



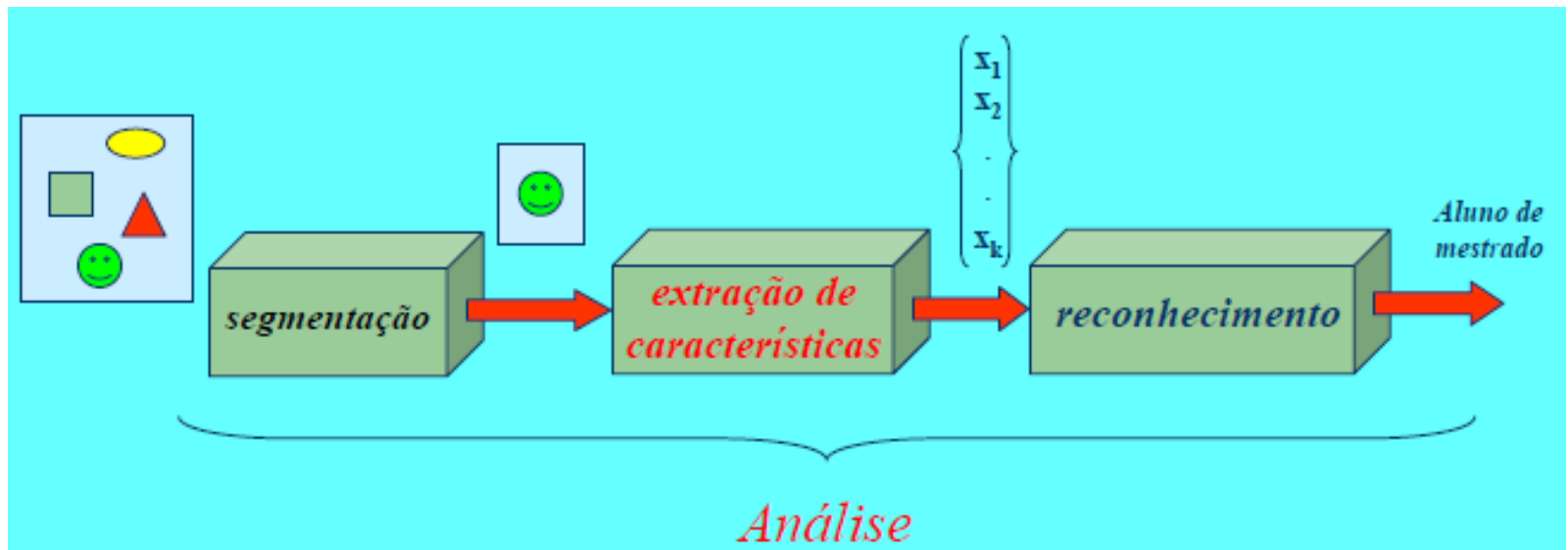
Segmentação

Guillermo Cámara-Chávez

O problema da segmentação



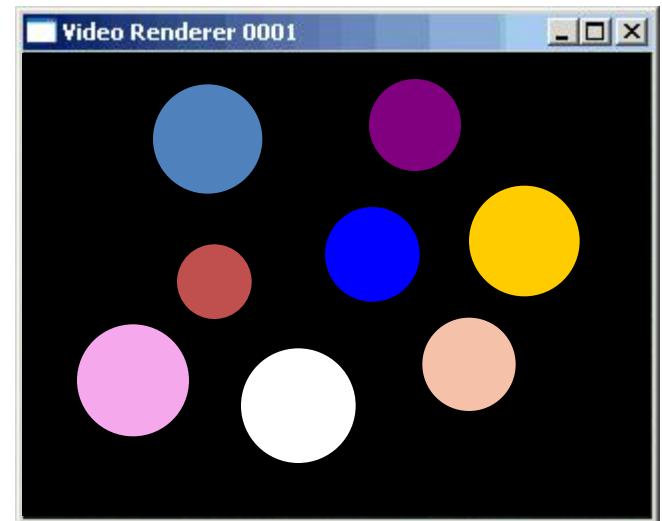
- Segmentação é uma tarefa básica no processo de análise de imagens



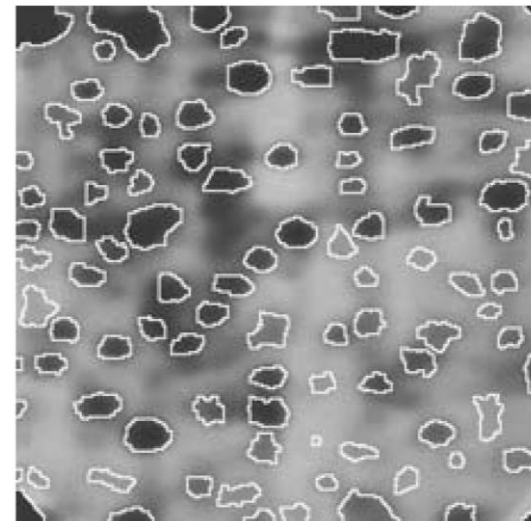
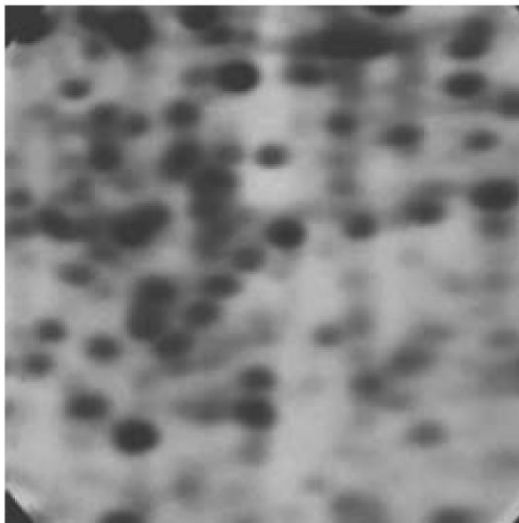
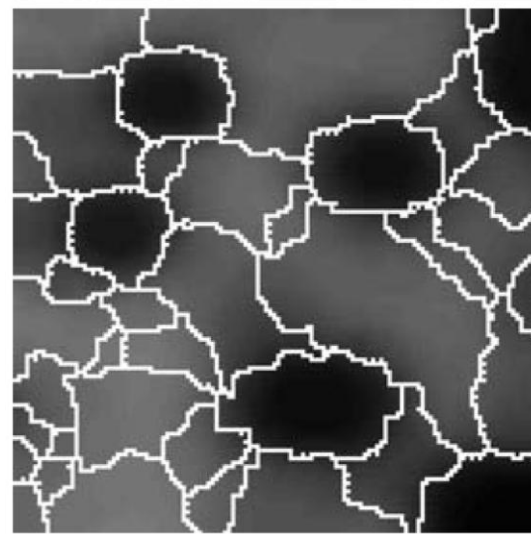
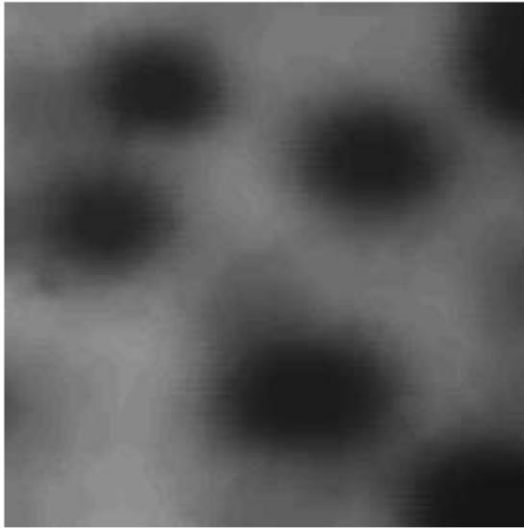
O problema da segmentação



- Métodos de segmentação são utilizados para **particionar** os pixels de uma imagem em **grupos**, geralmente **relacionados a objetos** em uma imagem.
- Tipicamente é o primeiro passo para diversas aplicações



O problema da segmentação



O problema da segmentação



- É uma das tarefas mais difíceis em processamento de imagens
- Com o **objetivo** de **aumentar a confiabilidade** e o resultado da segmentação, **deve-se fazer uso** de todo e qualquer **conhecimento prévio** sobre o problema

O problema da segmentação



- A segmentação determina o eventual sucesso ou fracasso da análise.
- Os algoritmos de segmentação são geralmente baseados na busca pelas **descontinuidades** ou pelas **similaridades** dos níveis de cinza.

Segmentação



Algoritmos de Segmentação

Mudanças bruscas
nos níveis de cinza

Semelhança nos
níveis de cinza

Descontinuidades

Similaridades

detecção de:

- pontos isolados
- linhas e
- Bordas na imagem

aplicação de:

- limiarização
- crescimento de regiões

Segmentação por Detecção de Descontinuidades



- Detecção de Pontos

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

$$|R| > T$$

- Feita através do uso de um filtro passa altas e um limiar de aceitação

Segmentação por Detecção de Descontinuidades



- Detecção de Linhas

-1	-1	-1
2	2	2
-1	-1	-1

horizontal

-1	-1	2
-1	2	-1
2	-1	-1

+45 graus

-1	2	-1
-1	2	-1
-1	2	-1

vertical

2	-1	-1
-1	2	-1
-1	-1	2

-45 graus

- Feita através do uso de um filtro passa altas direcionais

Segmentação por Detecção de Descontinuidades



- Detecção de Bordas:

“Uma borda é o limite entre duas regiões com propriedades relativamente distintas de nível de cinza”

Segmentação por Detecção de Descontinuidades



- É **assumido** que as **regiões** em questão são suficientemente **homogêneas** e que a **transição** entre as regiões podem ser **determinadas** com base apenas na **descontinuidade** dos níveis de cinza.

Segmentação por Detecção de Descontinuidades



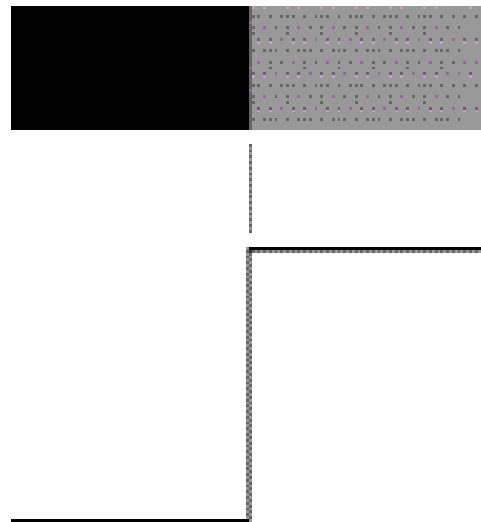
- A **idéia predominante** nas técnicas de detecção de bordas é a computação de um **operador diferencial**
- É importante ter em mente que **bordas em imagens digitais** são, geralmente, **levemente borradas** devido ao processo de **amostragem**

Segmentação por Detecção de Descontinuidades

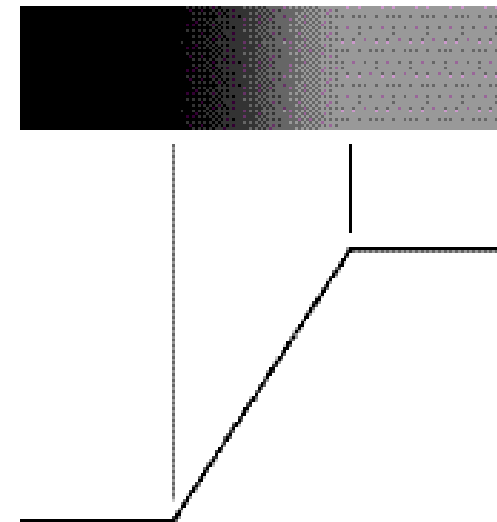


- A existência de imperfeições no processo de aquisição de imagem faz com que as bordas sejam “rampas”

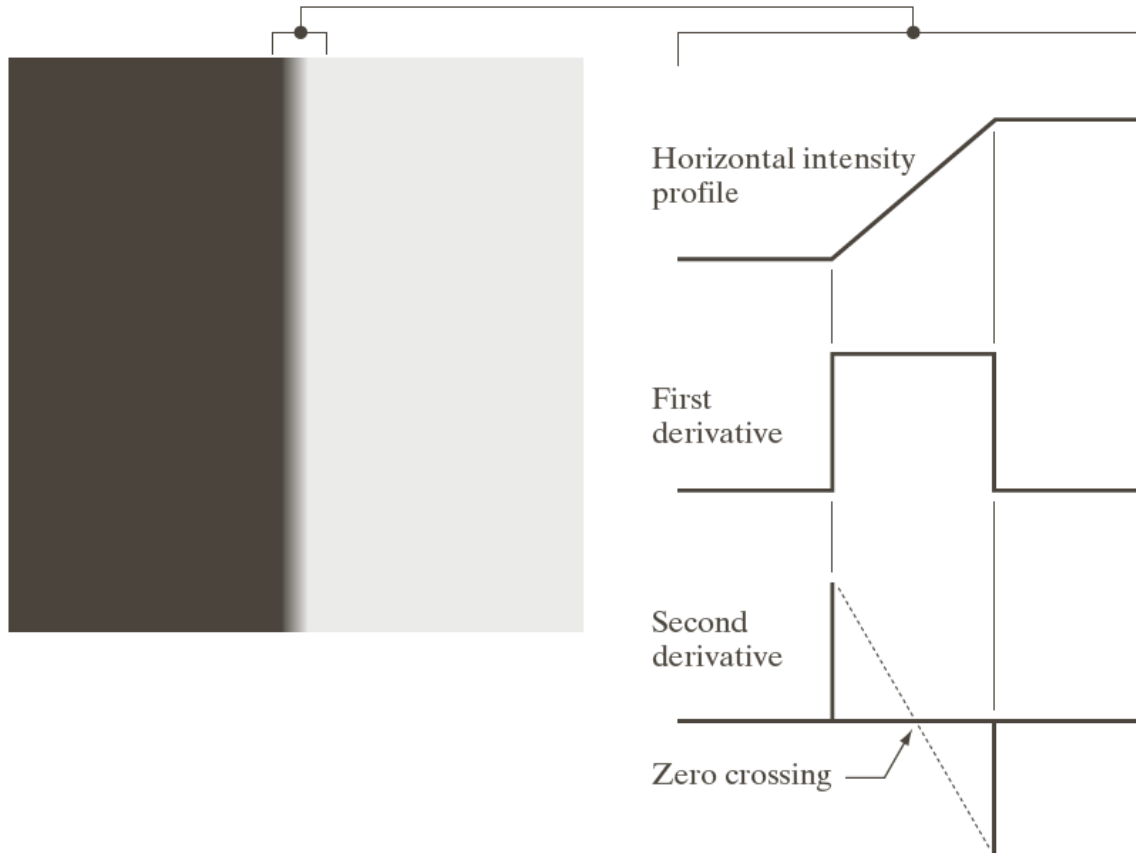
Modelos de uma borda digital ideal



Modelos de rampa em uma borda digital ideal



Operadores Diferenciais



a b

FIGURE 10.10

(a) Two regions of constant intensity separated by an ideal vertical ramp edge.
(b) Detail near the edge, showing a horizontal intensity profile, together with its first and second derivatives.

Operadores Diferenciais de 1ª ordem

1ª derivada - Gradiente



$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

$$|\nabla f| = \text{mag}(\nabla f) = \left[G_x^2 + G_y^2 \right]^{1/2} \approx |G_x| + |G_y|$$

$$\theta(x, y) = \tan^{-1} \left(\frac{G_y}{G_x} \right)$$

Ângulo de direção do vetor gradiente em relação ao eixo x

Operadores Diferenciais de 1ª ordem

1ª derivada - Gradiente



- Operador de Sobel

X

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

Y

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

$$g(x, y) = (X^2 + Y^2)^{1/2}$$

- O operador de Sobel tem um efeito secundário de suavização que é atrativo visto que a derivação aumenta o ruído
- Composição de: $[-1 \ 0 \ 1]$ diferenciação + $[1 \ 2 \ 1]$ suavização

Operadores Diferenciais de 1ª ordem

1ª derivada - Gradiente



```
img = imread('lenna.png');  
mask_x = fspecial('sobel');  
Gx = conv2(double(img), mask_x, 'same');  
Gy = conv2(double(img), mask_x', 'same');  
M = sqrt(Gx.^2 + Gy.^2);  
figure; imshow(M);  
figure; imshow(im2bw(M/max(M(:))), .2);
```

Operadores Diferenciais de 1ª ordem

1ª derivada - Gradiente



Magnitude



$M > 0.2$

Operadores Diferenciais de 1ª ordem

1ª derivada - Gradiente



- Operador de Prewitt

$$X$$

-1	-1	-1
0	0	0
1	1	1

$$Y$$

1	0	-1
1	0	-1
1	0	-1

$$g(x, y) = (X^2 + Y^2)^{1/2}$$

Operadores Diferenciais de 1ª ordem

1ª derivada - Gradiente



- Operador de Roberts

$$X \begin{array}{|c|c|} \hline 1 & 0 \\ \hline 0 & -1 \\ \hline \end{array}$$

$$Y \begin{array}{|c|c|} \hline 0 & 1 \\ \hline -1 & 0 \\ \hline \end{array}$$

$$g(x, y) = (X^2 + Y^2)^{1/2}$$

Operadores Diferenciais de 2ª ordem

2ª derivada



- Operador Laplaciano

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

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

Operadores Diferenciais de 2ª ordem

2ª derivada



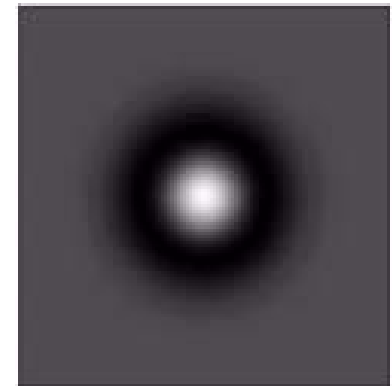
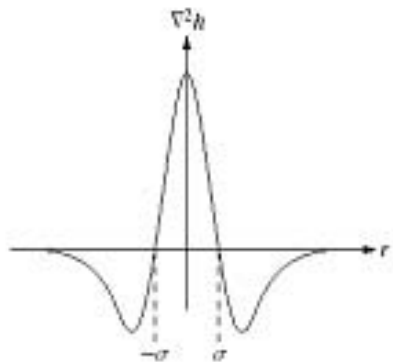
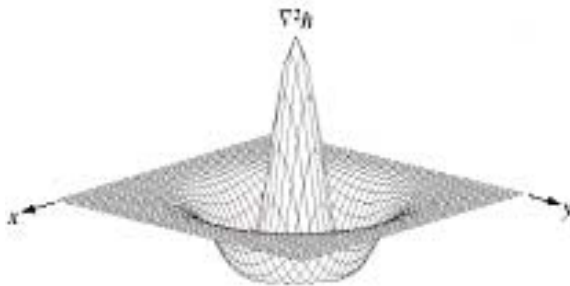
- Operadores de gradiente: funcionam bem em casos envolvendo imagens com transições de contraste e ruído relativamente baixo
- Operadores Laplaciano (cruzamento de zero): oferecem uma alternativa nos casos em que as bordas forem borradas ou quando o ruído for elevado

Operadores Diferenciais de 2ª ordem

2ª derivada



- Operador Laplaciano



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

Thresholding (limiarização)



- Thresholding é usualmente o primeiro passo em qualquer processo de segmentação
- Thresholding através de um único valor pode ser interpretado matematicamente como:

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

Exemplo de *Thresholding*



- Imagine um robô jogador de poker que necessita interpretar visualmente as cartas que possui:



Imagem original

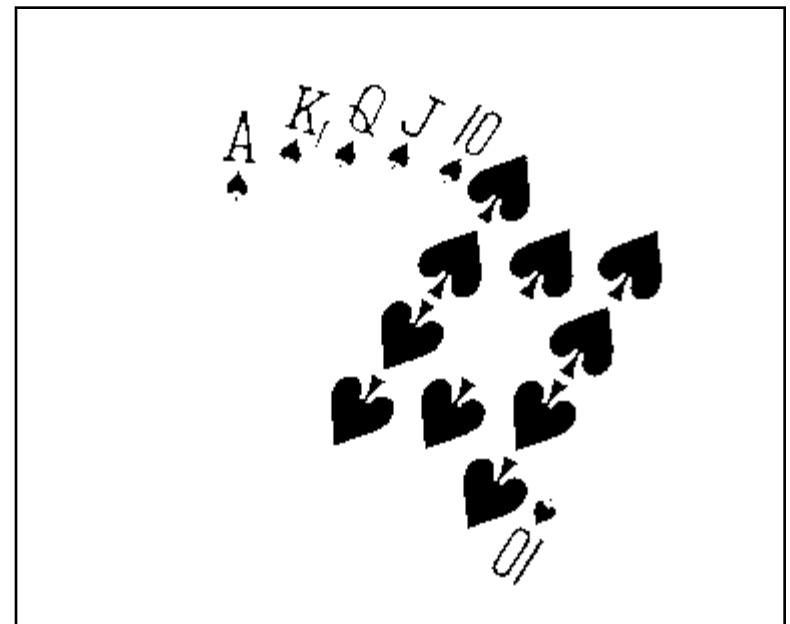
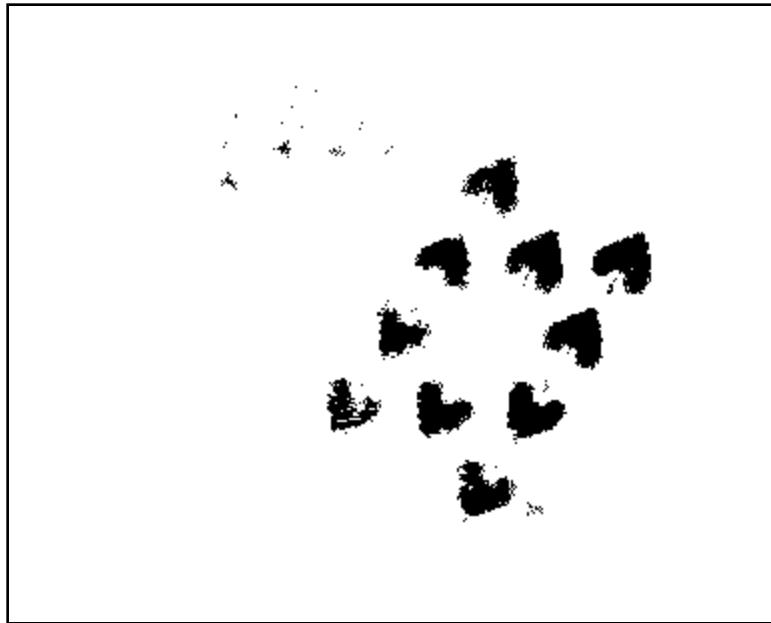


Imagem limiarizada

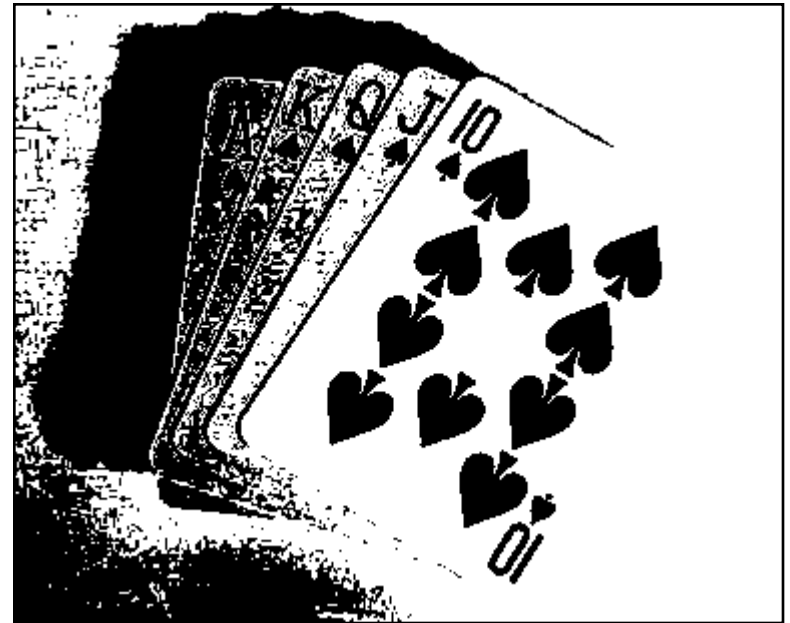
Cuidados na escolha do valor



- Se o valor a ser utilizado for incorreto os resultados serão insatisfatórios:



Threshold baixo demais



Threshold alto demais

Thresholding Global



- Baseado no histograma de imagem
- Particiona o histograma da imagem utilizando um único valor de *threshold*
- O sucesso desta técnica depende de quão bom o histograma de uma imagem pode ser particionado

Algoritmo para utilização do Threshold Global



- O valor do threshold, T , é calculado da seguinte forma:
 1. Selecione uma estimativa inicial de T (tipicamente o tom de cinza médio da imagem)
 2. Segmentando a imagem usando T produzirá dois grupos de pixels: G_1 com os pixels com valor $>T$ e G_2 com os pixels com valor $\leq T$
 3. Calcule a média dos tons de cinza dos pixels em G_1 para gerar μ_1 e G_2 para gerar μ_2

Algoritmo para utilização do Threshold Global

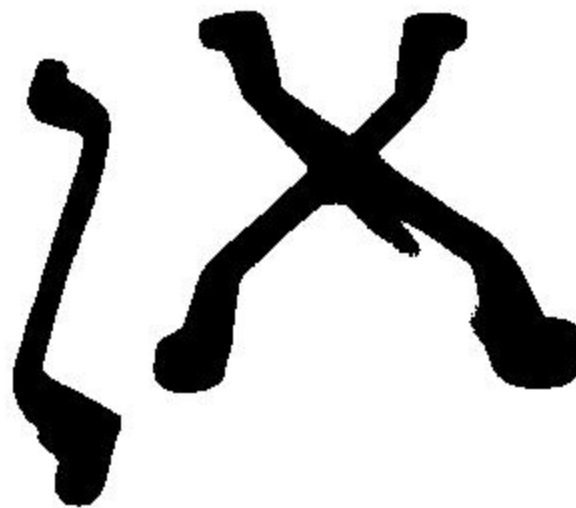
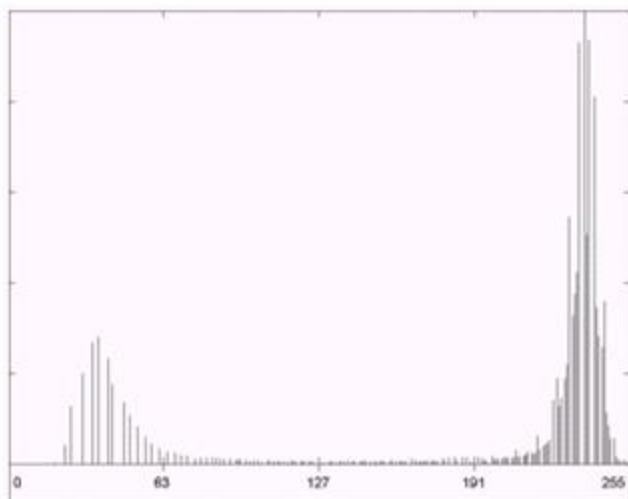
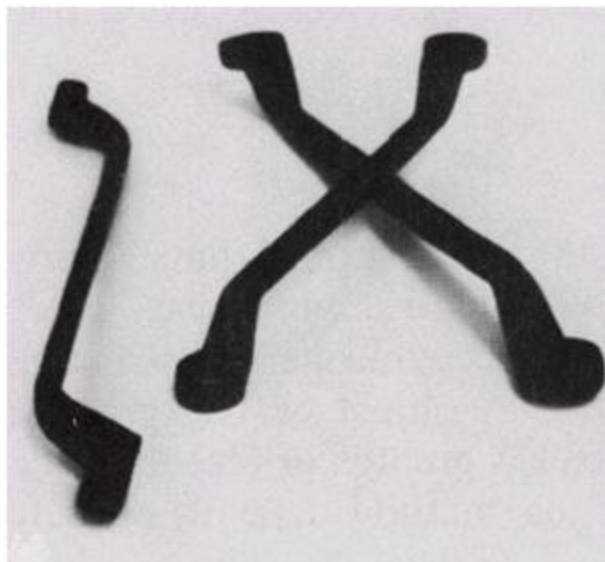


4. Calcule o novo Threshold

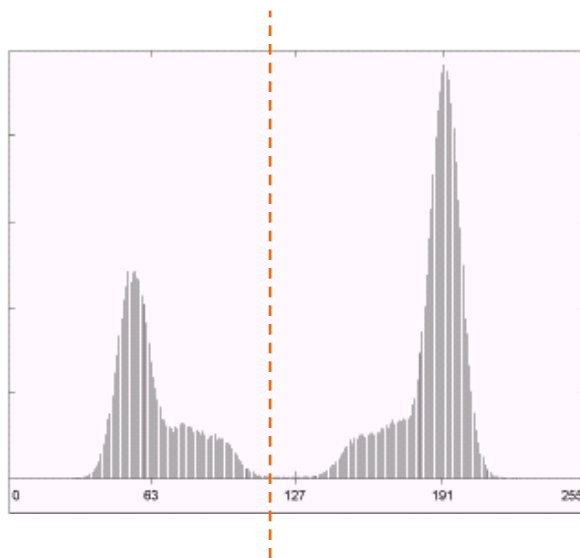
$$T = \frac{\mu_1 + \mu_2}{2}$$

5. Repita os passos 2 – 4 até que a diferença entre T nas próximas iterações seja menor do que um valor T_∞ pré-definido.
- Este algoritmo funciona bem para encontrar valores de threshold quando o histograma é adequado.

Thresholding Exemplo 1



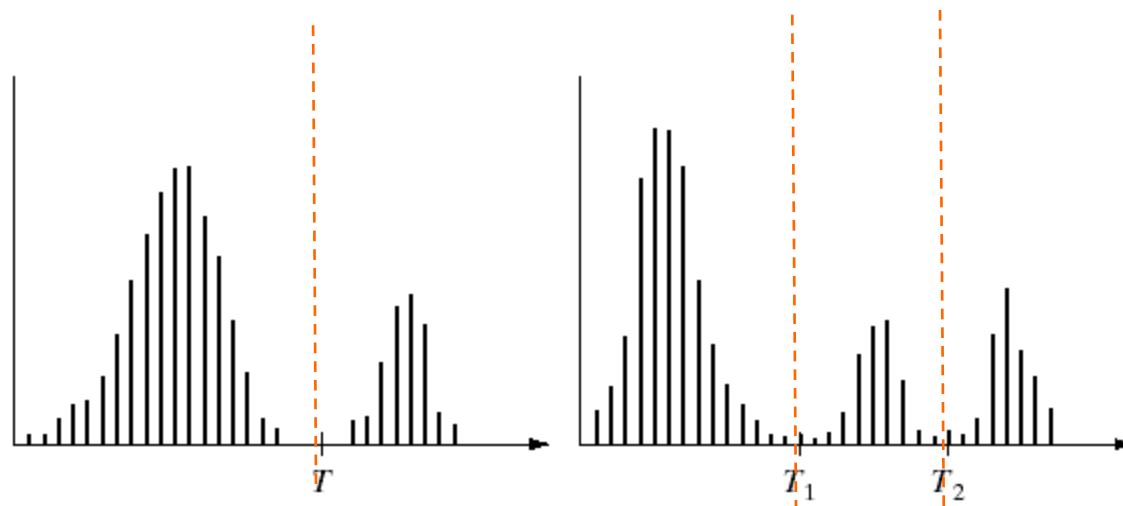
Thresholding Exemplo 2



Problemas com Threshold simples



- Thresholding com valor único só funciona com imagens bimodais
- Imagens com outro tipo de histograma necessitam de mais do que um único valor



Problemas com Threshold simples

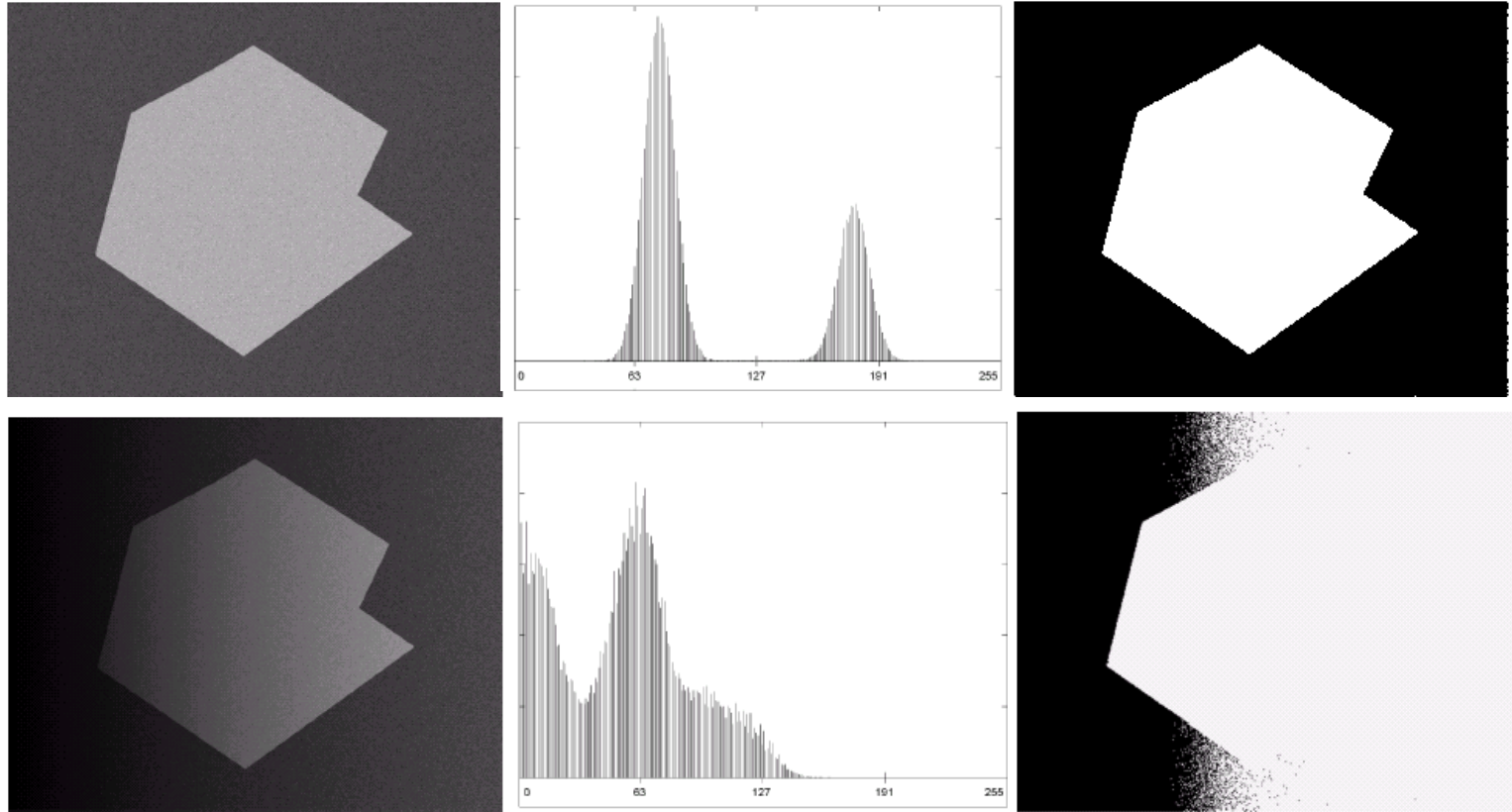


- Digamos que queremos isolar o conteúdo das garrafas



- Pense em como o histograma desta imagem seria
- O que aconteceria se utilizarmos apenas um valor?

Threshold único e iluminação



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

Thresholding Adaptativo

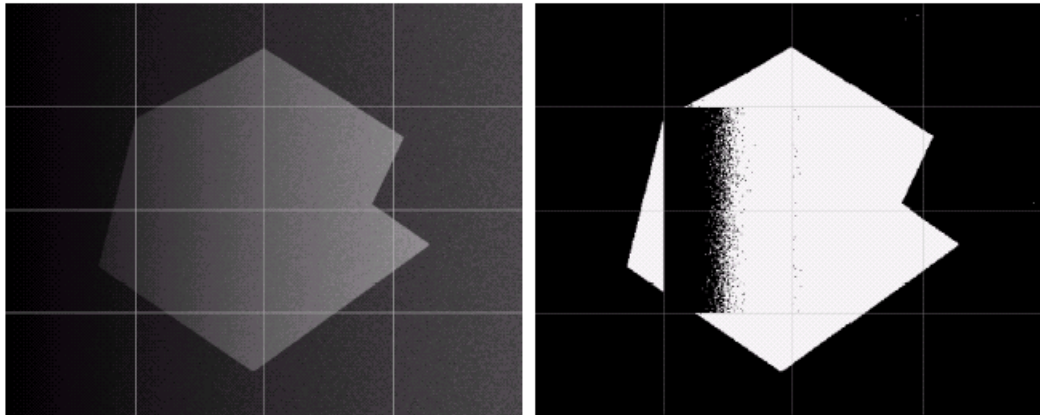


- Uma abordagem para tratar situações onde um **único valor não funciona** é **dividir** a imagem em **sub-imagens** e aplicar ***thresholds* individuais** em cada sub-imagem gerada
- Uma vez sendo o *threshold* para cada pixel dependente de sua localização em sua sub-imagem esta técnica é considerada adaptativa

Exemplo de Thresholding Adaptativo



- A imagem abaixo mostra um exemplo do uso de *threshold* adaptativo na imagem mostrada anteriormente



- Como pode ser observado, o sucesso é parcial
- Pode-se subdividir as áreas problemáticas para obtermos melhores resultados

Thresholding: Método de Otsu



- A operação de limiarização é considerada como sendo o particionamento dos pixels de uma imagem com L níveis de cinza em duas classes C_0 e C_1 .
- A partição se dará no nível de cinza t .
- Desta forma temos:
 $C_0 = \{0, 1, 2, \dots, t\}$ e $C_1 = \{t+1, t+2, \dots, L\}$

Thresholding: Método de Otsu



- Este método baseia-se no histograma normalizado com uma função de densidade de probabilidade

$$p_i = n_i / N, p_i \geq 0, \sum_{i=1}^L p_i = 1$$

$$N = n_1 + n_2 + n_3 + \dots + n_L$$

Thresholding: Método de Otsu



- O método de Otsu escolhe o threshold de valor k que maximiza a variância entre classes σ_B^2 que é definido como

$$\sigma_B^2(k) = \frac{(\mu_T \omega(k) - \mu(k))^2}{\omega(k)(1 - \omega(k))}$$

$$\omega(k) = \sum_{i=1}^k p_i$$

$$\mu(k) = \sum_{i=1}^k ip_i$$

$$\mu_T = \mu(L) = \sum_{i=1}^L ip_i$$

Thresholding: Método de Otsu



- O limiar ótimo k^* está determinado por:

$$\sigma_B^2(k^*) = \max_{1 \leq k < L} \sigma_B^2(k)$$

- Implementado em matlab através da função *graythresh()*

Thresholding: Método de Otsu



```
function T = Otsu(img)
h = imhist(img, 256);
p = h / sum(h);
omega = cumsum(p);
mu = cumsum(p .* [1:256]');
muT = mu(end);
sigma_b = (muT .* omega - mu).^2 ./ (omega .* (1 -
    omega));
m = max(sigma_b);
T = mean(find(sigma_b == m))
```

Segmentação por Crescimento de Regiões



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

Segmentação Baseada em Regiões



Premissas:

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

Segmentação Baseada em Regiões



Premissas:

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

Segmentação Baseada em Regiões



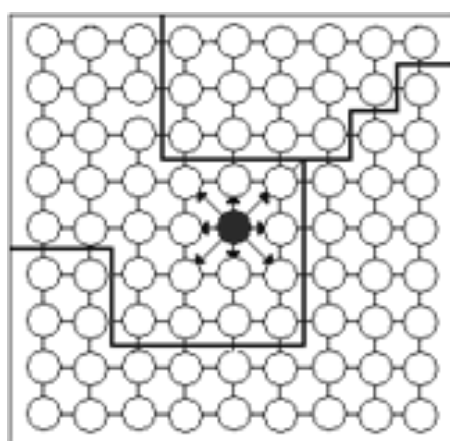
- Os algoritmos para a implementação do processo de crescimento de região estão baseados na comparação dos quatro (ou oito) primeiros pixels vizinhos

$a_{i-1,j-1}$	$a_{i-1,j}$	$a_{i-1,j+1}$
$a_{i,j-1}$	$a_{i,j}$	$a_{i,j+1}$
$a_{i+1,j-1}$	$a_{i+1,j}$	$a_{i+1,j+1}$

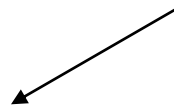
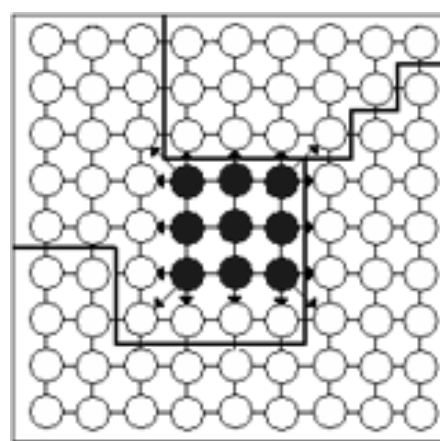
Segmentação Baseada em Regiões



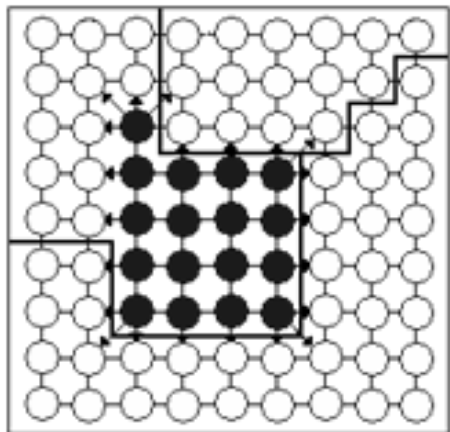
(1)



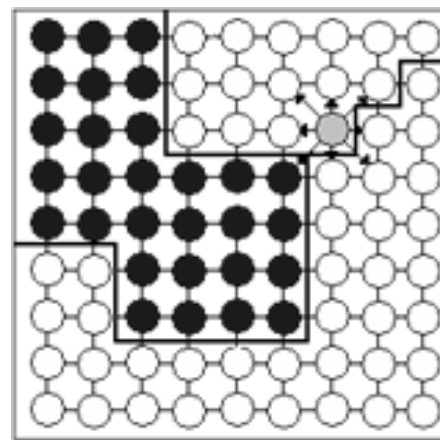
(2)



(3)



(4)



Segmentação Baseada em Regiões



O algoritmo deve:

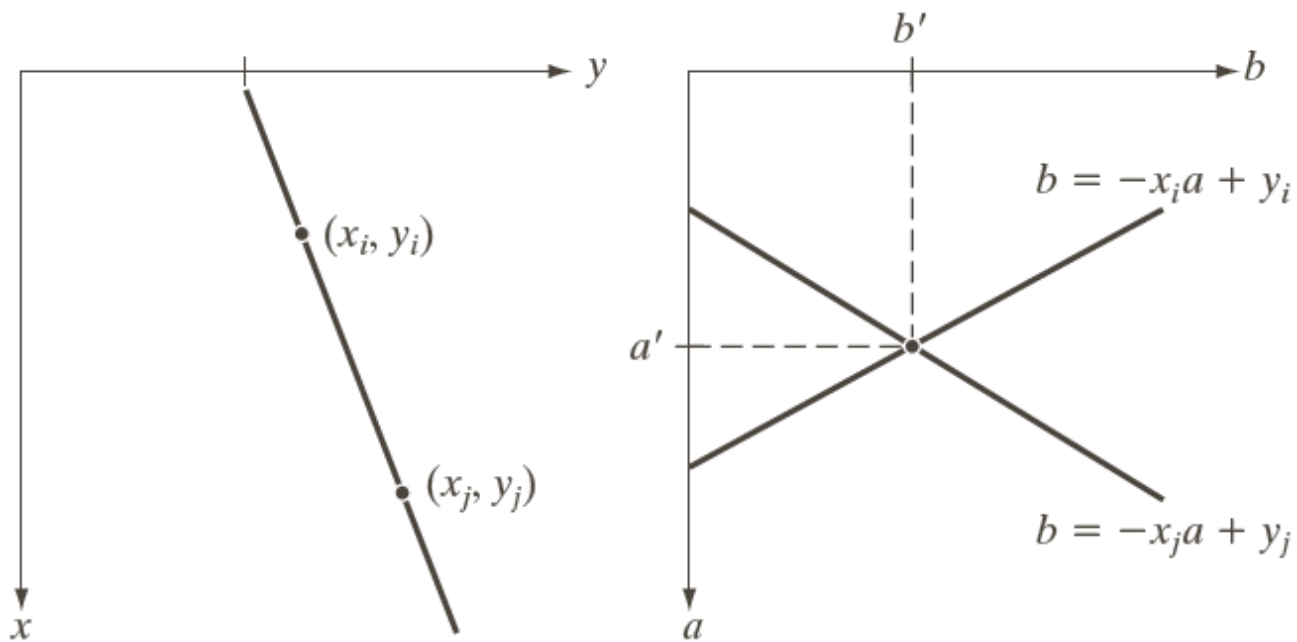
- realizar a comparação entre o pixel base e os seus vizinhos de acordo com algum critério de agregação;
 - incorporar ou não estes vizinhos à região e prosseguir para um próximo pixel para repetir o processo;
 - definir uma ordem de comparação com os vizinhos e como será escolhido o próximo pixel base.
-
- Nota-se que em cada passo do processo, podemos ter até quatro novos pixels candidatos em relação ao pixel base na próxima interação. Ao se escolher um, deve-se manter os demais em espera e retornar a eles em algum momento.

Segmentação Baseada em Regiões



- Duas maneiras de se implementar o processo de crescimento de região descrito são:
 - Com o uso de funções recursivas
 - Com pilhas.

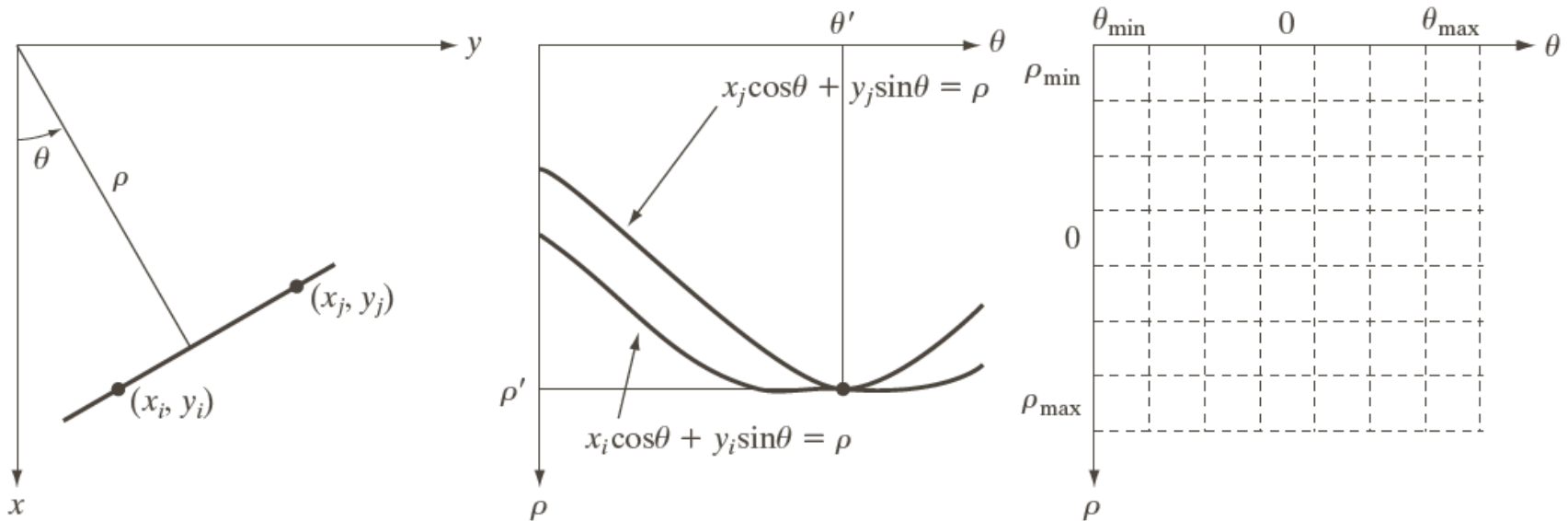
Transformada de Hough



a b

FIGURE 10.31
(a) xy -plane.
(b) Parameter space.

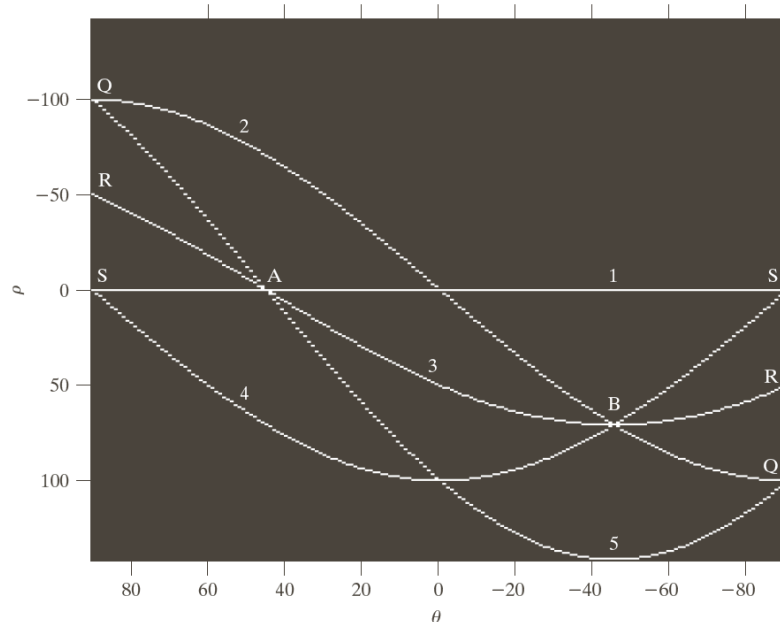
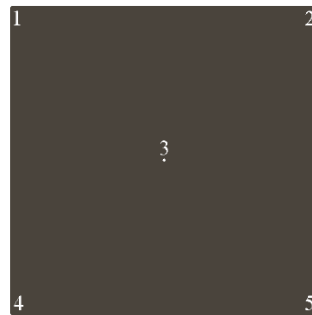
Transformada de Hough



a b c

FIGURE 10.32 (a) (ρ, θ) parameterization of line in the xy -plane. (b) Sinusoidal curves in the $\rho\theta$ -plane; the point of intersection (ρ', θ') corresponds to the line passing through points (x_i, y_i) and (x_j, y_j) in the xy -plane. (c) Division of the $\rho\theta$ -plane into accumulator cells.

Transformada de Hough



a
b

FIGURE 10.33

(a) Image of size 101×101 pixels, containing five points.

(b) Corresponding parameter space. (The points in (a) were enlarged to make them easier to see.)

Transformada de Hough

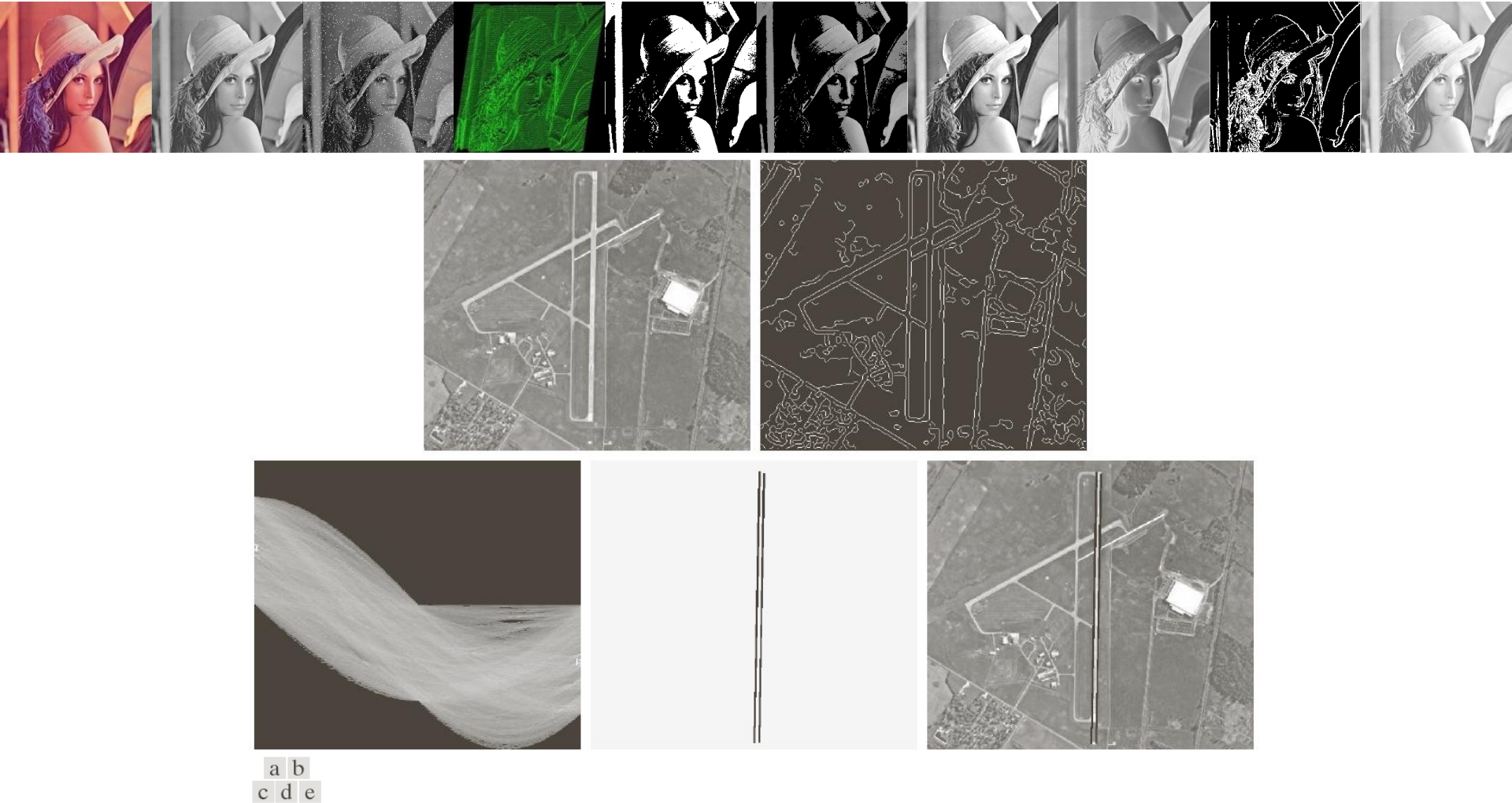


FIGURE 10.34 (a) A 502×564 aerial image of an airport. (b) Edge image obtained using Canny's algorithm. (c) Hough parameter space (the boxes highlight the points associated with long vertical lines). (d) Lines in the image plane corresponding to the points highlighted by the boxes). (e) Lines superimposed on the original image.