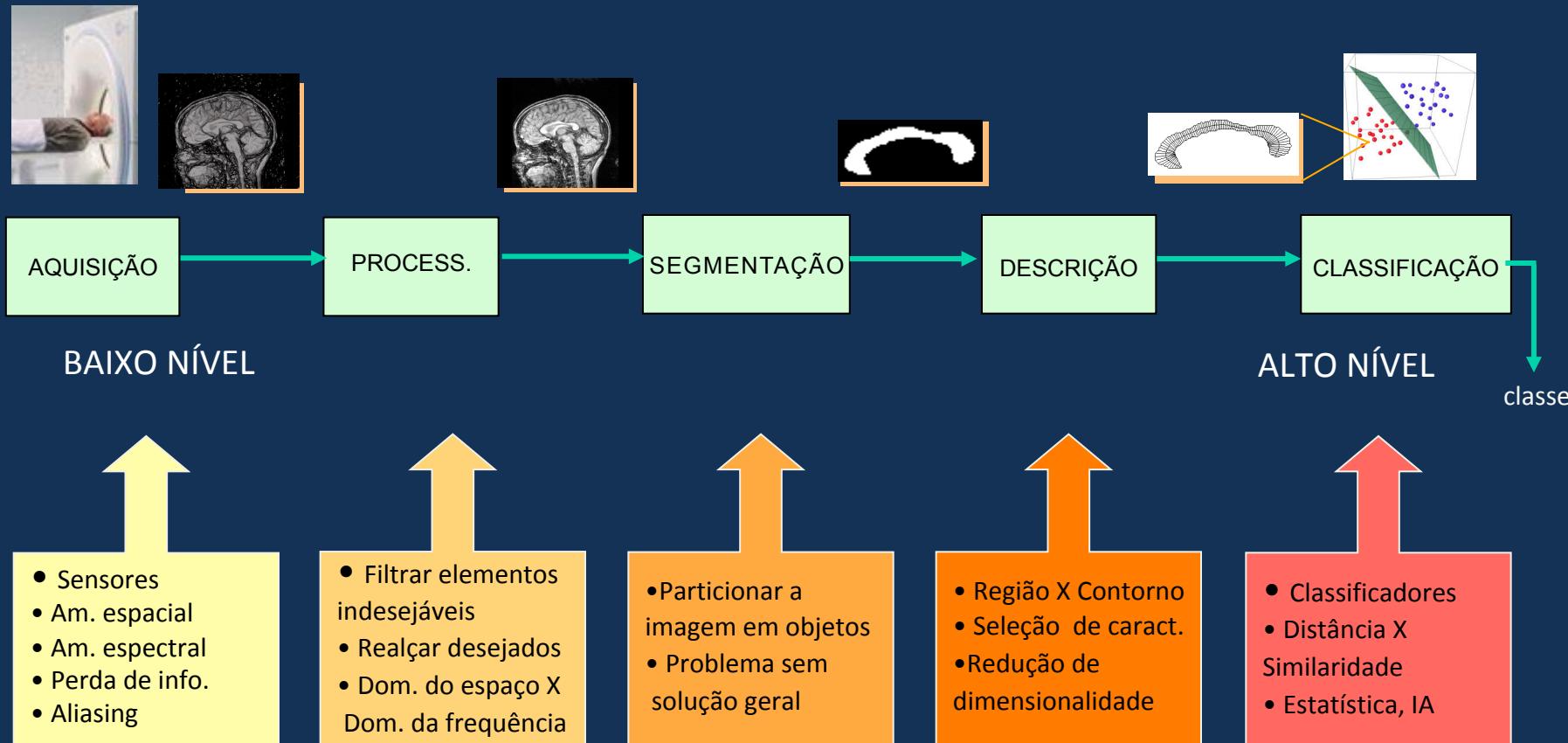


Processamento e Análise de Imagens

Pré-processamento

Prof. Alexei Machado
PUC Minas

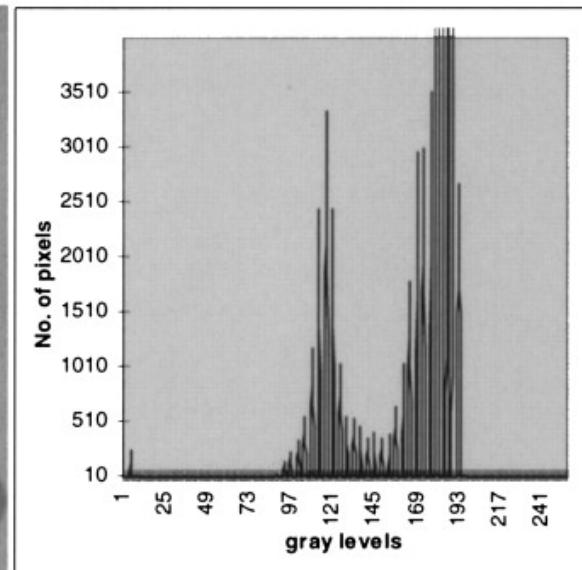
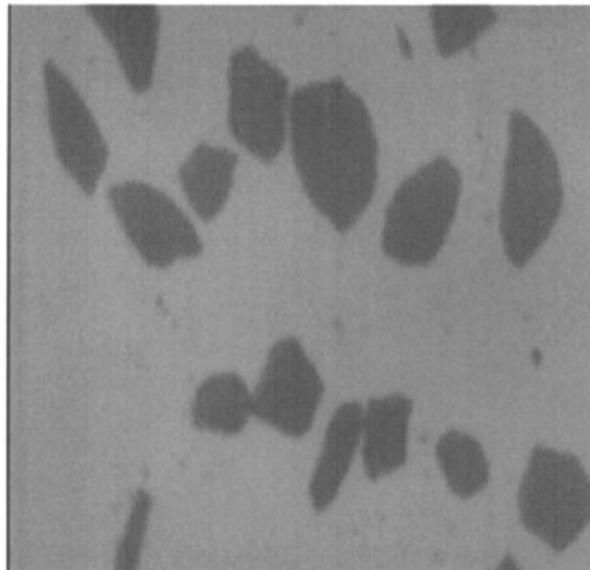
O Processo de Visão Computacional Clássico



Pré-processamento

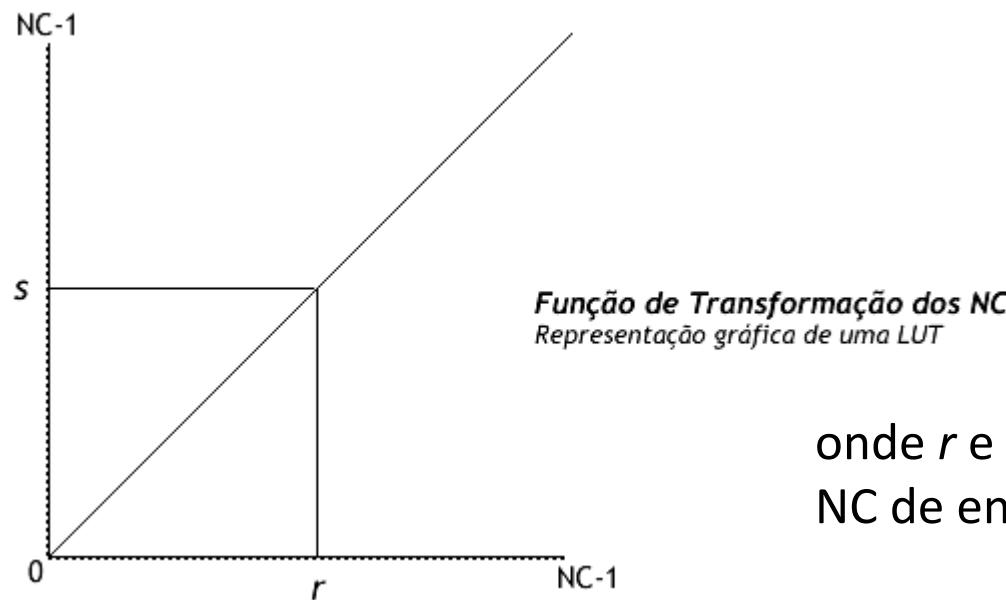
- Melhorar a qualidade da imagem suprimindo elementos indesejados e/ou realçando certas características importantes para o processamento posterior.

Histogramas



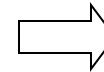
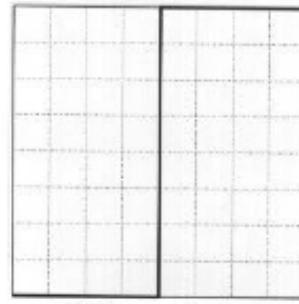
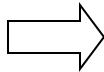
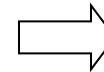
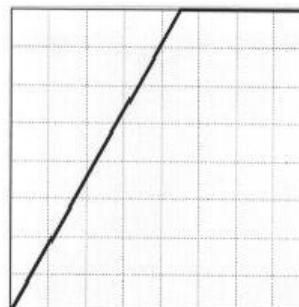
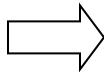
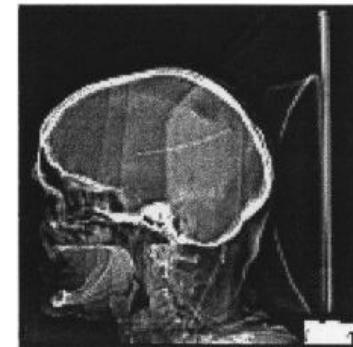
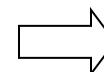
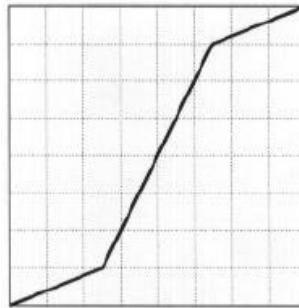
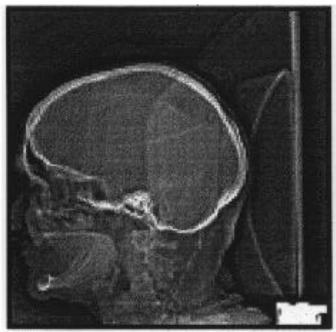
Transformações Radiométricas

- Função de transformação dos níveis de cinza
- Realce simples sobre a Intensidade: expansão de contraste, equalização, etc.

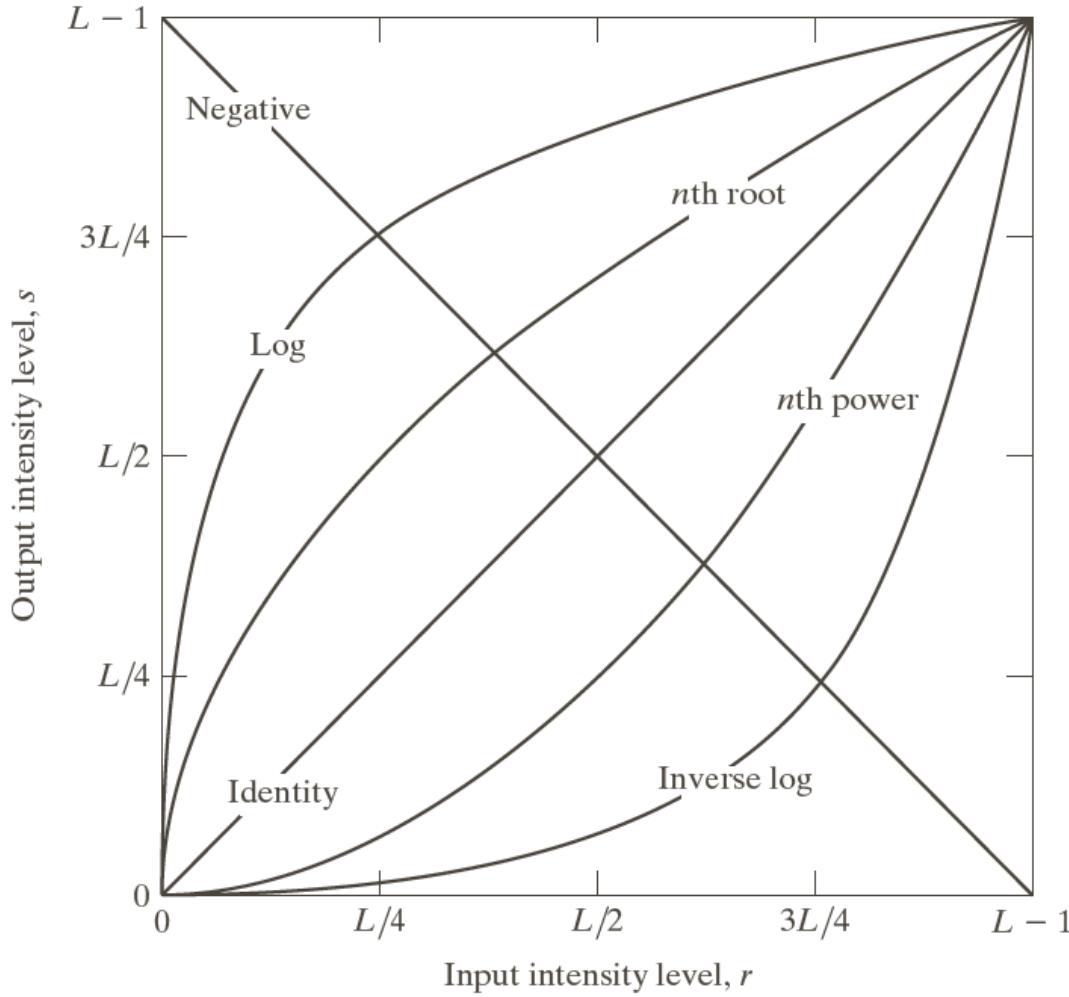


onde r e s denotam os
NC de entrada e saída

Transformações Radiométricas

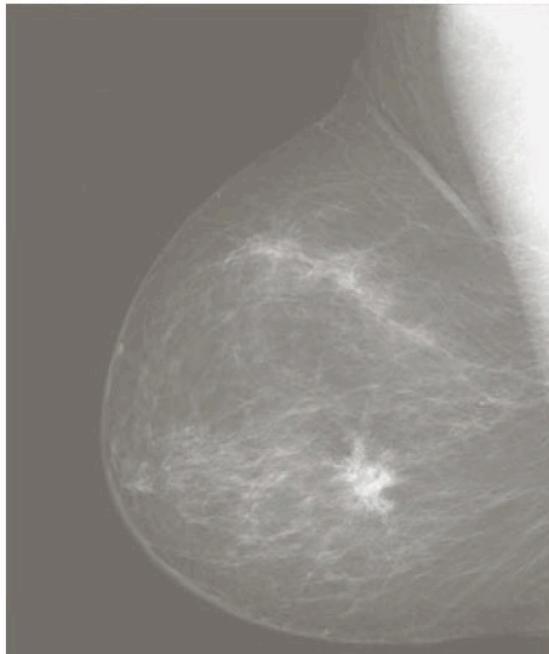


Funções de Transformação



Algumas funções básicas de transformação de intensidade. Todas as curvas foram escaladas para se enquadarem no intervalo mostrado.

Negativo



Mamografia original

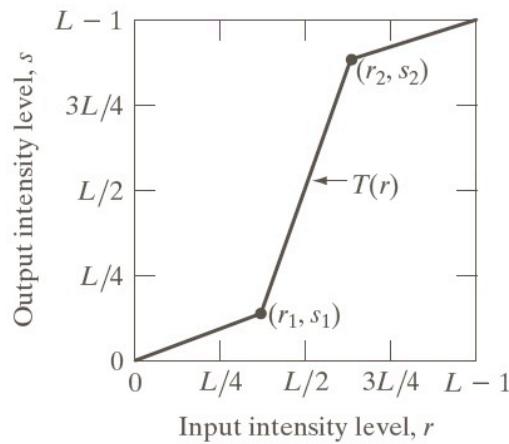


negativo

a b

FIGURE 3.4
(a) Original digital mammogram.
(b) Negative image obtained using the negative transformation in Eq. (3.2-1).
(Courtesy of G.E. Medical Systems.)

Transformação por partes

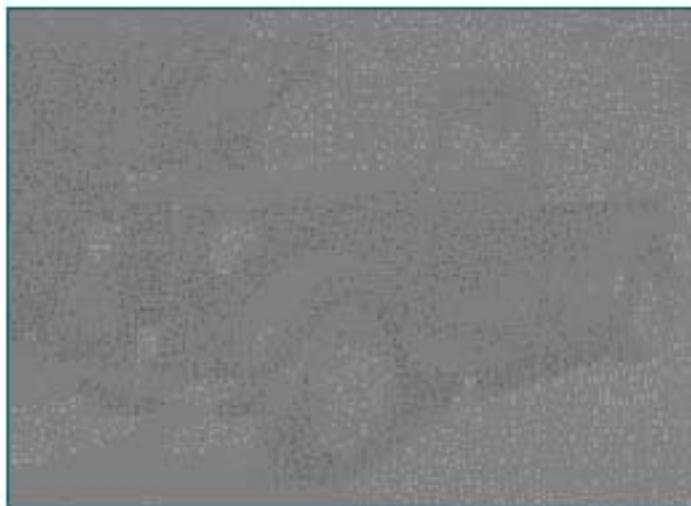


a
b
c
d

FIGURE 3.10
Contrast stretching.
(a) Form of transformation function.
(b) A low-contrast image.
(c) Result of contrast stretching.
(d) Result of thresholding.
(Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

- Extensão do contraste.**
- (a) Forma da função de transformação.**
 - (b) Imagem de baixo contraste.**
 - (c) Resultado da transformação.**
 - (d) Resultado da limiarização (thresholding)**

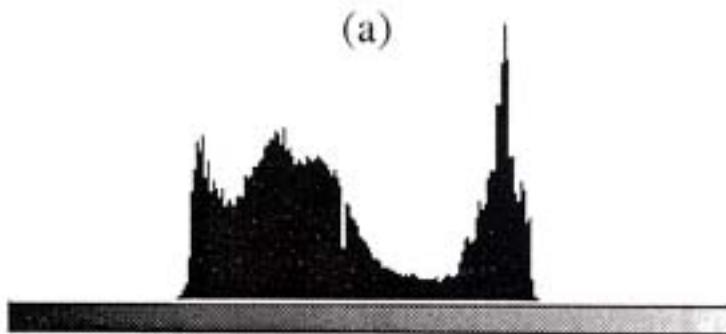
Expansão de Histograma



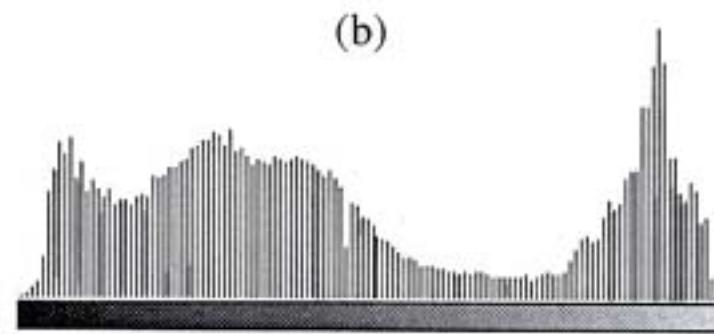
(a)



(b)

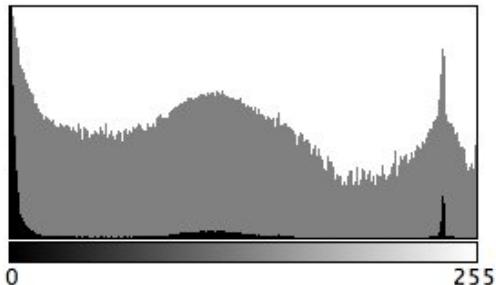
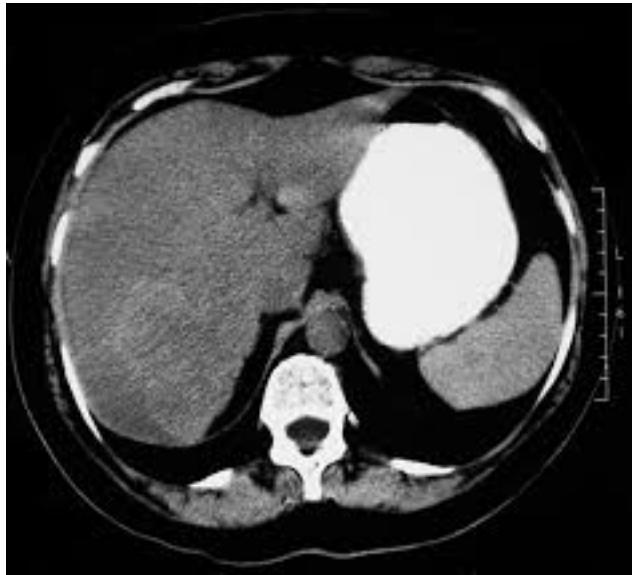


(a)



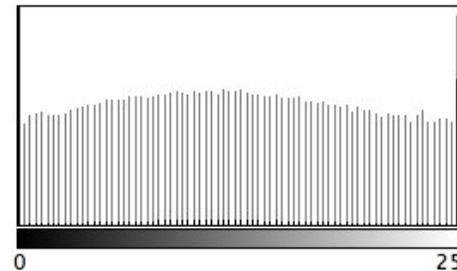
(b)

Janelamento



Count: 50290
Mean: 65.460
StdDev: 78.934

Min: 0
Max: 255
Mode: 0 (8120)



Count: 50290
Mean: 66.174
StdDev: 95.781

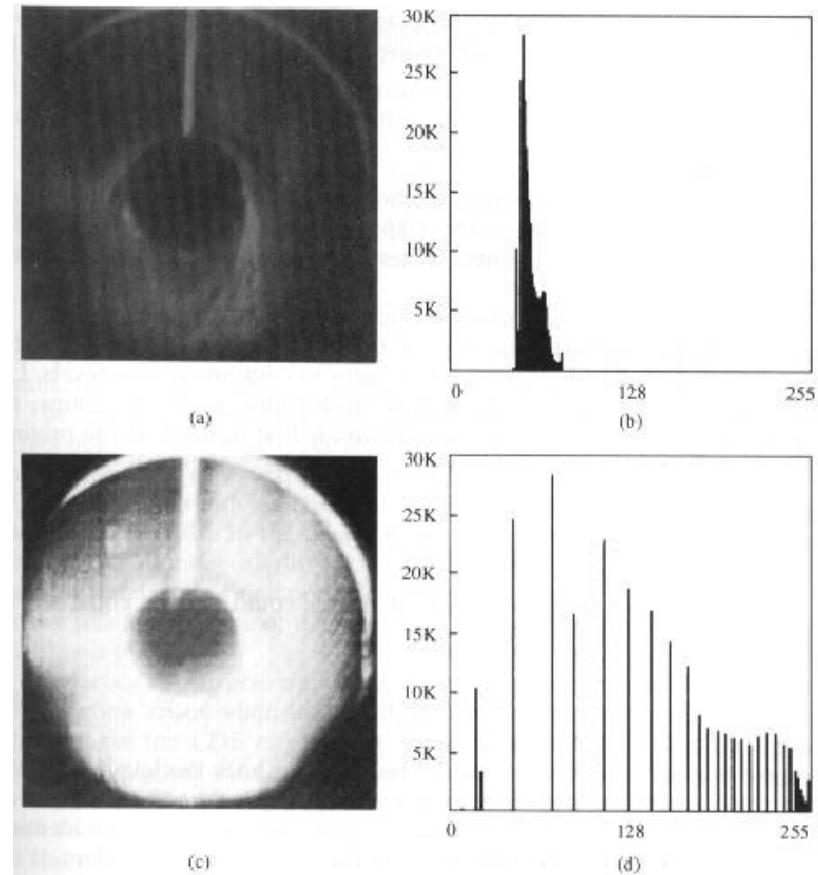
Min: 0
Max: 255
Mode: 0 (30188)

Equalização

$$r = (\text{MAX}/N) \sum_{i=0}^s h(i)$$

onde:

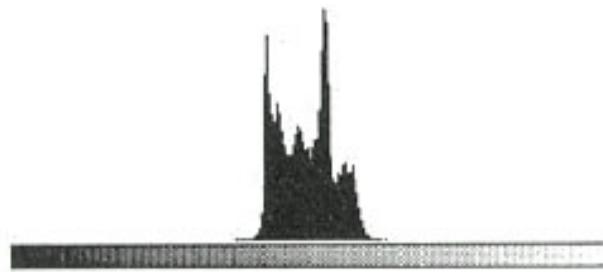
- r é o novo tom de cinza
- s é o tom antigo
- MAX é o novo tom máximo
- N é o tamanho da imagem
- $h(i)$ é o histograma



Equalização



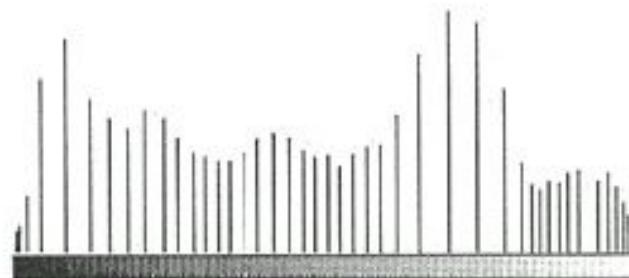
(a)



(b)



(c)

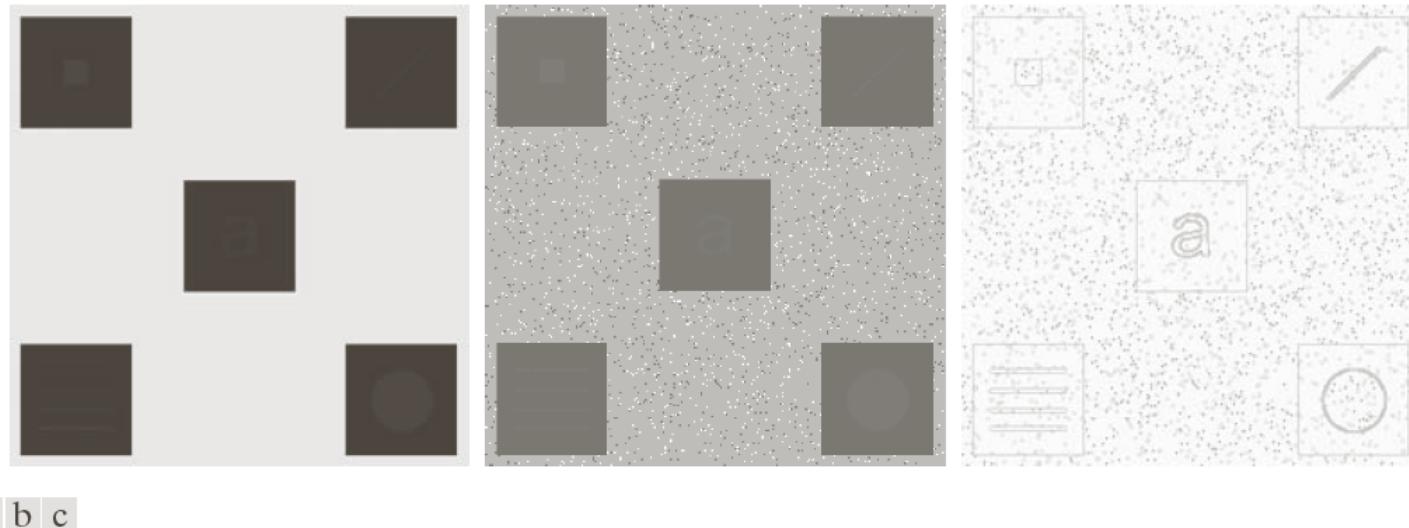


(d)

Processamento Local

- Os métodos de histograma vistos anteriormente são globais, ou seja, os pixels são modificados por uma função de transformação baseada na distribuição de intensidade da imagem inteira.
- Existem casos em que seja necessário melhorar detalhes sobre uma pequena área de uma imagem.
- O procedimento é definir uma vizinhança e mover o centro pixel a pixel. A cada posição, o histograma dos pontos da vizinhança é computado e uma função de transformação de equalização ou de especificação é obtida.
- Essa função é então usada para mapear a intensidade do pixel central da vizinhança.
- O centro da região de vizinhança é então movido a uma das posições adjacentes e o procedimento é repetido.

Processamento Local



a b c

FIGURE 3.26 (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization applied to (a), using a neighborhood of size 3×3 .

- (a) Imagem original.
- (b) resultado da equalização de histograma global
- (c) Resultado da equalização de histograma local em (a), usando uma vizinhança de tamanho 3x3.

Convolução

- Se $T(x,y)$ é uma máscara ($m \times n$) refletida em relação à origem e $I(x,y)$ é a Imagem ($M \times N$)
- A convolução de $T \otimes I$ é dada por:

$$T \otimes I(x, y) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} T(i, j) I(x + i, y + j)$$

Convolução

- Os valores da máscara são chamados de *pesos*.
- Algumas operações simples que podem ser feitas com este tipo de máscara são:
 - redução de ruído do tipo pulso e do tipo sal-pimenta
 - dilatação e erosão
 - borramento (suavização de bordas)
 - detecção de contornos etc



Convolução

As máscaras podem ter tamanhos diferentes mas, em geral, são quadradas e de dimensões ímpares (salvo algumas específicas).

Ex. desejamos computar a média de um pixel $p(x,y)$ usando uma máscara 3x3.

$$p' = \frac{1}{9} * (w_1 * p(x-1, y-1) + w_2 * p(x, y-1) + \dots + w_9 * p(x+1, y+1))$$

w_1	w_2	w_3

Exemplos de Convolução

☞ Filtro Passa-baixa

$$\begin{array}{|c|c|c|c|c|c|} \hline S & S & S & 9 & 9 & 9 \\ \hline S & S & S & 9 & 9 & 9 \\ \hline S & S & S & 9 & 9 & 9 \\ \hline \end{array} \otimes \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} = \begin{array}{|c|c|c|c|c|c|} \hline & & & 5 & 6 & 8 & 9 \\ \hline & & & & & & \\ \hline \end{array}$$

☞ Filtro Passa-alta

The diagram illustrates a convolution operation with padding. An input matrix of size 6x6 is multiplied by a kernel of size 3x3, resulting in an output matrix of size 4x4. The output matrix has a highlighted value '0' at position (2,2).

$$\begin{array}{c} \text{Input Image} \\ \otimes \\ \text{Kernel} \\ = \\ \text{Output Image} \end{array}$$

Convolução e Custo Computacional

- O custo computacional da convolução é alto:
 - Em uma imagem $M \times M$ e um template $N \times N$ são realizadas $M^2 \times N^2$.
 - Se a imagem é de 512×512 e o gabarito é de 16×16 , são necessárias 67108864 multiplicações.
 - Usar processamento paralelo

Filtros Estatísticos

Filtro MAX

Substitui a saída pelo maior valor da máscara. Efeito semelhante ao processo de dilatação

Filtro MIN

Substitui a saída pelo menor valor da máscara. Efeito semelhante ao processo de erosão

Filtro MODA

Calcula-se a frequência de cada NC da máscara e substitui pelo NC de maior frequência

Filtros Estatísticos

Filtro MEDIANA

Ordenam-se os pixels da máscara e substitui-se a saída pelo valor central. Usado na remoção de ruído do tipo sal e pimenta, com poucas perdas na definição das bordas

Filtro MÉDIA (*smooth, borramento, blur*)

Produz efeito de borramento com a suavização das bordas da imagem

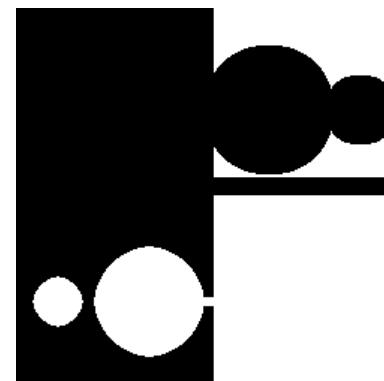
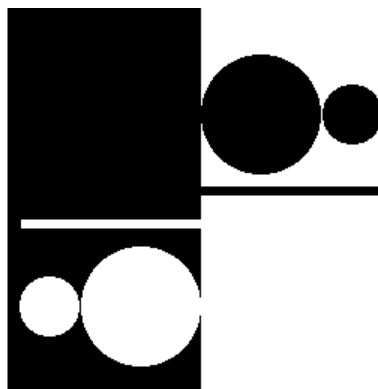
Filtros Estatísticos



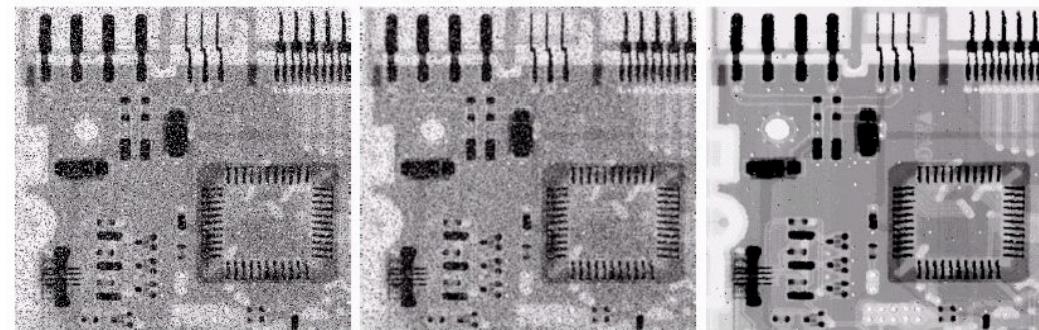
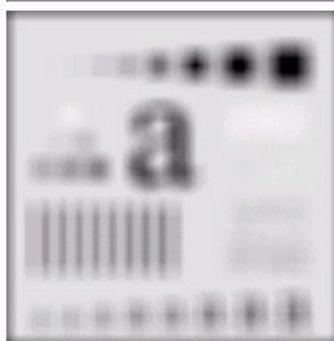
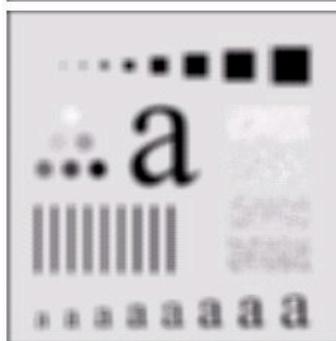
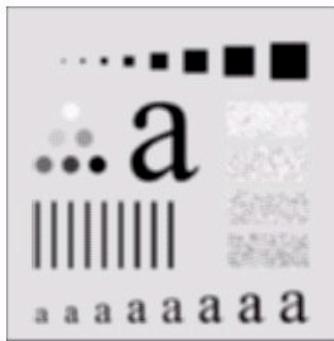
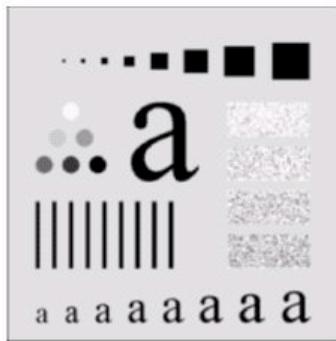
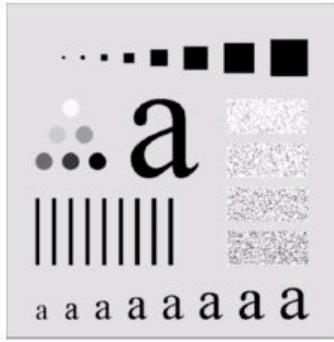
Imagen Original



Imagen Processada



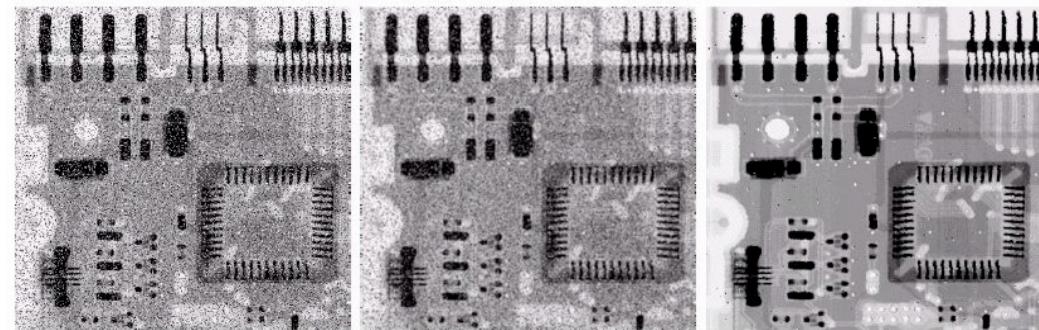
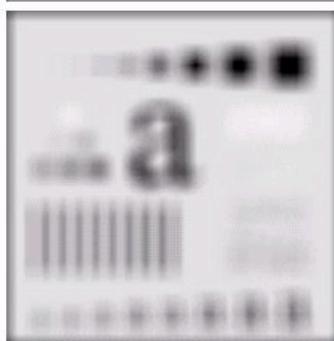
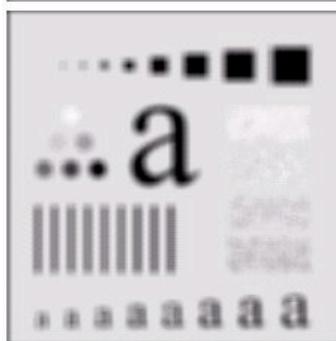
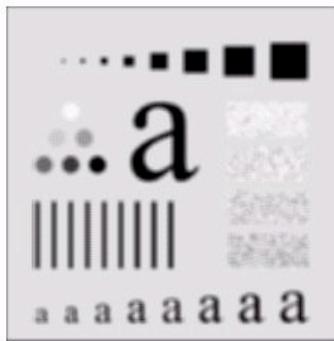
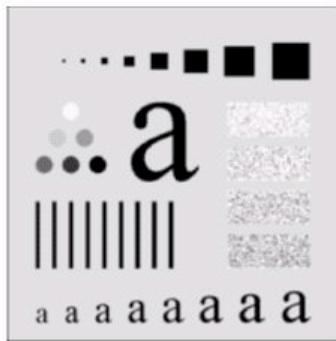
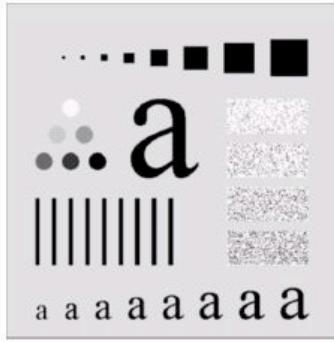
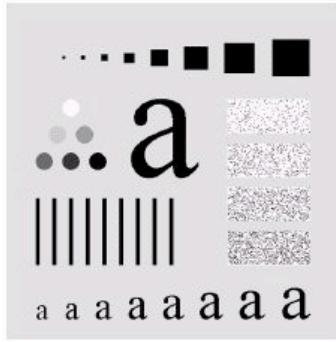
Filtros Estatísticos



Ex. Filtro mediana

Ex. Filtro média (máscara de tamanhos variados)

Filtros Estatísticos



Ex. Filtro mediana

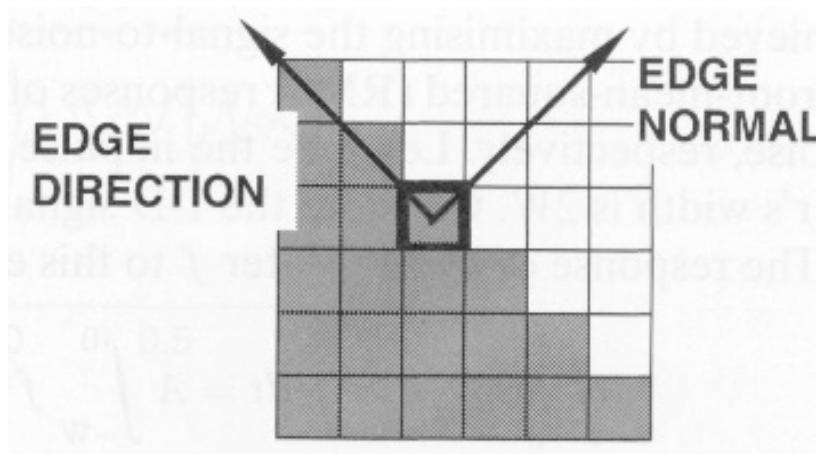
Ex. Filtro média (máscara de tamanhos variados)

Contorno X Borda

- *Edges* (aresta/borda): pixels nos quais há uma mudança brusca na intensidade.
- *Boundary* (contorno): linha fechada formada pelas bordas externas de um objeto

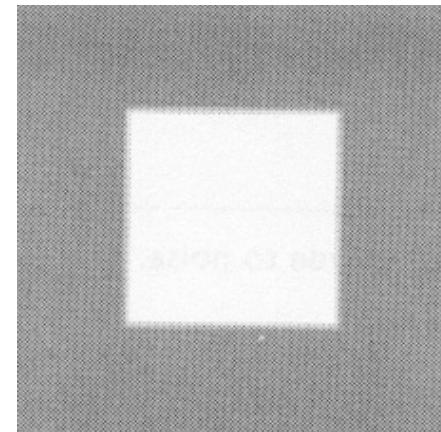
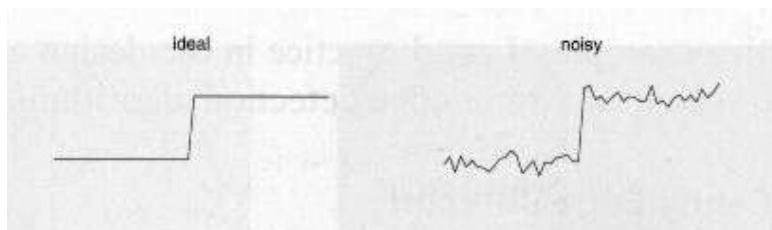
Descritores de Borda

- normal: vetor unitário na direção da mudança máxima de intensidade (gradiente).
- Direção: vetor unitário perpendicular à normal
- Posição: posição na imagem em que a borda está localizada.
- Magnitude: relacionado com o contraste local ao longo da normal



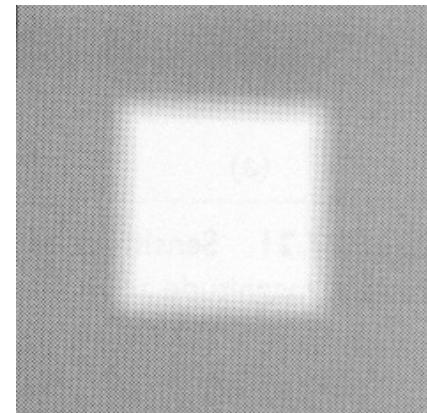
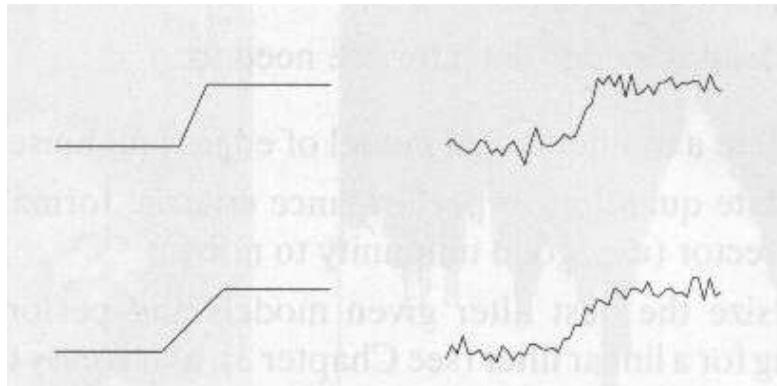
Borda Degrau

- A intensidade muda abruptamente de valor de um lado para o outro



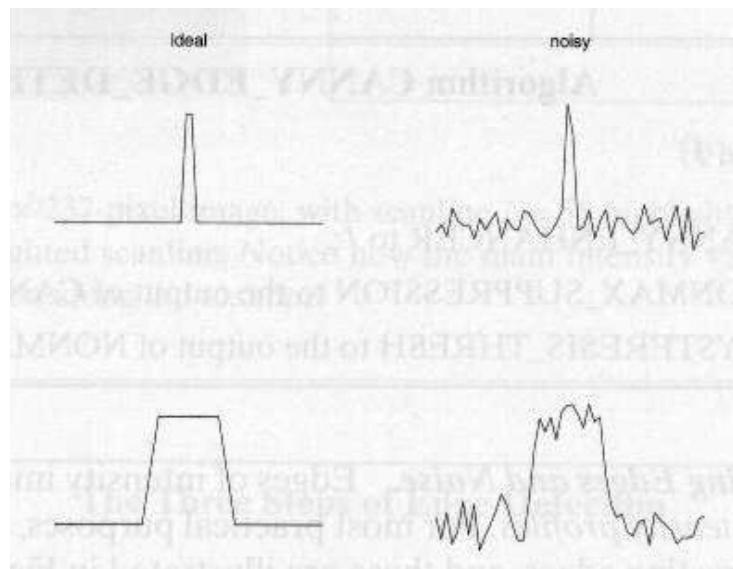
Borda em Rampa

- A intensidade muda gradativamente de valor de um lado para o outro



Ridge Edge

- Mudança abrupta de intensidade, voltando ao valor anterior em curta curta distância (geralmente causada por linhas)



Filtros detectores de bordas

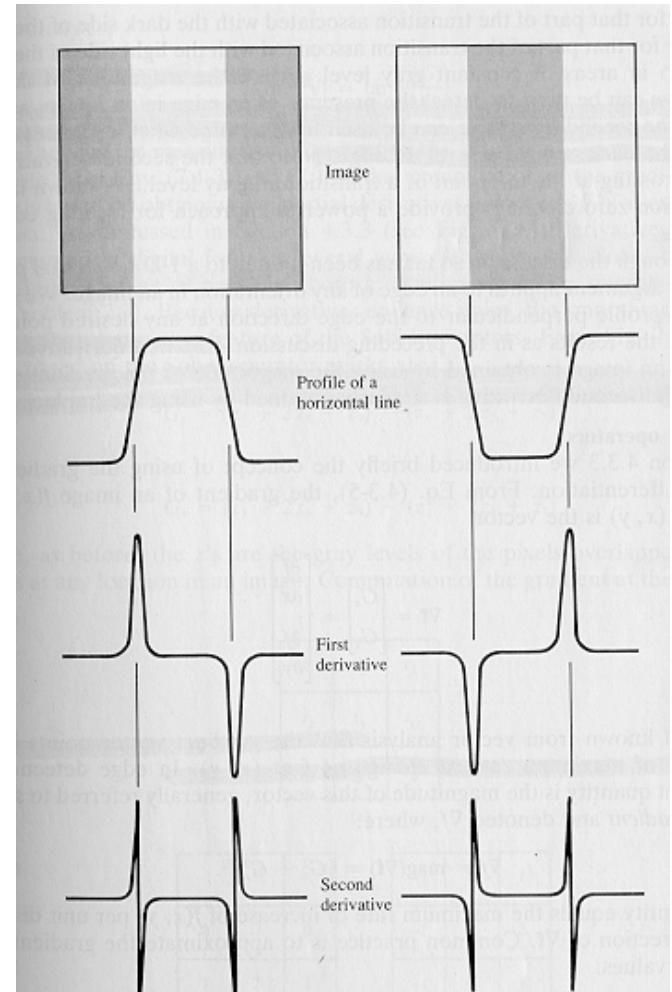
- Realçam detalhes, produzindo uma agudização (*sharpening*) da imagem, isto é, as transições entre regiões diferentes tornam-se mais nítidas. Estes filtros podem ser usados para realçar certas características presentes na imagem, tais como bordas, linhas curvas ou manchas, mas também enfatizam o ruído existente na imagem.

Considerações sobre bordas e derivadas

- Primeira ordem
 - Produzem bordas mais espessas
 - A magnitude da resposta é menor
- Segunda ordem
 - Realçam mais os detalhes (incluindo ruídos!)
 - Nas rampas ou degraus, apresenta sinais opostos (efeito borda dupla)
 - Sinal pode ser usado para determinar se uma borda é uma transição de claro p/ escuro (derivada negativa) ou escuro/claro (der. positiva)

Detecção de bordas por derivadas

- Pontos que se localizam nas bordas podem ser localizados:
 - (1) detectando-se a local maxima ou minima da primeira derivada.
 - (2) detectando-se o zero-crossings da segunda derivada.



Detecção de bordas usando 1^a derivada 1D

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} \approx f(x+1) - f(x) \quad (h=1) \quad \text{mask:} \quad [-1 \quad 1]$$

mask $M = [-1, 0, 1]$ (centrada em x)

Borda degrau para cima

S_1				12	12	12	12	12	24	24	24	24	24
S_1	\otimes	M		0	0	0	0	12	12	0	0	0	0

Borda degrau para baixo

S_2				24	24	24	24	24	12	12	12	12	12
S_2	\otimes	M		0	0	0	0	-12	-12	0	0	0	0

Borda rampa

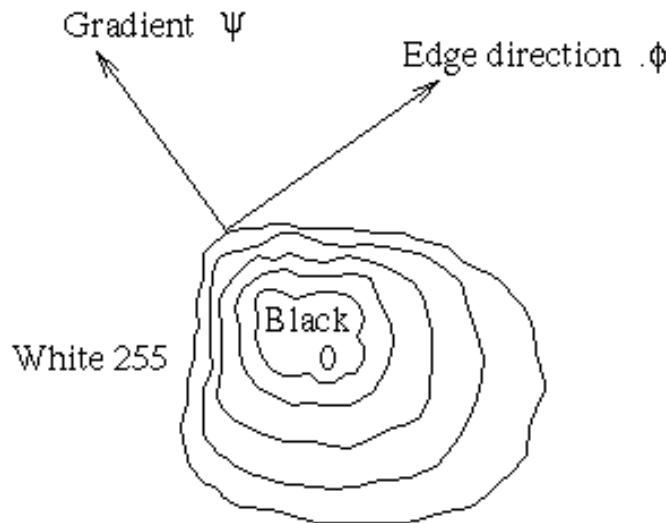
S_3				12	12	12	12	15	18	21	24	24	24
S_3	\otimes	M		0	0	0	3	6	6	6	3	0	0

Borda telhado

S_4				12	12	12	12	24	12	12	12	12	12
S_4	\otimes	M		0	0	0	12	0	-12	0	0	0	0

O Gradiente

- Em 2D as derivadas de primeira ordem são implementadas utilizando-se a magnitude do gradiente.
- A direção do gradiente é perpendicular a aresta e dá sempre a direção de crescimento da função: menor p/ maior intensidade !



$$|grad g(x, y)| = \sqrt{\left(\frac{\partial g}{\partial x}\right)^2 + \left(\frac{\partial g}{\partial y}\right)^2}$$

Detecção de bordas usando 1^a derivada 2D - Gradiente

- O Gradiente é definido como:

$$\frac{\partial f}{\partial x} = \lim_{h \rightarrow 0} \frac{f(x+h, y) - f(x, y)}{h} \quad \frac{\partial f}{\partial y} = \lim_{h \rightarrow 0} \frac{f(x, y+h) - f(x, y)}{h}$$

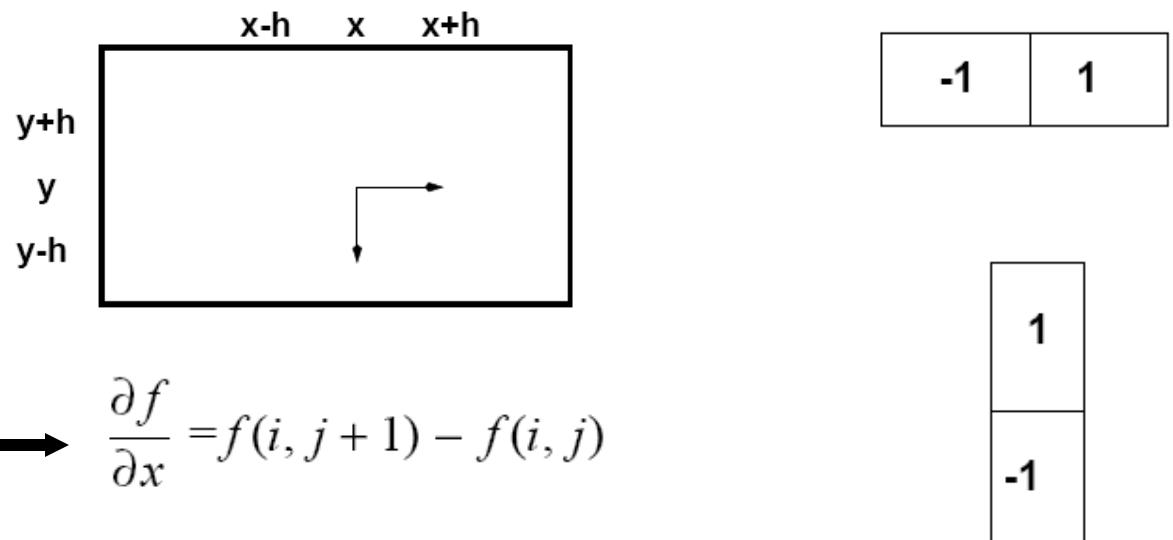
- Aproximação do gradiente por diferenças finitas:

$$\frac{\partial f}{\partial x} = \frac{f(x + h_x, y) - f(x, y)}{h_y} = f(x + 1, y) - f(x, y), \quad (h_x=1)$$

$$\frac{\partial f}{\partial y} = \frac{f(x, y + h_y) - f(x, y)}{h_y} = f(x, y + 1) - f(x, y), \quad (h_y=1)$$

Detecção de bordas usando 1^a derivada 2D - Gradiente

- Na notação de coordenada de pixel, j corresponde a direção x e i a direção negativa y.



$$f(x+1, y) - f(x, y) \rightarrow \frac{\partial f}{\partial x} = f(i, j+1) - f(i, j)$$

$$f(x, y+1) - f(x, y) \rightarrow \frac{\partial f}{\partial y} = f(i, j) - f(i+1, j)$$

Observações sobre o Gradiente

- Como os componentes do vetor gradiente são derivadas, estes são operadores lineares
- A magnitude, no entanto, não é linear (potência e raiz quadrada)
- As derivadas parciais são anisotrópicas, ou seja, não são invariantes a rotações. Mas a magnitude é invariante à rotação (isotrópica)

Operadores de primeira ordem

- Roberts
- Sobel
- Prewitt

Operador baseado na 1a Derivada - 2D

Roberts

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

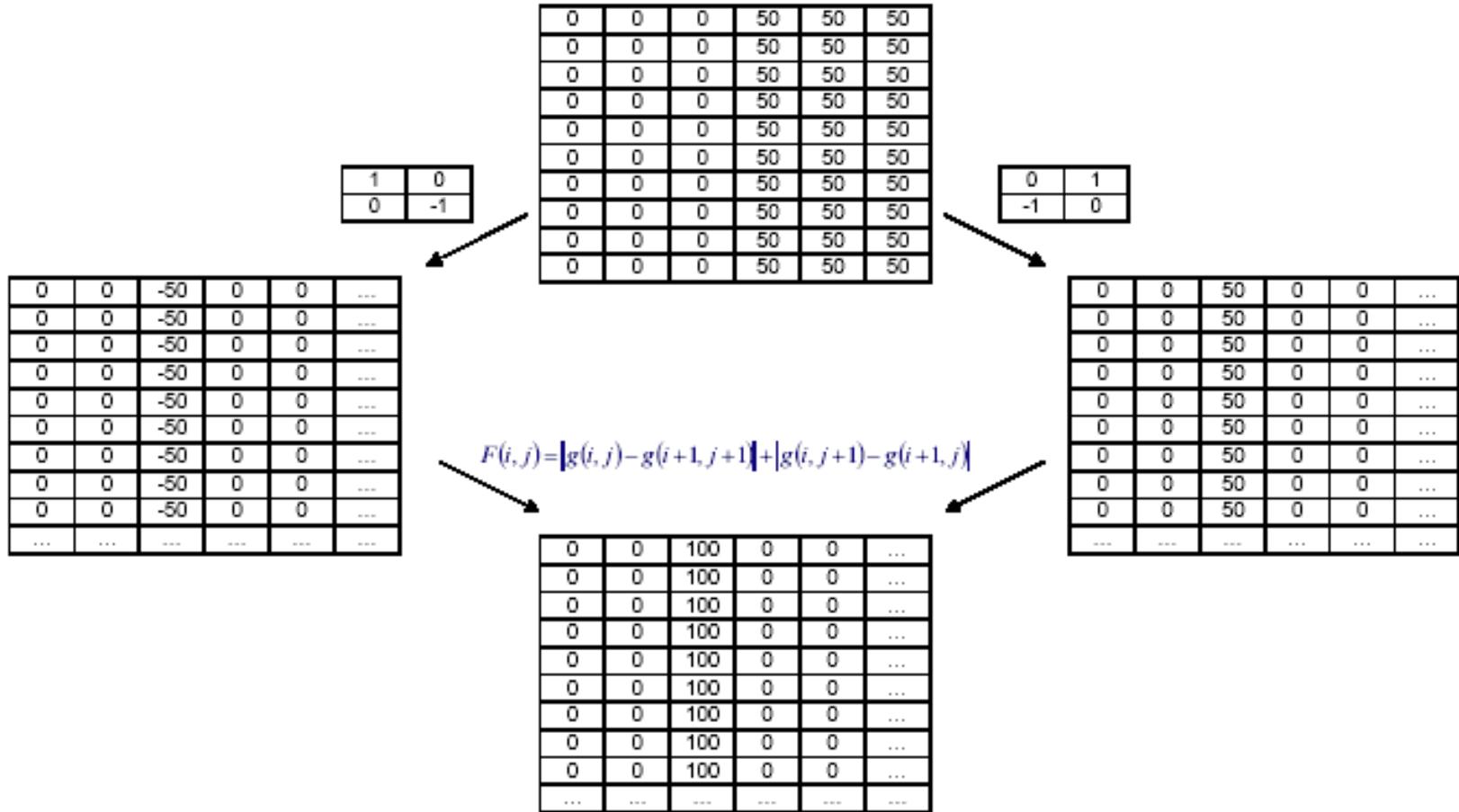
$$\frac{\partial f}{\partial y} = f(i + 1, j) - f(i, j + 1)$$

$$M_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad M_y = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

M_x e M_y são aproximações em $i + \frac{1}{2}$ e $j + \frac{1}{2}$

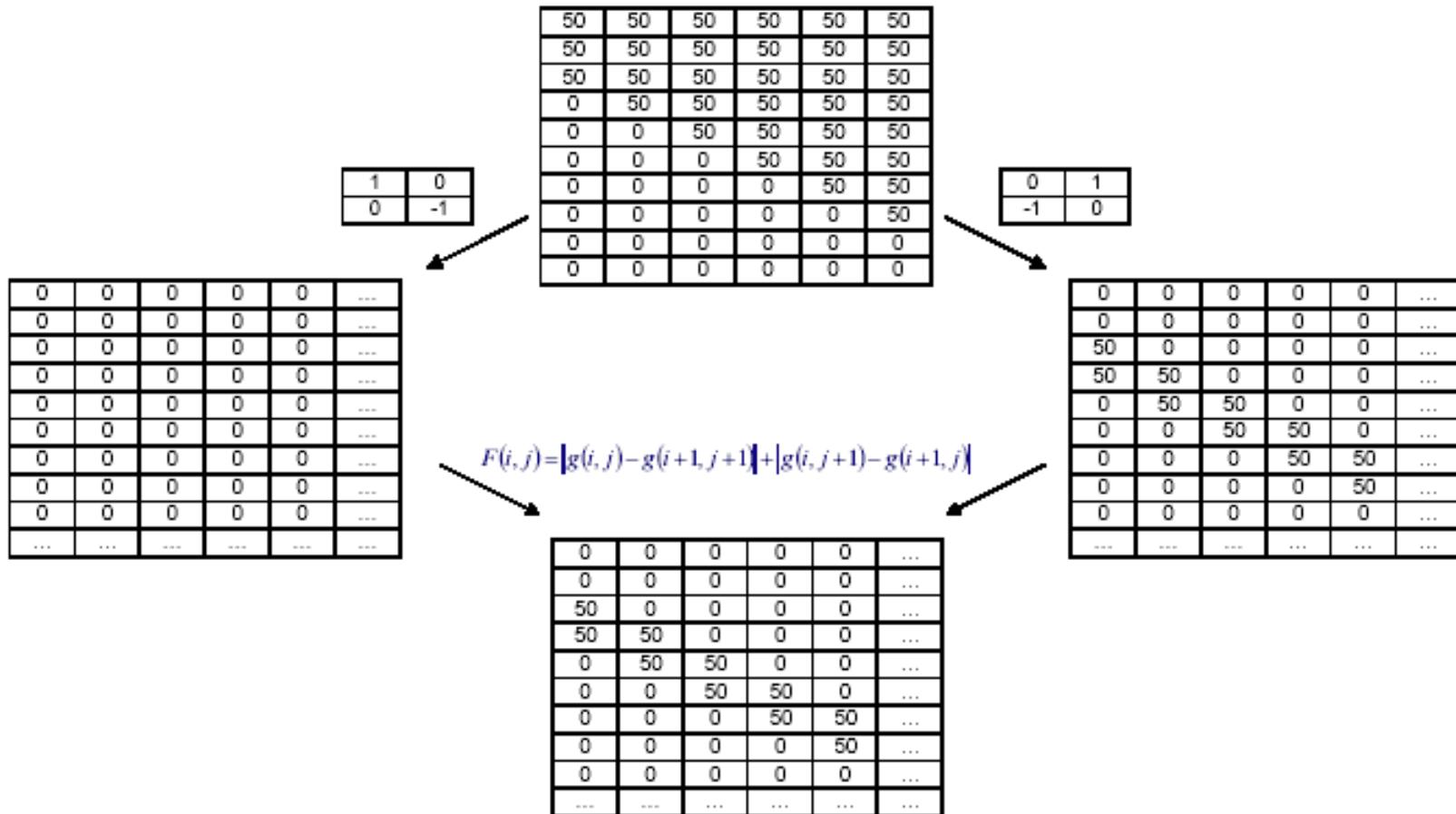
Operador baseado na 1a Derivada - 2D

Roberts



Operador baseado na 1a Derivada - 2D

Roberts



Operador baseado na 1a Derivada - 2D

Outra aproximação

- Considere o arranjo de pixels ao redor de (i, j) :

$$\begin{matrix} a_0 & a_1 & a_2 \\ a_7 & [i, j] & a_3 \\ a_6 & a_5 & a_4 \end{matrix}$$

- As derivadas parciais podem ser computadas como:

$$M_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6)$$
$$M_y = (a_6 + ca_5 + a_4) - (a_0 + ca_1 + a_2)$$

- A constante c implica que a ênfase foi dada ao pixel mais próximo ao centro da máscara.

Operador baseado na 1a Derivada - 2D

Prewitt

- Fazendo $c = 1$, temos o operador Prewitt

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad M_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

M_x e M_y são aproximações em (i, j)

Operador baseado na 1a Derivada - 2D

Sobel

- Fazendo $c = 2$, temos Sobel:

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad M_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

M_x e M_y são aproximações em (i, j)

Operador baseado na 1a Derivada - 2D

Sobel

a	b	c
d	e	f
g	h	i

$$S_x = (c + 2f + i) - (a + 2d + g)$$

$$S_y = (g + 2h + i) - (a + 2b + c)$$

$$S = \sqrt{S_x^2 + S_y^2}$$

Ou numa versão simplificada:

$$S = |S_x| + |S_y|$$

Operadores de Sobel



Bordas = thresholding

Operadores de Sobel

$$S_x = (c + 2f + i) - (a + 2d + g)$$

...
...	0	0	0	0	...
...	50	0	0	0	...
...	150	50	0	0	...
...	150	150	50	0	...
...	50	150	150	50	...
...	0	50	150	150	...
...	0	0	50	150	...
...	0	0	0	50	...
...

50	50	50	50	50	50
50	50	50	50	50	50
50	50	50	50	50	50
0	50	50	50	50	50
0	0	50	50	50	50
0	0	0	50	50	50
0	0	0	0	50	50
0	0	0	0	0	50
0	0	0	0	0	0

$$S_y = (g + 2h + i) - (a + 2b + c)$$

...
...	0	0	0	0	...
...	-50	0	0	0	...
...	-150	-50	0	0	...
...	-150	-150	-50	0	...
...	-50	-150	-150	-50	...
...	0	-50	-150	-150	...
...	0	0	-50	-150	...
...	0	0	0	-50	...
...

$$S = |S_x| + |S_y|$$

...
...	0	0	0	0	...
...	100	0	0	0	...
...	300	100	0	0	...
...	300	300	100	0	...
...	100	300	300	100	...
...	0	100	300	300	...
...	0	0	100	300	...
...	0	0	0	100	...
...

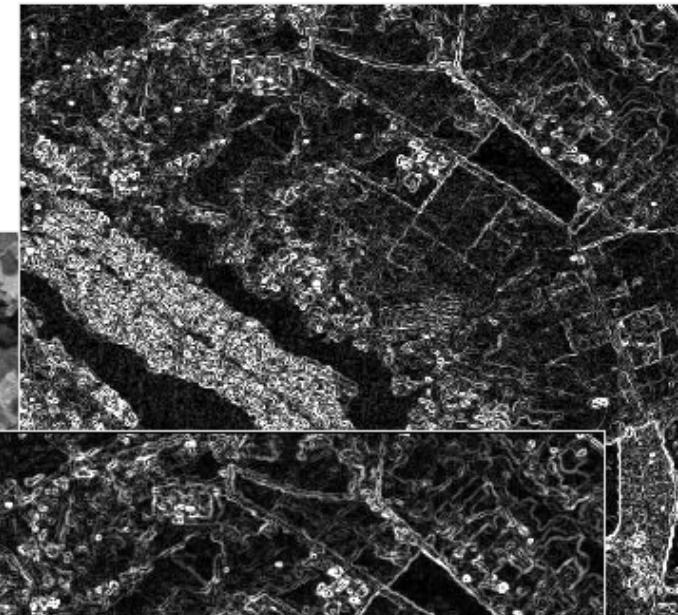
a	b	c
d	e	f
g	h	i

Comparação

Roberts



Original



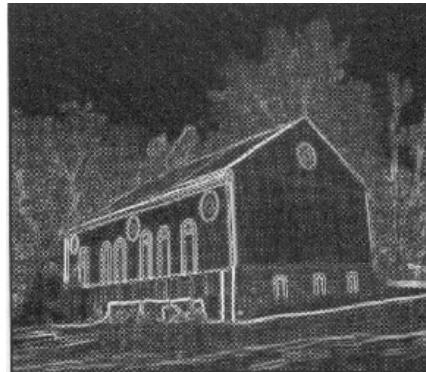
Sobel

Questões práticas

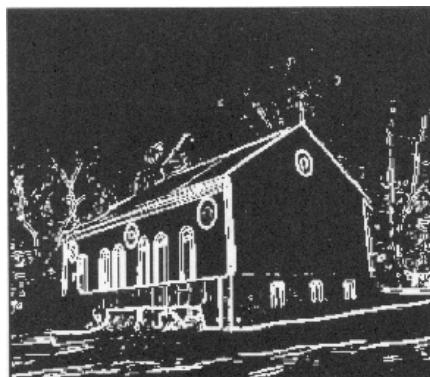
Como escolher o limiar ?



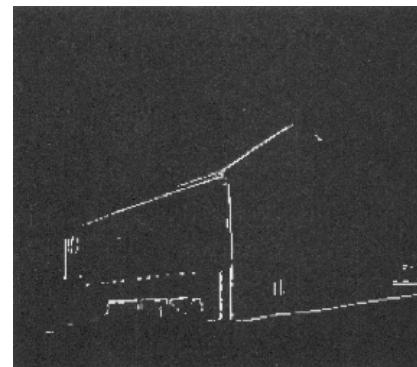
Magnitude do gradient e



Baixo threshold

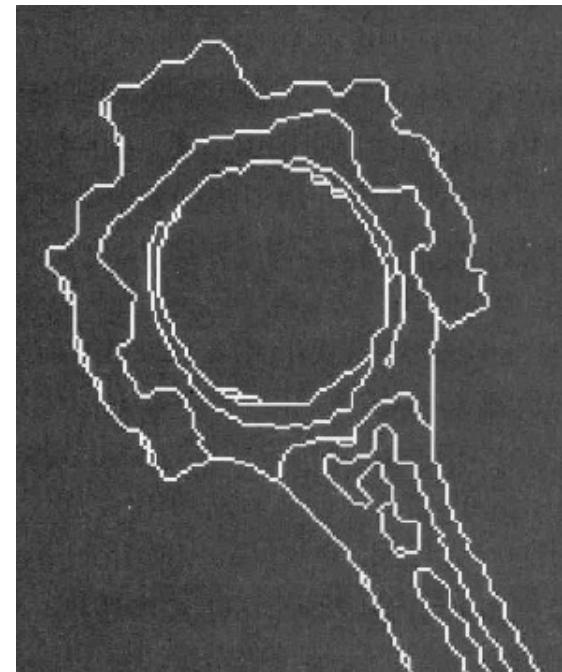
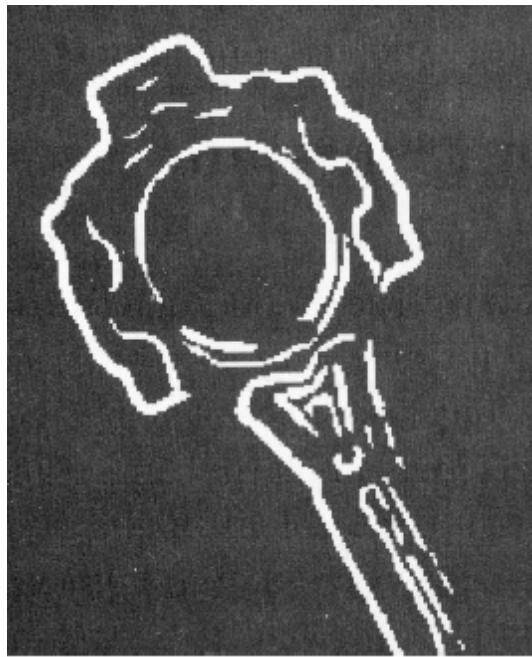


Alto threshold



Questões práticas

- Afinamento e fechamento normalmente são operações necessárias !



Detecção de bordas usando 2^a derivada 1D

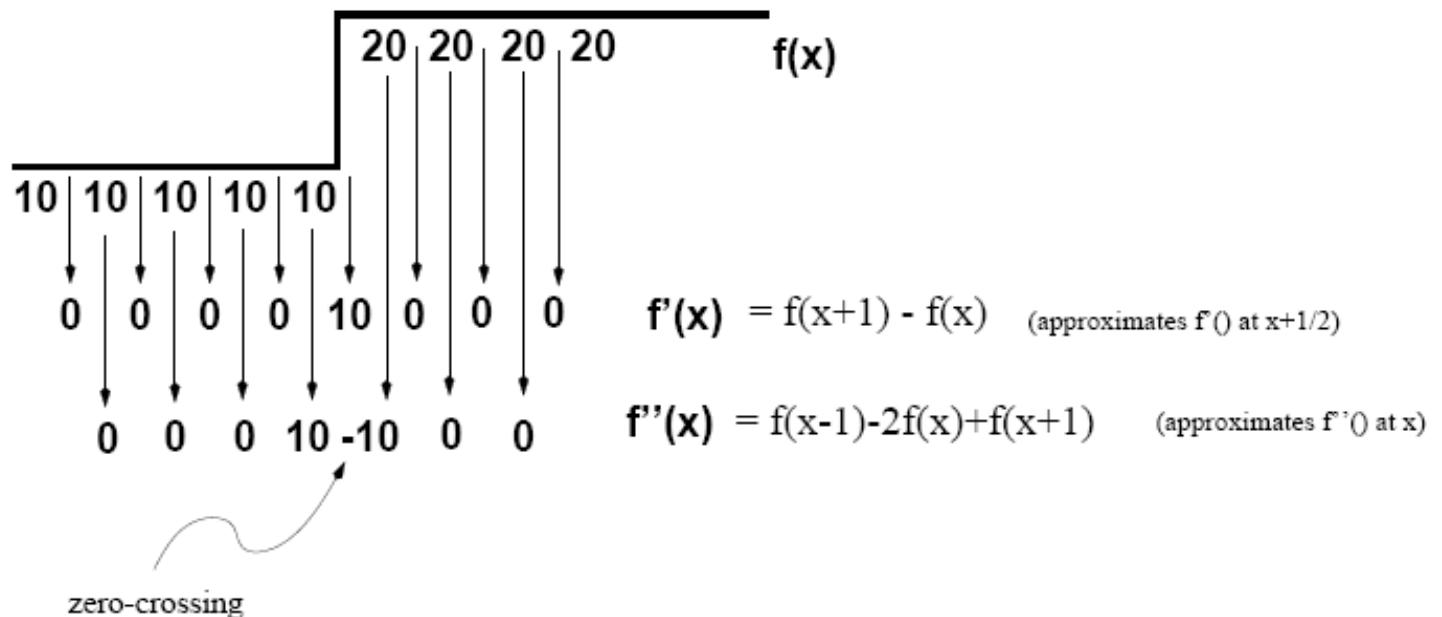
$$f''(x) = \lim_{h \rightarrow 0} \frac{f'(x+h) - f'(x)}{h} \approx f'(x+1) - f'(x) = \\ f(x+2) - 2f(x+1) + f(x) \quad (h=1)$$

Substituindo $x+1$ por x (ou seja, centrando em x):

$$f''(x) \approx f(x+1) - 2f(x) + f(x-1)$$

mask: [1 -2 1]

Detecção de bordas usando 2^a derivada 1D



Detecção de bordas usando 2^a derivada 1D

Borda degrau para cima

S_1			12	12	12	12	12	24	24	24	24	24
S_1	\otimes	M	0	0	0	0	-12	12	0	0	0	0

Borda degrau para baixo

S_2			24	24	24	24	24	12	12	12	12	12
S_2	\otimes	M	0	0	0	0	12	-12	0	0	0	0

Borda em rampa

S_3			12	12	12	12	15	18	21	24	24	24
S_3	\otimes	M	0	0	0	-3	0	0	0	3	0	0

Borda telhado

S_4			12	12	12	12	24	12	12	12	12	12
S_4	\otimes	M	0	0	0	-12	24	-12	0	0	0	0

Detecção de bordas usando 2^a derivada 2D - Laplaciano

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

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

$$\frac{\partial^2 f}{\partial y^2} = f(i+1, j) - 2f(i, j) + f(i-1, j)$$

$$\nabla^2 f = -4f(i, j) + f(i, j+1) + f(i, j-1) + f(i+1, j) + f(i-1, j)$$

máscara:

0	1	0
1	-4	1
0	1	0

Detecção de bordas usando 2^a derivada 2D - Laplaciano

detect zero-crossings

5	5	5	5	5	5
5	5	5	5	5	5
5	5	10	10	10	10
5	5	10	10	10	10
5	5	5	10	10	10
5	5	5	5	10	10

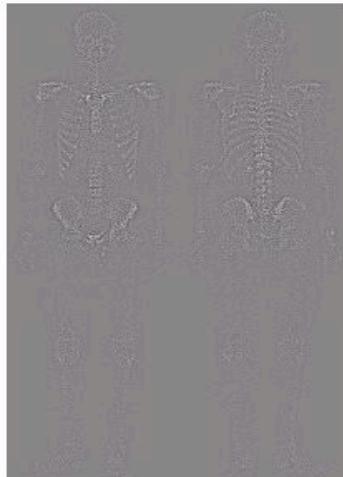
-	-	-	-	-	-
-	0	-5	-5	-5	-
-	-5	10	5	5	-
-	-5	10	0	0	-
-	0	-10	10	0	-
-	-	-	-	-	-

Uso combinado de filtros

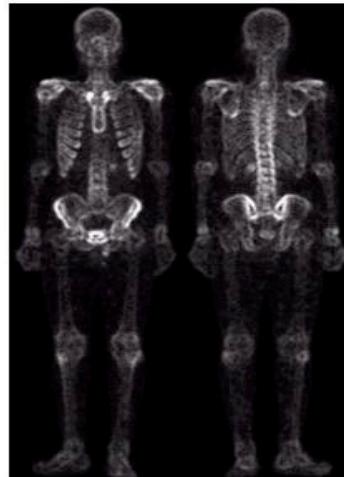
(a)



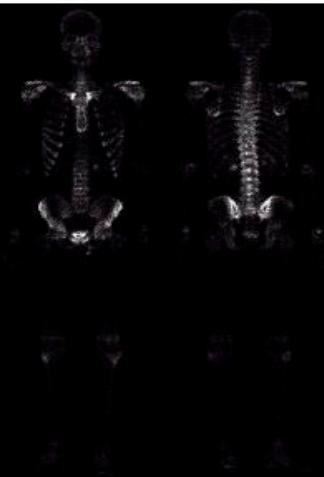
(b)



(e)

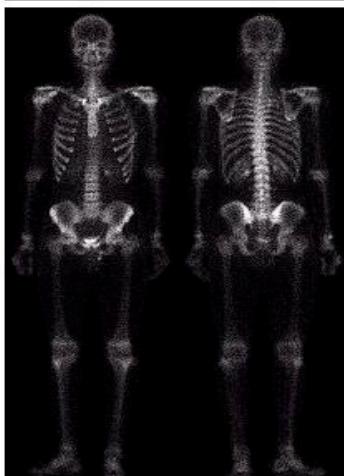


(f)



- Imagen original
b) Laplaciano em a
c) a + b
d) Sobel de a
e) Filtro média em d
f) c * e
g) a + f
h) g * g

(c)



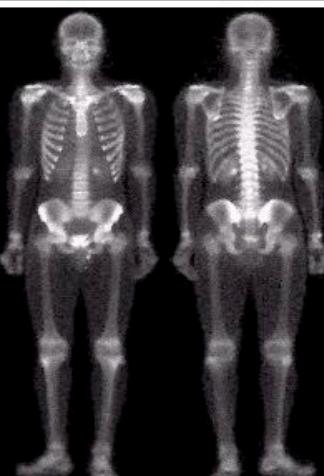
(d)



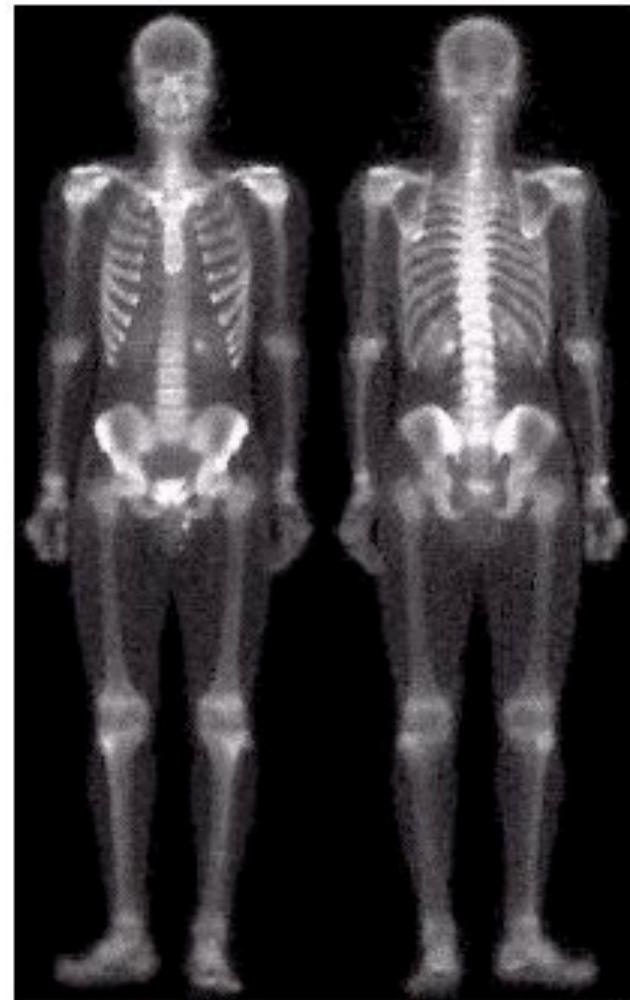
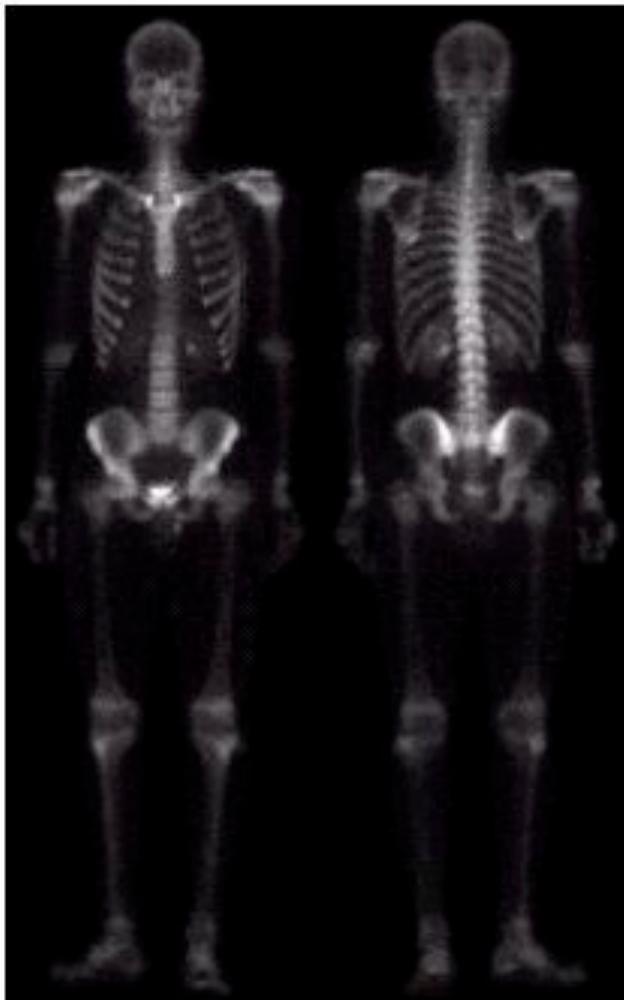
(g)



(h)



Uso combinado de filtros



Detector de bordas de Canny

- Taxonomicamente este operador deveria estar junto aos operadores de primeira derivada, mas devido a sua eficácia foi deixado para o final
 - Marr Hildreth (1980)
 - Canny (1986)
 1. Baixa taxa de erro : encontrar todas as bordas
 2. Boa localização: bordas detectadas o mais próximo das bordas reais
 3. Resposta única: um único ponto de borda retornado

Detector de bordas de Canny

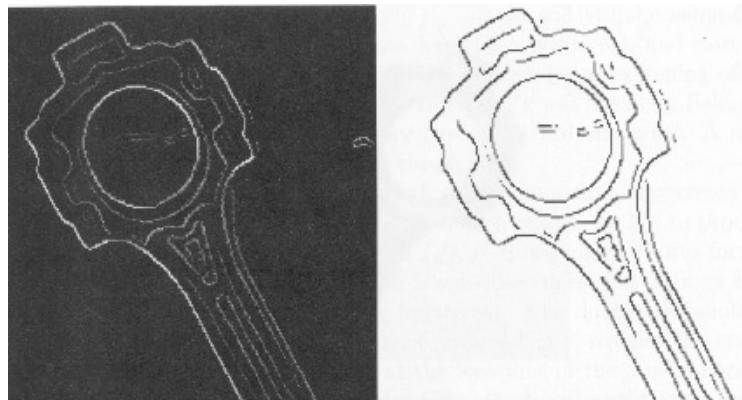
- Atacando os critérios taxa de erro e localização
 - Canny mostrou que uma boa aproximação para um detector ótimo de bordas de degrau é a primeira derivada de uma Gaussiana.
 - A análise é baseada em bordas de degrau (step-edges) corrompidas por ruído branco Gaussiano
 - (ruído branco Gaussiano: ruído branco em que a distribuição dos valores de amplitude é Gaussiana)
- Critério única resposta
 - Aproximação numérica

Canny - Etapas

1. Suavizar uma imagem f por uma Gaussiana
 - $F_s(x,y) = f(x,y) * G(x,y)$
2. Para cada pixel, computar a magnitude $M(x,y)$ e a direção do gradiente $\alpha(x,y)$
 - Usar qualquer filtro de primeira derivada
 - $M(x,y)$ fatalmente conterá cristas largas em torno dos máximos locais. Para afinar estas cristas utilize algoritmo de supressão dos não máximos, afinando $M(x,y)$
3. Limiarizar $M(x,y)$ e analisar conectividade.

Canny – Efeito de σ

(Canny - 7x7 Gaussian, more details)



(Canny - 31x31 Gaussian, less details)

