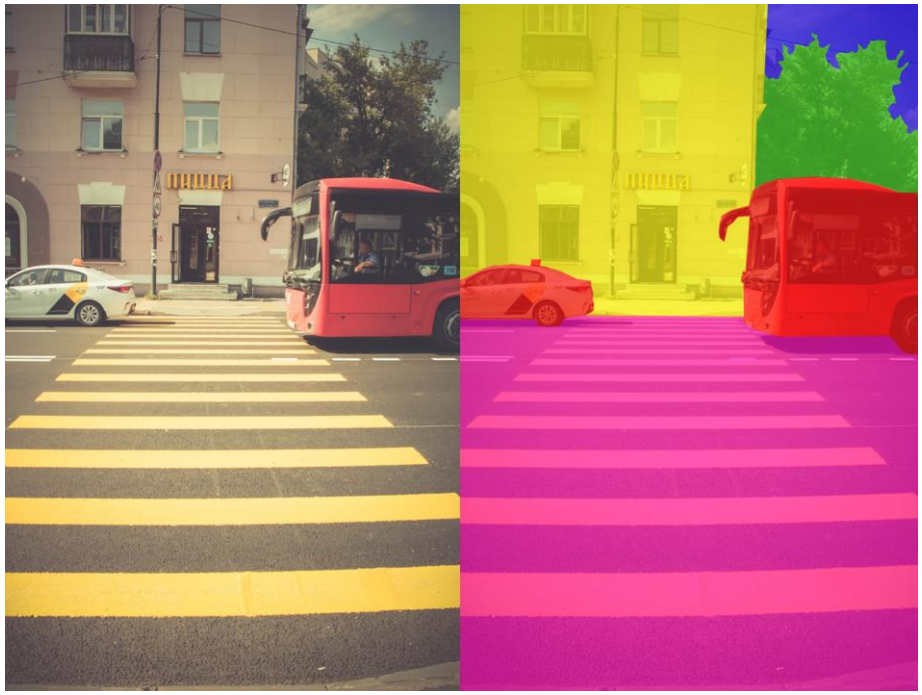
# Segmentação de Imagens

# Segmentação de Imagens

- Segmentação semântica: particionamento de uma imagem em classes distintas

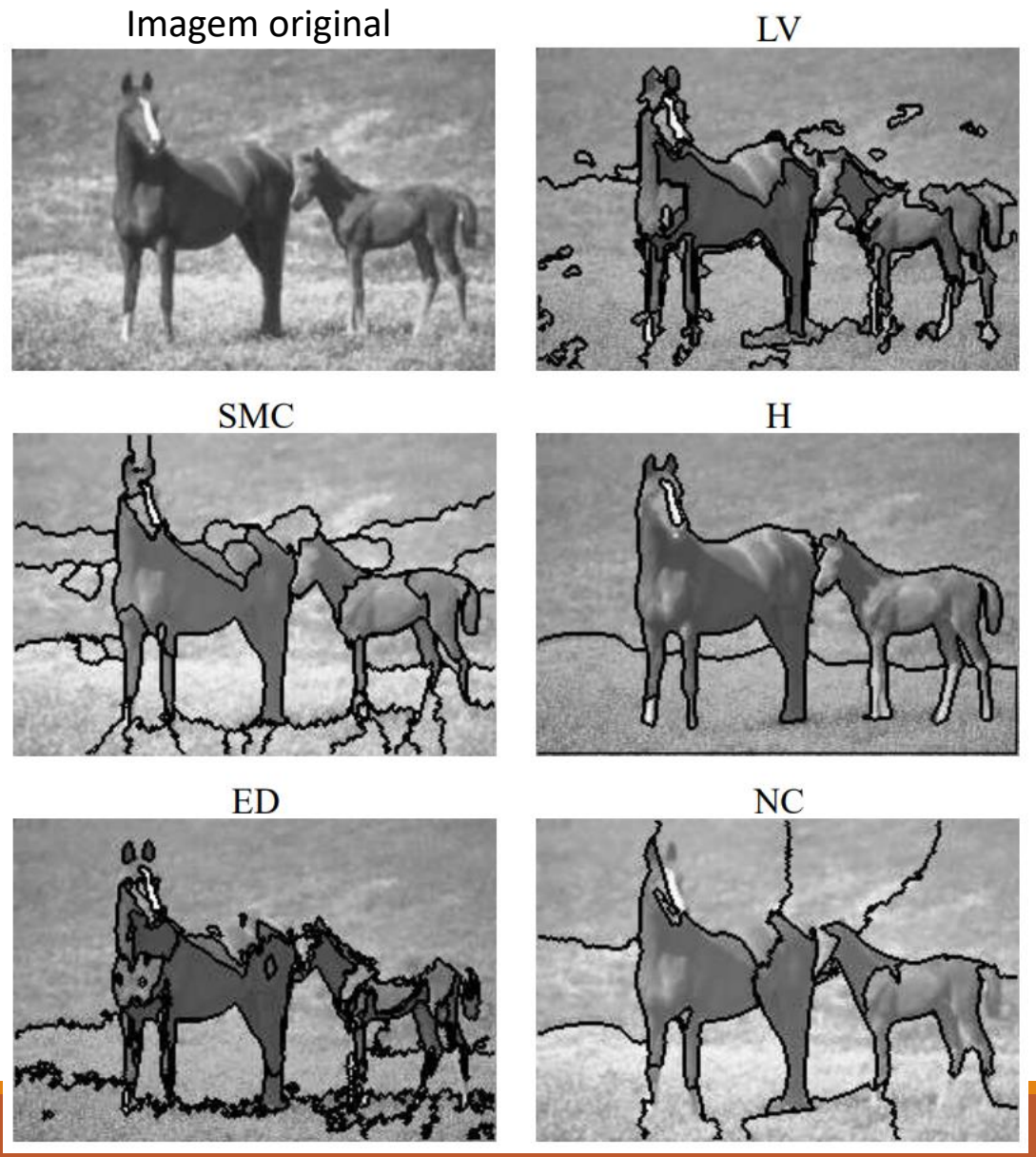


# Segmentação de Imagens

- Segmentação semântica: particionamento de uma imagem em classes distintas
- Segmentação de instância: identificação de todos os píxeis pertencentes a determinados objetos de uma imagem

# Segmentação de Imagens

- Diferentes critérios podem levar a diferentes resultados de segmentação



# Segmentação de Imagens

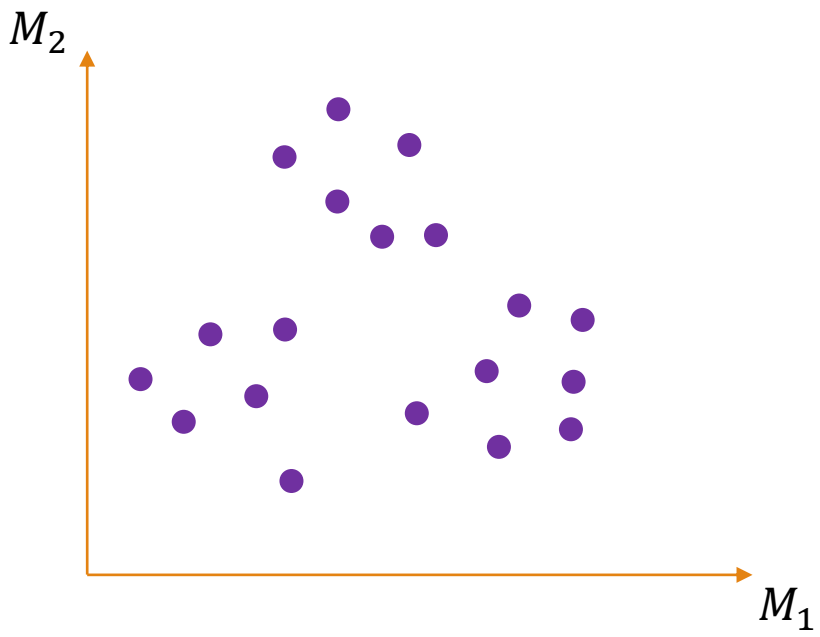
- Vimos ao longo do curso alguns procedimentos que podem ser utilizados para a tarefa de segmentação de imagens:
  - Detecção de pontos (ex: utilizando filtro Laplaciano)
    - Segmentação de objetos pequenos aproximadamente circulares
  - Detecção de bordas (ex: magnitude do gradiente)
    - A partir da borda, podemos encontrar o objeto
  - Detecção de retas (ex: transformada Hough)
    - Conjunto de retas pode formar um objeto
  - Cálculo do limiar ótimo (ex: método Otsu)
    - Componentes conexos representam objetos candidatos
  - Caracterização de textura (ex: GLCM)
    - Limiar sobre uma propriedade do GLCM leva a uma imagem binária

# Segmentação de Imagens

- Veremos algumas outras técnicas de segmentação:
  - Algoritmo K-médias sobre descritores
  - Algoritmo Mean shift
  - Modelos de mistura de gaussianas
  - Crescimento de região
  - Contornos ativos

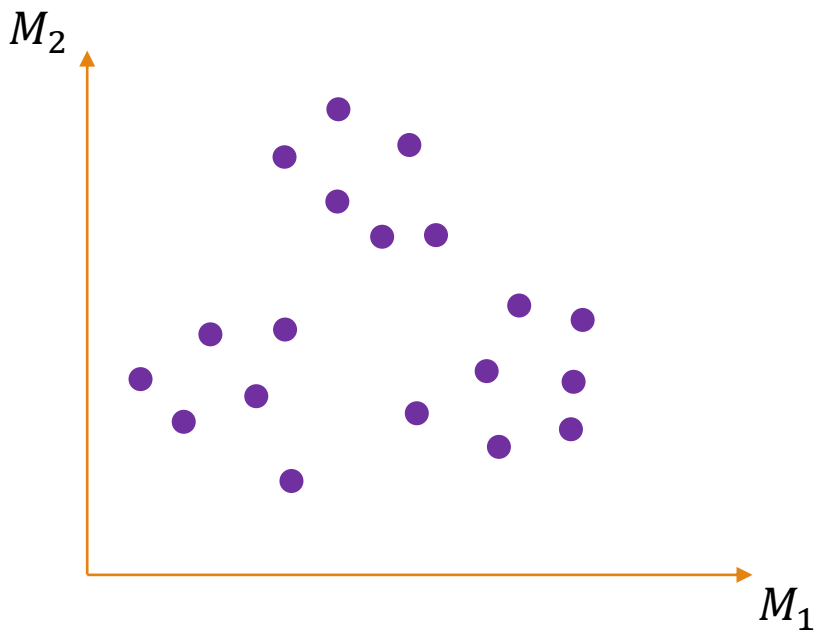
# Segmentação utilizando o algoritmo k-médias

# Clusterização k-médias



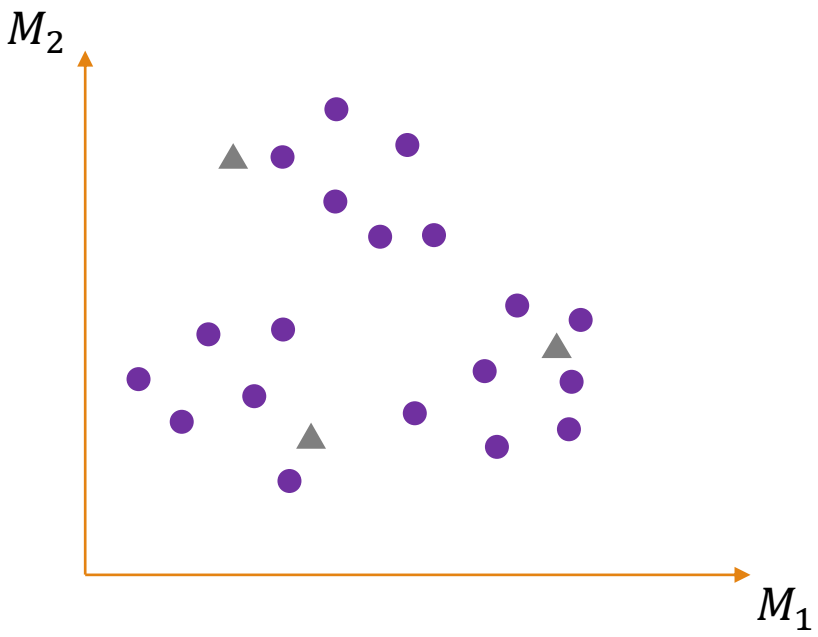

- Vamos imaginar que temos duas medidas associadas a um conjunto de objetos (ex: área e circularidade)
- Cada objeto pode ser representado como um ponto em um espaço 2D de medidas

# Clusterização k-médias



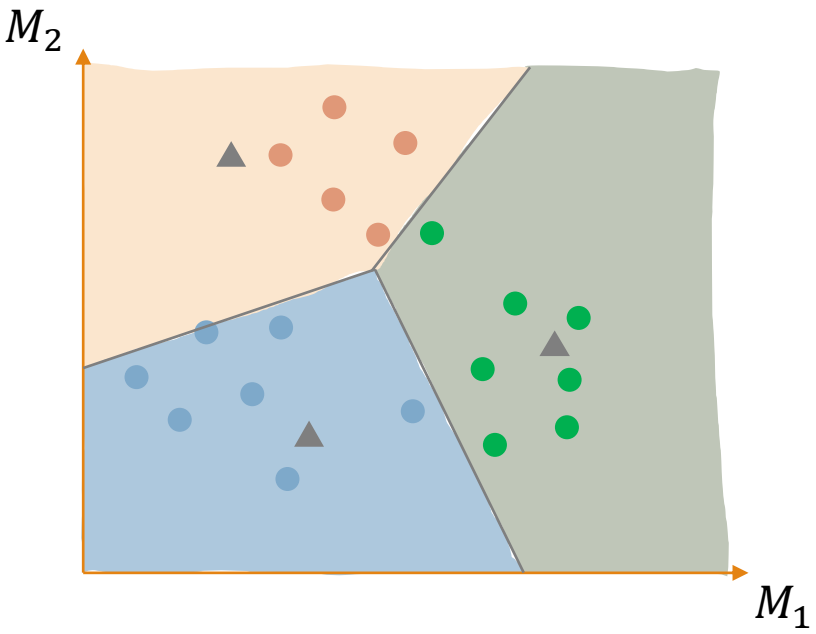

- Queremos identificar as classes às quais os pontos acima pertencem. Não conhecemos a classe de nenhum ponto (isto é, não há treinamento)
- Essa atividade é chamada de aprendizado não supervisionado

# Clusterização k-médias



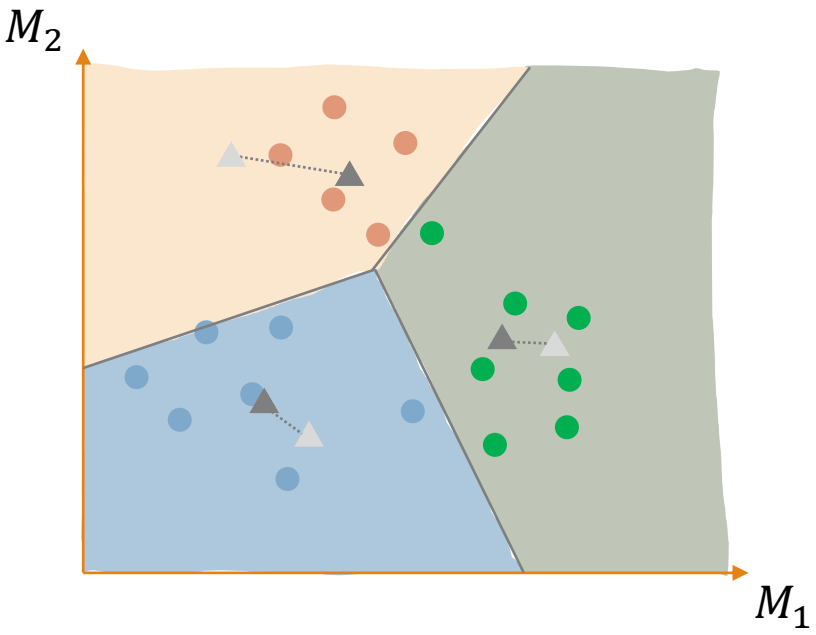

- Escolhemos um número de sementes, nesse caso, 3 sementes (k=3)
- A posição inicial das sementes é escolhida aleatoriamente

# Clusterização k-médias



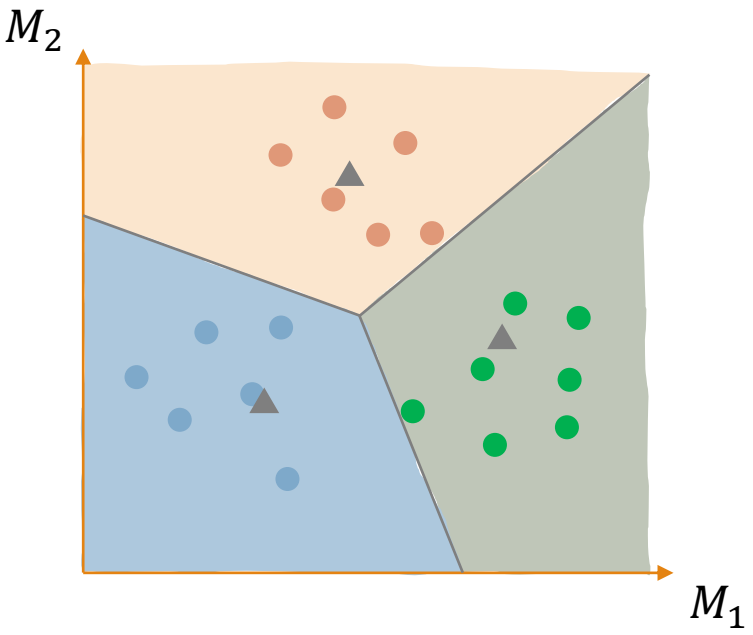

- O espaço é particionado em três regiões
- Cada região contém os pontos que estão mais próximos da semente contida na região do que qualquer outra semente
- Esse particionamento é conhecido como tesselação de Voronoi

# Clusterização k-médias



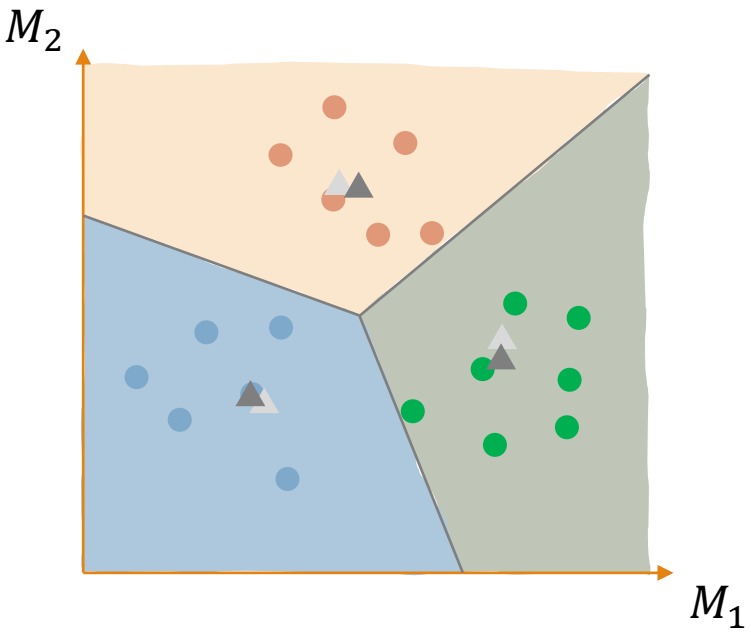

- A média das posições dos pontos pertencentes a cada região é calculada
- As sementes são reposicionadas para as posições médias calculadas

# Clusterização k-médias



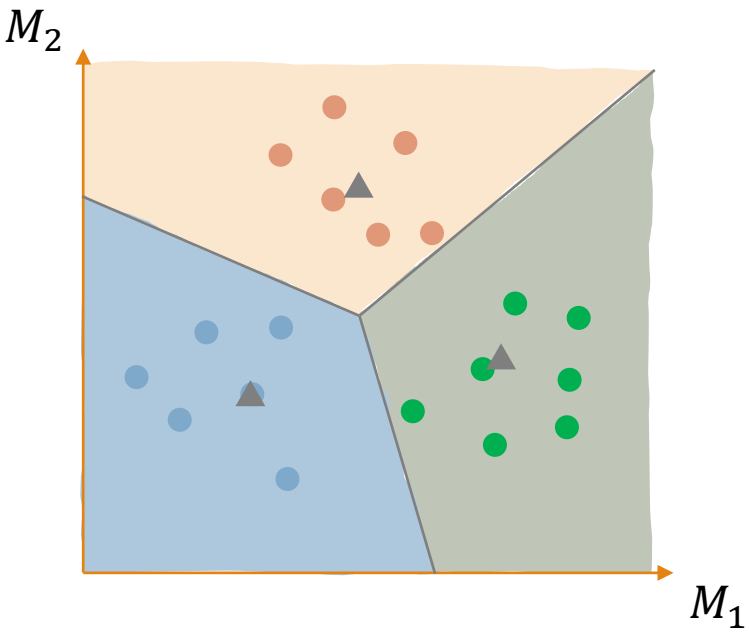

Uma nova partição do espaço é calculada

# Clusterização k-médias



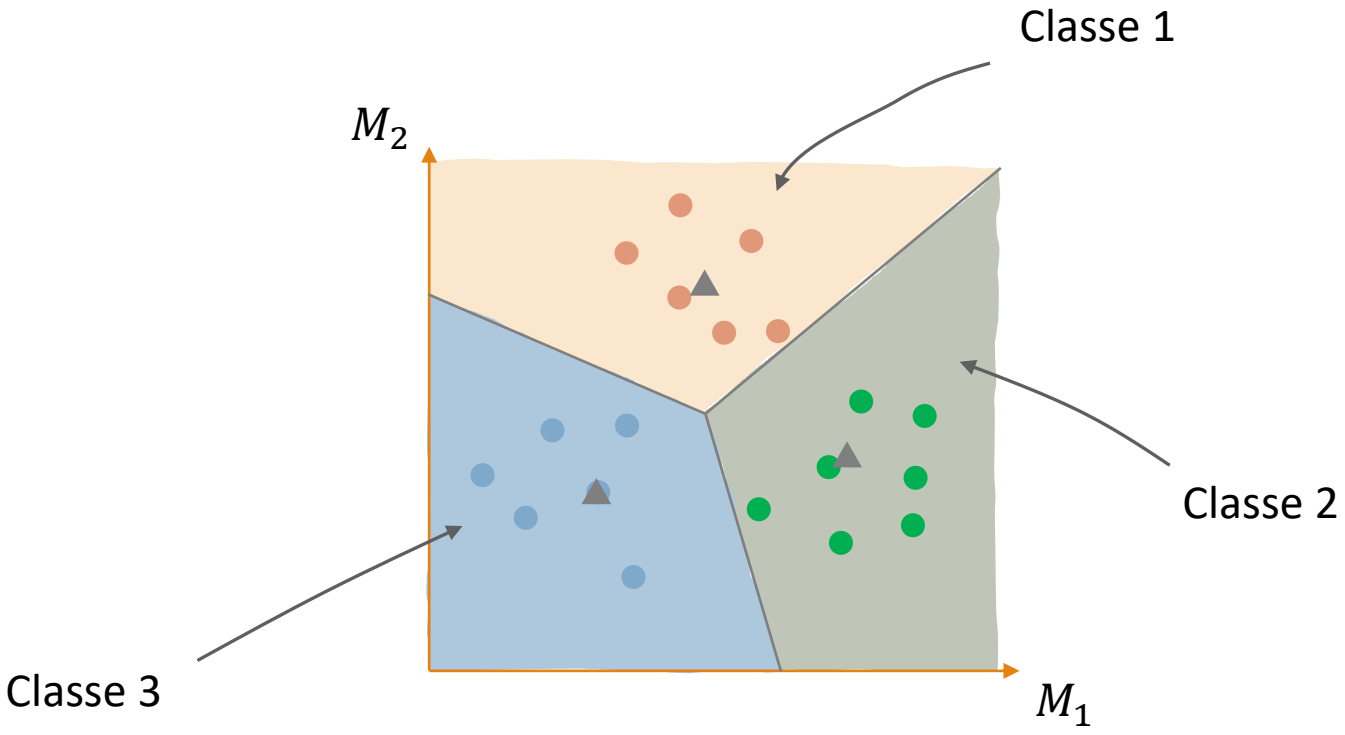

As sementes são movidas para a nova média dos pontos

# Clusterização k-médias




O processo continua até que as posições das sementes não mude consideravelmente, ou até que um número de iterações seja atingido

# Clusterização k-médias



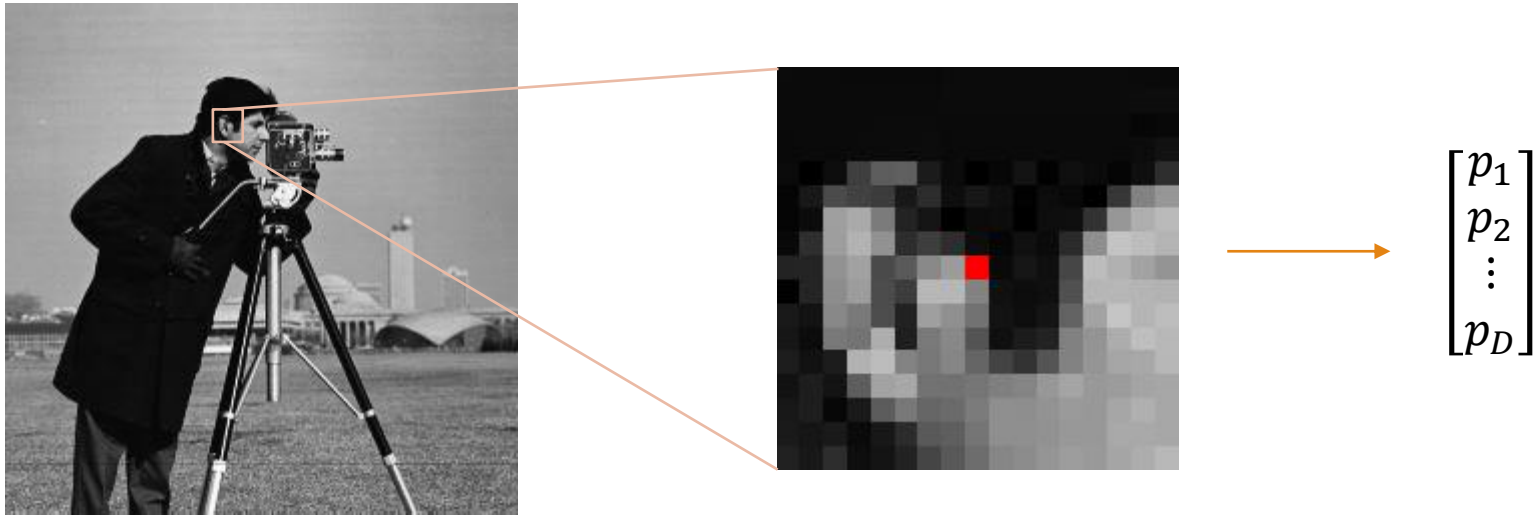
# Clusterização k-médias

Se as propriedades dos objetos possuem unidades diferentes, precisamos lembrar de normalizar os atributos

$$\tilde{x} = \frac{x - \mu_x}{\sigma_x}$$

# Segmentação utilizando k-médias

- A clusterização k-médias pode ser utilizada para segmentar imagens
- A ideia é caracterizar pixels ou regiões da imagem utilizando um conjunto de atributos, gerando um espaço de atributos de dimensão $D$. Em seguida, o algoritmo k-médias é aplicado aos valores calculados.



# Segmentação utilizando k-médias

- Para imagens coloridas, em muitos casos podemos aplicar o k-médias diretamente nas cores da imagem
- Cada cor representa uma propriedade, então em uma imagem RGB temos um espaço 3D onde cada pixel possui uma posição específica nesse espaço

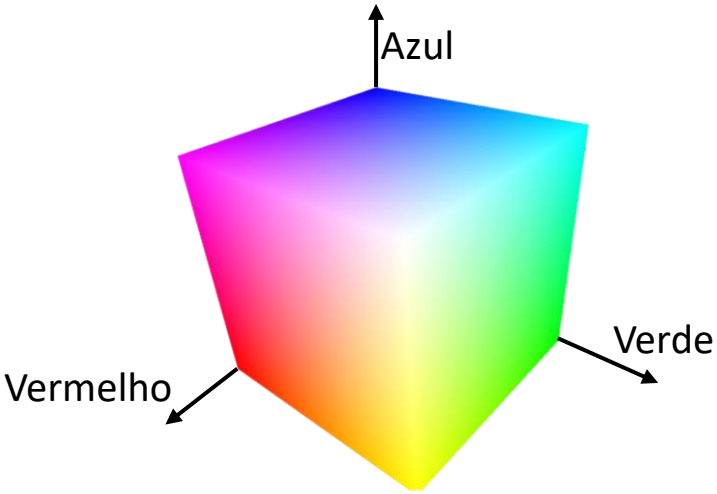
# Segmentação utilizando k-médias

Exemplo de segmentação utilizando o k-médias aplicado sobre as cores. Foram consideradas 3 sementes

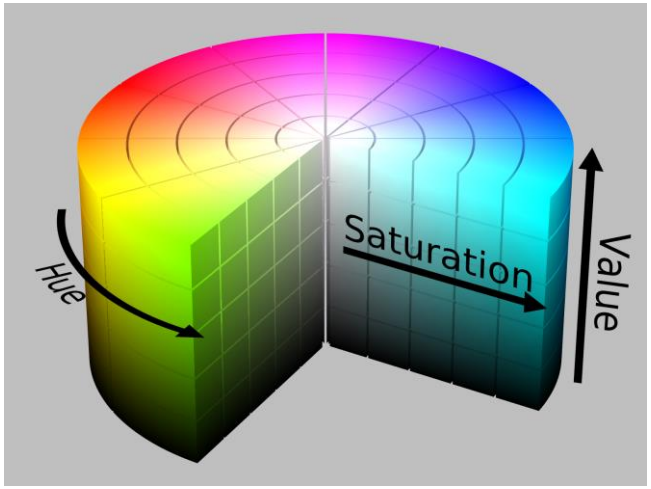
# Segmentação utilizando k-médias – Diferentes espaços de cores

- Imagens podem ser representadas em diferentes espaços de cores

Espaço de cores RGB




Espaço de cores HSV

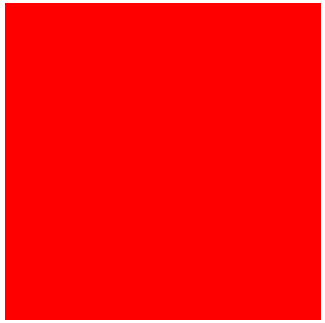


# Segmentação utilizando k-medias – Espaço de cores L*a*b*

- O espaço de cores L*a*b* é definido de forma a ser perceptualmente uniforme, isto é, uma variação $\Delta x$ de uma cor corresponde a uma variação similar na forma como percebemos essa cor
  - Isso não ocorre no espaço RGB. Por exemplo, para cores bem escuras, uma variação $\Delta x$ da cor pode nem mesmo ser percebida visualmente

# Segmentação utilizando k-medias – Espaço de cores L*a*b*

Cor (255, 0, 0)

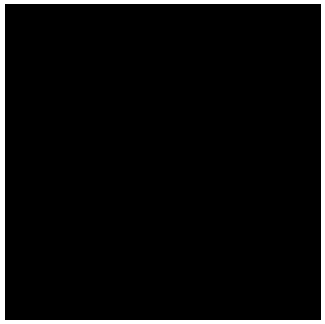


Cor (200, 0, 0)

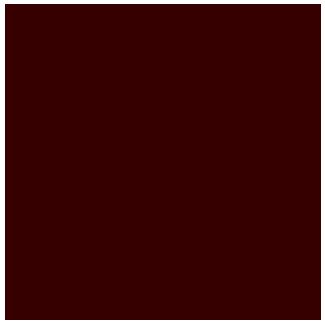
-55 de vermelho

A variação de vermelho é percebida de forma similar nessas imagens?

Cor (0, 0, 0)



Cor (55, 0, 0)



+55 de vermelho

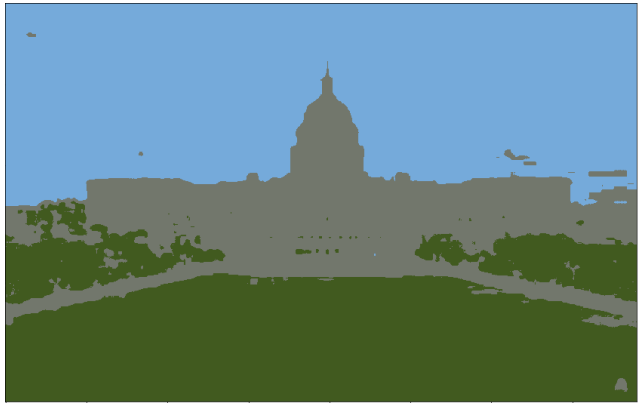
# Segmentação utilizando k-medias – Espaço de cores L*a*b*

- Quando realizamos segmentação utilizando cores, utilizar o espaço de cores L*a*b* pode proporcionar um resultado superior ao espaço RGB
- Usualmente, utilizamos os canais a* e b* desse espaço de cores

# Segmentação utilizando k-médias

K-médias em RGB

K-médias em L*a*b*



# Segmentação utilizando k-médias

Notebook “**Segmentação k-médias**”

# Mean shift

# Mean shift

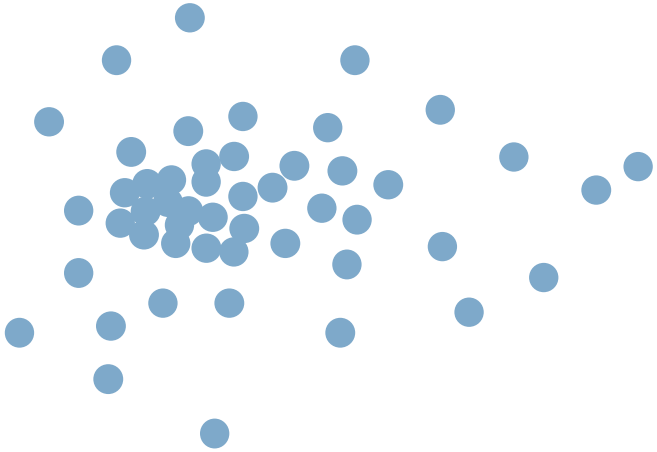
- Mean shift é uma técnica utilizada para a identificação do máximo de uma função de densidade de probabilidade
  - É um algoritmo conhecido de aprendizado de máquina
  - Portanto, ele pode ser utilizado para diversas outras aplicações além de processamento de imagens

- Ele é um método não-paramétrico, isto é, não é preciso especificarmos um modelo para os dados
- Ele é não-supervisionado, ou seja, não é preciso utilizarmos uma base de imagens para treinamento

# Mean shift

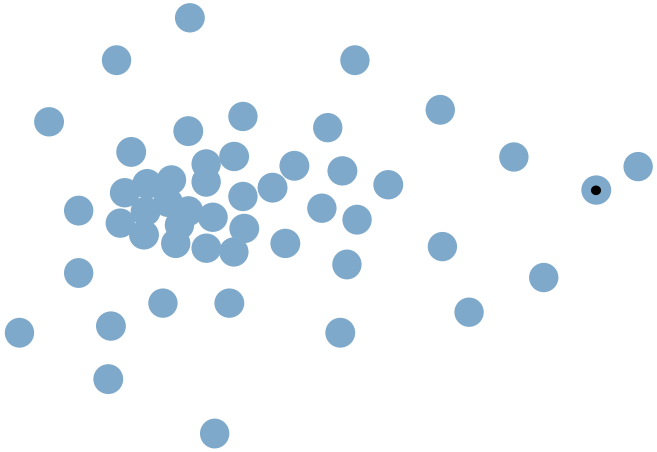
- Uma propriedade interessante do mean shift é que não é necessário saber o número de clusters na imagem

# Mean shift – Ilustração



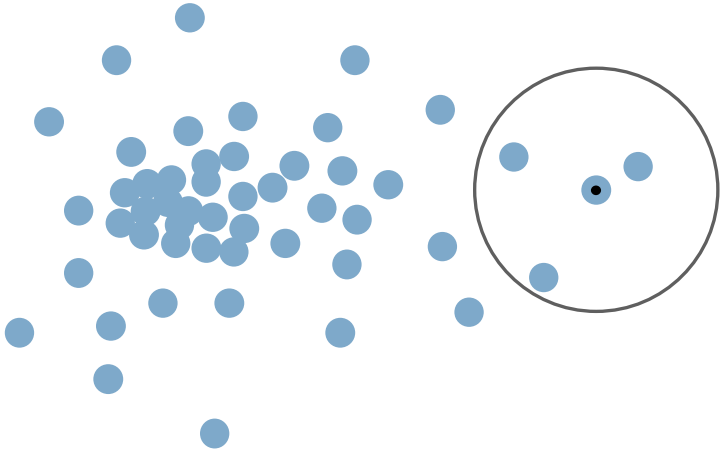
# Mean shift – Ilustração

1. Escolha uma semente inicial



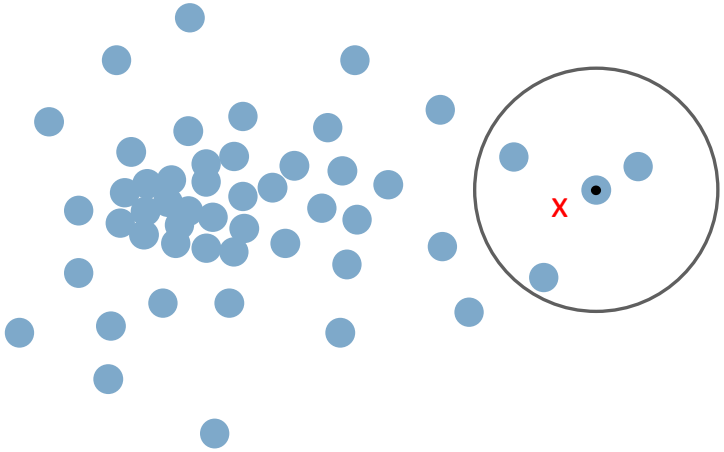
# Mean shift – Ilustração

1. Escolha uma semente inicial
2. Olhe para uma região ao redor da semente



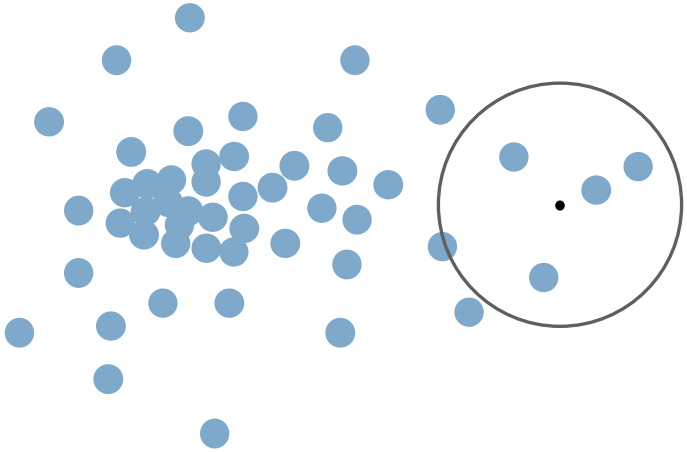
# Mean shift – Ilustração

1. Escolha uma semente inicial
2. Olhe para uma região ao redor da semente
3. Calcule a média das posições dos pontos vizinhos da semente



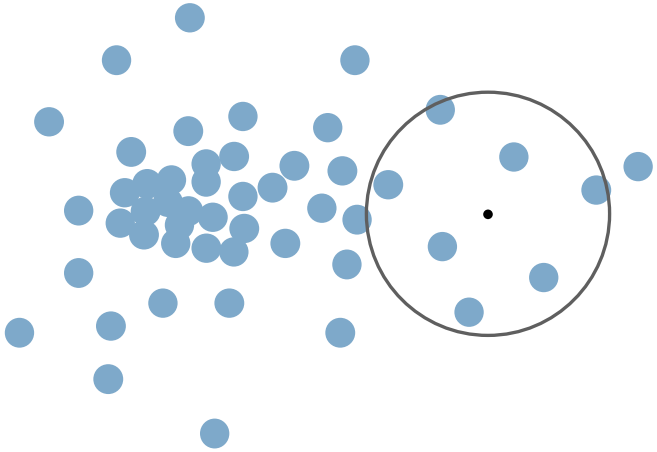
# Mean shift – Ilustração

1. Escolha uma semente inicial
2. Olhe para uma região ao redor da semente
3. Calcule a média das posições dos pontos vizinhos da semente
4. Atualize a posição da semente
5. Repita os passos 1 a 4 até que a posição da semente não mude mais



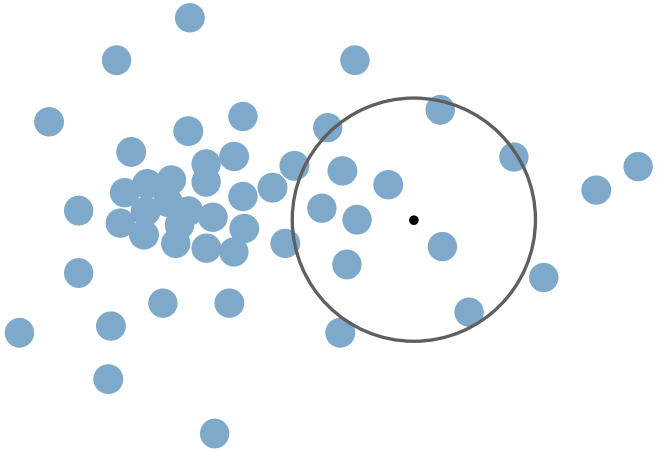
# Mean shift – Ilustração

1. Escolha uma semente inicial
2. Olhe para uma região ao redor da semente
3. Calcule a média das posições dos pontos vizinhos da semente
4. Atualize a posição da semente
5. Repita os passos 1 a 4 até que a posição da semente não mude mais



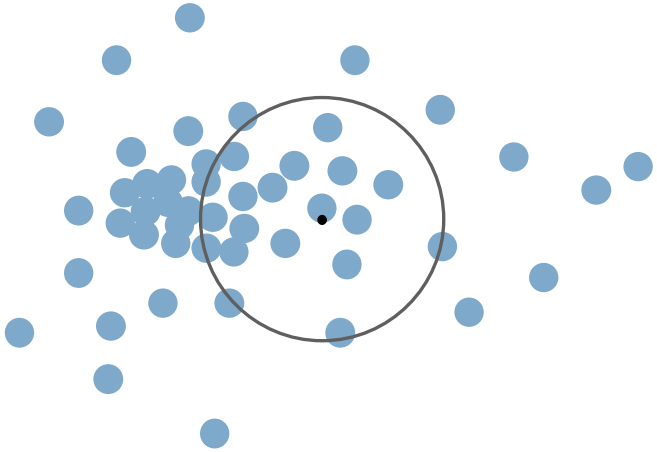
# Mean shift – Ilustração

1. Escolha uma semente inicial
2. Olhe para uma região ao redor da semente
3. Calcule a média das posições dos pontos vizinhos da semente
4. Atualize a posição da semente
5. Repita os passos 1 a 4 até que a posição da semente não mude mais



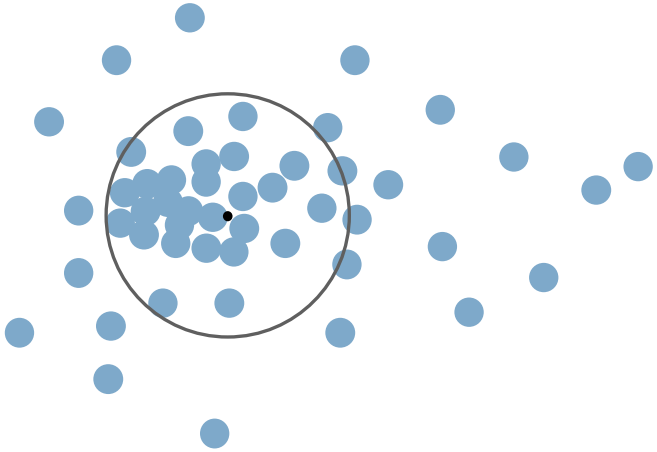
# Mean shift – Ilustração

1. Escolha uma semente inicial
2. Olhe para uma região ao redor da semente
3. Calcule a média das posições dos pontos vizinhos da semente
4. Atualize a posição da semente
5. Repita os passos 1 a 4 até que a posição da semente não mude mais



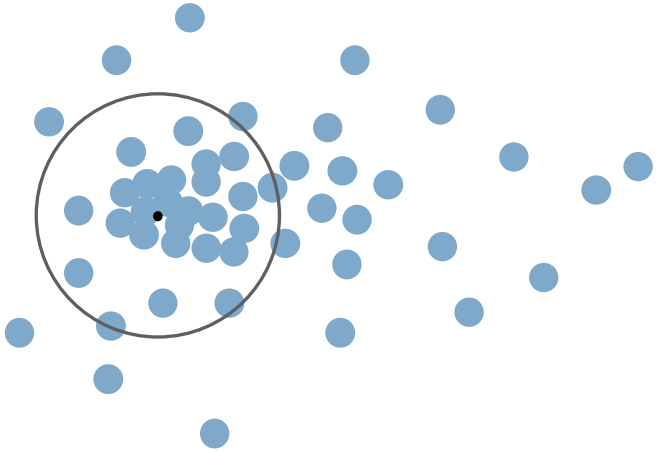
# Mean shift – Ilustração

1. Escolha uma semente inicial
2. Olhe para uma região ao redor da semente
3. Calcule a média das posições dos pontos vizinhos da semente
4. Atualize a posição da semente
5. Repita os passos 1 a 4 até que a posição da semente não mude mais



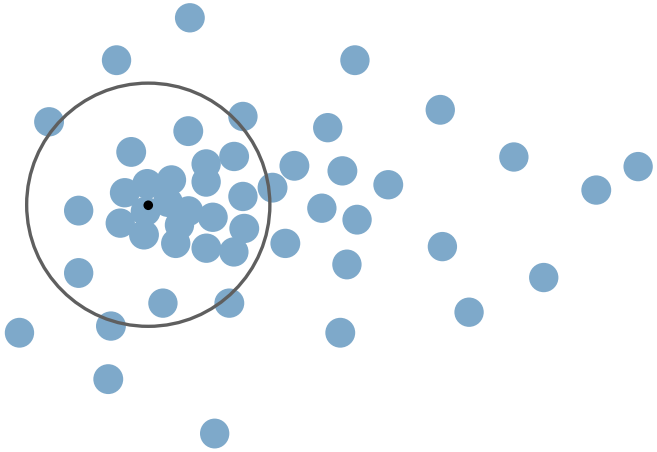
# Mean shift – Ilustração

1. Escolha uma semente inicial
2. Olhe para uma região ao redor da semente
3. Calcule a média das posições dos pontos vizinhos da semente
4. Atualize a posição da semente
5. Repita os passos 1 a 4 até que a posição da semente não mude mais

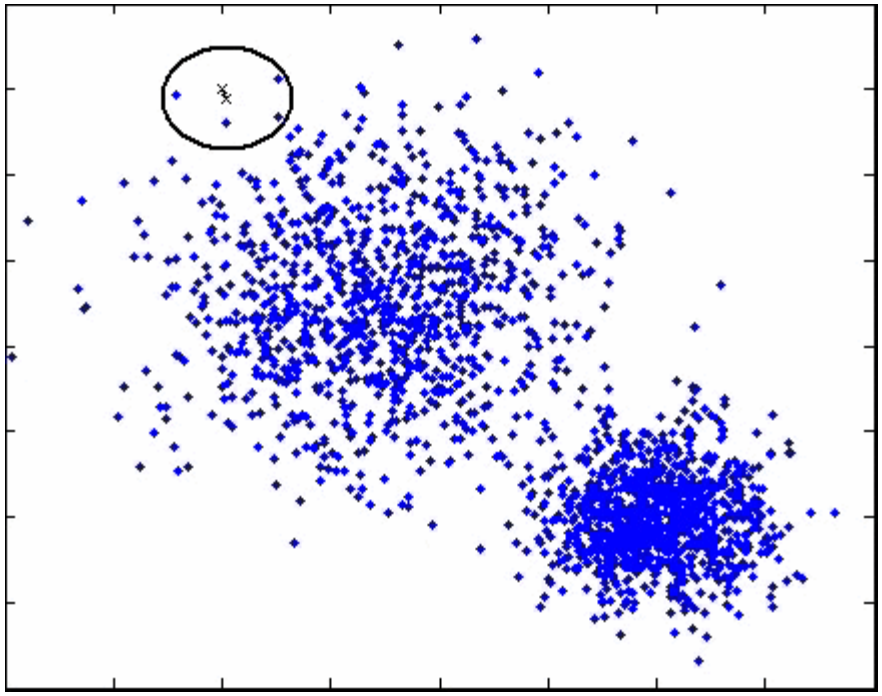


# Mean shift – Ilustração

1. Escolha uma semente inicial
2. Olhe para uma região ao redor da semente
3. Calcule a média das posições dos pontos vizinhos da semente
4. Atualize a posição da semente
5. Repita os passos 1 a 4 até que a posição da semente não mude mais



# Mean shift – Ilustração

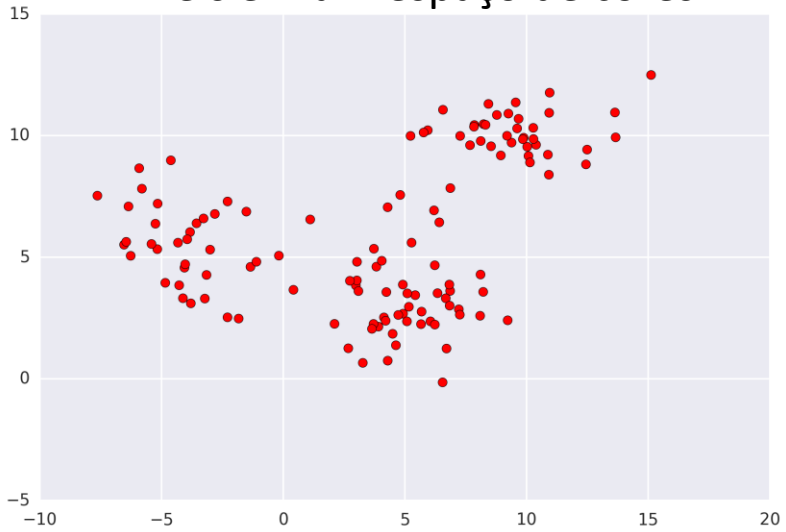


# Mean shift – Segmentação de imagens

- Como observado, a técnica mean shift pode ser utilizada para segmentar imagens

# Mean shift – Segmentação de imagens

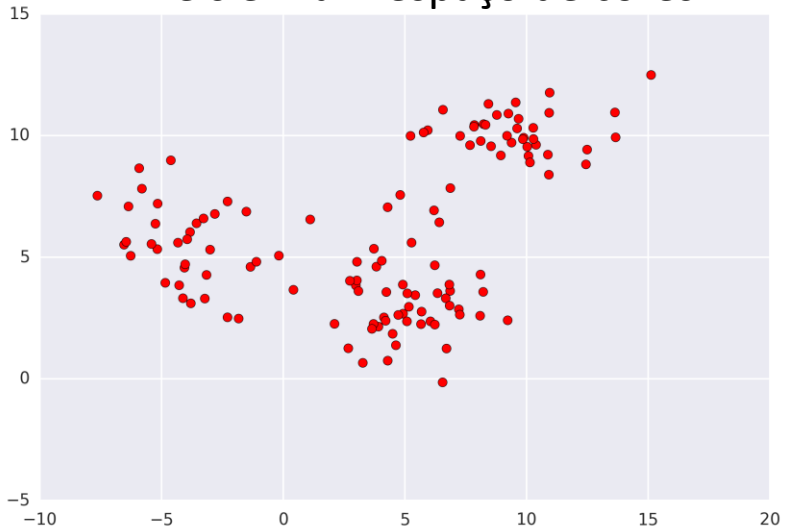
Pixels em um espaço de cores



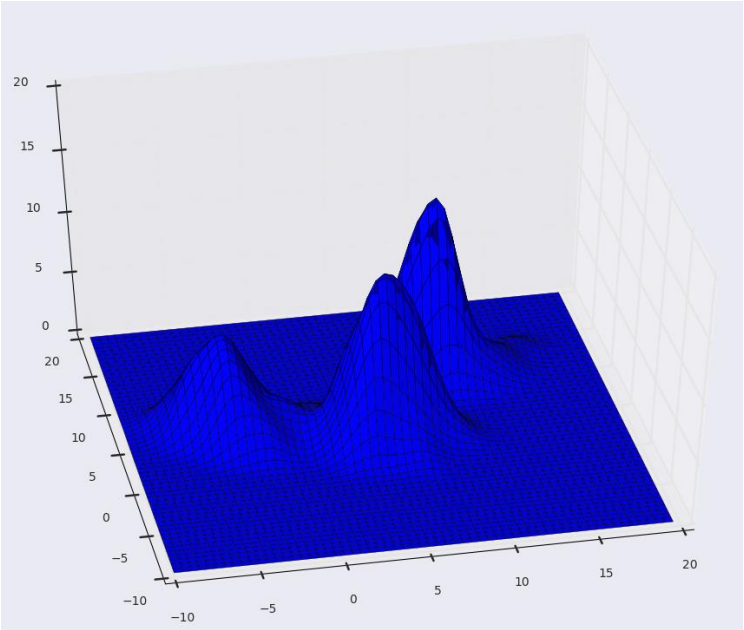

https://spin.atomicobject.com/2015/05/26/mean-shift-clustering/

# Mean shift – Segmentação de imagens

Pixels em um espaço de cores



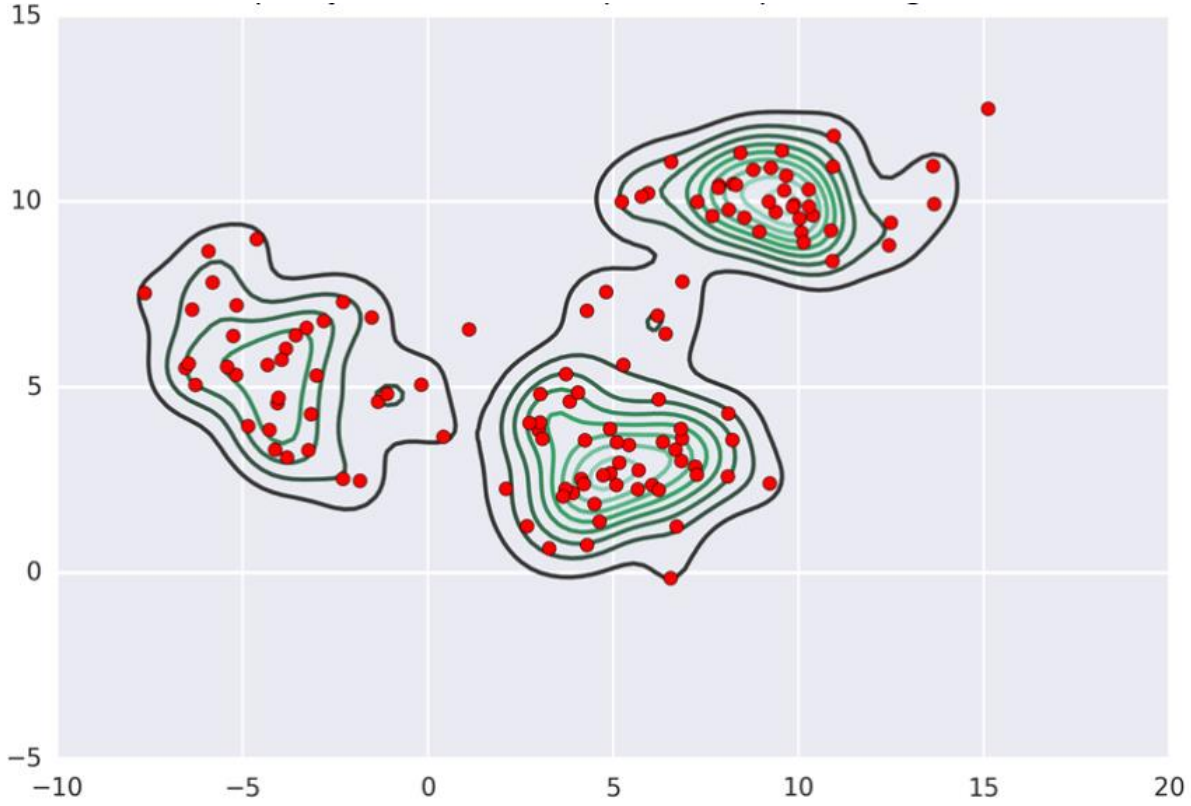

Estimação da densidade de probabilidade dos pontos




https://spin.atomicobject.com/2015/05/26/mean-shift-clustering/

# Mean shift – Segmentação de imagens

Pixels em um espaço de cores



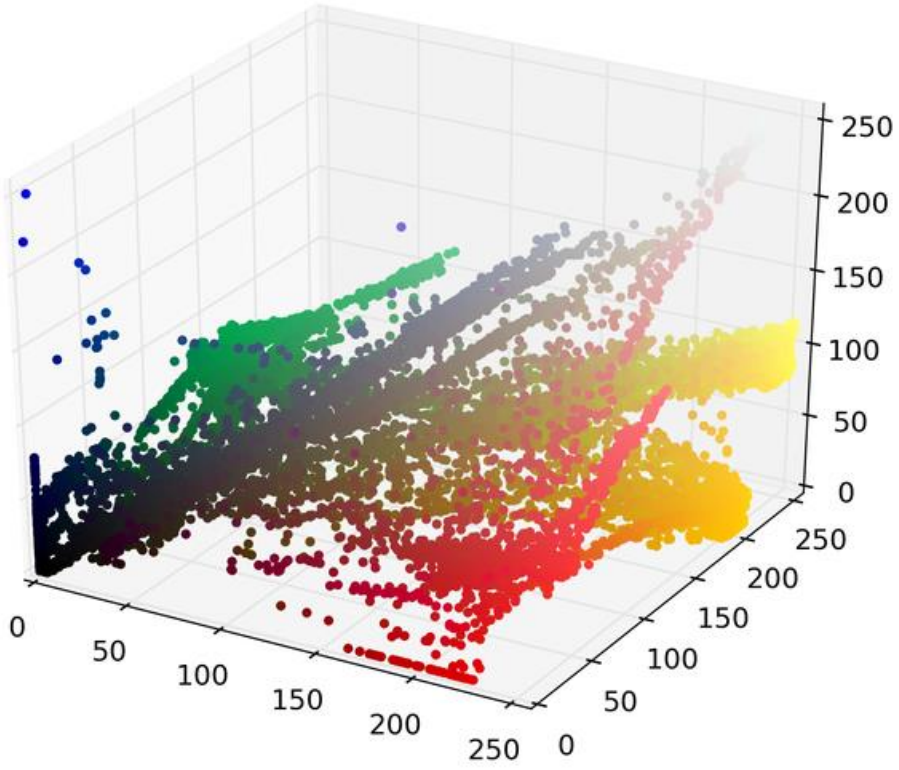
# Mean shift – Segmentação de imagens

# Mean shift – Segmentação de imagens


https://spin.atomicobject.com/2015/05/26/mean-shift-clustering/

# Mean shift – Segmentação de imagens

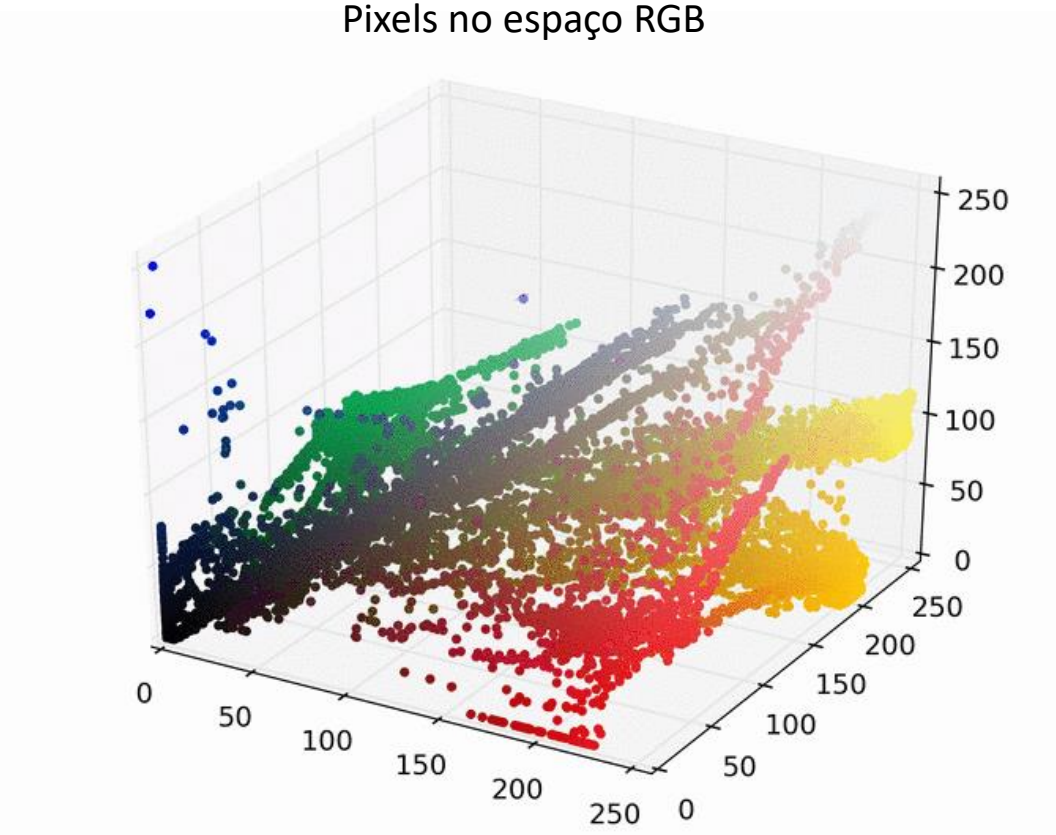
Pixels no espaço RGB




https://spin.atomicobject.com/2015/05/26/mean-shift-clustering/

# Mean shift – Segmentação de imagens

Pixels no espaço RGB




https://spin.atomicobject.com/2015/05/26/mean-shift-clustering/

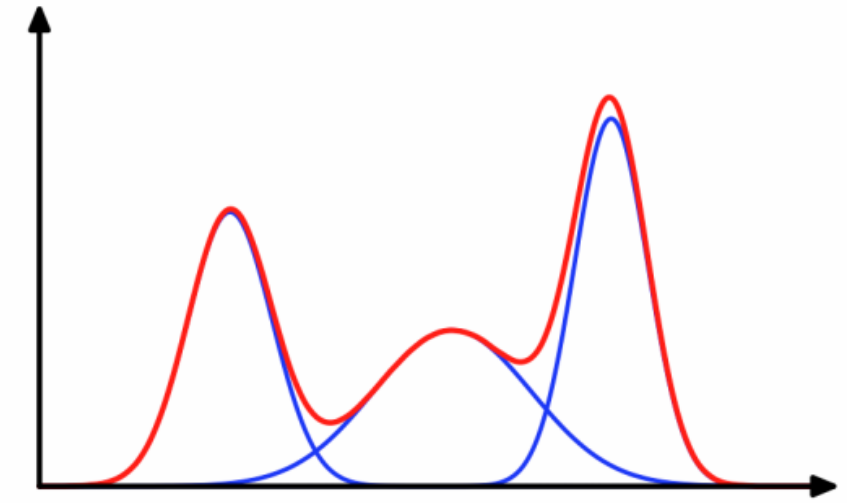
# Mean shift – Segmentação de imagens


https://spin.atomicobject.com/2015/05/26/mean-shift-clustering/

# Mistura de gaussianas

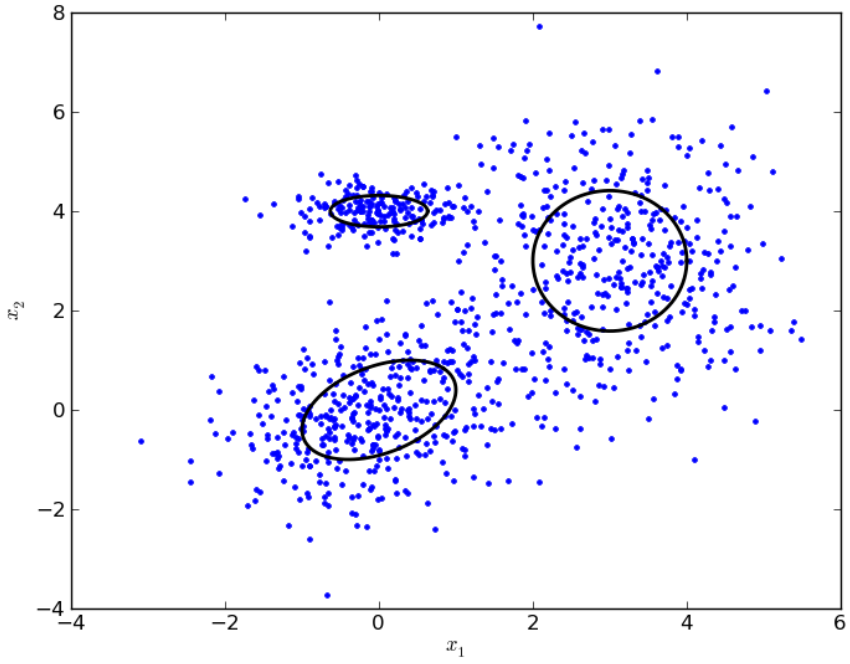
# Mistura de gaussianas

- Misturas de gaussianas podem ser utilizadas para representar funções possuindo diversos picos
- Por exemplo, a curva vermelha abaixo pode ser representada como a soma de três gaussianas, cada uma tendo diferentes amplitudes e larguras



# Mistura de gaussianas

Exemplo para um caso 2D, elipses representam um “corte” na gaussiana utilizada para representar o grupo



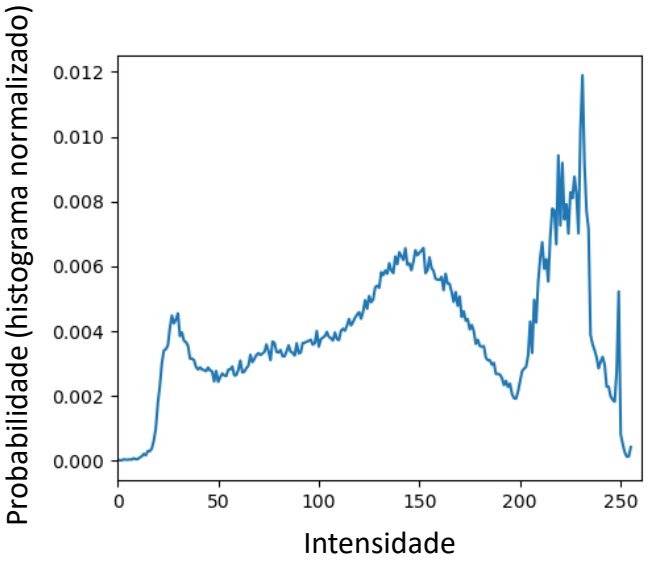
# Mistura de gaussianas

Assim como feito para o k-médias, podemos definir uma metodologia para ajustar K gaussianas a um conjunto de dados.

A principal diferença entre k-médias e mistura de gaussianas é que

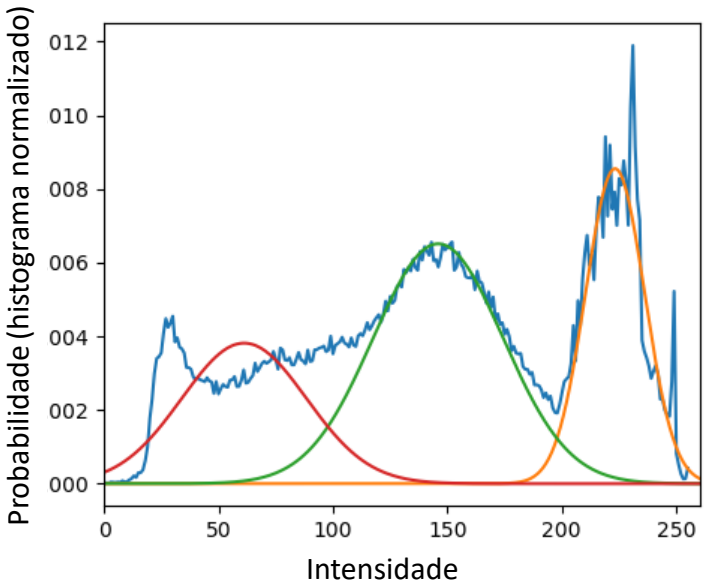
1. K-médias resulta em *hard clustering*, isto é, cada ponto é associado a um único cluster
2. Mistura de gaussianas resulta em *soft clustering*, isto é, temos a probabilidade de cada ponto pertencer a cada classe

# Exemplo de segmentação por mistura de gaussianas



# Exemplo de segmentação por mistura de gaussianas

Resultado do ajuste de três gaussianas



# Exemplo de segmentação por mistura de gaussianas

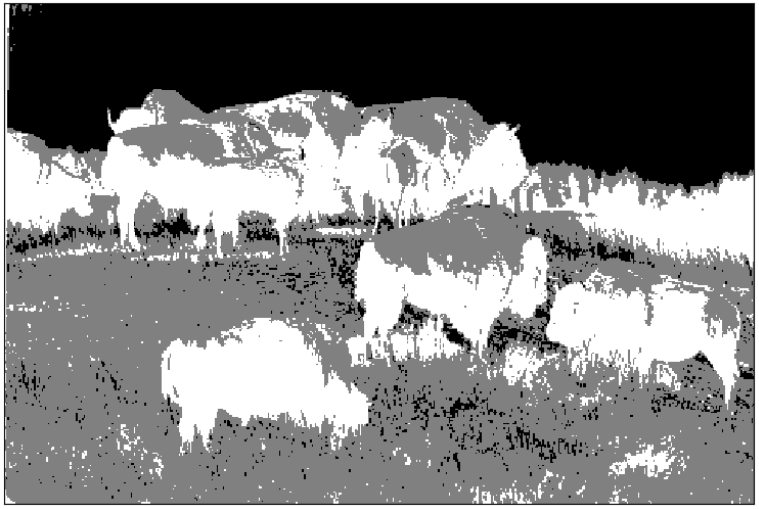
Probabilidades de cada pixel pertencer a cada uma das três classes

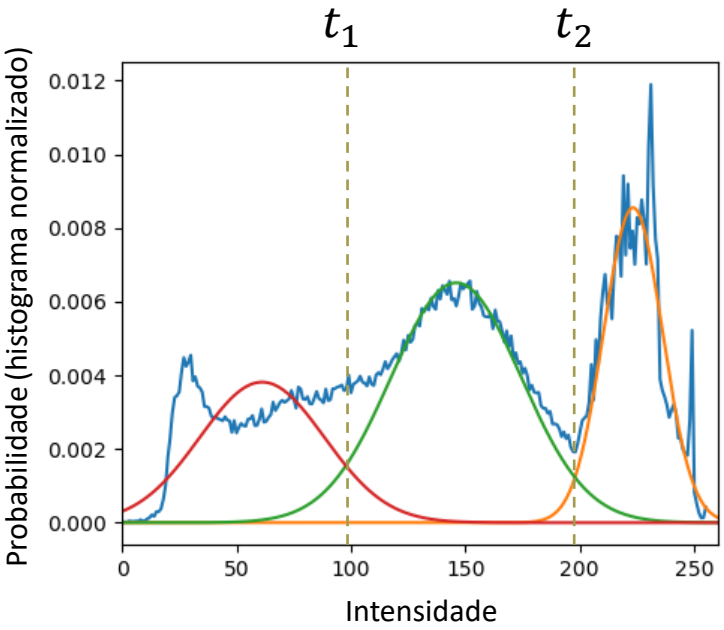
$P(I|\mu_1, \sigma_1)$

$P(I|\mu_2, \sigma_2)$

$P(I|\mu_3, \sigma_3)$

# Exemplo de segmentação por mistura de gaussianas





# Mistura de gaussianas

Notebook “**Mistura de Gaussianas**”

# Crescimento de região

# Crescimento de região

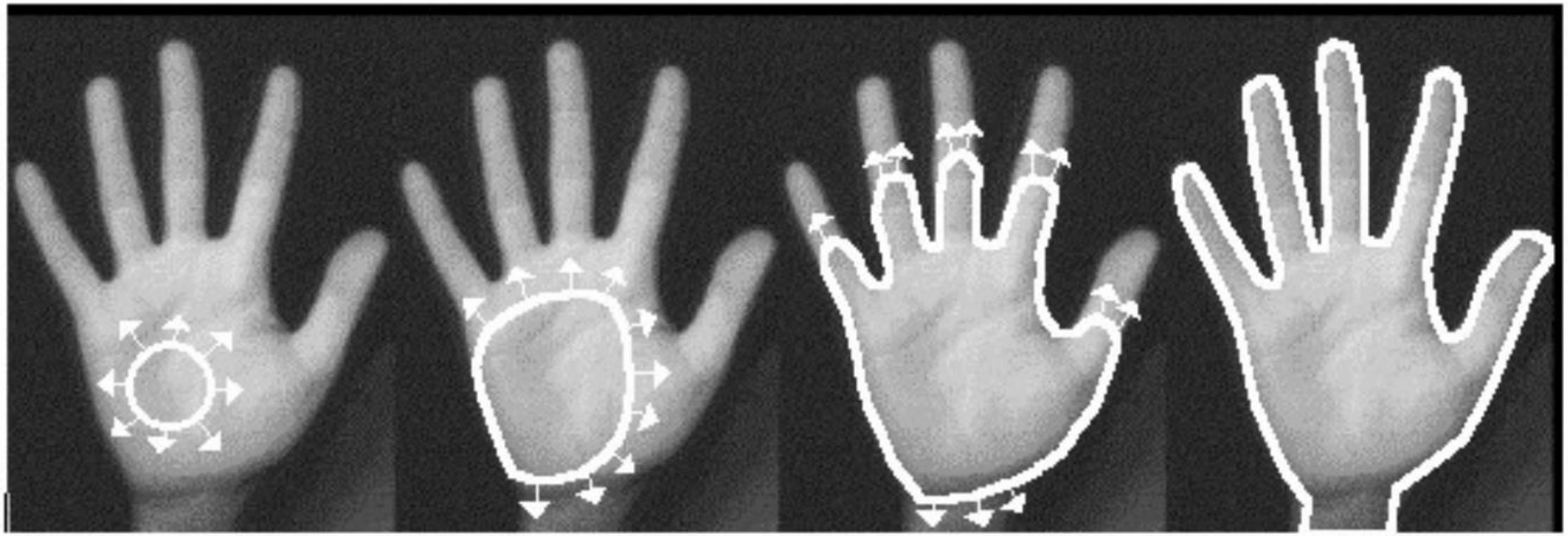
- A entrada de um algoritmo de crescimento de região é um pixel ou um conjunto de pixels semente, formando a região de segmentação inicial
- Cada pixel ao redor da região segmentada é verificado, e um desses pixels é adicionado à região se ele obedecer um critério pré-determinado

- Um possível critério de adição de pixel:
  - Dada uma região $R$, calcule a média e o desvio padrão das intensidades no interior da região
  - Para cada pixel ao redor da borda de $R$, tendo intensidade $I$, adicione-o à região se $\mu_R - \sigma_R < I < \mu_R + \sigma_R$

# Modelos de contorno ativo

# Modelos de contorno ativo

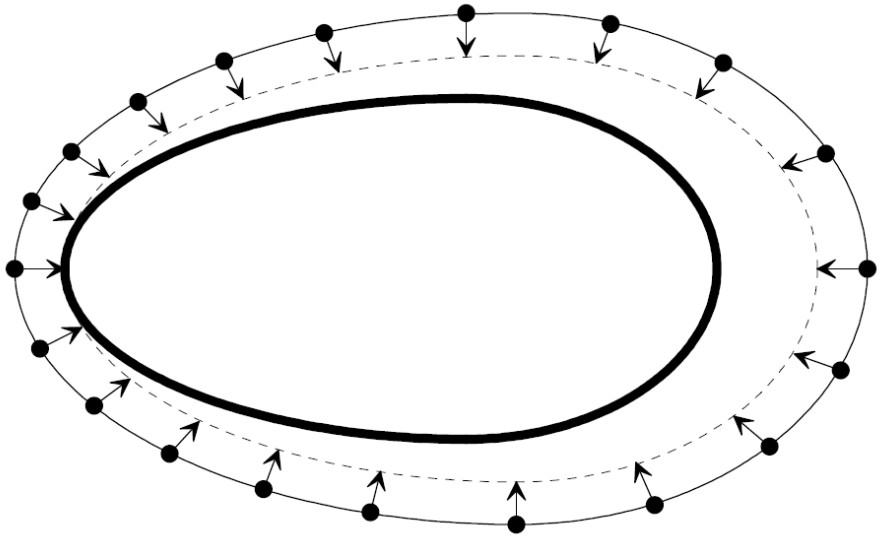
Técnica parecida com crescimento de região, mas são impostas restrições adicionais no formato do contorno. Por exemplo, podemos crescer um contorno com a restrição de que sua curvatura máxima não ultrapasse um determinado valor.



# Modelos de contorno ativo (Snakes)

- Um conjunto de pontos de controle evolui ao longo do tempo
- A cada iteração, os pontos de controle são modificados de forma que a seguinte energia seja minimizada:

$$E_{snake} = E_{interna} + E_{imagem}$$



# Modelos de contorno ativo (Snakes)

- Um conjunto de pontos de controle evolui ao longo do tempo
- A cada iteração, os pontos de controle são modificados de forma que a seguinte energia seja minimizada:

$$E_{snake} = E_{interna} + E_{imagem}$$

$E_{interna}$ está associada com a energia interna do contorno. Usualmente, ela é definida através das derivadas do contorno (curvatura), de forma que menores valores de $E_{interna}$ levem a contornos mais suaves

$E_{imagem}$ é associada com a imagem sendo segmentada. Por exemplo, ela pode ser inversamente proporcional a magnitude do gradiente, de tal forma que o contorno seja atraído para regiões de borda.

# Modelos de contorno ativo (Snakes)

- Evolução de contorno é uma poderosa metodologia na qual condições adicionais podem ser naturalmente impostas ao contorno. Por exemplo, pontos fixos do contorno ou uma forma ótima (elipse, quadrado, etc)
- Em muitos casos, o resultado é altamente dependente do contorno inicial

# Processamento de Vídeo Utilizando o OpenCV

# Processamento de vídeo utilizando o OpenCV

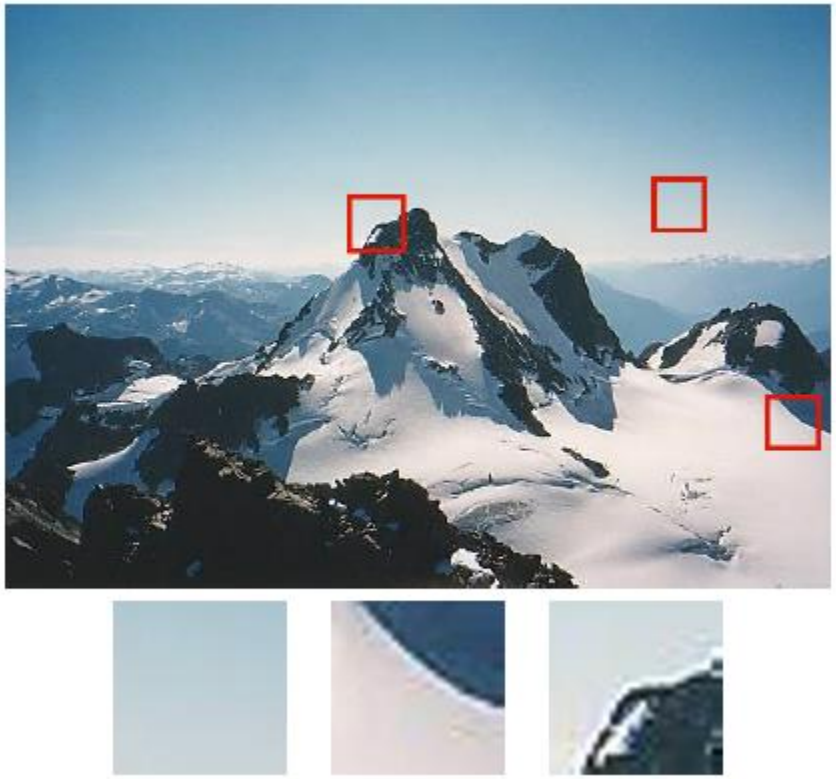
- Um vídeo nada mais é do que uma sequência de imagens. Portanto, todos os métodos que vimos sobre processamento de imagens podem ser utilizados em vídeos
- Por outro lado, existem diversos métodos que foram definidos para serem aplicados especificamente em vídeos. Tais métodos usualmente utilizam informações sobre a evolução temporal da imagem para obterem resultados melhores do que se apenas uma imagem estática fosse considerada

# Processamento de vídeo utilizando o OpenCV

Notebook “**Processamento de vídeo**”

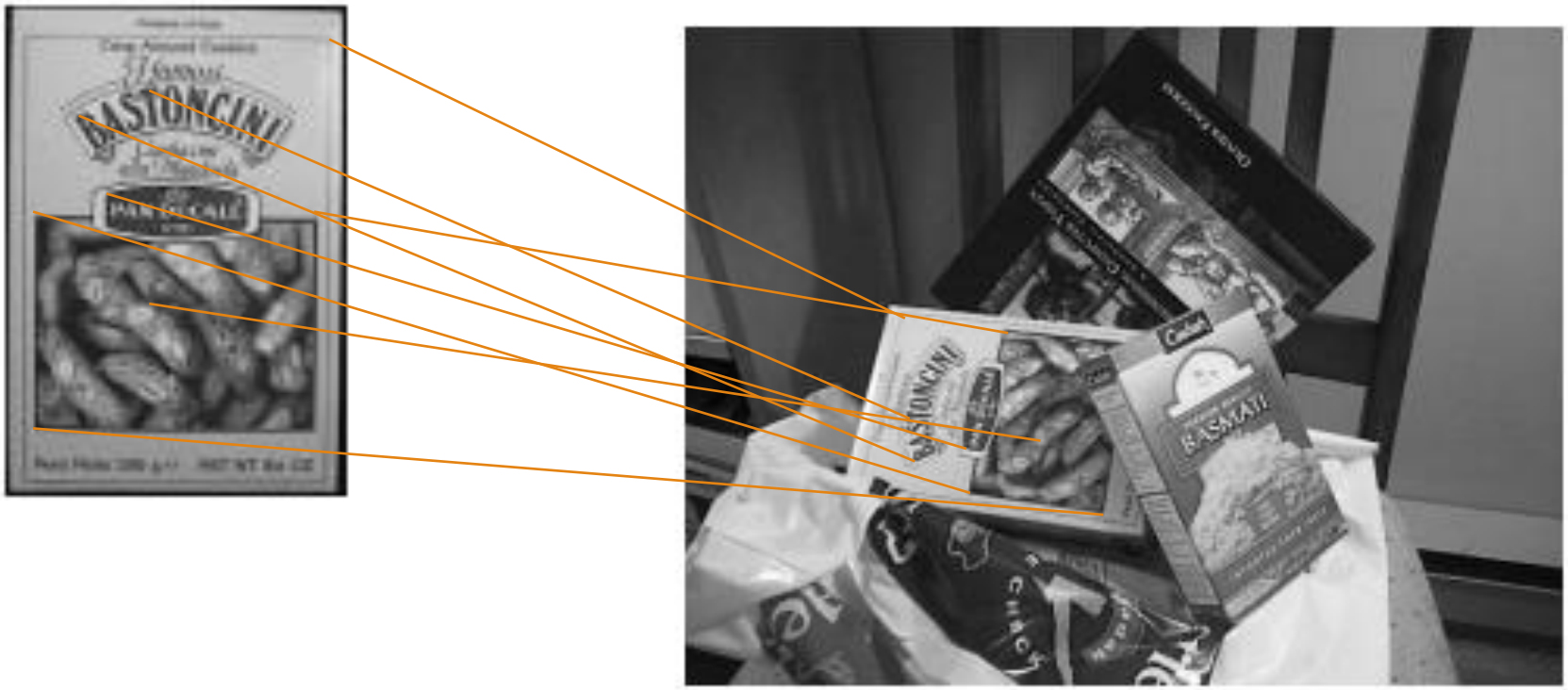
# Processamento de imagens. E o que mais?

# Detecção de pontos salientes



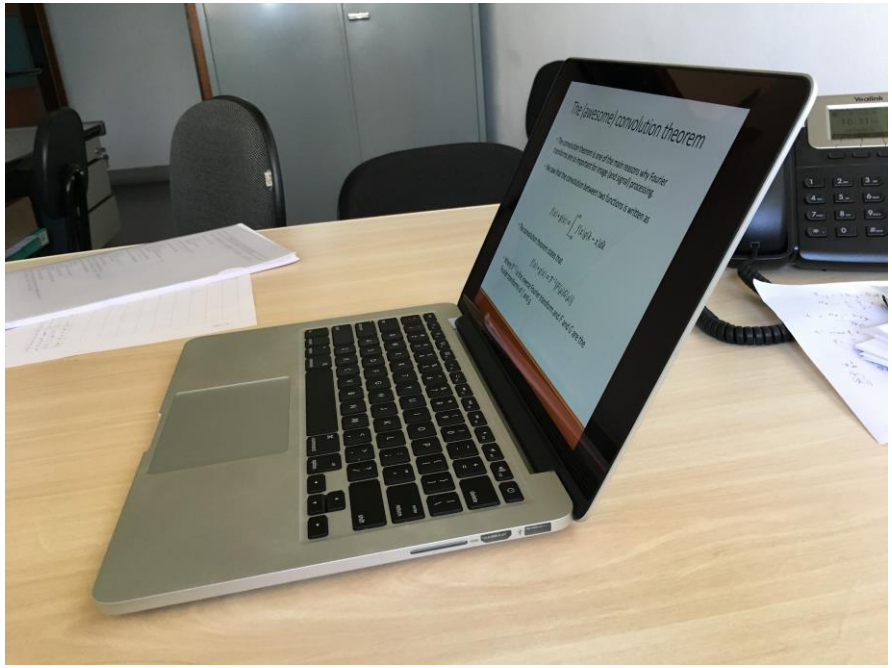
# Detecção de pontos salientes


# Casamento de pontos salientes

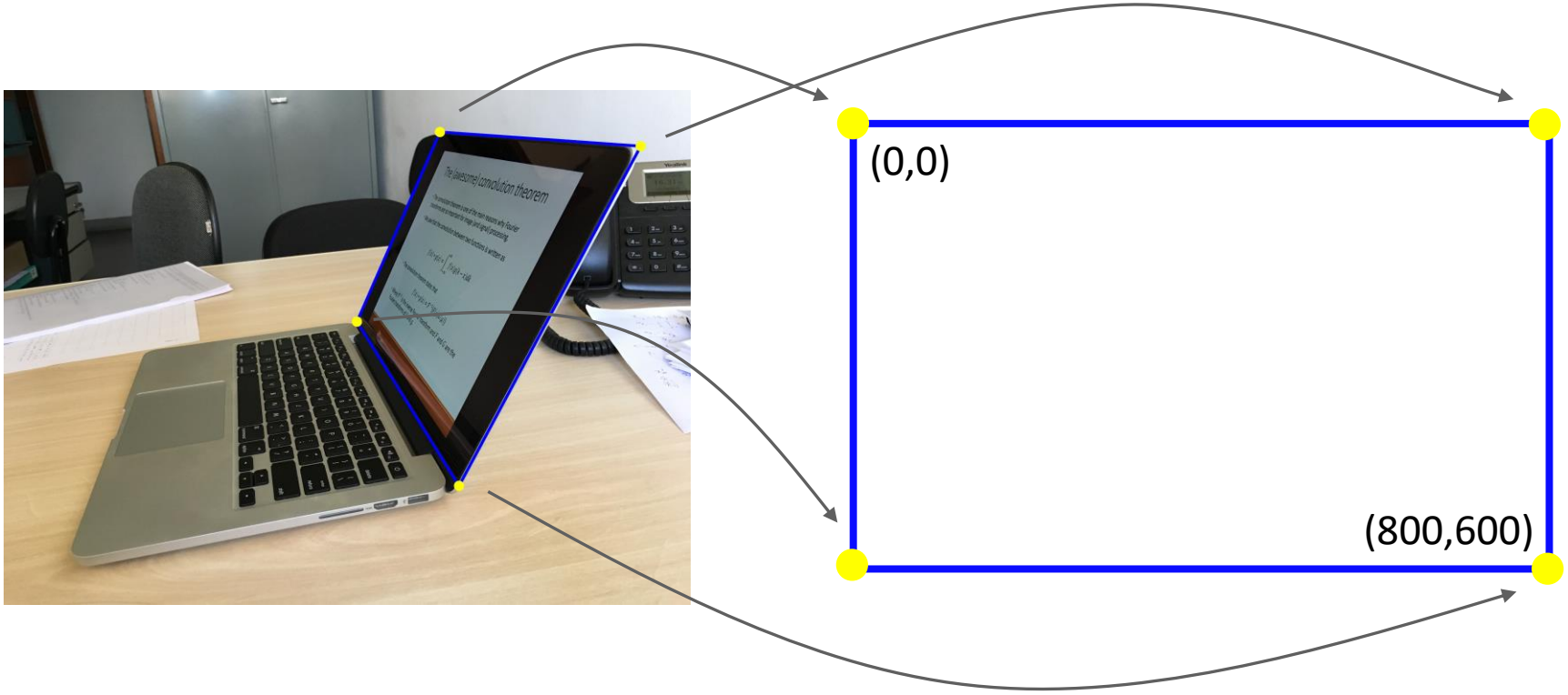


# Estimativa de transformações geométricas

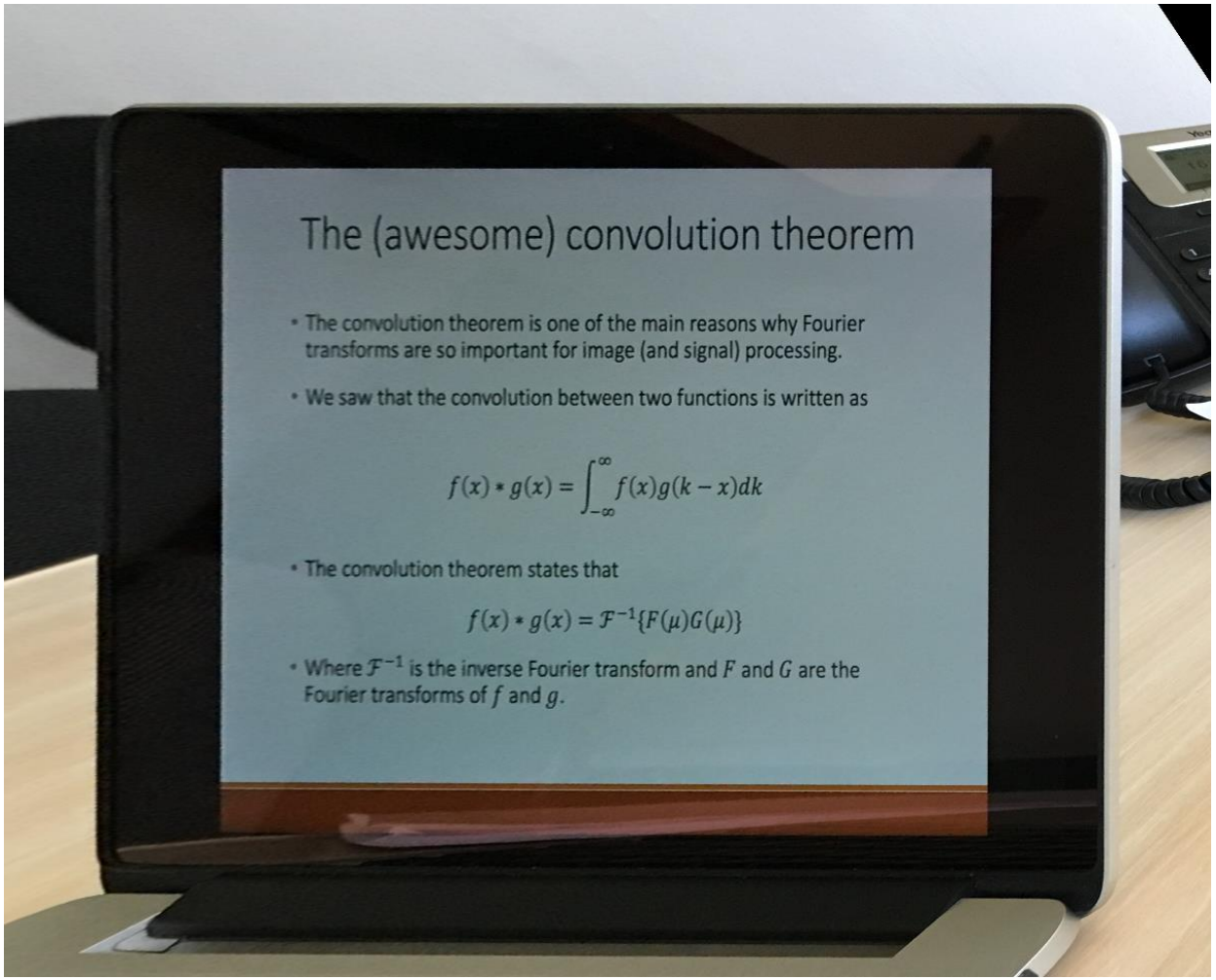

# Transformações geométricas



# Transformações geométricas



# Transformações geométricas



# Registro de Imagens

Registro de imagens pode ser utilizado para, por exemplo, unir diferentes imagens obtidas de partes de uma cena

# Registro de Imagens

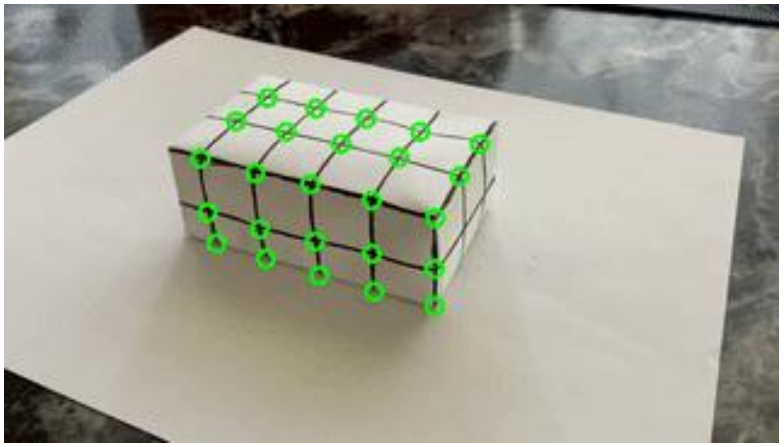
# Criação de mosaico de imagens

# Registro de Imagens – Estabilização de vídeo

# Registro de Imagens – Estabilização de vídeo

# Fluxo ótico

# Modelos de câmeras



# Modelos de câmeras

# Projeto 4

- Não utilizar funções prontas para implementar o principal conceito associado ao tema. Na dúvida, pergunte que funções/bibliotecas podem ser utilizadas no projeto.
- Entregáveis:
  - Código produzido (a organização do código também será avaliada!)
  - Um artigo escrito em Latex (∼6 páginas em coluna dupla ou ∼10 em coluna única) contendo:
    - Resumo
    - Introdução
      - Motivação do uso do método (porque usar? Em que situações ele é importante?)
    - Objetivos
      - O que será analisado sobre o método?
    - Metodologia
      - Explicação sobre a teoria do método
      - Explicação sobre a parte mais importante do código
    - Resultados
    - Conclusões
- Data de entrega: 28/03

# Projeto 4 - Temas

1. Implementação do descritor de textura Local binary pattern (LBP)
2. Segmentação por crescimento de região
3. Classificação de imagens obtidas pelos integrantes do grupo
4. Implementação do método non-local means
5. Utilizar PCA para analisar propriedades de imagens de folhas
6. Construir um sistema para desenhar no ar com um objeto colorido

# Projeto 4 – Tema 1

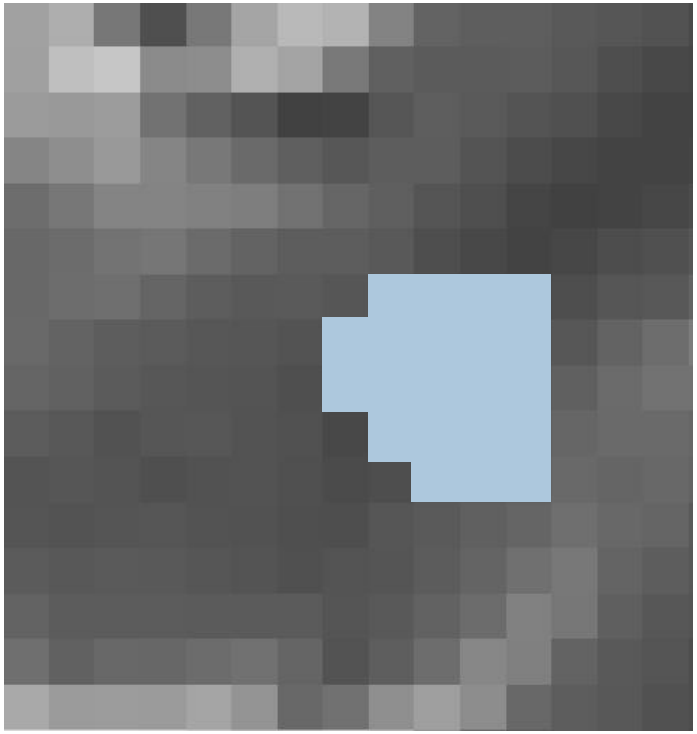
Implementação do descritor de textura Local binary pattern (LBP)

- Para uma dada imagem, calcular o vetor de atributos LBP, visto na aula passada.
- O vetor é dado pelos 256 valores do histograma associados aos valores LBP calculados para cada pixel de uma imagem
- Utilizar o LBP para classificar as imagens de dígitos

# Projeto 4 – Tema 2

Segmentação por crescimento de região

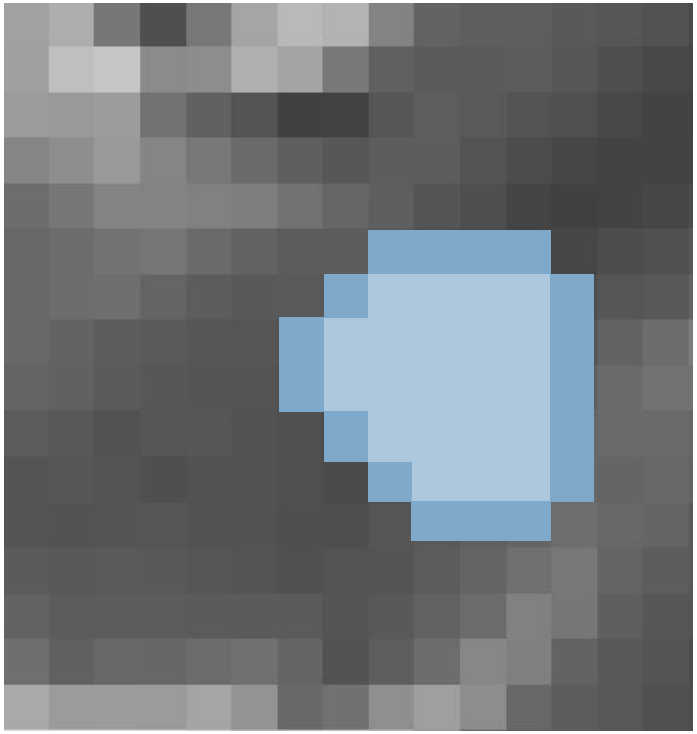
- A condição inicial é um pixel, ou uma região



# Projeto 4 – Tema 2

Segmentação por crescimento de região

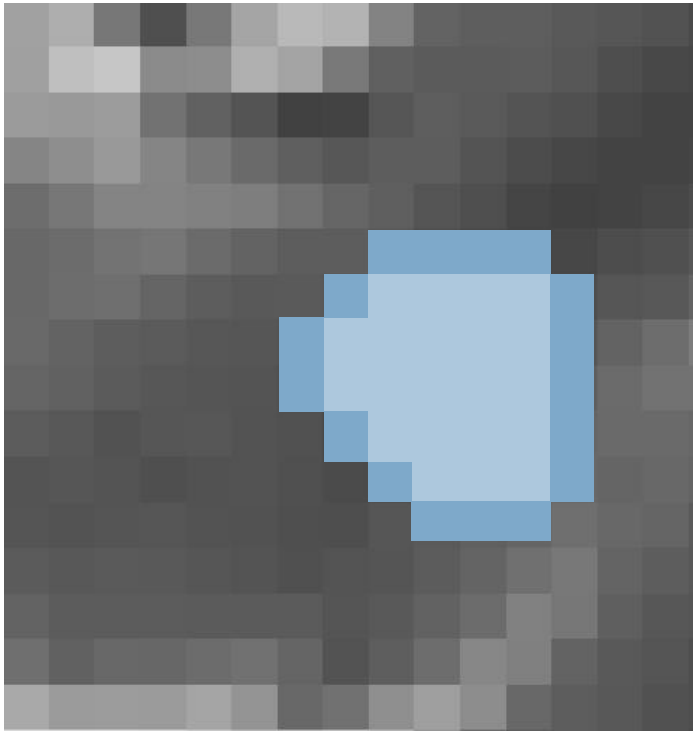
- A condição inicial é um pixel, ou uma região
- A borda externa da região é composta por pixels que estão fora da região e que possuem ao menos um vizinho-4 dentro da região



# Projeto 4 – Tema 2

Segmentação por crescimento de região

- A condição inicial é um pixel, ou uma região
- A borda externa da região é composta por pixels que estão fora da região e que possuem ao menos um vizinho-4 dentro da região
- Os pixels dentro da região possuem média de intensidade $\mu$ e desvio padrão de intensidade $\sigma$



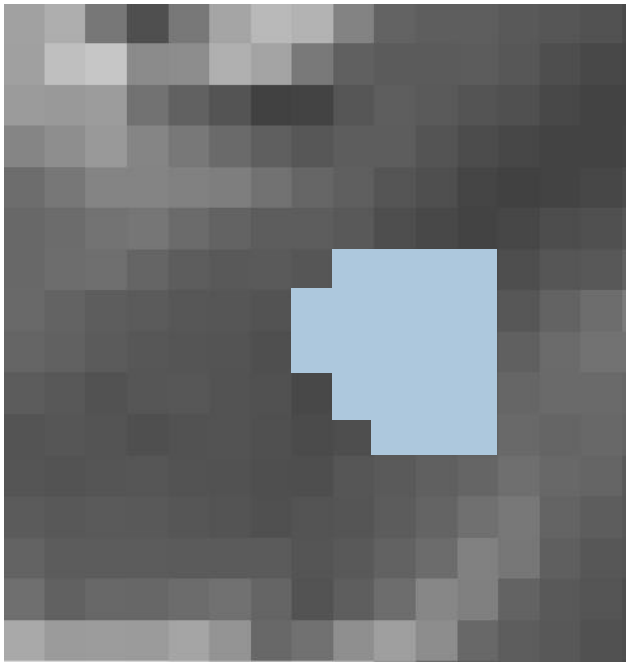
# Projeto 4 – Tema 2

Segmentação por crescimento de região

- Algoritmo:

1. Para cada pixel $p$ na borda externa da região
   a. Adicione $p$ à região se $\mu - f * \sigma < p < \mu + f * \sigma$
   b. Caso $p$ tenha sido adicionado, atualize a borda externa e também os valores de $\mu$ e $\sigma$
2. Repita 1. até que não haja nenhum pixel de borda externa a ser adicionado

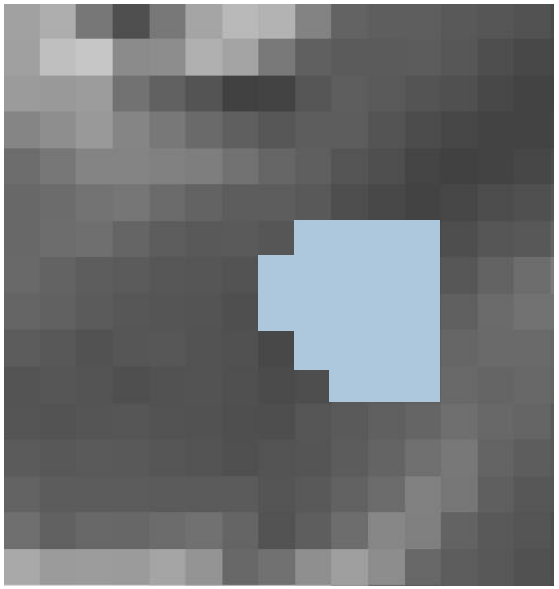
- $f$ é um parâmetro do método



# Projeto 4 – Tema 2

Segmentação por crescimento de região

- Dica: utilize uma lista para armazenar os pixels na borda interna da região e outra lista para armazenar os pixels da região
- Os vizinhos dos pixels de borda interna são verificados e adicionados ou não à lista de pixels borda
- Se um pixel não é mais borda, ele é removido da lista de pixels de borda



* Em Python, um conjunto de pixels pode ser representado utilizando a função set([(2, 3), (6, 3), (1, 8)]), que cria uma coleção de elementos.

# Projeto 4 – Tema 2

Segmentação por crescimento de região

- Dica2: ao adicionar um pixel à região, você não precisa percorrer novamente toda a região para calcular os novos valores de $\mu$ e $\sigma$
- Seja $\mu_i$ e $\sigma_i$ a média e desvio padrão das intensidades na região, e $I_p$ o valor do pixel adicionado à região, os novos valores de média e desvio são dados por

$$\mu_{i+1} = \frac{N\mu_i + I_p}{N+1}$$

$$\sigma_{i+1} = \sqrt{\frac{\left(\sigma_i^2 + \mu_i^2\right)N + I_p^2}{N+1} - \mu_{i+1}^2}$$

$N$ indica o número de pixels na região antes da adição do novo pixel.

# Projeto 4 – Tema 3

Classificação de imagens obtidas pelos integrantes do grupo

- Construir uma base de imagens contendo 5 classes e ao menos 40 imagens para cada classe. As imagens devem ser obtidas pelo grupo, isto é, não utilizar imagens da internet.
- Cada imagem deve possuir variação significativa em relação às demais.
- Para cada imagem, gerar 3 novas imagens aplicando as seguintes transformações: aumento de brilho (transformação gamma), redução do brilho (transformação gamma ) e borramento da imagem (filtro Gaussiano).
- A base final terá 800 imagens.
- Utilizar parte das imagens (ex: 80%) para treinar um classificador e tentar classificar as imagens restantes.
- No artigo, apresentar os critérios utilizados para adquirir as 200 imagens iniciais (ex: objeto com oclusão, sombra, longe da câmera, etc).

# Projeto 4 – Tema 4

Implementação do método non-local means básico

- O método non-local means é um filtro de suavização que utiliza pixels parecidos com o pixel referência para calcular o novo valor a ser associado ao pixel referência
- Os pixels parecidos não estão necessariamente próximos ao pixel referência

# Projeto 4 – Tema 4

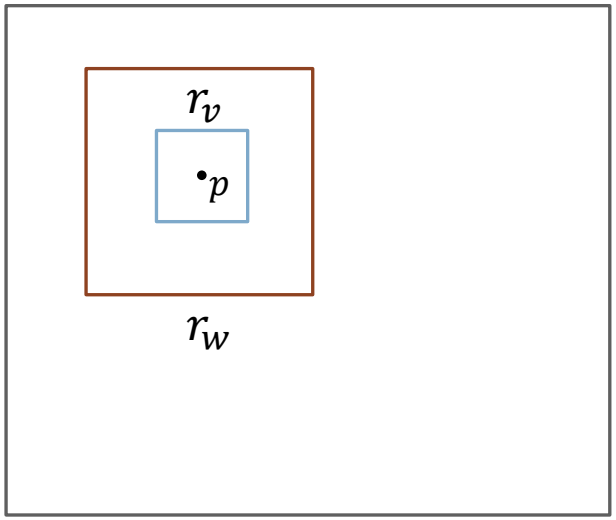
Implementação do método non-local means básico

- Parâmetros do método:
  - $r_v$: Tamanho da vizinhança a ser comparada
  - $r_w$: Tamanho da janela de busca para pixels parecidos
  - $h$: Distância típica de cut-off

# Projeto 4 – Tema 4

Implementação do método non-local means básico

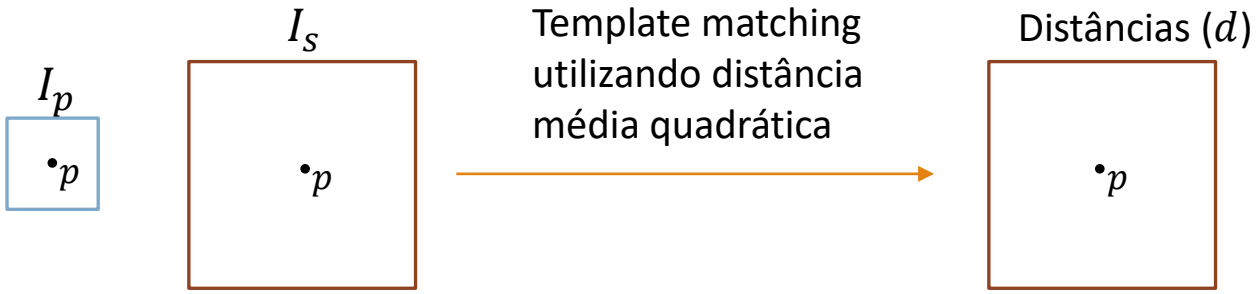
- Dado um pixel $p$ da imagem
- Obtemos uma nova imagem $I_p$ de tamanho $r_p \times r_p$ contendo a vizinhança do pixel $p$
- Obtemos uma nova imagem $I_s$ de tamanho $r_s \times r_s$ contendo a região em torno de $p$ a ser analisada



# Projeto 4 – Tema 4

Implementação do método non-local means básico

- Calculamos a distância quadrática média entre $I_p$ e $I_s$ para cada pixel de $I_s$ (essa operação é exatamente o cálculo da matriz distância do template matching)



# Projeto 4 – Tema 4

Implementação do método non-local means básico

- A matriz de distâncias é transformada utilizando as equações

$$w = e^{-d/h^2}$$
$$w = w/\text{numpy.sum}(w)$$

- O novo valor a ser associado ao pixel $p$ é então dado por

$$I(p) = \sum_{r=0}^{r_s} \sum_{c=0}^{r_s} w(r,c) * I_s(r,c)$$

$I_s$ é a subimagem de busca definida anteriormente

# Projeto 4 – Tema 4

Implementação do método non-local means básico

- O processo é repetido para todos os pixels da imagem

- Esse método corresponde a calcular uma média dos valores dos pixels em uma região em volta do pixel referência
- A média é ponderada pela similaridade entre a vizinhança do pixel referência e de cada pixel a ser utilizado na média

- Em geral, é preciso testar diversos valores de $h$ para obter resultados razoáveis

# Projeto 4 – Tema 4

Imagem $I_p$

Imagem $I_s$

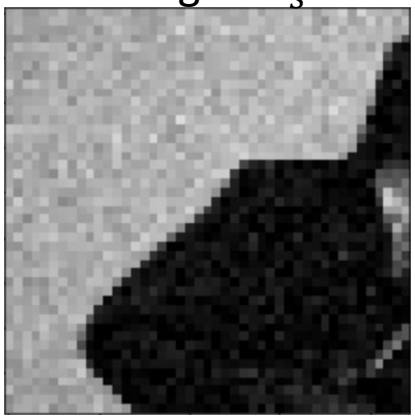


Imagem de distâncias

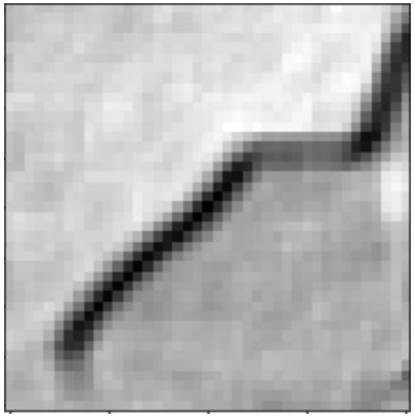
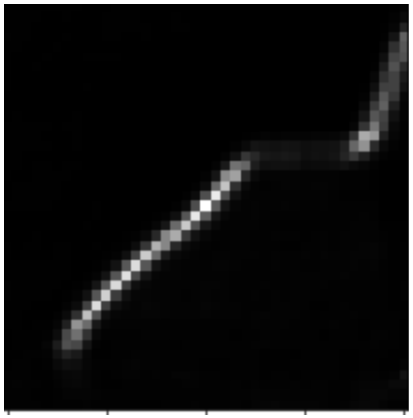
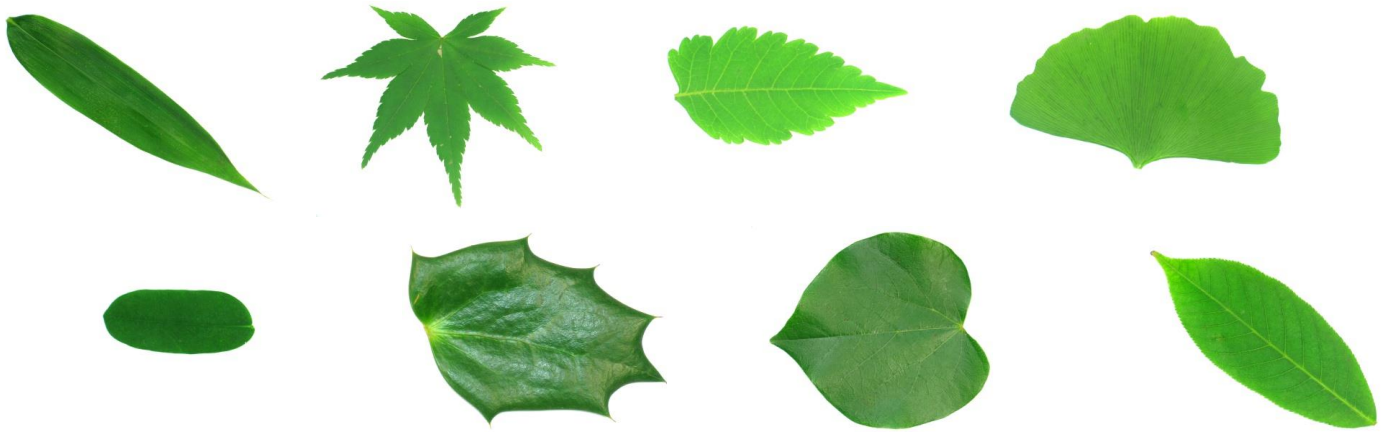


Imagem $w$



# Projeto 4 – Tema 5

- Utilizar PCA para encontrar bons atributos para uma base de imagens de folhas (disponível no AVA)
- Implementar 10 medidas morfológicas e aplicá-las nas imagens
- Utilizar PCA para projetar os dados em 2 dimensões, colorindo cada ponto com a classe respectiva
- A ideia é conseguir obter uma projeção na qual as classes estão razoavelmente separadas



# Projeto 4 – Tema 6

Construir um sistema para desenhar no ar com um objeto

- A ideia é possibilitar que uma pessoa com um objeto de cor específica (por exemplo, verde claro) faça desenhos na tela
- A pessoa é filmada por uma câmera, e ela movimenta o objeto para fazer os desenhos

- Utilizando a captura de imagens pelo OpenCV, a ideia é detectar o objeto, calcular o centro de massa, e pintar a imagem na posição do centro de massa

* Para pintar a imagem, utilize a função do OpenCV cv2.circle()
ou a função skimage.draw.disk() do scikit-image.