



Departamento de Engenharia Elétrica - EESC-USP



SEL-0339 Introdução à Visão Computacional

Aula 8

Segmentação de Imagens

Parte 2 - Similaridade

Prof. Dr. Marcelo Andrade da Costa Vieira
Rodrigo de Barros Vimieiro

- **Segmentação** é o processo que subdivide uma imagem em regiões que satisfaçam alguns critérios de similaridade ou descontinuidade pré-definidos.
- Geralmente a segmentação é baseada em propriedades dos níveis de cinza da imagem como:

Descontinuidade ou Similaridade

- **Descontinuidade:** Segmentação baseada em Bordas:
 - Detector de bordas, linhas ou pontos isolados
 - Baseado em filtros gradiente ou Laplaciano
- **Similaridade:** Segmentação baseada em Regiões:
 - Limiarização, Crescimento de Regiões, K-Means, etc..



Relações Básicas entre Pixels

Relações Básicas entre Pixels

Imagem \implies $f(x,y)$

Pixel \implies p,q,r,\dots

Sub-conjunto de Pixels \implies S

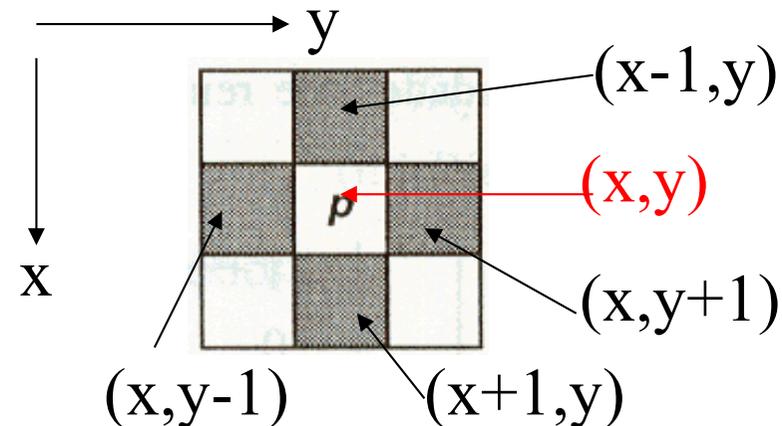
Adjacência entre Pixels:

Um par de Pixels de uma Imagem que compartilham uma borda é dito “**adjacente por borda**” ou **4-adjacente**.

Um par de Pixels de uma Imagem que compartilham um vértice é dito “**adjacente por vértice**” ou **8-adjacente**.

Vizinhança de um Pixel

Um Pixel p na coordenada (x,y) tem a seguinte vizinhança na vertical e na horizontal, chamada de $N_4(p)$

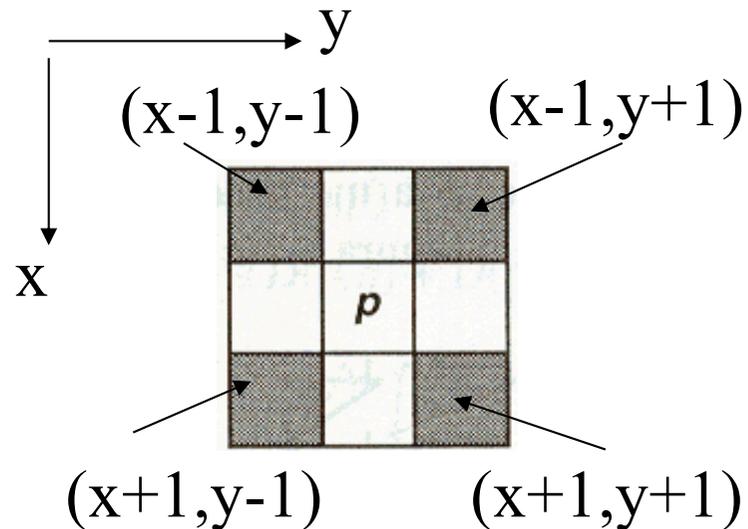


A **Vizinhança-4** de um Pixel p é um sub-conjunto de Pixels que são 4-adjacentes a ele mais o Pixel em si.

$$N_4(p) = (x-1,y) (x,y-1) (x,y) (x,y+1) (x+1,y)$$

Vizinhança de um Pixel

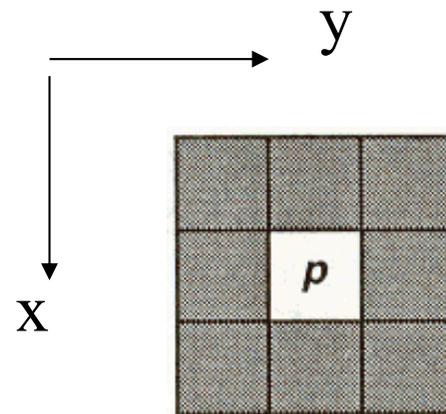
A **Vizinhança Diagonal** de p é formada pelos quatro Pixels que compartilham **somente um vértice** com p .



$$N_D(p) = (x-1, y-1) (x-1, y+1) (x+1, y-1) (x+1, y+1)$$

Vizinhança de um Pixel

A **Vizinhança-8** de um Pixel p é um sub-conjunto de 8 Pixels que são adjacentes por vértice (8-adjacentes) a ele, mais o Pixel em si.



$$N_8(p) = N_4(p) \cup N_D(p)$$

$$N_8(p) = (x-1, y-1) (x-1, y) (x-1, y+1) (x, y-1) (x, y) (x, y+1) (x+1, y-1) (x+1, y) (x+1, y+1)$$

Conceito usado para estabelecer fronteiras de objetos e regiões em uma Imagem.

Dois Pixels são conectados se:

1. São adjacentes
2. Seus níveis de cinza satisfazem a um critério especificado de similaridade

Exemplo: Imagem com 2 níveis de cinza (0 e 1). Dois pixels vizinhos de 4 são conectados se tiverem o mesmo valor.

Seja $V = \{ G_1, G_2, \dots, G_k \}$ o conjunto de “k” valores de níveis de cinza usados para definir a conectividade.

Define-se:

a) **Conectividade-4** : dois pixels p e q com valores em V e

$$q \supset N_4(p)$$

b) **Conectividade-8** : dois pixels p e q com valores em V e

$$q \supset N_8(p)$$

c) **Conectividade-m** : conectividade mista. Dois pixels p e q com valores em V e:

i) $q \supset N_4(p)$

ii) $q \supset N_D(p)$ e $N_4(p) \cap N_4(q) = \emptyset$

Exemplo

Seja $V = \{1,2\}$ os valores que definem a conectividade.

Conectividade-4

0	1	1
0	2	0
0	0	1

Conectividade-8

0	1	1
0	2	0
0	0	1

Pixel **p**

Conectividade-m

0	1	1
0	2	0
0	0	1

A **conectividade-m** elimina a dupla conexão ou o “caminho duplo” entre dois pixels **p** e **q**.

- ❑ Uma sequência de Pixels (p_1, p_2, \dots, p_n) é uma sequência 4-conectada se p_i é 4-adjacente a p_{i+1} .
- ❑ Uma sequência é 8-conectada se p_i é 8-adjacente a p_{i+1} .

Seja S um sub-conjunto não vazio de pixels de uma Imagem.

S é 4-conectado se:

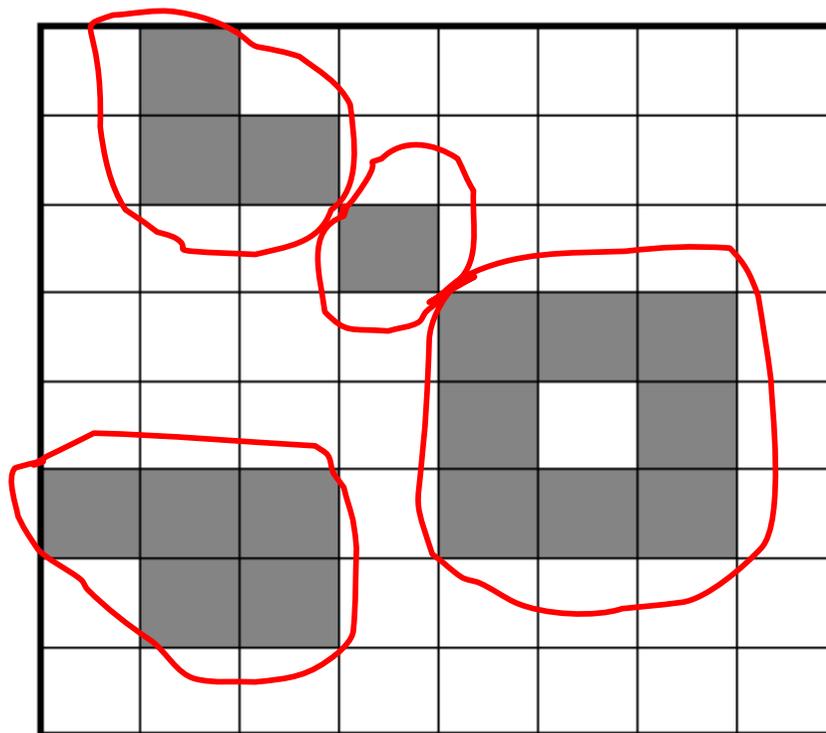
- 1) Para quaisquer dois pixels a e b , existe uma sequência de pixels 4-conectada ($p_1, p_2, p_3, \dots, p_n$) em S tal que $p_1 = a$ e $p_n = b$.
- 2) S contém somente um pixel.

S é 8-conectado se:

- 1) Quaisquer dois pixels em S são pontos de início e fim para uma sequência de pixels 8-conectada em S.
- 2) S é um sub-conjunto de um só tipo de pixel.

Exemplo

Sequências 4-conectadas para $V = \{1\}$



4 sequências de conectividade-4 para $V = \{1\}$

A distância entre dois pixels em uma imagem é uma medida fundamental para muitos algoritmos.

Dados os pixels {
p de coordenadas (x_1, y_1)
q de coordenadas (x_2, y_2)
t de coordenadas (x_3, y_3)

A função de distância D , obedece às seguintes propriedades:

- Não Negatividade $\rightarrow D(p, q) \geq 0$ para todo p e q
- Identidade $\rightarrow D(p, q) = 0$ somente se $p = q$
- Simetria $\rightarrow D(p, q) = D(q, p)$ para todo p e q
- Desigualdade Triangular $\rightarrow D(p, q) + D(q, t) \geq D(p, t)$ para todo p, q e t

Para mosaicos quadrados, as métricas mais utilizadas para dois pixels $p(x_1, y_1)$ e $q(x_2, y_2)$, são:

Distância Euclidiana:

$$D_e(p, q) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Distância City-Block (Distância D_4):

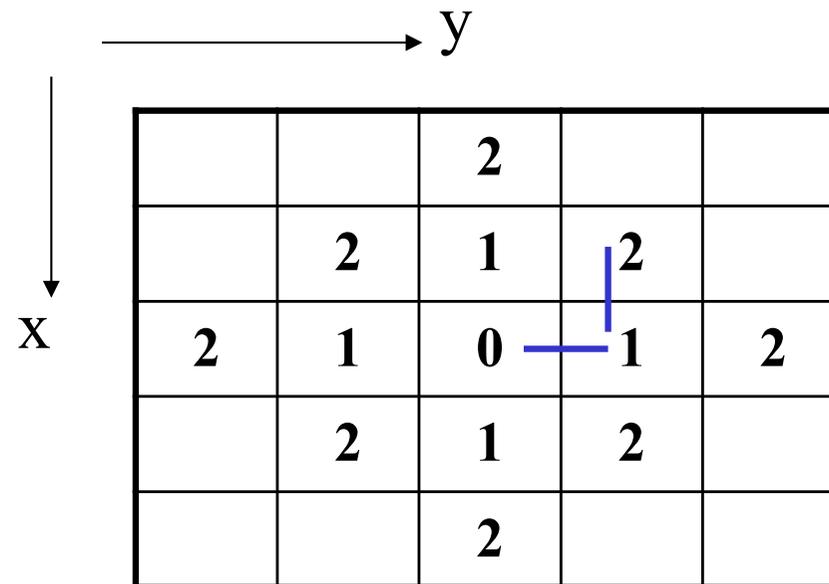
$$D_{cb}(p, q) = |x_1 - x_2| + |y_1 - y_2|$$

Distância Chessboard (Distância D_8):

$$D_{ch}(p, q) = \max \{ |x_1 - x_2|, |y_1 - y_2| \}$$

As distâncias D_4 e D_8 consideram apenas as coordenadas dos pixels, independente da existência de conectividade entre eles. A D_4 é medida pelo caminho de 4 mais curto e a D_8 pelo caminho de 8 mais curto.

Exemplo: Pixels cuja distância City Block $D_4 \leq 2$

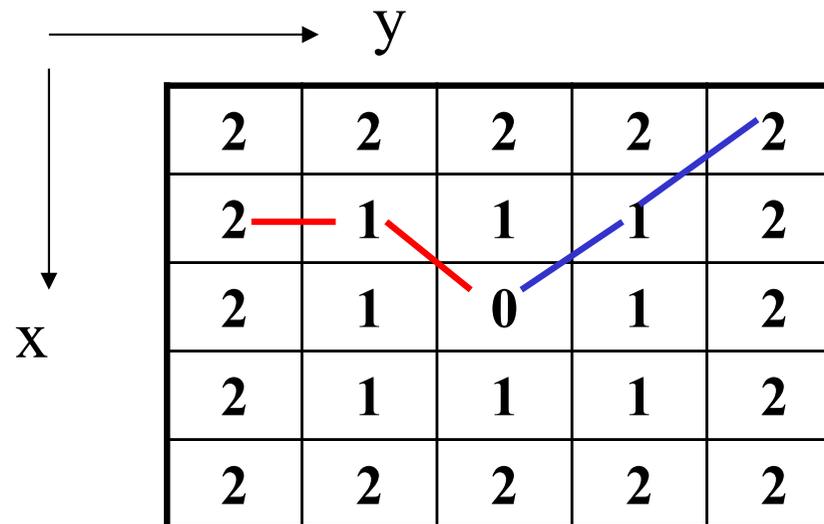


$$D_{cb}(p, q) = |x_1 - x_2| + |y_1 - y_2|$$

Exemplo

Pixels cuja distância Chessboard $D_8 \leq 2$

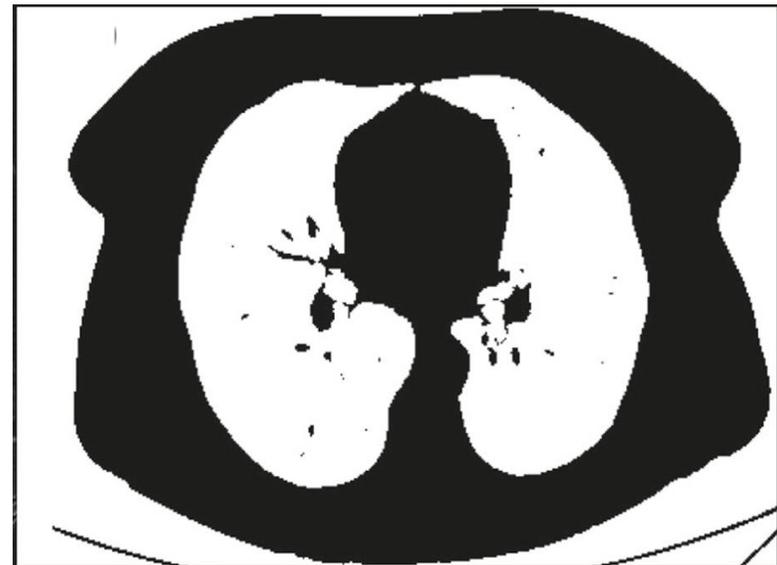
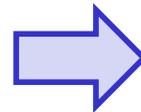
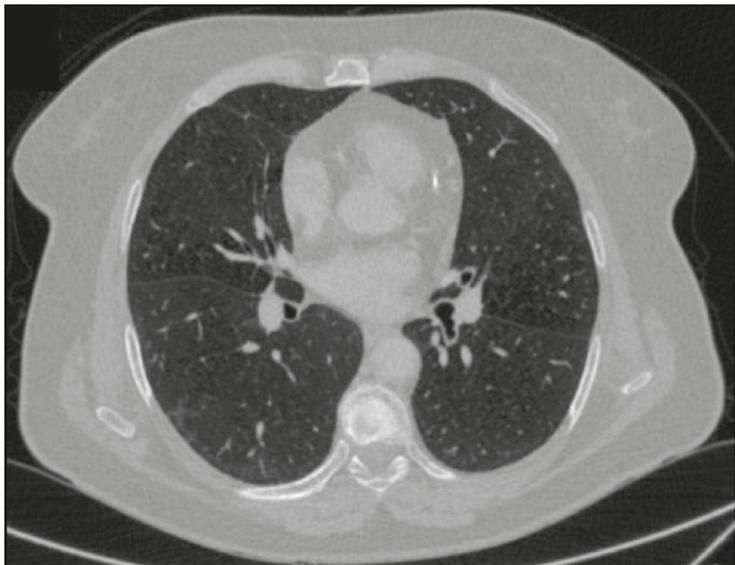
Caminho de 8 mais curto.



$$D_{ch}(p, q) = \max \{ |x_1 - x_2|, |y_1 - y_2| \}$$

Parte 2:

Segmentação por Similaridade



- **Similaridade:**
 - Baseado em regiões.
 - Agrupamento de pixels de regiões que são similares baseado em algum critério pré-definido (conectividade).
 - Exemplo: Limiarização (binarização), crescimento de região, K-means, entre outras.

1) Limiarização (Binarização ou *Thresholding*):

Modos de se escolher o limiar:

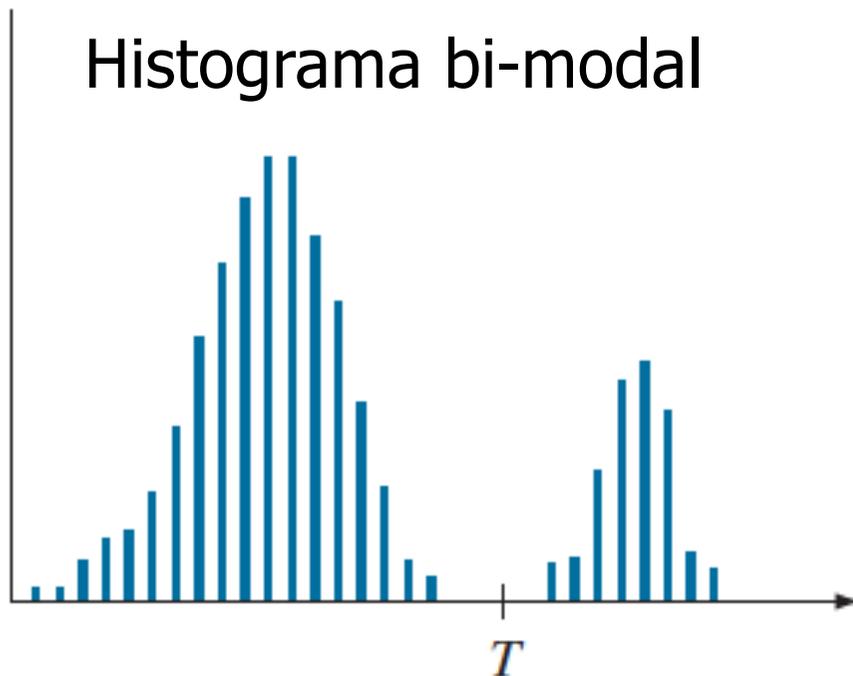
1. Inspeção visual do histograma
2. Limiar automático

Tipos de limiarização:

1. Limiarização global
2. Limiarização local (variável ou dinâmica)

1) Segmentação por Limiarização

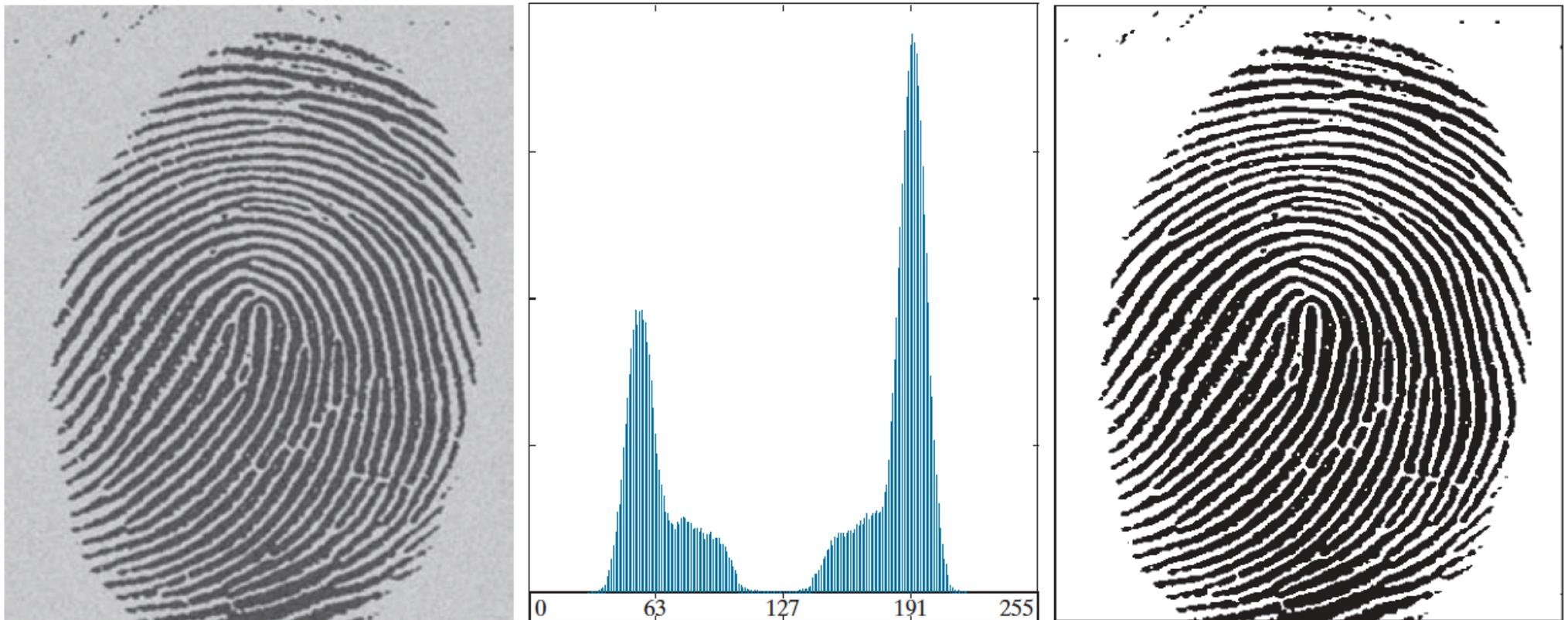
1. Inspeção visual do histograma: Método do Vale



- Imagem $f(x, y)$ composta de objetos brilhantes sobre fundo escuro
- Um ponto (x, y) é parte dos objetos se $f(x, y) > T$

O sucesso da segmentação depende da largura e da profundidade do vale que separa as duas classes

1) Método do Vale



$$T = 125$$

1) Segmentação por Limiarização:

2. Limiar automático por método interativo (tentativa e erro)

- Utiliza um processo interativo;
- O usuário testa diferentes níveis de *Threshold* até produzir um resultado satisfatório de acordo com um critério;
- **Limiarização global ótima:** método baseado na maximização da variância – (Método de Otsu, 1979)

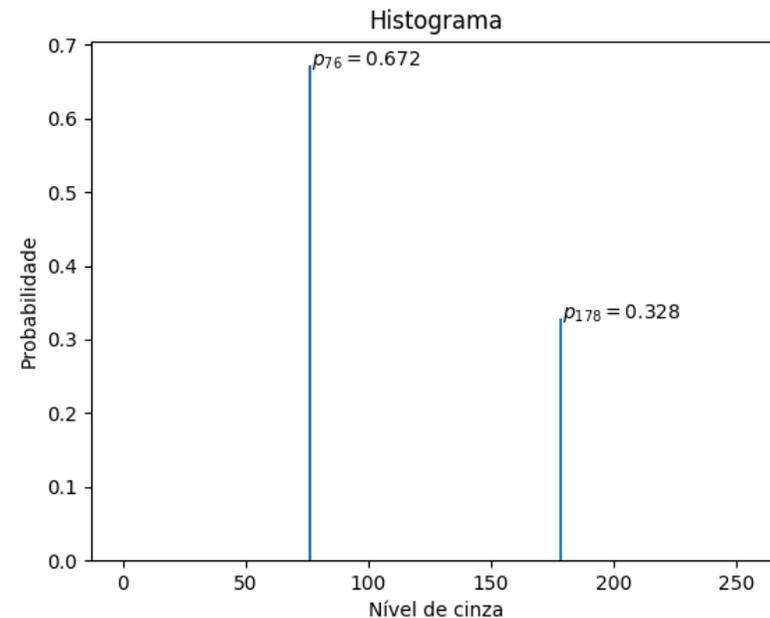
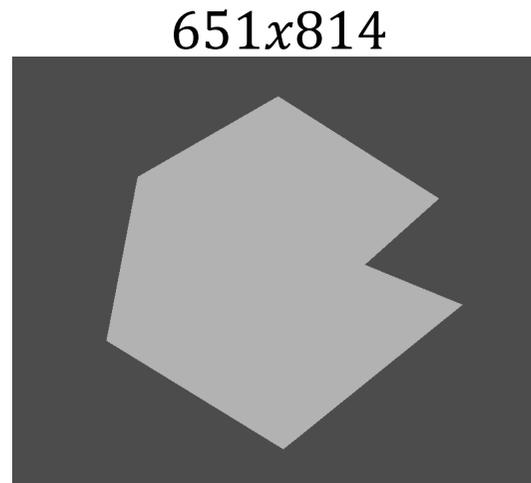
2) Limiarização Global Ótima: Método de Otsu

I. Tratar o Histograma da Imagem como uma Função de Probabilidade:

$$p_i = \frac{n_i}{M \times N}$$

$$i = 0, 1, 2, \dots, L-1$$

$$\sum_{i=0}^{L-1} p_i = 1$$



$M \times N$ = número total de pixels da imagem;

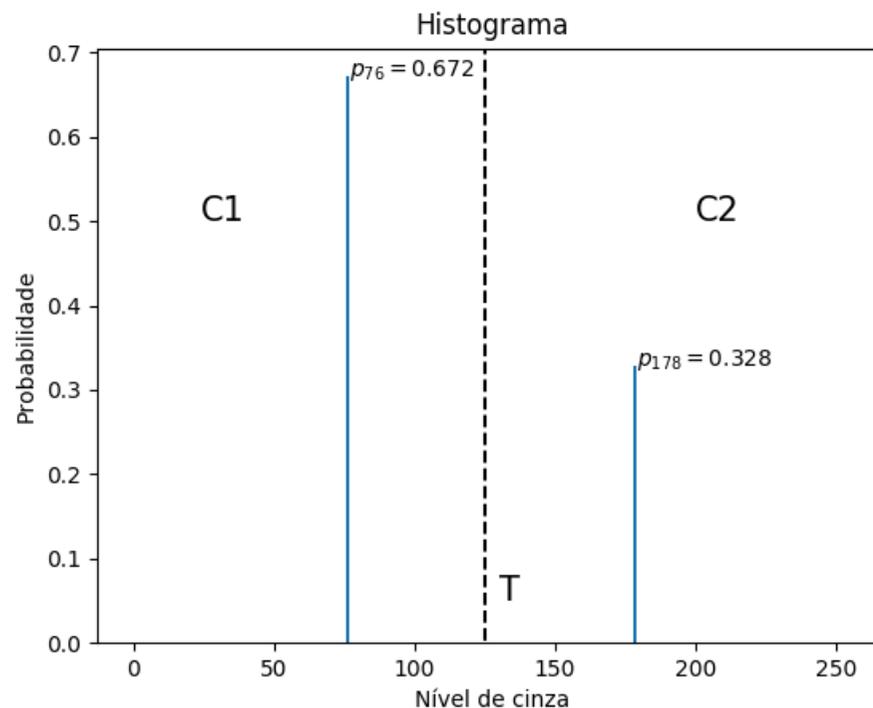
n_i = número de pixels com intensidade i ;

L = número total dos possíveis níveis de intensidade da imagem.

2) Limiarização Global Ótima: Método de Otsu

II. Escolher um valor T para o *Threshold* tal que:

- C1 seja a classe de pixels com níveis entre $[0, T]$
- C2 seja a classe de pixels com níveis entre $[T+1, L-1]$



2) Limiarização Global Ótima: Método de Otsu

III. A probabilidade de um pixel pertencer a classe C_1 é:

$$P_1(T) = \sum_{i=0}^T p_i$$

IV. A probabilidade de um pixel pertencer a classe C_2 é:

$$P_2(T) = \sum_{i=T+1}^{L-1} p_i = 1 - P_1(T)$$

2) Limiarização Global Ótima: Método de Otsu

V. A intensidade média dos pixels atribuídos à classe C_1 é:

$$m_1(T) = \frac{1}{P_1(T)} \sum_{i=0}^T ip_i$$

VI. A intensidade média dos pixels atribuídos à classe C_2 é:

$$m_2(T) = \frac{1}{P_2(T)} \sum_{i=T+1}^{L-1} ip_i$$

2) Limiarização Global Ótima: Método de Otsu

VII. A intensidade média global (de todo o histograma) é:

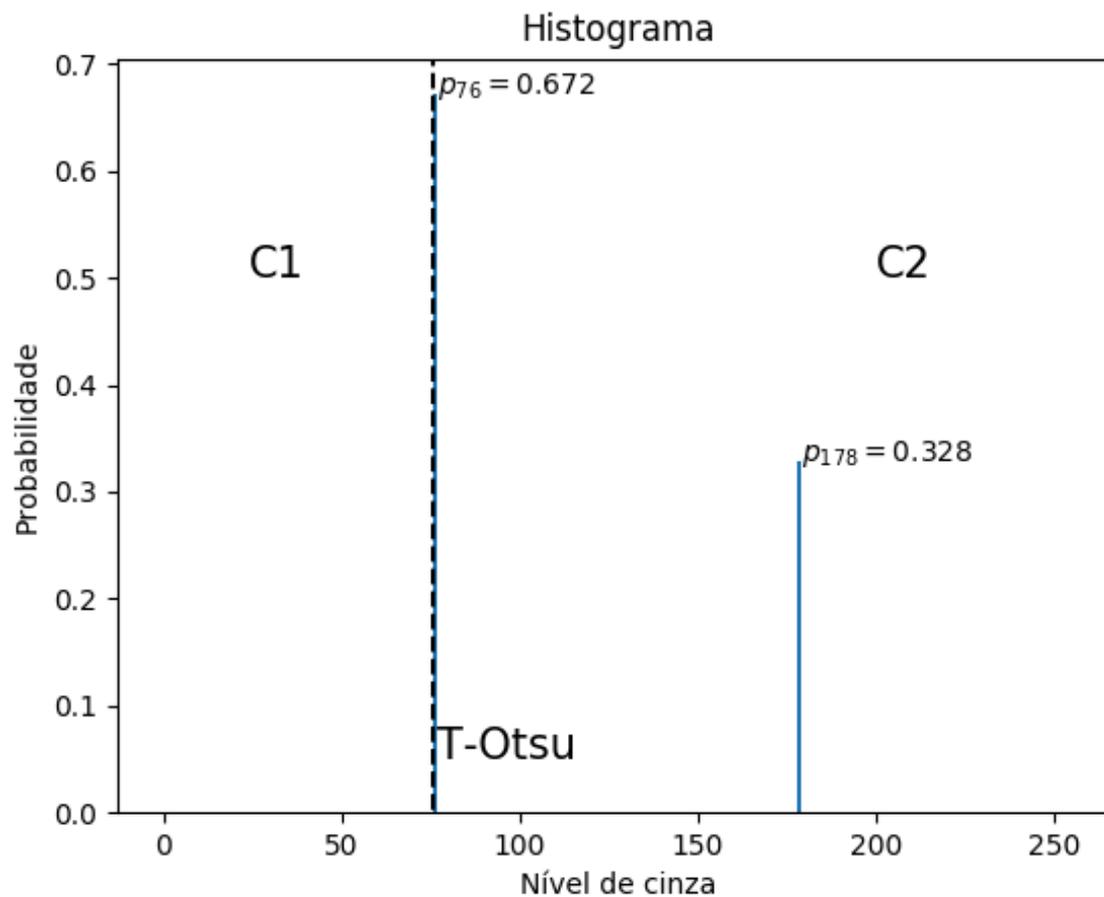
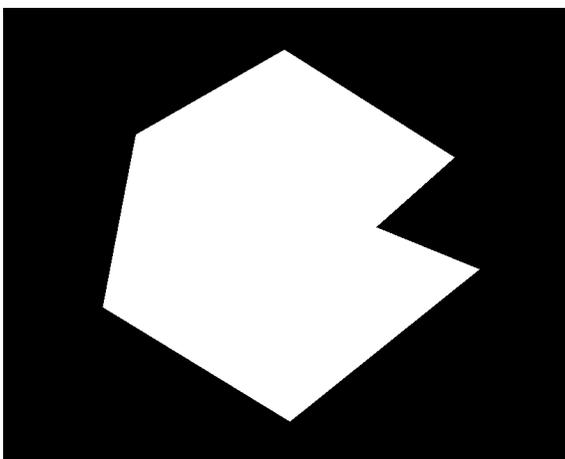
$$m_G = \sum_{i=0}^{L-1} ip_i$$

VIII. O método de Otsu escolhe **T** tal que maximize a variância entre as classes:

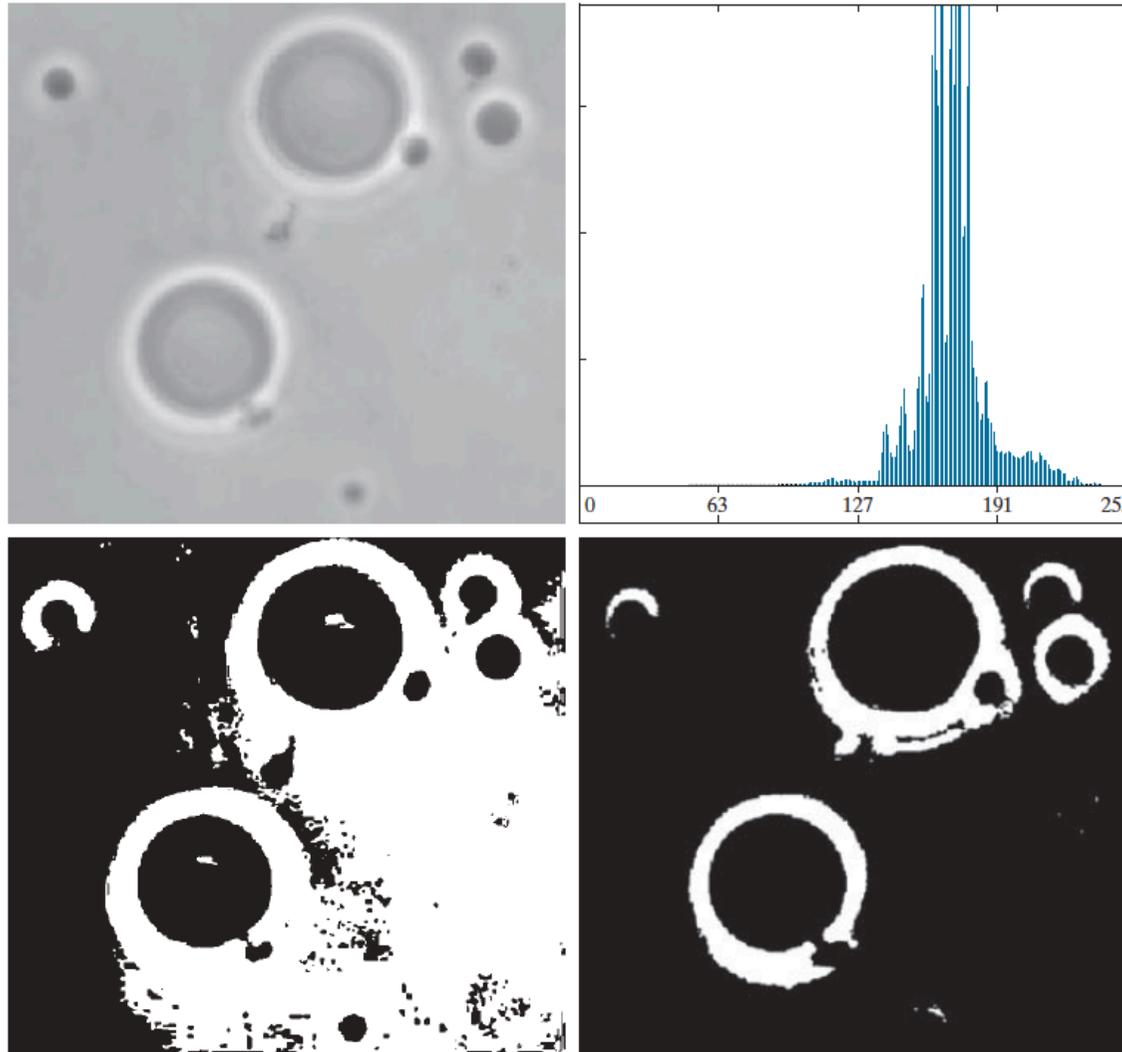
$$\sigma^2(T) = P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2$$

O algoritmo escolhe o primeiro valor de T (T = 1) e calcula a variância. Em seguida, vai incrementando o valor de T e calculando a variância para todos os casos. O valor de T escolhido é o que produz variância máxima.

2) Limiarização Global Ótima: Método de Otsu



2) Limiarização Global Ótima: Método de Otsu

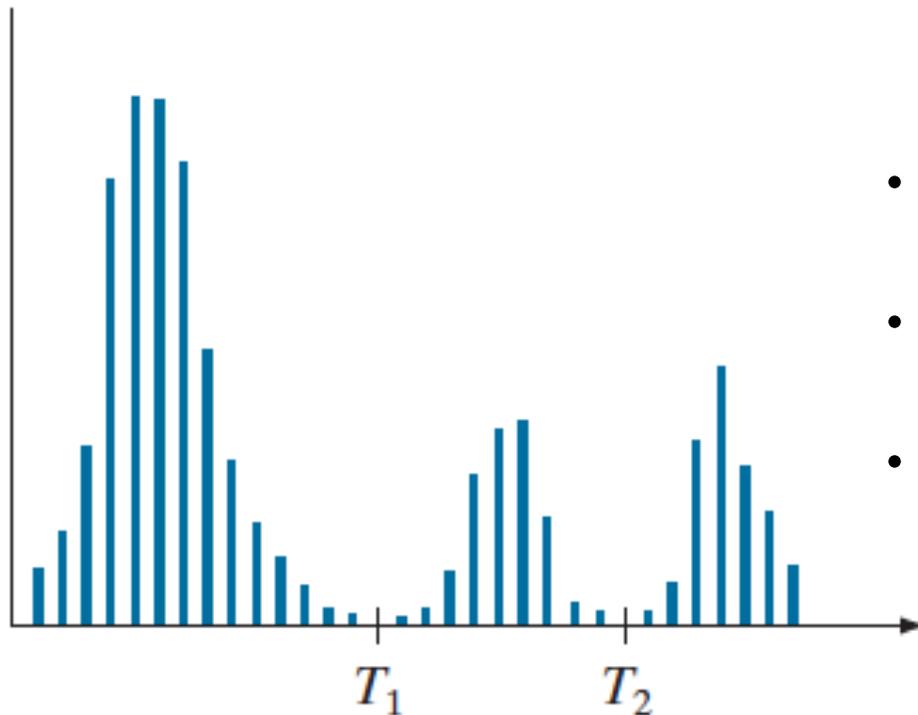


Limiarização
Simples
(pela Média)

Limiarização
Ótima
(pela Variância)

3) Limiarização Global Multinível: Método de Otsu

Histograma multi-modal



- Se $f(x, y) \leq T_1 \rightarrow$ o ponto (x, y) pertence ao fundo;
- Se $T_1 < f(x, y) \leq T_2 \rightarrow$ o ponto (x, y) pertence a uma classe de objetos;
- Se $f(x, y) > T_2 \rightarrow$ o ponto (x, y) pertence a outra classe.

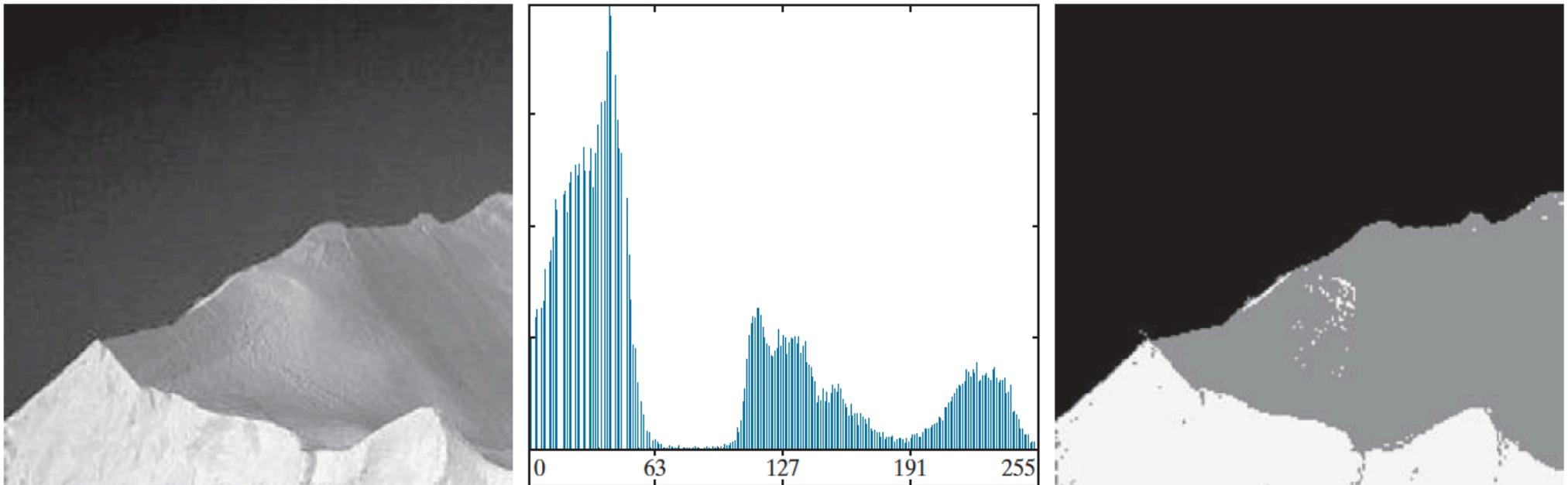
3) Limiarização Global Multinível: Método de Otsu

4. O método de Otsu escolhe T_1 e T_2 tal que maximize a variância entre as classes:

$$\sigma^2(T_1, T_2) = P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2 + P_3(m_3 - m_G)^2$$

- O algoritmo escolhe o primeiro valor de T_1 ($T_1 = 1$) e varia T_2 de 2 até $L-2$, calculando o valor da variância para todos os casos. Em seguida, incrementa o valor de T_1 e varia o valor de T_2 de T_1+1 até $L-2$. Repete-se esse procedimento até $T_1 = L-3$.
- O resultado desse processo é um arranjo 2D e os valores de T_1 e T_2 escolhidos são os que produzirem valor máximo nesse arranjo.

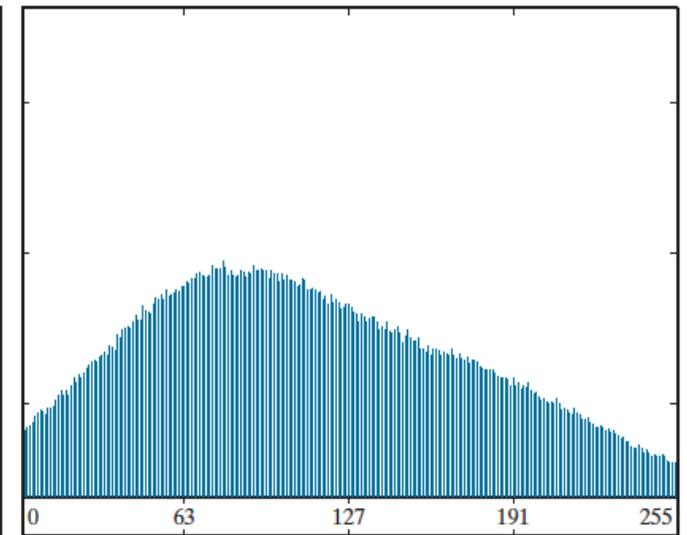
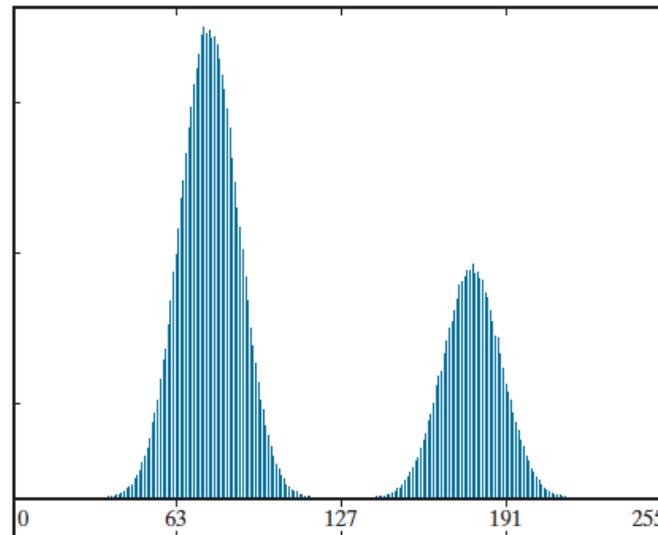
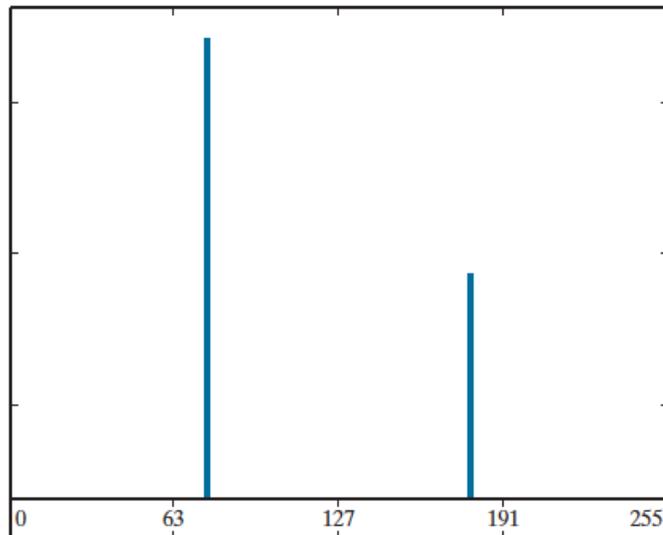
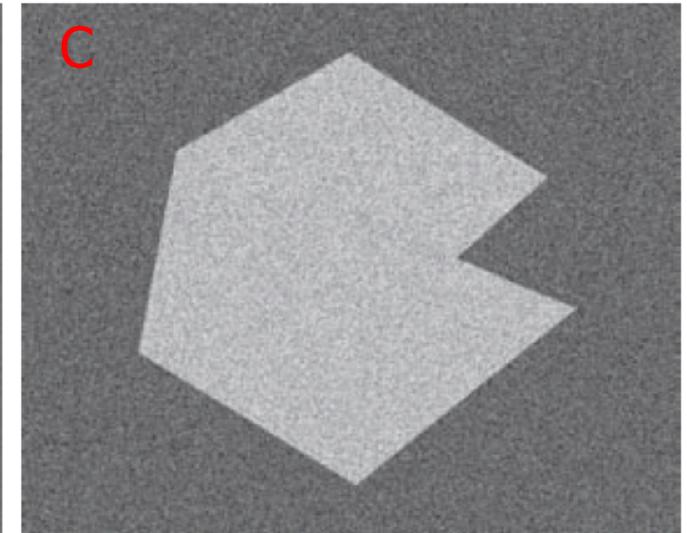
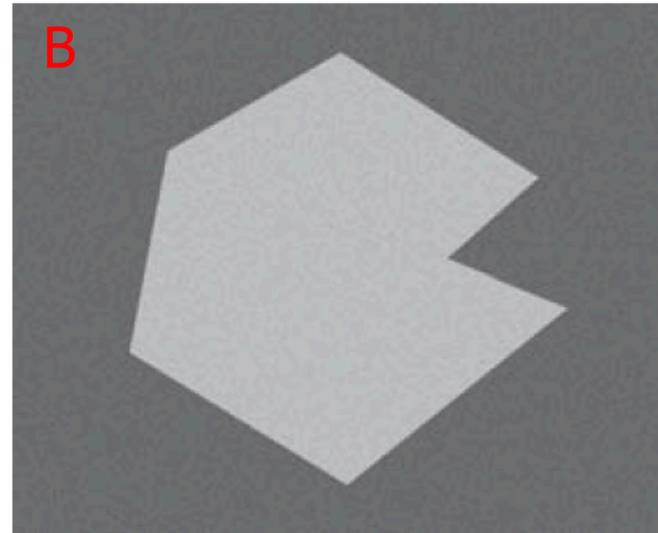
3) Limiarização Global Multinível: Método de Otsu



$$T_1 = 80$$

$$T_2 = 177$$

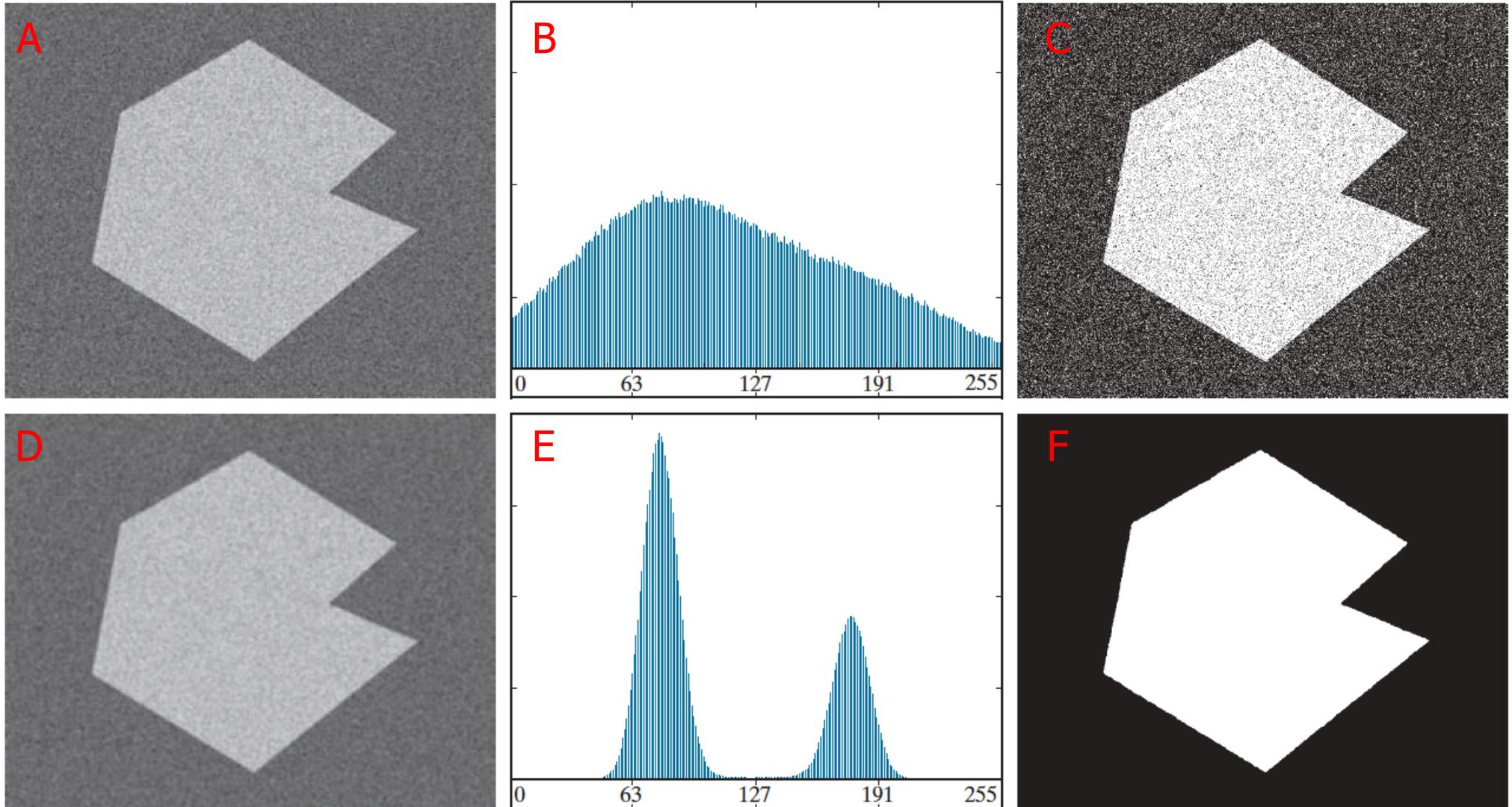
Influência do Ruído



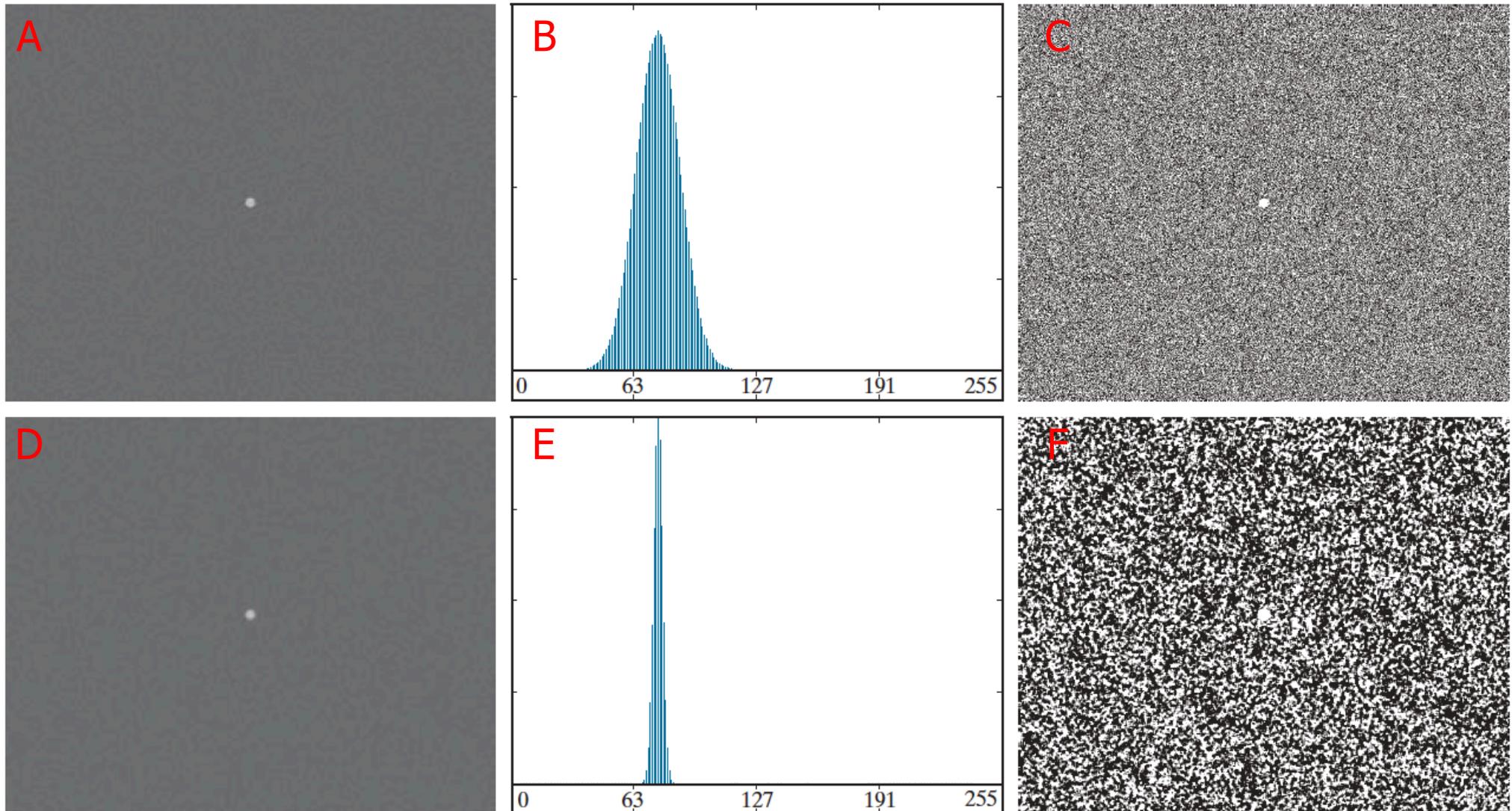
a) Sem ruído,

b) ruído de $\mu=0$ e $\sigma=10$

c) $\mu=0$ e $\sigma=50$



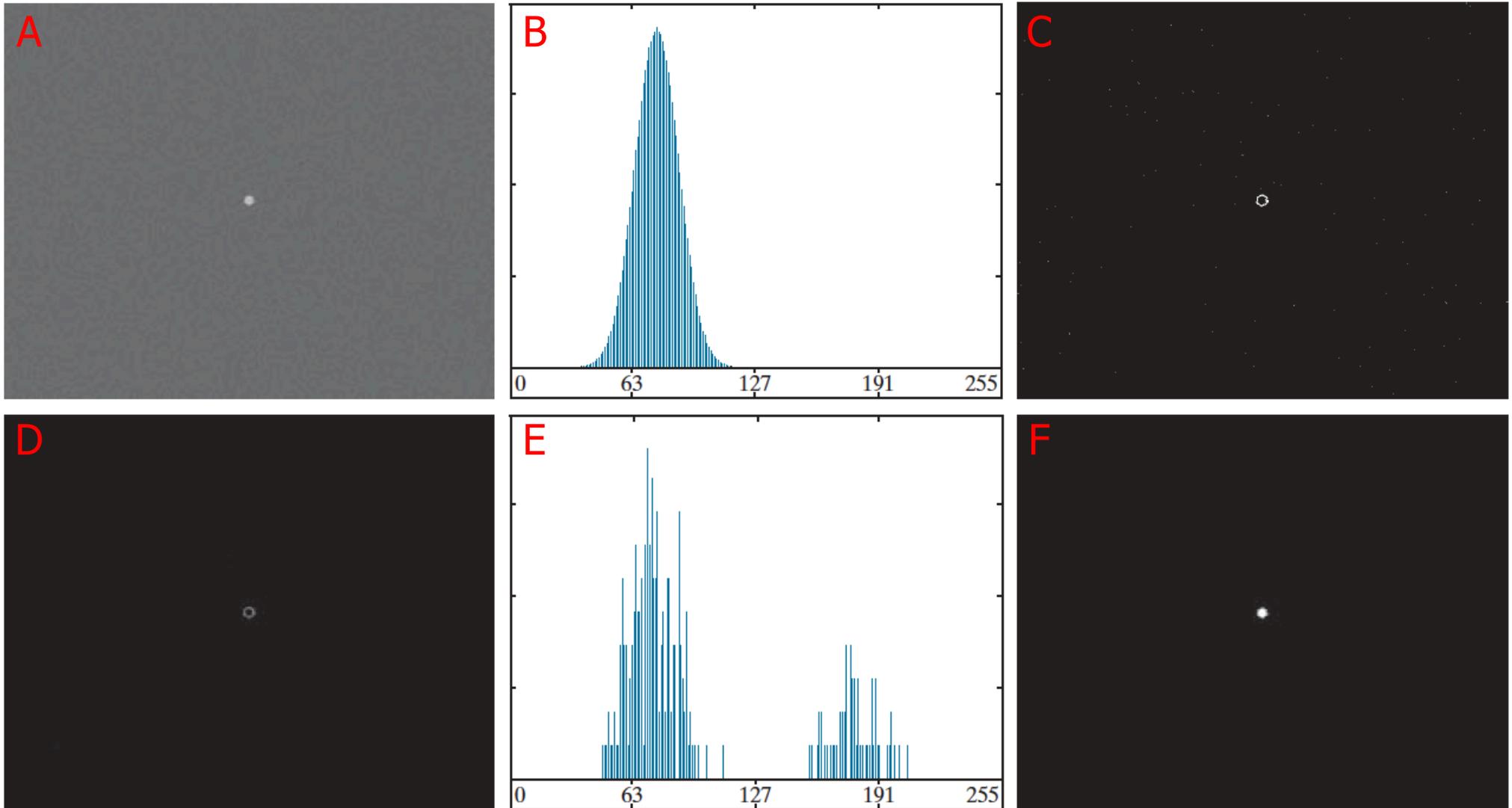
- a) Imagem ruidosa, b) seu histograma c) Otsu,
d) Imagem ruidosa suavizada $K = 5 \times 5$, e) seu histograma e f) Otsu



a) Imagem ruidosa, b) seu histograma c) Otsu, d) Imagem ruidosa suavizada $K = 5 \times 5$, e) seu histograma e f) Otsu. Falhou em ambos os casos.

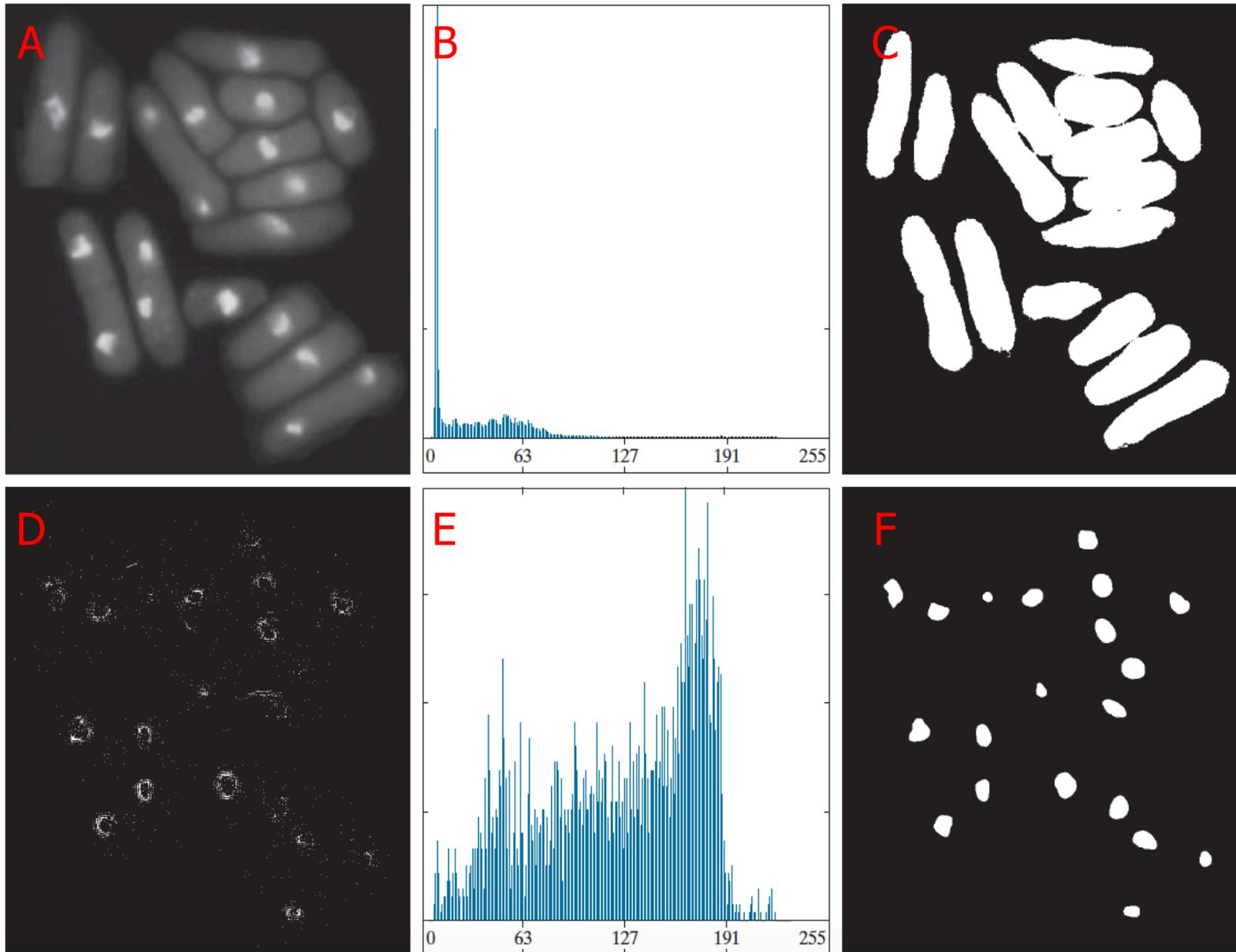
Solução: utilizar as informação das bordas

- Nos dois exemplos anteriores, o objeto não foi segmentado pois o seu tamanho faz com que seus pixels sejam estatisticamente insignificantes em relação ao espalhamento do ruído.
- Uma solução seria considerar no histograma apenas os **pixels próximos à bordas**, desprezando os pixels das regiões de baixa frequência.
- Pode ser feito utilizando um detector de bordas (geralmente Laplaciano) para encontrar as regiões de alta frequência na imagem e utilizar a imagem segmentada como uma “máscara” (Função lógica *AND*) com a imagem original para o cálculo do limiar.
- Isso faz com que as bordas sejam realçadas e o fundo eliminado do histograma, facilitando a separação das classes. Após o cálculo do limiar, **a binarização é realizada na imagem original.**



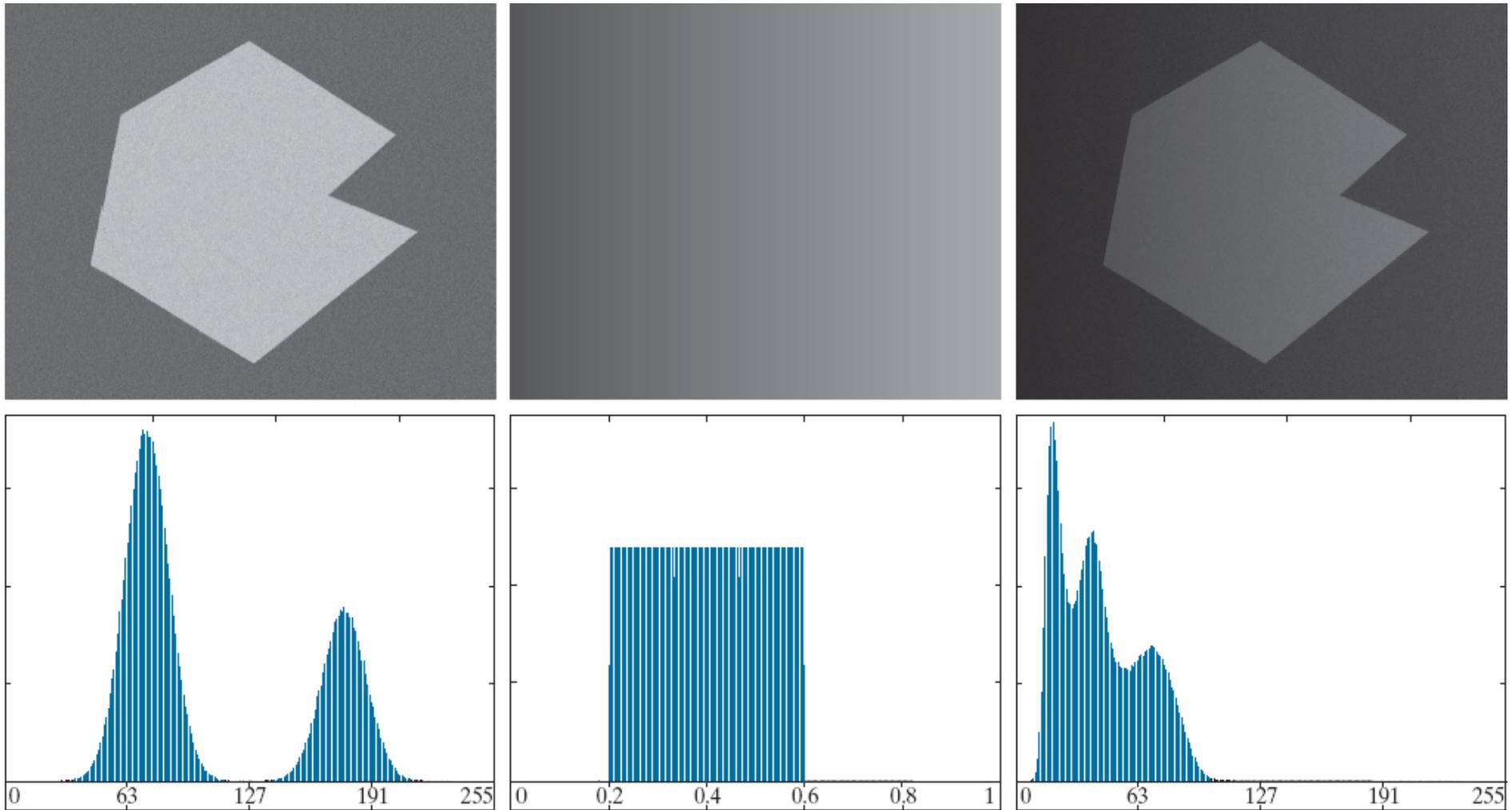
a) Imagem ruidosa, b) seu histograma c) Imagem gradiente binarizada, d) Imagem formada como produto de (a) e (c), e) histograma de (d) e f) Otsu (valor 134).

Usando o Laplaciano – 2º Ordem

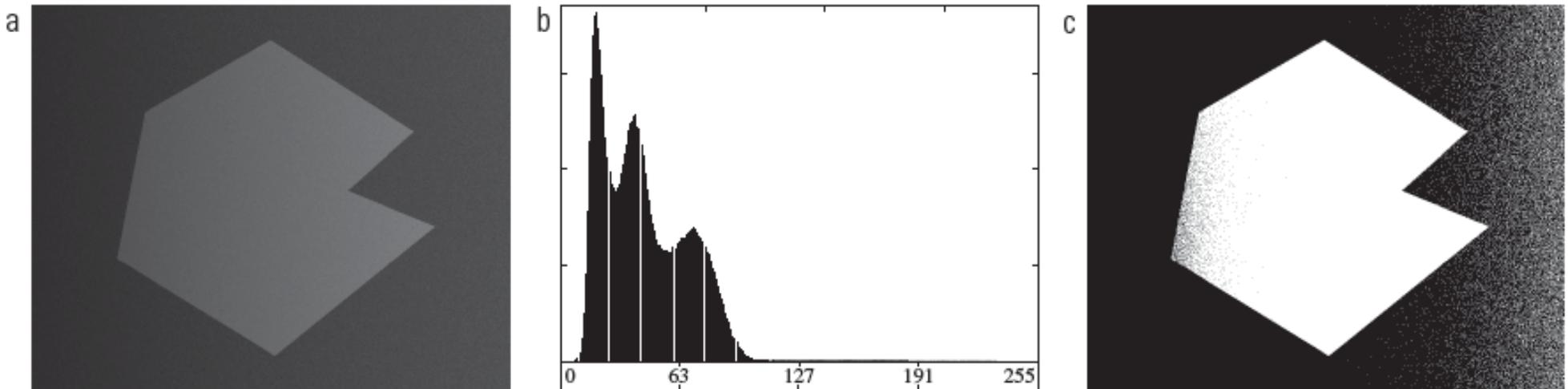


a) Imagem ruidosa, b) seu histograma c) Otsu em (b), d) Laplaciano binarizado, e) histograma do produto de (a) e (d), f) Otsu em (e).

Influência da Iluminação e/ou da Refletância



Influência da Iluminação e/ou da Refletância



Solução: utilizar limiarização local (não global)

- A Limiarização Global pode falhar quando a iluminação for não uniforme.
- Aplicar um *Threshold* Local é definir uma função $T(x,y)$ que varie o valor de T de acordo com as coordenadas (x,y) .

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

1. Subdivisão da imagem

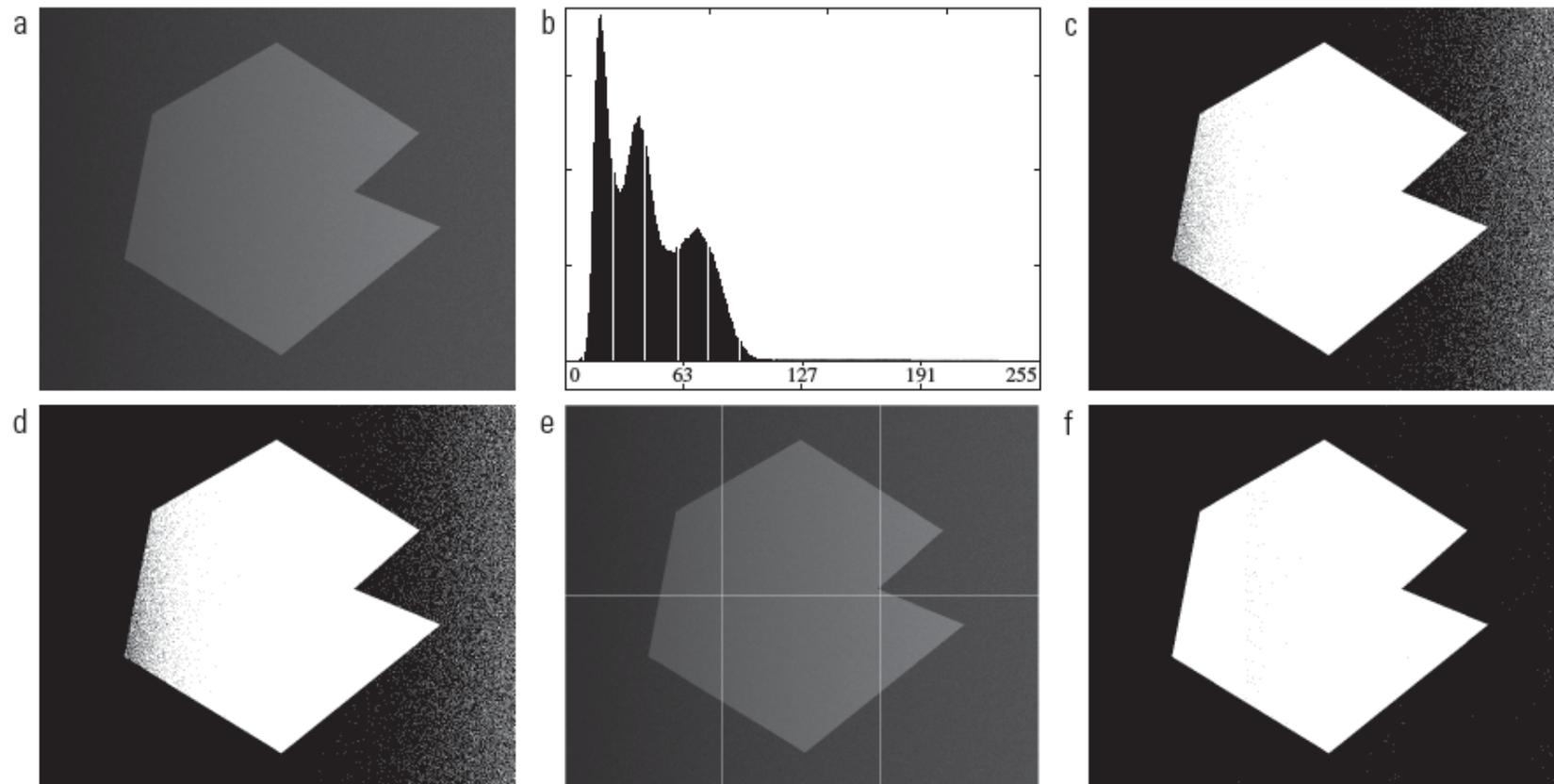


Figura 10.46 (a) Imagem ruidosa e sombreada. (b) Seu histograma. (c) Segmentação de (a) utilizando o algoritmo iterativo global da Seção 10.3.2. (d) Resultado obtido utilizando o método de Otsu. (e) Imagem subdividida em seis subimagens. (f) Resultado da aplicação do método de Otsu individualmente para cada subimagem.

1. Subdivisão da imagem

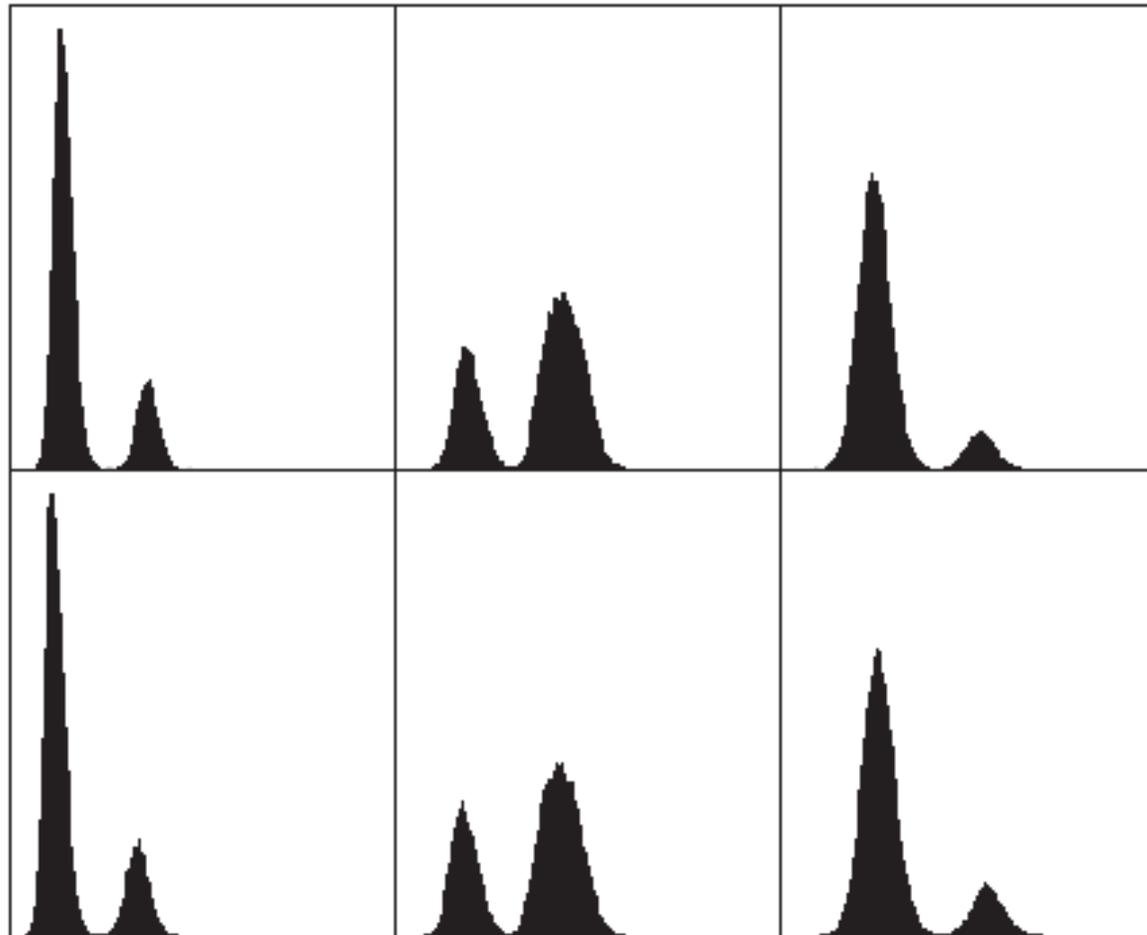


Figura 10.47 Histogramas das seis subimagens da Figura 10.46(e).

Crescimento de Região (*Region Growing*)

- Agrupamento de pixels ou grupo de pixels em regiões maiores.
- Os pixels a serem agrupados devem ter propriedades similares. (nível de cinza, textura, cor, etc...).
- Inicia-se com um conjunto de “sementes” em torno do qual as regiões crescem.

Crescimento de Região (*Region Growing*)

- Seja R a região completa de uma imagem e R_1, R_2, \dots, R_n sub-partições de R tal que:

a) $\bigcup_{i=1}^n R_i = R$ (todo pixel deve pertencer a uma região)

b) R_i é uma região conexa, $i = 1, 2, \dots, n$

c) $R_i \cap R_j = \emptyset$ para todo i e j , $i \neq j$

d) $P(R_i) = \text{VERDADEIRO}$ para $i = 1, 2, \dots, n$

$P(R_i) \rightarrow$ Propriedade definida para a região é verdadeira para todos os pixels da região
Ex: Intensidade igual.

Crescimento de Região (Region Growing)

Imagem:

	1	2	3	4	5
1	0	0	5	6	7
2	1	1	5	8	7
3	0	<u>1</u>	6	<u>7</u>	7
4	2	0	7	6	6
5	0	1	5	6	5

	R2				
R1	a	a	b	b	b
	a	a	b	b	b
	a	a	b	b	b
	a	a	b	b	b
	a	a	b	b	b
	a	a	b	b	b

T=3

Sementes \rightarrow (3, 2) e (3, 4)

Propriedade (P) $\rightarrow |I(x) - I(s)| < T$

$I(s) \rightarrow$ Intensidade da semente

T=8

a	a	a	a	a
a	a	a	a	a
a	a	a	a	a
a	a	a	a	a
a	a	a	a	a

Crescimento de Região (*Region Growing*)

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.

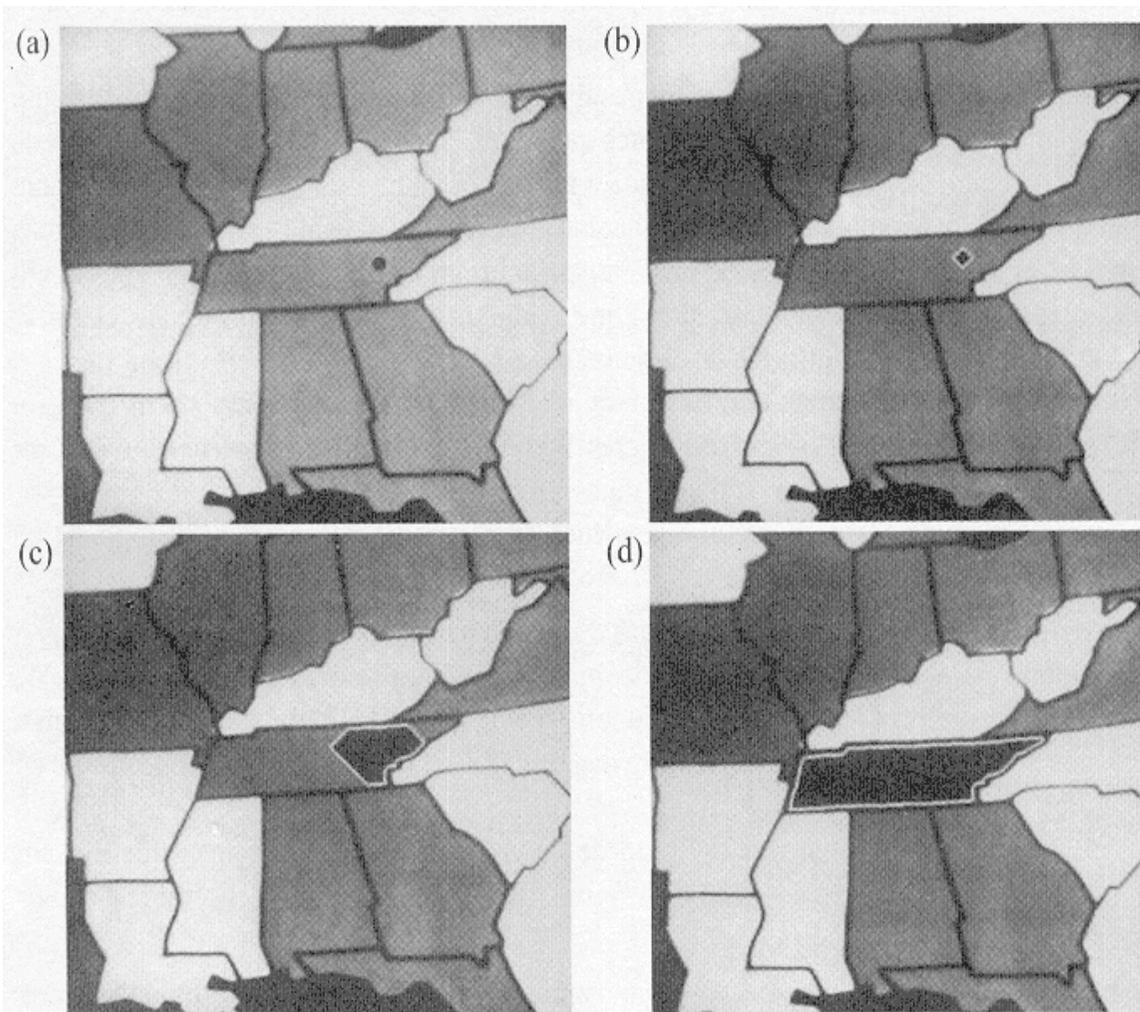
Crescimento de Região (*Region Growing*)

Problemas com a Técnica:

- 3) Utilização de conectividade e adjacência (4 ou 8);
- 4) Formulação de uma regra de parada: o crescimento de região deve parar quando não houver mais pixels satisfazendo os critérios de inclusão da região referida.

Crescimento de Região (*Region Growing*)

Exemplo:



a) Imagem $I(x, y)$ com semente $I(s)$

Critério:

$$|I(x, y) - I(s)| \leq 10\% (255 - \min)$$

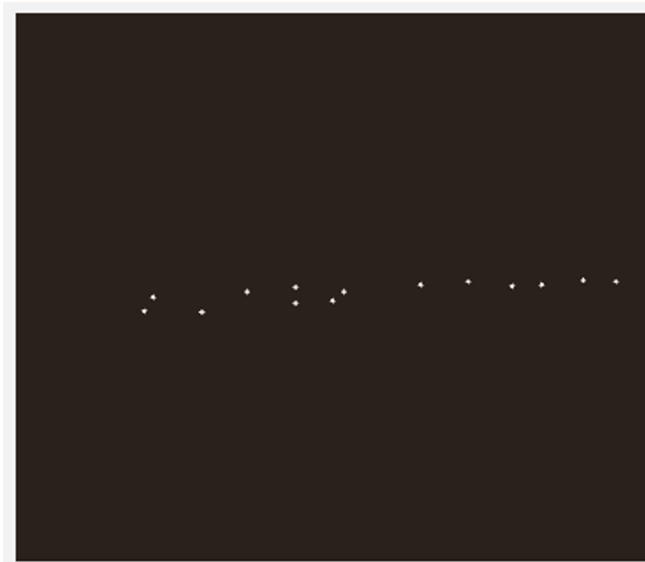
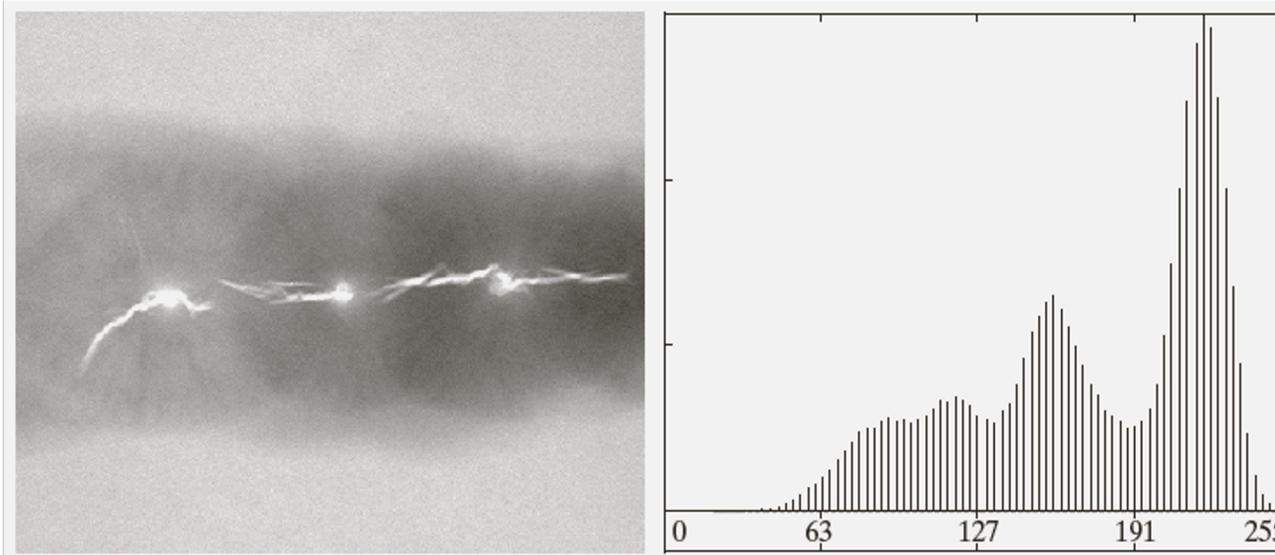
8 – conectada em cada pixel.

b) **Início do crescimento:** Pixels com a mesma distancia "city-block" da semente.

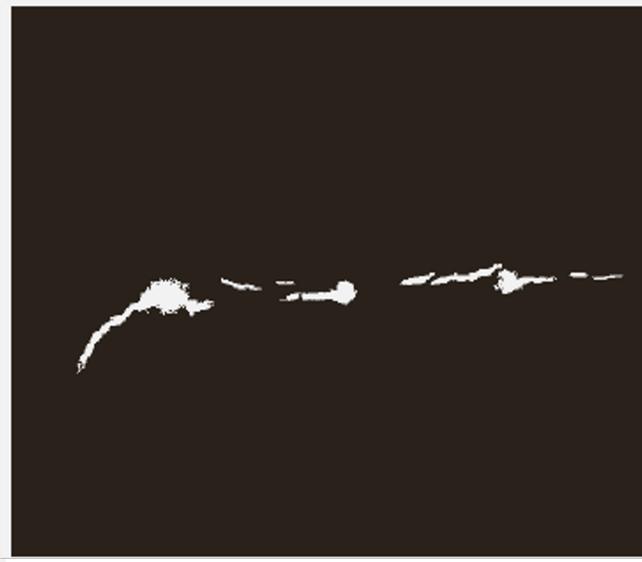
c) Estado intermediário de crescimento.

d) Região crescida completa: o processo para devido a borda de nível mais escuro que não atende ao critério de conectividade.

Exemplos

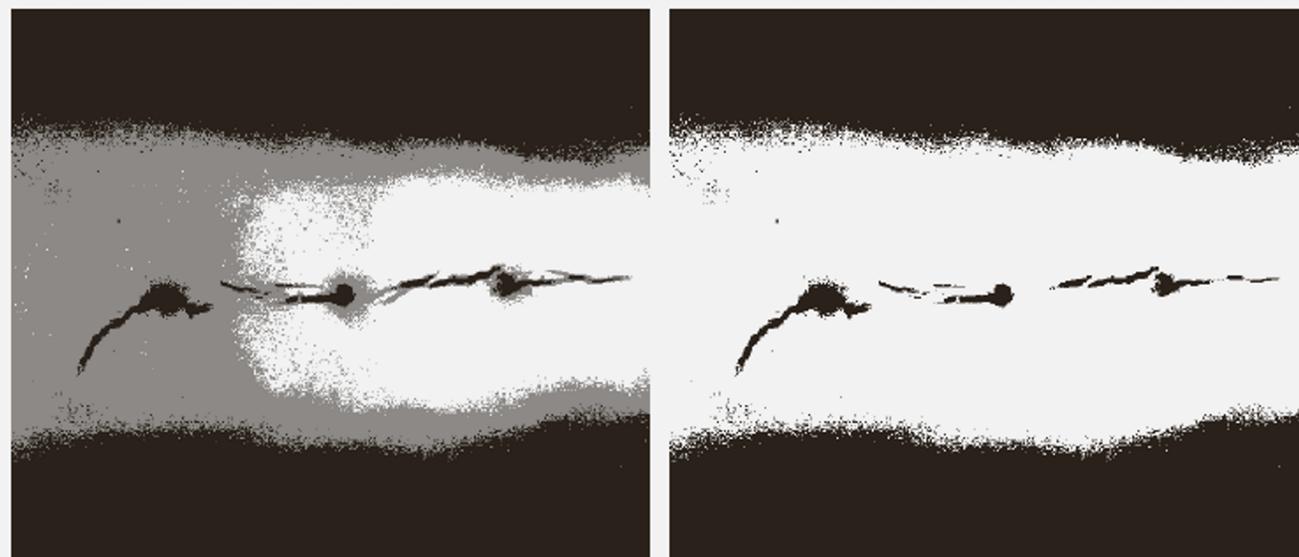
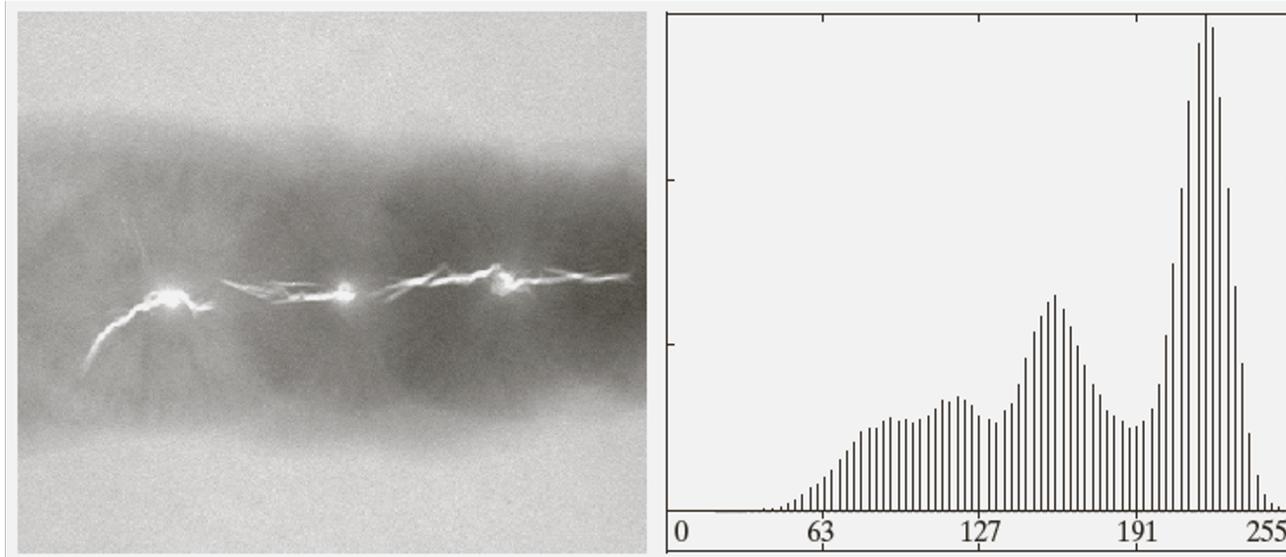


Sementes



Crescimento de região

Exemplos

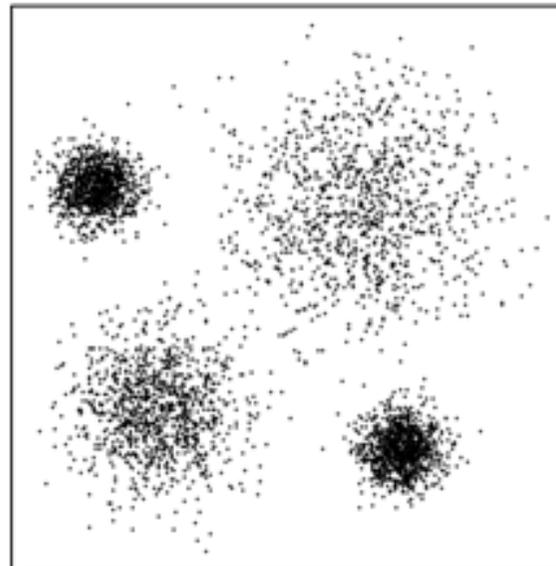


Limiar duplo (Otsu) – Limiar simples (Otsu)

K-means

- No agrupamento de *k-means*, cada observação (Pixel) é atribuída ao agrupamento com a média mais próxima (daí o nome do método).
- O algoritmo *k-means* é um procedimento iterativo que refina sucessivamente os resultados até que a convergência seja alcançada.

KMeans Iteration:



K-means

- O objetivo do agrupamento de k-means é particionar o conjunto Q de observações em k ($k \leq Q$) conjuntos de agrupamentos $C = \{C_1, C_2, \dots, C_k\}$, de modo que o seguinte critério de otimização seja satisfeito:

$$\arg \min_C \left(\sum_{i=1}^k \sum_{z \in C_i} \|z - \mathbf{m}_i\|^2 \right)$$

k = Número de conjuntos;

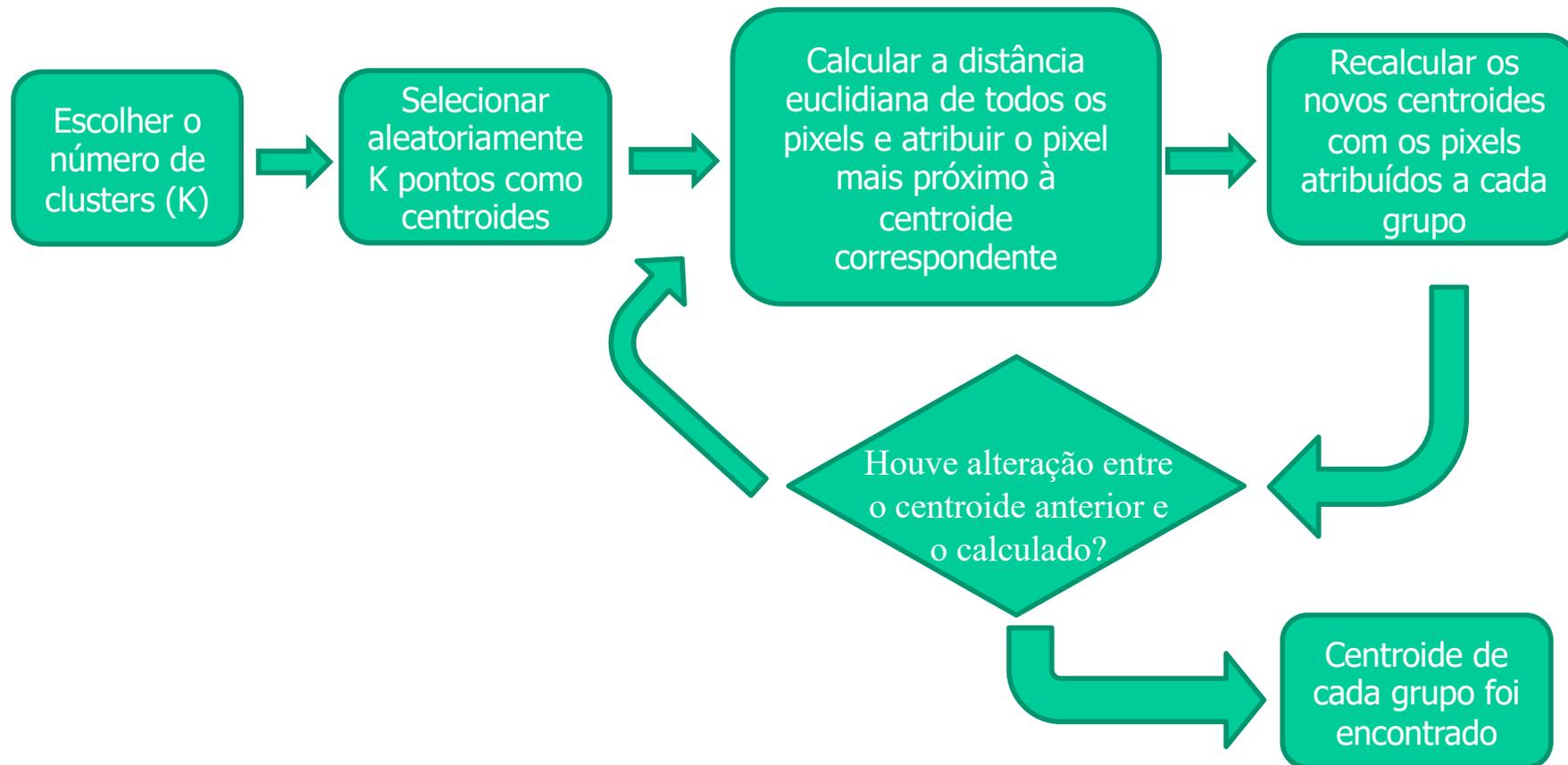
C_i = Conjunto i ;

z = Pixels de um conjunto C_i ;

\mathbf{m}_i = Média do conjunto C_i (centroide)

$\| \|^2$ = Distância Euclidiana.

K-means



Exemplo: K-Means (K=2)

- $V = \{3, 1, 2, 0, 2, 10, 12, 9, 8, 11\}$
- Início:
 - $M1 = 1$
 - $M2 = 3$
- Iteração
 - 1ª Iteração
 - $G1 = \{1, 2, 0, 2\}$
 - $G2 = \{3, 10, 12, 9, 8, 11\}$
 - $M1 = 1.25$
 - $M2 = 8.8$
 - 2ª Iteração
 - $G1 = \{3, 1, 2, 0, 2\}$
 - $G2 = \{10, 12, 9, 8, 11\}$
 - $M1 = 1.6$
 - $M2 = 10$

- Distâncias entre cada objeto e as Médias (M1 e M2)
- Matriz de Distâncias

1a. iteração

	3	1	2	0	2	10	12	9	8	11
M1=1	2	0	1	1	1	9	11	8	7	10
M2=3	0	2	1	3	1	7	9	6	5	8

$$M1 = (1 + 2 + 0 + 2) / 4 = \frac{5}{4} = 1.25$$

$$M2 = (3 + 10 + 12 + 9 + 8 + 11) / 6 = \frac{53}{6} = 8.8$$

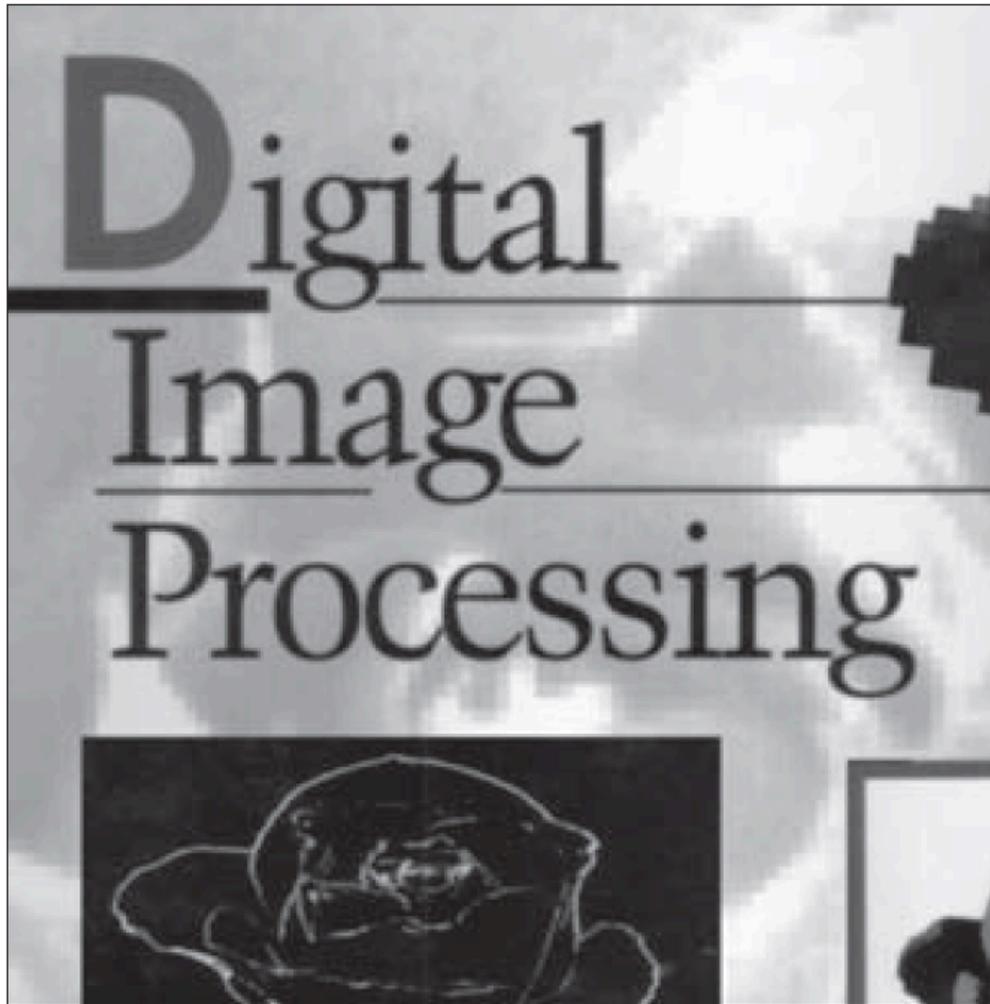
2a. iteração

	3	1	2	0	2	10	12	9	8	11
M1=1.25	1.75	0.25	0.75	1.25	0.75	8.75	10.75	7.75	6.75	9.75
M2=8.8	5.8	7.8	6.8	8.8	6.8	1.2	3.2	0.2	0.8	2.2

3a. iteração

	3	1	2	0	2	10	12	9	8	11
M1=1.6	1.4	0.6	0.4	1.6	0.4	8.4	10.4	7.4	6.4	9.4
M2=10	7	9	8	10	8	0	2	1	2	1

Segmentação por K-means (K=3)



Outras Técnicas para Segmentação de Imagens

- Segmentação por Textura;
- Segmentação por Cor (Imagens Coloridas);
- Segmentação utilizando Morfologia;
- Segmentação utilizando Contornos Ativos (*Snakes*);
- Segmentação utilizando Pirâmides de Resolução;
- Segmentação utilizando Superpixels (*Superpixels*);
- Segmentação utilizando Cortes de Grafos (*Graph Cuts*);
- Segmentação por watershed;
- Segmentação utilizando Redes Neurais Artificiais.

Mask Region-based convolutional neural networks (Mask-R-CNNs)

- Técnica que detecta com eficiência objetos em uma imagem enquanto gera simultaneamente uma máscara de segmentação de alta qualidade para cada instância.
- Utiliza Rede Neural Convolutiva.

Mask R-CNN

Kaiming He Georgia Gkioxari Piotr Dollár Ross Girshick

Facebook AI Research (FAIR)

Abstract

We present a conceptually simple, flexible, and general framework for object instance segmentation. Our approach efficiently detects objects in an image while simultaneously generating a high-quality segmentation mask for each instance. The method, called Mask R-CNN, extends Faster R-CNN by adding a branch for predicting an object mask in parallel with the existing branch for bounding box recognition. Mask R-CNN is simple to train and adds only a small overhead to Faster R-CNN, running at 5 fps. Moreover, Mask R-CNN is easy to generalize to other tasks, e.g., allowing us to estimate human poses in the same framework. We show top results in all three tracks of the COCO suite of challenges, including instance segmentation, bounding-box object detection, and person keypoint detection. Without bells and whistles, Mask R-CNN outperforms all existing, single-model entries on every task, including the COCO 2016 challenge winners. We hope our simple and effective approach will serve as a solid baseline and help ease future research in instance-level recognition. Code has been made available at: <https://github.com/facebookresearch/Detectron>.

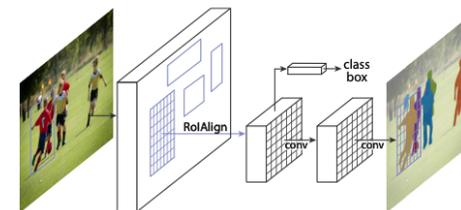
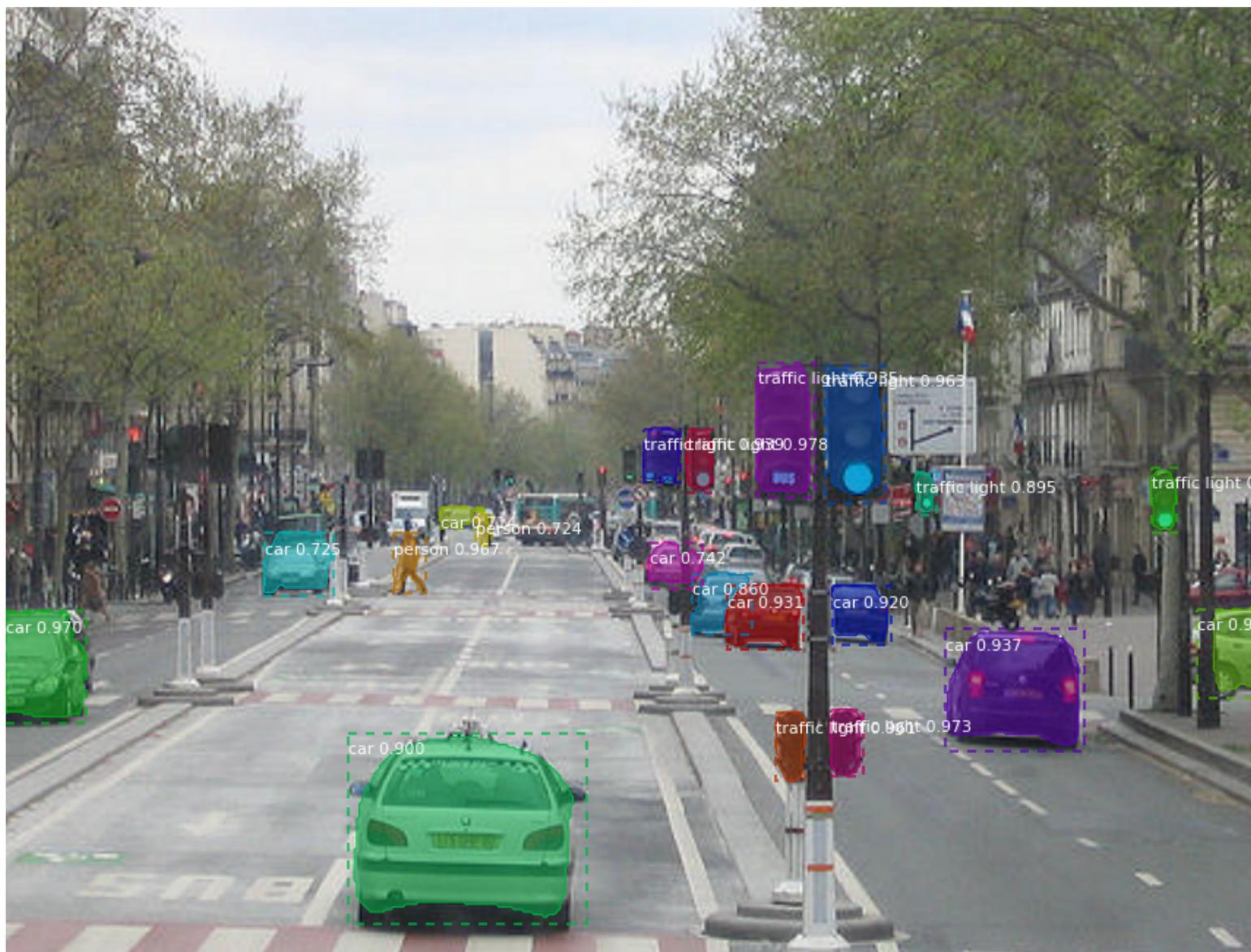


Figure 1. The Mask R-CNN framework for instance segmentation.

segmentation, where the goal is to classify each pixel into a fixed set of categories without differentiating object instances.¹ Given this, one might expect a complex method is required to achieve good results. However, we show that a surprisingly simple, flexible, and fast system can surpass prior state-of-the-art instance segmentation results.

Our method, called *Mask R-CNN*, extends Faster R-CNN [36] by adding a branch for predicting segmentation masks on each Region of Interest (RoI), in parallel with the existing branch for classification and bounding box regression (Figure 1). The mask branch is a small FCN applied to each RoI, predicting a segmentation mask in a pixel-to-

Mask-RCNNs



Mask-R-CNNs



https://github.com/matterport/Mask_RCNN

Mask-R-CNNs



https://github.com/matterport/Mask_RCNN

Mask-R-CNNs



https://github.com/matterport/Mask_RCNN



FIM