



SEGMENTAÇÃO DE IMAGENS

Adair Santa Catarina
Curso de Ciência da Computação
Unioeste – Campus de Cascavel – PR

Agosto/2023



Introdução

A **segmentação** divide uma imagem em regiões ou objetos que a compõem.

Uma segmentação realizada com sucesso é concluída quando os objetos de interesse forem separados. Aumentar o nível de detalhamento é desnecessário e pode comprometer o processo.

O processo de segmentação pode ser melhorado quando há alguma medida de controle no ambiente imageado.

Quando não se controla o ambiente pode ser necessário usar outros tipos de sensores: térmicos, infravermelhos, raios-x, laser, ondas de rádio (radar), etc.

Introdução

A maioria dos métodos de segmentação baseiam-se em duas propriedades básicas das intensidades registradas nas imagens: **descontinuidade** e **similaridade**.

Descontinuidade: a abordagem é dividir uma imagem com base nas mudanças bruscas de intensidades, visíveis nas bordas dos objetos.

Similaridade: nesta abordagem divide-se a imagem em regiões que sejam semelhantes de acordo com um conjunto de critérios pré-definidos.

Fundamentos

Se R representa toda a região espacial de uma imagem, a segmentação deve partitionar R em n sub-regiões (R_1, R_2, \dots, R_n) , tal que:

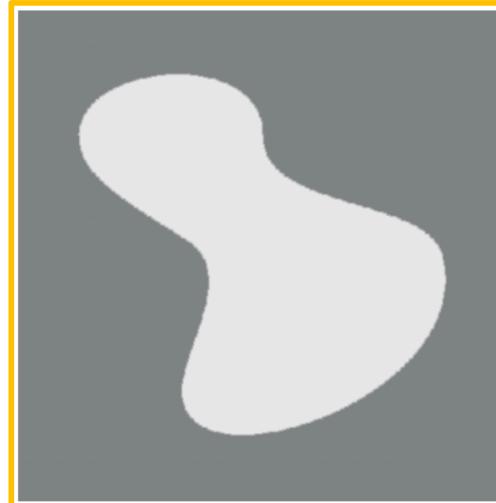
- a) $\bigcup_{i=1}^n R_i = R$
- b) R_i é um conjunto conectado, $i = 1, 2, \dots, n$
- c) $R_i \cap R_j = \emptyset$ para todo i e j , $i \neq j$
- d) $Q(R_i) = V$ para $i = 1, 2, \dots, n$ (Q = predicado)
- e) $Q(R_i \cup R_j) = F$ para quaisquer R_i e R_j adjacentes

Os conceitos de adjacência e conjunto conexo, como vistos nos Fundamentos de Imagens Digitais, também estão relacionados ao tipo de vizinhança adotada.

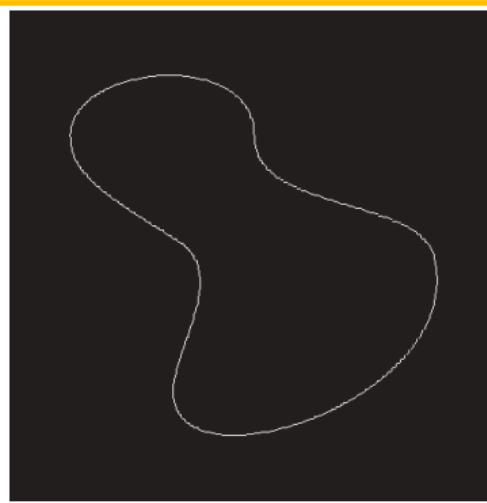
Segmentação Baseada em Descontinuidade

O pressuposto é que as fronteiras das regiões são suficientemente diferentes entre si e em relação ao fundo da imagem, permitindo identificar os limites com base na descontinuidade das intensidades.

Segmentação baseada nas bordas



Região com intensidade constante



Fronteira da região obtida por descontinuidade

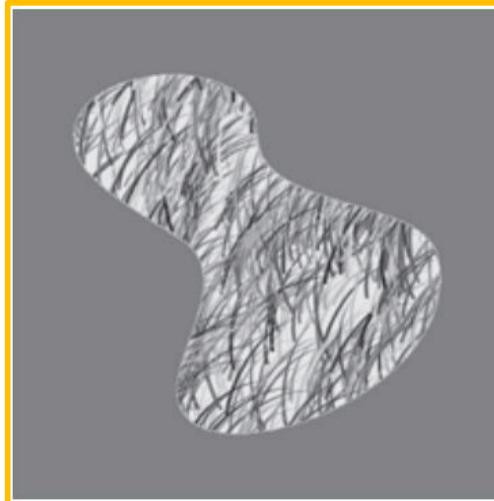


Imagem segmentada em duas regiões

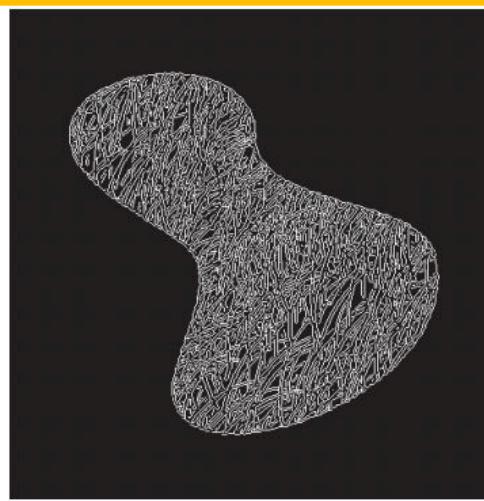
Segmentação Baseada em Similaridade

O pressuposto é que os elementos (pixels) de uma região pertencem a um mesmo conjunto e, por isso, podem ser considerados similares.

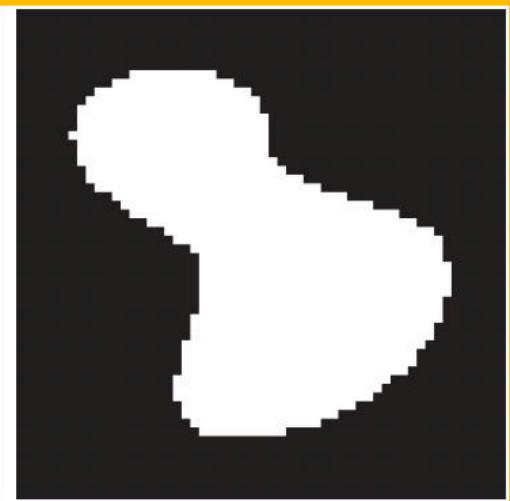
Segmentação baseada na região



A imagem contém
uma região com
textura



Cálculo das bordas
da região.
 Qual a fronteira?



Blocos 4x4.
Pixels iguais = Preto
Senão Branco.
Serrilhado!

Ponto, Linha e Borda

Pixels de borda são pixels em que a intensidade muda abruptamente. **Bordas** são conjuntos de pixels de borda conexos.

Linha é um segmento de borda em que a intensidade do fundo de cada lado da linha é diferente da intensidade da própria linha.

Ponto pode ser interpretado como uma linha de comprimento e largura igual ao pixel.

Linhos e pontos isolados dão origem às bordas em forma de telhado (*roof edges*).

Derivadas de 1^a e 2^a Ordem

Mudanças locais abruptas de intensidade podem ser detectadas pelas derivadas (diferenciação). No caso de imagens digitais pelas diferenças entre pixels.

$$\frac{\partial f}{\partial x} = f(x + 1) - f(x)$$

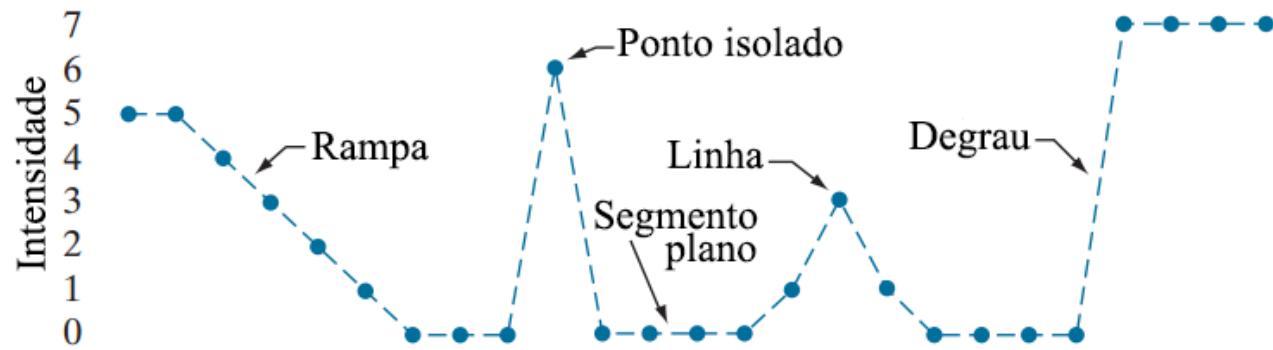
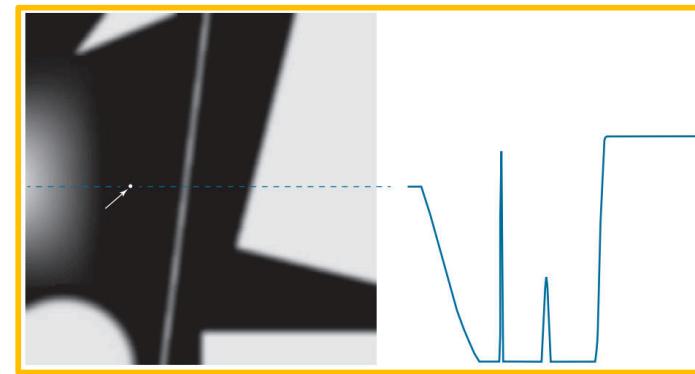
Derivada de 1^a ordem

$$\frac{\partial^2 f}{\partial x^2} = f(x + 1) + f(x - 1) - 2f(x)$$

Derivada de 2^a ordem

Os filtros de realce estudados anteriormente usavam estes mesmos princípios. A derivada de segunda ordem definia o filtro Laplaciano, enquanto a derivada de primeira ordem definia filtros baseados no gradiente (Roberts e Sobel, por exemplo).

Análise de Bordas com as Derivadas



Faixa da imagem [5 5 4 3 2 1 0 0 0 6 0 0 0 0 1 3 1 0 0 0 0 7 7 7 7 • •]

1^a derivada | -1 -1 -1 -1 -1 0 0 6 -6 0 0 0 1 2 -2 -1 0 0 0 7 0 0 0

2^a derivada | -1 0 0 0 0 1 0 6 -12 6 0 0 1 1 -4 1 1 0 0 7 -7 0 0

Detecção de Pontos Isolados

Usa-se a derivada de 2^a ordem, discretizada em um filtro laplaciano.

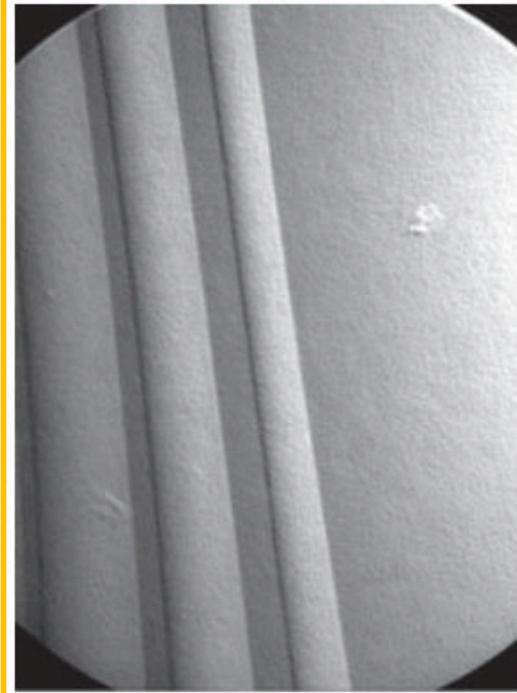
$$\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

$$\begin{aligned}\nabla^2 f(x, y) &= f(x + 1, y) + f(x - 1, y) + f(x, y + 1) \\ &\quad + f(x, y - 1) - 4f(x, y)\end{aligned}$$

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

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

Detecção de Pontos Isolados



Original

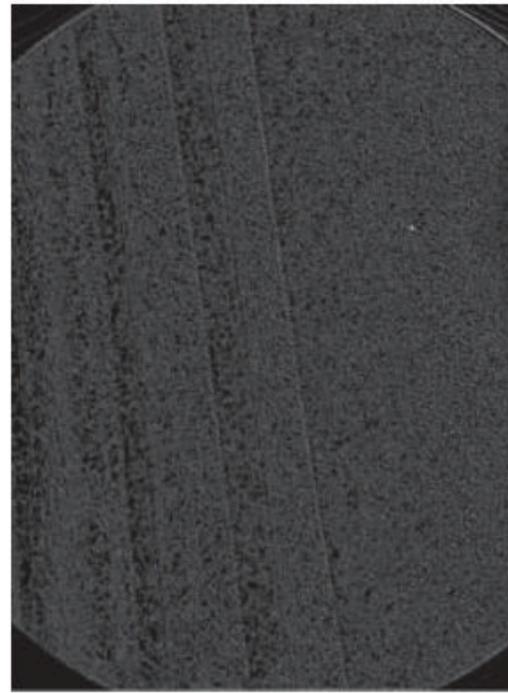
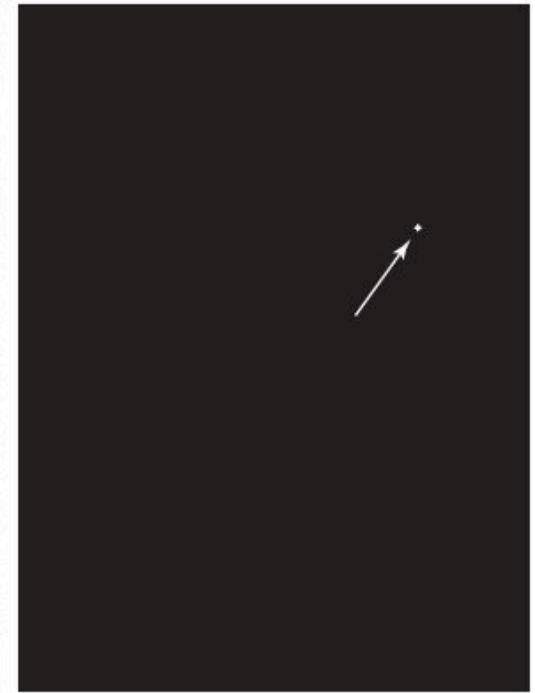


Imagen
filtrada



Após aplicação do
limiar

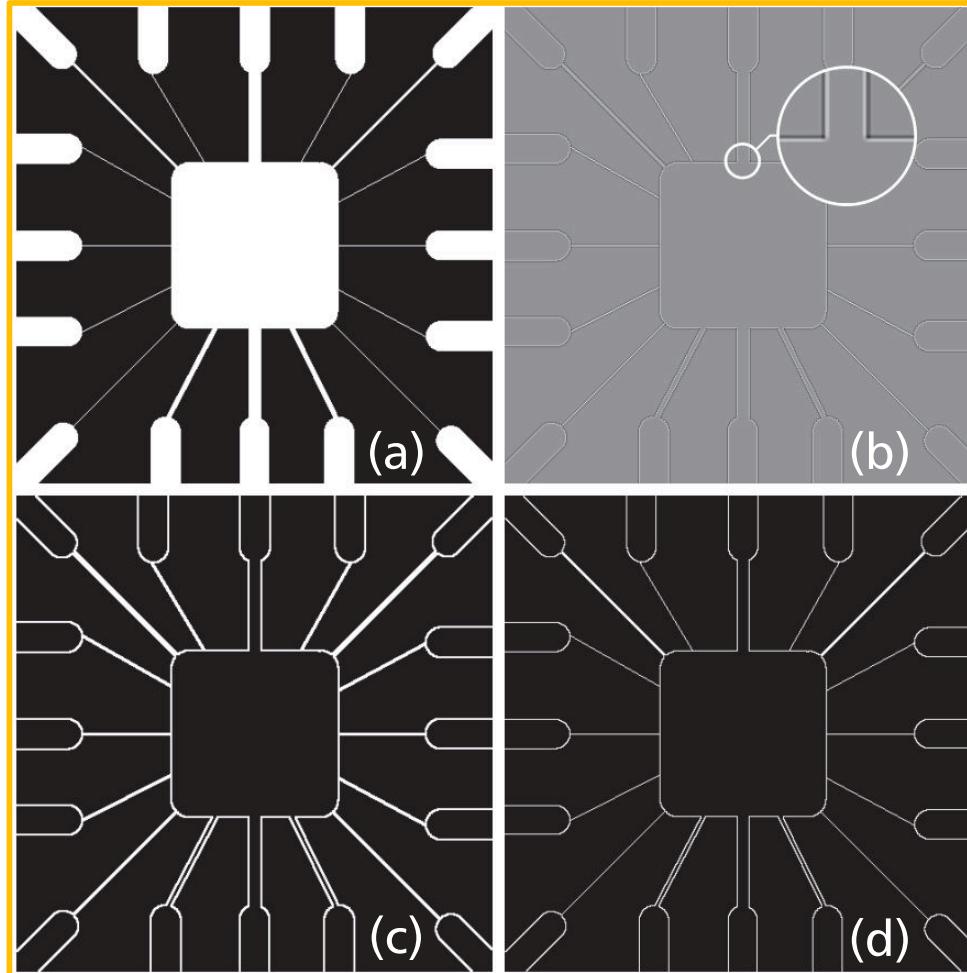
$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

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

$T = 90\%$ do maior valor encontrado na imagem filtrada

Detecção de Linhas

Também se utiliza a derivada de 2^a ordem (laplaciano).



- (a) Imagem original.
- (b) Imagem filtrada. Ajustada ao redor de 0 (cinza); valores negativos são escuros e valores positivos são claros. Observar as bordas duplas geradas pelo laplaciano.
- (c) Valor absoluto do laplaciano.
- (d) Valores positivos do laplaciano. Bordas simples, portanto mais finas.

Detecção de Linhas em Direções Específicas

Usando máscaras laplacianas direcionais podemos obter linhas horizontais, verticais e inclinadas em -45° e $+45^\circ$.

$$\begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix} \begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix}$$

Horizontal

$+45^\circ$

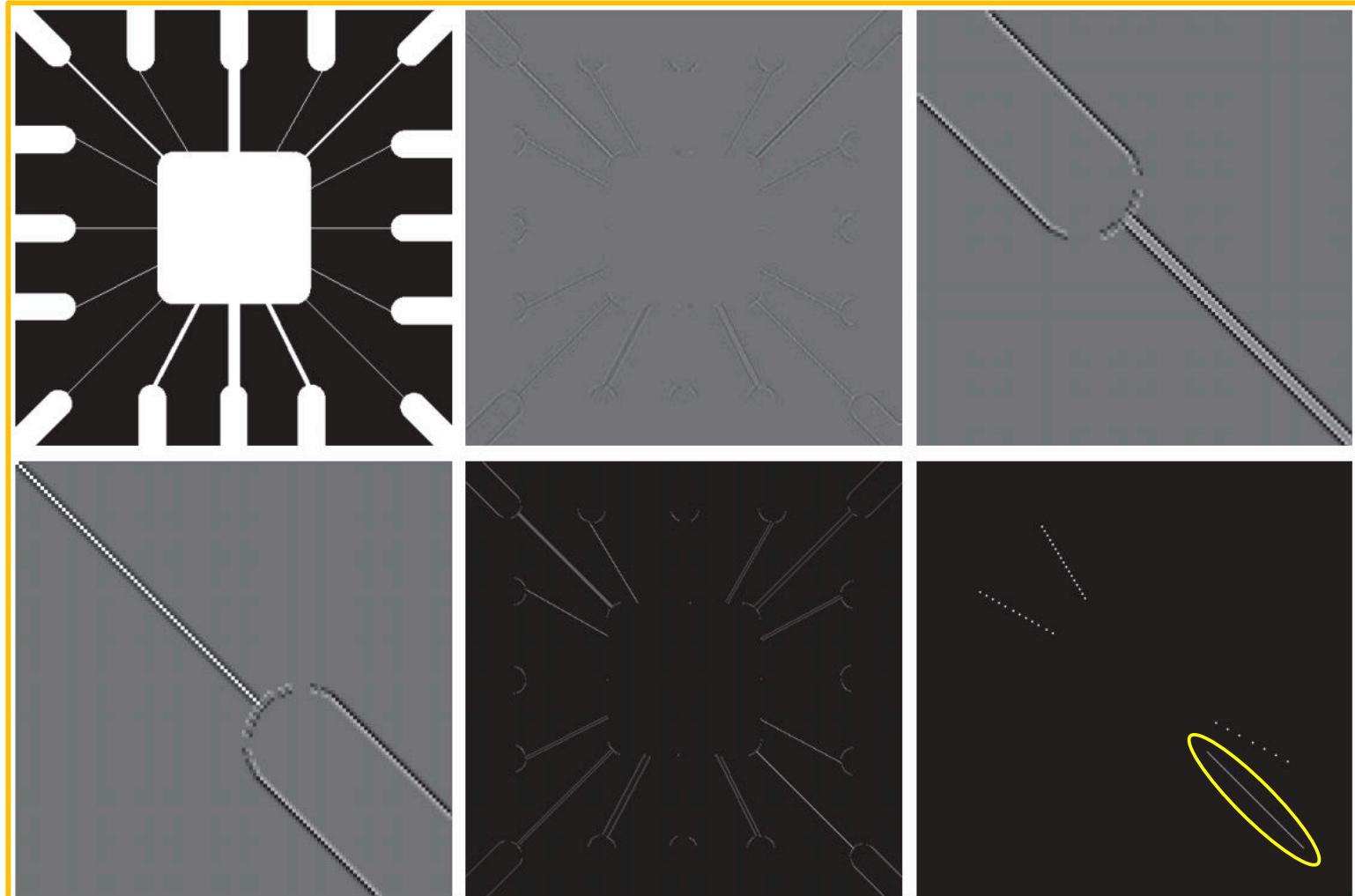
Vertical

-45°

Aplicamos as 4 máscaras sobre os pixels da imagem.

A máscara, cujo resultado ($|Z|$) for o maior, indica que aquele pixel tem maior probabilidade de pertencer a uma linha na direção daquela máscara.

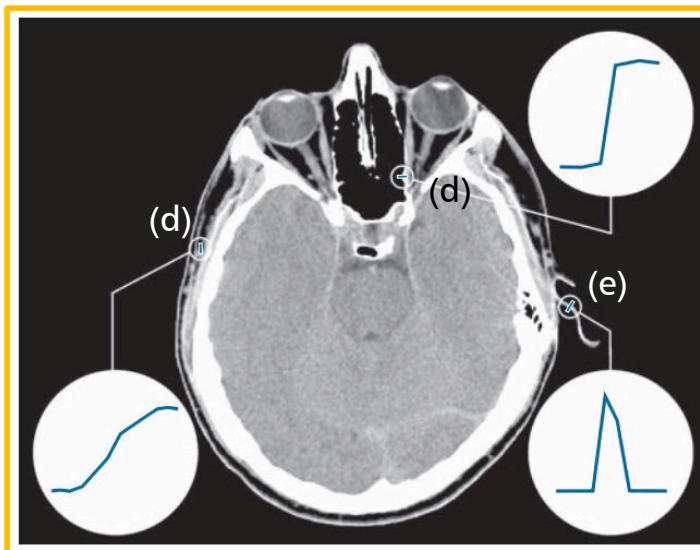
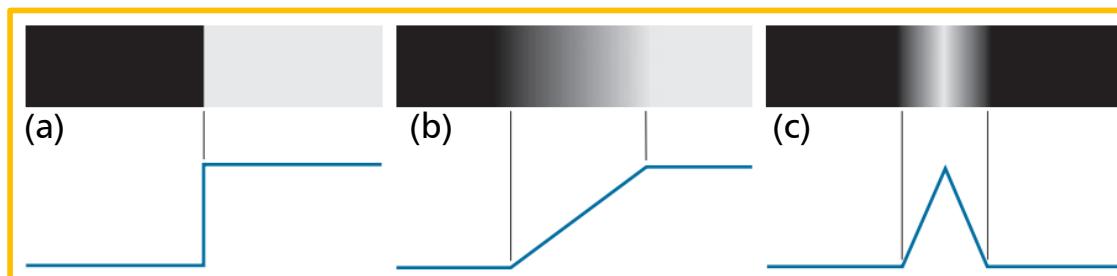
Detecção de Linhas em Direções Específicas



Detecção de linhas em $+45^\circ$

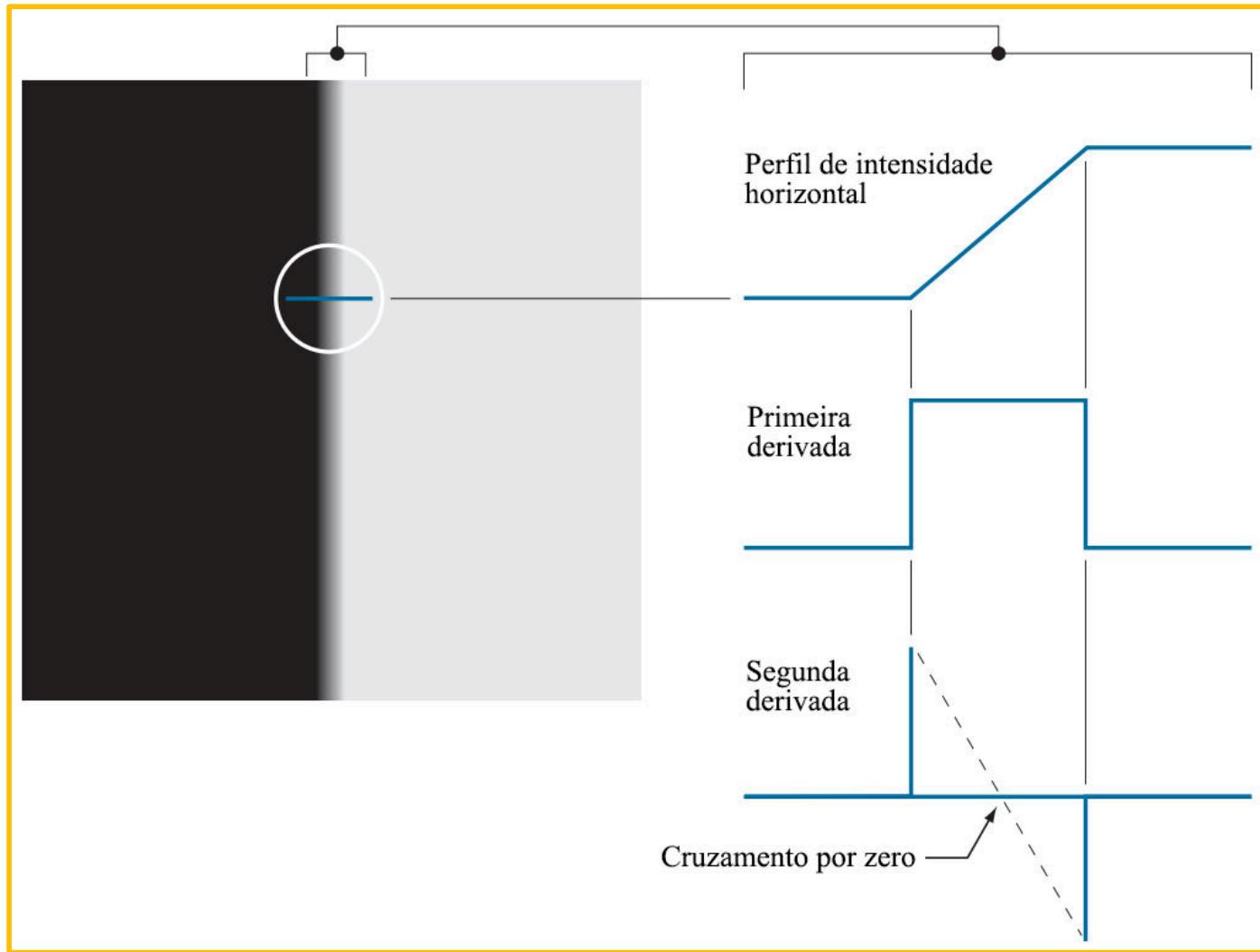
Modelos de Bordas

A detecção de bordas é o método mais usado para segmentar as imagens com base nas variações abruptas de intensidade.

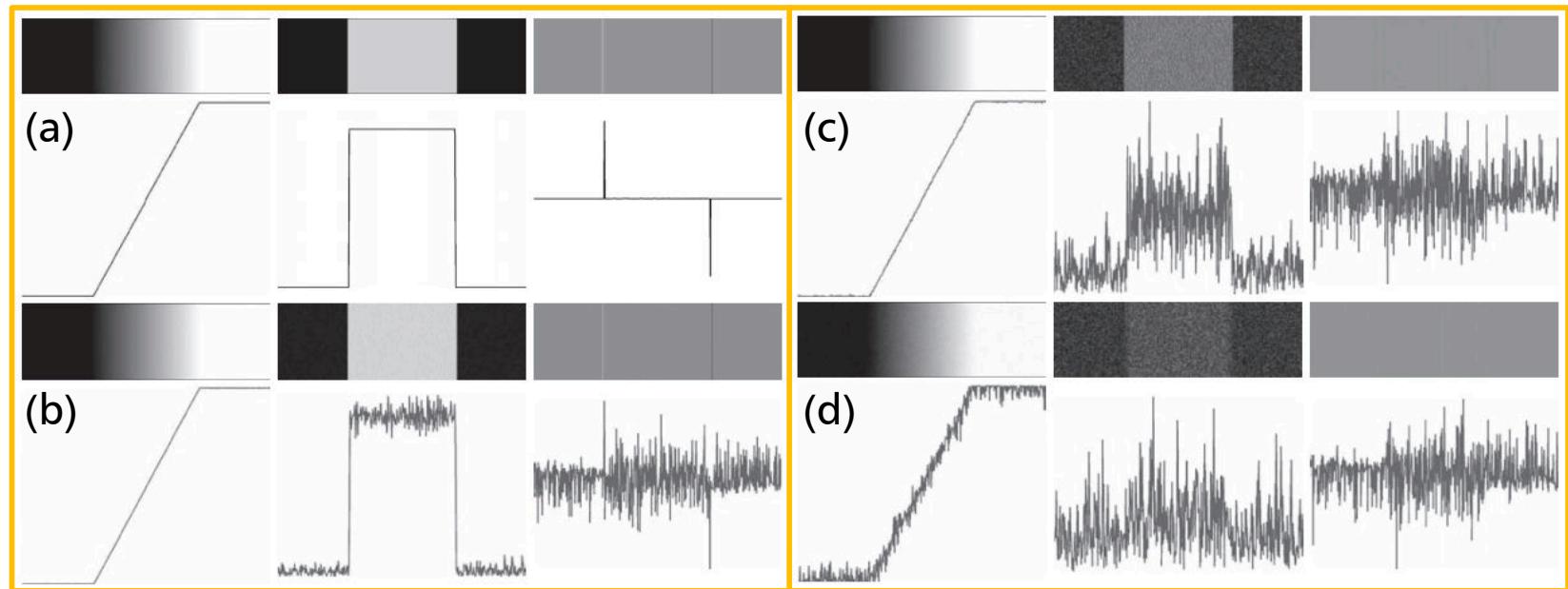


- (a) Borda em degrau
- (b) Borda em rampa
- (c) Borda em forma de telhado
- (d) Bordas em rampa cobrindo 9 e 2 pixels, respectivamente
- (e) Borda em forma de telhado cuja base tem 3 pixels.

Bordas e as Derivadas



Bordas com Ruído e as Derivadas



Na primeira coluna de cada grupo vemos imagens e perfis de intensidade. Na segunda e terceira coluna as mesmas informações para a primeira e segunda derivadas, respectivamente. (a) $\sigma=0.0$, (b) $\sigma=0.1$ (c) $\sigma=1.0$ e (d) $\sigma=10.0$.

Conclusão: É preciso suavizar o ruído!

Detecção de Bordas – Conclusões

Existem 3 passos fundamentais a serem considerados na detecção de bordas:

- 1) Suavização da imagem para redução do ruído;
- 2) Detecção dos pontos de borda: retirar da imagem todos os pontos que são candidatos a se tornarem pontos de borda;
- 3) Localização da borda: Selecionar, dentre os possíveis pontos de borda, apenas aqueles que de fato pertencem ao conjunto de pontos que formam uma borda.

Propriedades do Gradiente

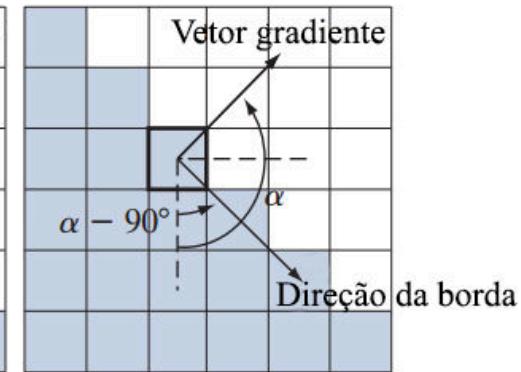
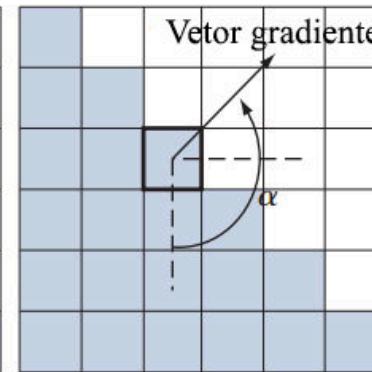
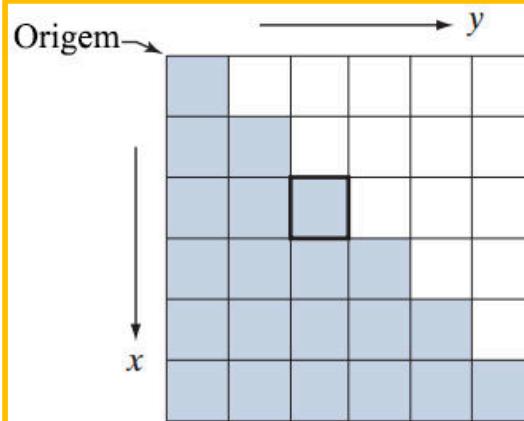
$$\nabla f(x, y) = \begin{bmatrix} g_x(x, y) \\ g_y(x, y) \end{bmatrix} = \begin{bmatrix} \frac{\partial f(x, y)}{\partial x} \\ \frac{\partial f(x, y)}{\partial y} \end{bmatrix}$$

Magnitude

Direção

$$M(x, y) = \text{mag}(\nabla f(x, y)) = \sqrt{g_x^2(x, y) + g_y^2(x, y)}$$

$$\alpha = \tan^{-1} \left(\frac{g_y(x, y)}{g_x(x, y)} \right)$$



$$g_x = \text{linha inferior} - \text{linha superior}$$

$$g_y = \text{coluna esquerda} - \text{coluna direita}$$

$$\nabla f = \begin{bmatrix} -2 \\ 2 \end{bmatrix}$$

$$\alpha = -45^\circ = 135^\circ$$

Gradientes Utilizados na Detecção de Bordas

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \text{ Roberts}$$

$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \text{ Prewitt}$$

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ Sobel}$$

$$\begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \text{ Prewitt diagonais}$$

$$\begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} \quad \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix} \text{ Sobel diagonais}$$

Gradientes Utilizados na Detecção de Bordas

$$\begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} \quad \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} \quad \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \quad \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$

N NW W SW

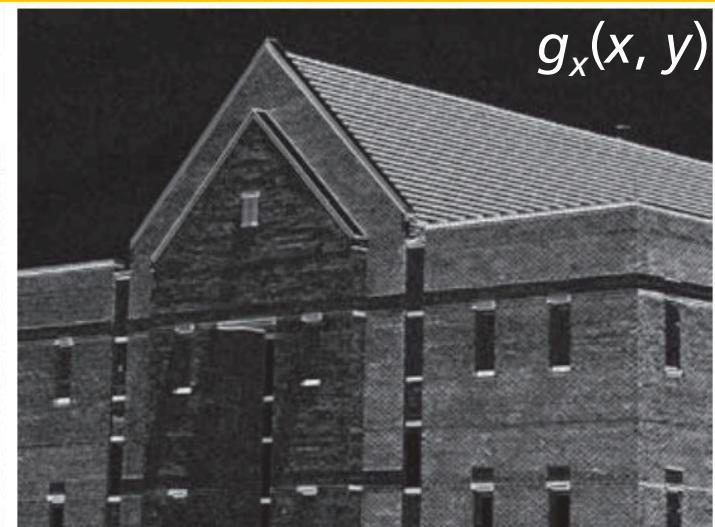
$$\begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} \quad \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix} \quad \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} \quad \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$$

S SE E NE

Gradientes direcionais de Kirsch

Para cada pixel da imagem aplicam-se as 8 máscaras e mantém-se o valor máximo (a magnitude do gradiente).

Detecção de Bordas (Gradiente de Sobel)



Detecção de Bordas (Box 5x5 + Sobel)

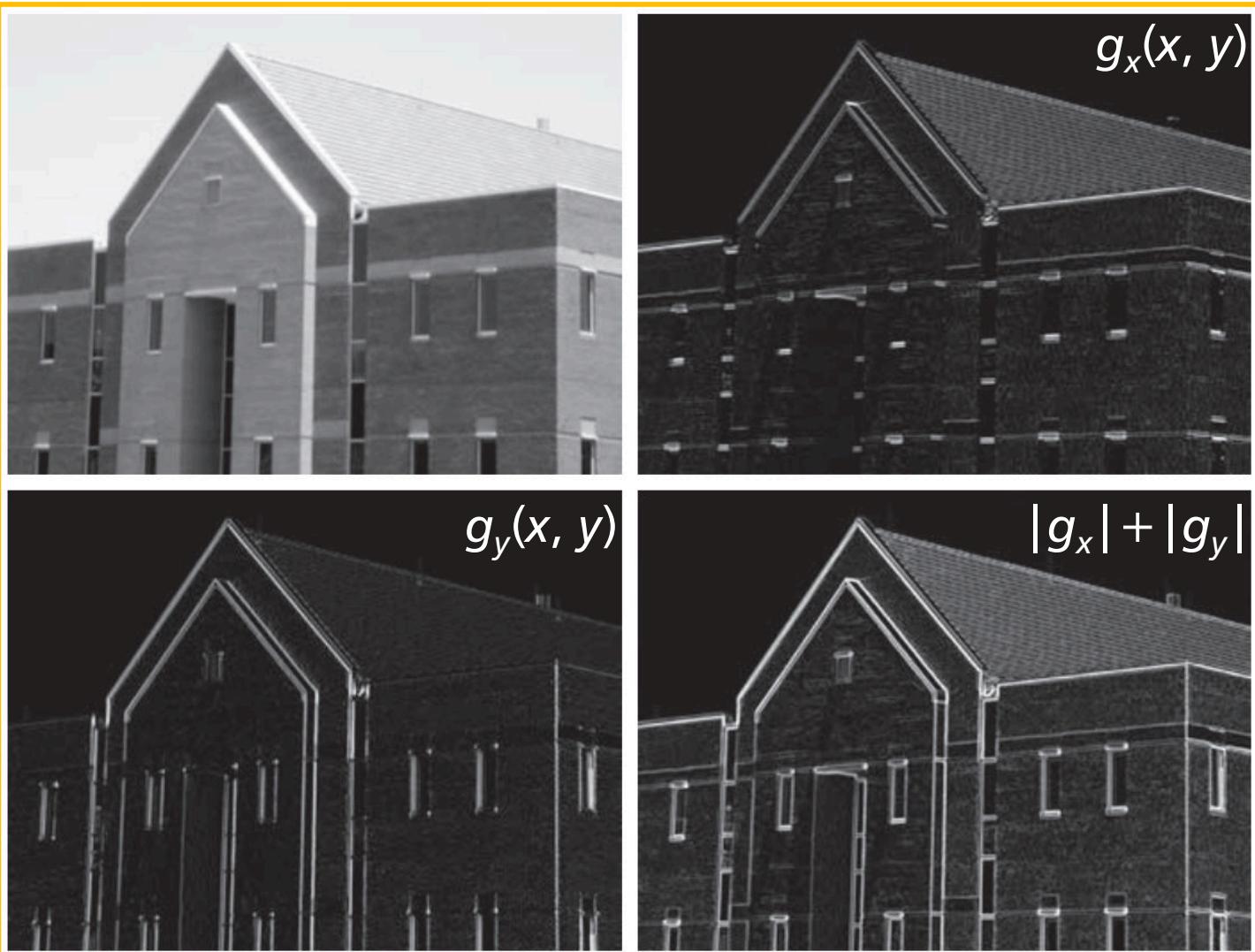
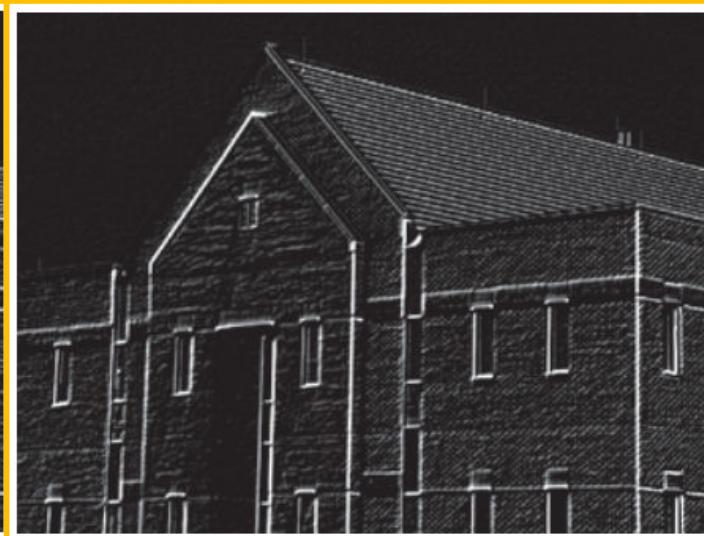
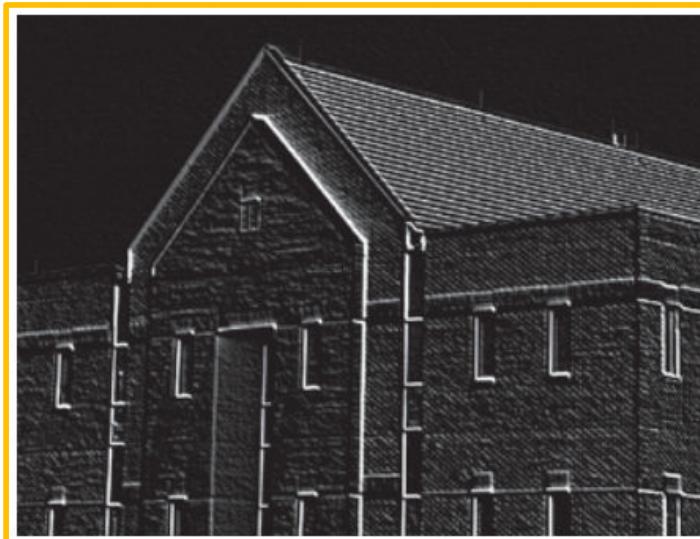


Imagen do Ângulo e Gradientes Direcionais



Imagen do ângulo
do gradiente

Regiões constantes
apontam a mesma
direção do vetor
gradiente

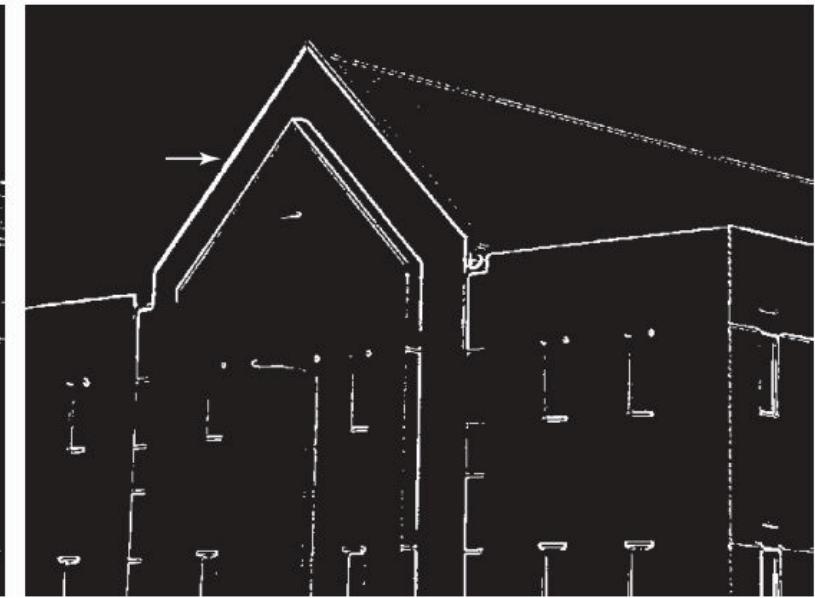


Kirsch
W e SW

Detecção de Bordas (Sobel + Limiarização)



Sobel +
Limiarização



Box 5x5 + Sobel +
Limiarização

Limiarização com $T = 33\%$ do pixel com intensidade máxima.

Se $g(x, y) > T \Rightarrow g(x, y) = \text{Branco}$, senão $g(x, y) = \text{Preto}$.

Detector de Bordas de Marr-Hildreth

Uma das primeiras tentativas de sucesso no processo de busca por borda (1980). Marr&Hildreth acreditavam que:

- 1) Mudanças de intensidade não são independentes de escala, portanto os operadores de detecção de bordas devem possuir diferentes tamanhos;
- 2) Uma mudança súbita de intensidade dará origem a um pico ou vale na 1^a derivada ou um cruzamento por zero na 2^a derivada.

Propuseram um filtro baseado em derivadas e ajustável em escala, o **Laplaciano da Gaussiana** (LoG).

A Função LoG

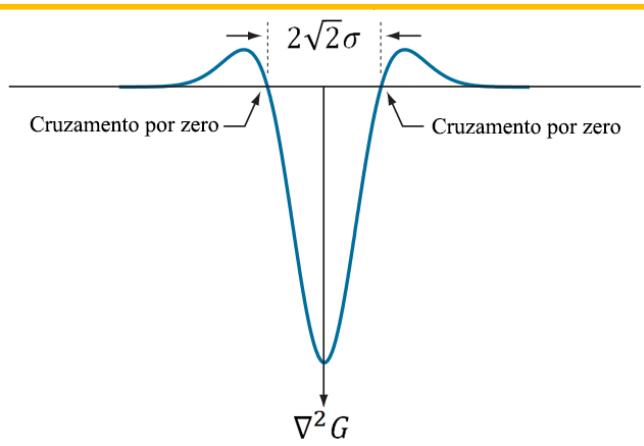
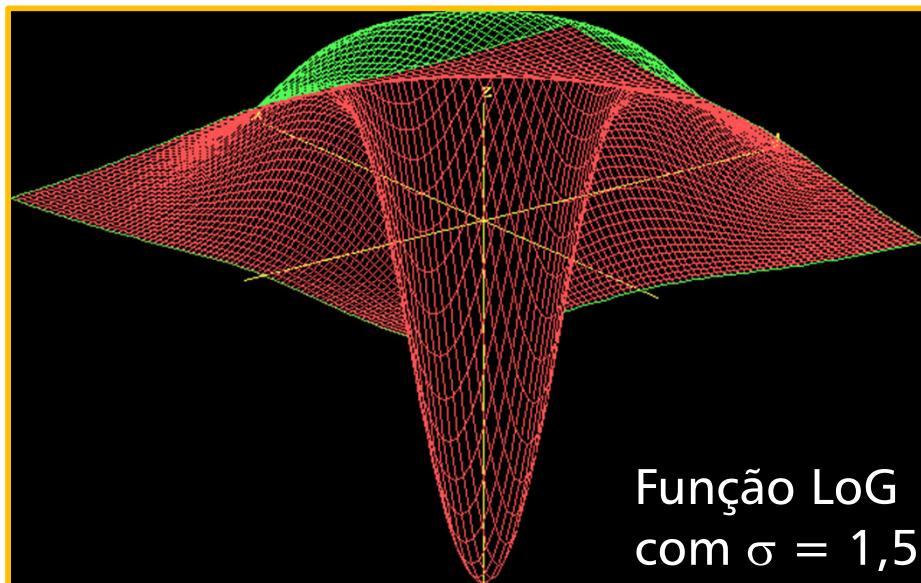
A função LoG ($\nabla^2 G$) é composta por dois elementos, o operador laplaciano e a função gaussiana em 2D.

$$\nabla^2 = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right)$$

$$G(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

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

Função LoG



A Função LoG Discretizada

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 1 & 2 & -16 & 2 & 1 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

LoG (5x5)

$$\begin{bmatrix} 0 & 1 & 1 & 2 & 2 & 2 & 1 & 1 & 0 \\ 1 & 2 & 4 & 5 & 5 & 5 & 4 & 2 & 1 \\ 1 & 4 & 5 & 3 & 0 & 3 & 5 & 4 & 1 \\ 2 & 5 & 3 & -12 & -24 & -12 & 3 & 5 & 2 \\ 2 & 5 & 0 & -24 & -40 & -24 & 0 & 5 & 2 \\ 2 & 5 & 3 & -12 & -24 & -12 & 3 & 5 & 2 \\ 1 & 4 & 5 & 3 & 0 & 3 & 5 & 4 & 1 \\ 1 & 2 & 4 & 5 & 5 & 5 & 4 & 2 & 1 \\ 0 & 1 & 1 & 2 & 2 & 2 & 1 & 1 & 0 \end{bmatrix}$$

LoG (9x9), com $\sigma = 1,4$

99,7% do volume sob uma superfície gaussiana 2D situa-se entre $\pm 3\sigma$. Assim o tamanho do filtro LoG ($n \times n$) deve ser tal que $n = (\text{menor número ímpar maior que } 6\sigma)$.

Geralmente adotamos σ como uma fração da menor dimensão da imagem, por exemplo 0,5%.

O Algoritmo (Marr-Hildreth)

$$g(x, y) = [\nabla^2 G(x, y)] * f(x, y) = \nabla^2[G(x, y) * f(x, y)]$$

1) Filtrar a imagem de entrada com um filtro ($n \times n$) gaussiano passa-baixa, obtido da discretização da função

$$G(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}};$$

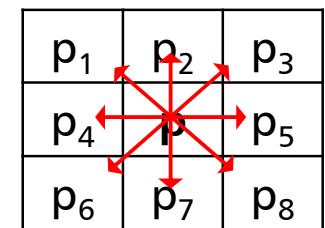
2) Calcular o laplaciano da imagem resultante da etapa 1, com a máscara $\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$, obtendo-se a imagem do LoG.

3) Encontrar os cruzamentos por zero na imagem do LoG.

O Cruzamento por Zero

Um cruzamento por zero implica que, na vizinhança-8 de um pixel p , os sinais de pelo menos dois de seus vizinhos opostos sejam diferentes.

Há quatro casos a testar: p_4p_5 , p_2p_7 , p_1p_8 e p_3p_6 .



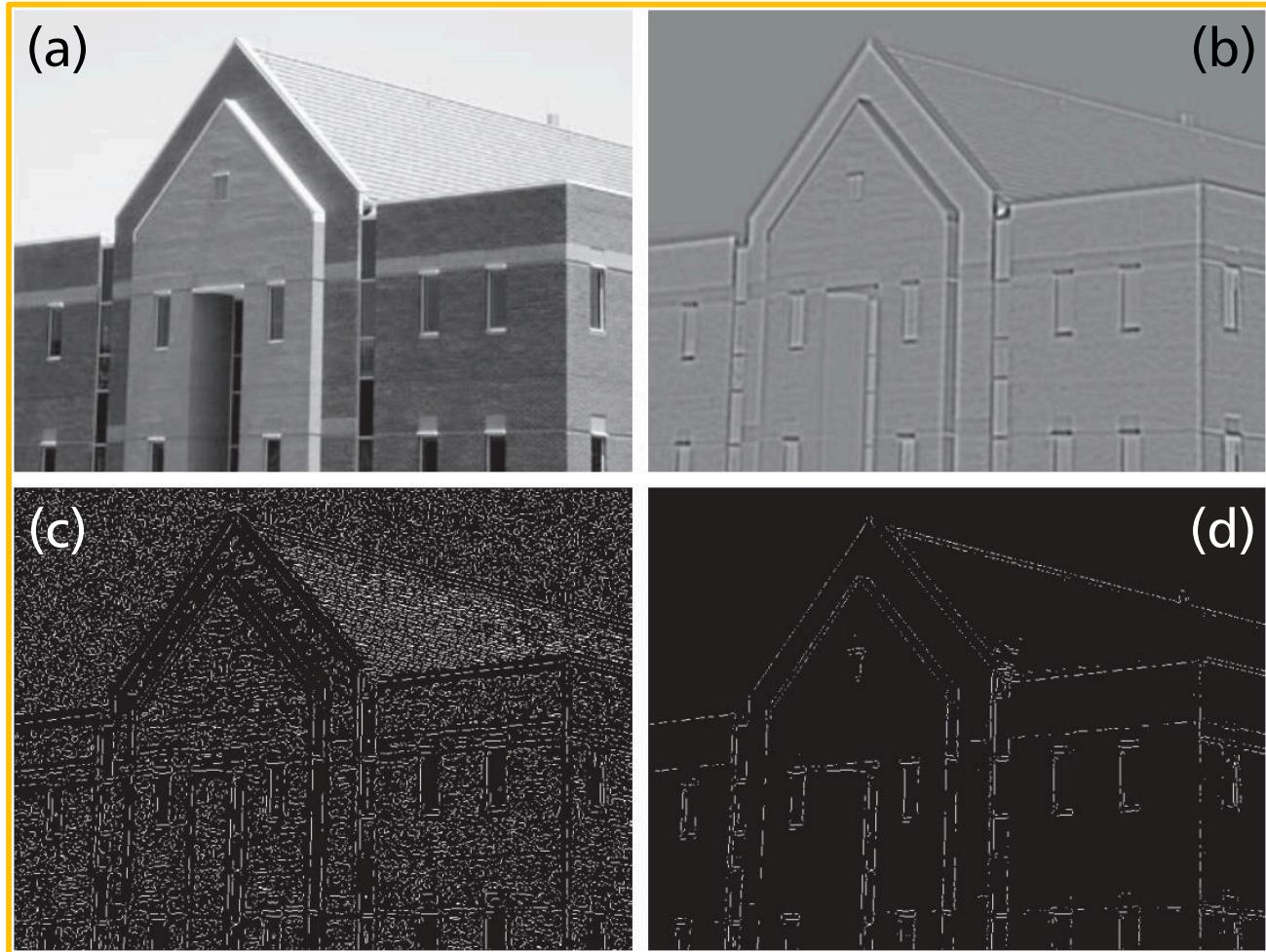
Encontrados pares com sinais diferentes calcula-se:

$$|p_i - p_j| > T$$

Se sim, p será rotulado como um ponto de borda.

O limiar T pode ser definido como uma fração do valor máximo encontrado na imagem LoG.

Detecção de Bordas com Marr-Hildreth



(a) Original. (b) Imagem LoG com $\sigma=4$, $n=25$ e tons de cinza ajustados ao redor de 127.
(c) Cruzamentos por zero com $T=0$. (d) Cruzamentos por zero com $T=4\%$ do $\max(\text{LoG})$.

Detector de Bordas de Canny

Criado por John F. Canny (1986) e considerado o melhor dos detectores de borda. Criado considerando 3 objetivos básicos:

- 1) Baixa taxa de erros: as bordas encontradas devem ser as mais próximas possível das bordas verdadeiras, sem trazer respostas espúrias;
- 2) Pontos de borda bem localizados: devem ser o mais próximos do centro da borda verdadeira;
- 3) Resposta em um único ponto de borda: retornar apenas um ponto para cada ponto de borda verdadeiro.

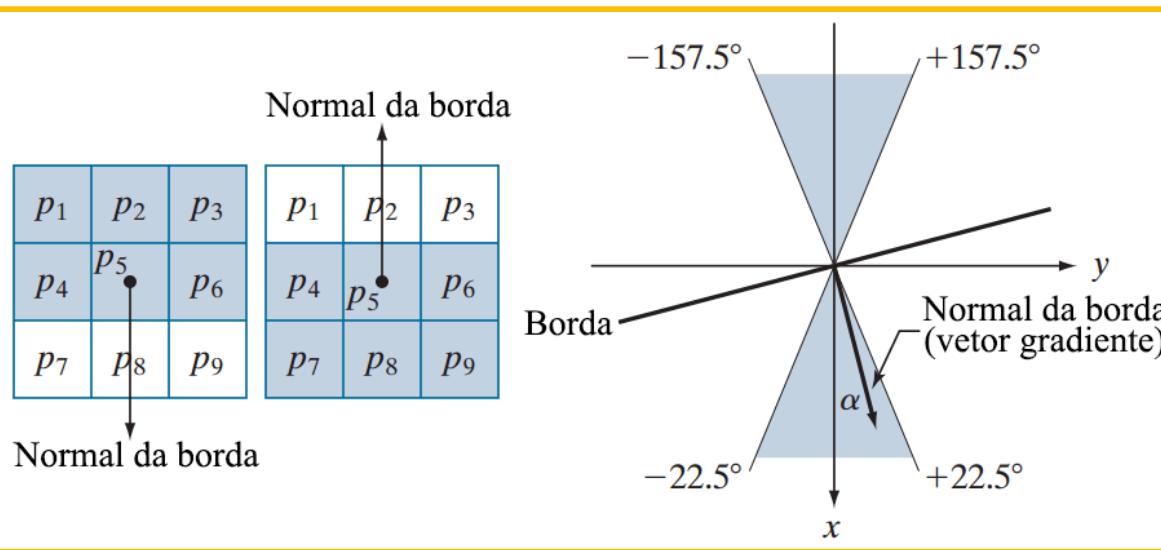
O Algoritmo (Canny)

- 1) Suavizar a imagem de entrada com um filtro ($n \times n$) gaussiano passa-baixa, obtido da discretização da função $G(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$, obtendo $f_s(x, y) = G(x, y) * f(x, y)$;
- 2) Calcular as imagens $M(x, y) = \sqrt{g_x^2 + g_y^2}$ e $\alpha(x, y) = \tan^{-1} \left[\frac{g_y}{g_x} \right]$ à partir de f_s . Os gradientes g_x e g_y podem ser calculados por Roberts, Prewitt ou Sobel;
- 3) Aplicar a supressão não máxima na imagem $M(x, y)$;
- 4) Usar a dupla limiarização e a análise de conectividade para detectar e conectar as bordas.

Supressão Não Máxima

Tem como objetivo afinar as bordas realçadas em $M(x, y)$.

O valor de $\alpha(x, y)$ indica vetor normal à borda predominante em $M(x, y)$, que pode estar em quatro direções: horizontal, -45° , vertical ou $+45^\circ$.

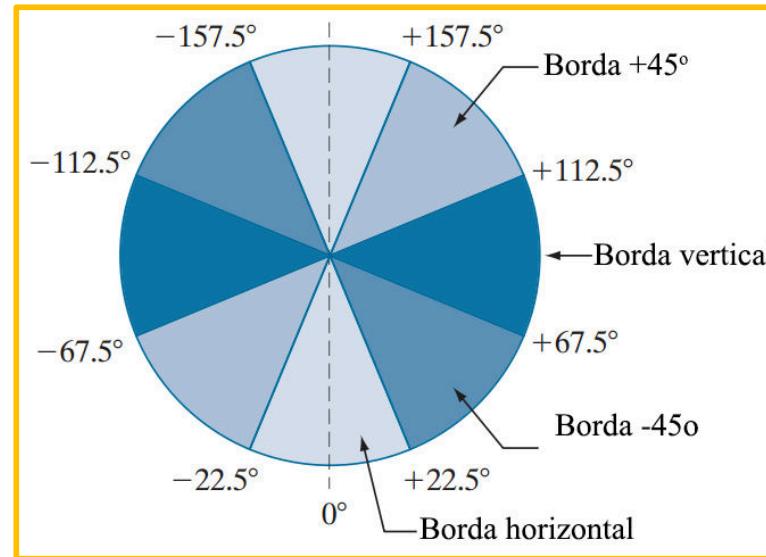


Neste exemplo o valor de $\alpha(x, y)$ (ângulo do vetor normal à borda) está entre $-22,5^\circ$ e $+22,5^\circ$, indicando que a borda é horizontal.

Como a borda é horizontal, a supressão não máxima deve analisar em $M(x, y)$ os pixels p_2 , p_5 e p_8 .

Supressão Não Máxima

A discretização de $\alpha(x, y)$ em 4 direções (d_1, d_2, d_3 e d_4), considera os limites da figura abaixo.



p_1	p_2	p_3
p_4	p_5 •	p_6
p_7	p_8	p_9

Se $M(x, y)$ for inferior a pelo menos um dos vizinhos ao longo de d_k , fazemos $g_N(x, y) = 0$, senão $g_N(x, y) = M(x, y)$.

Dupla Limiarização

Canny usa limiarização por histerese com dois limiares: T_L (baixo) e T_H (alto), na razão alto:baixo igual a 2:1 ou 3:1, criando 2 imagens: uma com pixels de bordas fortes - $g_{NH}(x, y)$ - e outra de bordas fracas - $g_{NL}(x, y)$.

$$\begin{aligned} g_{NH}(x, y) &= g_N(x, y) \geq T_H \text{ e} \\ g_{NL}(x, y) &= g_N(x, y) \geq T_L \end{aligned}$$

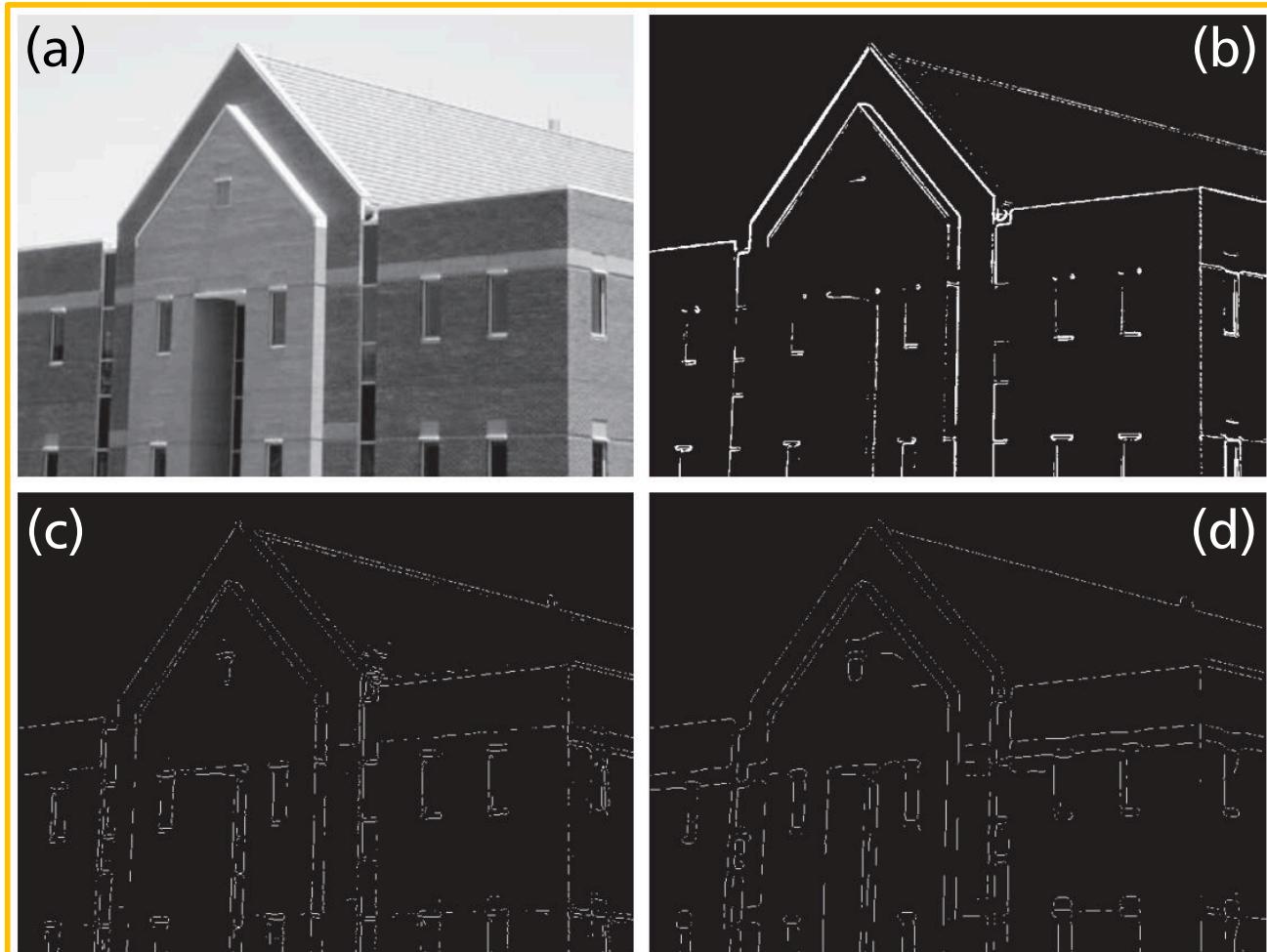
As bordas em $g_{NH}(x, y)$ são imediatamente rotulados como bordas válidas, mas podem apresentar falhas. Para gerar bordas longas usamos informações da imagem $g_{NL}(x, y)$, seguindo o algoritmo apresentado a seguir.

Algoritmo para Formar Bordas Longas

- 1) Localize o próximo pixel de borda a ser revisado, p , em $g_{NH}(x, y)$;
- 2) Marque como pixels de borda válidos todos os pixels fracos que estão conectados a p , usando a conectividade-8, por exemplo;
- 3) Se todos os pixels diferentes de zero em $g_{NH}(x, y)$ foram revisados, vá para o passo 4; senão volte para o passo 1;
- 4) Atribua zero a todos os pixels de $g_{NL}(x, y)$ que não foram marcados como pixels de bordas válidos.

A imagem final será formada atribuindo a $g_{NH}(x, y)$ todos os pixels não nulos em $g_{NL}(x, y)$.

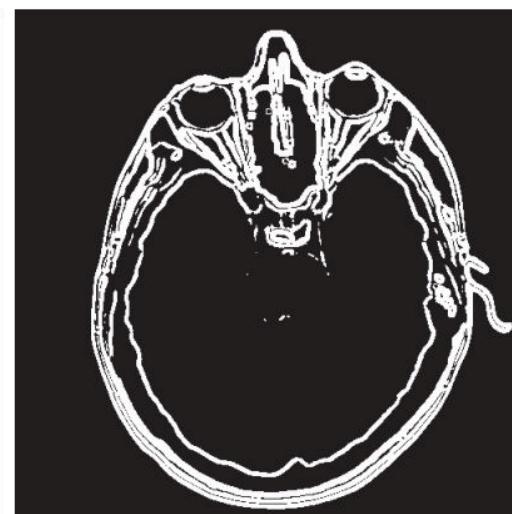
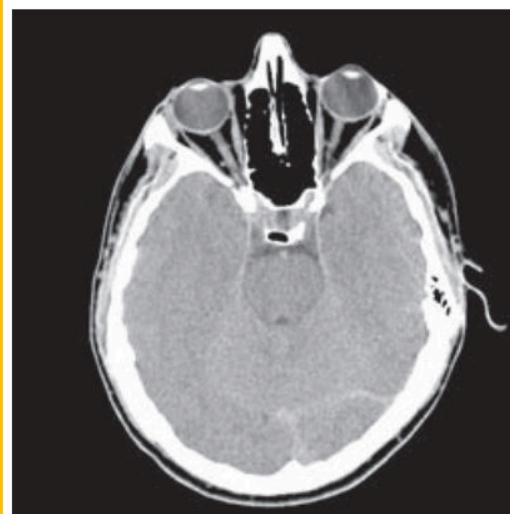
Detecção de Bordas com Canny



(a) Original (834 x 1114 pixels). (b) Sobel + Limiarização. (c) Marr-Hildreth. (d) Canny, com gaussiano $\sigma=4$ (25x25), $T_L=4\%$ e $T_H=10\%$.

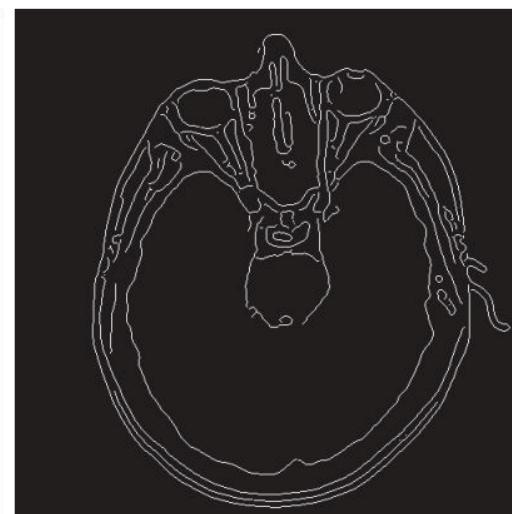
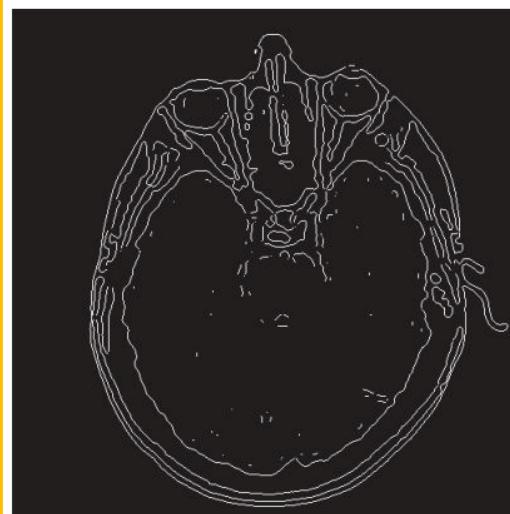
Comparação entre Detectores de Bordas

Imagen TC
(512 x 512)



Gradiente
limiarizado
da imagem
suavizada

Marr-Hildreth



Canny

Ligaçāo dos Pontos de Bordas

Processo executado após a detecção de bordas que visa corrigir as falhas ocasionadas por ruídos e outros efeitos que modificam as intensidades dos pixels.

Há 3 técnicas usadas neste processo:

- 1) Processamento local: exige conhecimento prévio sobre os pontos da borda em uma região local;
- 2) Processamento regional: requer que os pontos na fronteira de uma região sejam conhecidos;
- 3) Processamento global: trabalha com toda a imagem da borda.

Processamento Local das Bordas

Analisa a vizinhança de cada ponto da borda (x, y) e, considerando a **magnitude** e a **direção** (ângulo) do vetor gradiente, determina quais pixels da vizinhança são similares e serão integrados à borda.

Um pixel (s, t) na vizinhança do pixel (x, y) é similar em **magnitude** e **direção** se:

$$|M(s, t) - M(x, y)| \leq E, \text{ onde } E \text{ é um limiar positivo.}$$

$$|\alpha(s, t) - \alpha(x, y)| \leq A, \text{ onde } A \text{ é um limiar de ângulo positivo.}$$

Este processo é computacionalmente caro pois todos os pixels vizinhos de cada (x, y) deve ser examinado. Em aplicações de tempo real emprega-se uma simplificação.

Processamento Local das Bordas

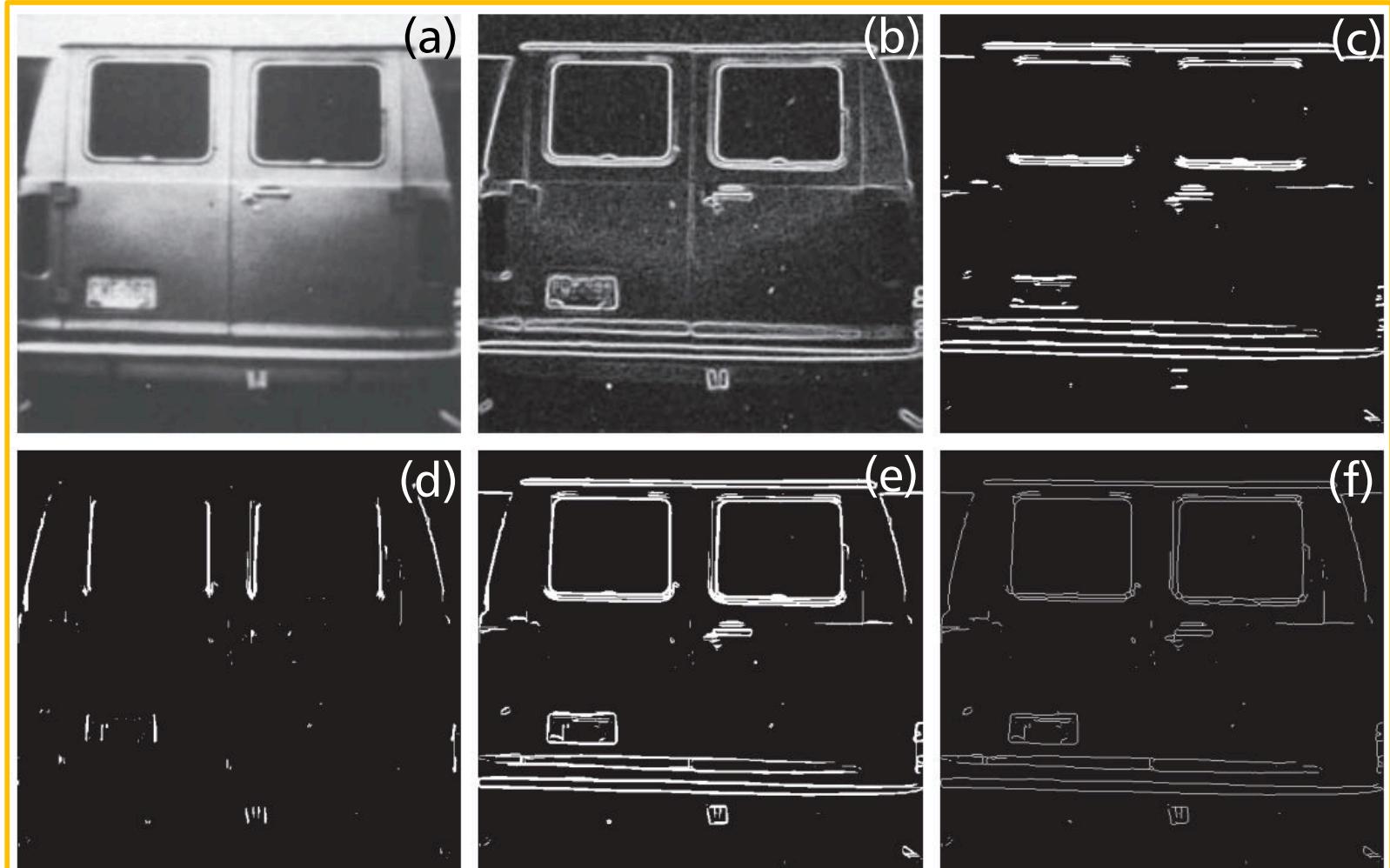
- 1) Calcule as matrizes $M(x, y)$ e $\alpha(x, y)$ da imagem de entrada $f(x, y)$;
- 2) Crie uma imagem binária $g(x, y)$ com a regra:

$$g(x, y) = \begin{cases} 1, & \text{se } M(x, y) > T_M \text{ e } \alpha(x, y) = A \pm T_A \\ 0, & \text{caso contrário} \end{cases}$$

T_M é um limiar. A é a direção angular específica e $\pm T_A$ define uma faixa de direções aceitáveis ao redor de A .

- 3) Percorra as linhas de $g(x, y)$ e marque com 1 as falhas (conjuntos de zeros com extremos iguais a 1) que não excedam um limite K .
- 4) Para corrigir as falhas em qualquer direção θ , faça $R(\theta)$ em $g(x, y)$ e execute novamente a etapa 3. Para retornar à inclinação original faça $R(-\theta)$ em $g(x, y)$.

Processamento Local das Bordas



(a) Original (534×566 pixels). (b) $M(x, y)$. (c) e (d) Etapas 3 e 4 do algoritmo com $T_M = 0,3(\max(b))$, $A = 90^\circ$, $T_A = 45^\circ$ e $K \leq 25$ pixels. (e) (c)OU(d). (f) Afinamento de (e).

Processamento Regional das Bordas

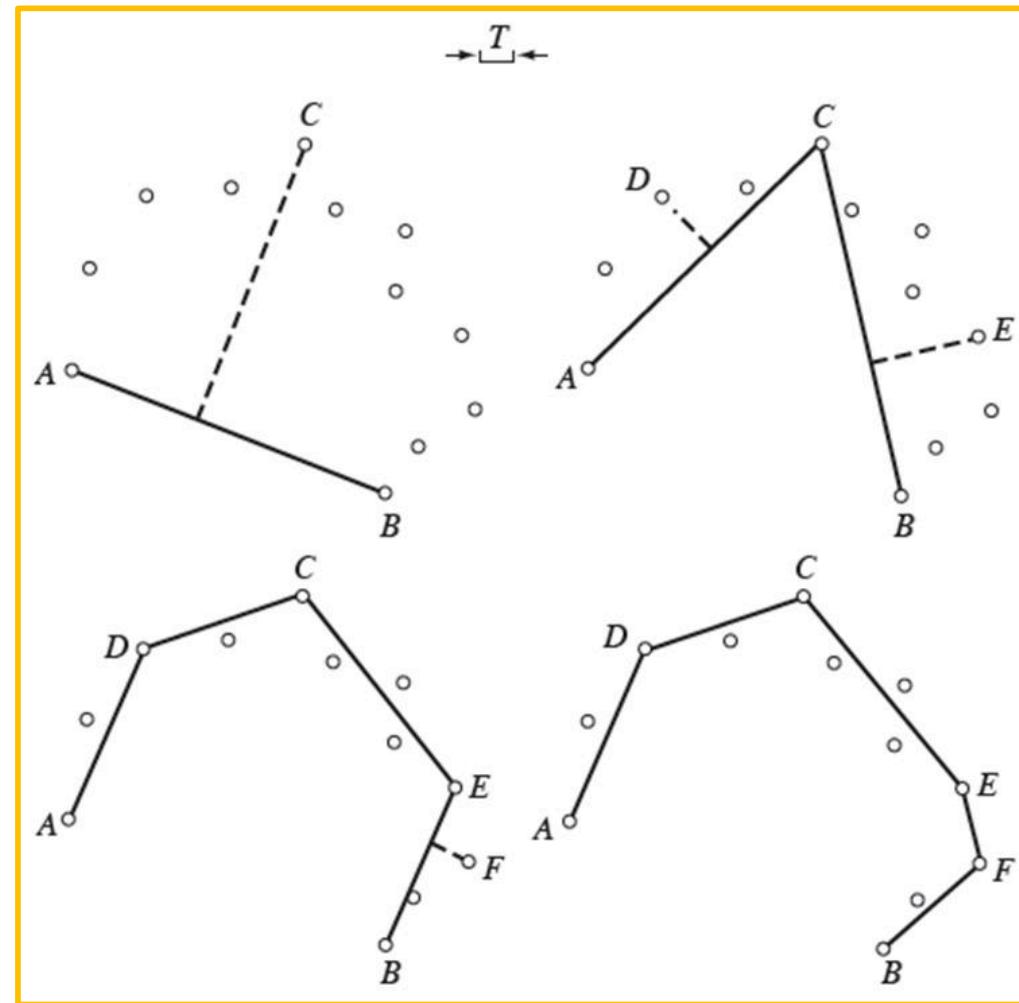
Usado quando se sabe que o conjunto de pixels a serem analisados pertencem às bordas de um mesmo objeto.

O processo consiste em encontrar uma aproximação poligonal do conjunto de pixels de borda, partindo de dois pontos extremos (A e B). O conjunto de pixels deve estar ordenado em sentido horário ou anti-horário.

Um limiar T deve ser definido, estabelecendo a precisão da aproximação poligonal a ser obtida. $\downarrow T \Rightarrow \uparrow \text{precisão}$.

Considerando AB um segmento inicial, calcula-se a distância perpendicular entre AB e todos os demais pixels. O ponto mais distante ($d \geq T$) define um novo vértice na aproximação poligonal. Repete-se o processo até não ser possível adicionar novo vértice.

Processamento Regional das Bordas



Neste exemplo o conjunto de pontos define uma curva aberta.

Processamento Regional das Bordas

Nem sempre sabemos, à priori, se conjunto de pontos definem uma curva aberta ou a fronteira de um objeto (curva fechada).

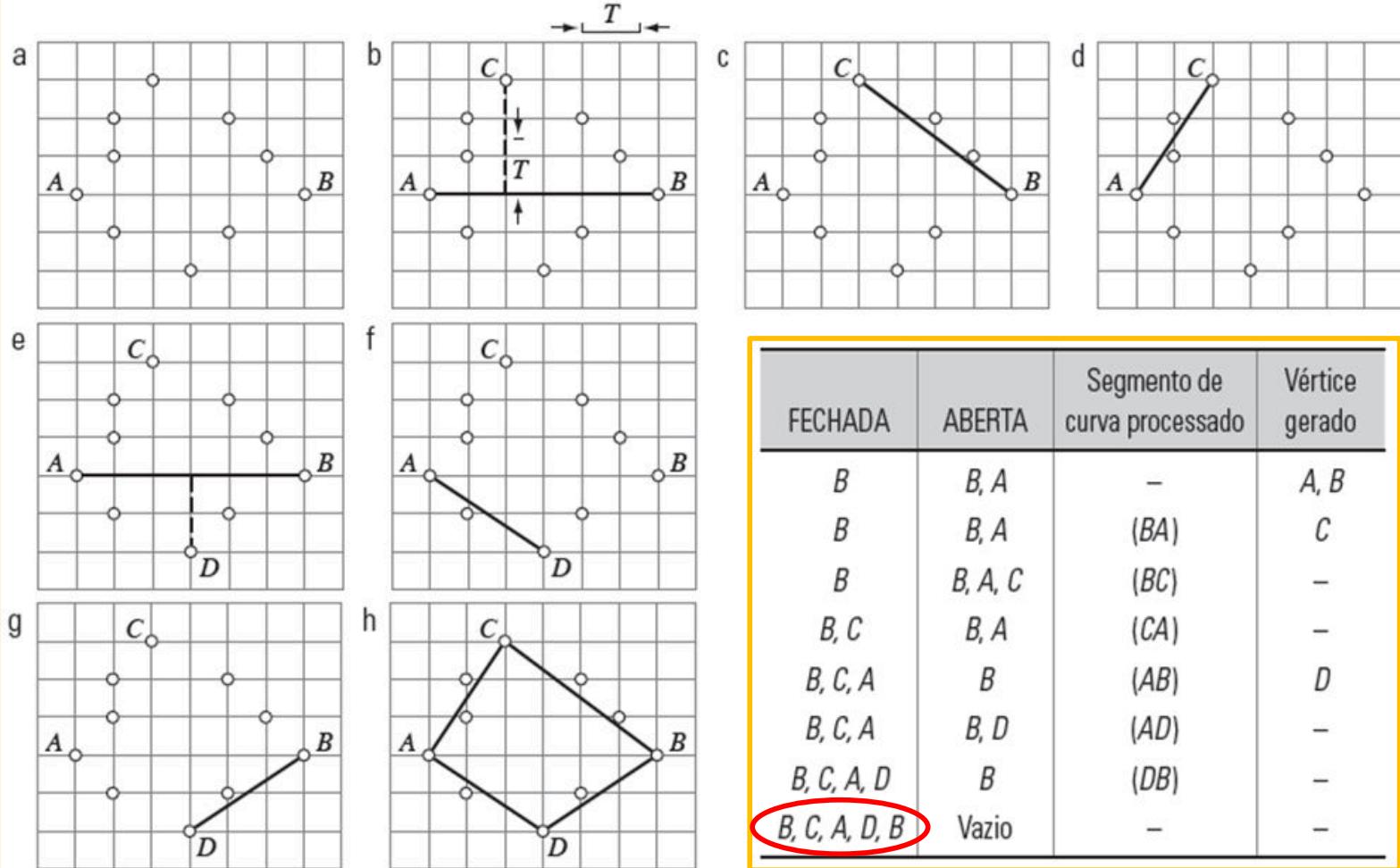
Se há um par de pontos consecutivos cuja distância é grande, quando comparada às distâncias entre os demais pares de pontos consecutivos, intuímos que se trata de uma curva aberta. Os pontos finais são utilizados para iniciar o procedimento.

Se as distâncias entre os pares de pontos consecutivos forem homogêneas intuímos que temos uma curva fechada. Nesses caso os pontos mais à esquerda e à direita são os pontos de partida para o procedimento.

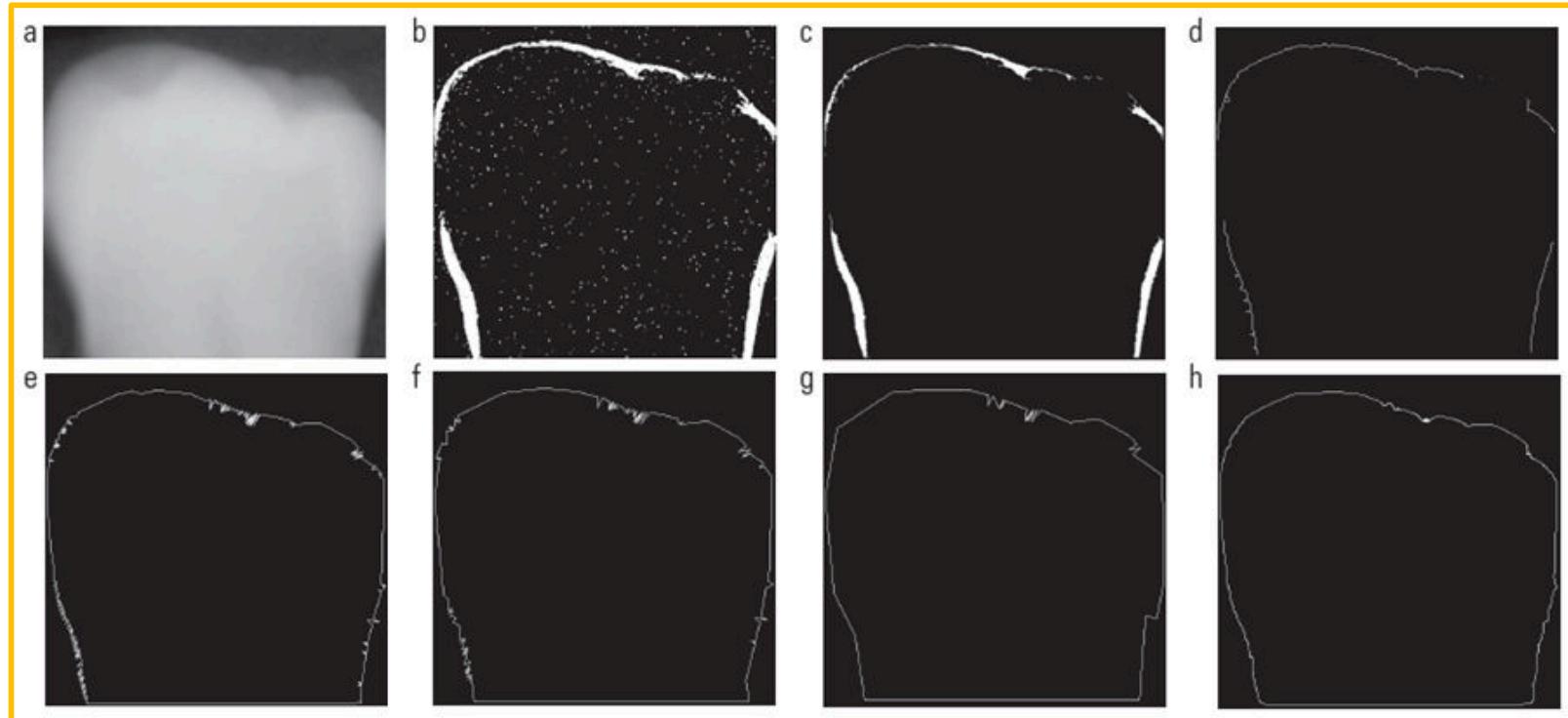
Processamento Regional das Bordas

- 1) P é um conjunto ordenado de pontos. A e B são os pontos de partida;
- 2) Definimos o limiar T e duas pilhas vazias (**Ab** e **Fe**);
- 3) Se P define uma curva fechada empilhamos B em **Ab** e **Fe** e A em **Ab**. Se a curva for aberta colocamos A em **Ab** e B em **Fe**;
- 4) Calculamos os parâmetros da reta que passa pelos vértices no topo de **Fe** e no topo de **Ab**;
- 5) Para todos os pontos, entre aqueles vértices obtidos em (4), calculamos suas distâncias em relação à reta obtida em (4). Selecionamos o ponto V_{max} , com distância D_{max} ;
- 6) Se $D_{max} > T$, empilhamos o vértice V_{max} em **Ab** e retornamos a (4);
- 7) Senão removemos o vértice no topo de **Ab**, empilhando-o em **Fe**;
- 8) Se a pilha **Ab** não estiver vazia, retornamos a (4);
- 9) Caso contrário, saímos. Os vértices em **Fe** definem a aproximação poligonal do conjunto de pontos P.

Processamento Regional das Bordas



Processamento Regional das Bordas



(a) Original (550 x 566 pixels). (b) Sobel + Limiarização (10% de $\max(M(x, y))$).
(c) Remoção de ruídos. (d) Esqueletização + Poda. (e, f, g) Aproximações poligonais
com $T = 3, 6$ e 12 . (h) Filtro da média-1D (1×31 pixels) sobre (f).

Processamento Global das Bordas

O processamento local e regional das bordas faz uso de algum conhecimento prévio acerca dos pixels pertencentes a objetos individuais.

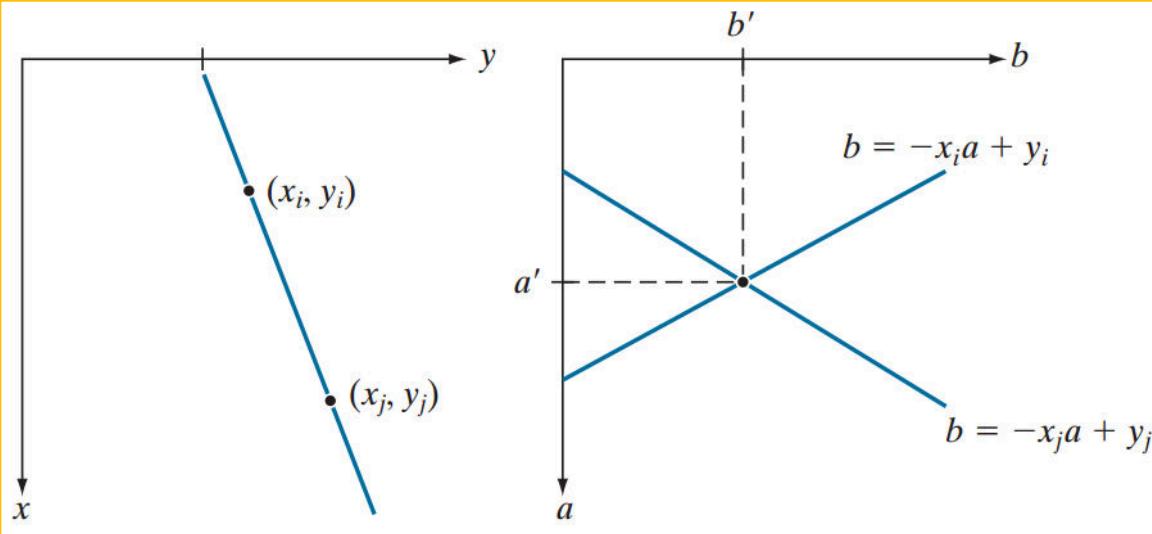
Mas quando temos apenas a imagem das bordas, sem conhecimento da localização dos objetos de interesse, todos os pixels são candidatos a serem ligados.

Nesta condição temos de aceitar ou eliminar pixels com base em propriedades globais pré-definidas.

Exemplos de propriedades globais pré-definidas são a linearidade e formas geométricas como círculos, elipses ou curvas.

Transformada de Hough

Desenvolvida em 1962 por Paul Hough. A **transformada de Hough** cria um espaço parametrizado onde pontos colineares geram retas que se cruzam em um ponto do espaço de parâmetros, permitindo identificar a reta suporte dos pontos colineares.

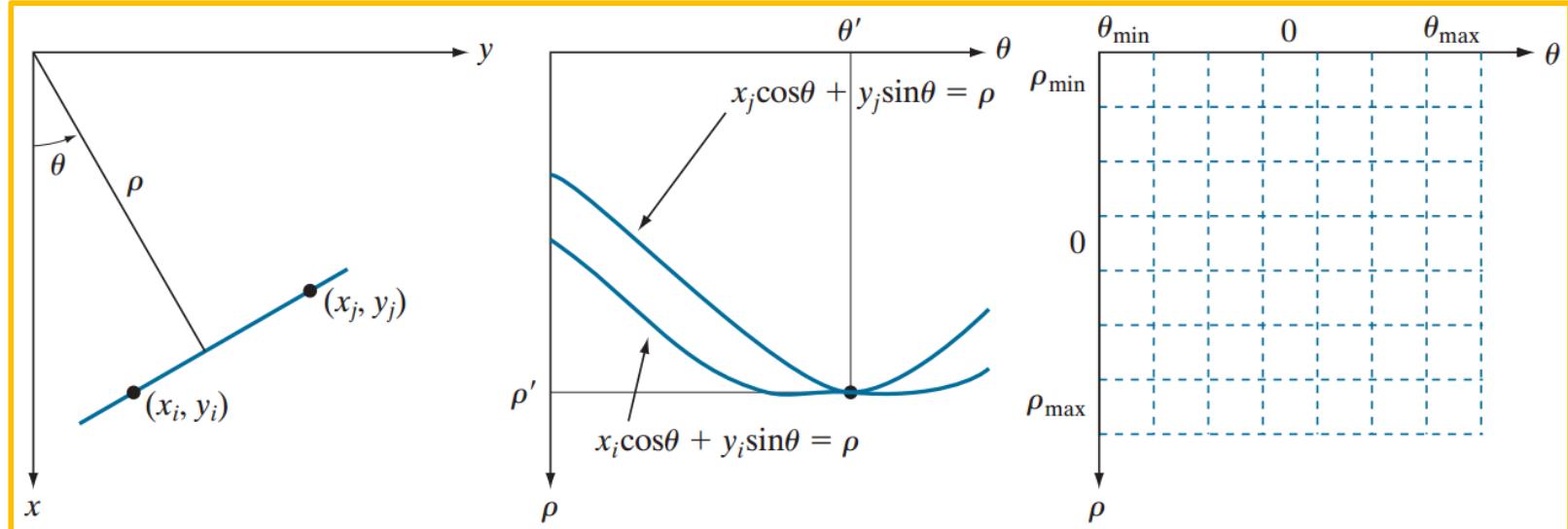


$$\begin{aligned}P &= (x_i, y_i) \\y_i &= ax_i + b \\b &= -x_i a + y_i \\Q &= (x_j, y_j) \\b &= -x_j a + y_j\end{aligned}$$

Transformada de Hough

Para contornar o problema das retas com inclinação vertical (coeficiente angular tende a infinito), usa-se outra parametrização, com coordenadas polares.

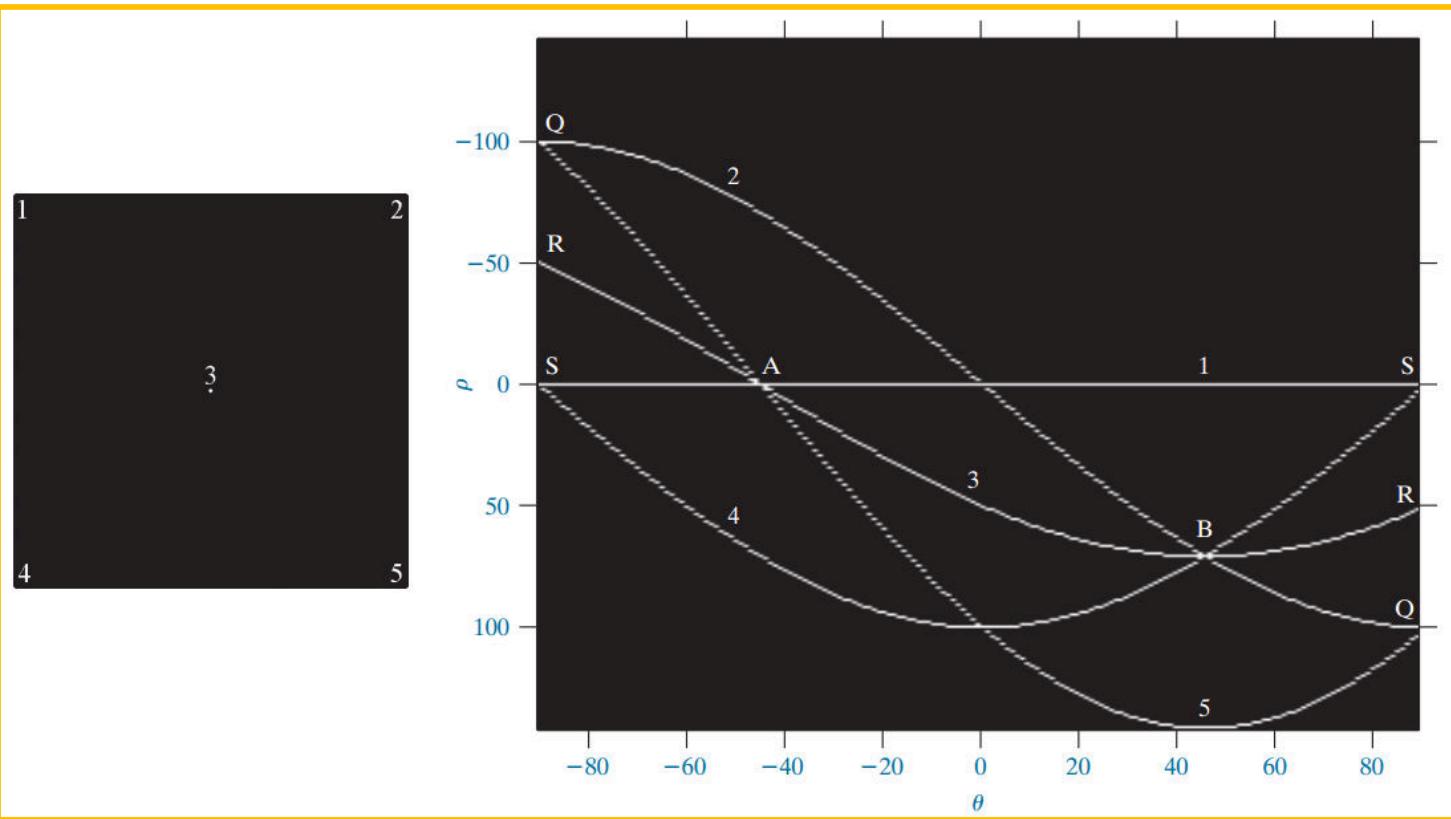
$$x\cos(\theta) + y\sin(\theta) = \rho$$



$$-90^\circ \leq \theta \leq +90^\circ, -D \leq \rho \leq +D$$

com D = distância entre os cantos opostos da imagem

Transformada de Hough



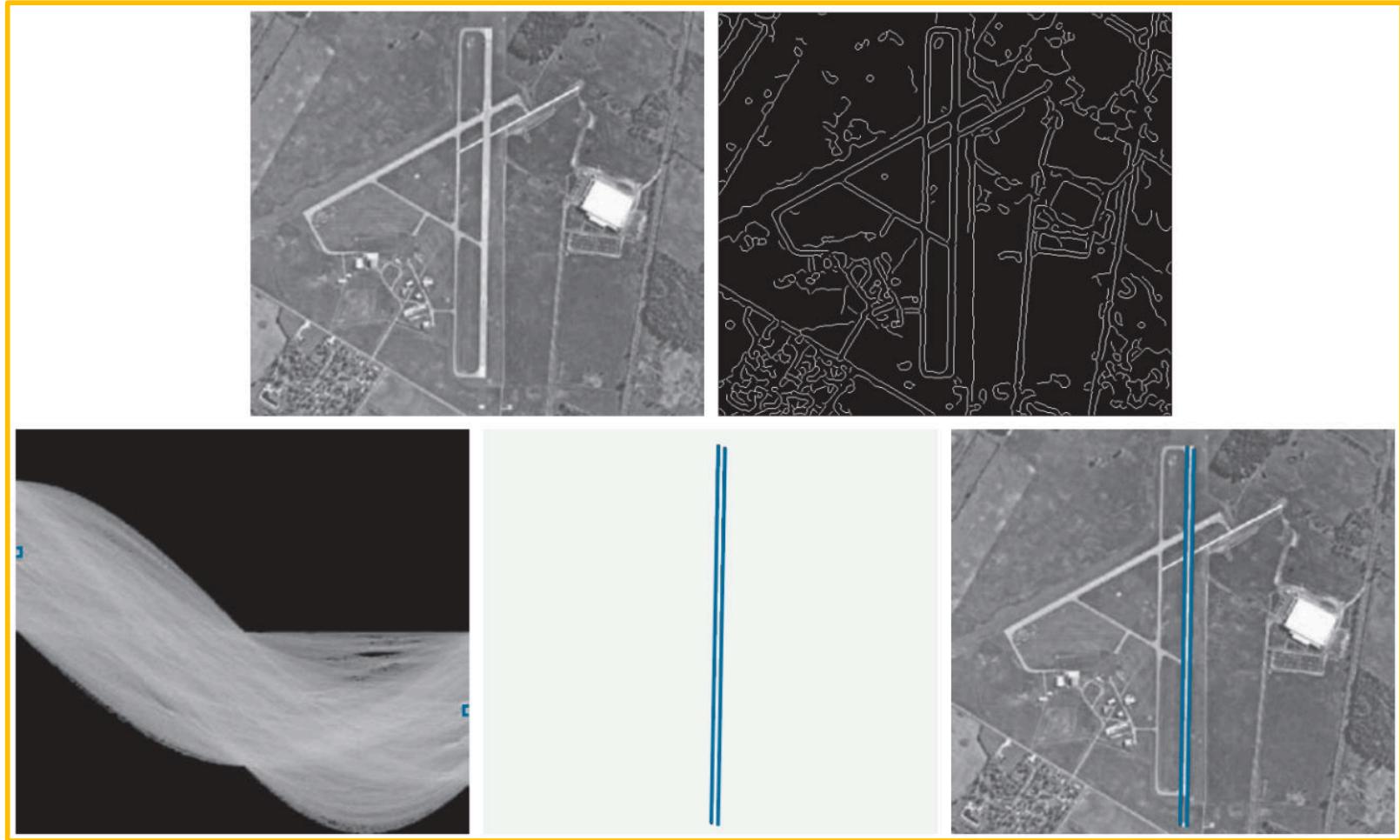
Sobre um ponto passam infinitas retas, com diferentes valores para ρ e θ , dando origem a uma senoide. O cruzamento entre as senoides indicam pontos colineares.

Transformada de Hough

Um algoritmo possível para detectar linhas é o que segue:

- 1) Obtenha uma imagem binária com as bordas da imagem;
- 2) Defina como o plano $\rho\theta$ será dividido (estrutura da matriz acumuladora);
- 3) Aplique a parametrização aos pontos da imagem das bordas, atualizando a matriz acumuladora;
- 4) Examine a matriz acumuladora em busca de células com valores elevados;
- 5) Examine a relação (principalmente as de continuidade) entre os pixels oriundos das células escolhidas em (4).

Transformada de Hough



Transformada de Hough para Circunferências

A equação de uma circunferência pode ser escrita como:

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

$$r = \sqrt{(x - a)^2 + (y - b)^2}$$

(a, b) = centro da circunferência; r = raio.

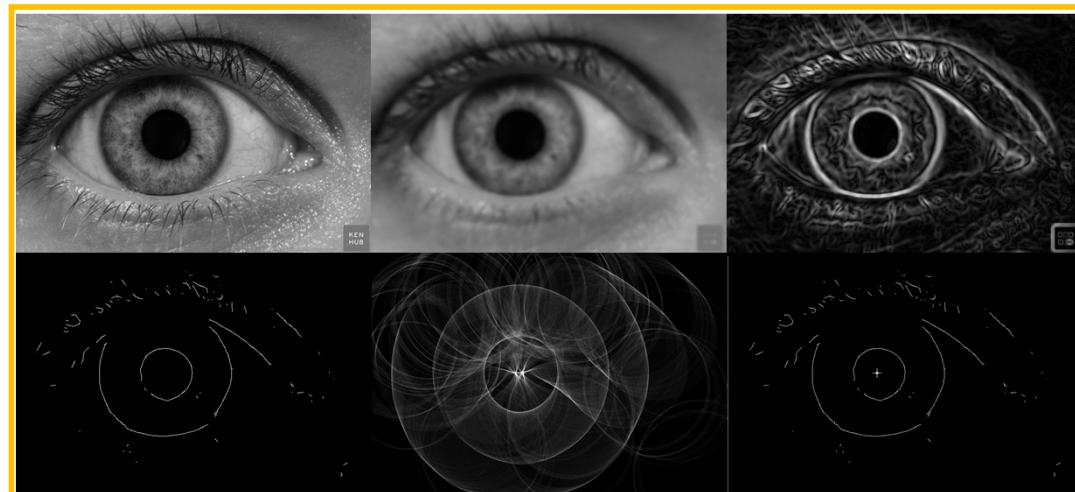
$$x_0 = x + \rho \cos \theta$$

$$y_0 = y + \rho \sin \theta$$

$\rho = [\rho_{min} \dots \rho_{max}]$; $\theta = [0..2\pi]$; (x, y) = pixels da imagem; (x_0, y_0) = centros calculados

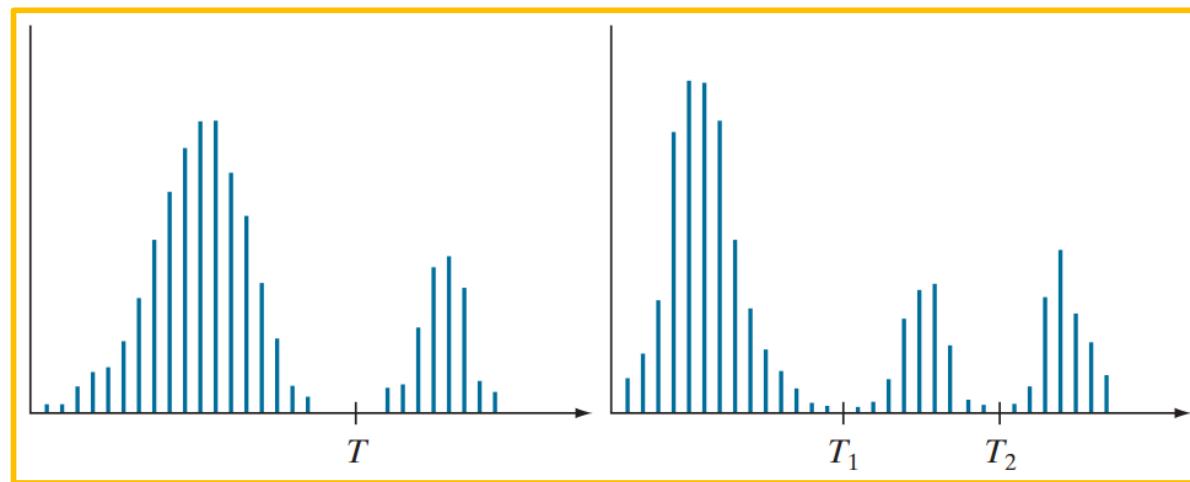
Matriz acumuladora [a, b, r].

Matriz acumuladora [x_0, y_0, ρ]



Limiarização

A **limiarização** é um processo que visa separar uma imagem em regiões com base nos valores de intensidade.



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

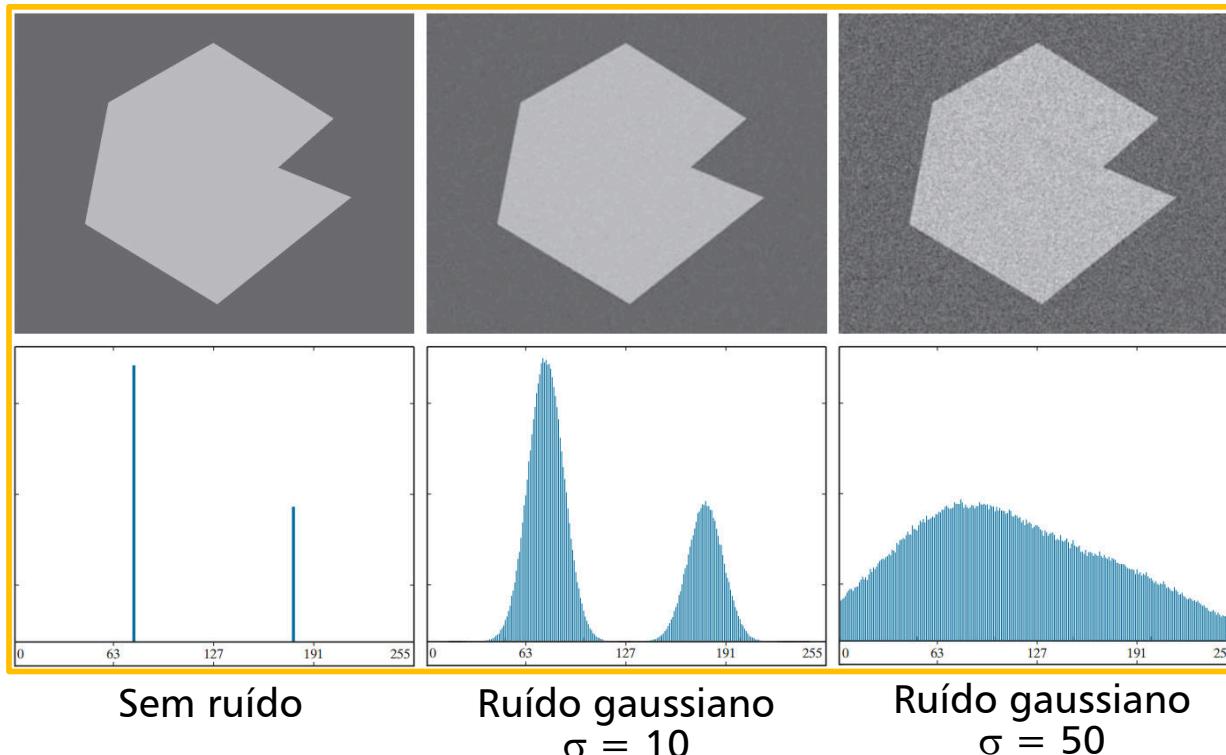
Limiarização global

$$g(x, y) = \begin{cases} a & \text{se } f(x, y) > T_2 \\ b & \text{se } T_1 < f(x, y) \leq T_2 \\ c & \text{se } f(x, y) \leq T_1 \end{cases}$$

Limiarização múltipla

Limiarização

A limiarização é sensível ao índice de ruído da imagem, ao tamanho relativo dos objetos e do fundo, à uniformidade da fonte de iluminação e à uniformidade das propriedades de reflexão dos objetos imageados.



Limiarização

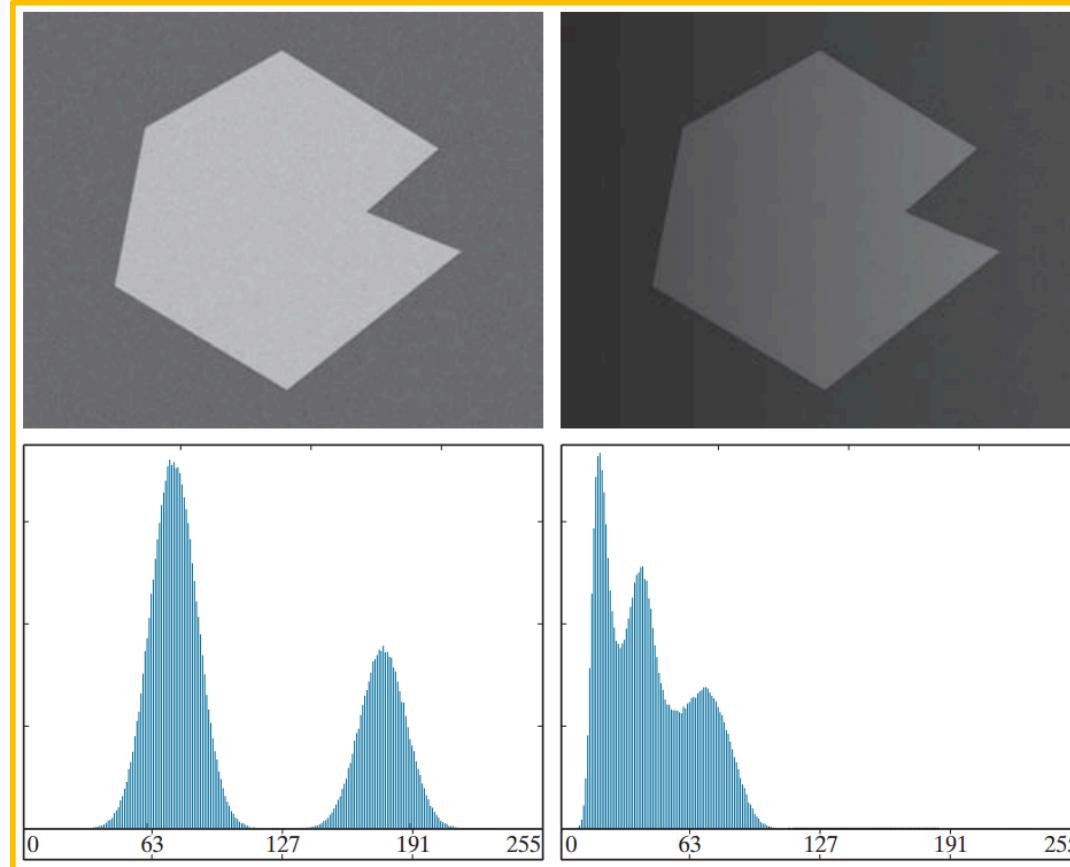


Imagen ruidosa
com iluminação
constante

A mesma imagem
ruidosa com
iluminação variável

Efeito do
iluminação na
limiarização

Limiarização Global Simples

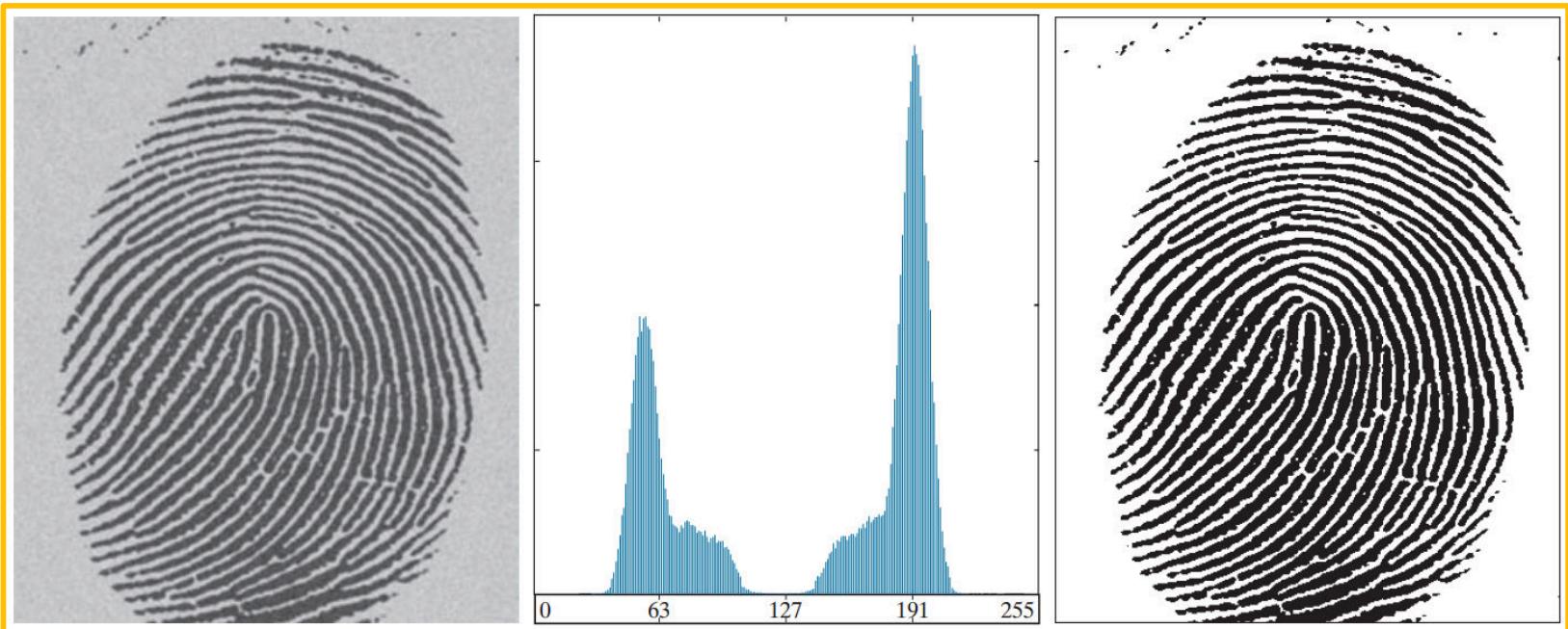
Método usando quando os objetos e o fundo da imagem são suficientemente distintos”.

- 1) Selecionar uma estimativa inicial para o limiar global T ;
- 2) Segmentar a imagem usando

$$g(x, y) = \begin{cases} G_1, & \text{se } f(x, y) > T \\ G_2 & \text{se } f(x, y) \leq T \end{cases}$$

- 3) Calcular os valores de intensidade média m_1 e m_2 , para os grupos G_1 e G_2 , respectivamente;
- 4) Calcular um novo valor de limiar $T = \frac{1}{2}(m_1 + m_2)$;
- 5) Repetir as etapas 2 a 4 até que a diferença entre os valores de T em interações sucessivas seja menor que o parâmetro pré-definido ΔT .

Limiarização Global Simples



Impressão digital ruidosa

Histograma

Segmentação usando um limiar global

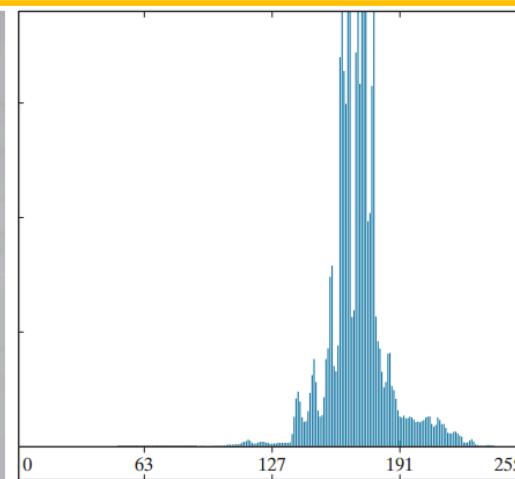
Limiarização Global Ótima (Método de Otsu)

O **método de Otsu** usa a função densidade de probabilidade para determinar um limiar ótimo que maximize a variância entre as classes.

- 1) Construir o histograma da imagem;
- 2) Fazer $\text{MaxVar} = T = \text{Sum} = 0$;
- 3) Calcular $\text{Sum} = \sum_{i=0}^{L-1} (i * f_i)$;
- 4) para i (0 a $L-1$)
 - 4.1) $wb = wb + f_i$;
 - 4.2) se ($wb = 0$) continue;
 - 4.3) $wf = \text{Número de pixels da imagem} - wb$;
 - 4.4) se ($wf = 0$) break;
 - 4.5) $\text{Sumb} = \text{Sumb} + i * f_i$; $mb = \text{Sumb}/wb$;
 $mf = (\text{sum} - \text{Sumb})/wf$; $AVar = wb * wf * (mb - mf)^2$;
 - 4.6) se ($AVar > \text{MaxVar}$) $\text{MaxVar} = AVar$ e $\text{Limiar} = i$.

Limiarização Global Ótima (Método de Otsu)

Imagen
original



Histograma da
imagen com
os picos
cortados para
visualizar os
detalhes nos
valores mais
baixos

Imagen
segmentada
com um
limiar global
simples



Imagen
segmentada
com um limiar
ótimo obtido
com o método
de Otsu

Usando a Suavização de Imagens para Melhorar a Limiarização Global

Como visto anteriormente, o ruído pode interferir no processo de limiarização. Melhoramos a segmentação aplicando um filtro de suavização na imagem ruidosa.

Imagen
ruidosa

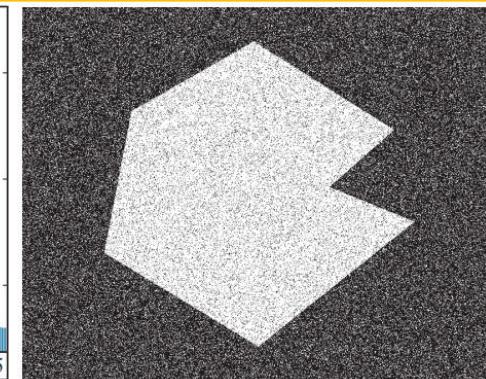
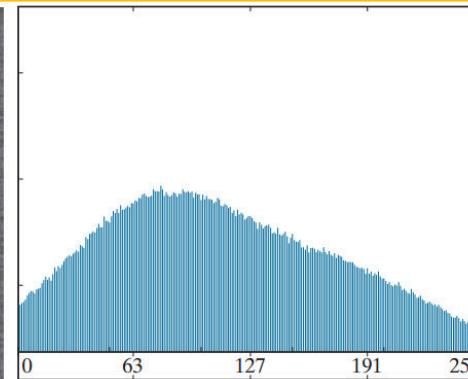
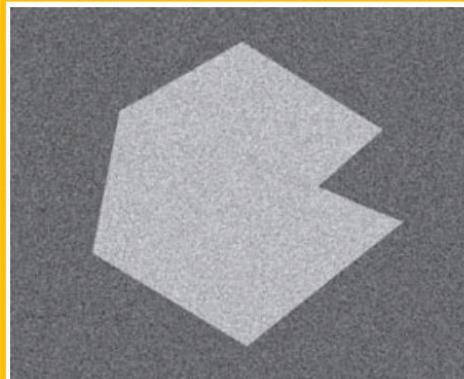
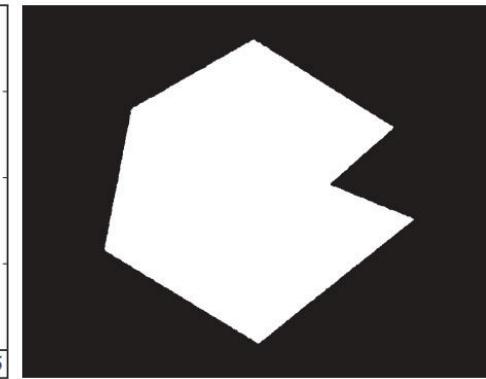
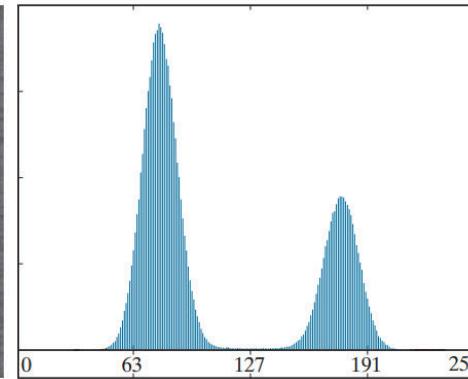
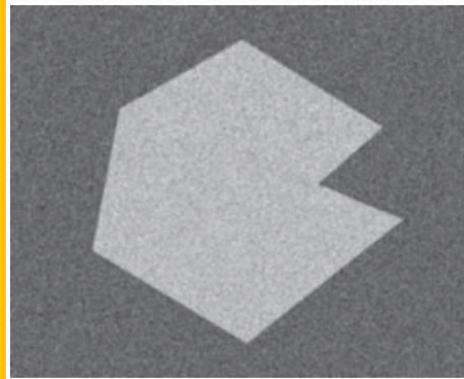


Imagen
filtrada
com
filtro
Box
(5x5)

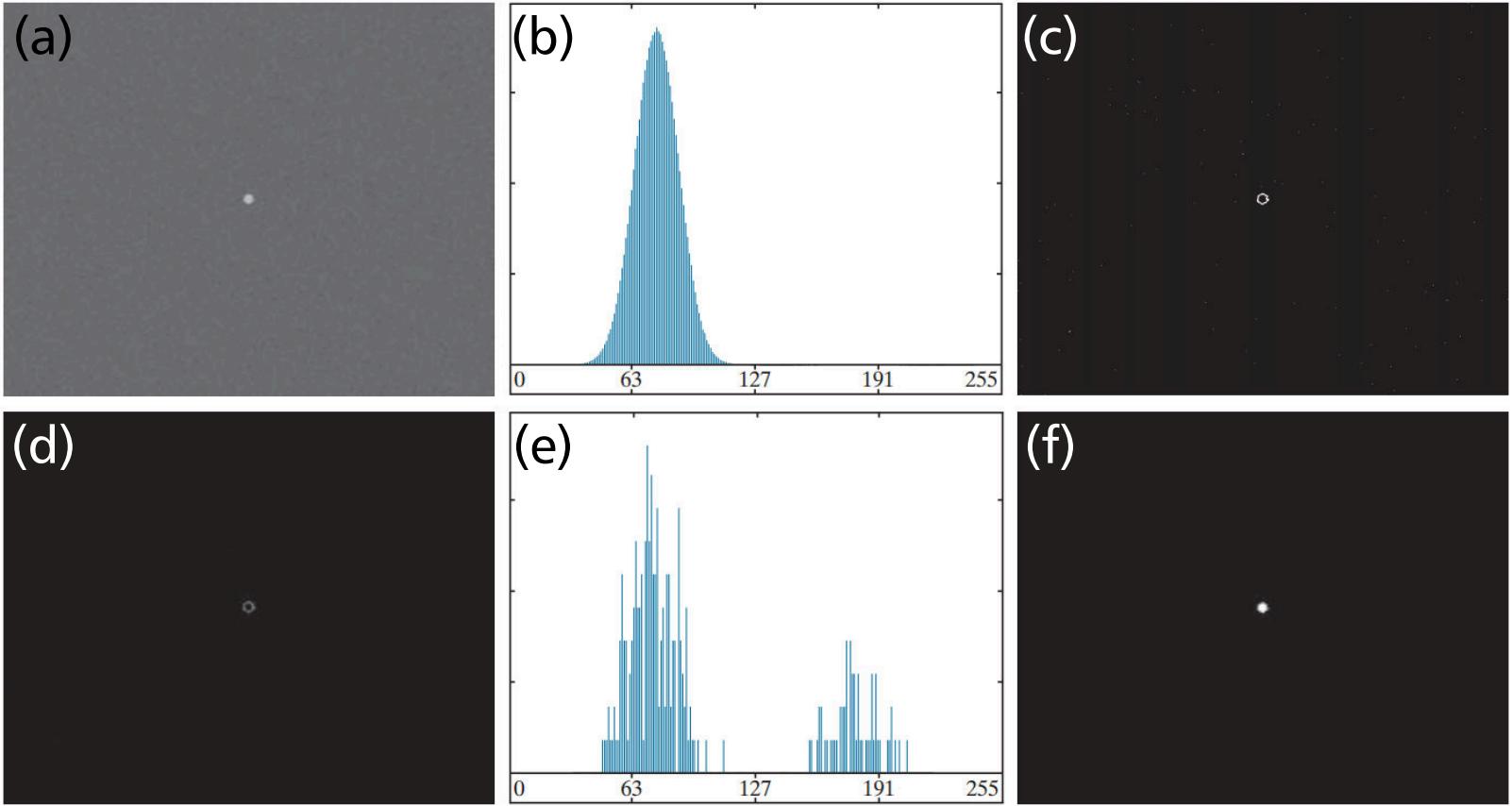


Usando Bordas para Melhorar a Limiarização Global

Quando os objetos são muito pequenos em relação ao fundo pode não ser possível estabelecer um limiar global adequado. O uso das bordas pode melhorar o processo.

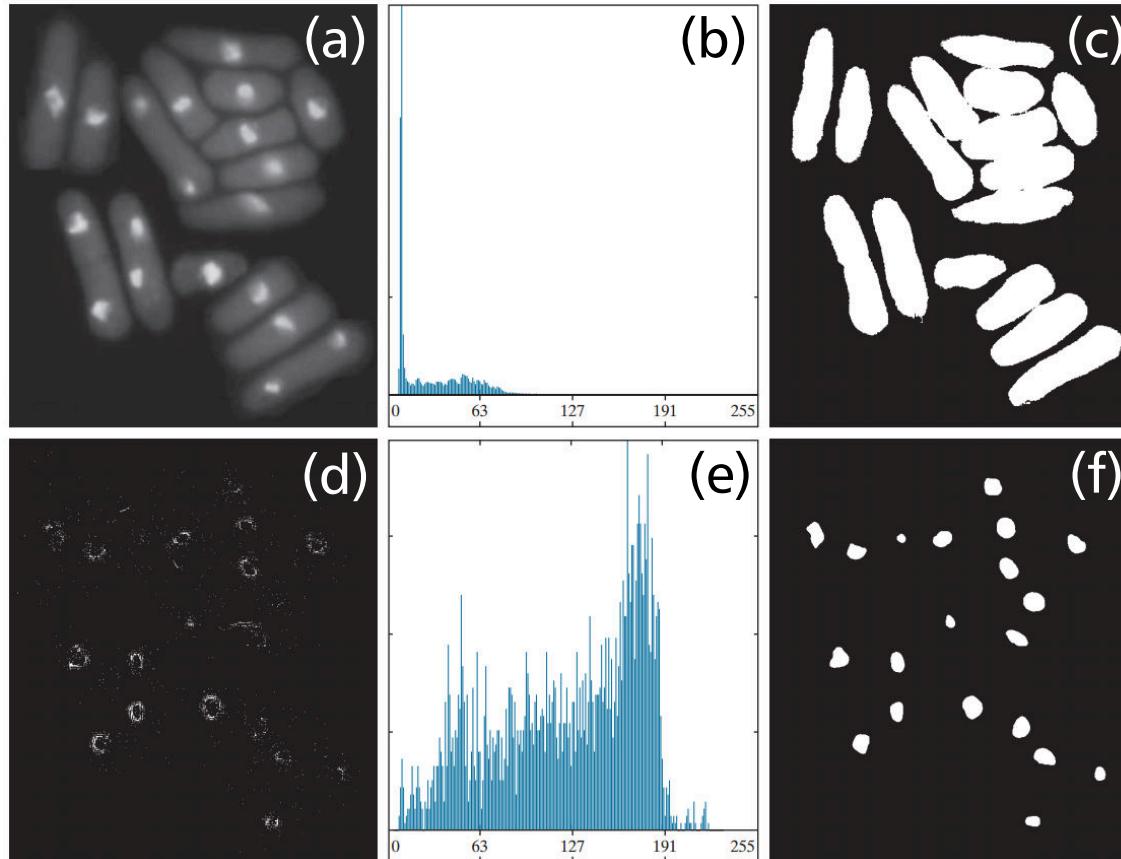
- 1) Calcular uma imagem das bordas $g(x, y)$ a partir da imagem $f(x, y)$ usando a magnitude do gradiente ou o valor absoluto do laplaciano;
- 2) Limiarizar $g(x, y)$ usando um limiar T , gerando uma imagem $g_T(x, y)$. Os pixels com valor 1 em $g_T(x, y)$ correspondem aos pixels “fortes” da borda em $f(x, y)$;
- 3) Construir um histograma que use apenas os pixels de $f(x, y)$ que seja pixels “fortes” da borda;
- 4) Usar o histograma da etapa (3) para obter o limiar global usando, por exemplo, o método de Otsu.

Usando a Magnitude do Gradiente para Melhorar a Limiarização Global



- (a) Imagem ruidosa com um objeto pequeno em relação ao fundo da imagem.
(b) Histograma da imagem. (c) Imagem de $M(x,y)$ limiarizada com percentil 99,7.
(d) Imagem formada pelo produto entre (a) e (c). (e) Histograma dos pixels diferentes de zero em (d). (f) Resultado da segmentação da imagem (a) usando o limiar ótimo encontrado com o método de Otsu ($T=134$) obtido a partir do histograma em (e).

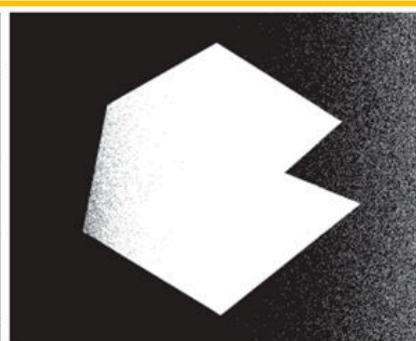
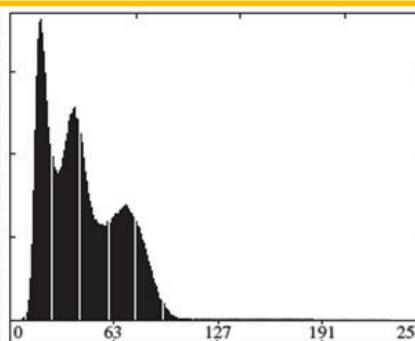
Usando o Laplaciano para Melhorar a Limiarização Global



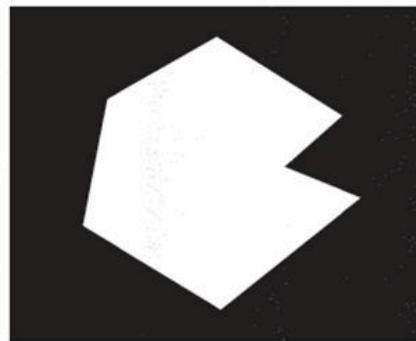
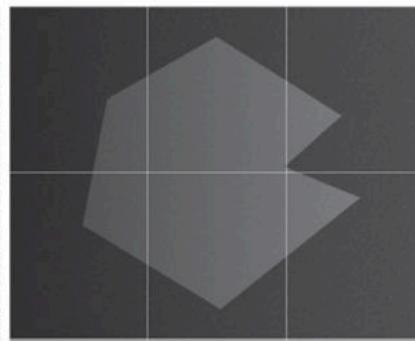
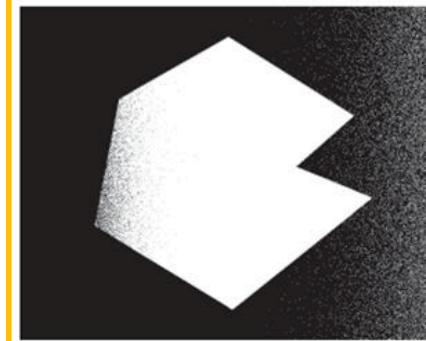
- (a) Imagem das células de levedura. (b) Histograma da imagem. (c) Segmentação de (a) usando o método de Otsu ($T=42$). (d) Laplaciano absoluto limiarizado ($T=115$). (e) Histograma dos pixels diferentes de zero na imagem produto entre (a) e (d). (f) Resultado da segmentação da imagem (a) usando o limiar ótimo encontrado com o método de Otsu ($T=115$) obtido a partir do histograma em (e).

Limiarização Variável

A **limiarização variável** é usada quando há problemas de iluminação, cujo efeito não pôde ser corrigido por métodos de pré-processamento.



Limiarização da imagem com iluminação variável com um limiar global simples e com Otsu



Limiarização da imagem usando o método de Otsu calculado em cada partição

Limiarização Variável – Propriedades Locais

Na limiarização variável baseada nas propriedades locais da imagem um limiar é calculado para cada ponto (x, y) da imagem, com base em uma ou mais propriedades calculadas em sua vizinhança.

Uma abordagem simples desta técnica utiliza a média (m_{xy}) e o desvio padrão (σ_{xy}) dos pixels na vizinhança. O limiar T_{xy} pode ser calculado por:

$$T_{xy} = a\sigma_{xy} + bm_{xy} \text{ ou } T_{xy} = a\sigma_{xy} + bm_g$$

m_g é a média global da imagem. a e b são constantes não negativas. A imagem segmentada é calculada como:

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

Limiarização Variável – Propriedades Locais

Imagen das células de levedura

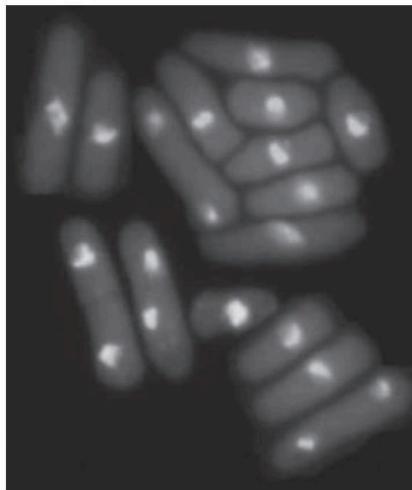


Imagen dos desvios padrões locais (3x3)

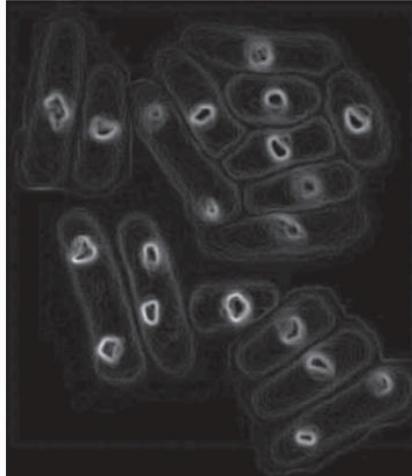
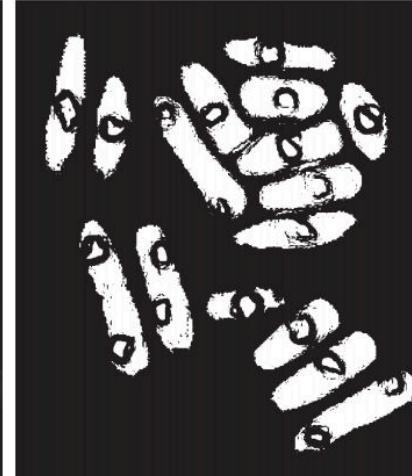


Imagen segmentada usando duplo limiar



Imagen segmentada usando limiarização variável usando propriedades locais com $a=0,3$ e $b=0,15$ nos predicados abaixo



$$g(x, y) = \begin{cases} 1 & \text{se } [f(x, y) > a\sigma_{xy} \text{ e } f(x, y) > bm_g] \\ 0 & \text{caso contrário} \end{cases}$$

Limiarização Variável – Médias Móveis

Na limiarização variável baseada em médias móveis o limiar é calculado com base na média de uma janela linear com n pixels que se move ao longo de cada linha da imagem.

$$m(k+1) = \frac{1}{n} \sum_{i=k+2-n}^{k+1} z_i, \text{ para } k \geq n-1$$

$$m(k+1) = m(k) + \frac{1}{n} (z_{k+1} - z_{k-n}), \text{ para } k \geq n+1$$

n é o número de pontos usados no cálculo da média móvel. $m(1)=z_1$. Nas bordas a média móvel é calculada com os pixels disponíveis. A limiarização é dada por:

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

c é uma constante positiva e m_{xy} é a média móvel na janela em (x, y) .

Limiarização Variável – Médias Móveis

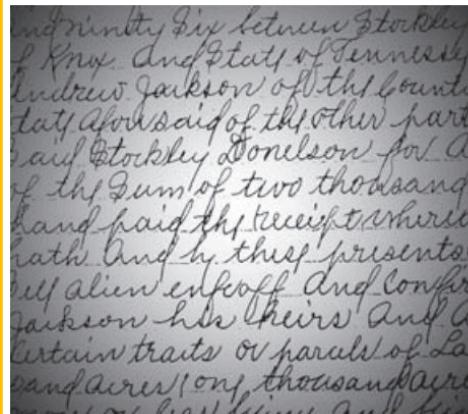
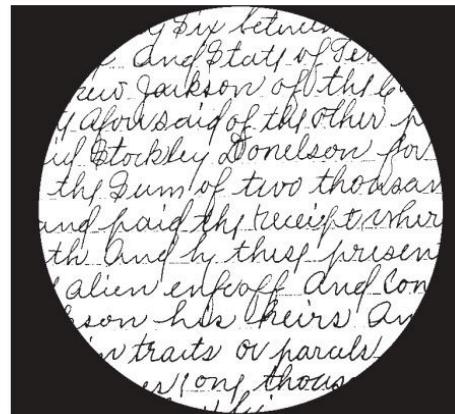
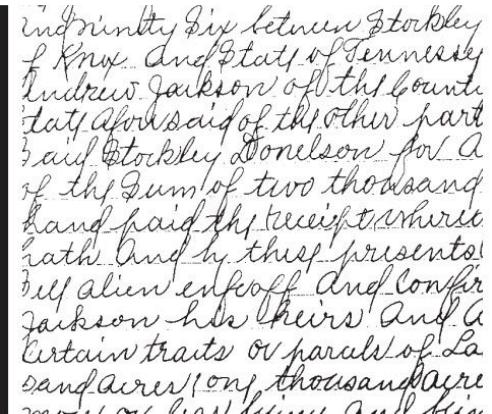


Imagen de texto corrompida com sombreamento pontual



Resultado da limiarização usando o método de Otsu



Limiarização usando médias móveis, com $n=20$ e $c=0,5$

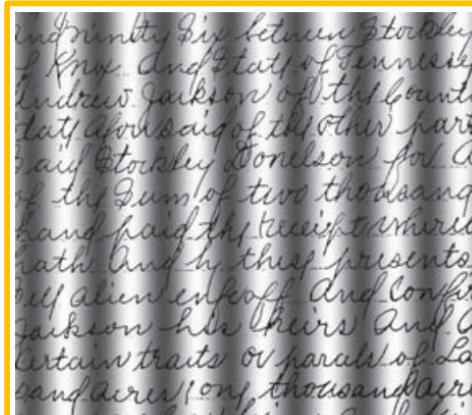
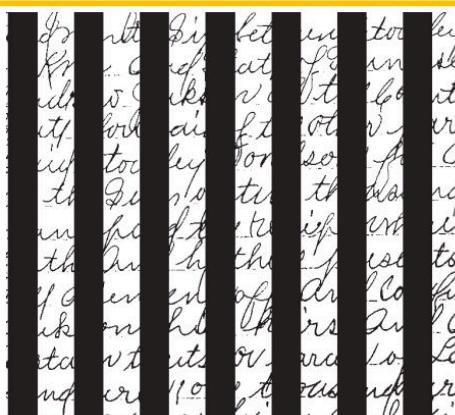
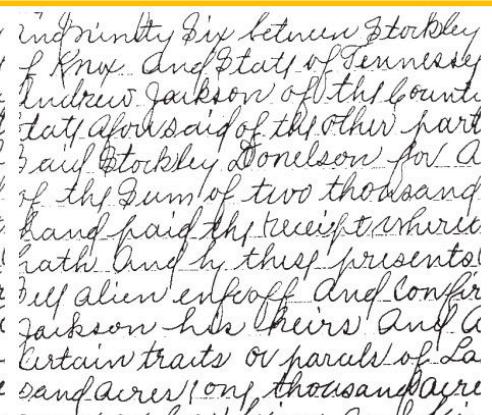


Imagen de texto corrompida com sombreamento senoidal



Resultado da limiarização usando o método de Otsu



Limiarização usando médias móveis, com $n=20$ e $c=0,5$

Segmentação Baseada na Região

Na **segmentação baseada na região** estamos interessados em encontrar as regiões de forma direta, sem usar as informações de bordas ou limiares.

Os métodos tradicionais para segmentar regiões são: crescimento de região, divisão e fusão de região e *watershed* morfológico.

Há outras técnicas empregadas no processo de segmentação baseada na região, mas que não serão aqui estudadas, como: segmentação usando algoritmos de clusterização, segmentação usando superpixels e segmentação usando particionamento ótimo em grafos.

Crescimento de Regiões

Neste método os pixels são agrupados formando regiões maiores, com base em critérios pré-definidos, como vizinhança e similaridade.

O processo se inicia com um conjunto de pontos (sementes) aos quais são anexados os pixels vizinhos semelhantes à cada semente.

Quando não se tem informação à priori sobre os objetos da imagem, a seleção das sementes pode ser aleatória ou definida pelo usuário.

Para que um pixel seja anexado a uma região ele deve estar conectado a ela e possuir um predicado (propriedades) que indicam sua pertença àquela região.

Crescimento de Regiões

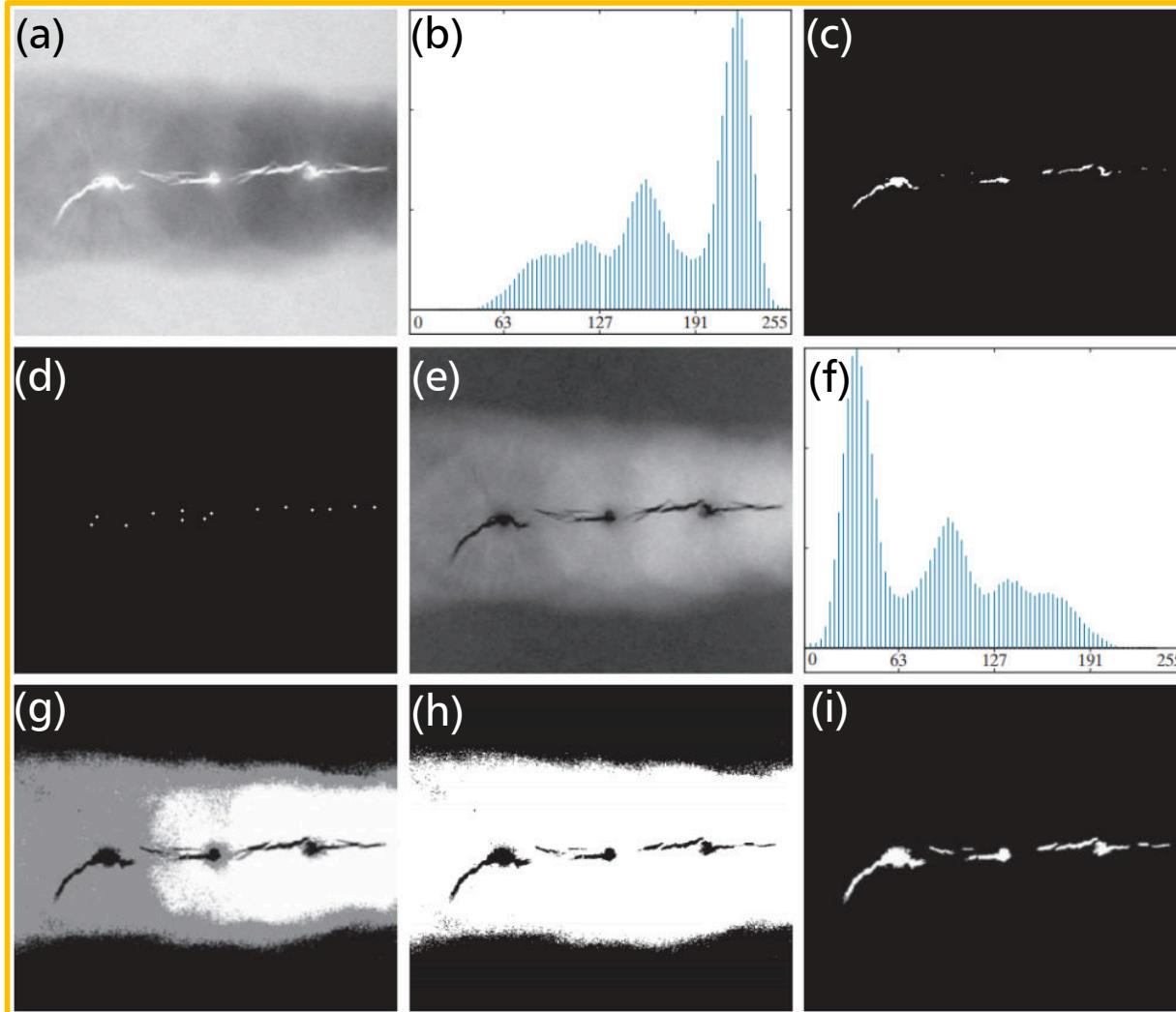
O predicado abaixo representa a similaridade de um pixel (x, y) se sua diferença absoluta em relação à semente da região R for menor ou igual a um limiar T .

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

No predicado de Haralick e Shapiro (1992) uma região é definida em função da média e da variância do conjunto. N é o número de pixels na região, z o pixel testado.

$$P(R) = \begin{cases} V & \text{se } \sqrt{\frac{(N-1)N}{N+1} \cdot \frac{(z - \mu_R)^2}{\sigma_R^2}} \text{ não for valor elevado} \\ F & \text{caso contrário} \end{cases}$$

Crescimento de Regiões



- (a) Imagem de uma solda com falhas, em raios-X
- (b) Histograma da imagem
- (c) Imagem limiarizada com as sementes ($T=254$)
- (d) Imagem erodida até 1 pixel/semente
- (e) $|(a) - (c)|$
- (f) Histograma de (e)
- (g) Imagem (e) limiarizada com dois limiares ótimos (68 e 126)
- (h) Imagem (e) limiarizada com o menor dos dois limiares ótimos (68)
- (i) Imagem segmentada usado crescimento de regiões.

Crescimento de Regiões – Um Algoritmo

- 1) Obter uma imagem $S(x, y)$ das regiões de interesse a segmentar e que contenham as possíveis sementes;
- 2) Encontrar todos os componentes conectados em $S(x, y)$ e erodi-los para o tamanho máximo de 1 pixel. Rotular esses pontos como 1 e os demais como 0. Em $S(x, y)$ restam apenas as sementes;
- 3) Formar uma imagem f_Q onde $f_Q(x, y) = 1$ se $f(x, y)$ satisfaz o critério de similaridade, senão $f_Q(x, y) = 0$;
- 4) Formar a imagem $g(x, y)$ anexando a cada semente em $S(x, y)$ todos os pontos rotulados com 1 em f_Q e que estão 8-conectados à essa semente;
- 5) Rotular cada componente conectado em g com uma diferente etiqueta de região (1, 2, 3, ...).

Os passos 3, 4 e 5 devem ser repetidos para cada semente quando seu valor original afeta a medida de similaridade.

Crescimento de Regiões – Outro Algoritmo

- 1) Obter um conjunto S com as tuplas (x, y) das sementes;
- 2) Criar uma imagem $g(x, y)$ toda nula. $i = 1$;
- 3) Para cada semente $p = (x, y)$ do conjunto S e $g(p) = 0$ faça
 - 3.1) $g(p) = i$ e inserir p em Q (fila)
 - 3.2) Enquanto Q não estiver vazia faça
 - 3.2.1) Remover p de Q ;
 - 3.2.2) Para todo pixel $f(q)$, $q = (x, y)$, pertencente à vizinhança e similar a $f(p)$ e $g(q) = 0$ faça
 - 3.2.2.1) $g(q) = g(p)$;
 - 3.2.2.2) inserir q em Q ;
 - 3.3) $i = i + 1$;

Na imagem $g(x, y)$ estarão rotulados os conjuntos conexos ($i=1, 2, \dots, n$) que cresceram à partir das sementes. Quando duas ou mais sementes forem similares e seus conjuntos conexos forem adjacentes, estas regiões assumirão o mesmo rótulo.

Divisão e Fusão de Regiões

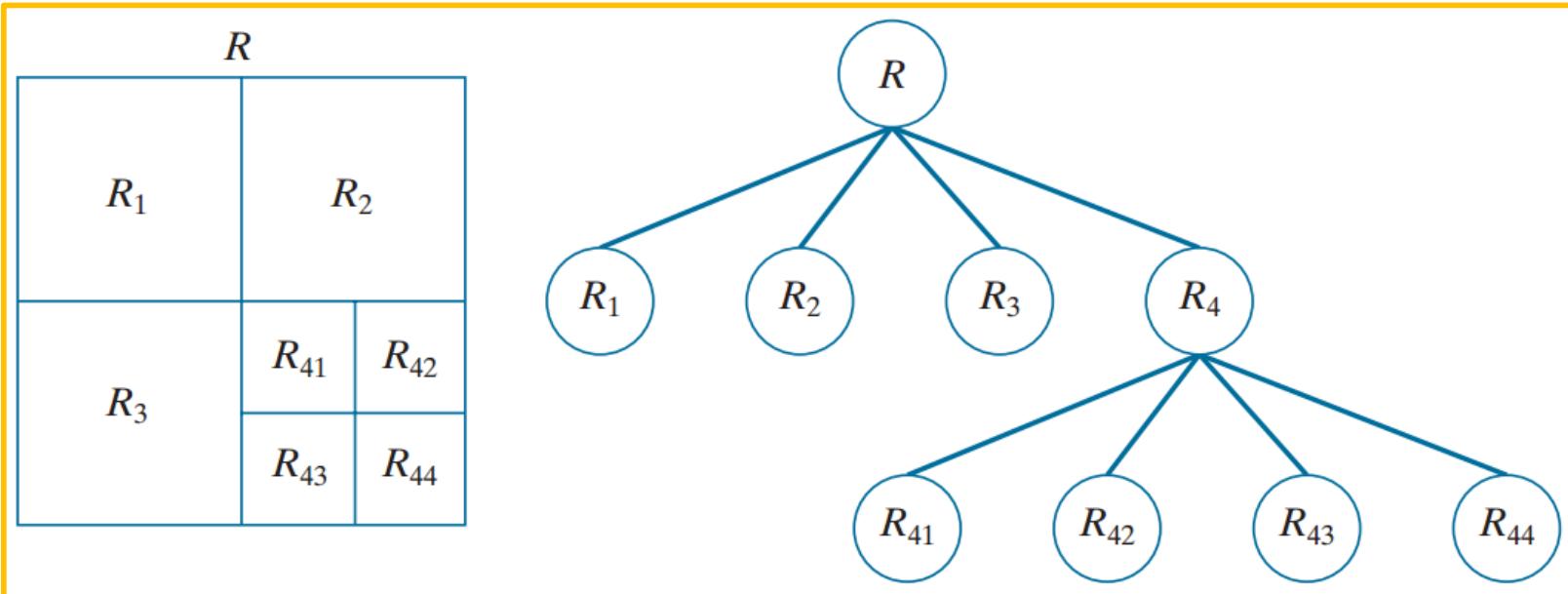
Neste método a imagem é subdividida em um conjunto de regiões distintas e arbitrárias e, em seguida, são fundidas e/ou divididas visando satisfazer os critérios de particionamento vistos no slide 4.

Para dividir ou fundir regiões observa-se a adjacência e o predicado que avalia a similaridade das regiões.

Se todos os pixels de uma região R_i não atendem ao predicado ($P(R_i) = F$), então R_i deve ser dividida. Normalmente dividimos a região em 4 sub-regiões, dando origem a uma quadtree.

Caso duas sub-regiões adjacentes levem a $P(R_i \cup R_j) = V$, então as sub-regiões R_i e R_j serão unidas, cessando a divisão desta sub-região.

Divisão e Fusão de Regiões



O processo de divisão vai continuar enquanto o $P(R_i) = F$ ou até que a região atinja um tamanho mínimo e não possa mais ser dividida.

A segmentação termina quando não for mais possível dividir regiões ou fundir regiões adjacentes.

Divisão e Fusão de Regiões

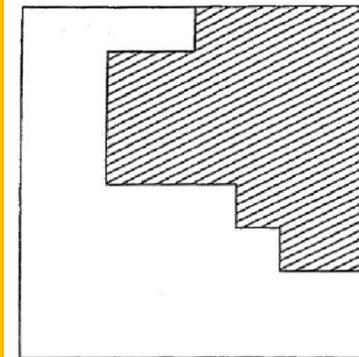


Imagen binária

0	0	0	0	1	1	1	1
0	0	1	1	1	1	1	1
0	0	1	1	1	1	1	1
0	0	1	1	1	1	1	1
0	0	0	0	0	1	1	1
0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Matriz da imagem

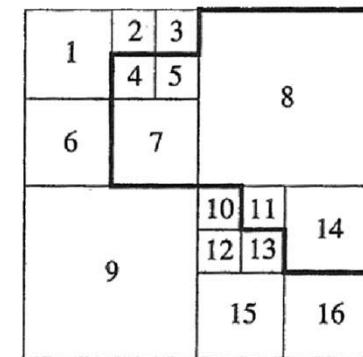
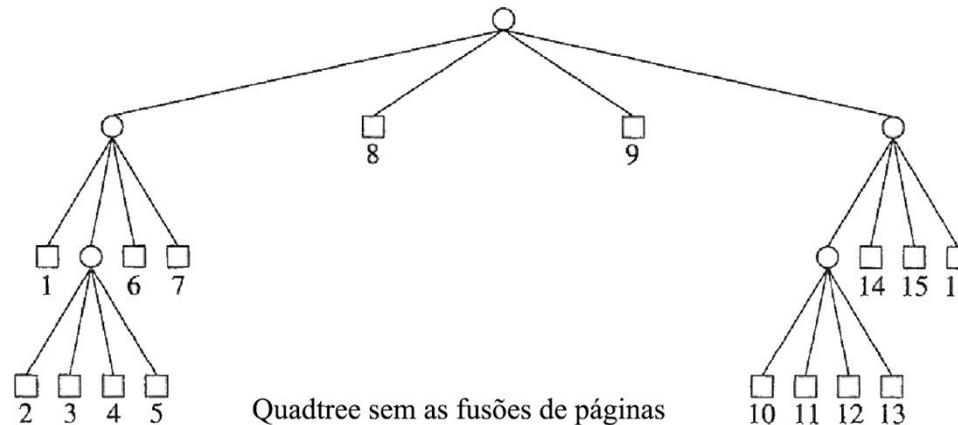


Imagen segmentada



Quadtree sem as fusões de páginas

Para gerar a imagem segmentada, algumas regiões poderiam ser fundidas logo após a divisão (4 e 5) ou ao final do processo (4+5 com 7) quando estão em níveis distintos da quadtree.

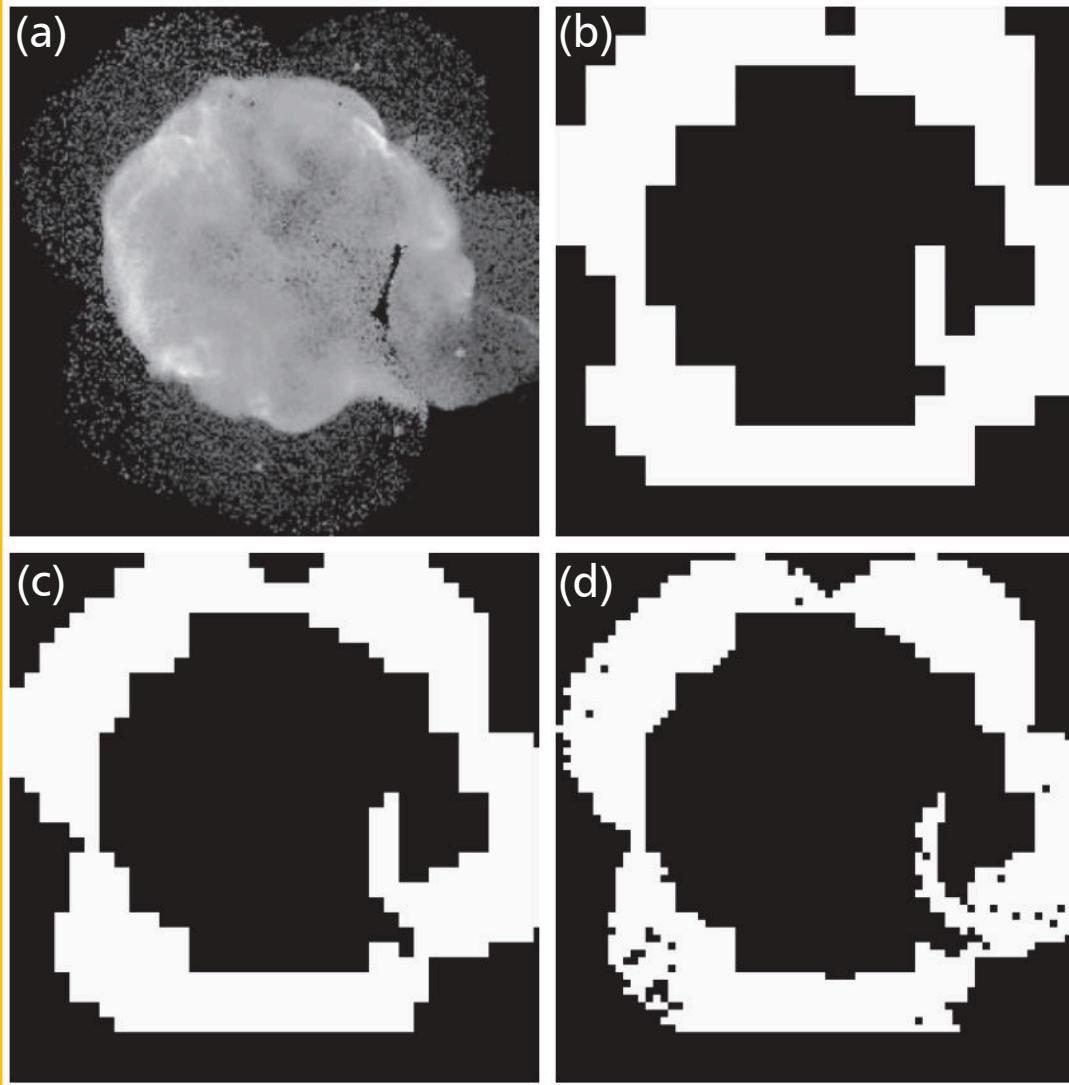
Divisão e Fusão de Regiões – Algoritmo

- 1) Dividir em 4 quadrantes separados qualquer região R_i para a qual $P(R_i) = F$;
- 2) Quando não for possível continuar dividindo, fundir as regiões adjacentes R_j e R_k para as quais $P(R_j \cup R_k) = V$;
- 3) Parar quando a fusão não for mais possível.

O processo de divisão e fusão de regiões admite diversas variações. Por exemplo, a etapa 2 poderia permitir a fusão de duas regiões adjacentes R_j e R_k se $P(R_j) = V$ e $P(R_k) = V$, o que é mais rápido que verificar $P(R_j \cup R_k) = V$. Esta simplificação ainda é capaz de produzir bons resultados de segmentação.



Divisão e Fusão de Regiões



Objetivo: segmentar o anéis de matéria menos densa em torno do núcleo denso da supernova Cygnus Loop.

O predicado usado na segmentação foi:

$$P = \begin{cases} V & \text{se } \sigma_R > a \text{ e } 0 < m_R < b \\ F & \text{caso contrário} \end{cases}$$

$$a = 10 \text{ e } b = 125$$

(a) Imagem da supernova Cygnus Loop em raios-X

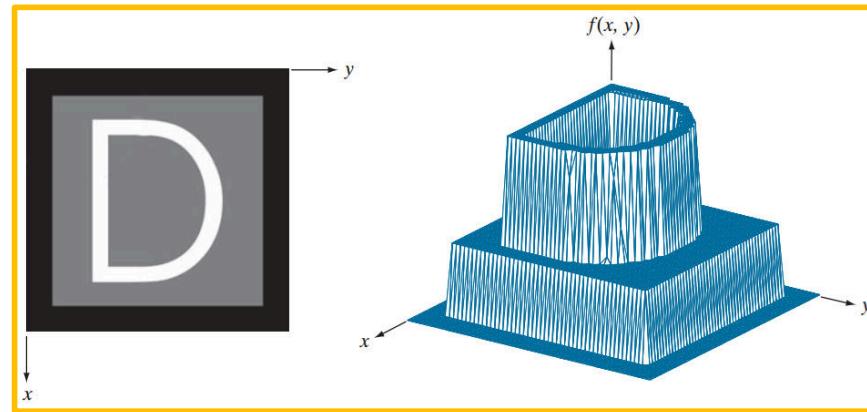
(b) Imagem limiarizada da segmentação com sub-região mínima 32x32

(c) Imagem limiarizada da segmentação com sub-região mínima 16x16

(d) Imagem limiarizada da segmentação com sub-região mínima 8x8 **83**

Watershed Morfológico

O conceito de *watershed* (bacia hidrográfica) baseia-se na visão de uma imagem em 3 dimensões: duas coordenadas espaciais vs intensidade.



Nesta interpretação topográfica há 3 tipos de pixels. a) pontos que definem mínimo regional; b) pontos, onde despejada uma gota d'água, ela escorreria para um dos mínimos; c) pontos que estão sobre o divisor de águas.

Watershed Morfológico

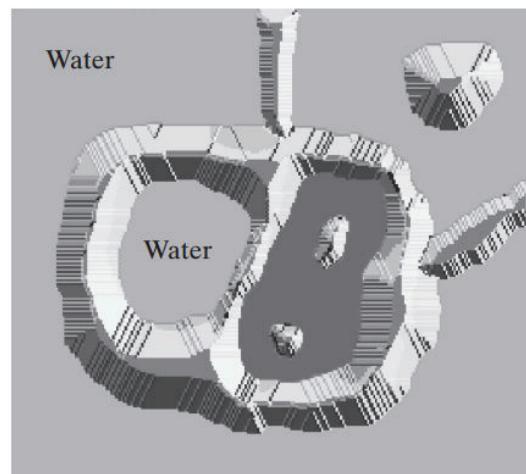
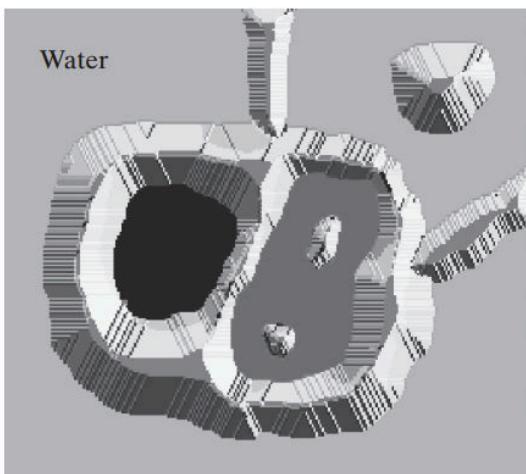
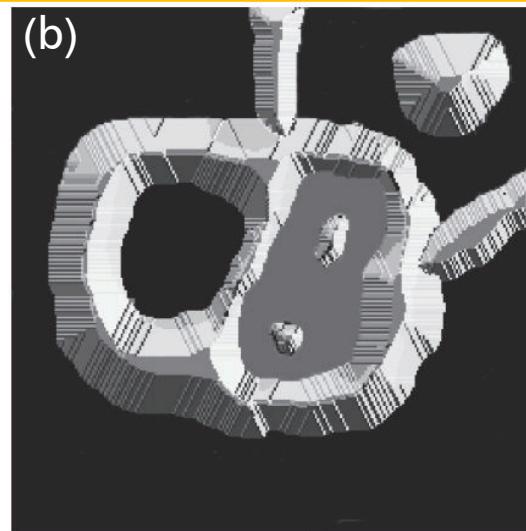
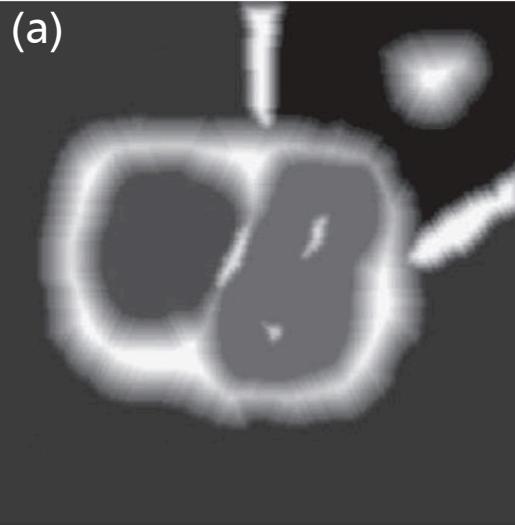
Para um ponto de mínimo regional, todos os pontos do tipo b) formam a bacia hidrográfica ou *watershed* deste ponto de mínimo regional.

Os pontos do tipo c) forma as linhas de *watershed* ou o divisor de águas entre as bacias hidrográficas.

O objetivo do método é encontrar os divisores de água, que definem as fronteiras (conectadas) e que separam as bacias hidrográficas (objetos segmentados)

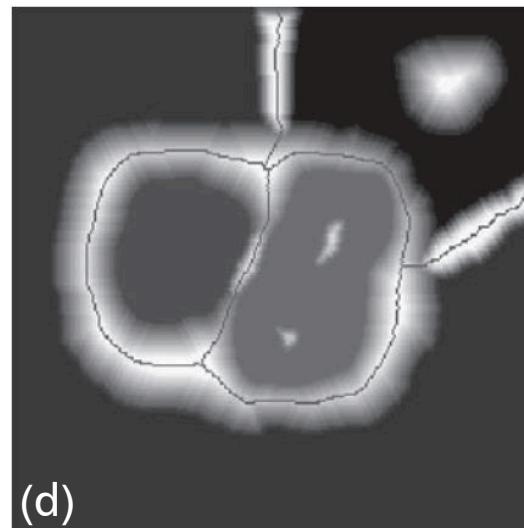
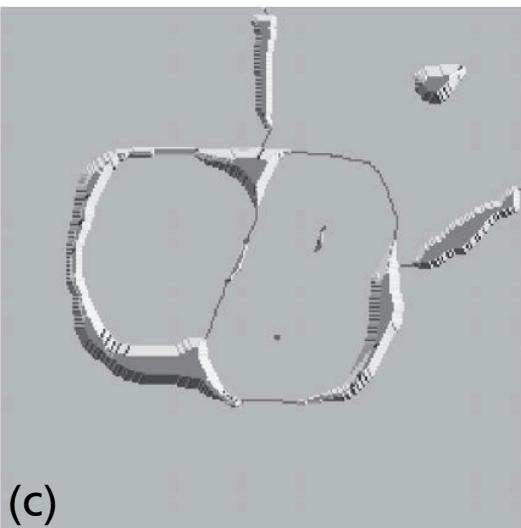
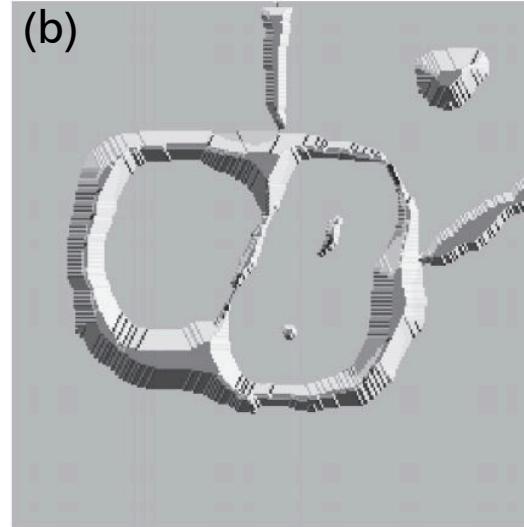
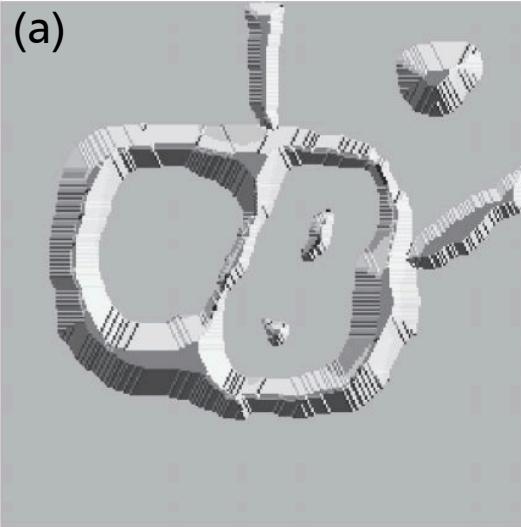
Por analogia, suponhamos que um orifício seja perfurado em cada mínimo regional e que a topografia seja inundada de baixo para cima a uma taxa constante.

Watershed Morfológico



- (a) Imagem original.
- (b) Vista topográfica, com o fundo em preto. A bacia da esquerda possui um tom de cinza escuro, mas não preto.
- (c) Estágio da inundação onde o nível da água ainda não chegou no mínimo das bacias.
- (d) Estágio da inundação onde o nível da água chegou no mínimo regional da bacia esquerda.

Watershed Morfológico



Aplicação da Segmentação de *Watershed*

A segmentação de *watershed* é adequada para separar os objetos com textura quase uniforme do fundo das imagens.

Quando o gradiente é calculado em regiões uniformes, seus valores são baixos.

Por esse motivo, a segmentação de *watershed* é normalmente aplicada na imagem gradiente, pois os mínimos regionais têm correlação direta com os baixos valores de gradiente dos objetos de interesse.

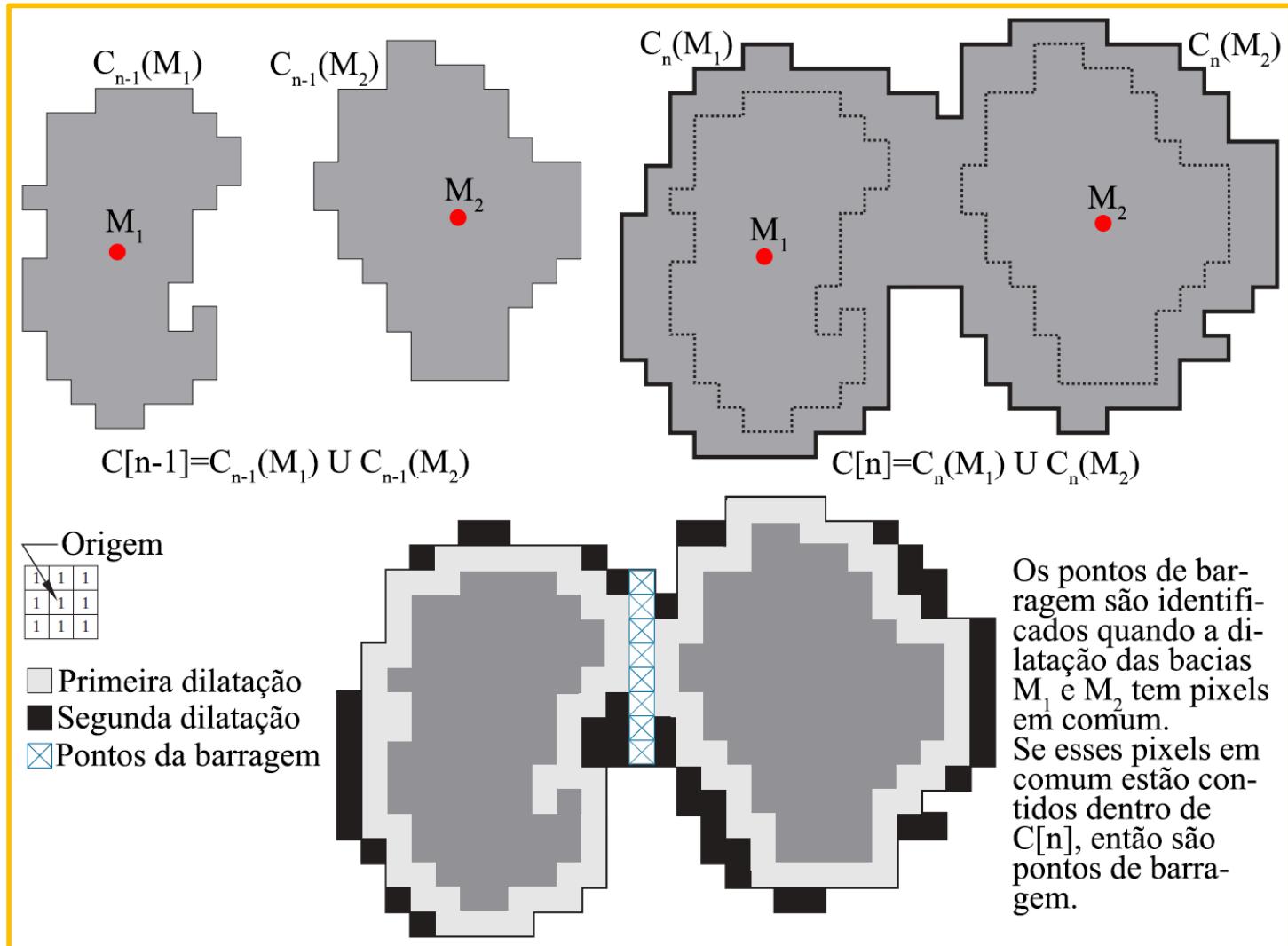
A Construção das Barragens

A construção de barragens acontece quando duas bacias adjacentes estão para se unir devido ao transbordamento.

Assim, deve-se identificar em quais pontos haverá este transbordamento e substituir seus valores por outros que representem uma barragem elevada.

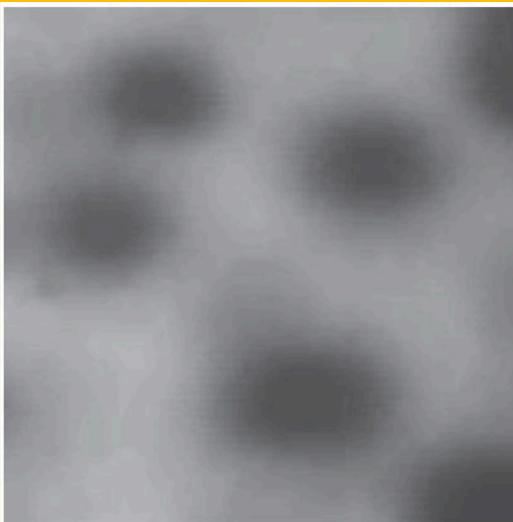
Para identificar o transbordamento devemos manter informações sobre os pixels inundados em cada bacia em dois momentos. No momento $(n-1)$ (anterior) e no momento n (atual).

A Construção das Barragens



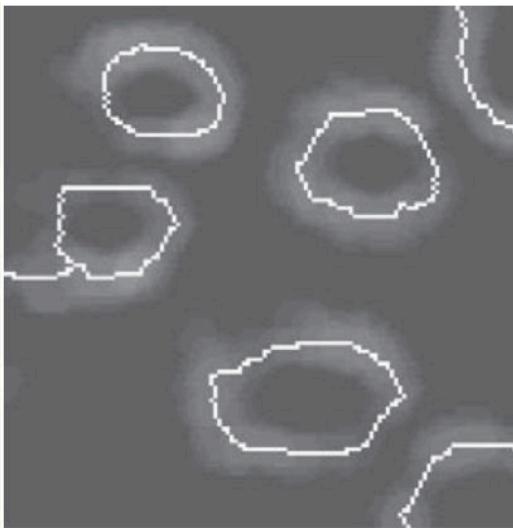
Watershed Morfológico

Imagen
original,
contendo
gotas

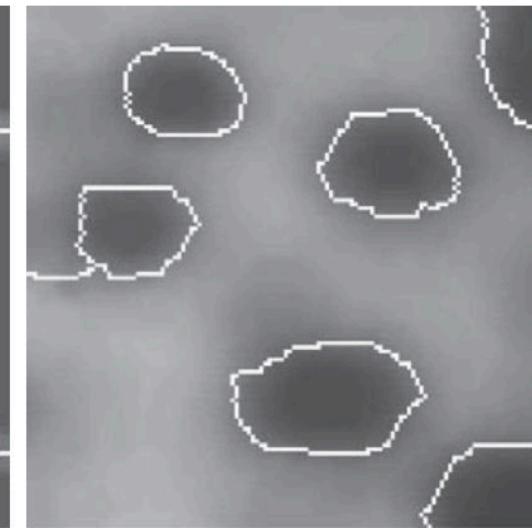
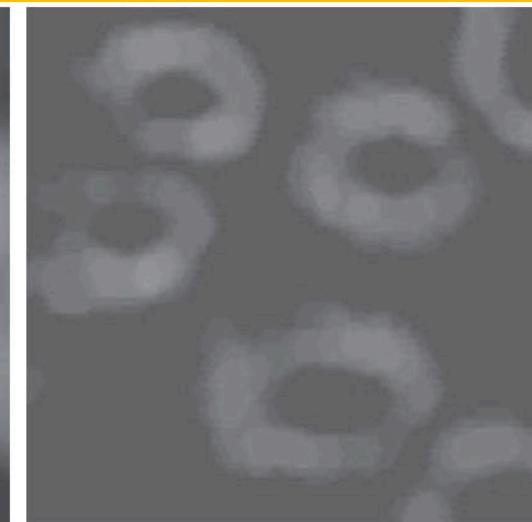


Gradiente
da imagem
original

Linhos de
watershed
obtidas do
gradiente
da imagem
original



Linhos de
watershed
sobrepostos
na imagem
original



Uso de Marcadores na Segmentação de Bacias

A segmentação de bacias é muito sensível ao ruído e outras irregularidades locais do gradiente, pois esses fatores levam a um número elevado de mínimos locais.

O excesso de mínimos locais conduz a um processo de super-segmentação, que torna a imagem segmentada praticamente inútil.

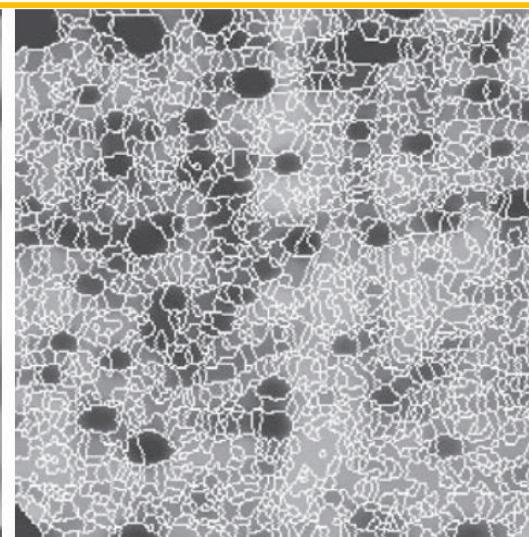
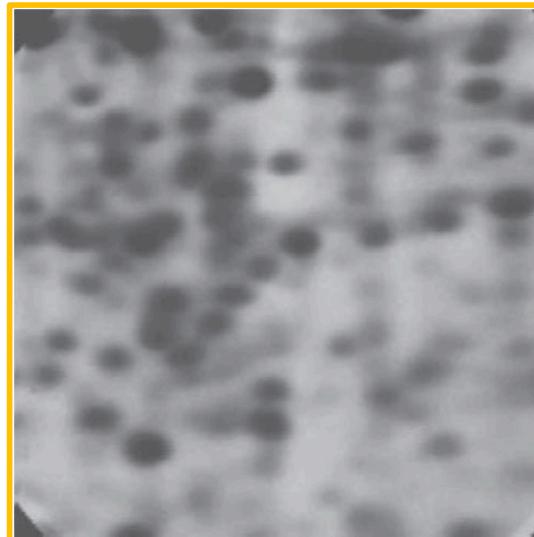


Imagen
super-segmentada

Uso de Marcadores na Segmentação de Bacias

Para contornar o problema da super-segmentação usamos marcadores. **Marcadores** são componentes conectados que pertencem a uma imagem.

Para selecionar os marcadores devemos realizar 2 etapas:

- 1) Pré-processamento;
- 2) Definição dos critérios a serem atendidos pelos marcadores.

A etapa de pré-processamento envolve realizar tarefas para reduzir o número de mínimos locais. Para isso usamos um filtro de suavização sobre a imagem a ser segmentada.

Uso de Marcadores na Segmentação de Bacias

Os critérios definidos para os marcadores representam conhecimento do especialista sobre o que será segmentando.

Estes critérios podem estar baseados em informações simples como similaridade e conectividade, ou informações mais complexas como tamanho, formato, distâncias, texturas, etc.

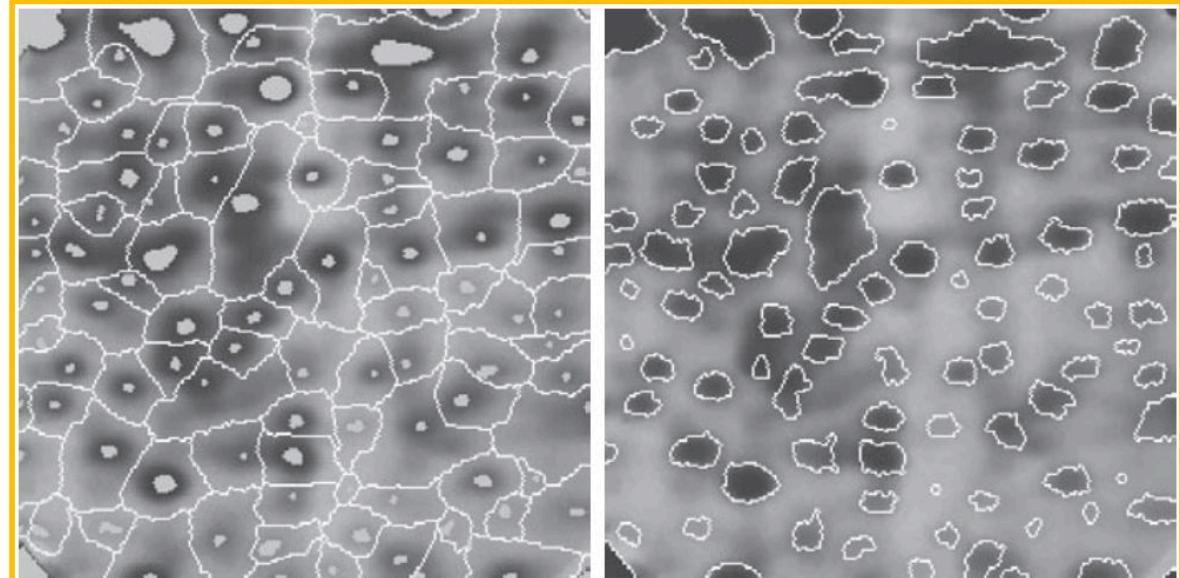
Os marcadores podem estar relacionados aos objetos de interesse (marcadores internos) ou ao fundo (marcadores externos).

Uso de Marcadores na Segmentação de Bacias

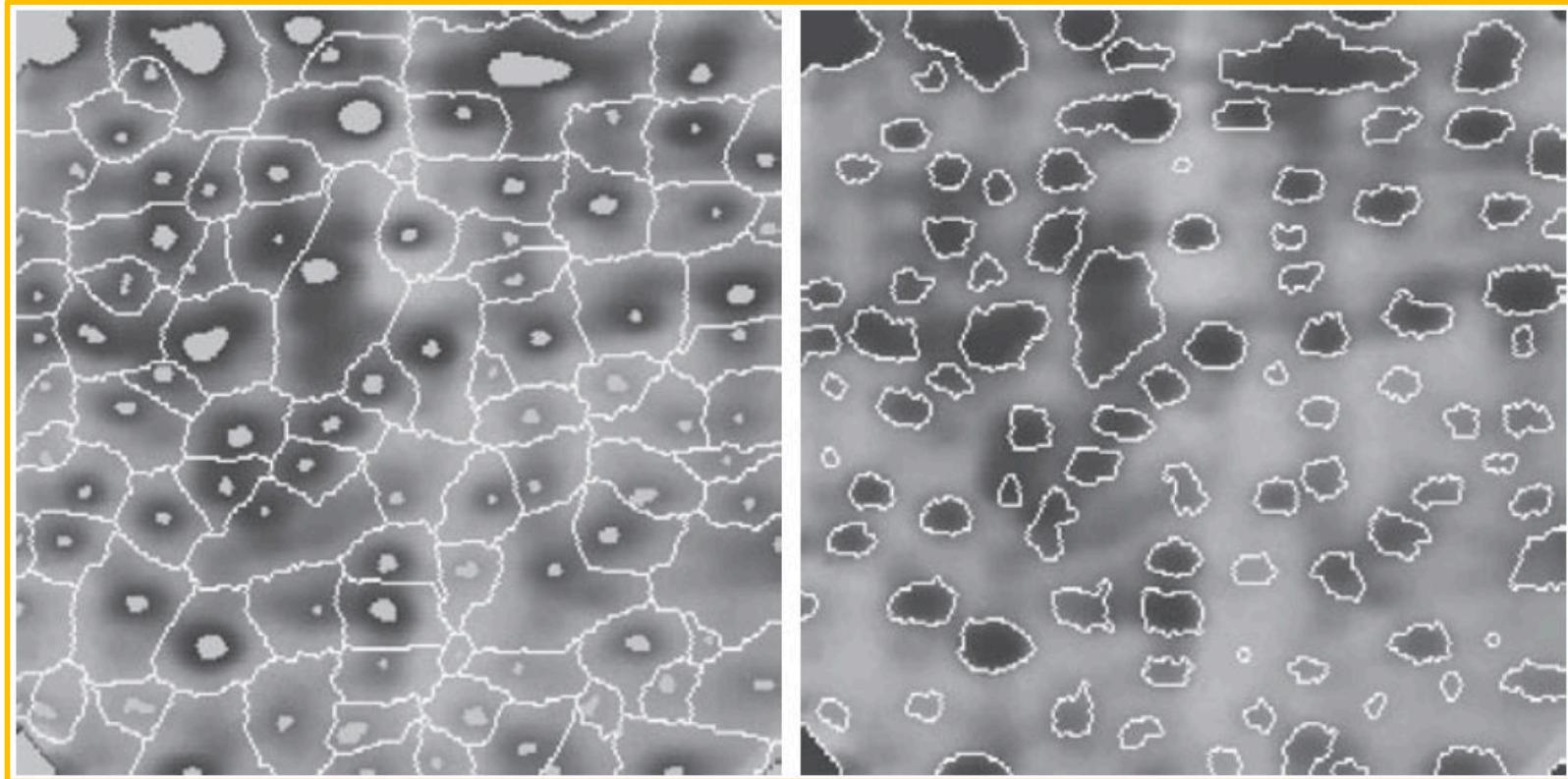
Um exemplo de critérios para definição de marcadores internos pode ser:

- 1) Uma região cercada por pontos de maior altitude;
- 2) O marcador é um componente conectado;
- 3) Os pontos do marcador tem a mesma intensidade.

Os marcadores internos (regiões em cinza claro na imagem à esquerda) definem os mínimos regionais para o processo de watershed, realizado sobre a imagem original suavizada. As linhas de barragem obtidas definem marcadores externos.



Uso de Marcadores na Segmentação de Bacias



As regiões delimitadas pelos marcadores externos (linhas de barragem) pode ser segmentadas individualmente, usando das técnicas apresentadas anteriormente.

Neste exemplo, aplicou-se a segmentação por *watershed* sobre o gradiente da imagem original suavizada, em cada região que contém um marcador interno.



Referências

GONZALEZ, R. C.; WOODS, R. E. **Processamento digital de imagens**. 3. ed. São Paulo: Pearson Prentice Hall, 2010.

PEDRINI, H.; SCHWARTZ, W. R. **Análise de imagens digitais: princípios, algoritmos e aplicações**. São Paulo: Thomson Pioneira, 2007.