

Segmentação de imagens

Agostinho Brito

2022

1 Detecção de pontos e bordas

2 Detecção de linhas

3 Aglomeração de regiões

Segmentação de imagens

- A segmentação subdivide uma imagem em suas partes constituintes ou objetos.
- É uma das tarefas mais complicadas em processamento de imagens.
- Algoritmos são baseados nas propriedades de descontinuidade e similaridade.

Descontinuidade fronteiras das regiões são destacadas.

Similaridade pixels em uma região pode ser aglomerados por critério semelhante.

Exemplos

- Detecção de pontos e linhas
- Detecção de arestas
- Aglomeração de regiões



Segmentação de imagens

O processo de segmentação divide uma região R em n subregiões, $R_1, R_2, R_3, \dots, R_n$, de modo que:

- 1 $\bigcup_{i=1}^n R_i = R$
- 2 R_i é um conjunto conectado, para $i = 1, 2, \dots, n$.
- 3 $R_i \cap R_j = \emptyset$, para todo i e j , $i \neq j$.
- 4 Uma propriedade P aplicada a uma região R_i satisfaz à relação $P(R_i) = VERDADEIRO$ para $i = 1, 2, \dots, n$.
- 5 Para esta mesma propriedade, $P(R_i \cup R_j) = FALSO$ para duas regiões adjacentes R_i e R_j .

Detecção de pontos e bordas

Detecção de pontos

- Detecção de pontos: uso de derivadas de segunda ordem (laplaciano)

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

- Pode-se assumir que um ponto ocorre quando

$$g(x, y) = \begin{cases} 1 & \text{se } |R(x, y)| \geq T \\ 0 & \text{caso contrário} \end{cases}$$

- $R(x, y)$ é o resultado da resposta da máscara no ponto central (x, y) de uma região e T é um limiar não negativo que regula a intensidade da resposta que será considerada como ponto isolado. Ex: T igual a 90% do maior valor absoluto presente na imagem.

Detector de bordas de Marr-Hildreth

- Mudanças de intensidade são independentes de escala e as bordas são de 1 pixel de espessura.
- Uso do operador laplaciano do gaussiano ($\nabla^2 G$), ou LoG.

$$\nabla^2 G = \frac{\partial^2 G(x, y)}{\partial x^2} + \frac{\partial^2 G(x, y)}{\partial y^2}$$

$$G(x, y) = \exp^{-\frac{x^2+y^2}{2\sigma^2}} \quad \nabla^2 G(x, y) = \left[\frac{x^2 + y^2 - \sigma^2}{\sigma^4} \right] \exp^{-\frac{x^2+y^2}{2\sigma^2}}$$

- Negativo do LoG para máscara 5×5

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

Detector de bordas de Marr-Hildreth

- Encontrar os cruzamentos de zero. Para uma vizinhança 3×3 , um pixel é de cruzamento se pelo menos os sinais de dois de seus vizinhos opostos (norte/sul, leste/oeste, diagonais) são diferentes.
- Um limiar para o cruzamento de zero que seja diferente de “zero” pode ser escolhido. Ex: percentual do máximo valor do LoG.
- Cerca de 99,7% do volume de uma gaussiana encontra-se em $\pm 3\sigma$ ao redor da média. Um filtro de tamanho $n \times n$, com n ímpar, deve ser escolhido tal que $n > 6\sigma$. Ex: $\sigma = 4 \rightarrow n = 25$.
- Problemas: pode gerar falsas arestas e erros graves em arestas curvas.



Detector de bordas de Canny - Objetivos

- Baixa taxa de erros
- Pontos de borda devem estar bem localizados
- Resposta de um único ponto de borda

Detector de bordas de Canny - Etapas

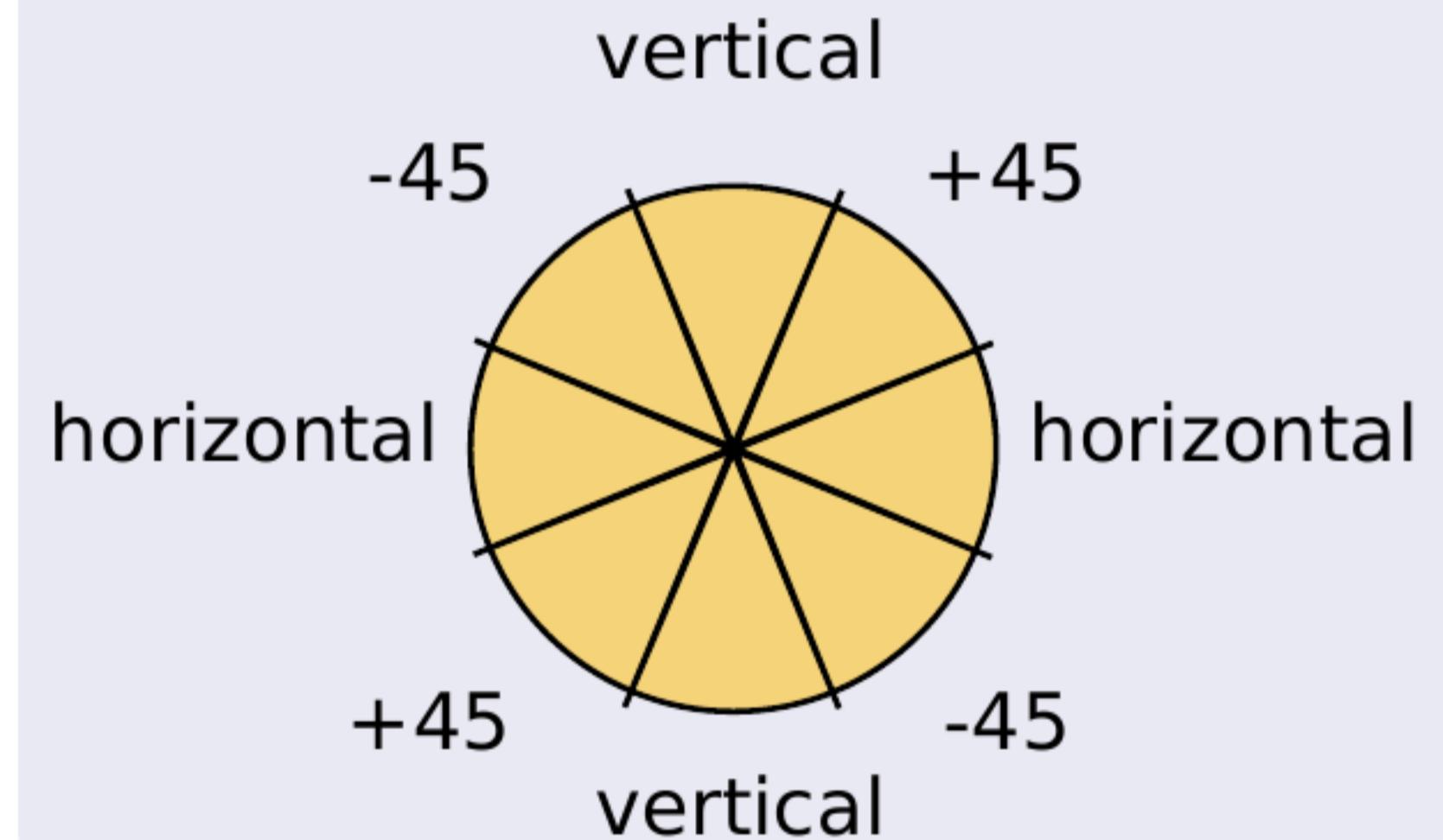
Borramento para redução de ruídos

- Convolução com o filtro Gaussiano, cálculo da magnitude e ângulo do gradiente.

Afinação das cristas largas do gradiente

- Classificação dos pontos quanto às orientações Horizontal, Vertical, $+45^\circ$, -45° (intervalos de $\pm 22.5^\circ$).
- Para os vizinhos na orientação determinada para o pixel, verificar os seus gradientes.
- Supressão de não máximos: se o valor da magnitude do gradiente $M(x, y)$ for inferior a pelo menos um de seus vizinhos, faça $g_N(x, y) = 0$; caso contrário, faça $g_N(x, y) = M(x, y)$. $g_N(x, y)$ é a imagem com supressão.

Direções dos ângulos



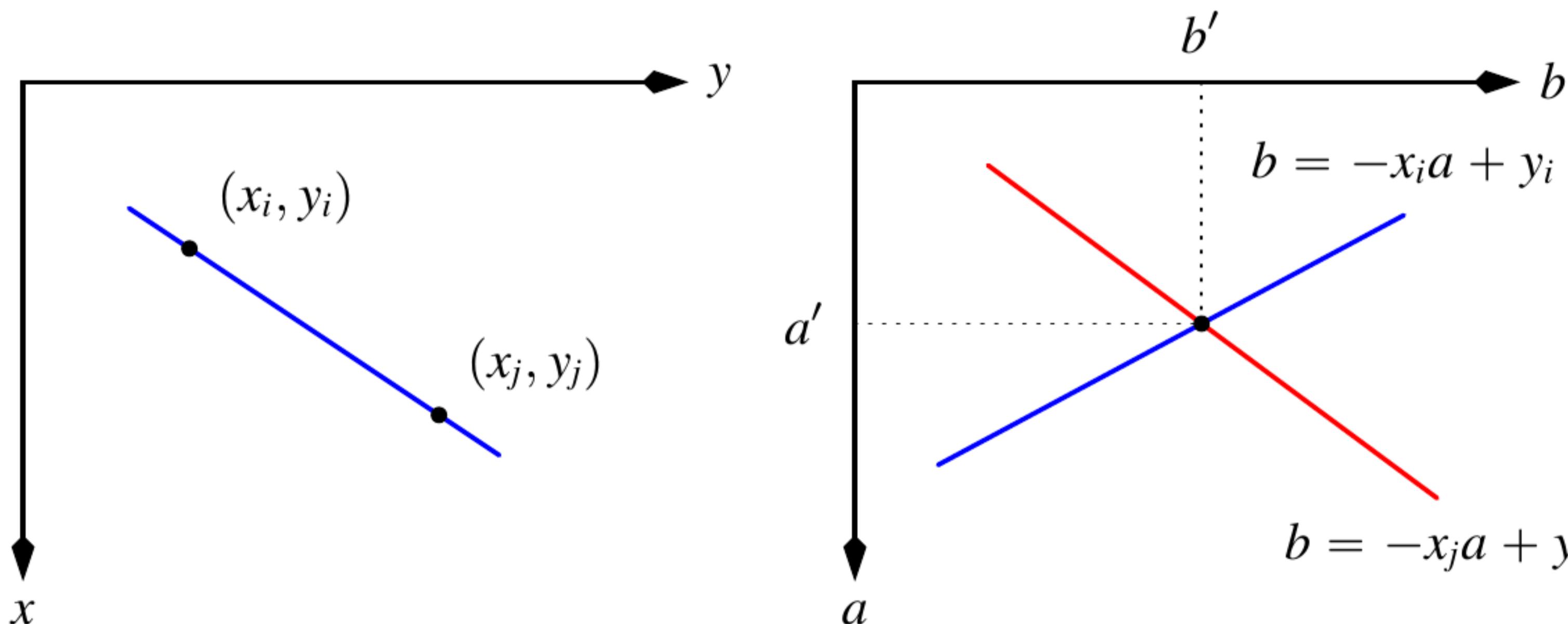
Limiarização com histerese é usada para a quebra do contorno (borda tracejada)

- Dois limiares T_1 e T_2 . $T_1 > T_2$ são usados.
- Se o pixel é tal que $g_N(x, y) \geq T_1$, é assumido como ponto de borda forte.
- Para os pixels restantes, aqueles em que $g_N(x, y) \geq T_2$, são assumidos como ponto de borda fraco.
- Para todos os vizinhos dos pontos de borda fraco, procurar nos seus 8-vizinhos se há algum ponto de borda forte. Caso haja, este é marcado como parte da fronteira.
- Sugestão de Canny: $T_H/T_L = 3/1$ ou $T_H/T_L = 2/1$.

Detecção de linhas

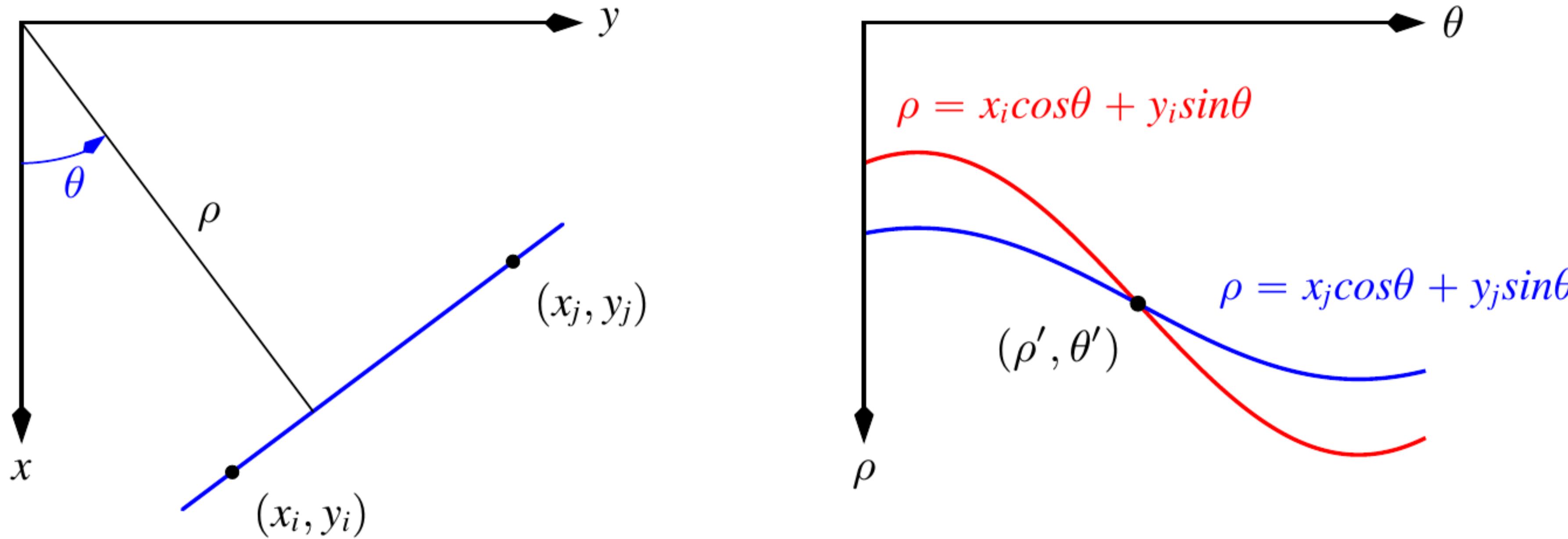
Transformada de Hough

- Tem por objetivo encontrar as equações das retas presentes em uma imagem.
- Considere dois pontos (x_i, y_i) e (x_j, y_j) . Apenas uma reta da forma $y_i = ax_i + b$ passa por ambos os pontos.
- Se a equação da reta for reescrita da forma $b = -ax_i + y_i$, a representação no espaço de parâmetros mostra uma interseção das novas retas formadas por (x_i, y_i) e (x_j, y_j) .



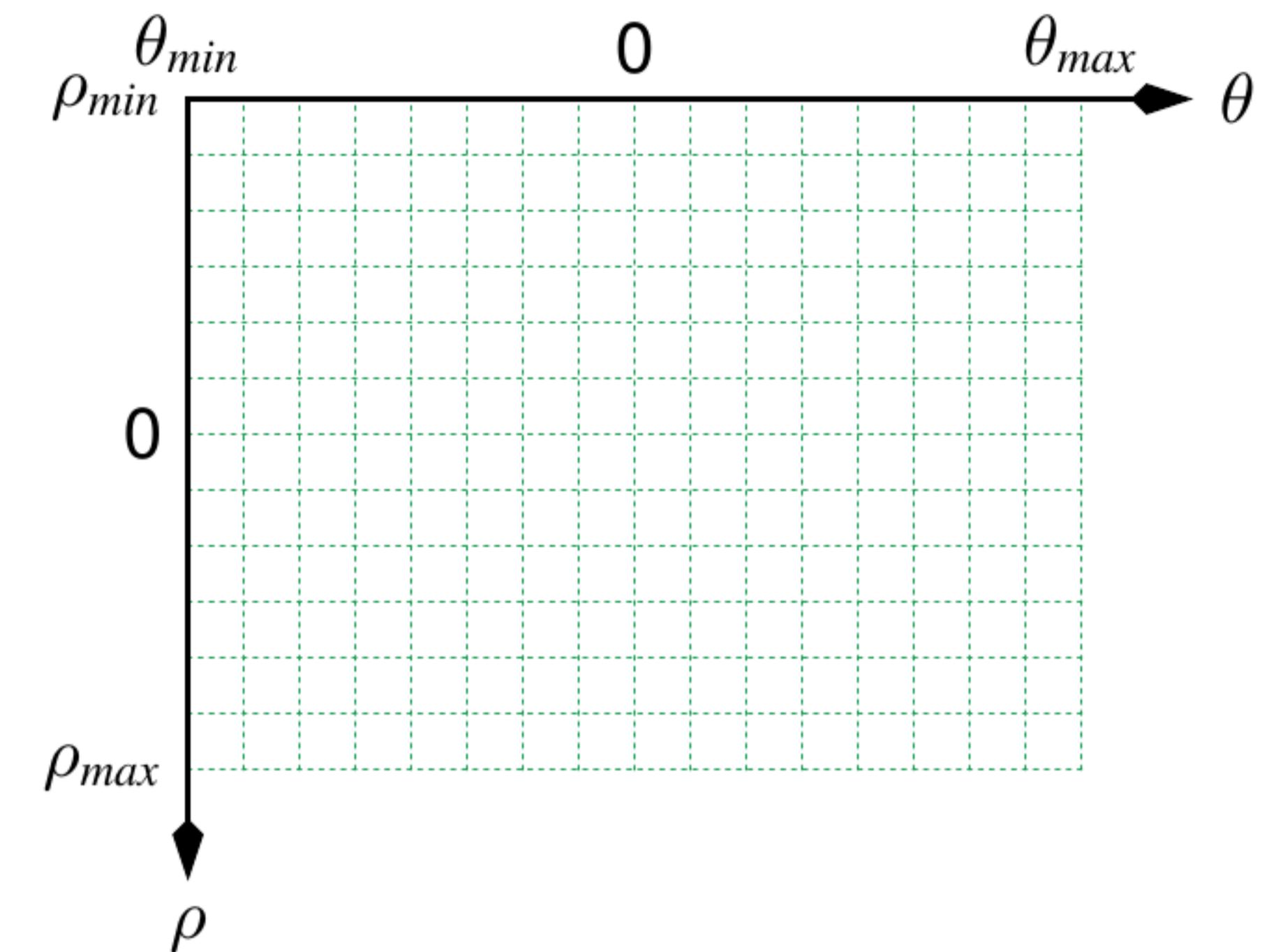
Transformada de Hough

- Problema: retas se aproximando da inclinação vertical ($a \rightarrow \infty$).
- Solução: representação em coordenadas na forma normal $[\theta, \rho]$.

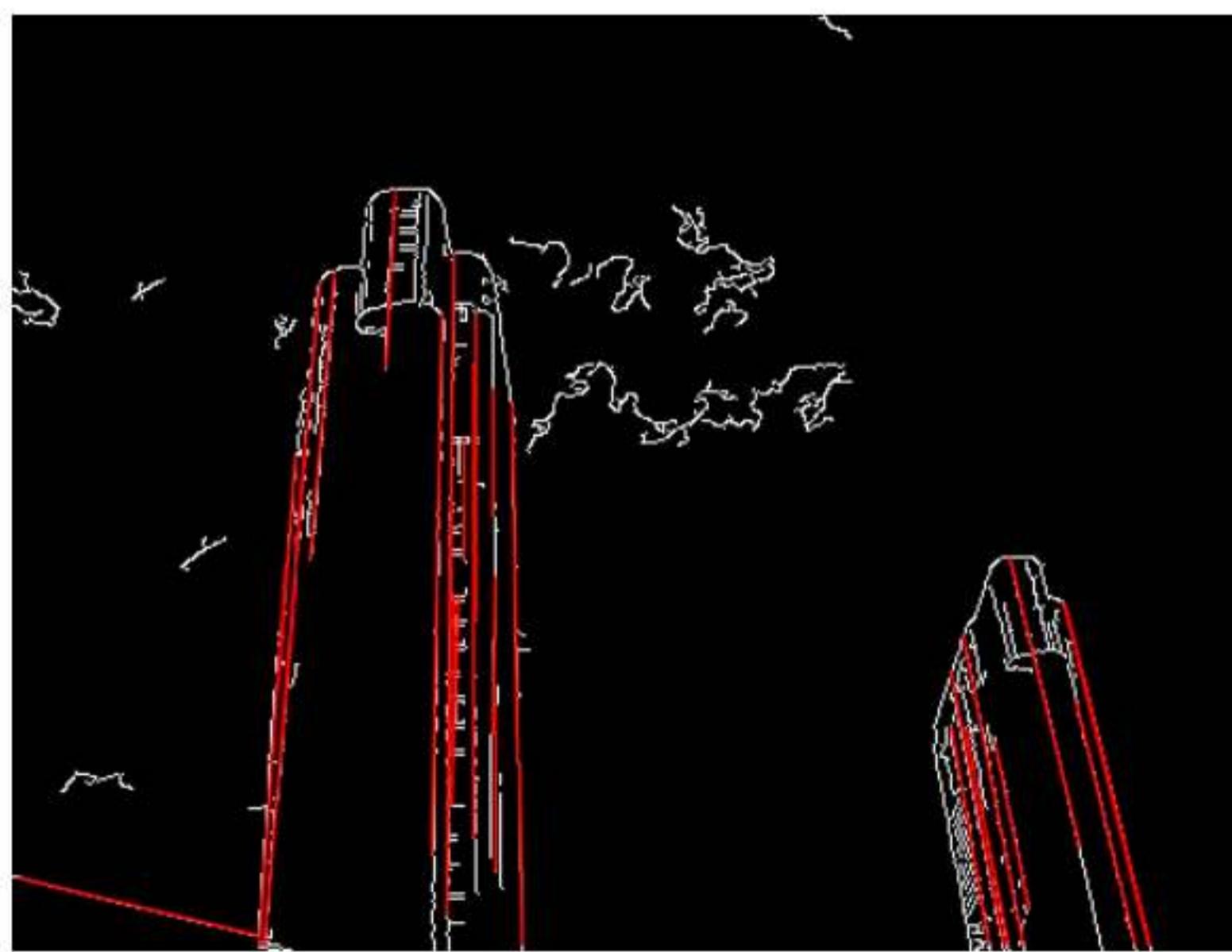
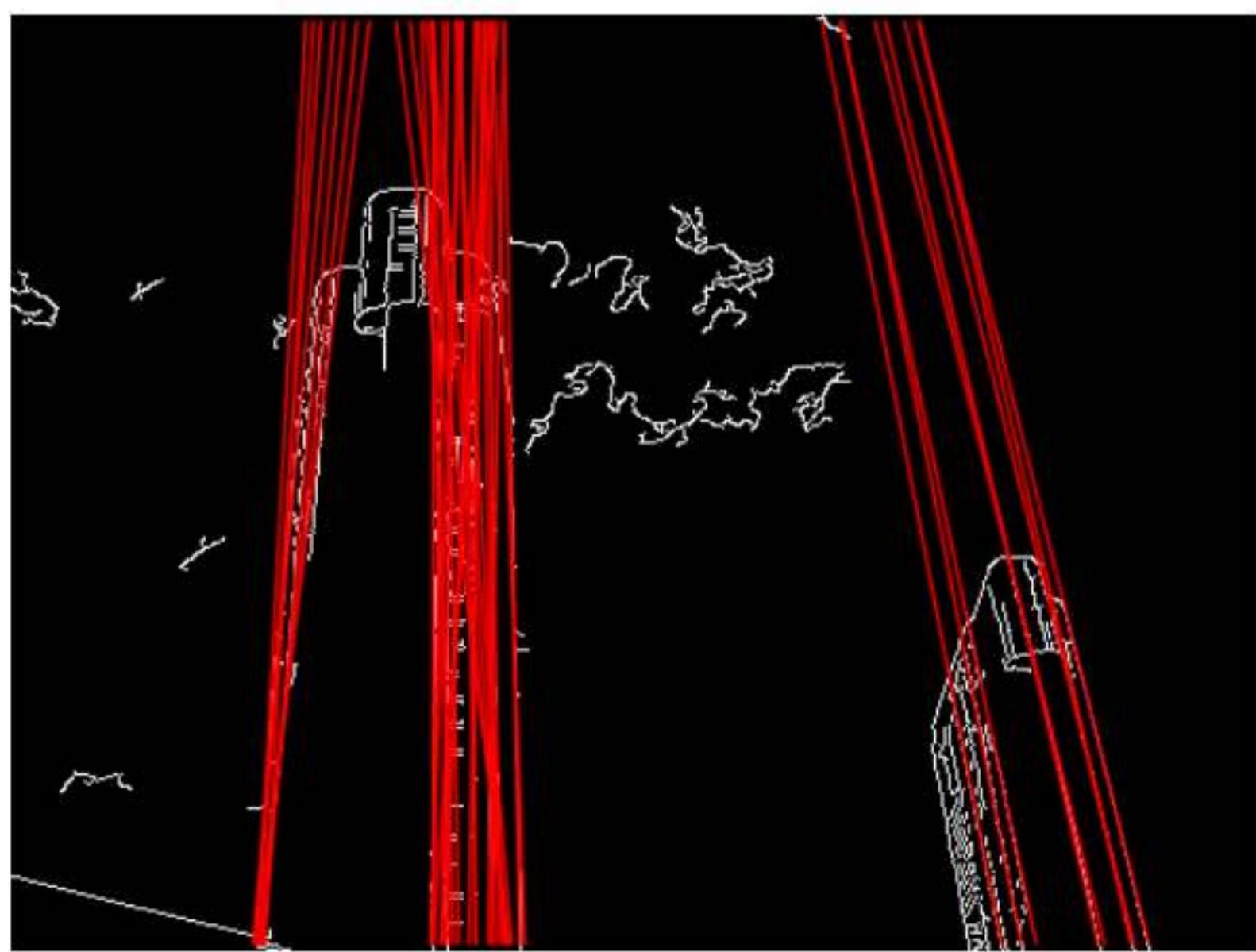


Transformada de Hough

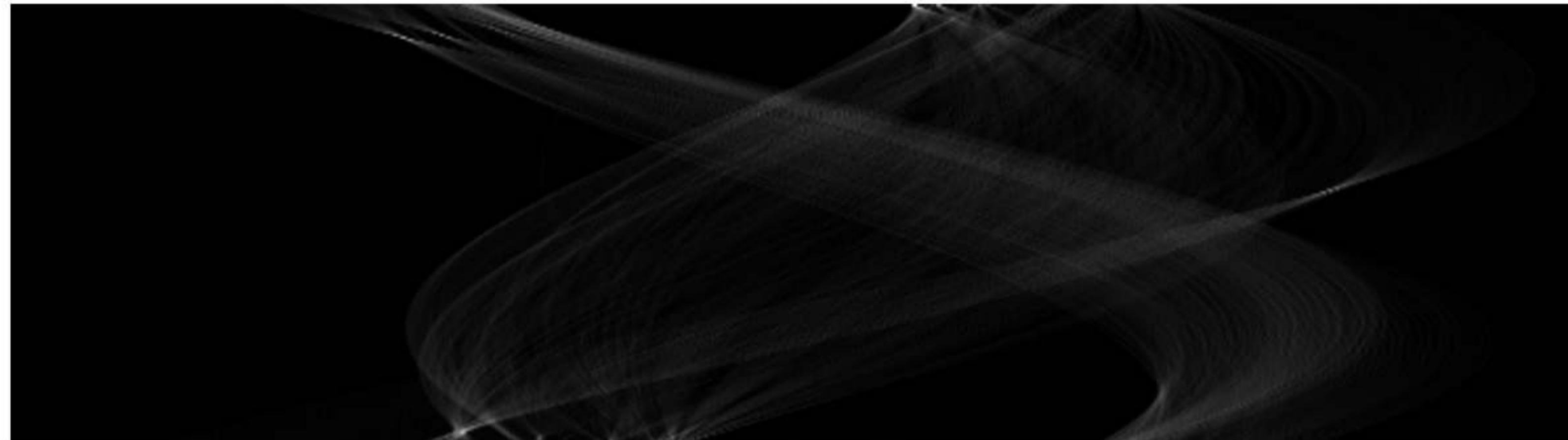
- Uso de matriz acumuladora $A(p, q)$ que subdivide o espaço de parâmetros $\rho\theta$.
- A matriz assume variações $-90^\circ \leq \theta \leq 90^\circ$ (ou $0^\circ \leq \theta \leq 180^\circ$) e $-D \leq \rho \leq D$, onde D é o diâmetro da imagem.
- Inicia-se o processo de cálculo da transformada com $A(p, q) = 0$.
- O processamento é realizado em uma imagem de borda (Ex: obtida pelo filtro de Canny).
- Para cada pixel de borda, uma função paramétrica é traçada. Para cada valor de θ incrementando, arredonda-se o valor de ρ resultante para a posição mais próxima na matriz, (p, q) . Acrescenta-se em 1 o valor de $A(p, q)$, ou seja $A(p, q) = A(p, q) + 1$.
- Examinar a matriz acumuladora em busca de máximos.
- Examinar a conectividade entre pixels de uma reta associada a uma célula escolhida.



Transformada de Hough

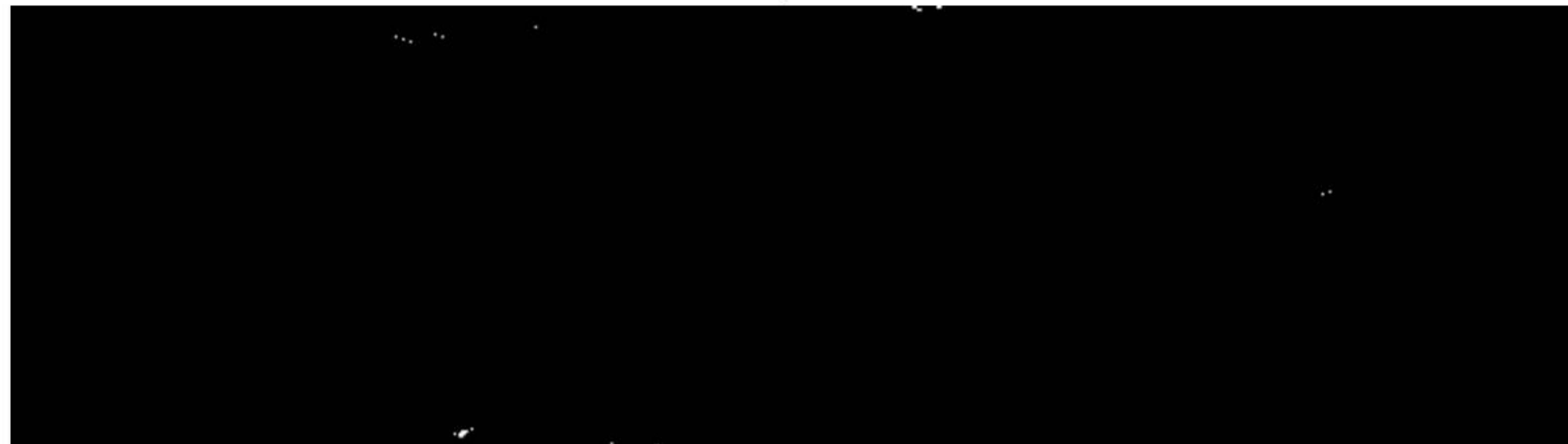


Transformada de Hough



θ

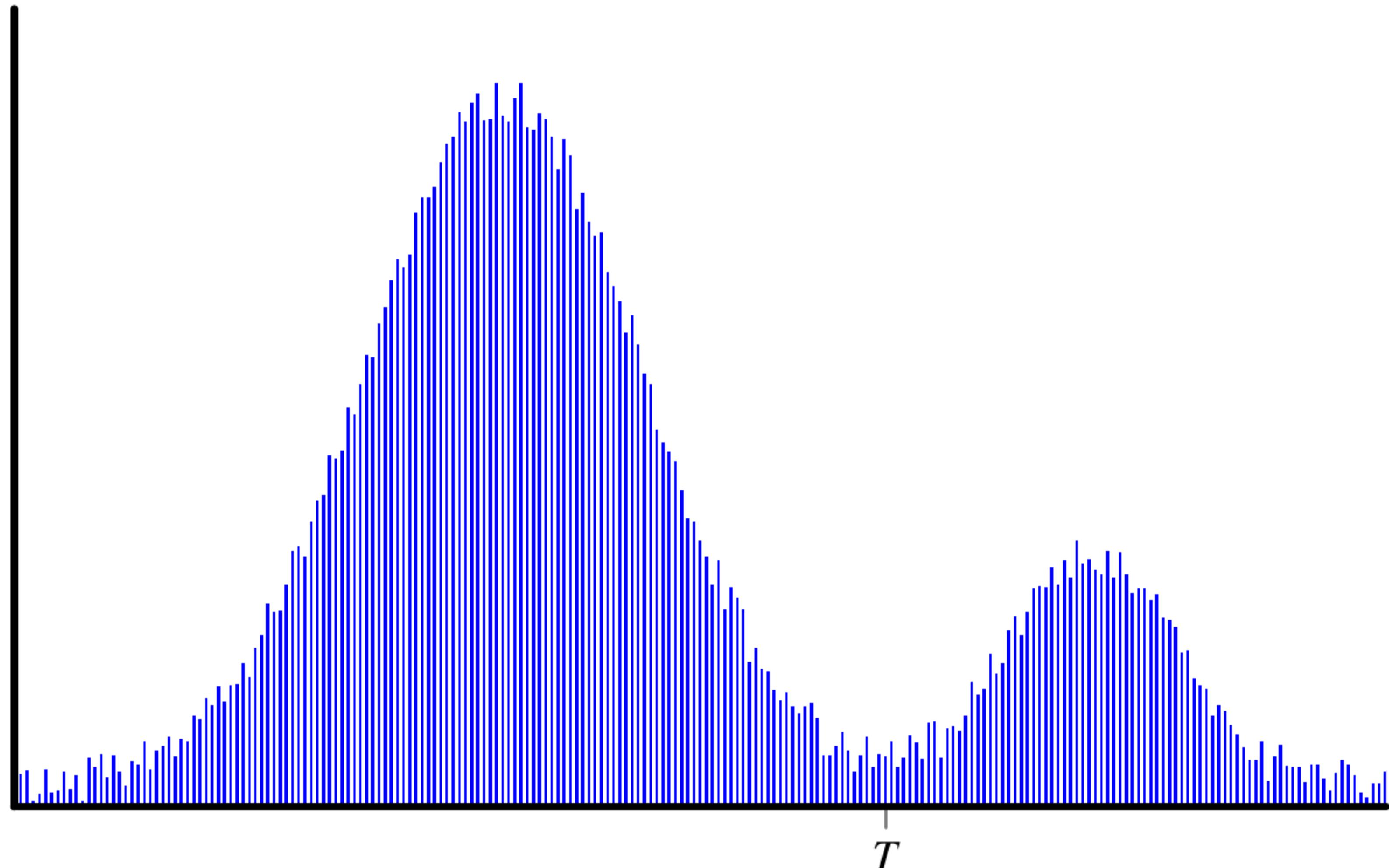
ρ



θ



Limiarização com Histogramas de intensidade



Limiarização global simples

- Selecionar uma estimativa global para o limiar, T . Ex: valor médio dos pixels da imagem.
- Segmentar a imagem em dois grupos de pixels usando T .
- Calcular as médias de tons de cinza de cada grupo, m_1 e m_2 .
- Calcular um novo valor de limiar

$$T = \frac{1}{2}(m_1 + m_2)$$

- Repetir as segmentação até que a diferença entre os limiares obtidos em duas iterações consecutivas, T_k e T_{k+1} , seja menor que um parâmetro pré-definido ΔT .
- É rápido, mas pode falhar em imagens com grandes diferenças entre as quantidades de pixels de cada grupo (objeto e fundo).

Método de Otsu

- Assume que o histograma pode ser representado por duas funções de densidade de probabilidade.
- Procura determinar um limiar ótimo de modo a maximizar a variância entre classes.
- Para um dado limiar k , a variância entre classes pode ser definida como

$$\sigma_B^2(k) = P_1(k)(m_1(k) - m_G)^2 + P_2(k)(m_2(k) - m_G)^2$$

$$\text{ou} \quad \sigma_B^2(k) = \frac{[m_G P_1(k) - m(k)]^2}{P_1(k)[1 - P_1(k)]}$$

- $m(k)$ é a média acumulada até o nível k .
- m_G é a média geral dos pixels na imagem.
- Para n_i pixels no nível i , $P_1(k)$ é dado pela soma cumulativa

$$P_1(k) = \sum_{i=0}^k p_i = \sum_{i=0}^k \frac{n_i}{MN}$$

- A variância entre classes é uma medida de SEPARABILIDADE entre as classes.

Método de Otsu

- Calcular o histograma normalizado da imagem de entrada, p_i .
- Calcular as somas acumuladas $P_i(k)$.
- Calcular as médias acumuladas $m(k)$.
- Calcular a média global m_G .
- Calcular a variância entre classes $\sigma_B^2(k)$, para $k = 0, 1, 2, \dots, L - 1$ (L níveis de cinza).
- Procurar nos valores de $\sigma_B^2(k)$ aquele que manifesta valor máximo. Se mais de um máximo for encontrado, o limiar ótimo será a média dos valores de k que produziram os máximos.
- Caso contrário, escolha k como limiar.

Melhorando a segmentação

- Suavização prévia.
- Uso de informação de bordas para processar o histograma.
- Limiarização local.
 - Particionando a imagem em blocos.
 - Realizando a limiarização em uma vizinhança de tamanho fixo.

Médias de movimento para segmentação de documentos

- Documentos normalmente possuem uma região de fundo uniforme, com texto preto superposto em um fundo branco.
- Mesmo assim, a digitalização pode produzir imagens com iluminação variável, o que pode dificultar a segmentação. Ex: documentos adquiridos por intermédio de fotografia.
- Solução: realizar a segmentação linha por linha, pixel a pixel, caminhando de um tom de cinza z_k para outro z_{k+1} , comparando o tom do píxel com uma média na sua vizinhança.
- Média em cada ponto (considerando uma vizinhança de n pontos) é dada por:

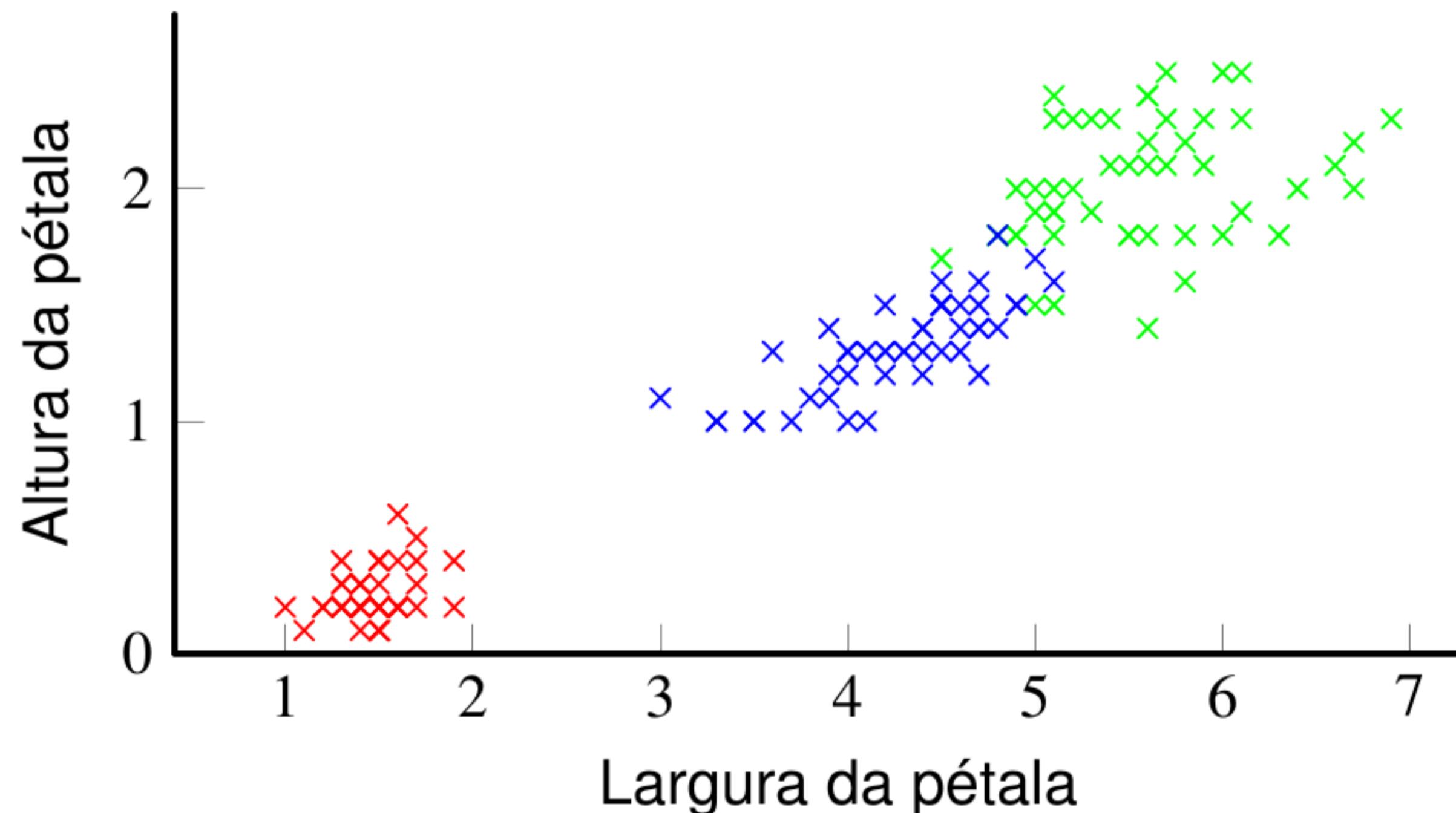
$$\begin{aligned} m(k+1) &= \frac{1}{n} \sum_{i=(k+1)-(n-1)}^{k+1} z_i \\ &= m(k) + \frac{1}{n} (z_{k+1} - z_{k-n}) \end{aligned}$$

- A ideia é remover da média o valor do pixel mais antigo e adicionar o valor do pixel mais recente.
- Uso do limiar $T_{xy} = b \times m_{xy}$, com $b = 0.5$, por exemplo. O valor de n é, via de regra, 4 a 5 vezes a largura média do traço.

Aglomeração de regiões

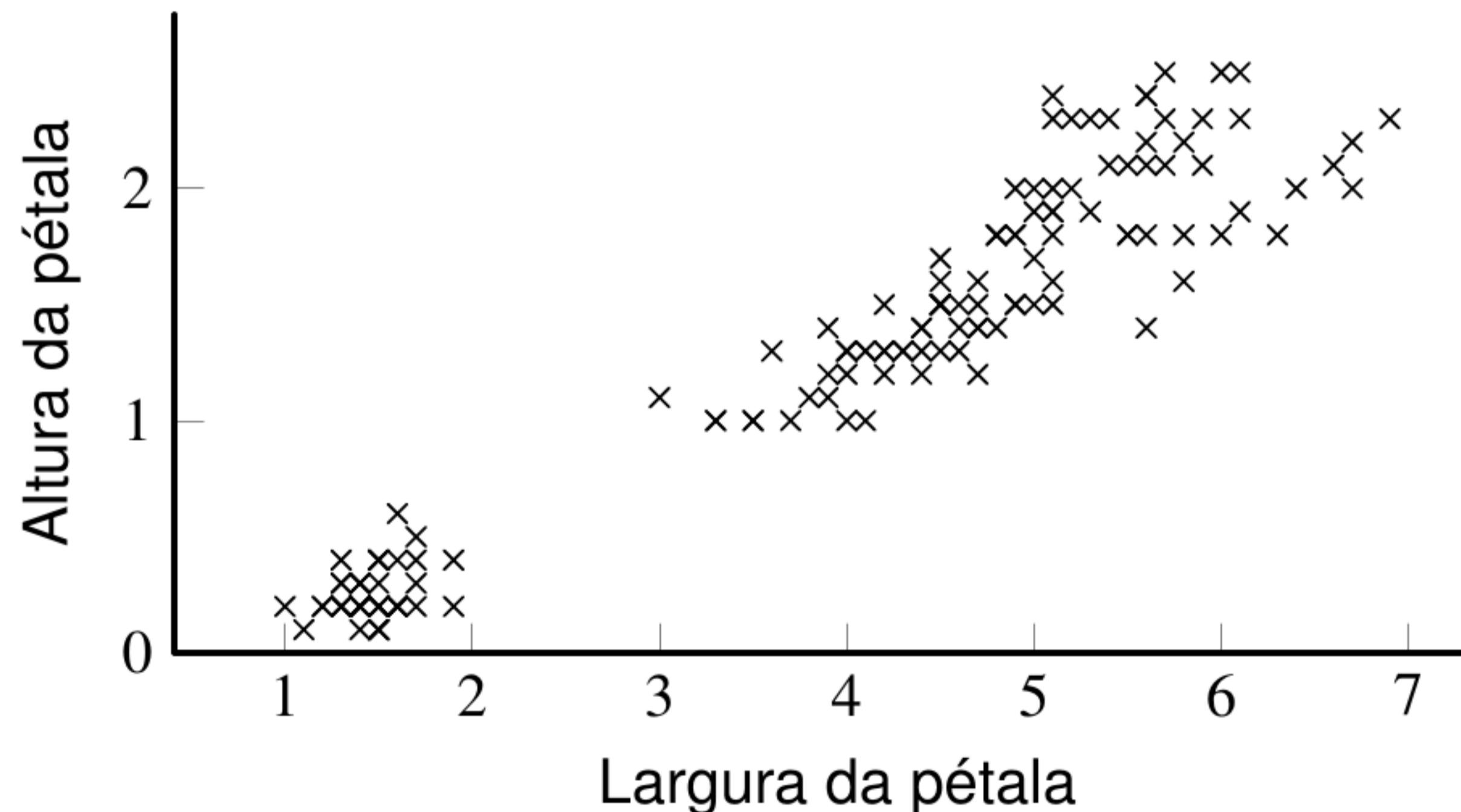
Segmentação por agrupamento com k-médias

- Algoritmo iterativo que classifica um conjunto de vetores em k classes.
- A classificação é realizada com base na distância do vetor ao centro do grupo.
- O algoritmo minimiza a soma de todas as distâncias euclidianas entre cada vetor e o centróide do seu grupo.
- Nenhum dos vetores pertence inicialmente a uma classe específica. Ex: "Iris dataset"(Iris virginica, Iris versicolor e Iris Setosa): Largura x Altura da pétala.



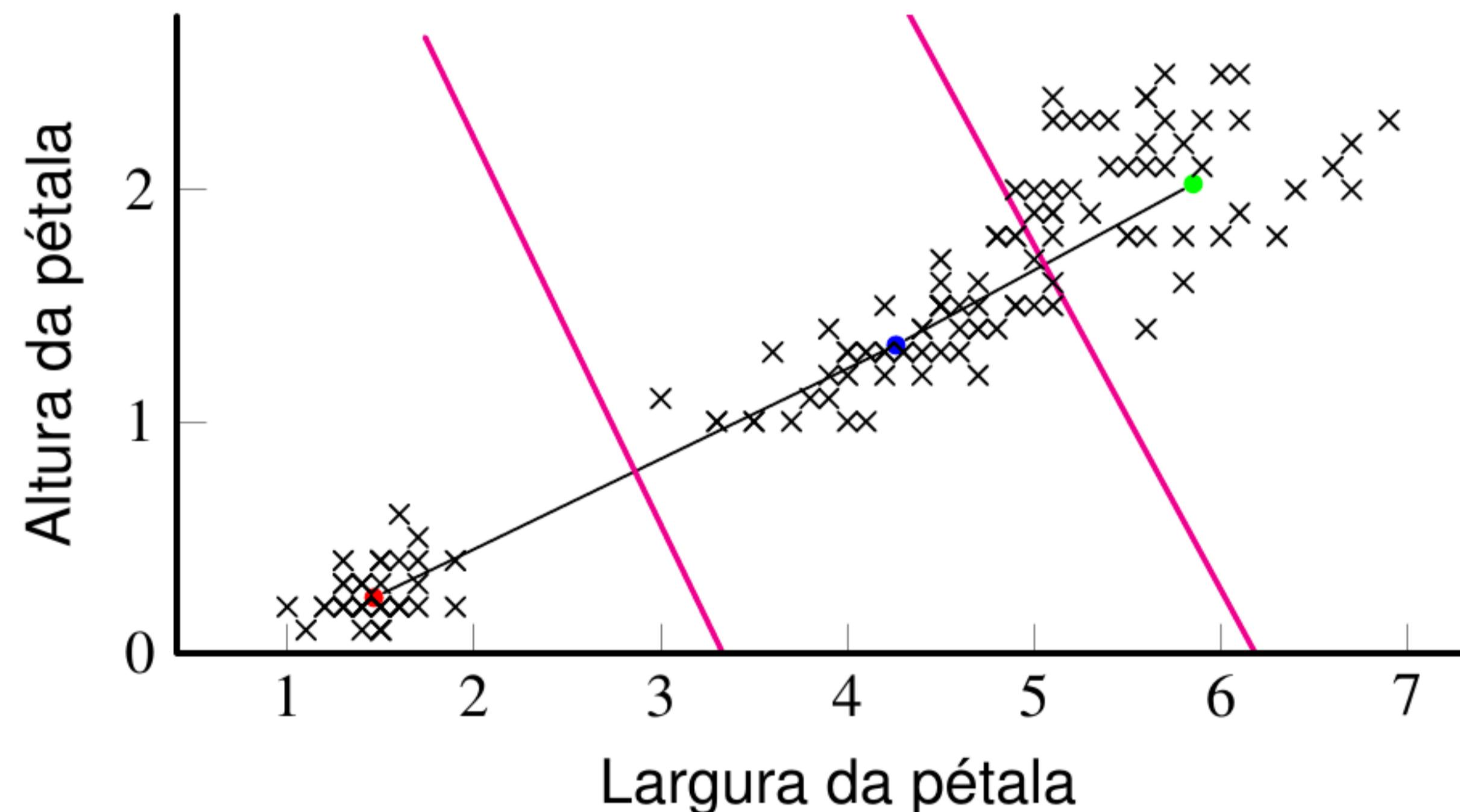
Segmentação por agrupamento com k-médias

- Algoritmo iterativo que classifica um conjunto de vetores em k classes.
- A classificação é realizada com base na distância do vetor ao centro do grupo.
- O algoritmo minimiza a soma de todas as distâncias euclidianas entre cada vetor e o centróide do seu grupo.
- Nenhum dos vetores pertence inicialmente a uma classe específica. Ex: "Iris dataset"(Iris virginica, Iris versicolor e Iris Setosa): Largura x Altura da pétala.



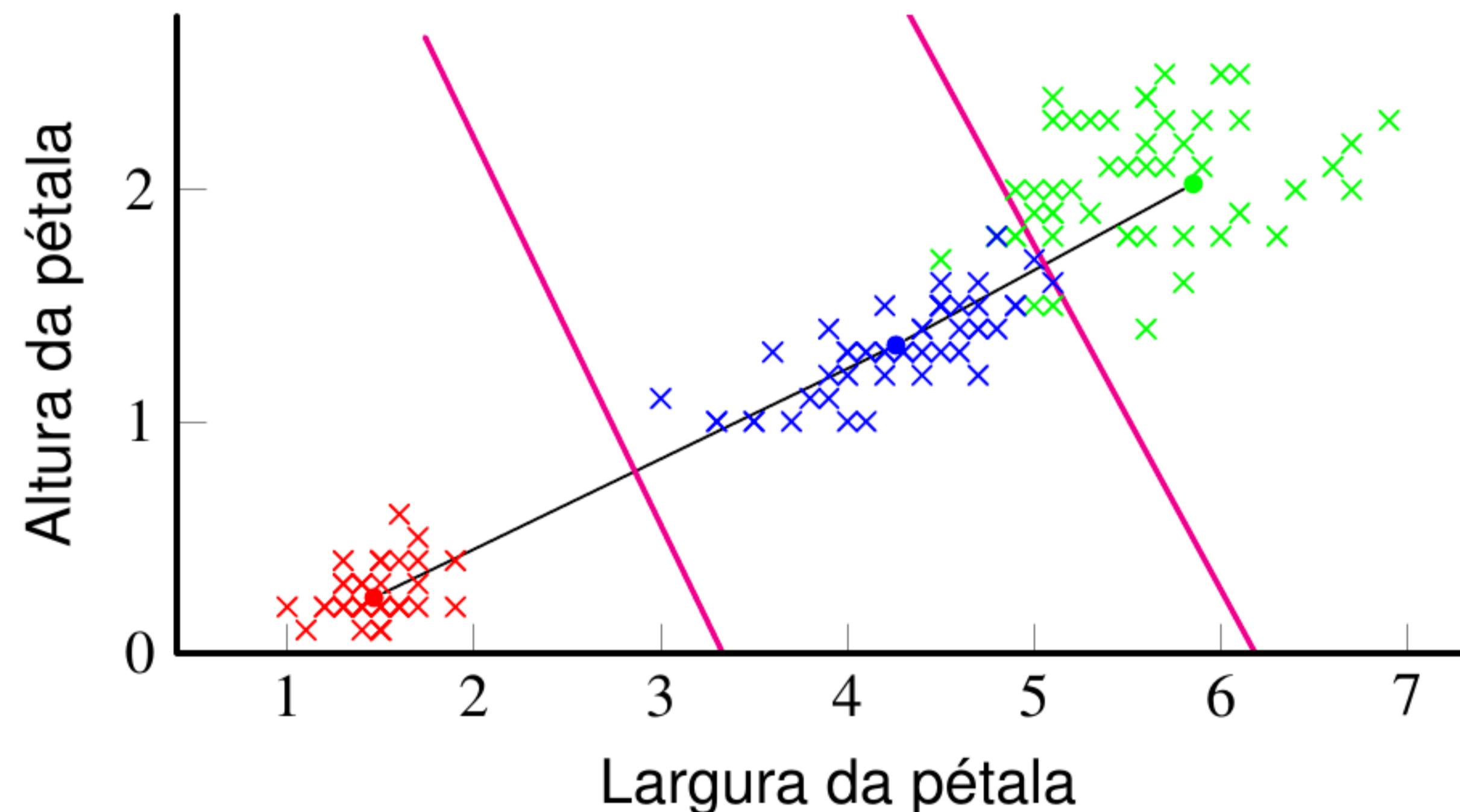
Segmentação por agrupamento com k-médias

- Algoritmo iterativo que classifica um conjunto de vetores em k classes.
- A classificação é realizada com base na distância do vetor ao centro do grupo.
- O algoritmo minimiza a soma de todas as distâncias euclidianas entre cada vetor e o centróide do seu grupo.
- Nenhum dos vetores pertence inicialmente a uma classe específica. Ex: "Iris dataset"(Iris virginica, Iris versicolor e Iris Setosa): Largura x Altura da pétala.

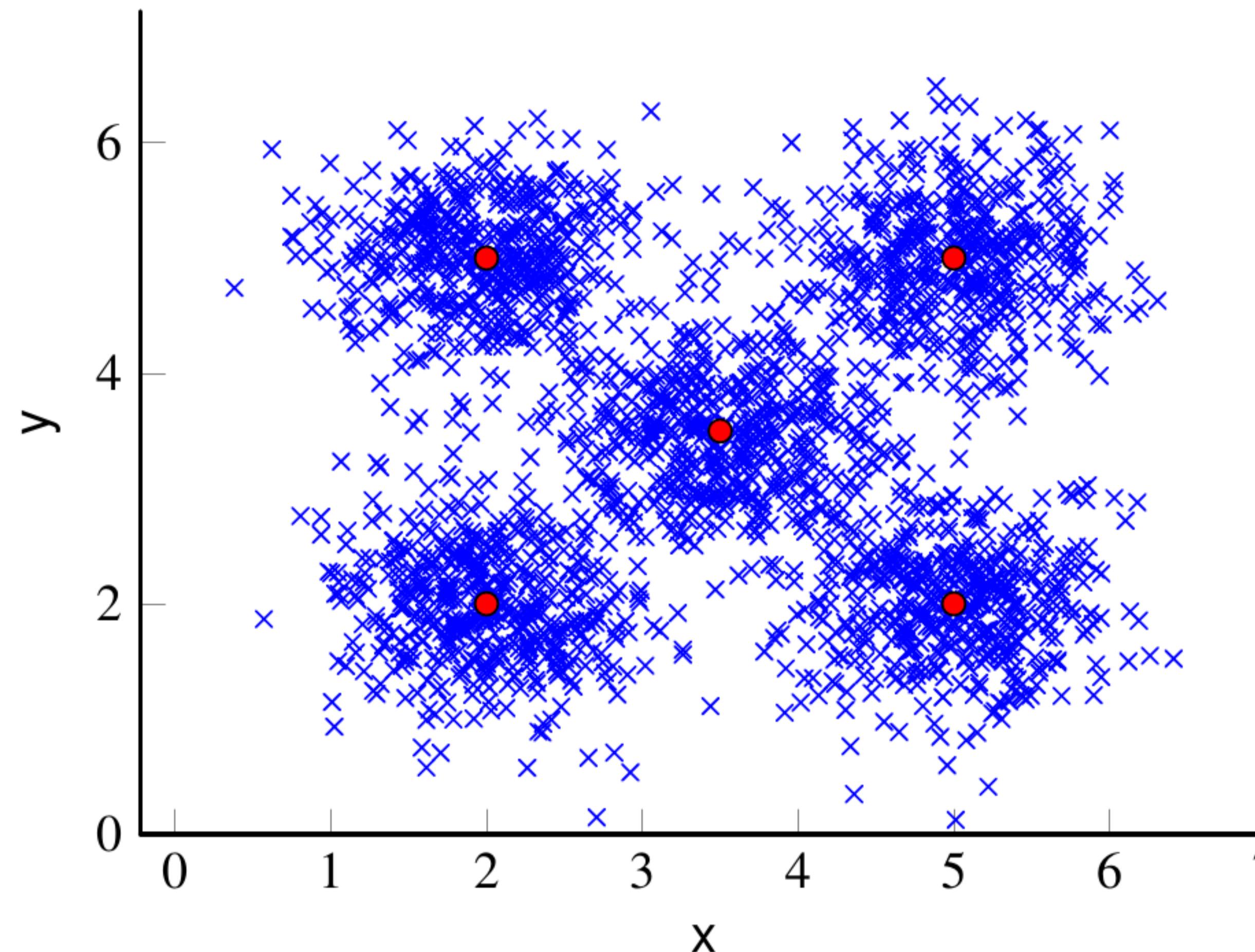


Segmentação por agrupamento com k-médias

- Algoritmo iterativo que classifica um conjunto de vetores em k classes.
- A classificação é realizada com base na distância do vetor ao centro do grupo.
- O algoritmo minimiza a soma de todas as distâncias euclidianas entre cada vetor e o centróide do seu grupo.
- Nenhum dos vetores pertence inicialmente a uma classe específica. Ex: "Iris dataset"(Iris virginica, Iris versicolor e Iris Setosa): Largura x Altura da pétala.



Segmentação por agrupamento com k-médias



Segmentação por agrupamento com k-médias

- 1 Escolha o número de classes para vetores \mathbf{x}_i de N características, $i = 1, 2, \dots, N_{amostras}$.
- 2 Escolha $\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_k$ como aproximações iniciais para os centros das classes.
- 3 Classifique cada amostra \mathbf{x}_i usando, por exemplo, um classificador de distância mínima.
- 4 Recalcule as médias \mathbf{m}_j usando o resultado de 3.
- 5 Se as novas médias são consistentes, finalize o algoritmo. Caso contrário, repita o passo 3 com as novas médias obtidas.

Observação

Diferentes execuções do algoritmo k-médias podem levar a diferentes centros. É comum executar o algoritmo algumas vezes e escolher a melhor compactação dos dados.

Segmentação por agrupamento com k-médias

Parâmetros

8 centros

1000 iterações



Segmentação por agrupamento com k-médias



2

4

8

16

32