0)



# K-MEANS

- **Vantagens:**

- Simples e Rápido
- Fácil de implementar

- **Contra:**

- Precisa escolher K
- Sensível a outliers

# MEAN-SHIFT

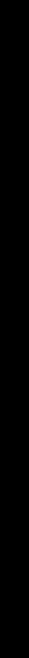
- Uma técnica versátil e avançada pra fazer segmentação baseada em clustering (agrupamento)



# MEAN-SHIFT

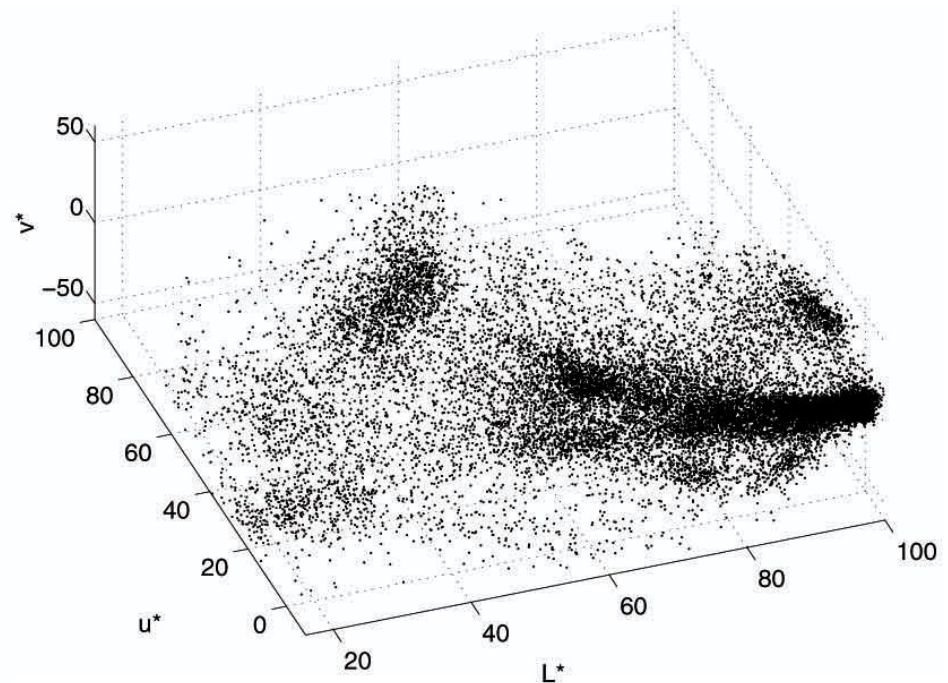
O algoritmo mean shift procura por meios ou máximos locais de densidade no espaço de características

Imagem

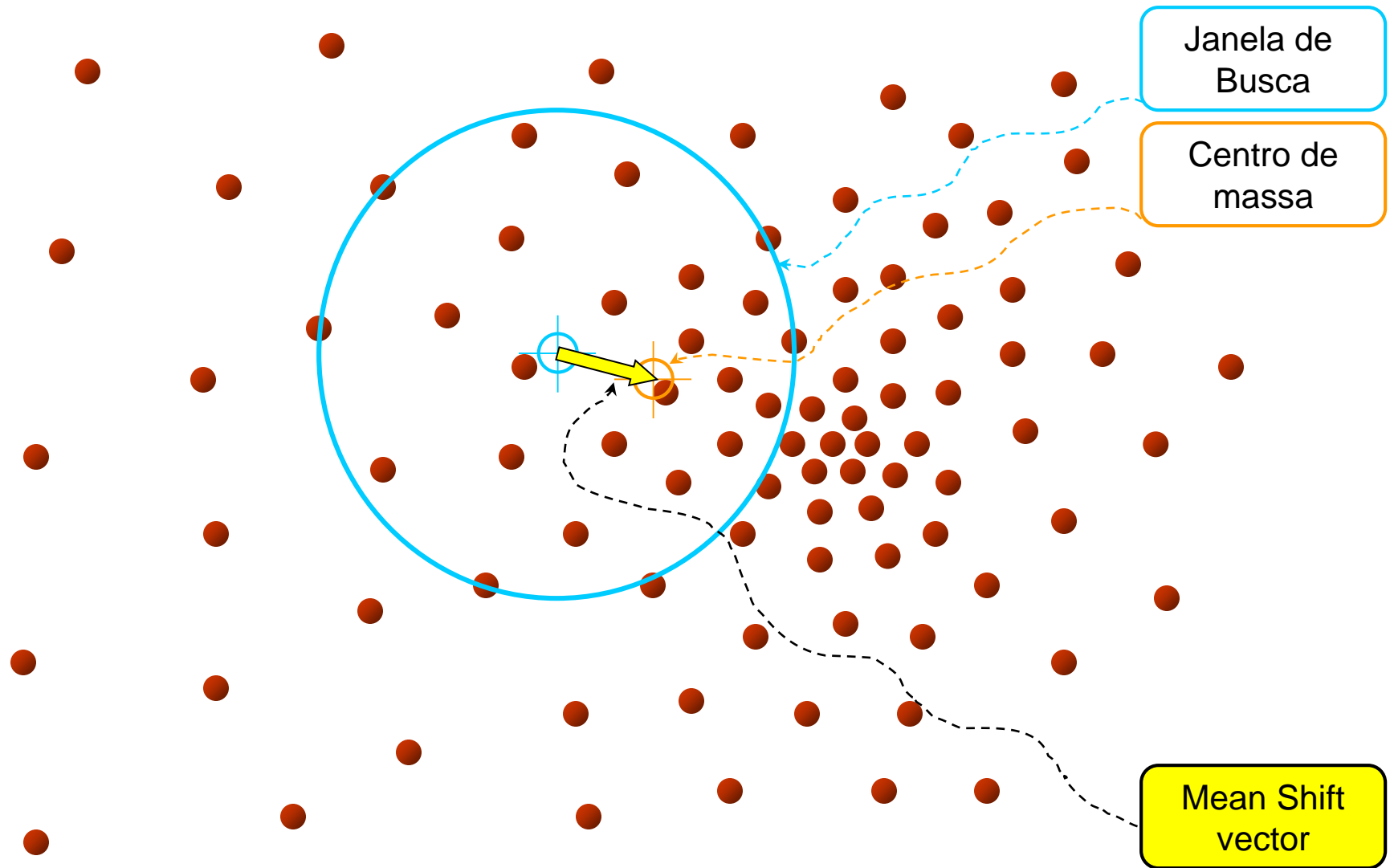


Espaço de características

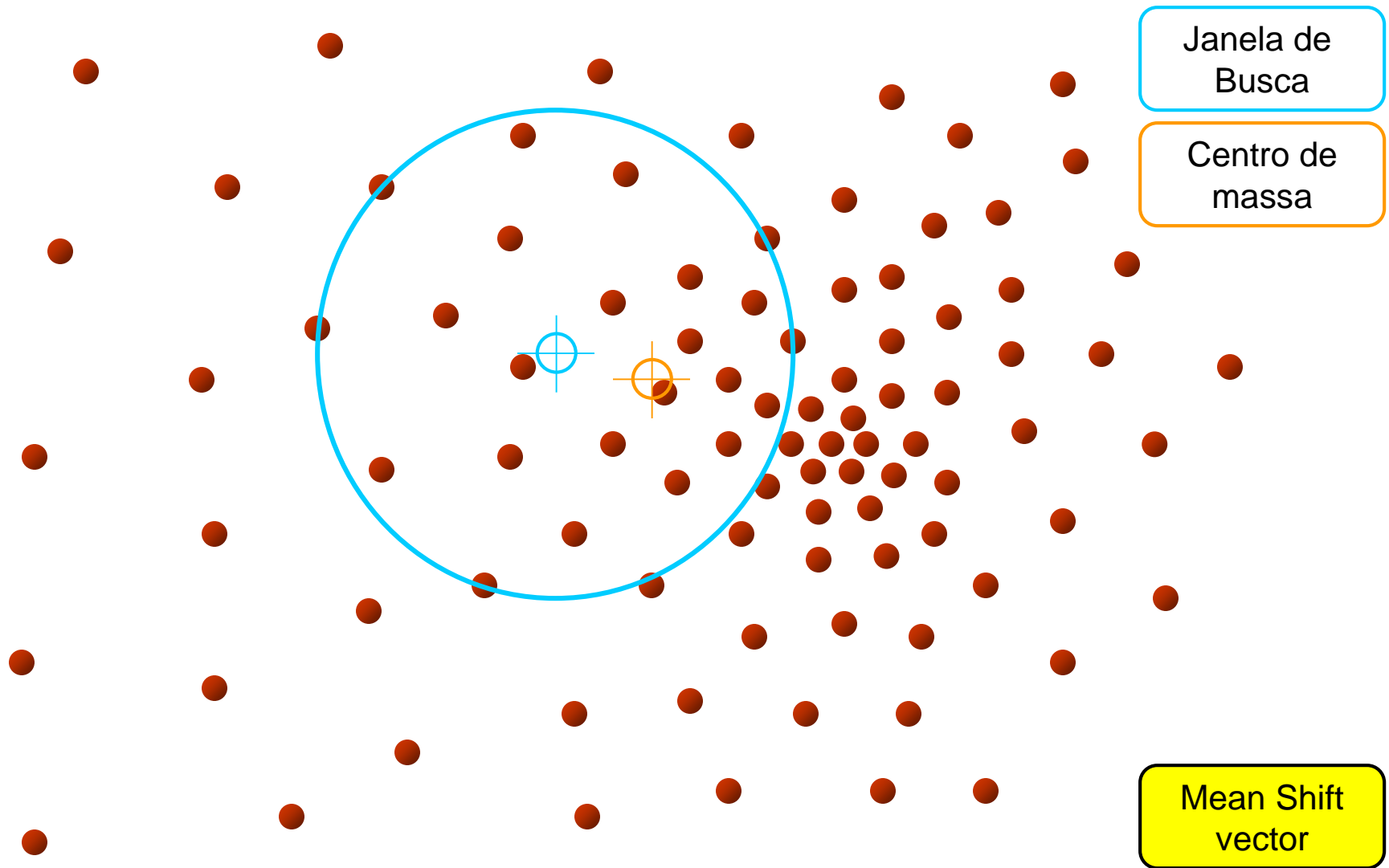
( $L^*u^*v$  Valor de Cores)



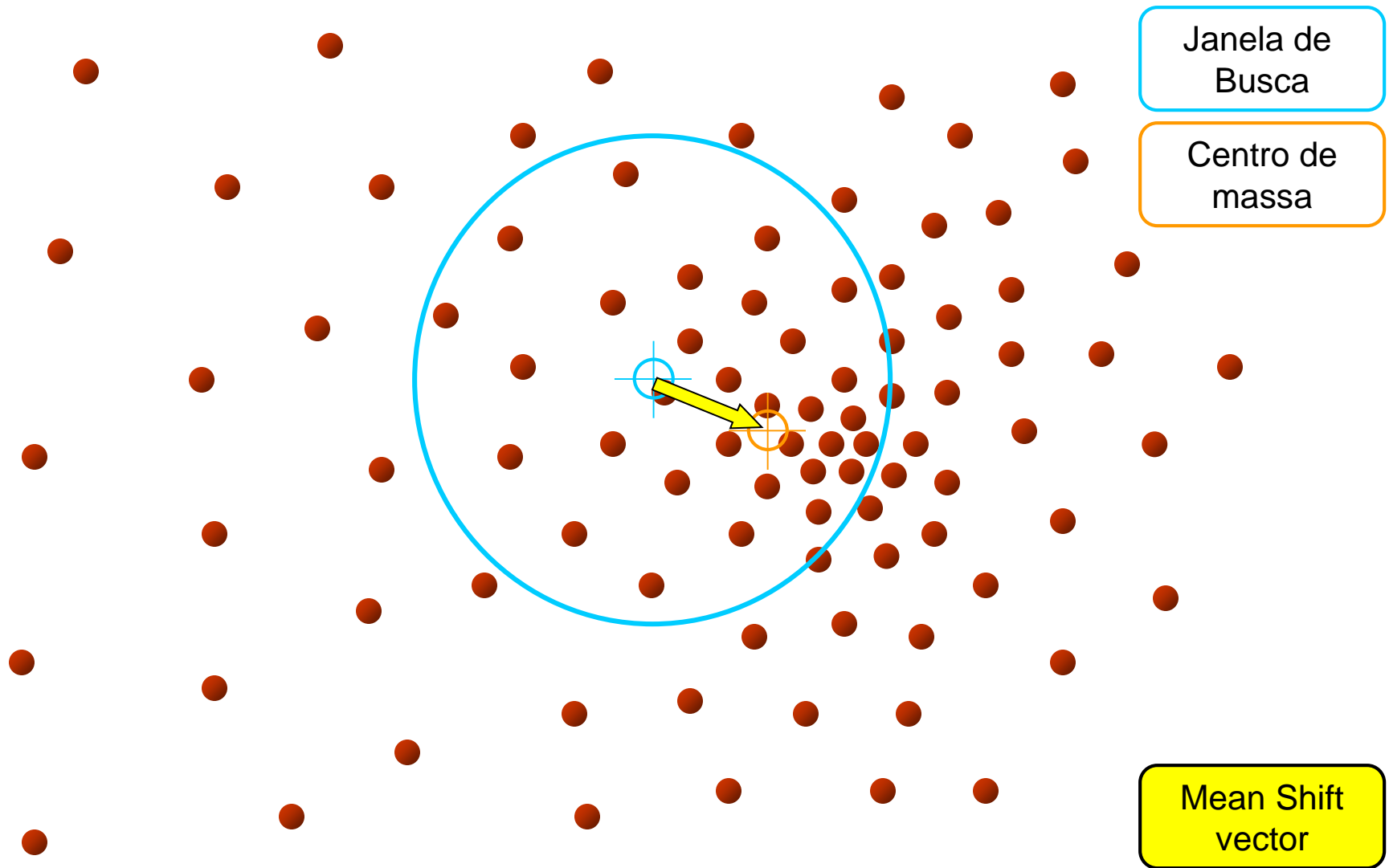
# MEAN SHIFT



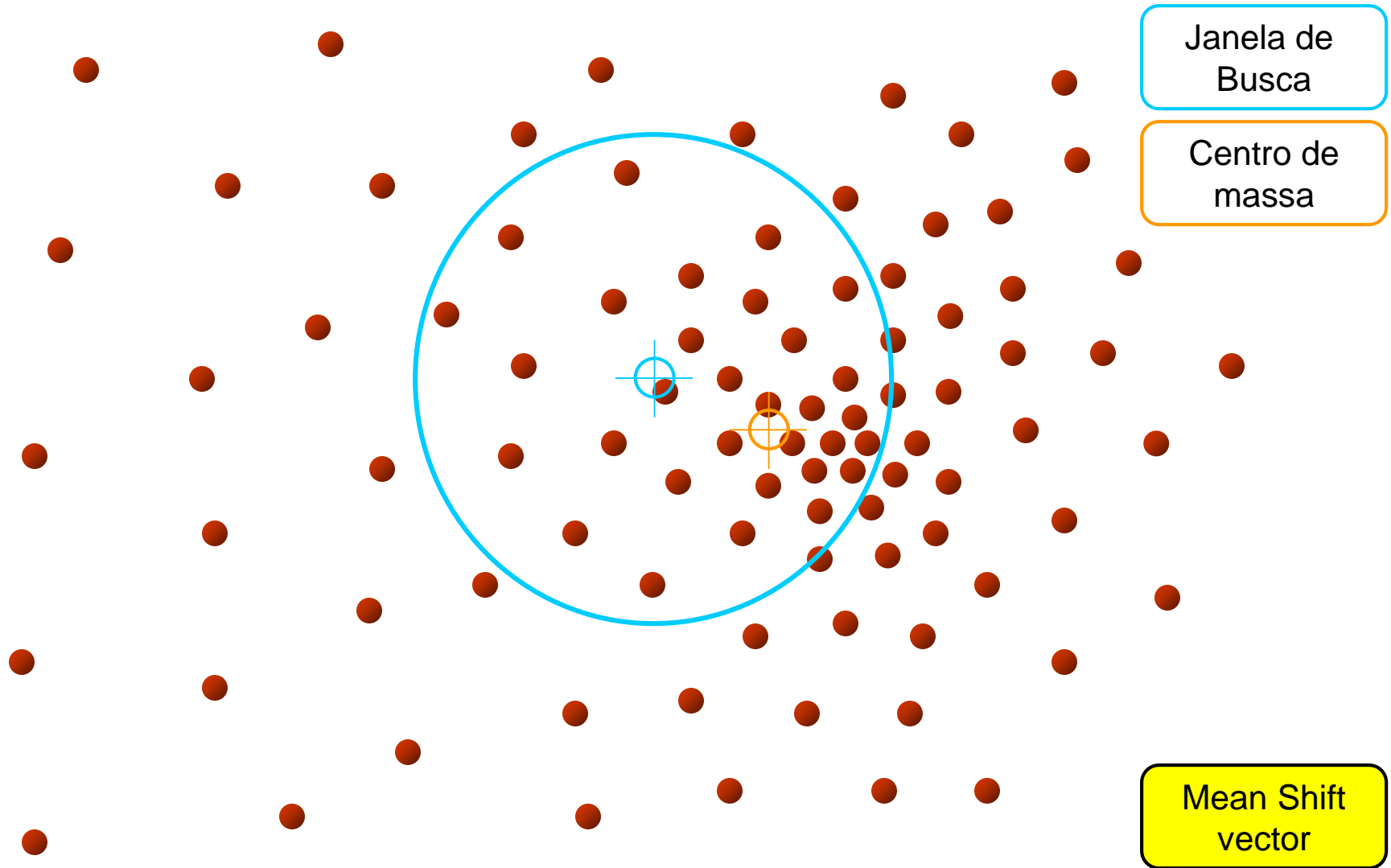
# MEAN SHIFT



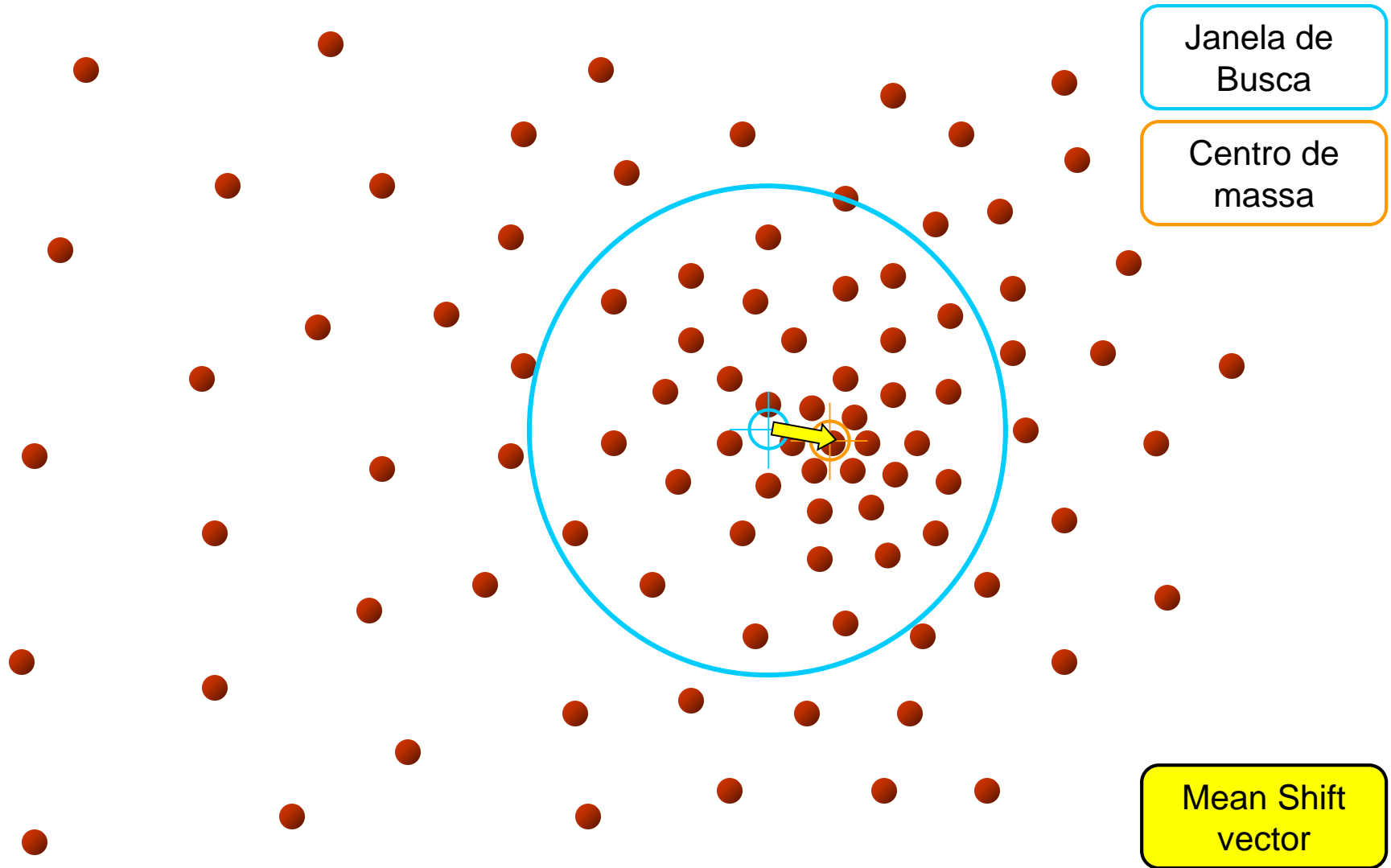
# MEAN SHIFT



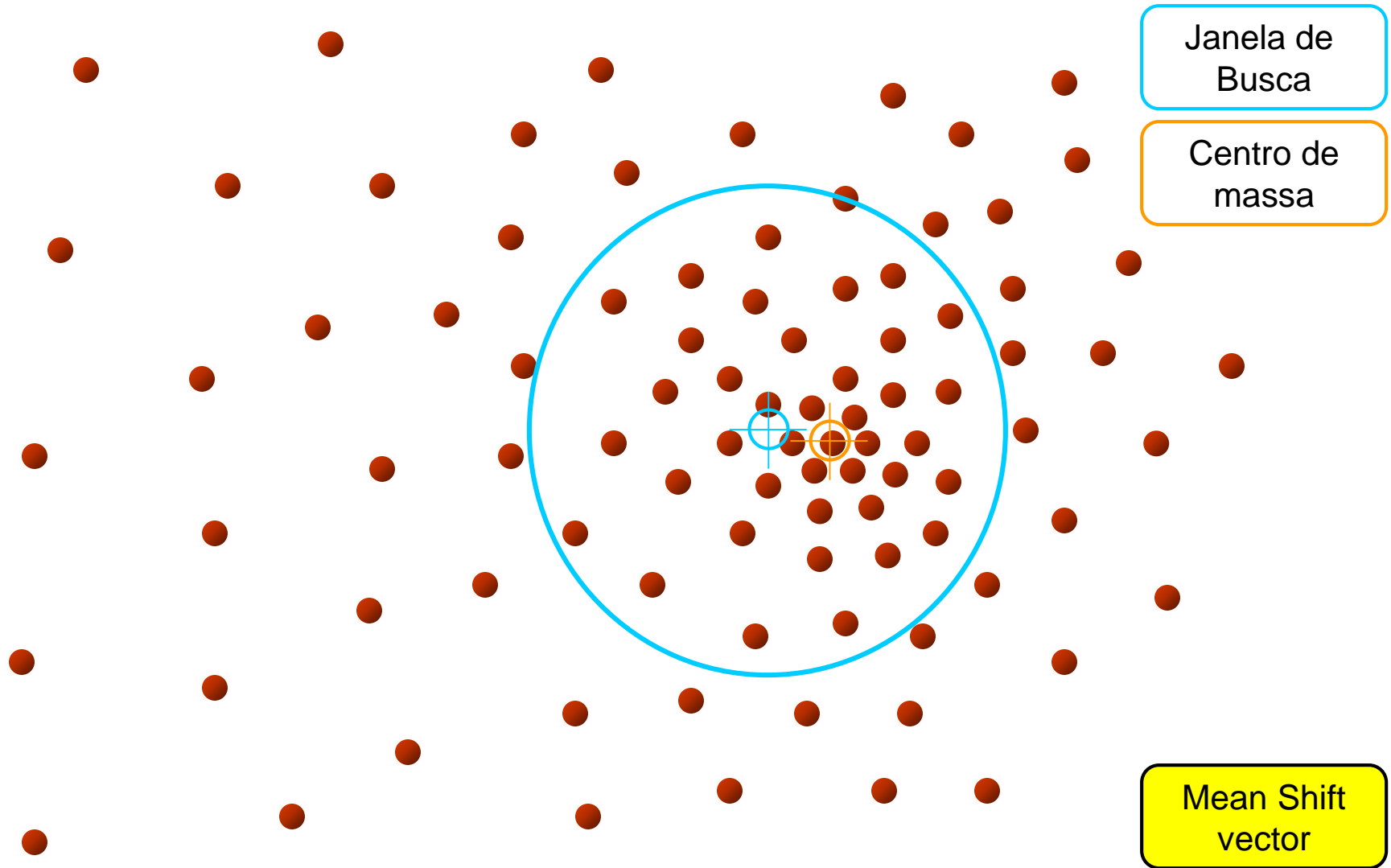
# MEAN SHIFT



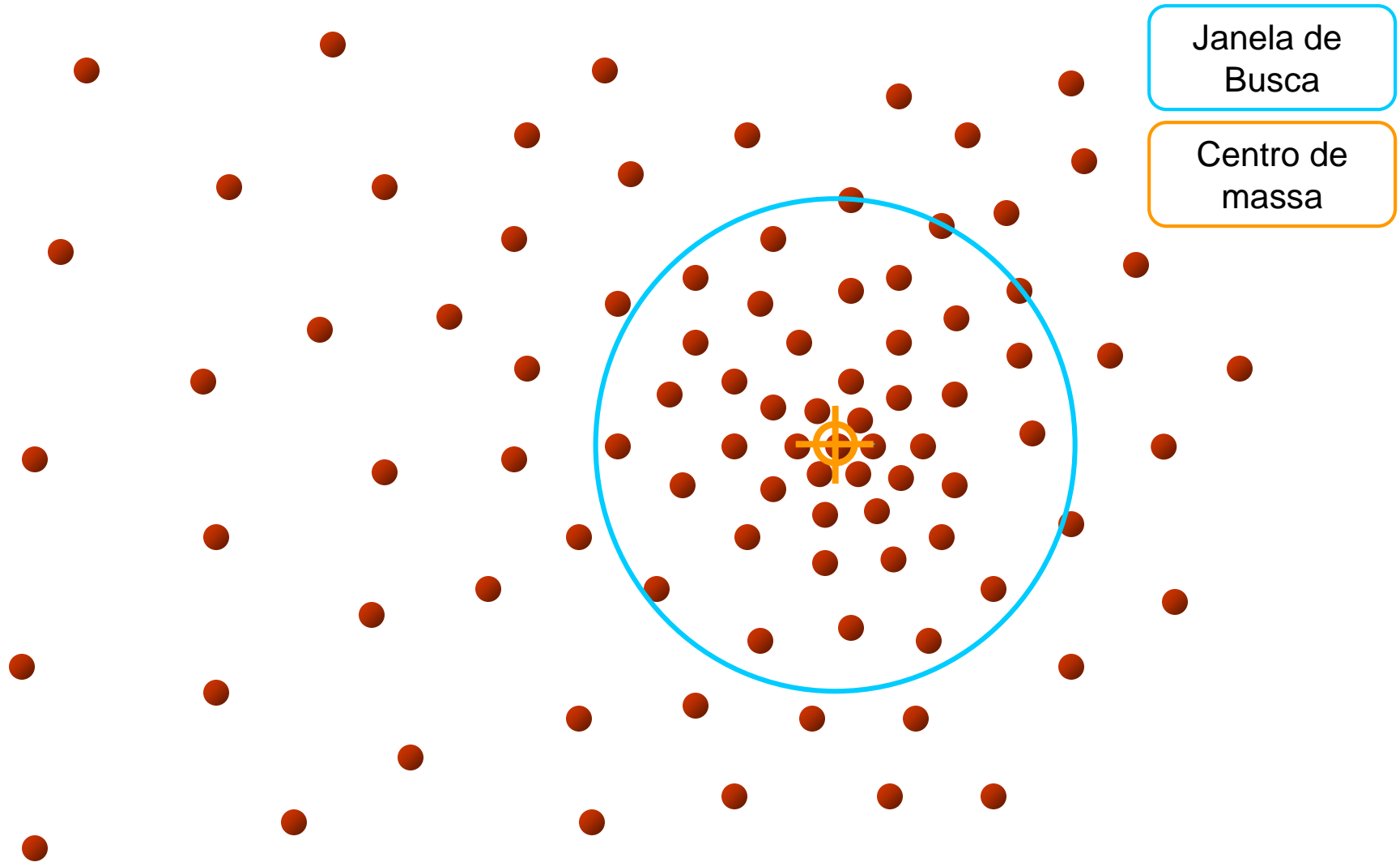
# MEAN SHIFT



# MEAN SHIFT



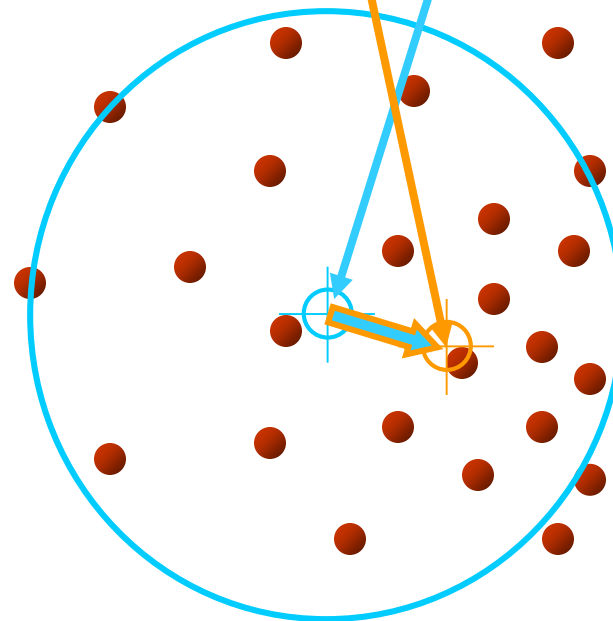
# MEAN SHIFT





# Calculando a Média

$$\nabla P(\mathbf{x}) = \frac{c}{n} \sum_{i=1}^n \nabla k_i = \frac{c}{n} \left[ \sum_{i=1}^n g_i \right] \left[ \frac{\sum_{i=1}^n \mathbf{x}_i g_i}{\sum_{i=1}^n g_i} - \mathbf{x} \right]$$



$$g(\mathbf{x}) = -k'(\mathbf{x})$$

# MEAN-SHIFT

## Algoritmo:

1. Selecionar um pixel  $\mathbf{x}$  da imagem
2. Calcular o vetor da média  $\mathbf{m}(\mathbf{x})$

$$\mathbf{m}(\mathbf{x}) = \frac{\sum_{i=1}^n \mathbf{x}_i g\left(\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{h}\right)}{\sum_{i=1}^n g\left(\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{h}\right)} - \mathbf{x}$$

Tamanho da Janela de Busca

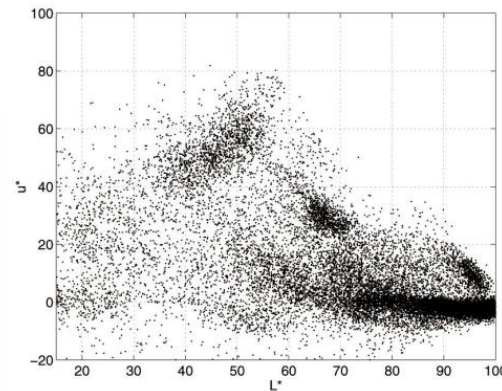
3. Mover a janela do kernel por  $\mathbf{m}(\mathbf{x})$

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{m}(\mathbf{x}_i^t).$$

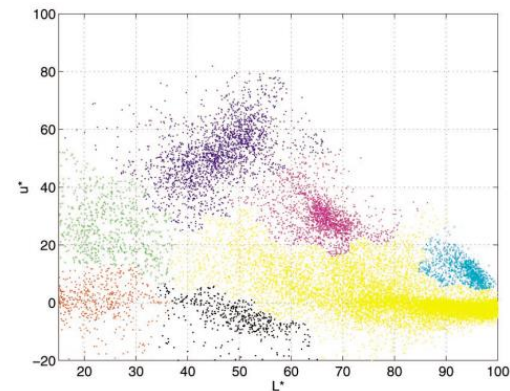
4. Repita as etapas 2 e 3 até que o movimento seja menor do que um pixel por iteração
5. Repita os passos 1 a 4 para cada pixel na imagem.
6. Atribua todos os pixels para os quais  $\mathbf{x}$  termina no mesmo ponto (dentro do raio  $h$ ) para o mesmo cluster.

# MEAN-SHIFT

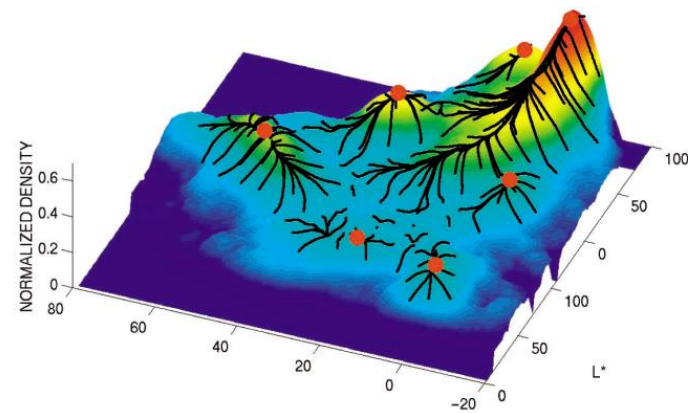
- Encontre as características (Cor, gradiente, textura, etc.)
- Iniciar a janela de busca com pontos individuais de características
- Executar mean shift para cada janela até convergir
- Fundir janelas que acabam próximas do mesmo "pico" ou moda



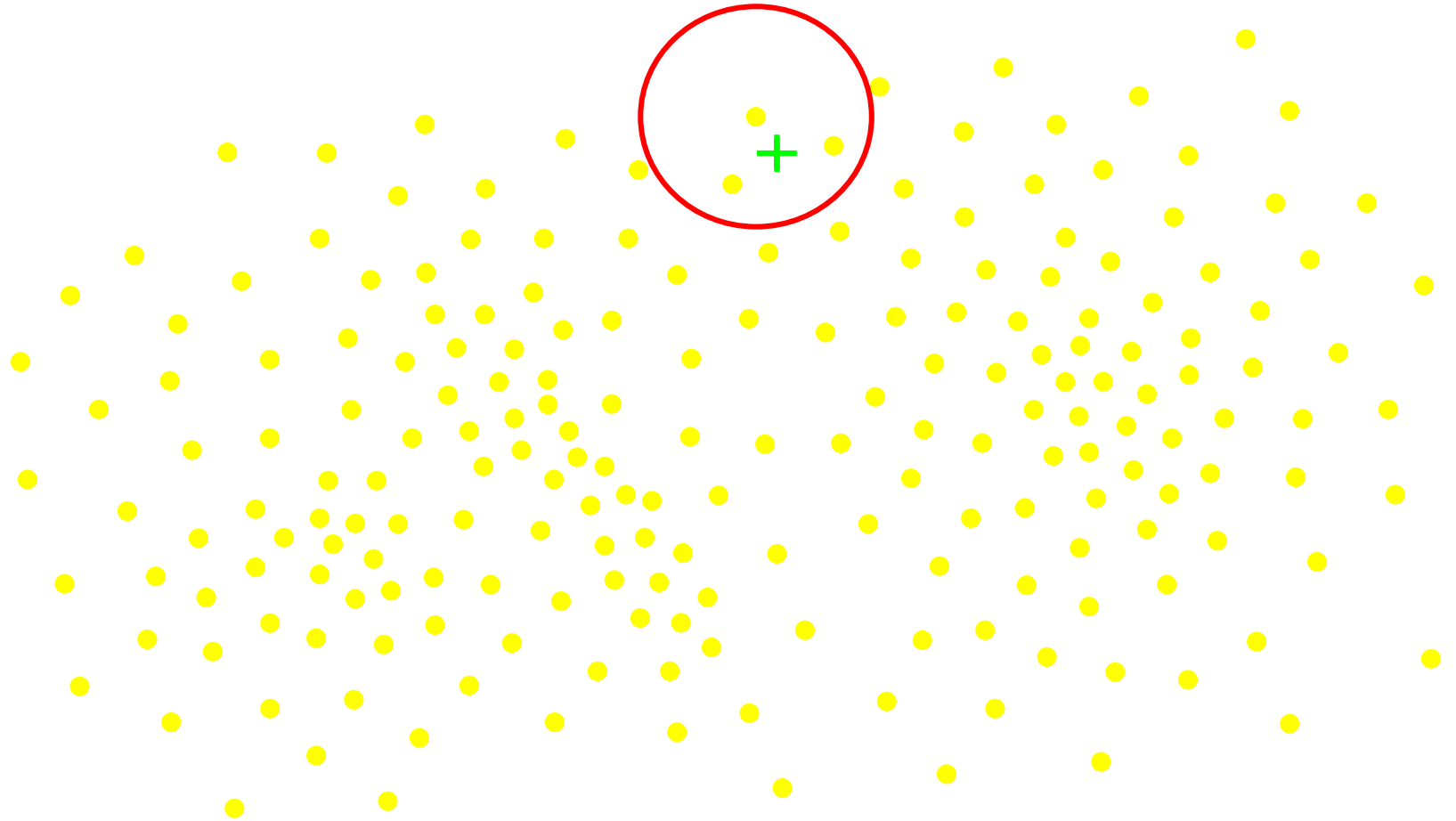
(a)



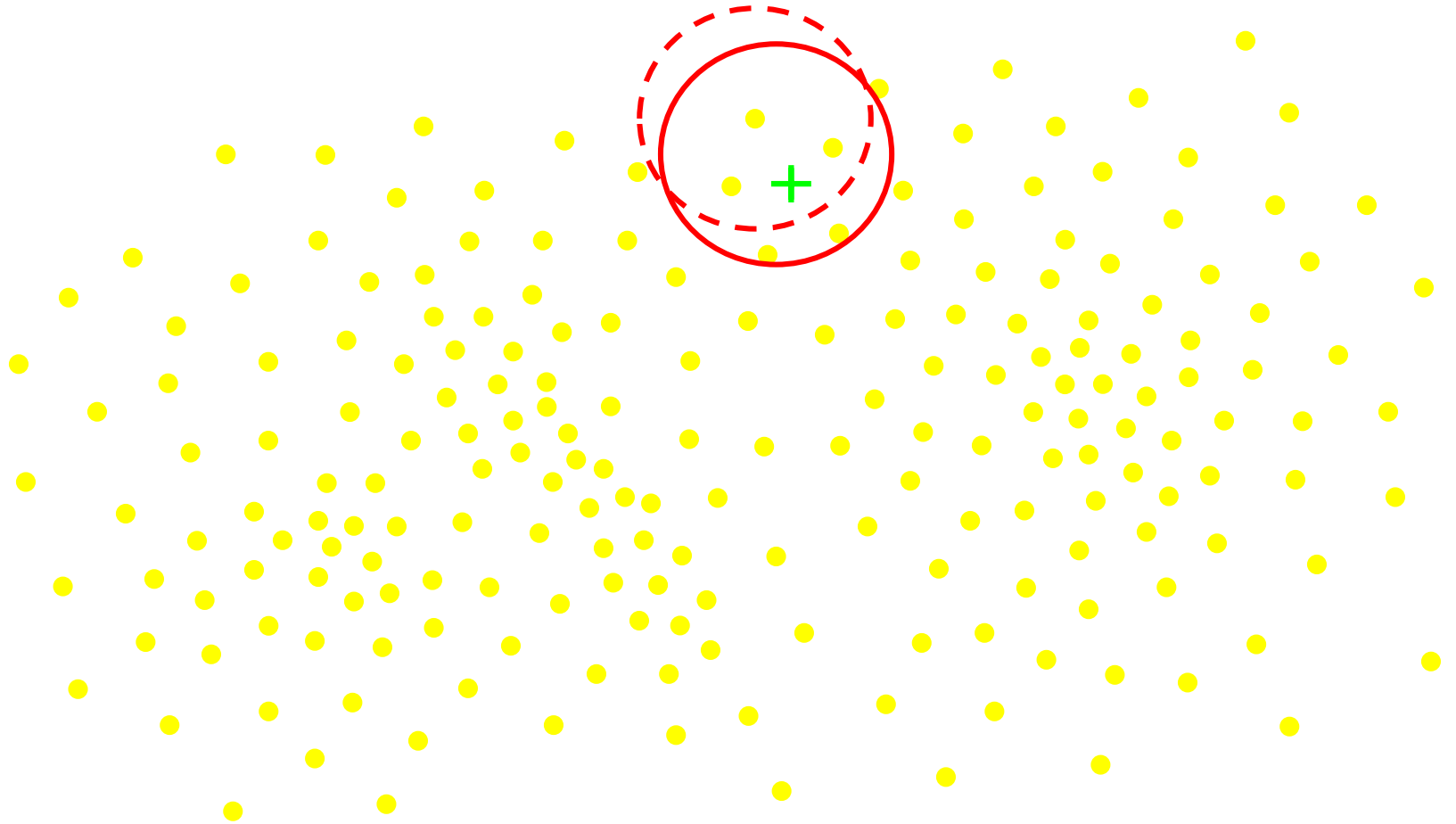
(b)



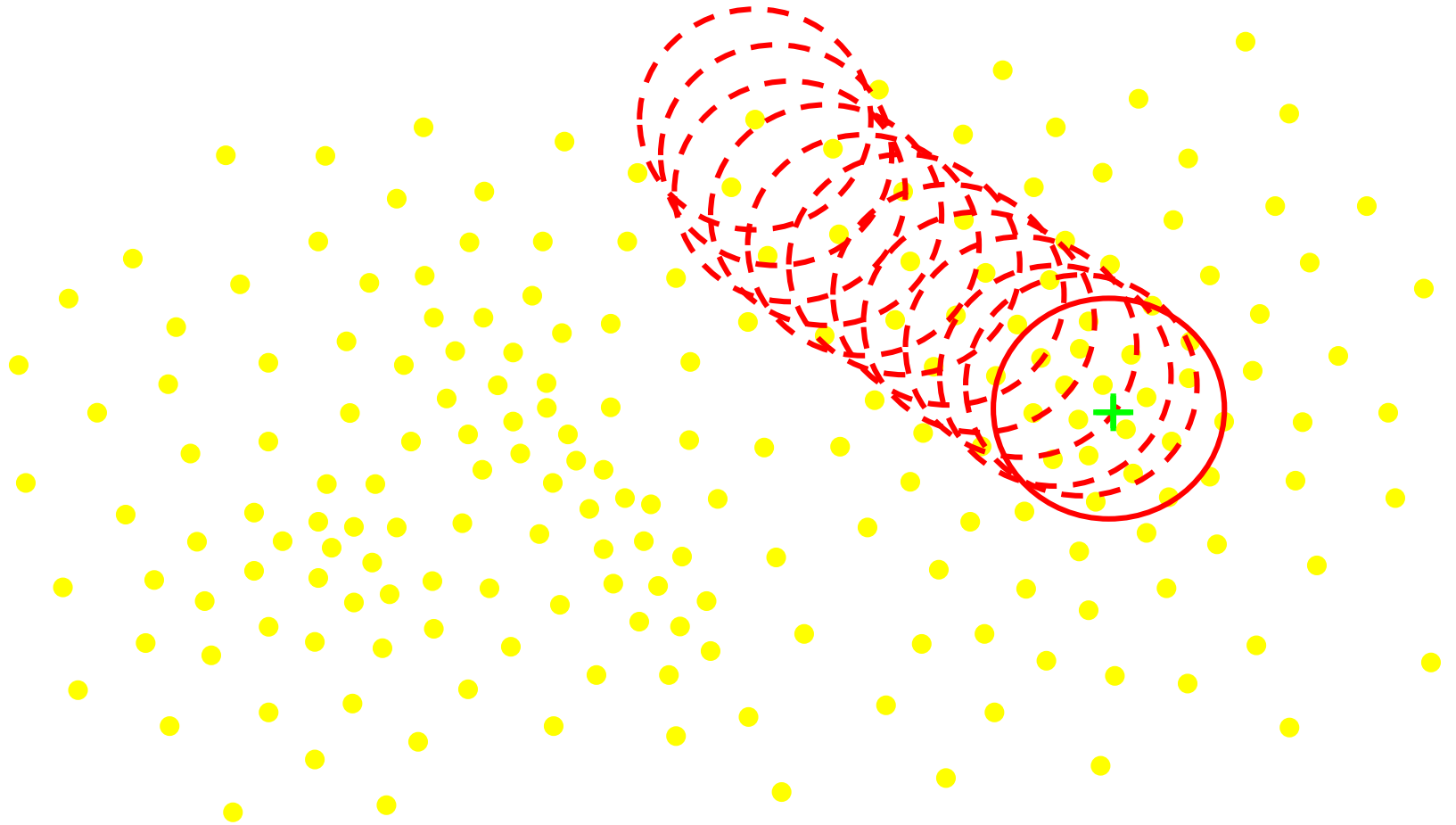
# MEAN SHIFT CLUSTERING



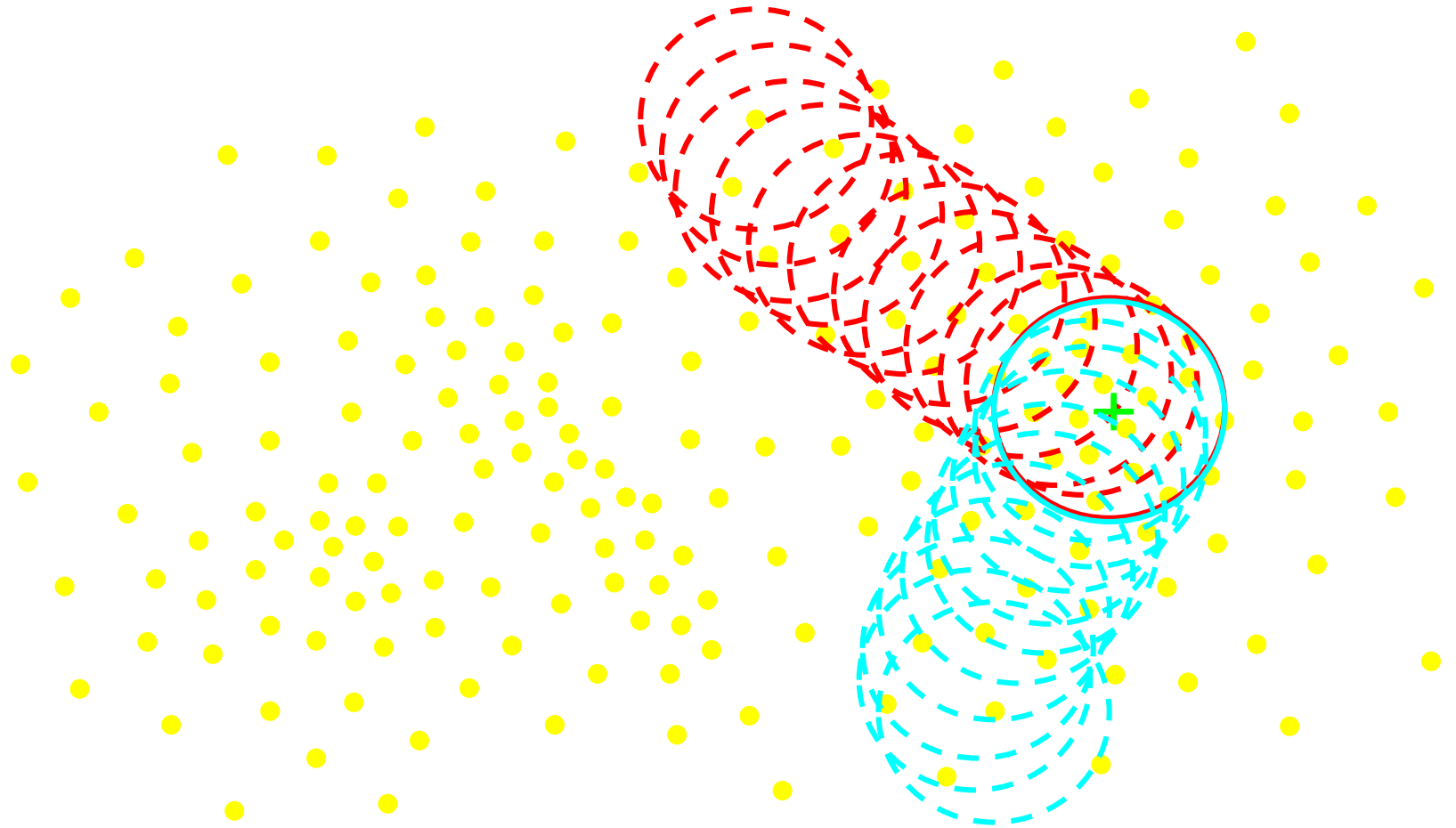
# MEAN SHIFT CLUSTERING



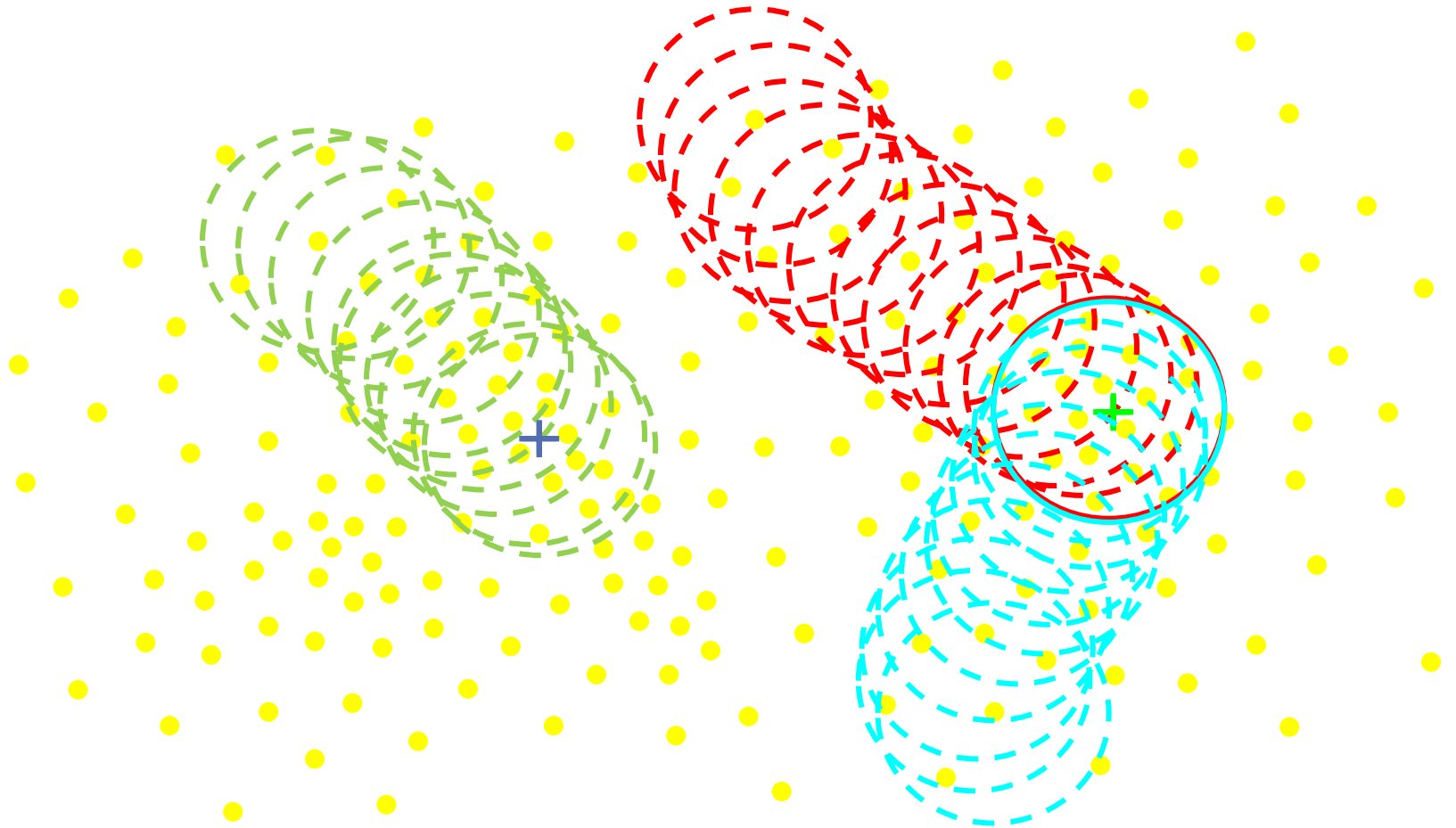
# MEAN SHIFT CLUSTERING



# MEAN SHIFT CLUSTERING

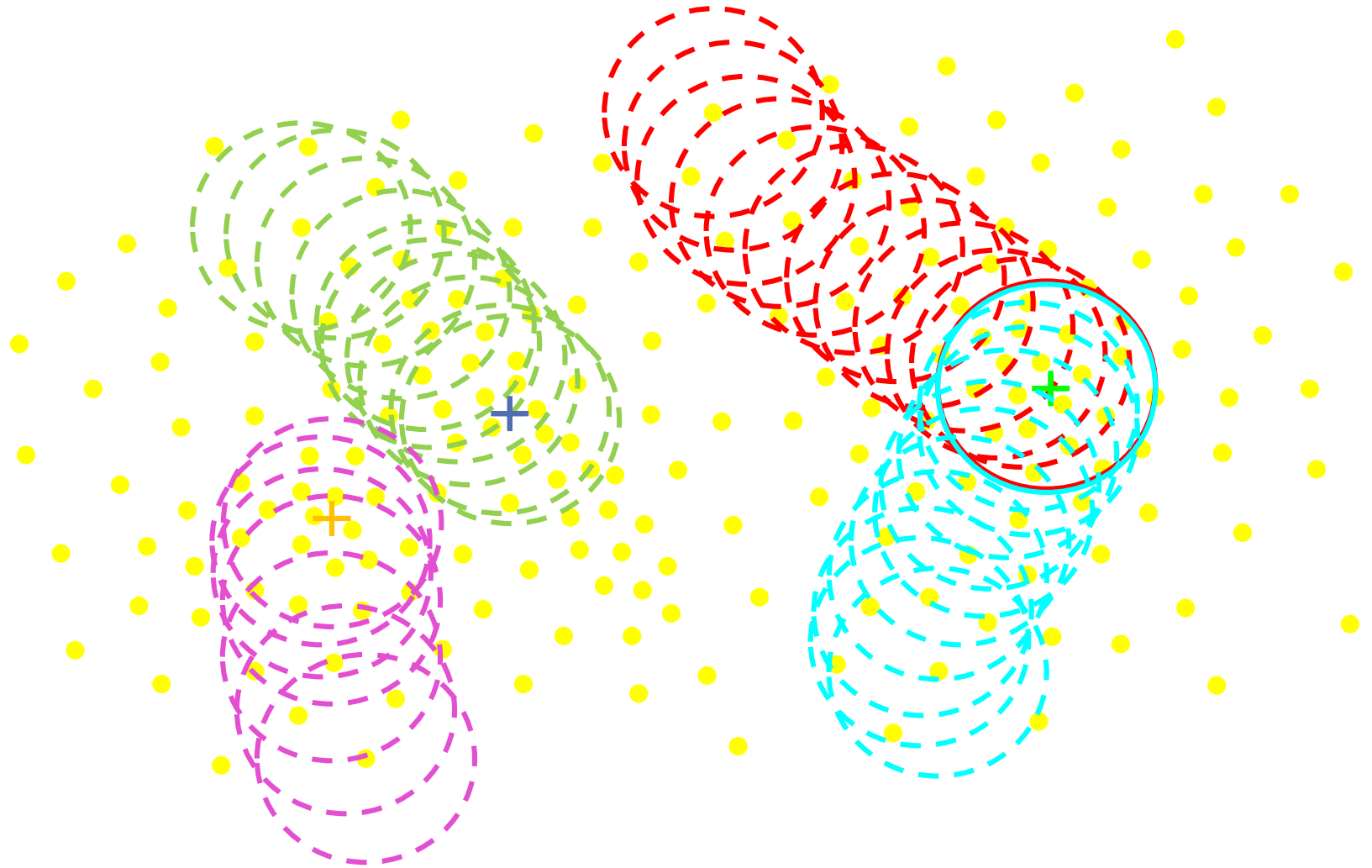


# MEAN SHIFT CLUSTERING

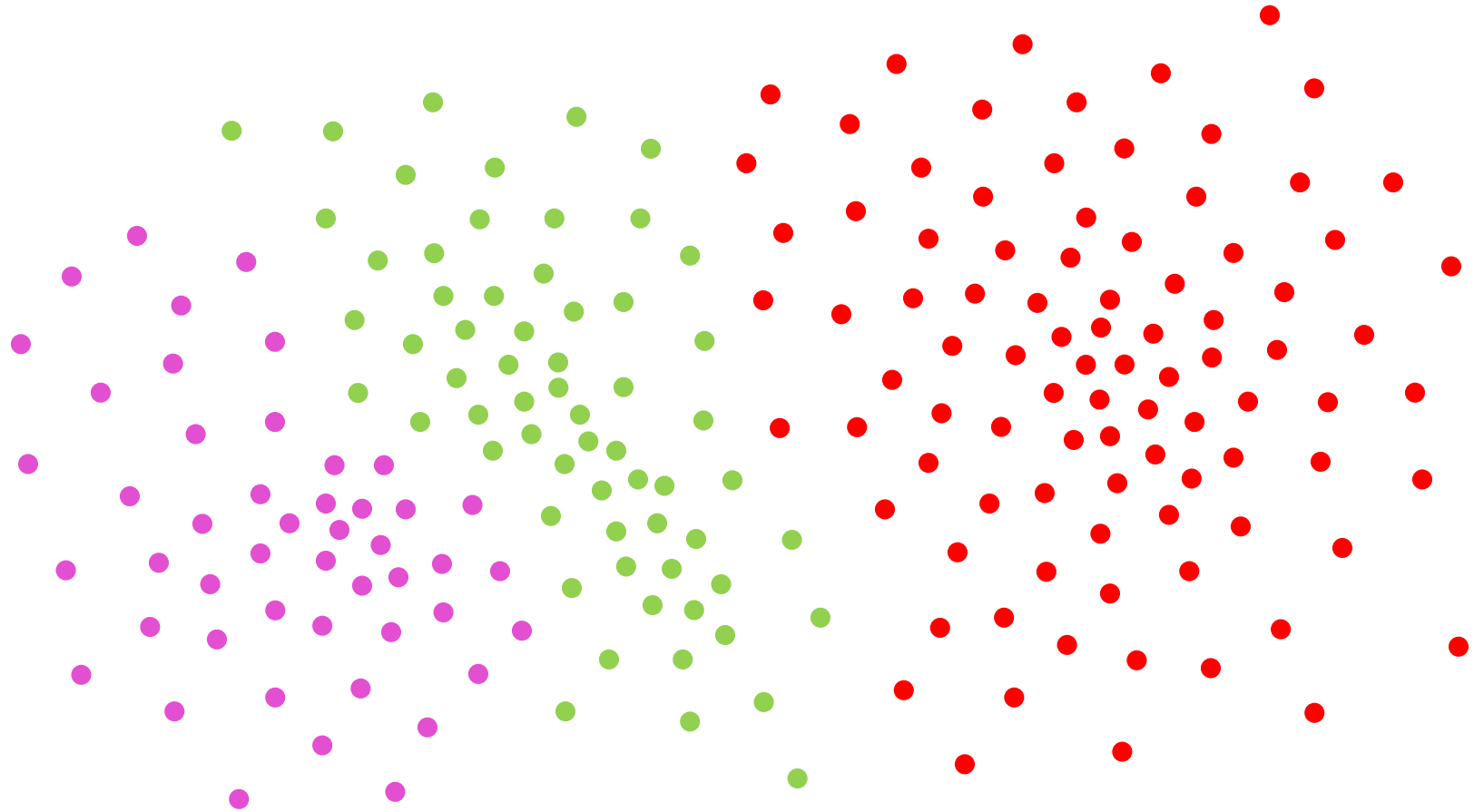




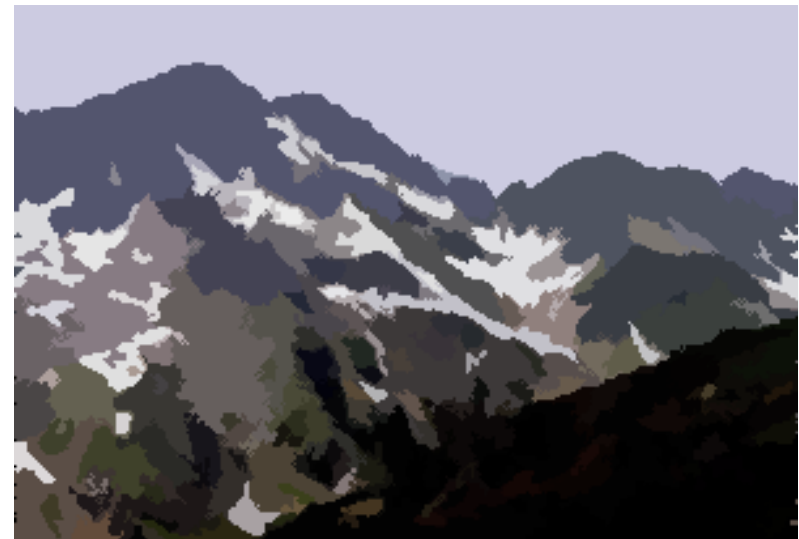
# MEAN SHIFT CLUSTERING



# MEAN SHIFT CLUSTERING



# RESULTADOS COM MEAN-SHIFT



# RESULTADOS COM MEAN-SHIFT





# RESULTADOS COM MEAN-SHIFT



# VANTAGENS E DESVANTAGENS DO MEAN-SHIFT

- **Vantagens**

- Não assume clusters esféricos
- Apenas um único parâmetro (tamanho da janela)
- Encontra um número variável de tons
- Robusto para outliers

- **Desvantagens**

- Saída depende do tamanho da janela
- Computacionalmente caro
- Não escala bem com dimensão de espaço de características

# **QUANDO UTILIZAR**

- **Super segmentação**
- **Segmentações múltiplas**
- **Rastreamento e outras aplicações de agrupamento**

# MOVIMENTO

- O movimento é uma poderosa pista usada pelos seres humanos e animais para a extração de um objeto de interesse de um fundo de detalhes irrelevantes
- O movimento de objetos em “frames” de vídeo fornece meio para segmentação do objeto e/ou do fundo da cena.
- **Abordagem básica**
  - A abordagem mais simples para a detecção de mudanças entre dois quadros de imagem  $f(x,y,t_i)$  e  $f(x,y,t_j)$  tomados em instantes  $t_i$  e  $t_j$ , respectivamente, é através da comparação de imagens pixel a pixel.



# MOVIMENTO

$$d_{ij}(x, y) = \begin{cases} 1 & \text{se } |f(x, y, t_i) - f(x, y, t_j)| > \Theta \\ 0 & \text{caso contrario} \end{cases}$$

- **Abordagem básica:**
  - Essa abordagem pode ser aplicada quando a iluminação é relativamente constante
  - Além de objetos em movimento, a imagem da diferença pode contar com pequenos ruídos e objetos que se movem vagarosamente

# MOVIMENTO

- **Subtração do fundo**
  - Fundo Simples: Obtém-se um modelo do fundo que é subtraído de cada quadro
  - Fundo Complexo: Modelo de Mistura de Gaussianas (GMM)

# MOVIMENTO

- **Fundo Simples:**

- Fazer a média (mediana) de várias imagens afim de criar uma aproximação do fundo
- Subtrair os quadros subsequentes desta aproximação
- Objetos que não fazem parte do fundo da cena apareceram como resultado desta operação de segmentação
- Este método é muito popular para operações de rastreamento em tempo real por ser simples e rápido. No entanto, tem dois grandes inconvenientes. Ele não pode lidar com ruídos intermitentes (por exemplo, mudanças de iluminação) e pode gerar objetos fantasmas.

038 MONITOR 0004  
LIVERPOOL STREET  
TICKET OFFICE B



person 4



person 16



person 17



person 2



person 1



person 5



person 3

**FIM**

## SEGMENTATION

