

# Introdução ao Processamento Digital de Imagem MC920 / MO443

Prof. Hélio Pedrini

Instituto de Computação

UNICAMP

<http://www.ic.unicamp.br/~helio>

1º Semestre de 2020

## 1 Segmentação

- Detecção de Descontinuidades
- Detecção de Pontos, Retas e Bordas
- Operadores de Gradiente
- Operador de Canny
- Detecção de Junções ou Cantos
- Transformada de Hough
- Limiarização Global e Local
- Segmentação de Regiões
- Segmentação por Divisor de Águas

# Segmentação de Imagens

- O processo de *segmentação* particiona o conjunto de dados de entrada em estruturas com conteúdo semântico relevante para a aplicação em questão, de modo a facilitar a tarefa complexa de interpretação dos dados contidos em imagens digitais.
- Essas estruturas correspondem a objetos ou partes de objetos que auxiliarão o processo de interpretação das imagens.
- Após a segmentação, cada objeto é descrito por meio de suas propriedades geométricas e topológicas.
  - ▶ por exemplo: atributos como área, forma e textura podem ser extraídos dos objetos e utilizados no processo de análise.
- O processo de análise de imagens é considerado:
  - ▶ *supervisionado*: quando há regiões da imagem em que se dispõe de informações que permitem a identificação de uma ou mais classes de interesse.
  - ▶ *não supervisionado*: quando a região é associada a uma classe que é aprendida com base na similaridade entre as amostras de treinamento, ou seja, são utilizadas técnicas para reconhecer as classes presentes na imagem.

# Segmentação de Imagens

- Processar uma imagem de modo a segmentar um número de objetos é uma tarefa difícil e extremamente dependente da correta extração de características dos objetos, especialmente em imagens ruidosas.
- As abordagens convencionais para segmentação de imagens são normalmente baseadas nas propriedades básicas dos níveis de cinza da imagem, buscando detectar:
  - ▶ *descontinuidades*: os métodos visam particionar a imagem com base em mudanças abruptas nos níveis de cinza, caracterizadas pela presença de pontos isolados, linhas ou bordas na imagem.
  - ▶ *similaridades*: os métodos procuram agrupar pontos da imagem que apresentam valores similares para um determinado conjunto de características.

## Detecção de Descontinuidades

- Os tipos básicos de descontinuidades normalmente detectadas em imagens digitais são pontos, segmentos de retas, junções e bordas.
- Uma maneira comum de identificação de descontinuidades é por meio da varredura da imagem por uma máscara, ou seja, o processo de convolução descrito anteriormente.
- No caso de uma máscara  $w$  com tamanho de  $3 \times 3$  pixels, ilustrada abaixo, esse procedimento envolve o cálculo da soma dos produtos dos coeficientes da máscara pelos níveis de cinza da região delimitada pela máscara.

$$w = \begin{array}{|c|c|c|} \hline w_1 & w_2 & w_3 \\ \hline w_4 & w_5 & w_6 \\ \hline w_7 & w_8 & w_9 \\ \hline \end{array}$$

**Figura:** Máscara de  $3 \times 3$  pixels.

- Dessa forma, a resposta  $R$  da máscara posicionada sobre um ponto da imagem é dada por

$$R = w_1 z_1 + w_2 z_2 + \dots + w_9 z_9 = \sum_{i=1}^9 w_i z_i \quad (1)$$

em que  $z_i$  é o nível de cinza associado com o coeficiente  $w_i$  da máscara.

- A resposta da máscara é definida em relação à sua posição central.
- Quando a máscara é posicionada em um pixel na fronteira da imagem, a resposta é calculada, utilizando-se a vizinhança parcial apropriada.

## Detecção de Pontos

- A detecção de pontos isolados em uma imagem pode ser realizada pela aplicação direta da máscara  $h$  definida como

$$h = \begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline -1 & 8 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array}$$

- Um ponto é detectado na posição central da máscara se

$$|R| > T \quad (2)$$

em que  $T$  é um limiar não-negativo e  $R$  é dado pela equação 1.

- A partir do cálculo dessas diferenças ponderadas entre os valores do ponto central e de seus vizinhos, um ponto é detectado se houver uma discrepância entre seu valor de nível de cinza e de seus vizinhos.
- Quando a máscara  $h$  é posicionada sobre uma região homogênea da imagem, ou seja, todos os pixels pertencentes à região possuem a mesma intensidade, a resposta da máscara é nula ( $R = 0$ ).

- Segmentos de retas também podem ser detectados pelo uso de máscaras.
- A máscara  $h_1$  abaixo pode ser utilizada para detectar retas orientadas horizontalmente.
  - ▶ Para uma imagem com intensidade de fundo constante, a resposta máxima da máscara  $h_1$  ocorre quando a reta (com largura de um pixel) passa pela linha central da máscara.
- Analogamente, a máscara  $h_2$  responde melhor a retas orientadas a  $45^\circ$ , a máscara  $h_3$  a retas verticais e a máscara  $h_4$  a retas orientadas a  $135^\circ$ .

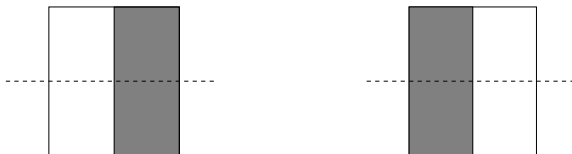
$$h_1 = \begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix} \quad h_2 = \begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix} \quad h_3 = \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix} \quad h_4 = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$



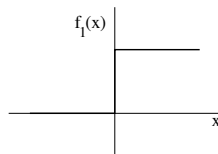
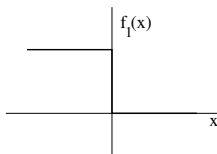
# Detecção de Bordas

- Uma borda é o limite ou a fronteira entre duas regiões com propriedades relativamente distintas de nível de cinza.
- Assume-se que as regiões em questão são suficientemente homogêneas, de maneira que a transição entre duas regiões pode ser determinada com base apenas na descontinuidade dos níveis de cinza.
- Basicamente, a ideia associada à maioria das técnicas para detecção de bordas é o cálculo de um operador local diferencial.
- A figura a seguir mostra:
  - (a) duas imagens formadas por uma região clara e uma região escura.
  - (b) o perfil representando o nível de cinza ao longo de uma linha de varredura horizontal da imagem é modelado como uma mudança abrupta dos níveis de cinza.
  - (c) perfil modelado como uma mudança suave dos níveis de cinza.
  - (d) derivada primeira do perfil de cinza ao longo da linha de varredura horizontal correspondente à imagem (c).
  - (e) derivada segunda.

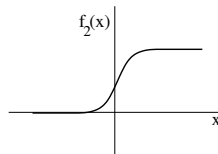
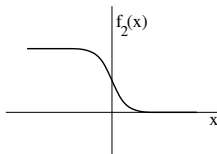
# Detecção de Bordas



(a) imagens contendo uma região clara e uma região escura

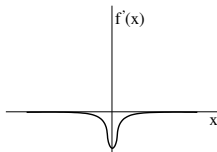


(b) perfil mostrando variação abrupta de intensidade de cinza ao longo de uma linha de varredura horizontal

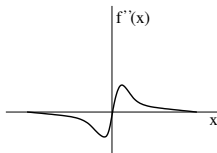
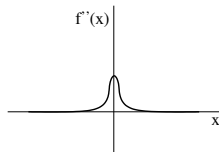


(c) perfil mostrando variação suave dos níveis de cinza

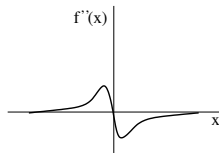
# Detecção de Bordas



(d) derivada primeira do perfil de níveis de cinza



(e) derivada segunda do perfil de níveis de cinza



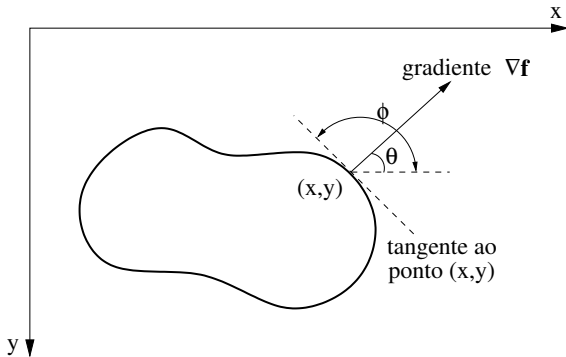
- A derivada primeira é positiva nas transições da região escura para a região clara, negativa nas transições da região clara para escura e nula nas áreas de nível de cinza constante.
- A derivada segunda é positiva na parte da transição associada ao lado escuro da borda, negativa na parte da transição associada ao lado claro da borda e nula nas áreas de nível de cinza constante.
- Assim, a magnitude da derivada primeira pode ser utilizada na detecção de uma borda em uma imagem, enquanto o sinal da derivada segunda possui um cruzamento em zero, ou seja, uma indicação de que há uma mudança de sinal na transição dos níveis de cinza, permitindo a localização das bordas em uma imagem.
- A derivada primeira em qualquer ponto da imagem é obtida usando-se a magnitude do gradiente naquele ponto.
- A derivada segunda é obtida similarmente utilizando-se o operador Laplaciano.

# Operadores de Gradiente

- A detecção de bordas é, essencialmente, a operação de identificação de mudanças locais significativas nos níveis de cinza da imagem.
- Essas mudanças podem ser descritas por meio do conceito de derivada.
- Como uma imagem depende de duas coordenadas espaciais, as bordas da imagem podem ser expressas por derivadas parciais.
- Um operador comumente utilizado em diferenciação de imagens é o *gradiente*, que é um vetor cuja direção indica os locais nos quais os níveis de cinza sofrem maior variação.

## Operadores de Gradiente

- A figura a seguir mostra que a direção do gradiente é sempre perpendicular à direção tangente da borda.



**Figura:** Gradiente em uma borda.

- Dessa forma

$$\theta = \phi \pm \frac{\pi}{2} \quad (3)$$

O vetor gradiente  $\nabla f(x, y)$  de uma imagem na posição  $(x, y)$  pode ser calculado pelas derivadas parciais

$$\nabla f(x, y) = \frac{\partial f(x, y)}{\partial x} \mathbf{i} + \frac{\partial f(x, y)}{\partial y} \mathbf{j} \quad (4)$$

em que  $\mathbf{i}$  e  $\mathbf{j}$  são vetores unitários nas direções  $x$  e  $y$ , respectivamente.

- Desse modo, uma variação rápida de  $f(x, y)$  ao longo da direção  $x$  e lenta ao longo da direção  $y$  indica a presença de uma borda praticamente vertical.

- Algoritmo para identificação de pontos de borda baseada no conceito de gradiente.

---

**Algoritmo 1** Determinação de pontos de borda em uma imagem

---

```
1: entradas: uma imagem de entrada  $f$  com dimensões  $M \times N$  pixels e um limiar  $T$ .
2: for  $x = 0$  até  $M - 1$  do
3:   for  $y = 0$  até  $N - 1$  do
4:     // calcular a magnitude do gradiente  $\nabla f(x, y)$ 
5:     
$$\nabla f(x, y) = \sqrt{\left(\frac{\partial f(x, y)}{\partial x}\right)^2 + \left(\frac{\partial f(x, y)}{\partial y}\right)^2}$$

6:     // efetuar a limiarização
7:     if  $\nabla f(x, y) > T$  then
8:        $(x, y)$  é um ponto da borda
9:     end if
10:   end for
11: end for
```

---



- Na forma matricial, o gradiente da imagem pode ser expresso como

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

- Em detecção de bordas, a magnitude do vetor gradiente é uma medida importante, em geral, chamada simplesmente de *gradiente* e denotada por  $\nabla f$ , sendo

$$\nabla f = \text{mag}(\nabla \mathbf{f}) = \sqrt{G_x^2 + G_y^2} = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \quad (5)$$

- A magnitude do gradiente equivale à maior taxa de variação de  $f(x, y)$  por unidade de distância na direção de  $\nabla \mathbf{f}$ .

## Operadores de Gradiente

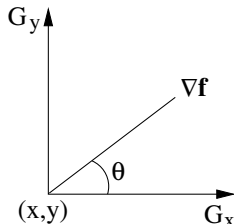
- Em razão do custo computacional requerido pela equação 5, a magnitude do gradiente pode ser aproximada pelo uso dos valores absolutos

$$\nabla f \approx |G_x| + |G_y|$$

ou do valor máximo entre os gradientes na direção  $x$  e  $y$

$$\nabla f \approx \max(|G_x|, |G_y|)$$

- A direção do vetor gradiente é também uma medida importante.
- Seja  $\theta(x, y)$  o ângulo da direção do vetor  $\nabla f$  na posição  $(x, y)$ , conforme ilustrado na figura a seguir.



- Então, tem-se que

$$\theta(x, y) = \arctan \left( \frac{G_y}{G_x} \right)$$

em que o ângulo  $\theta$  é medido em relação ao eixo  $x$ .

- Uma mudança em intensidade pode ser detectada pela diferença entre os valores de pixels adjacentes.
- Bordas *verticais* podem ser detectadas pela diferença *horizontal* entre pontos, enquanto bordas *horizontais* podem ser detectadas pela diferença *vertical* entre pontos adjacentes da imagem.

## Operadores de Gradiente

- Seja a região da imagem mostrada na figura abaixo, em que os valores denotam os níveis de cinza dos pixels.

$f(x-1, y-1)$	$f(x, y-1)$	$f(x+1, y-1)$
$f(x-1, y)$	$f(x, y)$	$f(x+1, y)$
$f(x-1, y+1)$	$f(x, y+1)$	$f(x+1, y+1)$

**Figura:** Região da imagem formada por  $3 \times 3$  pixels.

- A magnitude do gradiente, mostrada na equação 5, pode ser aproximada no ponto  $f(x, y)$  de várias maneiras.
- Uma forma simples consiste em usar a diferença  $f(x, y) - f(x+1, y)$  na direção  $x$  e  $f(x, y) - f(x, y+1)$  na direção  $y$ , combinadas como

$$\nabla f \approx \sqrt{[f(x, y) - f(x+1, y)]^2 + [f(x, y) - f(x, y+1)]^2} \quad (6)$$

## Operadores de Gradiente

- Uma outra abordagem para aproximação da equação 5 é usar as diferenças cruzadas

$$\nabla f \approx \sqrt{[f(x, y) - f(x + 1, y + 1)]^2 + [f(x, y + 1) - f(x + 1, y)]^2} \quad (7)$$

ou usar os valores absolutos

$$\nabla f \approx |f(x, y) - f(x + 1, y + 1)| + |f(x, y + 1) - f(x + 1, y)| \quad (8)$$

- As equações 6 a 8 podem ser implementadas por meio de máscaras de tamanho  $2 \times 2$  pixels.
- Por exemplo, a equação 8 pode ser implementada tomando-se o valor absoluto das respostas das duas máscaras mostradas em 9 e somando-se os resultados.
- Essas máscaras são chamadas de operadores cruzados de *gradiente de Roberts*.

$$G_x = \begin{array}{|c|c|} \hline 1 & 0 \\ \hline 0 & -1 \\ \hline \end{array} \quad G_y = \begin{array}{|c|c|} \hline 0 & -1 \\ \hline 1 & 0 \\ \hline \end{array} \quad (9)$$

## Operadores de Gradiente

- Uma aproximação para a equação 5 também no ponto  $f(x, y)$ , mas agora usando uma vizinhança de  $3 \times 3$  pixels é dada por

$$\begin{aligned} \nabla f \approx & |[f(x+1, y-1) + f(x+1, y) + f(x+1, y+1)] - \\ & [f(x-1, y-1) + f(x-1, y) + f(x-1, y+1)]| + \\ & |[f(x-1, y+1) + f(x, y+1) + f(x+1, y+1)] - \\ & [f(x-1, y-1) + f(x, y-1) + f(x+1, y-1)]| \end{aligned} \quad (10)$$

- A diferença entre a terceira e a primeira colunas da região de  $3 \times 3$  pixels aproxima a derivada na direção  $x$ , enquanto a diferença entre a terceira e a primeira linhas aproxima a derivada na direção  $y$ .
- As máscaras mostradas em 11, chamadas de *operadores de Prewitt*, podem ser usadas para implementar a equação 10.

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (11)$$

## Operadores de Gradiente

- O *operador de Sobel* aproxima a magnitude do gradiente como a diferença de valores ponderados dos níveis de cinza da imagem como

$$\begin{aligned}G_x &\approx [f(x+1, y-1) + 2f(x+1, y) + f(x+1, y+1)] - \\&\quad [f(x-1, y-1) + 2f(x-1, y) + f(x-1, y+1)] \\G_y &\approx [f(x-1, y+1) + 2f(x, y+1) + f(x+1, y+1)] - \\&\quad [f(x-1, y-1) + 2f(x, y-1) + f(x+1, y-1)]\end{aligned}\tag{12}$$

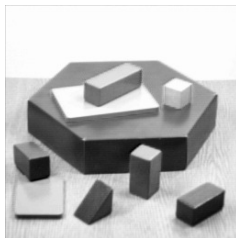
- As máscaras mostradas em 13 implementam o operador de Sobel

$$G_x = \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array} \qquad G_y = \begin{array}{|c|c|c|} \hline -1 & -2 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 2 & 1 \\ \hline \end{array}\tag{13}$$

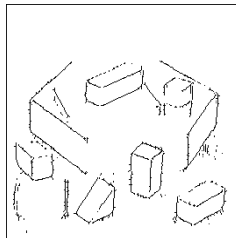
em que, como anteriormente, os níveis de cinza dos pixels de uma região são sobrepostos pelas máscaras centradas no pixel  $(x, y)$  da imagem.

# Operadores de Gradiente

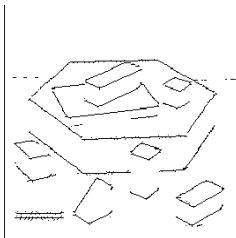
- Ilustração da detecção de bordas com o operador de Sobel.



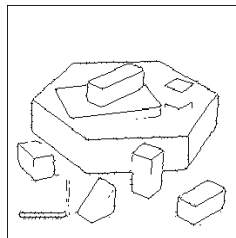
(a) imagem original



(b) resultado de  $G_x$



(c) resultado de  $G_y$



(d) combinação de  $G_x$  e  $G_y$



## Operadores de Gradiente

- O *operador de Kirsch* consiste em oito máscaras de convolução orientadas em  $45^\circ$ , mostradas a seguir.

5	-3	-3
5	0	-3
5	-3	-3

-3	-3	-3
5	0	-3
5	5	-3

-3	-3	-3
-3	0	-3
5	5	5

-3	-3	-3
-3	0	5
-3	5	5

-3	-3	5
-3	0	5
-3	-3	5

-3	5	5
-3	0	5
-3	-3	-3

5	5	5
-3	0	-3
-3	-3	-3

5	5	-3
5	0	-3
-3	-3	-3

- Para cada pixel da imagem, o operador aplica cada uma das oito orientações da máscara de derivadas e mantém o valor máximo.
- Assim, o cálculo da magnitude do gradiente, mostrado no algoritmo 1, não determina valores separados para  $G_x$  e  $G_y$ , o gradiente de cada pixel é obtido pela maior resposta do conjunto de oito máscaras.

## Operadores de Gradiente

- Robinson (1977) propôs o conjunto de máscaras descritas a seguir.

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

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

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

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

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

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

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

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

- A magnitude do gradiente é calculada de maneira similar àquela realizada pelas máscaras de Kirsch, ou seja, tomando-se o valor máximo dentre as oito máscaras.

- O *operador de Frei-Chen* (1977) utiliza um conjunto de nove máscaras que são cuidadosamente construídas para formar uma *base ortogonal*.
- Denotando as máscaras como  $M_1$  a  $M_9$ , então qualquer região de  $3 \times 3$  pixels da imagem pode ser expressa como uma combinação linear dessas máscaras, ou seja

$$w_1 M_1 + w_2 M_2 + \dots + w_9 M_9$$

- Para uma pequena região da imagem, os valores dos pesos  $w_i$  indicam a intensidade da contribuição feita para aquela região pela máscara  $M_i$ .

# Operadores de Gradiente

- As máscaras de Frei-Chen são mostradas a seguir.

$$M_1 = \begin{bmatrix} 1 & \sqrt{2} & 1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & -1 \end{bmatrix}$$

$$M_4 = \begin{bmatrix} \sqrt{2} & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & -\sqrt{2} \end{bmatrix}$$

$$M_7 = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

$$M_2 = \begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix}$$

$$M_5 = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$M_8 = \begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 0 & -1 & \sqrt{2} \\ 1 & 0 & -1 \\ -\sqrt{2} & 1 & 0 \end{bmatrix}$$

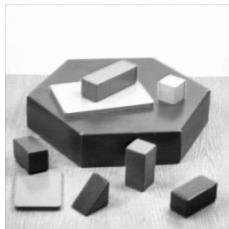
$$M_6 = \begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix}$$

$$M_9 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

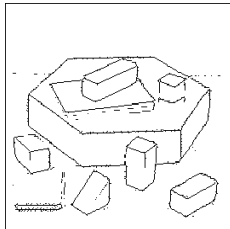
- As máscaras  $M_1$  a  $M_4$  são utilizadas para detectar bordas,  $M_5$  a  $M_8$  detectam retas, enquanto  $M_9$  representa uma média dos pixels na região de  $3 \times 3$  pixels.

## Operadores de Gradiente

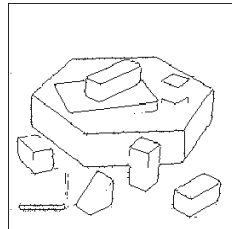
- Comparação dos resultados após a aplicação dos operadores de Roberts, Prewitt, Sobel, Kirsch e Frei-Chen em uma imagem.



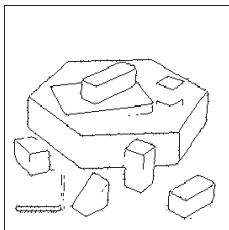
(a) imagem original



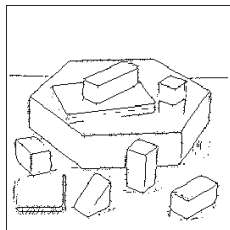
(b) Roberts



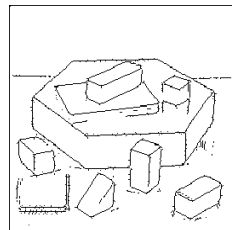
(c) Prewitt



(d) Sobel



(e) Kirsch



(f) Frei-Chen

## Laplaciano

- O operador *Laplaciano* de uma função bidimensional contínua  $f(x, y)$  é definido por uma derivada de segunda ordem como

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (14)$$

- Assim como no caso do gradiente, a equação 14 pode ser aproximada na forma discreta de maneiras diferentes.
- Para o caso de uma região de  $3 \times 3$  pixels, uma forma comum é

$$\frac{\partial^2 f}{\partial x^2} \approx f(x-1, y) - 2f(x, y) + f(x+1, y) \quad (15)$$

$$\frac{\partial^2 f}{\partial y^2} \approx f(x, y-1) - 2f(x, y) + f(x, y+1) \quad (16)$$

- Combinando as equações 15 e 16 em um único operador, o Laplaciano pode ser aproximado como

$$\nabla^2 f \approx f(x, y-1) + f(x-1, y) + f(x+1, y) + f(x, y+1) - 4f(x, y) \quad (17)$$

- A exigência para a definição do Laplaciano na forma discreta é que o coeficiente associado ao pixel central seja positivo e que os outros pixels externos sejam negativos.
- A resposta do operador Laplaciano é nula quando o ponto central e seus vizinhos tiverem os mesmos valores de intensidade, ou seja, pertencerem a uma região homogênea da imagem.
- A máscara  $h_1$ , mostrada a seguir, pode ser usada na implementação da equação 17, tal que as duas matrizes que compõem a máscara correspondem às derivadas segundas ao longo de todas as linhas e colunas, respectivamente, assim como no Laplaciano contínuo expresso na equação 14.

$$h_1 = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 2 & -1 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & -1 & 0 \\ 0 & 2 & 0 \\ 0 & -1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

# Laplaciano

- Em certas situações, é desejável dar maior peso aos pontos vizinhos mais próximos do pixel central.
- Uma aproximação do Laplaciano com tal característica é dada pela máscara  $h_2$ , mostrada a seguir.

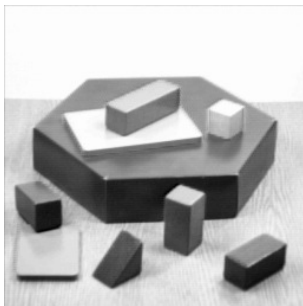
$$h_2 = \begin{array}{|c|c|c|} \hline -1 & -4 & -1 \\ \hline -4 & 20 & -4 \\ \hline -1 & -4 & -1 \\ \hline \end{array}$$

- Embora o operador Laplaciano responda a transições de intensidade, ele raramente é utilizado na prática para detecção de bordas.
- Por ser uma derivada de segunda ordem, o Laplaciano é tipicamente sensível a ruído de maneira inaceitável.
- Além disso, o Laplaciano usualmente cumpre papel secundário como um detector para determinar se um pixel está no lado claro ou escuro de uma borda.

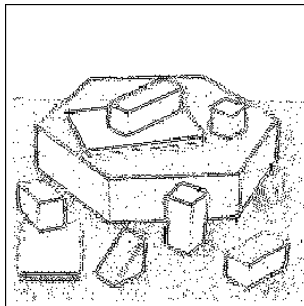


# Laplaciano

- Um uso mais geral do Laplaciano é a localização de bordas baseada na propriedade de mudança de sinal da derivada segunda, conhecida como cruzamento em zero.
- Uma aplicação do operador Laplaciano é ilustrada na figura a seguir.



(a) imagem original



(b) mapa de bordas

## Laplaciano do Gaussiano e Diferença do Gaussiano

- Como mencionado anteriormente, o uso de derivada de segunda ordem da intensidade da imagem para detectar pontos da borda é muito sensível a ruído.
- Portanto, é desejável filtrar o ruído antes da detecção de bordas.
- Para fazer isso, o *Laplaciano do Gaussiano*, proposto por Marr (1980), combina a filtragem Gaussiana com o operador Laplaciano para localizar bordas.
- Após a suavização da imagem por meio de um filtro Gaussiano, as bordas são identificadas pela presença de um cruzamento em zero na derivada segunda com um pico acentuado correspondente à derivada primeira.
- A resposta do operador Laplaciano do Gaussiano é obtida pela operação de convolução

$$\nabla^2((G(x, y) * f(x, y)))$$

em que  $f(x, y)$  é uma imagem suavizada por uma função Gaussiana.

# Laplaciano do Gaussiano e Diferença do Gaussiano

- A ordem na realização da diferenciação e da convolução pode ser alterada em razão da linearidade das operações envolvidas, resultando

$$(\nabla^2 G(x, y)) * f(x, y)$$

- Por simplicidade, pode-se usar  $r^2 = x^2 + y^2$ , em que  $r$  mede a distância das coordenadas à origem.
- Isso é razoável já que a Gaussiana<sup>1</sup> é circularmente simétrica.

---

1

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(\frac{-(x^2 + y^2)}{2\sigma^2}\right)$$

## Laplaciano do Gaussiano e Diferença do Gaussiano

- Portanto, a equação da função Gaussiana torna-se

$$G(r) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{r^2}{2\sigma^2}\right)$$

- A derivada primeira,  $G'(r)$ , é

$$G'(r) = -\frac{r}{2\pi\sigma^4} \exp\left(-\frac{r^2}{2\sigma^2}\right)$$

- A derivada segunda,  $G''(r)$ , ou seja, o Laplaciano do Gaussiano, é

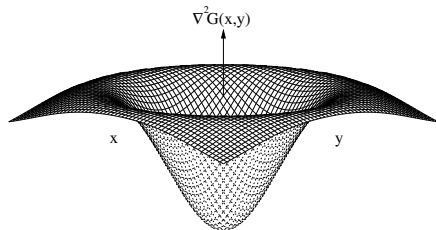
$$G''(r) = \nabla^2 G(r) = -\frac{1}{2\pi\sigma^4} \left(1 - \frac{r^2}{\sigma^2}\right) \exp\left(-\frac{r^2}{2\sigma^2}\right)$$

## Laplaciano do Gaussiano e Diferença do Gaussiano

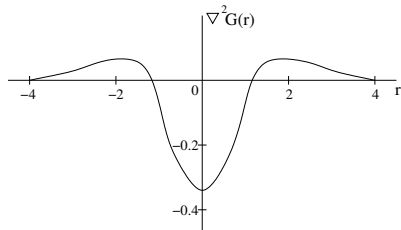
- Substituindo  $r^2$  de volta por  $x^2 + y^2$ , obtém-se

$$\nabla^2 G(x, y) = -\frac{1}{2\pi\sigma^4} \left(1 - \frac{x^2 + y^2}{\sigma^2}\right) \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (18)$$

que é normalmente conhecido como operador *chapéu mexicano*, mostrado na figura a seguir.



(a) Laplaciano do Gaussiano ( $\nabla^2 G$ ) para  $\sigma = 1.0$



(b) seção transversal do operador

# Laplaciano do Gaussiano e Diferença do Gaussiano

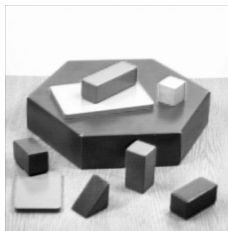
- O parâmetro  $\sigma$  controla o grau de suavização do filtro Gaussiano.
- Um exemplo de máscara com  $5 \times 5$  pixels para aproximar o Laplaciano do Gaussiano é mostrado a seguir.

$$h =$$

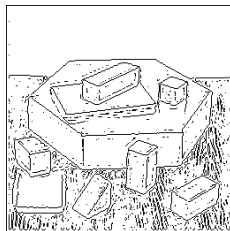
0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

# Laplaciano do Gaussiano e Diferença do Gaussiano

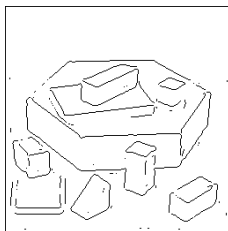
- Ilustração da aplicação do operador Laplaciano do Gaussiano para diferentes graus de suavização.



(a) imagem original



(b)  $\sigma = 1.0$



(c)  $\sigma = 2.0$



(d)  $\sigma = 3.0$

## Laplaciano do Gaussiano e Diferença do Gaussiano

- Há evidências de que o sistema visual humano realiza operações muito semelhantes às realizadas pelo operador  $\nabla^2 G$ .
- Huertas e Medioni (1986) mostraram que o Laplaciano do Gaussiano pode ser descrito como a soma de dois filtros.
- Tal decomposição acelera consideravelmente o cálculo do operador  $\nabla^2 G$ . A equação 18 pode ser reescrita como

$$\nabla^2 G(x, y) = k \left( 1 - \frac{x^2 + y^2}{\sigma^2} \right) \exp \left( \frac{-(x^2 + y^2)}{2\sigma^2} \right) \quad (19)$$

em que  $k$  é um fator que normaliza a soma dos elementos da máscara para zero.



- A decomposição é realizada pela soma de dois filtros separáveis  $h_1$  e  $h_2$  como

$$\nabla^2 G(x, y) = h_1(x)h_2(y) + h_2(x)h_1(y) \quad (20)$$

sendo

$$h_1(\xi) = -\sqrt{k} \left( 1 - \frac{\xi^2}{\sigma^2} \right) \exp \left( -\frac{\xi^2}{2\sigma^2} \right) \quad (21)$$

$$h_2(\xi) = \sqrt{k} \exp \left( -\frac{\xi^2}{2\sigma^2} \right) \quad (22)$$

## Laplaciano do Gaussiano e Diferença do Gaussiano

- O operador Laplaciano do Gaussiano,  $\nabla^2 G(x, y)$ , pode ser aproximado por convolução com uma máscara que é a diferença entre duas funções Gaussianas com valores  $\sigma_1$  e  $\sigma_2$  significativamente diferentes.
- Essa técnica, conhecida como *Diferença das Gaussianas*, é definida como

$$DoG(\sigma_1, \sigma_2) = \frac{1}{2\pi\sigma_1^2} \exp\left(\frac{-(x^2 + y^2)}{2\sigma_1^2}\right) - \frac{1}{2\pi\sigma_2^2} \exp\left(\frac{-(x^2 + y^2)}{2\sigma_2^2}\right)$$

- Marr (1980) mostram que uma boa escolha é  $\sigma_2/\sigma_1 = 1.6$ , para que o operador se aproxime do Laplaciano do Gaussiano.

# Laplaciano do Gaussiano e Diferença do Gaussiano

- Um exemplo de máscara com  $7 \times 7$  pixels para aproximar a Diferença das Gaussianas é mostrada a seguir.

$$h =$$

0	0	-1	-1	-1	0	0
0	-2	-3	-3	-3	-2	0
-1	-3	5	5	5	-3	-1
-1	-3	5	16	5	-3	-1
-1	-3	5	5	5	-3	-1
0	-2	-3	-3	-3	-2	0
0	0	-1	-1	-1	0	0

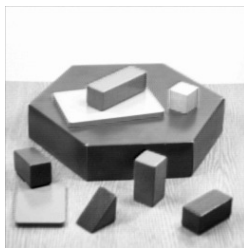
- Canny (1986) propôs um método para detecção de bordas que procura otimizar a localização de pontos da borda na presença de ruído.
- Inicialmente, a imagem é suavizada por meio de um filtro Gaussiano.
- Em seguida, a magnitude e a direção do gradiente são calculadas utilizando aproximações baseadas em diferenças finitas para as derivadas parciais, de maneira similar aos métodos de gradiente apresentados anteriormente.
- Após o cálculo do gradiente, a borda é localizada tomando-se apenas os pontos cuja magnitude seja localmente máxima na direção do gradiente.
- Essa operação, chamada de *supressão não-máxima*, reduz a espessura das bordas. Entretanto, a borda pode ainda conter certos fragmentos espúrios causados pela presença de ruído ou textura fina.

# Operador de Canny

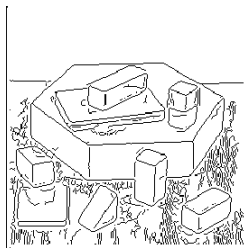
- Uma abordagem possível para remover esses fragmentos é utilizar um limiar durante a etapa de supressão não-máxima.
- A escolha do valor desse limiar é uma tarefa complexa, podendo acarretar em bordas falsas caso o limiar seja muito baixo ou, por outro lado, em perda de fragmentos da borda caso o limiar seja muito alto.
- Para resolver esse problema, o operador de Canny utiliza dois limiares diferentes,  $T_1$  e  $T_2$ , com  $T_2 > T_1$ .
- Essa etapa é conhecida como *limiarização com histerese*.
- Pontos da borda que possuem gradiente maior que  $T_2$  são mantidos como pontos da borda.
- Qualquer outro ponto conectado a esses pontos da borda é considerado como pertencente à borda se a magnitude de seu gradiente estiver acima de  $T_1$ .
- Isso evita que as bordas fiquem fragmentadas em múltiplos segmentos.
- A escolha dos limiares  $T_1$  e  $T_2$  é feita com base em uma estimativa da relação sinal-ruído.

# Operador de Canny

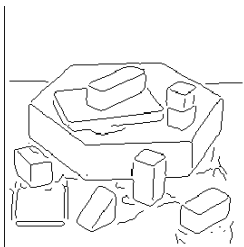
- Ilustração da aplicação do operador de Canny em uma imagem.



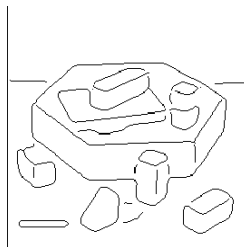
(a) imagem original



(b)  $\sigma = 0.5$



(c)  $\sigma = 1.0$



(d)  $\sigma = 2.0$

- Boie e Cox (1987) propuseram um algoritmo similar ao desenvolvido por Canny, em que o detector de máximos locais foi substituído pelo cruzamento em zero da derivada segunda e a direção do gradiente foi substituída pela direção da derivada primeira cuja magnitude é máxima.
- Seja  $M$  a magnitude máxima da derivada primeira,  $F$  e  $S$  os valores das derivadas primeira e segunda em uma dada direção e  $(dx, dy)$  o deslocamento de um pixel na mesma direção.
- Para um ponto  $(x, y)$  a ser marcado como borda nessa direção, a derivada primeira deve estar próxima do valor máximo, ou seja

$$|F(x, y)| > T_1 \quad \text{e} \quad M(x, y) - |F(x, y)| \leq T_2$$

- O ponto deve também estar no cruzamento em zero da derivada segunda, ou seja, uma das condições a seguir deve ser satisfeita:

a)  $(x + \frac{dx}{2}, y + \frac{dy}{2}) > T_3$  e  $S(x - \frac{dx}{2}, y - \frac{dy}{2}) < -T_3$

b)  $\left| S(x + \frac{dx}{2}, y + \frac{dy}{2}) \right| \leq T_3$  e  $S(x + \frac{3dx}{2}, y + \frac{3dy}{2}) < -T_3$  e  
 $S(x - \frac{dx}{2}, y - \frac{dy}{2}) > T_3$

c)  $\left| S(x - \frac{dx}{2}, y - \frac{dy}{2}) \right| \leq T_3$  e  $S(x - \frac{3dx}{2}, y - \frac{3dy}{2}) < -T_3$  e  
 $S(x + \frac{dx}{2}, y + \frac{dy}{2}) > T_3$



# Transformada de Hough

- Um problema comum na área de análise de imagens é a detecção de um conjunto de pontos na imagem que pertençam a uma curva específica, tal como segmento de reta, circunferência, elipse e outras.
- Essas curvas constituem uma família representada como

$$f(\mathbf{v}, \mathbf{p}) = 0 \quad (23)$$

em que  $\mathbf{v}$  é um vetor de coordenadas e  $\mathbf{p}$  é um vetor de parâmetros característicos da curva.

- A partir de um número de pontos  $p_i(x_i, y_i)$  na imagem, o objetivo é identificar se subconjuntos de pontos pertencem a uma determinada curva.

# Transformada de Hough

- Para o caso particular de segmentos de retas, o problema consiste em achar subconjuntos de pontos que sejam colineares.
- Uma solução possível é encontrar todos os segmentos de retas formados entre cada par de pontos e buscar os subconjuntos de pontos que estejam próximos dos segmentos de retas.
- Entretanto, essa abordagem envolve um número elevado de iterações, sendo proibitiva computacionalmente para a maior parte das aplicações.
- Hough (1962) propôs um método, comumente conhecido como *transformada de Hough*, para detectar curvas da forma representada pela equação 23.

## Detecção de Retas pela Transformada de Hough

- A equação da reta na forma declividade-intersecção é representada como

$$y = mx + b \quad (24)$$

em que  $m$  é a declividade da reta e  $b$  é o ponto de intersecção da reta com o eixo  $y$ .

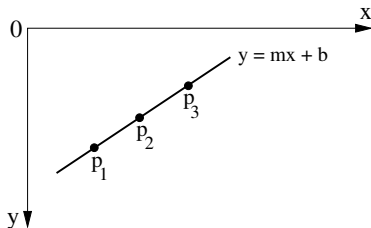
- Para valores diferentes de  $m$  e  $b$ , infinitas retas passam por um ponto  $p_1(x_1, y_1)$ , todas elas satisfazendo a equação  $y_1 = mx_1 + b$ .
- Analogamente, infinitas retas que passam por um ponto  $p_2(x_2, y_2)$  podem ser expressas pela equação  $y_2 = mx_2 + b$ .
- Rearranjando a equação 24 de tal forma que  $m$  e  $b$  sejam os parâmetros e que  $x$  e  $y$  sejam constantes, tem-se  $b = y - mx$ .

# Detecção de Retas pela Transformada de Hough

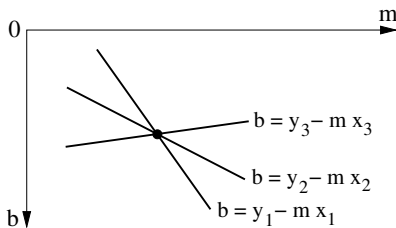
- O plano  $mb$  é conhecido como *espaço de parâmetros*.
- Todas as retas que passam pelo ponto  $p_1$  são representadas no espaço de parâmetros pela equação  $b = y_1 - mx_1$ .
- Do mesmo modo, a equação  $b = y_2 - mx_2$  representa o conjunto de todas as retas que passam pelo ponto  $p_2$  no plano da imagem.
- O ponto  $(m, b)$  no espaço de parâmetros é comum a essas duas retas associadas aos pontos  $p_1$  e  $p_2$ .
- De fato, todos os pontos que são colineares no plano da imagem se interceptam em um mesmo ponto no espaço de parâmetros.

## Detecção de Retas pela Transformada de Hough

- A figura a seguir ilustra o mapeamento entre o plano da imagem e o espaço de parâmetros.



(a) plano da imagem



(b) espaço de parâmetros

- Pontos nos quais muitas retas se interceptam no espaço de parâmetros correspondem a muitos pixels colineares, os quais potencialmente formam um segmento de reta na imagem.
- Isso significa que as coordenadas no espaço  $(m, b)$  do ponto de intersecção fornecem os parâmetros da reta no espaço  $(x, y)$  que contém esses pontos.

## Detecção de Retas pela Transformada de Hough

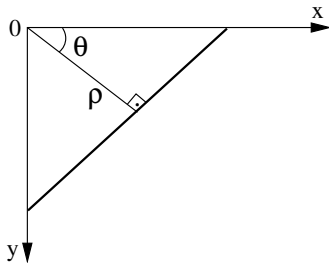
- Esse conceito forma a base da transformada de Hough para detecção de retas.
- Pixels são convertidos em retas no espaço  $(m, b)$  e, então, os pontos de intersecção de várias retas são localizados e agrupados em segmentos de reta.
- A equação 24 apresenta problema quando as retas forem aproximadamente verticais, ou seja, quando a declividade tender a infinito.
- Uma maneira de contornar tal situação é utilizar a equação da reta na forma polar

$$\rho = x \cos \theta + y \sin \theta \quad (25)$$

em que  $\rho$  é a distância perpendicular da origem  $(0, 0)$  à reta e  $\theta$  é o ângulo formado entre a reta perpendicular e o eixo  $x$ .

# Detecção de Retas pela Transformada de Hough

- A figura a seguir ilustra a representação de uma reta na forma polar.



## Detecção de Retas pela Transformada de Hough

- A conversão da equação 24 para a forma polar pode ser feita por meio das transformações

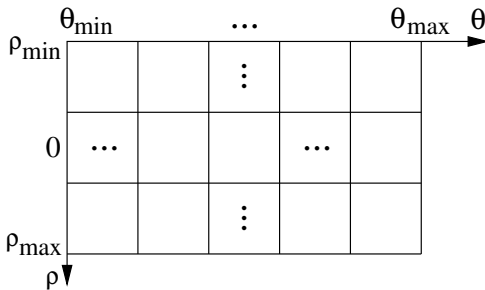
$$\sin \theta = \frac{-1}{\sqrt{m^2 + 1}} \quad \cos \theta = \frac{m}{\sqrt{m^2 + 1}} \quad \rho = \frac{-b}{\sqrt{m^2 + 1}}$$

- Em vez de utilizar coordenadas no espaço  $(m, b)$ , coordenadas  $(\rho, \theta)$  são então utilizadas.
- No espaço  $(\rho, \theta)$ , ou *espaço de Hough*, pontos colineares no espaço  $(x, y)$  correspondem agora a curvas senoidais que se interceptam no espaço  $(\rho, \theta)$ .



## Detecção de Retas pela Transformada de Hough

- Para implementar a transformada de Hough, o espaço  $(\rho, \theta)$  deve ser discretizado, conforme ilustra a figura a seguir.



(a) quantização do plano  $\rho\theta$

- Desde que  $\theta$  é medido com respeito ao eixo  $x$ , os valores possíveis de  $\theta$  variam de  $0^\circ$  a  $180^\circ$ . Se o intervalo for dividido em incrementos de  $1^\circ$ , então 181 ângulos serão possíveis.
- O domínio do eixo  $\rho$  varia de 0 a  $\sqrt{M^2 + N^2}$ , para uma imagem com dimensões de  $M \times N$  pixels.

# Detecção de Retas pela Transformada de Hough

- Após essa discretização do espaço  $(\rho, \theta)$ , cada célula do espaço é considerada uma célula de acumulação.
- Inicialmente, essas células possuem valor zero.
- Então, para cada ponto  $(x, y)$  no espaço da imagem,  $k$  pontos colineares de uma reta  $x \cos \theta + y \sin \theta = \rho$  levam a  $k$  curvas senoidais no plano  $\rho\theta$  (em contraste com retas no plano  $mb$ ) que se interceptam em  $(\rho, \theta)$  no espaço de parâmetros.
- Incrementando-se  $\theta$  e resolvendo para o valor de  $\rho$  correspondente, há  $k$  posições no acumulador associadas à célula determinada pelo ponto de intersecção  $(\rho, \theta)$ .
- Ao final desse procedimento, um valor  $k$  em uma célula corresponde a  $k$  pontos no plano  $\rho\theta$  que satisfazem a equação da reta.

## Detecção de Retas pela Transformada de Hough

- A precisão da colinearidade desses pontos é determinada pelo número de subdivisões no plano  $\rho\theta$ .
- Os valores mais altos (picos) no espaço de Hough correspondem aos parâmetros que caracterizam as retas da imagem.
- Após a detecção dos picos, deve-se determinar os segmentos de reta correspondentes a cada par de parâmetros.
- Uma vez que a equação de reta representa uma reta com comprimento infinito, segmentos reais devem ser determinados examinando-se os pixels de borda na imagem.

# Detecção de Retas pela Transformada de Hough

- Algoritmo para detecção de retas pela transformada de Hough.

---

## Algoritmo 2 Detecção de retas pela transformada de Hough

---

- 1: Discretizar o espaço de parâmetros  $(\rho, \theta)$  em intervalos finitos. Cada célula  $M(\rho, \theta)$  no espaço de parâmetros é um acumulador.
  - 2: Inicializar todas as células do acumulador com valor zero.
  - 3: Para cada ponto  $(x, y)$  no espaço da imagem, calcular os valores de  $\rho$  e  $\theta$  que satisfazem a equação da reta.
  - 4: Incrementar em 1 o acumulador  $M(\rho, \theta)$ .
  - 5: Após a determinação dos parâmetros de todos os pontos no espaço da imagem, os pontos de máximo (picos) na matriz de acumulação indicam forte evidência de retas na imagem.
-

## Detecção de Retas pela Transformada de Hough

- Se existirem várias retas que são detectadas pelo modelo, então existirão vários picos no espaço de parâmetros.
- Cada pico pode ser detectado e as retas associadas às retas correspondentes aos picos podem ser removidas.
- O processo continua para as retas remanescentes, enquanto existirem picos relevantes.
- Entretanto, pode ser difícil determinar se um pico é ou não relevante.
- Outra dificuldade com a transformada de Hough é sua dependência com relação ao número de parâmetros.
- Para um arco circular, por exemplo, o espaço de parâmetros possui três dimensões; para outras curvas mais complexas, a dimensionalidade pode ser ainda maior.

## Detecção de Retas pela Transformada de Hough

- Uma característica interessante da transformada de Hough é que mesmo segmentos que apresentam regiões obstruídas por outros objetos podem ser detectados.
- As regiões de descontinuidade na reta causarão apenas uma redução no valor do máximo local, devido ao menor número de pixels da reta que contribuem para a célula de acumulação correspondente.
- A transformada de Hough é pouco sensível à presença de ruído, já que os pontos da imagem corrompidos por ruído normalmente não serão mapeados em uma mesma célula de acumulação.

## Detecção de Retas pela Transformada de Hough

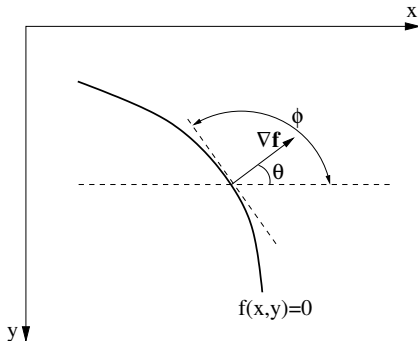
- Uma técnica para reduzir o número de parâmetros na transformada de Hough é utilizar a informação de gradiente, ou seja, a direção das bordas.
- Caso a direção do gradiente esteja disponível, a dimensão no espaço de parâmetros pode ser reduzida.
- Relembrando que a direção tangente  $\phi$  de qualquer função  $f(x, y)$  está relacionada à derivada  $dy/dx$ , então

$$\tan \phi = \frac{dy}{dx} = \frac{\partial f / \partial y}{\partial f / \partial x} = \nabla f \quad (26)$$

em que  $\partial f / \partial x$  e  $\partial f / \partial y$  são as derivadas parciais da função  $f(x, y)$  com respeito a  $x$  e  $y$ , respectivamente.

## Detecção de Retas pela Transformada de Hough

- Para um ponto arbitrário na curva especificada pela equação  $f(x, y) = 0$ , a direção  $\theta$  do gradiente é sempre perpendicular à direção tangente  $\phi$ , como pode ser observado na figura a seguir.



- O valor do parâmetro  $\theta$  coincide, então, com a direção do gradiente no ponto.
- Dessa forma, o passo 3 do algoritmo 2 torna-se mais simples, uma vez que apenas os pontos da borda devem ser considerados.

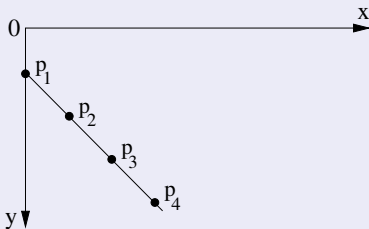


# Detecção de Retas pela Transformada de Hough

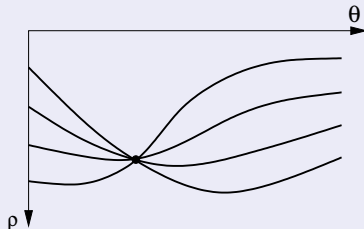
## Exemplo:

Sejam  $P_1(0, 1)$ ,  $P_2(1, 2)$ ,  $P_3(2, 3)$  e  $P_4(3, 4)$  quatro pixels que formam um segmento de reta.

- Esses pontos são mapeados para o espaço de parâmetros como quatro curvas senoidais, mostradas na figura a seguir.



(a) plano da imagem



(b) espaço de parâmetros

# Detecção de Retas pela Transformada de Hough

## Exemplo:

- Como os quatro pontos são colineares, as curvas se interceptam no ponto  $(\rho = \sqrt{2}/2, \theta = 3\pi/4)$ , representando a reta da equação 25.
- Portanto

$$\frac{\sqrt{2}}{2} = x \cos \frac{3\pi}{4} + y \sin \frac{3\pi}{4}$$

ou seja,  $y = x + 1$ .

## Detecção de Circunferências pela Transformada de Hough

- A transformada de Hough pode ser utilizada para detectar circunferências em uma imagem.
- Uma formulação possível de uma circunferência em coordenadas cartesianas é dada por

$$(x - a)^2 + (y - b)^2 = r^2 \quad (27)$$

em que  $(a, b)$  e  $r$  representam as coordenadas do centro e o raio da circunferência, respectivamente.

- Portanto, a estrutura de acumulação é tridimensional para o caso de detecção de circunferências.
- Para cada pixel  $(x, y)$ , a célula de acumulação  $(a, b, r)$  é incrementada se o ponto  $(a, b)$  estiver à distância  $r$  do ponto  $(x, y)$ .
- Se um centro  $(a, b)$  de uma circunferência de raio  $r$  é frequentemente encontrado no espaço de parâmetros, é altamente provável que uma circunferência com raio  $r$  e centro  $(a, b)$  exista na imagem.
- Dessa forma, picos no espaço de parâmetros corresponderão aos centros das circunferências no plano da imagem.

## Detecção de Circunferências pela Transformada de Hough

- As equações paramétricas da circunferência em coordenadas polares são

$$\begin{aligned}x &= a + r \cos \theta \\ y &= b + r \sin \theta\end{aligned}\tag{28}$$

- Resolvendo para os parâmetros da circunferência, tem-se que

$$\begin{aligned}a &= x - r \cos \theta \\ b &= y - r \sin \theta\end{aligned}\tag{29}$$

- Se o raio da circunferência for previamente conhecido, então é necessário incrementar o acumulador para o ponto  $(a, b)$  dado pelas equações 28 e 29.
- Nesse caso, circunferências com valor de raio  $r$  poderão ser detectadas a cada aplicação do algoritmo.
- Isso é aceitável se apenas valores conhecidos de raio ocorrem na imagem.

## Detecção de Circunferências pela Transformada de Hough

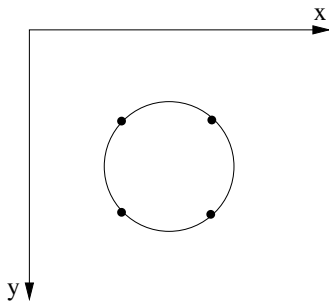
- Assim como no caso de detecção de retas, o esforço computacional pode ser reduzido se a informação de gradiente estiver disponível.
- Dada a direção  $\theta$  do gradiente em um ponto  $(x, y)$  da borda, pode-se calcular os valores de  $\sin \theta$  e  $\cos \theta$ .
- Unindo-se as equações 28 e 29, o raio da circunferência pode ser eliminado

$$b = a \tan \theta - x \tan \theta + y \quad (30)$$

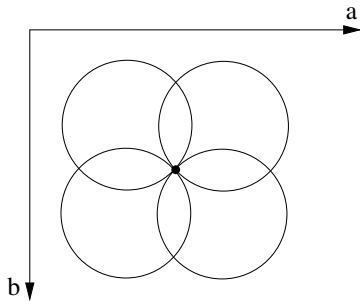
- A partir da direção  $\theta$  do gradiente em cada ponto da borda  $(x, y)$ , as células de acumulação no espaço de parâmetros  $(a, b)$  são incrementadas de acordo com a equação 30.

## Detecção de Circunferências pela Transformada de Hough

- A figura a seguir ilustra a detecção de circunferências por meio da transformada de Hough, em que são apresentados tanto o plano da imagem  $xy$  quanto o espaço de parâmetros  $ab$ .



(a) dados originais no espaço  $(x, y)$



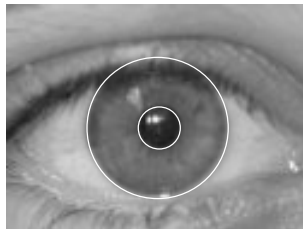
(b) correspondentes células de acumulação no espaço  $(a, b)$

## Detecção de Circunferências pela Transformada de Hough

- Uma aplicação prática da transformada de Hough é a detecção da pupila e da íris em imagens digitais (fotografias) obtidas a partir de iluminação infra-vermelha.
- Em sistemas biométricos, o reconhecimento da íris pode ser utilizado, por exemplo, na identificação criminal, autenticação e controle de acesso de usuários a ambientes ou locais específicos, como prédios ou salas.
- A figura a seguir ilustra a detecção da íris e da pupila pela transformada de Hough.



(a) imagem original



(b) detecção da íris e pupila

# Detecção de Circunferências pela Transformada de Hough

- Algoritmo para detecção de circunferências pela transformada de Hough.

---

**Algoritmo 3** Detecção de circunferências pela transformada de Hough

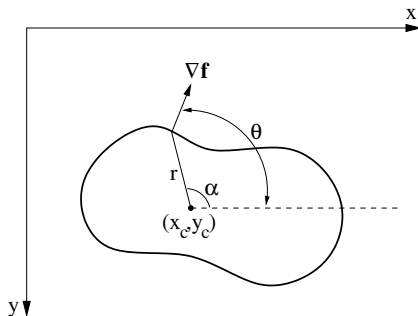
---

- 1: Quantizar o espaço de parâmetros  $a$  e  $b$ .
  - 2: Inicializar células de acumulação  $M(a, b)$  com zero.
  - 3: Calcular a magnitude do gradiente  $\nabla f(x, y)$  e ângulo  $\theta(x, y)$ .
  - 4: Para cada ponto em  $\nabla f(x, y)$ , incrementar todos pontos nas células de acumulação  $M(a, b)$  de acordo com a equação  $b = a \tan \theta - x \tan \theta + y$ .
  - 5: Picos locais na matriz de acumulação correspondem aos centros das circunferências na imagem.
-



## Generalização da Transformada de Hough

- A transformada de Hough pode ser generalizada para detectar curvas com forma arbitrária, ou seja, que não apresentam representação paramétrica.
- Na generalização da transformada de Hough proposta por Ballard (1981), um ponto de referência  $(x_c, y_c)$  é escolhido no interior do objeto, por exemplo, o centróide do objeto.
- Um segmento de reta arbitrário pode ser construído unindo o ponto de referência a um ponto da borda, conforme figura a seguir.



**Figura:** Ponto de referência em um objeto.

## Generalização da Transformada de Hough

- A direção do gradiente pode ser calculada na intersecção do segmento de reta com a borda do objeto.
- Uma tabela de referência, conhecida como *tabela-R*, é construída para armazenar dois parâmetros  $r$  e  $\alpha$  como uma função da direção da borda no ponto de intersecção.
- Para cada ponto da borda  $(x, y)$ , os parâmetros  $r$  e  $\alpha$  são calculados como

$$\begin{aligned} r &= \sqrt{(x - x_c)^2 + (y - y_c)^2} \\ \alpha &= \arctan\left(\frac{y - y_c}{x - x_c}\right) \end{aligned} \quad (31)$$

- A tabela resultante pode ser ordenada de acordo com as direções da borda nos pontos de intersecção.
- Deve-se notar que diferentes pontos  $P_i$  e  $P_j$  da borda do objeto podem ter a mesma direção da borda, ou seja,  $\theta_{P_i} = \theta_{P_j}$ .
- Isso significa que pode existir mais de um par  $(r, \alpha)$  para cada  $\theta$  tal que possa determinar as coordenadas de um ponto de referência.

# Generalização da Transformada de Hough

- Uma representação da tabela-R é ilustrada a seguir.

$\theta_1$	$(r_1^1, \alpha_1^1), (r_2^1, \alpha_2^1), \dots$
$\theta_2$	$(r_1^2, \alpha_1^2), (r_2^2, \alpha_2^2), \dots$
$\vdots$	$\vdots$
$\theta_k$	$(r_1^k, \alpha_1^k), (r_2^k, \alpha_2^k), \dots$

- A detecção da forma e sua localização na imagem pode ser realizada buscando-se, para cada ponto  $(x, y)$  da borda, a entrada da tabela cujo ângulo  $\theta_i$  é mais próximo à direção do gradiente no ponto  $(x, y)$ .

## Generalização da Transformada de Hough

- Para cada par  $(r, \alpha)$  da entrada indexada por  $\theta_i$ , deve-se calcular as coordenadas dos pontos de referência candidatos, ou seja

$$\begin{aligned}x_c &= x + r \cos \alpha \\y_c &= y + r \sin \alpha\end{aligned}\tag{32}$$

- Considerando  $M$  uma matriz de acumulação que armazena as células do espaço de parâmetros quantizado como  $M(x_{c_{\min}} \dots x_{c_{\max}}, y_{c_{\min}} \dots y_{c_{\max}})$ , a célula  $M(x_c, y_c)$  deve ser incrementada em 1.
- Após repetir esse procedimento para cada ponto  $(x, y)$  da borda, os valores das células da matriz  $M$  que excedem um limiar  $T$  representam as possíveis localizações da forma do objeto na imagem.
- Para tornar a detecção da forma de um objeto invariante quanto à rotação e à escala, dois parâmetros adicionais devem ser introduzidos, o fator de escala  $S$  e o ângulo de rotação  $\phi$ .

## Generalização da Transformada de Hough

- O espaço de parâmetros possui agora dimensão 4, sendo que a matriz de acumulação é denotada  $M(x_c, y_c, S, \phi)$ .
- A detecção da forma do objeto é similar à descrita anteriormente, alterando-se o cálculo das coordenadas dos pontos de referência candidatos.
- Para cada ponto da borda  $(x, y)$ , busca-se a entrada da tabela cujo ângulo  $\theta_i$  esteja próximo à direção do gradiente no ponto  $(x, y)$ .
- Para cada par  $(r, \alpha)$  associado a essa entrada, deve-se calcular

$$\begin{aligned}x_c &= x + r S \cos(\alpha + \phi) \\y_c &= y + r S \sin(\alpha + \phi)\end{aligned}\tag{33}$$

- A célula  $M(x_c, y_c, S, \phi)$  é incrementada em 1.
- Ao final do processo, todas as células da matriz  $M$  que excedem um limiar  $T$  representam o fator de escala  $S$ , o ângulo de rotação  $\phi$  da forma, bem como a localização do ponto de referência  $(x_c, y_c)$  na imagem.

# Generalização da Transformada de Hough

- Os principais passos da transformada de Hough generalizada são mostrados no algoritmo a seguir.

---

## Algoritmo 4 Transformada de Hough generalizada

---

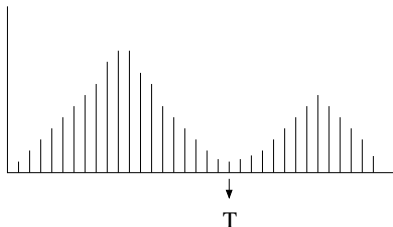
- 1: Criar uma tabela com  $k$  entradas, cada uma delas indexada por um ângulo  $\theta_i$ ,  $i = 1, 2, \dots, k$ , que varia de 0 a  $180^\circ$  com incremento de  $180/k$ , em que  $k$  é a resolução da orientação do gradiente.
  - 2: Escolher um ponto de referência  $(x_c, y_c)$  no interior do objeto.
  - 3: Para cada ponto  $(x, y)$  pertencente à borda do objeto, calcular:
    - 4:  $r = \sqrt{(x - x_c)^2 + (y - y_c)^2}$
    - 5:  $\alpha = \arctan((y - y_c)/(x - x_c))$
    - 6:  $\theta$ : direção do gradiente
  - 7: Adicionar o par  $(r, \alpha)$  na entrada da tabela-R com  $\theta_i$  mais próximo de  $\theta$ .
  - 8: Quantizar o espaço de parâmetros como uma matriz de acumulação,  $M(x_{c_{\min}} \dots x_{c_{\max}}, y_{c_{\min}} \dots y_{c_{\max}}, S_{\min} \dots S_{\max}, \phi_{\min} \dots \phi_{\max})$ , inicializada com zero.
-

## Generalização da Transformada de Hough

```
9: for cada ponto  $(x, y)$  da borda do
10:     Recuperar todos os pares  $(r, \alpha)$  da tabela-R cuja entrada  $\theta_i$ 
11:     esteja próxima do valor do gradiente no ponto  $(x, y)$ 
12:     Para cada par  $(r, \alpha)$ , calcular os pontos de referência candidatos:
13:     for  $\phi = \phi_{\min}$  até  $\phi = \phi_{\max}$  do
14:         for  $S = S_{\min}$  até  $S_{\max}$  do
15:              $x_c = x + r S \cos(\alpha + \phi)$ 
16:              $y_c = y + r S \sin(\alpha + \phi)$ 
17:              $M(x_c, y_c, S, \phi) = M(x_c, y_c, S, \phi) + 1$ 
18:         end for
19:     end for
20: end for
21: Se  $M(x_c, y_c, S, \phi) > T$ , o contorno do objeto é localizado no ponto  $(x_c, y_c)$ , com
    respeito ao fator de escala  $S$  e ao ângulo de rotação  $\phi$ .
```

# Limiarização

- A *limiarização* é uma das técnicas mais simples de segmentação e consiste na classificação dos pixels de uma imagem de acordo com a especificação de um ou mais limiares.
- Seja o histograma de níveis de cinza, mostrado na figura a seguir, correspondente a uma imagem  $f(x, y)$ , de maneira que os pixels dos objetos e os do fundo tenham seus níveis de cinza separados em dois grupos dominantes.



- Uma maneira óbvia de extrair os objetos do fundo é por meio da seleção de um limiar  $T$  que separe os dois grupos.



## Limiarização

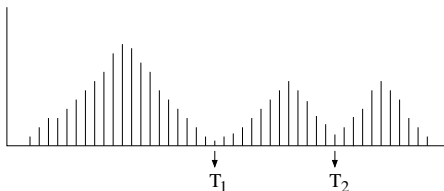
- Então, cada ponto  $(x, y)$  tal que  $f(x, y) > T$  é denominado um ponto do objeto; caso contrário, o ponto é denominado um ponto do fundo.
- A imagem limiarizada  $g(x, y)$  pode ser definida como

$$g(x, y) = \begin{cases} 0, & \text{se } f(x, y) \leq T \\ 1, & \text{se } f(x, y) > T \end{cases} \quad (34)$$

- Portanto, os pixels rotulados como 1 (ou qualquer outro nível de cinza conveniente) correspondem aos objetos, enquanto que aqueles rotulados como 0 correspondem ao fundo.
- Nesse caso, a limiarização é denominada *binarização*, pois a imagem resultante possui apenas dois valores de intensidade, 0 (preto) ou 1 (branco).

## Limiarização

- A figura a seguir mostra um caso ligeiramente mais geral dessa abordagem, em que três grupos dominantes caracterizam o histograma da imagem (por exemplo, dois tipos de objetos iluminados sobre um fundo escuro).



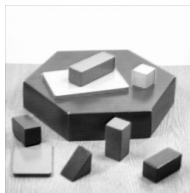
- A limiarização da imagem pode ser definida como

$$g(x, y) = \begin{cases} l_1, & \text{se } f(x, y) \leq T_1 \\ l_2, & \text{se } T_1 < f(x, y) \leq T_2 \\ l_3, & \text{se } f(x, y) > T_2 \end{cases} \quad (35)$$

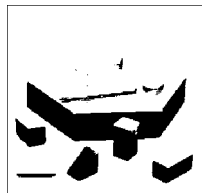
tal que, para cada intervalo, é especificado um nível de cinza correspondente.

# Limiarização

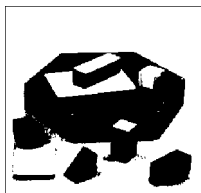
- A seleção correta do valor de limiar é crucial para que o processo de segmentação baseada na limiarização produza bons resultados.
- A figura a seguir ilustra a segmentação de uma imagem com o uso de diferentes valores de limiar.



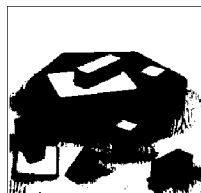
(a) imagem original



(b)  $T = 108$



(c)  $T = 179$

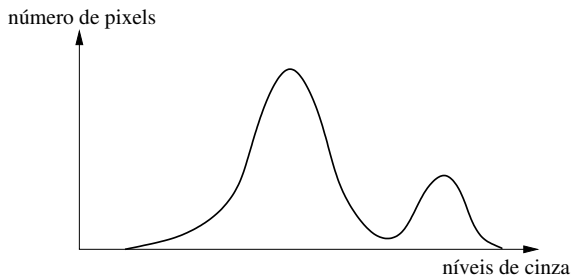


(d)  $T = 213$

- *Limiarização global*: a utilização de um único valor de limiar para segmentar toda a imagem não é em geral adequada, pois as imagens podem conter variações nos níveis de cinza dos objetos e do fundo por causa da iluminação não-uniforme, ruído, parâmetros do dispositivo de aquisição não-uniformes ou outros fatores.
- *Limiarização local*: valores de limiar podem variar sobre a imagem como uma função de suas características locais.

## Limiarização Global

- Como mencionado anteriormente, a maneira mais direta de selecionar um valor global de limiar é a partir da distribuição das intensidades dos pixels na imagem.
- Para uma imagem consistindo de objetos com níveis de cinza aproximadamente iguais, diferindo do nível de cinza do fundo, o histograma possuirá dois picos distintos, um formado pelos pixels dos objetos e outro pelos pixels do fundo.
- Um histograma com tal distribuição é chamado *bimodal*.



(a) histograma bimodal

- Intuitivamente, o valor do limiar corresponde ao nível de cinza que melhor separa os dois picos.
- Vários métodos têm sido propostos para encontrar os pontos de máximo (picos) e, então, selecionar o valor de limiar como o ponto de mínimo (vale) entre eles.
- Há várias desvantagens em selecionar o limiar global por meio da distribuição das intensidades dos pixels.
  - ▶ Nem sempre as intensidades dos objetos e do fundo da imagem são bem distintas em virtude, por exemplo, da ocorrência de baixo contraste ou ruído.
  - ▶ Além disso, a determinação de picos e vales não é um problema trivial. Por exemplo, muitos pontos de mínimo ou máximo locais podem existir na imagem.

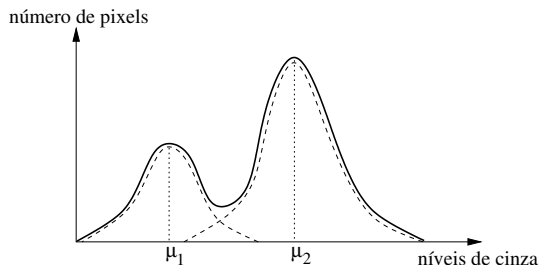
- Uma abordagem proposta por Mason et al. (1975) consiste em modificar o histograma de acordo com uma função do valor de contraste local.
- O histograma das intensidades dos pixels é ponderado pelo inverso dos seus valores de gradiente.
- As intensidades de regiões de pixels com valores de gradiente baixos são mais influenciadas do que as intensidades com valores de gradiente mais altos, desse modo, os picos do histograma causados por tais regiões podem ser removidos, facilitando a determinação do limiar entre eles.

- Uma técnica similar é ponderar as intensidades de pixels com valores de gradiente e, então, escolher o limiar no pico do histograma, correspondendo à transição entre as regiões.
- Esse método requer a detecção de um ponto de máximo, o qual é normalmente mais fácil do que a detecção de vales entre dois picos. Outra abordagem utiliza o operador Laplaciano, que é um operador de diferenciação de segunda ordem.
- O Laplaciano resulta em valores altos para as transições das regiões em bordas.
- Caso alguma propriedade da imagem a ser segmentada seja previamente conhecida, a seleção do limiar pode ser simplificada, já que o valor do limiar é escolhido para assegurar que essa propriedade seja satisfeita.
- Conhecendo-se a percentagem  $p$  de pixels do objeto na imagem, essa informação pode ser utilizada para escolher o limiar  $T$ , tal que o valor  $p$  da área da imagem tenha níveis de cinza menores que  $T$  e o restante dos níveis de cinza maiores que  $T$ .



## Limiarização Global

- Algumas técnicas determinam o valor de limiar por meio da otimização de certas medidas de separação entre as classes de objetos na imagem.
- As distribuições de intensidade dos pixels podem ser modeladas como funções de densidade de probabilidades, por exemplo, a função Gaussiana.



- Para cada distribuição, a média e a variância são calculadas, e o limiar é escolhido tal que o erro definido por essas funções de probabilidade seja minimizado.

- Métodos para escolha de limiar que resultam em segmentação com erro mínimo entre os picos de duas ou mais funções de probabilidade com distribuição normal foram propostos por vários autores.
- Uma dificuldade com esses métodos é a estimativa de parâmetros das distribuições normais, bem como a incerteza inerente ao se considerar a função de probabilidade como uma distribuição normal.
- Um método para selecionar iterativamente o valor de limiar baseado na combinação de modelos de duas distribuições Gaussianas foi proposto por Ridler e Calvard (1978).
  - ▶ Após uma estimativa inicial para o limiar, o método refina o cálculo do limiar baseado na média das distribuições dos níveis de cinza do objeto ( $\mu_1$ ) e do fundo ( $\mu_2$ ), até que esses valores não sofram alterações em iterações sucessivas.
  - ▶ Na prática, as iterações terminam quando a diferença  $|T_{i+1} - T_i|$  entre os limiares  $T$  nas iterações  $i$  e  $i + 1$  torna-se suficientemente pequena.

- O algoritmo a seguir apresenta os principais passos do método de Ridler e Calvard (1978).

---

### Algoritmo 5 Limiarização global iterativa

---

- 1: Estimar um valor inicial,  $T = T_0$ , para o limiar, por exemplo, igual à intensidade média da imagem.
  - 2: Particionar a imagem em duas regiões,  $R_1$  e  $R_2$ , com o limiar  $T$ , tal que todos os pixels com intensidade menor que  $T$  sejam atribuídos à região  $R_1$  e todos os outros sejam atribuídos à região  $R_2$ .
  - 3: Na iteração  $i$ , calcular valores de intensidade média  $\mu_1$  e  $\mu_2$  associados às partições  $R_1$  e  $R_2$ .
  - 4: Calcular um novo limiar como  $T = T_{i+1} = \frac{\mu_1^i + \mu_2^i}{2}$ .
  - 5: Repetir os passos (2) a (4) até que o limiar  $T$  não sofra alteração significativa em iterações sucessivas, ou seja,  $T_{i+1} = T_i$ .
-

- Trussel (1979), baseando-se no algoritmo de Ridler e Calvard, propôs que o novo valor de limiar  $T_{i+1}$  na iteração  $i + 1$  fosse calculado como

$$T_{i+1} = \frac{\mu_1^i + \mu_2^i}{2} = \frac{1}{2} \left( \frac{\sum_{i=0}^{T_i} i p_i}{\sum_{i=0}^{T_i} p_i} + \frac{\sum_{i=T_{i+1}}^{L-1} i p_i}{\sum_{i=T_{i+1}}^{L-1} p_i} \right)$$

em que  $\mu_1^i$  e  $\mu_2^i$  são as intensidades médias dos níveis de cinza abaixo e acima, respectivamente, do limiar  $T_i$  e  $p_i$  é a probabilidade de se encontrar um número de pixel  $n_i$  com nível de cinza  $i$  em uma imagem com  $n$  pixels, ou seja,  $p_i = n_i/n$ .

- Lloyd (1985) propôs uma modificação no método de Ridler e Calvard para seleção iterativa do limiar.
- O passo (4) do Algoritmo 5 é substituído por

$$T = \frac{1}{2} (\mu_1 + \mu_2) + \frac{\sigma^2}{(\mu_1 - \mu_2)} \log \frac{n_2}{n_1}$$

em que  $\mu_1$  e  $\mu_2$  são os níveis de cinza médios associados aos números de pixels  $n_1$  e  $n_2$  das regiões  $R_1$  e  $R_2$ , respectivamente.

- A variância  $\sigma^2$  pode ser estimada a partir dos dados, em particular,  $\sigma^2 = \sigma_1^2 = \sigma_2^2$ .

- O método de Otsu (1979) é baseado no fato de que o histograma de uma imagem é composto de duas classes, os pixels do objeto e os pixels do fundo.
  - ▶ Cada classe possui suas próprias características, ou seja, sua média e desvio padrão.
  - ▶ A variância  $\sigma_T^2$  e a média  $\mu_T$  globais da imagem são calculadas.
  - ▶ O método procura maximizar a razão  $\eta(T)$  da variância entre as classes  $\sigma_B^2$  em relação à variância total, para todos os valores possíveis de limiar  $T$ , sendo

$$\eta(T) = \frac{\sigma_B^2}{\sigma_T^2} \quad (36)$$

tal que

$$\begin{aligned} \sigma_T^2 &= \sum_{i=0}^{L-1} (i - \mu_T)^2 p_i & \mu_T &= \sum_{i=0}^{L-1} i p_i & \sigma_B^2 &= \omega_1 \omega_2 (\mu_1 - \mu_2)^2 & \omega_1 &= \sum_{i=0}^T p_i \\ \omega_2 &= 1 - \omega_1 & \mu_1 &= \frac{\mu_T}{\omega_1} & \mu_2 &= \frac{\mu_T - \mu_S}{\omega_2} & \mu_S &= \sum_{i=0}^T i p_i & \sum_{i=0}^{L-1} p_i &= 1, & p_i &= \frac{n_i}{n} \end{aligned}$$

em que  $n_i$  é o número de pixels com nível de cinza  $i$ ,  $n$  é o número total de pixels da imagem,  $p_i$  representa a probabilidade do nível de cinza  $i$  ser encontrado na imagem e  $L$  é o número de níveis de cinza da imagem.

- Conforme equação 36, a razão  $\eta(T)$  é calculada para todos os valores possíveis de  $T$ .
- O limiar ótimo pode ser determinado como

$$T = \arg \max \eta(T)$$

- O método de Otsu possui bom desempenho em imagens com maior variância de intensidade.
- Uma desvantagem é que o método assume que o histograma da imagem seja bimodal.

- Um método proposto por Yanni e Horne (1994) assume que dois picos distintos  $g_{\text{pico1}}$  e  $g_{\text{pico2}}$  nos níveis de cinza são identificáveis na distribuição de probabilidade.
- Um ponto intermediário é inicialmente estabelecido como  $\hat{g}^* = (g_{\text{max}} + g_{\text{min}})/2$ , em que  $g_{\text{min}}$  é o nível de cinza mais baixo e  $g_{\text{max}}$  o nível de cinza mais alto diferente de zero.
- Esse ponto intermediário é atualizado pela média dos dois picos localizados à esquerda e à direita, ou seja,  $\hat{g}^* = (g_{\text{pico1}} + g_{\text{pico2}})/2$ .
- O limiar é então

$$T = (g_{\text{max}} - g_{\text{min}}) \sum_{g=g_{\text{min}}}^{\hat{g}^*} p_g$$



- Um outro enfoque similar ao de Otsu (1979) para selecionar valores de limiar emprega a entropia.
- A maximização da entropia da imagem limiarizada é interpretada como um indicativo de transferência máxima de informação.
- No método de Pun (1980), procura-se a maximização da entropia da imagem, dividindo-se o seu histograma em duas distribuições de probabilidade representando as duas classes presentes na imagem, ou seja, fundo e objeto.

- As entropias  $H_b(T)$  e  $H_w(T)$  associadas aos pixels do fundo e dos objetos, respectivamente, são dadas por

$$H_b(T) = - \sum_{i=0}^T p_i \log p_i \qquad H_w(T) = - \sum_{i=T+1}^{L-1} p_i \log p_i$$

sendo  $\sum_{i=0}^{L-1} p_i = 1$ ,  $p_i = \frac{n_i}{n}$ , tal que  $n_i$  é o número de pixels com nível de cinza  $i$ ,  $p_i$  representa a probabilidade do nível de cinza  $i$  ser encontrado na imagem,  $n$  é o número de pixels da imagem e  $L$  é o número de níveis de cinza da imagem.

- A escala de níveis de cinza é analisada de forma a encontrar um valor de limiar  $T$ , tal que  $T = H_b(T) + H_w(T)$  seja máximo, procurando-se uma boa separação entre o objeto e o fundo da imagem, ou seja

$$T = \arg \max [H_b(T) + H_w(T)] \qquad (37)$$

- Kapur et al. (1985) propõem que as entropias  $H_b(T)$  e  $H_w(T)$  sejam calculadas como

$$H_b(T) = - \sum_{i=0}^T \frac{p_i}{p_1(T)} \log \frac{p_i}{p_1(T)}$$
$$H_w(T) = - \sum_{i=T+1}^{L-1} \frac{p_i}{p_2(T)} \log \frac{p_i}{p_2(T)}$$

sendo

$$p_1(T) = \sum_{i=0}^T p_i \quad p_2(T) = \sum_{i=T+1}^{L-1} p_i \quad p_1(T) + p_2(T) = 1$$

- A partir das duas entropias  $H_b(T)$  e  $H_w(T)$ , realiza-se um processo de seleção exaustiva do limiar na escala de cinza, com o objetivo de encontrar o valor que maximize a soma das entropias, como no método definido por Pun, dado pela equação 37.

- Tsai (1985) utilizou a preservação de momentos para obter um valor de limiar sem a necessidade de iteração.
- O momento de ordem  $k$ , denotado  $m_k$ , de uma imagem  $\mathbf{f}$  com  $L$  níveis de cinza é definido como

$$m_k = \sum_{i=0}^{L-1} p_i i^k \quad k = 0, 1, 2, 3 \quad (38)$$

sendo  $p_i = n_i/n$ , tal que  $n_i$  é o número de pixels com nível de cinza  $i$ ,  $n$  é o número de pixels da imagem,  $L$  é o número de níveis de cinza da imagem e  $m_0 = 1$ .

- A imagem  $\mathbf{f}$  pode ser vista como uma versão suavizada de uma imagem ideal  $\mathbf{g}$  que apresenta histograma bimodal, cujos picos estão em  $h_1$  e  $h_2$ ,  $h_1 < h_2$ .
- O método seleciona um limiar  $T$  tal que, se todas as intensidades abaixo do limiar na imagem  $\mathbf{f}$  forem trocadas por  $h_1$  e todas as intensidades acima do limiar forem trocadas por  $h_2$ , então os três primeiros momentos de  $\mathbf{f}$  serão preservados na imagem  $\mathbf{g}$ .

- Considerando  $p_1$  e  $p_2$  as frações dos pixels que estão abaixo e acima do limiar na imagem  $\mathbf{f}$ , então os três primeiros momentos da imagem binária  $\mathbf{g}$  são dados por

$$m_k = p_1 h_1^k + p_2 h_2^k \quad k = 0, 1, 2, 3 \quad (39)$$

- Preservando os momentos e utilizando o fato de que  $p_1 + p_2 = 1 = m_0$ , um conjunto de quatro equações contendo os parâmetros  $p_1$ ,  $p_2$ ,  $h_1$  e  $h_2$  é definido como

$$p_1 h_1 + p_2 h_2 = m_1$$

$$p_1 h_1^2 + p_2 h_2^2 = m_2$$

$$p_1 h_1^3 + p_2 h_2^3 = m_3$$

$$p_1 + p_2 = 1$$

- No caso de um histograma bimodal, as equações são resolvidas por meio das expressões

$$h_1 = \frac{-c_1 - \sqrt{c_1^2 - 4c_0}}{2} \quad h_2 = \frac{-c_1 + \sqrt{c_1^2 - 4c_0}}{2} \quad p_1 = \frac{\begin{vmatrix} 1 & 1 \\ m_1 & h_2 \end{vmatrix}}{\begin{vmatrix} 1 & 1 \\ h_1 & h_2 \end{vmatrix}}$$

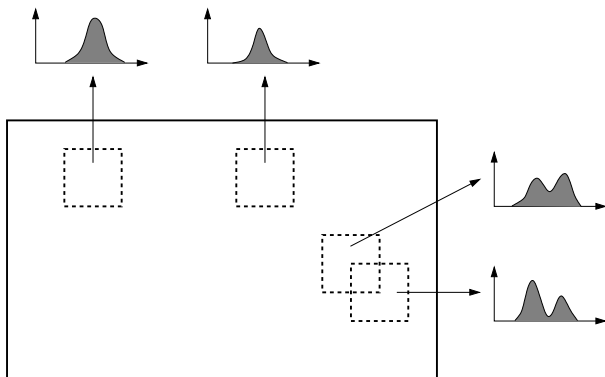
em que

$$c_0 = \frac{\begin{vmatrix} -m_2 & m_1 \\ -m_3 & m_2 \end{vmatrix}}{\begin{vmatrix} m_0 & m_1 \\ m_1 & m_2 \end{vmatrix}} \quad c_1 = \frac{\begin{vmatrix} m_0 & -m_2 \\ m_1 & -m_3 \end{vmatrix}}{\begin{vmatrix} m_0 & m_1 \\ m_1 & m_2 \end{vmatrix}}$$

- O valor do limiar  $T$  é escolhido como o nível de cinza mais próximo da fração  $p_1$  no histograma da imagem  $f$ .

## Limiarização Local

- Em casos nos quais a utilização de um único limiar para segmentar toda a imagem não produza bons resultados, a *limiarização local* pode ser uma alternativa mais apropriada.
- Uma forma comum de realizar a limiarização local é analisar as intensidades dos pixels em uma região da imagem para determinar limiares locais. Esse processo é ilustrado na figura a seguir.



## Limiarização Local

- Algumas medidas estatísticas simples para calcular um limiar local são a média dos valores de intensidade em uma vizinhança local da imagem

$$T = \text{média}_v(p) \quad (40)$$

a mediana dos valores

$$T = \text{mediana}_v(p) \quad (41)$$

ou a média dos valores mínimo e máximo

$$T = \frac{\min_v(p) + \max_v(p)}{2} \quad (42)$$

em que  $v$  é uma vizinhança local ao ponto  $p$  na imagem.

- Um problema crítico encontrado nas técnicas de limiarização local é a escolha do tamanho da janela.
- O tamanho selecionado deve ser suficientemente grande para garantir uma boa estimativa do valor da média, entretanto, sem sofrer distorções por causa da não-uniformidade do fundo.



- Yasuda et al. (1980) propuseram um método que expande a escala de níveis de cinza da imagem, seguido de uma suavização não-linear que preserve bordas.
- A suavização consiste em substituir cada pixel pela média de seus oito vizinhos, dado que o intervalo do pixel (definido como a diferença entre os valores máximo e mínimo locais) está abaixo de um limiar  $T_1$ .
- Um limiar adaptativo é aplicado tal que o valor de cada pixel seja atribuído como fundo se o intervalo local estiver abaixo de um limiar  $T_2$  ou se o valor do pixel estiver acima de uma média local, ambos calculados sobre uma vizinhança de  $n \times n$  pixels. Caso contrário, a escala de nível de cinza deve ser expandida convenientemente.
- Finalmente, a imagem é binarizada atribuindo-se um pixel como objeto se seu valor mínimo em uma vizinhança de  $3 \times 3$  pixels estiver abaixo de  $T_3$  ou se sua variância local estiver acima de  $T_4$ .
- Kamel e Zhao (1989) sugerem os valores  $T_1 = 50$ ,  $n = 16$ ,  $T_2 = 16$ ,  $T_3 = 128$  e  $T_4 = 35$ .

- No método de Bernsen (1986), o limiar para cada pixel  $(x, y)$  é calculado como

$$T(x, y) = \frac{z_{\min} + z_{\max}}{2} \quad (43)$$

em que  $z_{\min}$  e  $z_{\max}$  são os valores de níveis de cinza mínimo e máximo em uma vizinhança de  $n \times n$  pixels centrada em  $(x, y)$ .

## Limiarização Local

- No método de Niblack (1986), o valor do limiar em um pixel  $(x, y)$  é baseado na média local e no desvio padrão calculados sobre uma vizinhança de tamanho  $n \times n$  pixels, expresso como

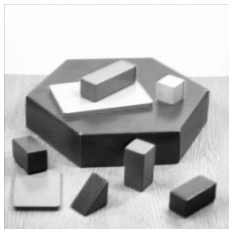
$$T(x, y) = \mu(x, y) + k \sigma(x, y) \quad (44)$$

em que  $\mu(x, y)$  e  $\sigma(x, y)$  são a média e o desvio padrão dos valores sobre a vizinhança, respectivamente.

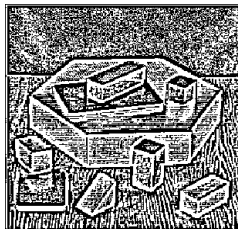
- O valor para a constante  $k$  é ajustado conforme o tipo de aplicação ou característica da imagem sendo processada.
- A escolha desse valor proporciona um equilíbrio entre supressão de ruído e preservação de detalhes, além de atribuir menor ou maior presença de contorno do objeto na imagem limiarizada.
- O tamanho da vizinhança deveria ser suficientemente pequeno para preservar detalhes locais, mas, ao mesmo tempo, suficientemente grande para suprimir ruído.
- Trier e Jain (1995) sugerem que resultados satisfatórios podem ser obtidos com uma janela de tamanho  $n = 15$  e  $k = -0.2$ .

## Limiarização Local

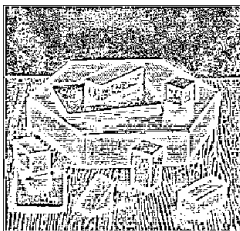
- Exemplo de imagem limiarizada pelo método de Niblack com diferentes valores de ajuste, para um mesmo tamanho de janela.



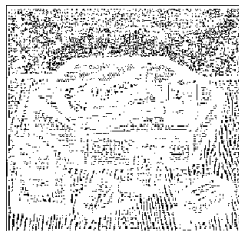
(a) imagem original



(b)  $n = 25, k = 0.1$



(c)  $n = 25, k = 0.5$



(d)  $n = 25, k = 0.8$

## Limiarização Local

- O método de Sauvola e Pietaksinen (2000) procura melhorar os resultados do método de Niblack, particularmente para imagens de documentos apresentando má iluminação.
- O limiar se adapta de acordo com a média local e o desvio padrão sobre uma janela de tamanho  $n \times n$  pixels.
- O limiar em um pixel  $(x, y)$  é calculado como

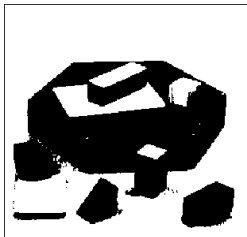
$$T(x, y) = \mu(x, y) \left[ 1 + k \left( \frac{\sigma(x, y)}{R} - 1 \right) \right] \quad (45)$$

em que  $\mu(x, y)$  e  $\sigma(x, y)$  são definidos como em Niblack.

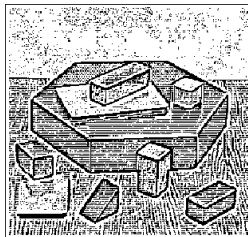
- Sauvola e Pietaksinen sugerem valores de  $k = 0.5$  e  $R = 128$  para as imagens utilizadas em seus experimentos.
- Dessa forma, a contribuição do desvio padrão torna-se adaptativa.

# Limiarização Global × Local

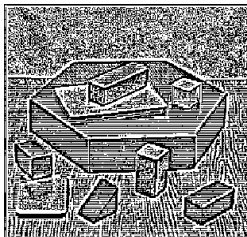
- Comparação entre diferentes técnicas de limiarização.



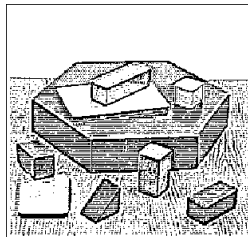
(a) global ( $T = 179$ )



(b) Bernsen ( $n = 9$ )



(c) Niblack ( $n = 9$ ,  $k = 0.01$ )



(d) Sauvola e Pietakinen ( $n = 9$ ,  
 $k = 0.5$ ,  $R = 128$ )

# Segmentação de Regiões

- Em contraste com os métodos de segmentação descritos anteriormente, os métodos a seguir detectam regiões diretamente nas imagens, ao invés de encontrar as bordas que delimitam as regiões.
- Pontos apresentando propriedades similares são agrupados para formar uma região.
- Diversas propriedades têm sido propostas para caracterizar uma região, tais como intensidade de cinza, cor, informação semântica ou textura.

# Segmentação de Regiões

- Seja  $R$  a região definida pela própria imagem de entrada. Pode-se imaginar a segmentação como o processo de particionar  $R$  em  $n$  regiões  $R_1, R_2, \dots, R_n$ , tal que:

(a)  $\bigcup_{i=1}^n R_i = R$

(b)  $R_i$  é uma função 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  $i = 1, 2, \dots, n$

(e)  $P(R_i \cup R_j) = \text{FALSO}$  para  $i \neq j$  e  $R_i$  adjacente a  $R_j$ .

em que  $P(R_i)$  é um predicado lógico sobre os pontos do conjunto  $R_i$  e  $\emptyset$  é o conjunto vazio.

- O predicado é utilizado como uma medida de similaridade para agrupar os pixels em uma região.

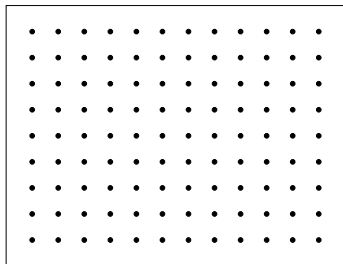


# Segmentação de Regiões

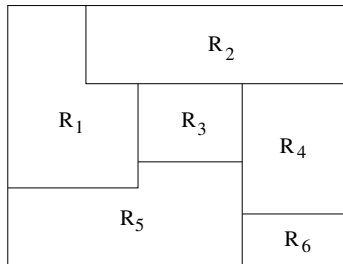
- A condição (a) indica que cada pixel deve pertencer a uma região da imagem.
- A condição (b) requer que os pixels de uma região satisfaçam critérios de conectividade.
- A condição (c) estabelece que as regiões devem ser disjuntas.
- A condição (d) refere-se às propriedades que devem ser satisfeitas pelos pixels em uma região segmentada. Por exemplo,  $P(R_i) = \text{VERDADEIRO}$  se todos os pixels da região  $R_i$  possuírem a mesma intensidade.
- Finalmente, a condição (e) estabelece que as regiões adjacentes  $R_i$  e  $R_j$  são diferentes em relação ao predicado  $P$ .

# Segmentação de Regiões

- O processo de segmentação de imagens em regiões homogêneas disjuntas é ilustrado a seguir.



(a) imagem de entrada



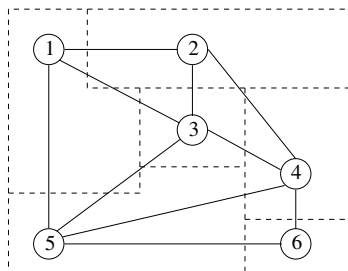
(b) imagem segmentada em regiões

## Segmentação de Regiões

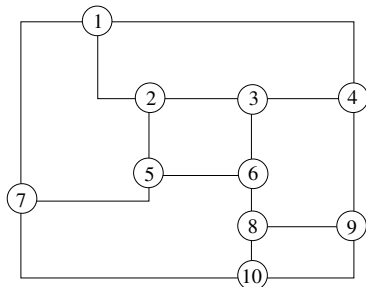
- A relação de adjacência entre as regiões de uma imagem pode ser representada por um *grafo de adjacência de regiões*.
- As regiões da imagem são representadas por um conjunto de nós  $V = \{V_1, V_2, \dots, V_m\}$ , tal que a região  $R_i$  na imagem e suas propriedades são associadas ao nó  $V_i$ .
- A aresta  $E_{ij}$  entre os nós  $V_i$  e  $V_j$  representa a adjacência entre as regiões  $R_i$  e  $R_j$ .
- Conforme já descrito, duas regiões  $R_i$  e  $R_j$  são adjacentes se existir um pixel na região  $R_i$  que seja adjacente a um pixel na região  $R_j$  por meio de vizinhança-4 ou vizinhança-8.
- A relação de adjacência é reflexiva e simétrica, mas não necessariamente transitiva.
- O *grafo dual de adjacência de regiões* também pode ser utilizado em algumas situações, em que os nós representam as fronteiras e as arestas representam as regiões que são separadas pelas fronteiras.

# Segmentação de Regiões

- Exemplo de representação da imagem mostrada na figura anterior (b) por meio de um grafo de adjacência de regiões e seu dual.



(a) grafo de adjacência de regiões



(b) grafo dual

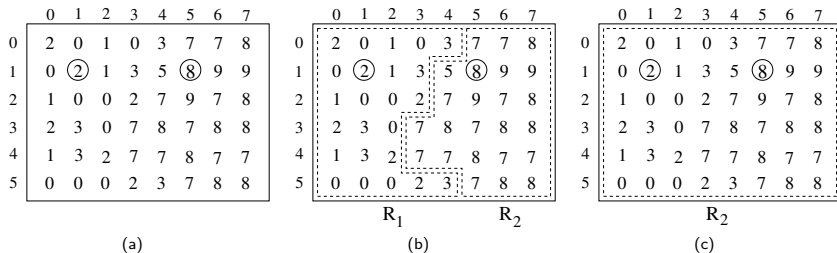
- Os principais métodos de segmentação baseada em regiões podem ser classificados em *crescimento de regiões*, *divisão de regiões*, *divisão e fusão de regiões* e *divisor de águas*, discutidos a seguir.

# Crescimento de Regiões

- A segmentação baseada em *crescimento de regiões* é um procedimento que agrega pixels com propriedades similares em regiões.
- Uma abordagem simples é iniciar com um conjunto de pixels denominados *sementes* e, a partir deles, crescer as regiões anexando a cada ponto semente outros pixels que possuam propriedades similares.
- Os pixels sementes podem ser escolhidos de maneira aleatória, determinística ou selecionados pelo usuário.

## Crescimento de Regiões

- A figura a seguir ilustra o procedimento de crescimento de regiões, em que os valores das células na matriz representam os níveis de cinza dos pixels.



**Figura:** Exemplo de crescimento de regiões. (a) imagem original; (b) segmentação utilizando uma diferença absoluta menor que 4 entre os níveis de cinza; (c) segmentação utilizando uma diferença absoluta menor que 8.

## Crescimento de Regiões

- Sejam os pontos de coordenadas  $(1, 1)$  e  $(5, 1)$  os pixels sementes, conforme marcados na figura (a).
- A utilização de dois pontos iniciais resulta em uma segmentação consistindo em, no máximo, duas regiões:  $R_1$  associada à semente  $(1, 1)$  e  $R_2$  associada à semente  $(5, 1)$ .
- O predicado  $P$  a ser usado para agregar um pixel em uma das regiões verifica se a diferença absoluta entre os níveis de cinza desse pixel e o da semente é menor que um dado limiar  $T$ , ou seja

$$P(R) = \begin{cases} \text{VERDADEIRO}, & \text{se } |f(x, y) - f(r, s)| \leq T \\ \text{FALSO}, & \text{caso contrário} \end{cases}$$

em que  $f(r, s)$  representa o pixel semente e  $f(x, y)$  representa os pixels conectados ao pixel semente por vizinhança-8.

- Qualquer pixel que satisfaça essa propriedade simultaneamente para ambas é (arbitrariamente) atribuído à região  $R_1$ .

- A figura (b) mostra o resultado obtido usando-se  $T = 4$ . Nesse caso, a segmentação consiste em duas regiões, tal que os pontos pertencentes à região  $R_1$  são denotados por  $a$ , enquanto os pertencentes à região  $R_2$  são denotados por  $b$ .
- Deve-se notar que qualquer ponto inicial, em qualquer uma das regiões resultantes, levaria ao mesmo resultado.
- Por outro lado, a escolha de  $T = 8$  resultaria em uma única região, como mostrado na figura (c).



- No critério de similaridade descrito por Haralick e Shapiro (1992), uma região é definida como um conjunto de pixels conexos pela média e variância do conjunto. Um pixel é agregado à região se o critério dado pela equação 46 for satisfeito, tal que

$$T = \sqrt{\frac{(N-1)N}{N+1} \frac{(z - \mu_R)^2}{\sigma_R^2}} \quad (46)$$

em que  $N$  é o número de pixels da região,  $z$  é o valor do pixel sendo testado,

$$\mu_R = \frac{1}{N} \sum_{(x,y) \in R} f(x,y) \text{ e } \sigma_R^2 = \frac{1}{N} \sum_{(x,y) \in R} (f(x,y) - \mu_R)^2.$$

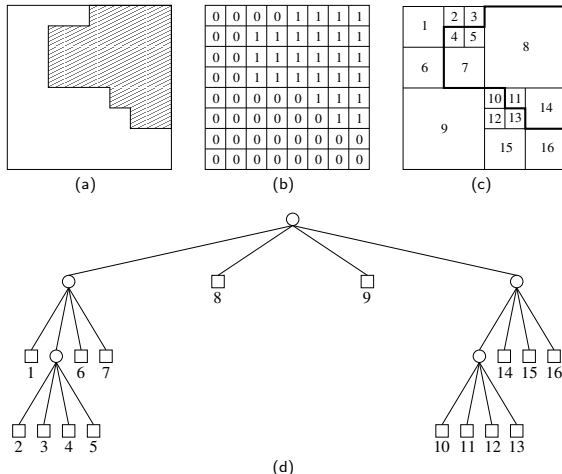
- Um valor alto de  $T$  indica que o pixel representado por  $z$  não deve ser associado à região  $R$ ; caso contrário, o pixel pode ser agregado.
- Duas dificuldades associadas ao crescimento de regiões são a seleção de sementes que representem adequadamente as regiões de interesse, bem como a seleção de propriedades apropriadas para a inclusão de pontos nas várias regiões durante o processo de crescimento.

# Divisão de Regiões

- A segmentação baseada na *divisão de regiões* inicia-se com regiões formadas por pixels da imagem e, recursivamente, subdivide as regiões não-homogêneas em áreas menores.
- Em muitos casos, a imagem inteira pode ser considerada como a região inicial.
- O processo de subdivisão termina quando todas as regiões satisfizerem o critério de similaridade.
- Uma técnica comum de subdivisão da imagem em regiões homogêneas utiliza a representação *quadtree*, que é uma estrutura hierárquica baseada na decomposição recursiva e regular da imagem em quadrantes, de maneira que, para qualquer região  $R_i$ ,  $P(R_i) = \text{VERDADEIRO}$ . Ou seja, se  $P(R) = \text{FALSO}$ , então a imagem deve ser dividida em quadrantes. Se o predicado  $P$  for FALSO para qualquer quadrante, o quadrante deve ser subdividido em subquadrantes e assim por diante.

## Segmentação de Regiões

- Seja a imagem mostrada na figura 6(a), a qual é representada por uma matriz binária de  $2^3 \times 2^3$ , mostrada na figura 6(b). As regiões homogêneas resultantes da decomposição da *quadtree* são mostradas na figura 6(c) e a *quadtree* correspondente na figura 6(d).



**Figura:** Divisão de regiões pela técnica de decomposição regular. (a) imagem original; (b)

# Divisão e União de Regiões

- Os procedimentos de crescimento e divisão de regiões podem ser combinados de modo a reunir as vantagens de ambas as técnicas.
- Os métodos de *divisão e união de regiões* são particularmente úteis para a segmentação de imagens complexas.
- Uma abordagem alternativa é subdividir a imagem em um conjunto de regiões arbitrárias e disjuntas e, então, realizar a divisão e/ou união das regiões na tentativa de satisfazer as condições requeridas.
- Caso apenas a divisão (subdivisão) fosse usada, a partição final eventualmente poderia conter regiões adjacentes apresentando propriedades similares.
- Esse problema pode ser minimizado permitindo-se a união da mesma maneira que a divisão. O intuito de satisfazer as condições já mencionadas requer que a união seja realizada em regiões adjacentes cujos pixels combinados satisfaçam o predicado  $P$ , ou seja, duas regiões adjacentes  $R_i$  e  $R_j$  são unidas apenas se  $P(R_i \cup R_j) = \text{VERDADEIRO}$ .

- Esse procedimento de divisão e união de regiões é resumido no algoritmo a seguir.

---

**Algoritmo 6** Divisão e união de regiões

---

- 1: Dividir em sub-regiões distintas qualquer região  $R_i$  em que  $P(R_i) = \text{FALSO}$ .
  - 2: Unir quaisquer regiões adjacentes  $R_i$  e  $R_j$  tal que  $P(R_i \cup R_j) = \text{VERDADEIRO}$ .
  - 3: Parar quando nenhuma divisão ou nenhuma fusão for mais possível.
-

# Divisão e União de Regiões

- Diferentes variações dessa abordagem básica são possíveis.
- Por exemplo, uma possibilidade é dividir inicialmente a imagem em blocos quadrados.
- Divisões adicionais são realizadas como foi descrito, mas as fusões são inicialmente limitadas a grupos de quatro blocos que sejam descendentes na representação *quadtrees* e que satisfaçam o predicado  $P$ .
- Quando fusões desse tipo não forem mais possíveis, o procedimento é terminado por uma união final das regiões que satisfaçam o passo mostrado na linha 2 do algoritmo de divisão e união.
- Nesse ponto, as regiões unidas podem ser de diferentes tamanhos.
- A principal vantagem dessa abordagem é a de usar a mesma *quadtrees* para a divisão e para a união até a união final.

# Segmentação por Divisor de Águas

- Para facilitar o entendimento da técnica de segmentação por *divisor de águas*, pode-se interpretar a imagem a ser segmentada como sendo uma superfície topográfica, em que as intensidades dos pixels correspondem a valores de altitude ou elevação dos pontos.
- Dessa forma, o conjunto de pixels da imagem forma uma superfície composta por vales e picos com diversas elevações.
- Um processo de imersão simula a inundação da superfície a partir de perfurações localizadas nas regiões mais baixas da superfície.
- À medida que a água penetra nessas regiões, conhecidas como mínimos regionais, vales são gradativamente inundados, formando bacias de retenção.
- Quando as águas de duas bacias vizinhas entram em contato, uma linha de contenção é criada entre essas bacias.

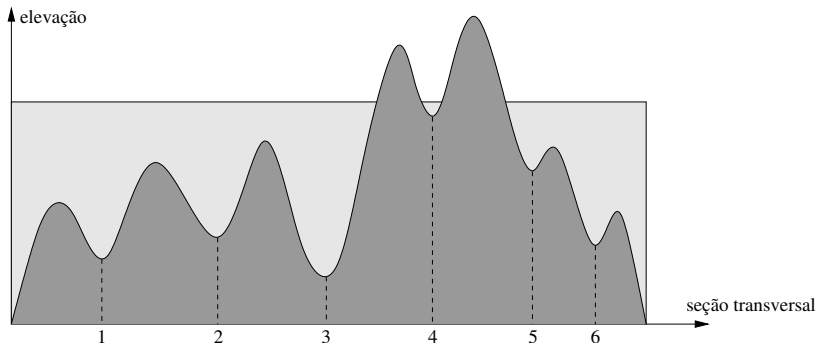
## Segmentação por Divisor de Águas

- O processo de imersão continua até que toda a superfície esteja sob a água, tal que haverá uma linha de contenção delimitando cada bacia de retenção.
- Ao final do processo, um conjunto de linhas de contenção de águas (*watershed*) é formado para evitar o transbordamento das águas entre diferentes bacias.
- Essas linhas definem o contorno dos objetos da imagem.



## Segmentação por Divisor de Águas

- O processo de simulação de imersão para uma seção transversal da imagem é mostrado na figura a seguir. A imersão é realizada a partir de mínimos regionais localizados na base das bacias.



**Figura:** Segmentação por divisor de águas.

# Segmentação por Divisor de Águas

- Vários fatores afetam a qualidade da segmentação por divisor de águas.
- O processo de imersão é sensível à presença de ruído, o qual pode ocorrer na forma de valores mínimos ou máximos na imagem, comprometendo o resultado da segmentação.
- Irregularidades nas bordas das regiões podem permitir vazamentos de água no processo de imersão e, assim como ocorre quando a imagem é corrompida por ruído, o excesso de mínimos locais pode resultar em um grande número de regiões (*sobre-segmentação*).
- Vários algoritmos sequenciais e paralelos têm sido desenvolvidos para segmentar imagens por divisor de águas.