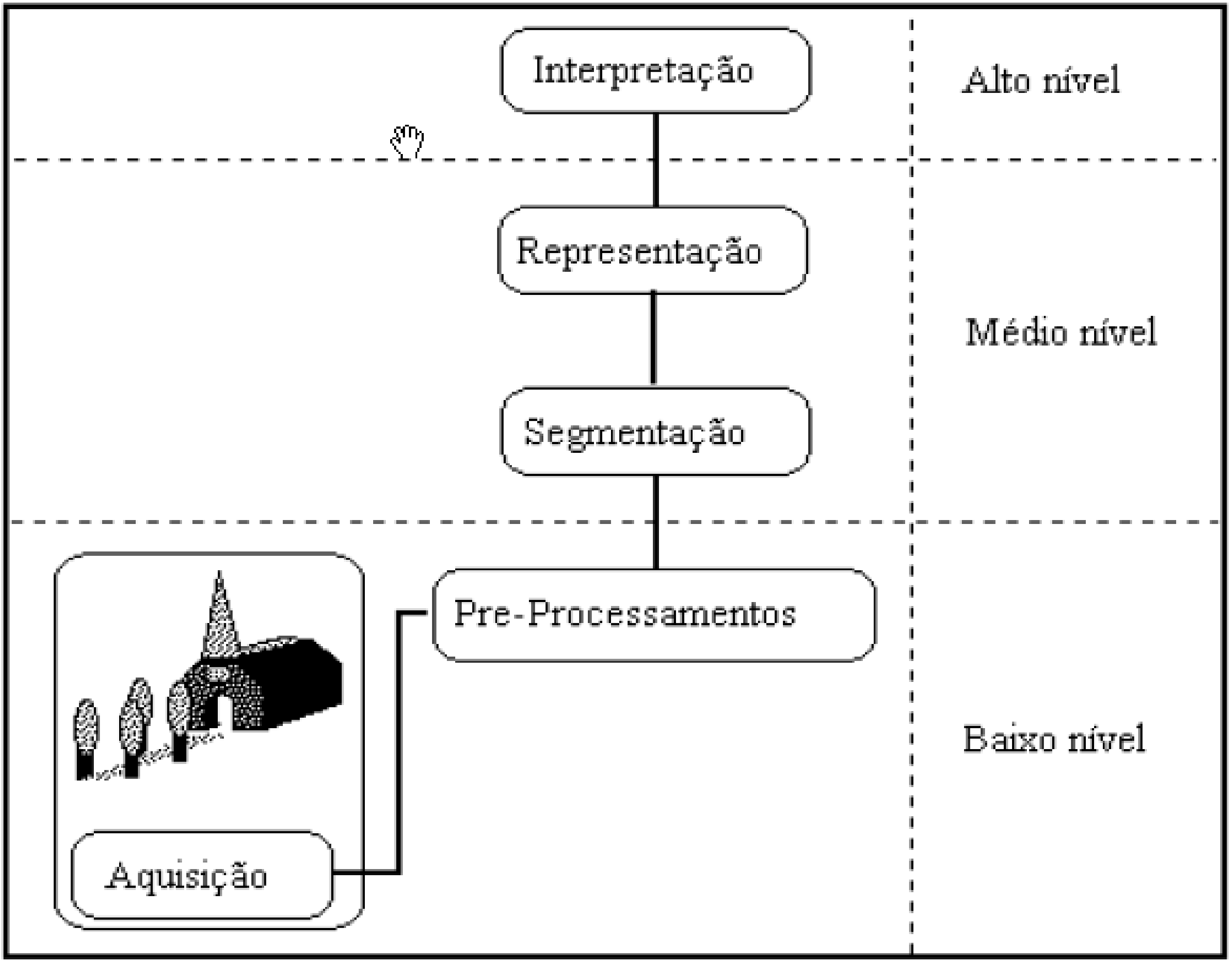
Processamento de Imagens

SCC0251

Pré-processamento

Material baseado nos livros do Sonka e do Gonzalez

http://www.icaen.uiowa.edu/~dip/LECTURE/lecture.html


Interpretação
Alto nível
Representação
Médio nível
Segmentação
Pre-Processamentos
Aquisição
Baixo nível

Propriedades de uma Imagem digital

- Vizinhança
- Conectividade
- Medição de Distância
- Operações Lógicas e Aritméticas

Vizinhança

- $N_4(p)$: pixel p (x,y) com 4 vizinhos
  - (x+1,y), (x-1,y), (x,y+1) e (x,y-1)
  - 4-vizinhança
- $N_8(p)$: pixel p (x,y) com 8 vizinhos
  - Os acima +
  - (x+1,y+1), (x-1,y-1), (x-1,y+1) e (x+1,y-1)
  - 8-vizinhança
- Vizinhança diagonal

Conectividade

- Importante para estabelecer limites de regiões e objetos em uma imagem
  - 4-conectividade: pixels p e q são 4-conectados se $q \in N_4(p)$
  - 8-conectividade: pixels p e q são 8-conectados se $q \in N_8(p)$

Distâncias

- Dados pixels p (x,y); q (s,t) e z (u,v)
- Distância D
  - $D(p,q) \geq 0$ (positividade) $D(p,q) = 0$, se somente se $p=q$)
  - $D(p,q) = D(q,p)$ (simetria)
  - $D(p,z) \leq D(p,q) + D(q,z)$ (desigualdade triangular)

Distâncias

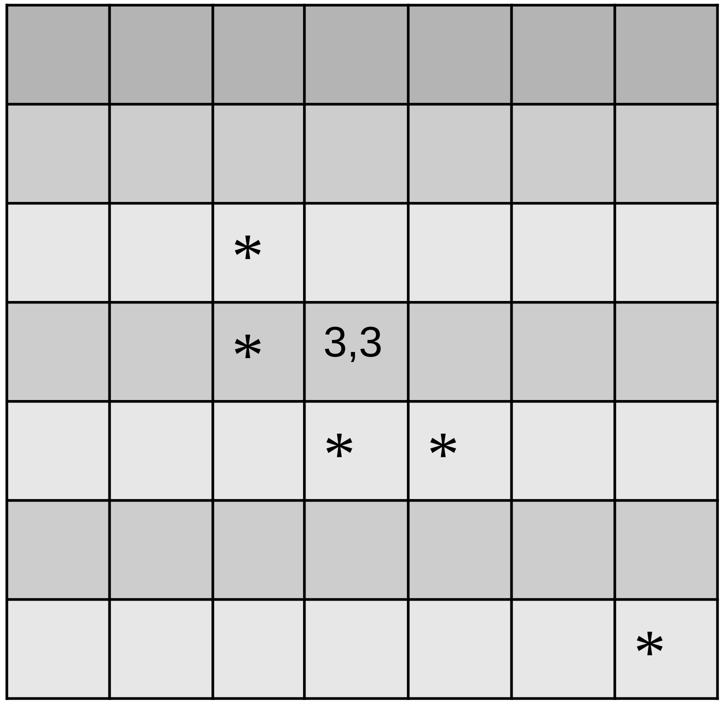
- Euclidiana: $D_e(p,q) = \sqrt{(x-s)^2 + (y-t)^2}$

- Distância D_4 (city-block/Manhattan):

$$|x-s| + |y-t|$$

- Distância D_8 (tab. xadrez):

$$\max(|x-s|, |y-t|)$$



Para o pixel de coordenadas (x,y) = (3,3), compute as 3 distâncias para os pixels vizinhos marcados

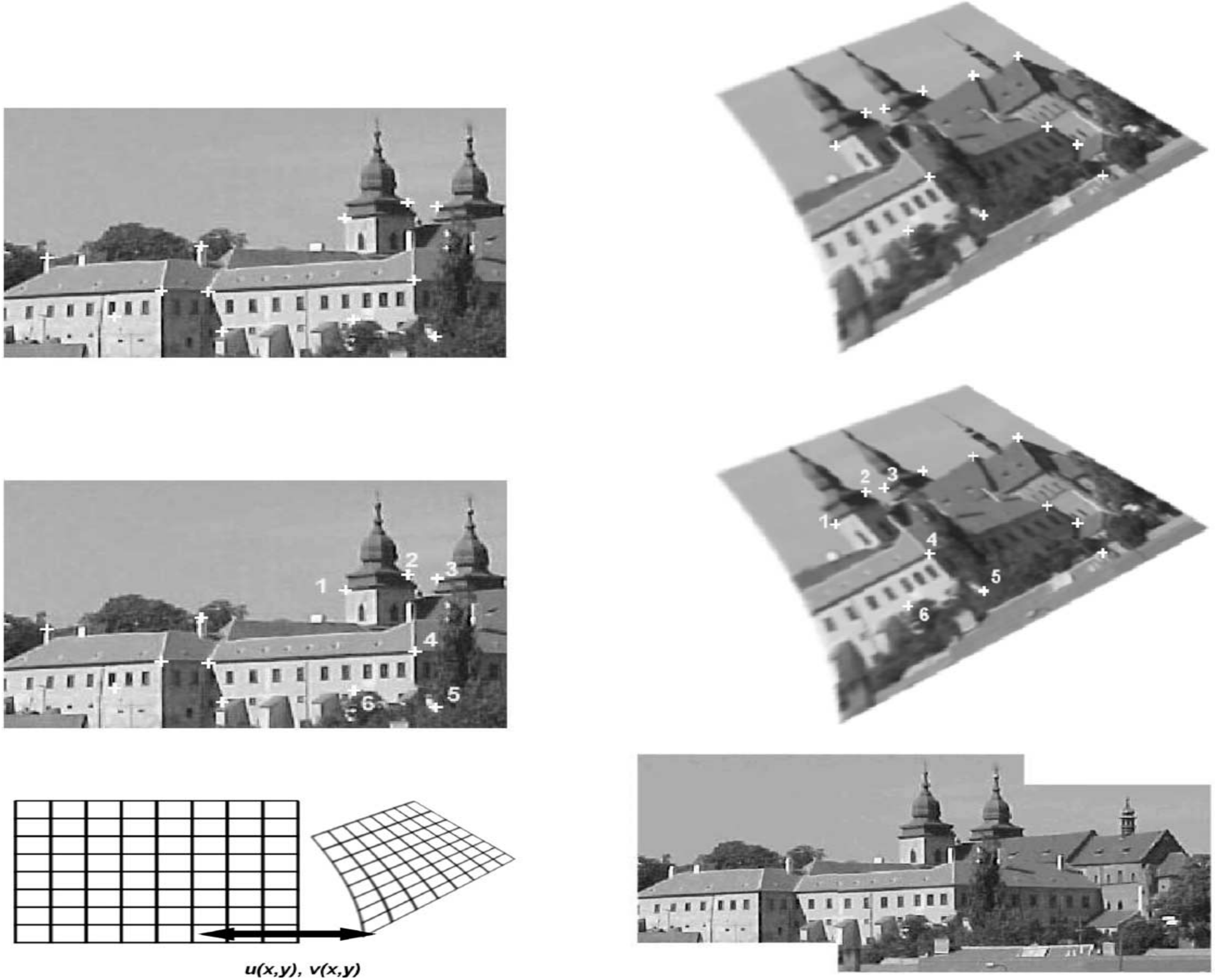
Operações Aritméticas e Lógicas

- Soma, subtração, multiplicação, divisão
  - Operações úteis, por exemplo, para filtrar ruído!
- AND, OR, XOR
  - Identificação de máscaras
- Transformações geométricas
  - Registro de imagens!

Registro de Imagens

- Alinhar duas ou mais imagens de uma cena
- Dadas duas imagens, qual a transformação geométrica que leva a imagem de entrada na imagem de saída?
- Problema: estimar a função de transformação e utilizá-la para registrar as 2 imagens
    - Imagem de entrada: a que queremos transformar
    - Imagem de referência: em relação à qual queremos registrar a entrada

Registro de imagens



Pré-processamento

- Não aumenta o conteúdo de informação em uma imagem
- Às vezes, diminui o conteúdo
- O melhor pré-processamento é o não pré-processamento
  - concentrar-se no processo de aquisição de imagens
- É, mesmo assim, útil por eliminar informações redundantes

Pré-processamento

- Melhorar a qualidade da imagem (dados) suprimindo distorções indesejadas e ou realçando certas características importantes para o processamento posterior

  {note que transformações geométricas (rotação, escala, translação, etc.) são classificadas como pré-processamento}

- 4 categorias → tamanho da vizinhança considerada no cálculo do novo nível cinza

4 Categorias

- Transformações de brilho (histograma, etc.)
- Transformações geométricas (distorções)
- Métodos de filtragem espacial que consideram uma vizinhança local a cada pixel (suavização, filtros)
- Restauração (requer conhecimento sobre a imagem investigada)

Transformadas de brilho do pixel

- Dependem exclusivamente da propriedade do próprio pixel
- Há duas classes de transformadas
  - Correção de brilho (*brightness correction*)
  - transformada de nível de cinza (*grey-scale transformation*)

Transformadas

- Correção de brilho (*position-dependent*)
  - modifica o brilho do pixel levando em conta o seu brilho original e sua posição na imagem

- Transformada de nível de cinza
  - modifica o brilho do pixel sem considerar sua posição na imagem (equalização do histograma, por exemplo)

Correção de Brilho

- Foto-sensitividade do sensor não é uniforme (câmeras CCD, por exemplo)
  - Na prática, dispositivos de aquisição e digitalização atenuam mais a luz quanto mais longe esta está do eixo óptico
- Iluminação desigual dos objetos também é fonte de degradação

Correção da degradação

(quando sistemática e estável)

$$g(x,y) = f(x,y) \bullet h(x,y)$$

g(x,y): imagem obtida, com degradação

h(x,y): coeficiente de erro

f(x,y): imagem original (ideal) não degradada (desejada)

$$f(x,y) = g(x,y) \bullet h^{-1}(x,y)$$

h(x,y) estimado ao capturar uma imagem g(x,y) de brilho conhecido (imagem com brilho constante c)

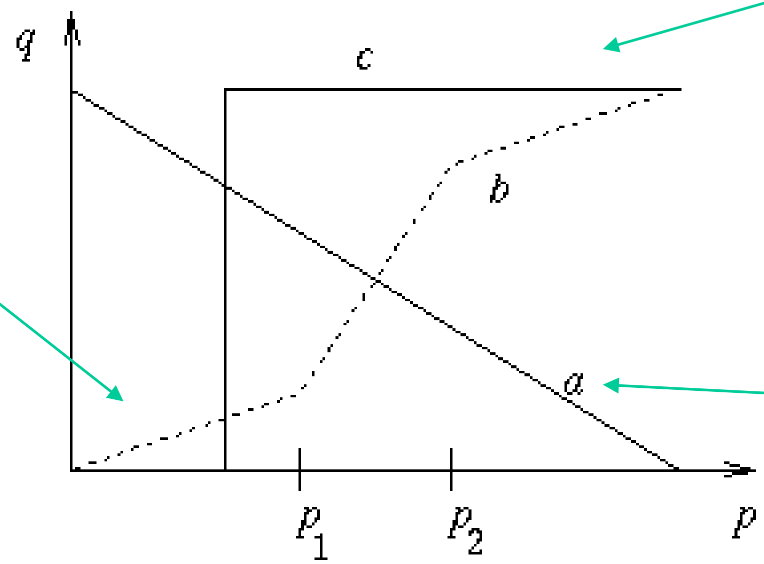
Transformadas

- Correção de brilho (*position-dependent*)
  - modifica o brilho do pixel levando em conta o seu brilho original e sua posição na imagem

- Transformada de nível de cinza
  - modifica o brilho do pixel sem considerar sua posição na imagem (equalização do histograma, por exemplo)

Transformadas do nível de cinza

$$q = T(p) \qquad (4.3)$$



Thresholding

(**limiarização**)

Piecewise Linear Function:

Realce de contraste na faixa (p1,p2)

Transformada negativa

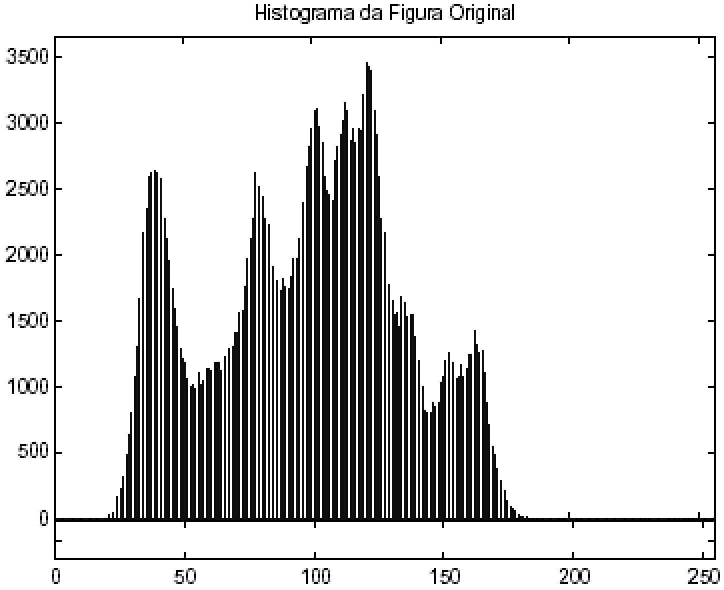
T: Transformação
p: pixel original (brilho original)
q: novo valor

Histogramas em Imagens Digitais

$p_r(r_k) = n_k / n$ onde

- $0 \leq r_k \leq 1$
- k = 0,1,..., L-1, onde L é o nro de níveis de cinza da imagem
- n = nro total de pixels na imagem
- $p_r(r_k)$ = probabilidade do k-ésimo nível de cinza
- $n_{k\,=}$ nro de pixels cujo nível de cinza corresponde a k

Histograma em imagens digitais

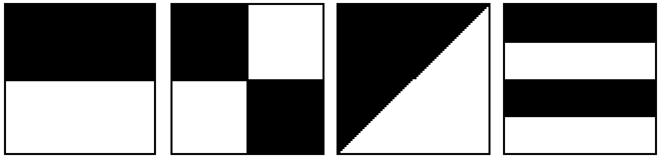


Propriedades de Histogramas

- O histograma não preserva a informação espacial da distribuição dos pixels, pois registra apenas a quantidade de pixels com um determinado nível de cinza, mas não a sua posição na imagem

- Área do histograma = área da imagem $\int_0^{\infty} H(p)\,dp = \acute{A}rea\ da\ Imagem$

- Imagens distintas podem apresentar histogramas idênticos

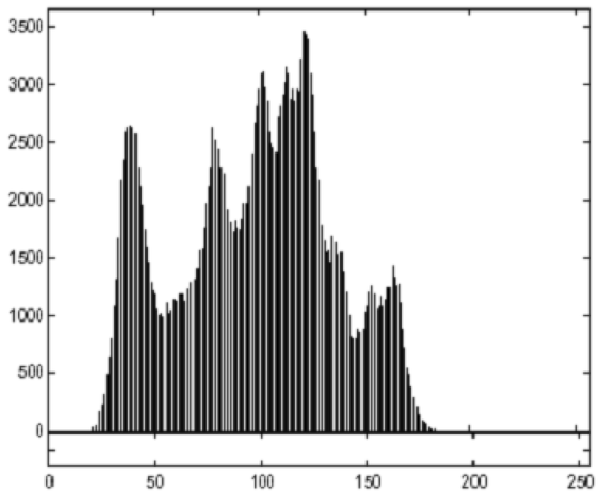


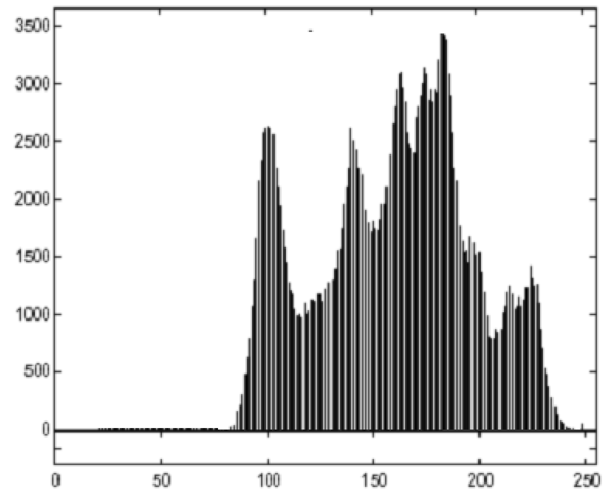
- Os histogramas podem ser utilizados para fins de reconhecimento de padrões

Técnicas de processamento com Histogramas

- Ajuste de brilho e contraste
- Equalização / Refinamento Imagem
- Segmentação
- Mapas de pseudocores

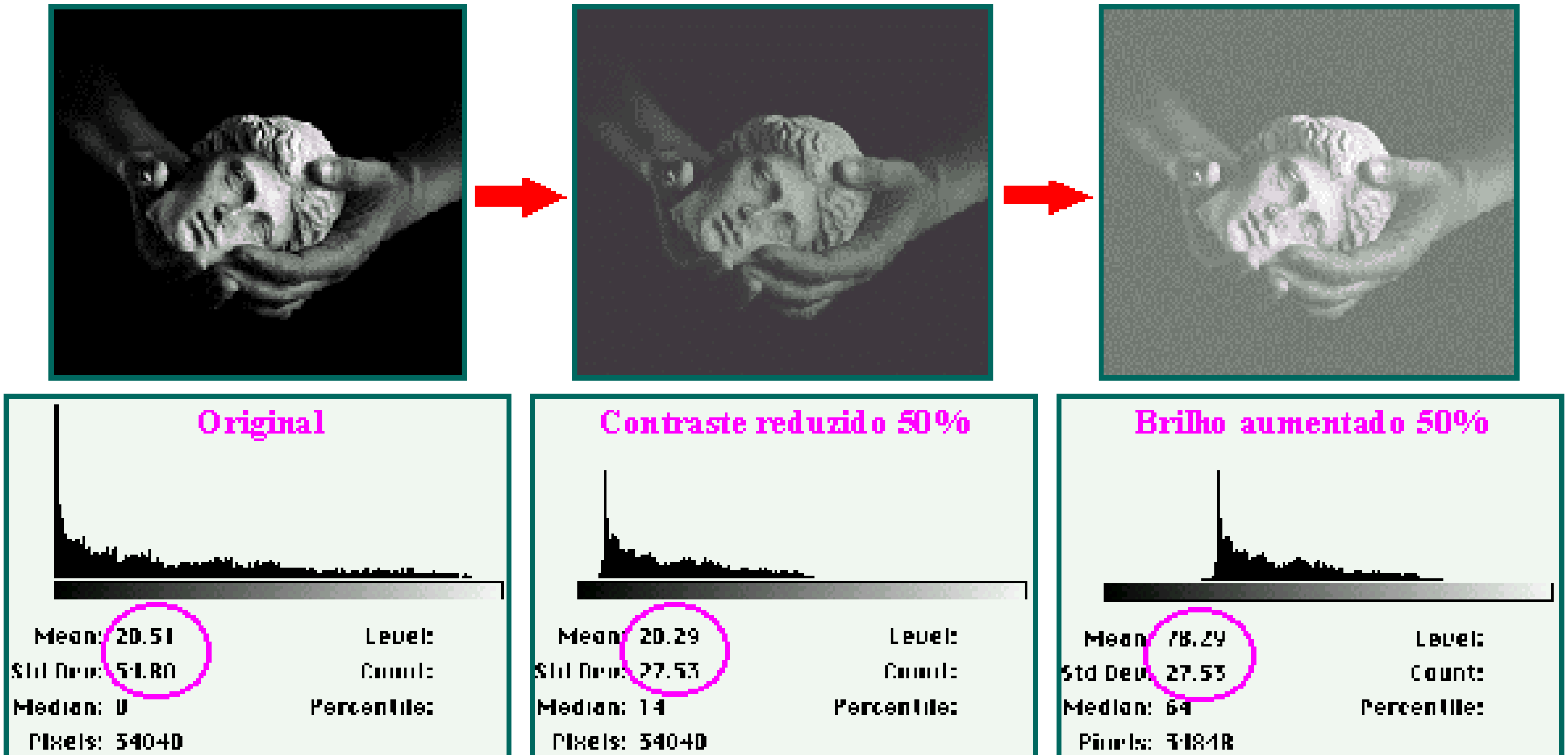
Ajustes de brilho





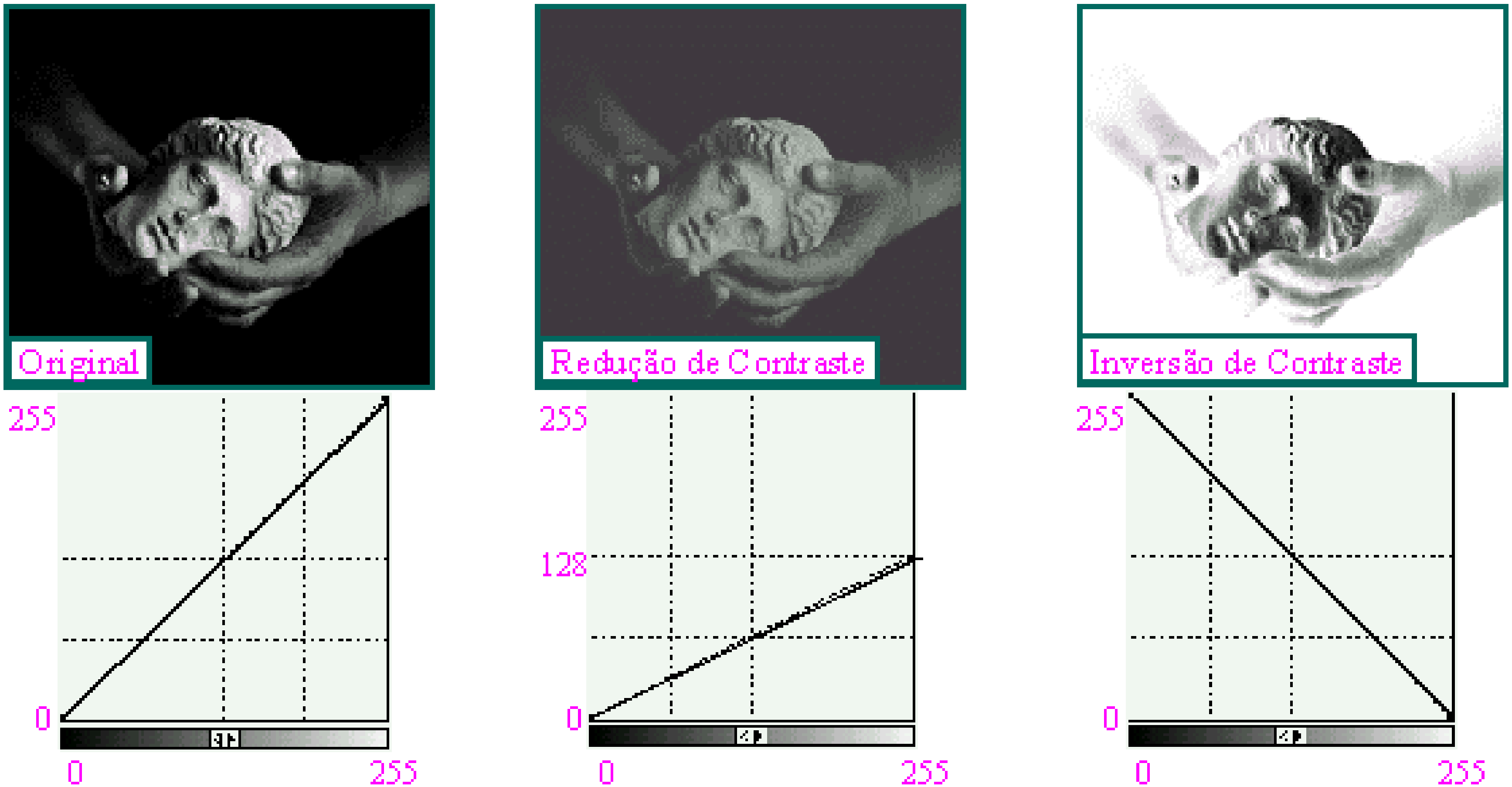
- Brilho
  - Pode ser realizado através da soma/subtração pontual na intensidade dos pixels

Histograma

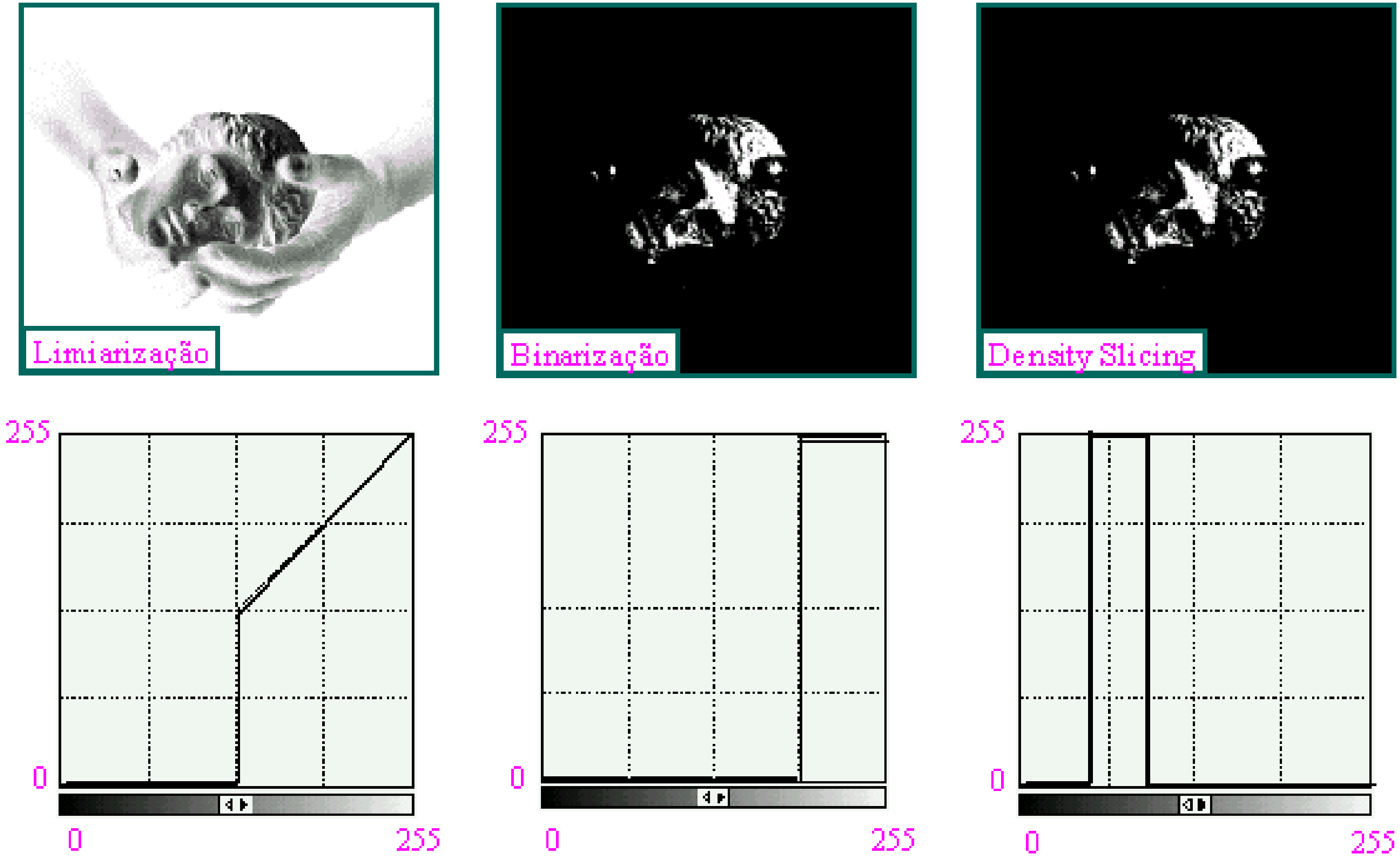


Original
Contraste reduzido 50%
Brilho aumentado 50%
Mean: 20.51
Median: 0
Pixels: 34040
Mean: 20.29
27.53
Median: 14
Pixels: 34040
Mean: 78.29
Std Dev: 27.53
Median: 64
Level:
Count:
Percentile:

Algumas transformações contraste lineares



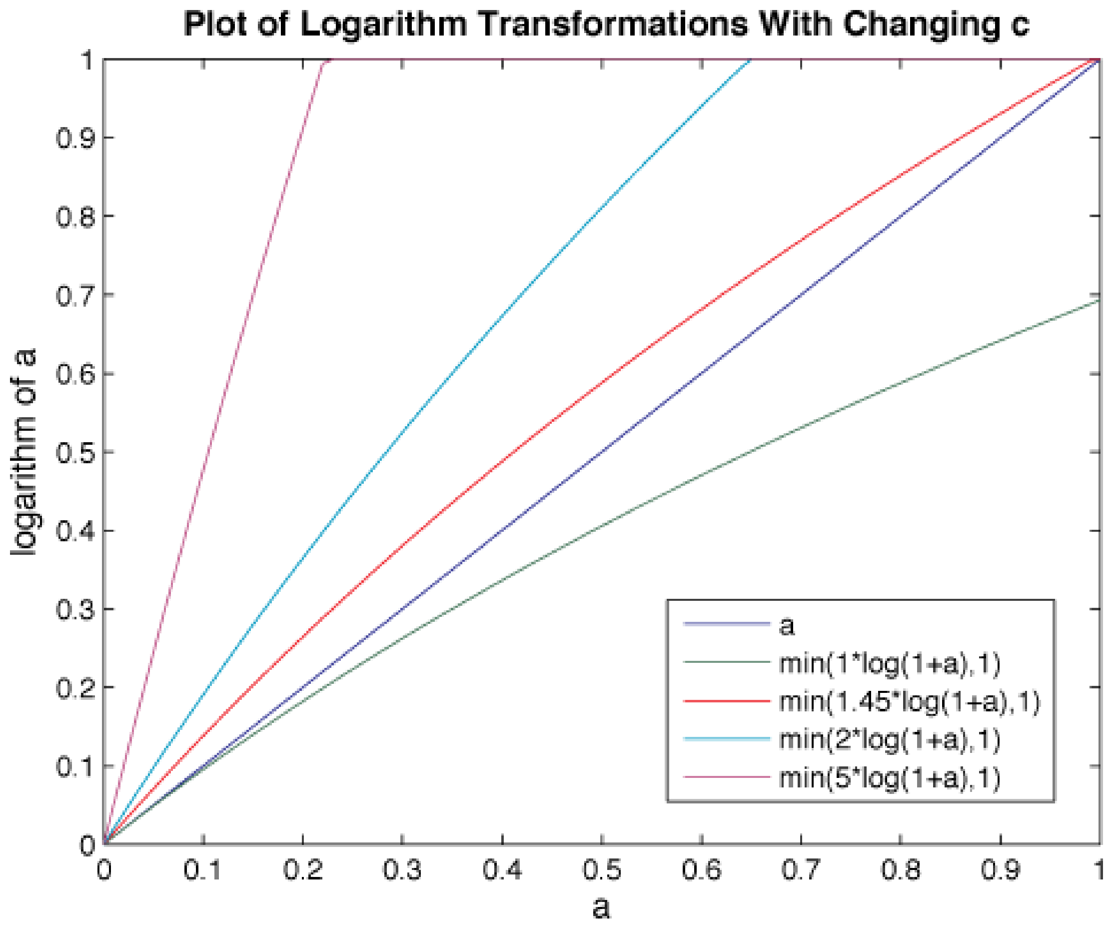
Outras transformações lineares



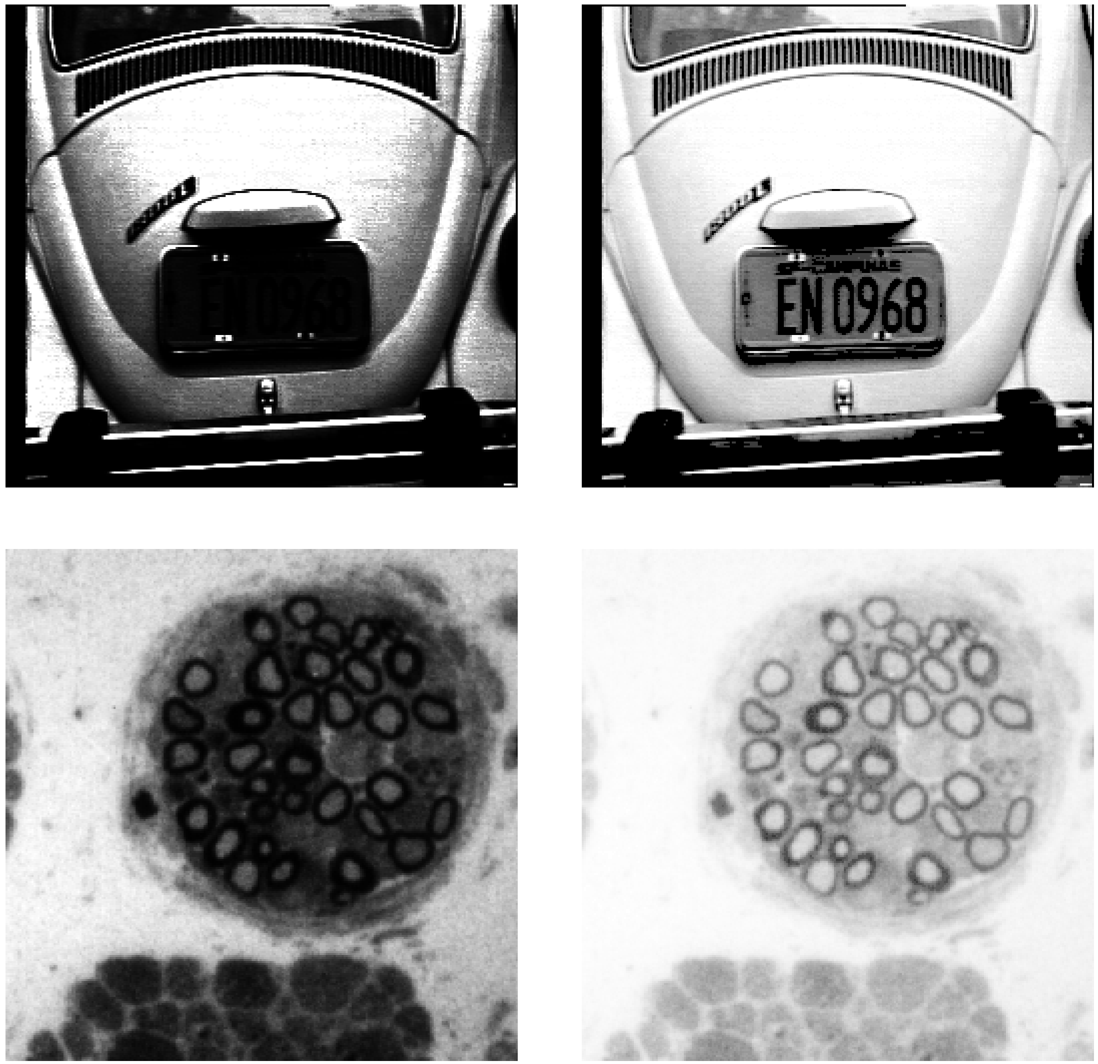
Ajustes de contraste não lineares

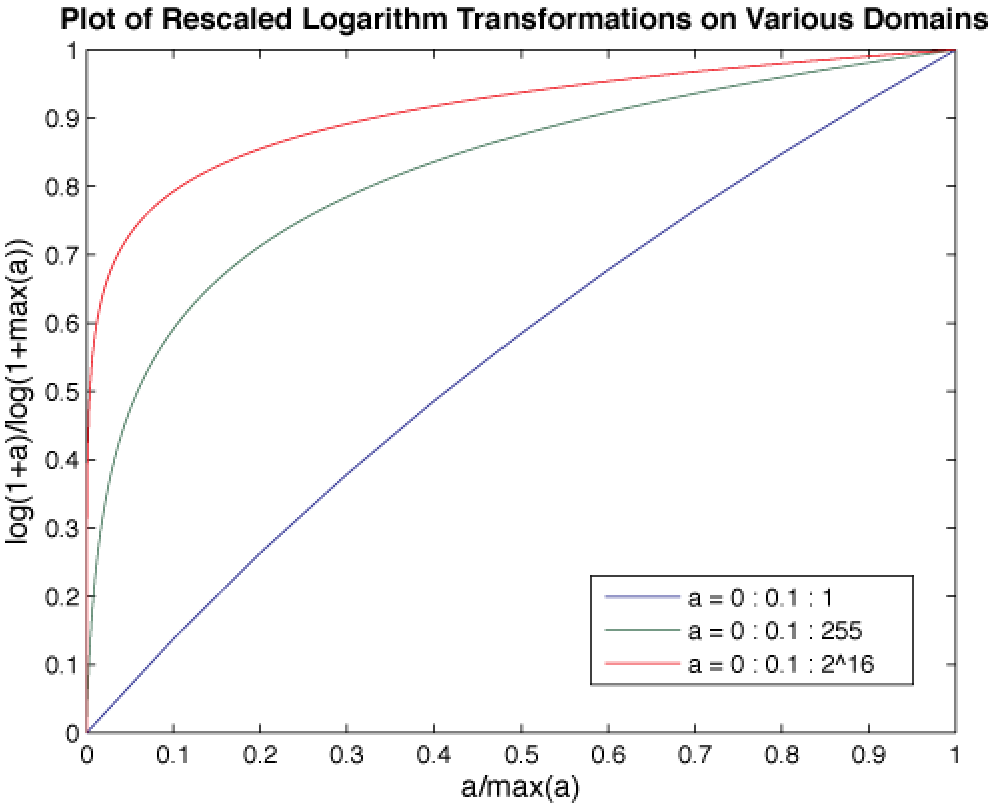
- Escala logarítmica de contraste

  $s = c \log(1+p)$ [Gonzalez & Woods, 1993]
- aplica uma escala logarítmica e normaliza o resultado para o histograma 256
- Esta técnica apresenta bons resultados para aumentar o contraste de imagens
- Ferramentas de processamento permitem que o usuário ‘desenhe’ a função logarítmica


Plot of Logarithm Transformations With Changing c
logarithm of a
a
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
a
min(1*log(1+a),1)
min(1.45*log(1+a),1)
min(2*log(1+a),1)
min(5*log(1+a),1)

Exemplos escala log




Plot of Rescaled Logarithm Transformations on Various Domains
log(1+a)/log(1+max(a))
a/max(a)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
a = 0 : 0.1 : 1
a = 0 : 0.1 : 255
a = 0 : 0.1 : 2^16

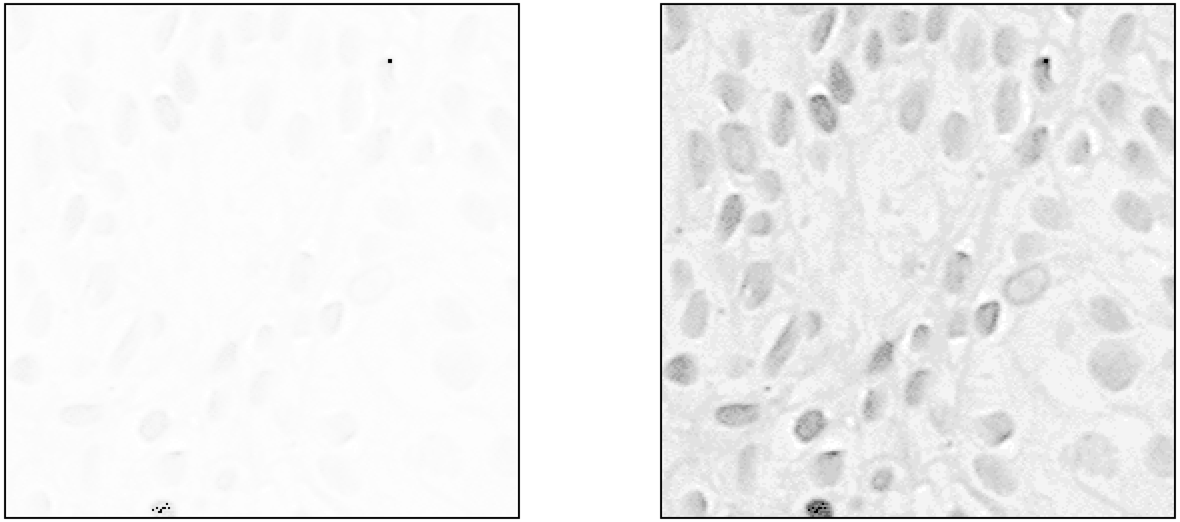
Ajustes de brilho e contraste

- Escala exponencial de contraste

$$s = c \exp(1+p)$$

- Similar à técnica logarítmica, no entanto é empregada em imagens com muito brilho

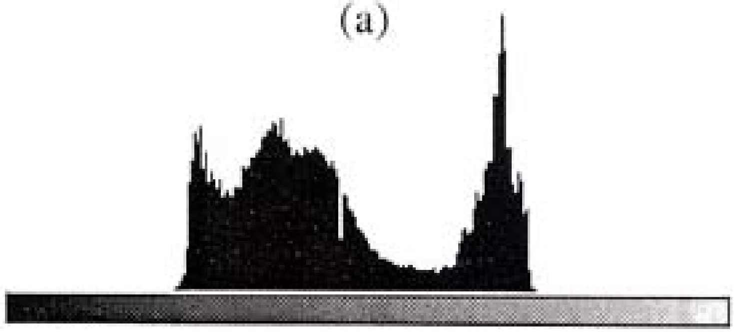
Exemplos escala exponencial



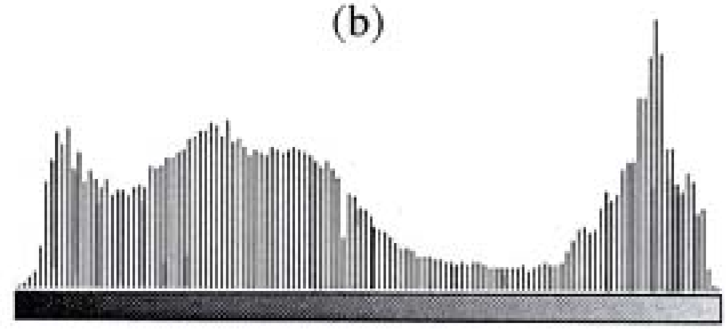
Expansão de histograma

(a)

(b)

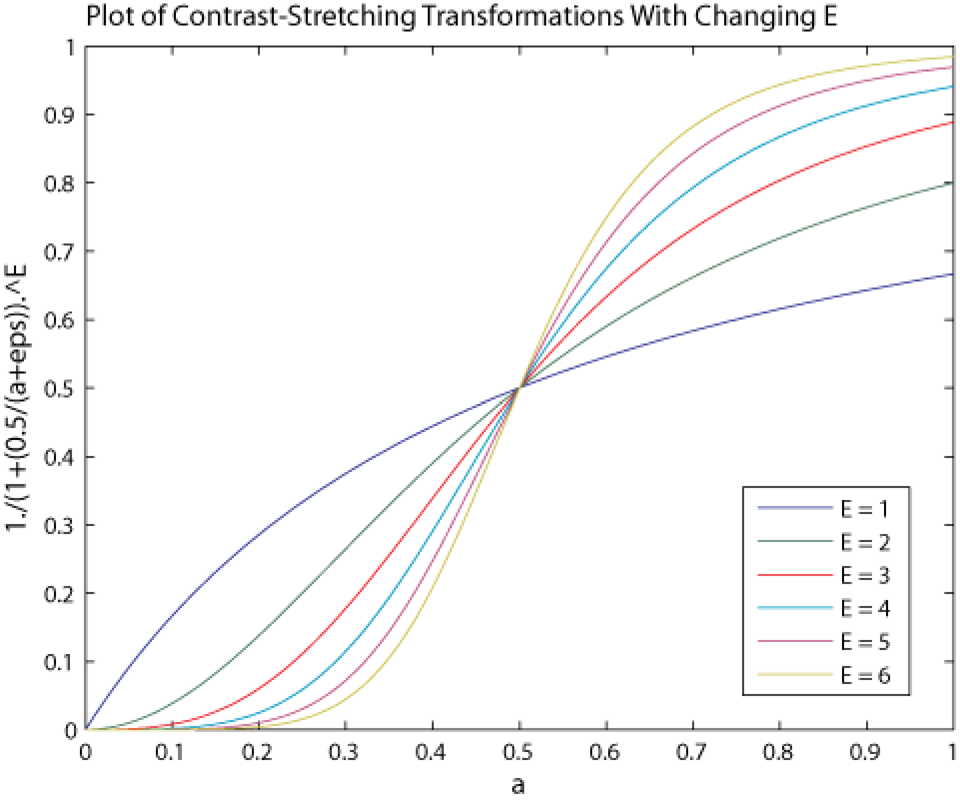


(a)



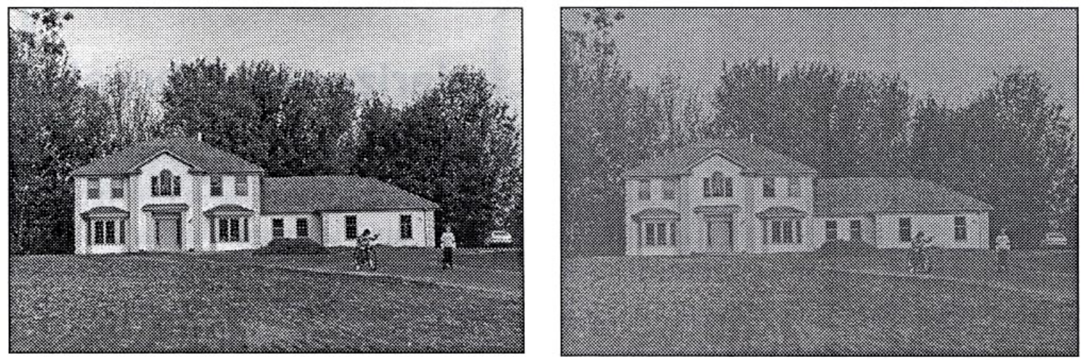
(b)

Função para transformação de *contrast stretching*

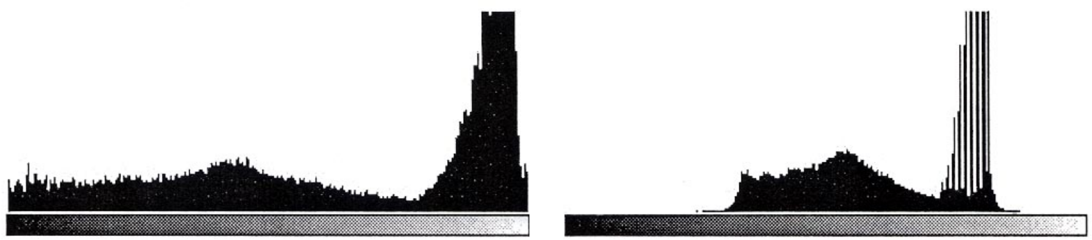


$$s = \frac{1}{(1+\frac{m}{(p+\epsilon)})^{E}}$$

Compressão de histograma



(a) (b)

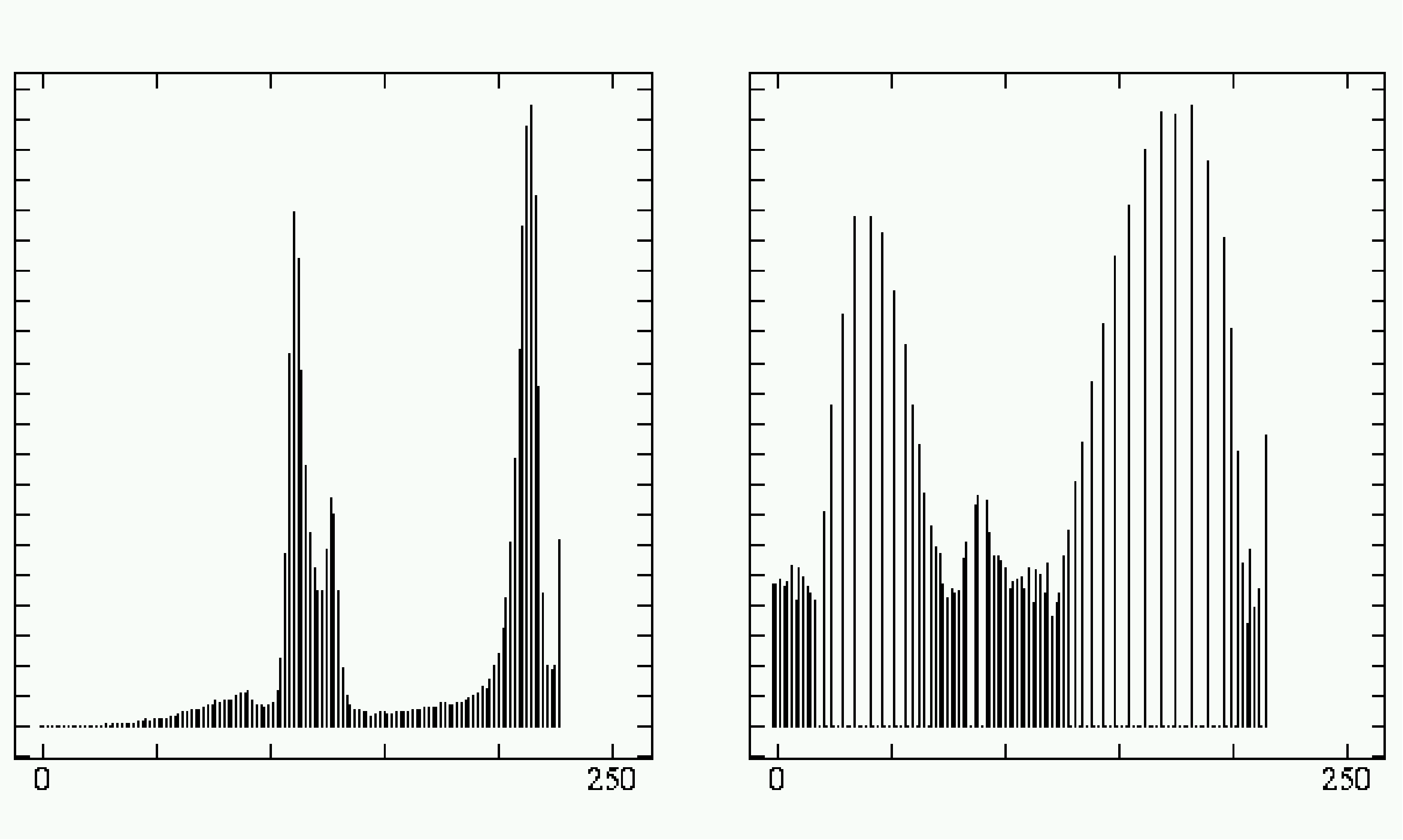


(a) (b)

Equalização de Histograma

- Cria uma imagem com níveis de cinza igualmente distribuídos ao longo da escala de cinza
- Muitas vezes melhora a qualidade visual da imagem
- Utiliza função auxiliar (ou de transformação)
  - cdf: função de distribuição acumulada, por exemplo

Equalização



Equalização

- Histograma acumulado
- Função de distribuição acumulada (*cumulative distribution function*)

$$S_k = T(r_k) = \Sigma \ n_j/n = \Sigma \ p_r(r_j) \quad (0 \geq j \geq k)$$

Histograma

Nível de Cinza (r_k)	n_k	$p_r(r_k)$
0	1120	0,068
1/7 (0,142)	3214	0,196
2/7 (0,285)	4850	0,296
3/7 (0,428)	3425	0,209
4/7 (0,571)	1995	0,122
5/7 (0,714)	784	0,048
6/7 (0,857)	541	0,033
1	455	0,028
TOTAL	**16384**	**1**

Fazendo-se somas sucessivas com a função **cdf**, teremos:

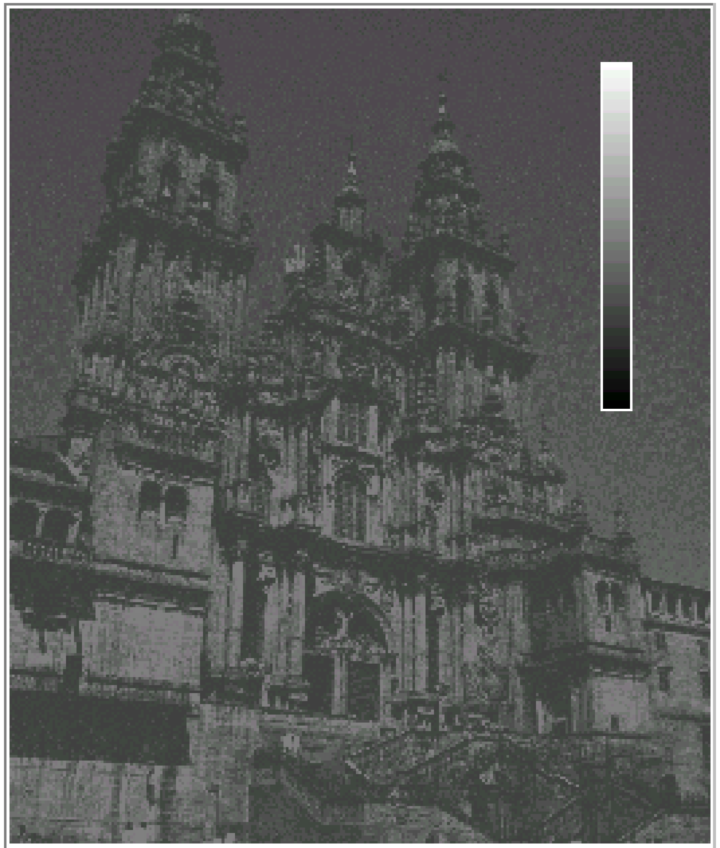
S_0 = 0.068, S_1=0.264, S_2=0.560, ... S_6=0,972, S_7=1,000

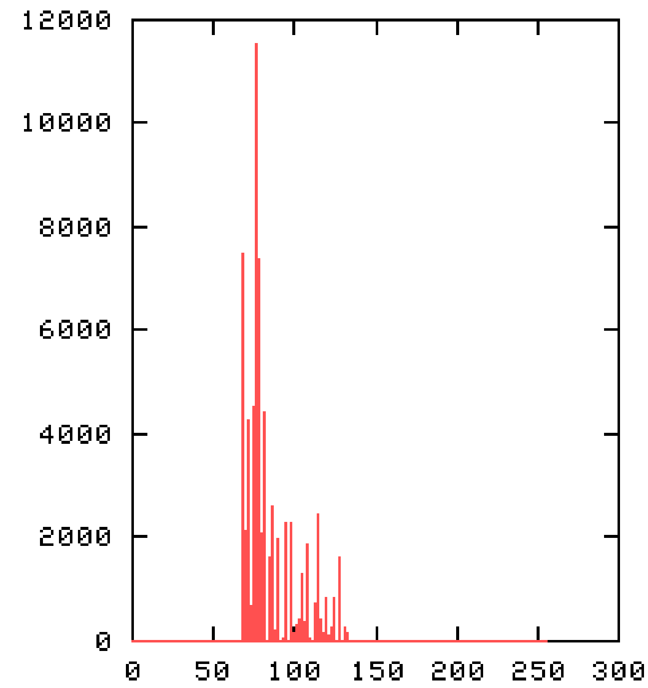
Equalização

- Imagem quantizada com 8 níveis de cinza
  - cada valor s_j deverá ser arredondado para o valor válido mais próximo
    - $s_0 \approx 0$, $s_1 \approx 2/7$, $s_2 \approx 4/7$, $s_3 \approx 5/7$
    - $s_4 \approx 6/7$, $s_5 \approx 1$, $s_6 \approx 1$, $s_7 \approx 1$

Após equalização

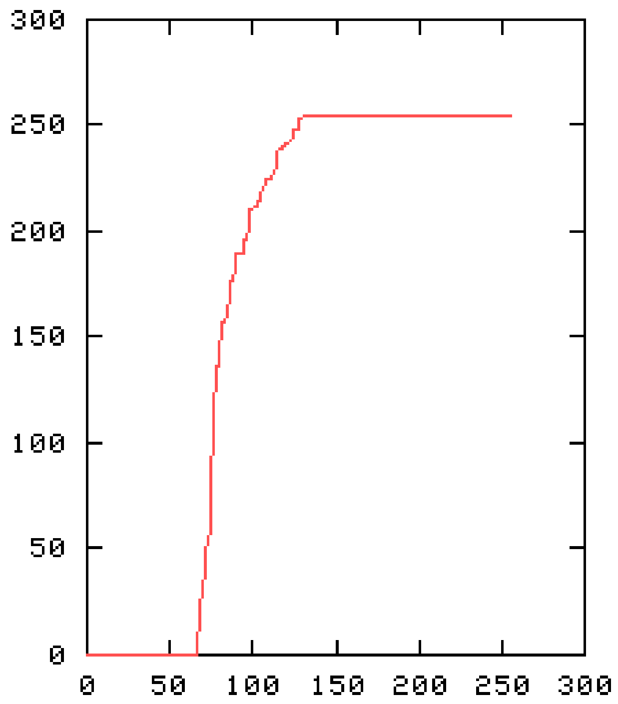
Nível de Cinza (r_k)	n_k	$p_r(r_k)$
0	1120	0,068
1/7	0	0,000
2/7	3214	0,196
3/7	0	0,000
4/7	4850	0,296
5/7	3425	0,209
6/7	1995	0,122
1	1780	0,109
TOTAL	**16384**	**1**





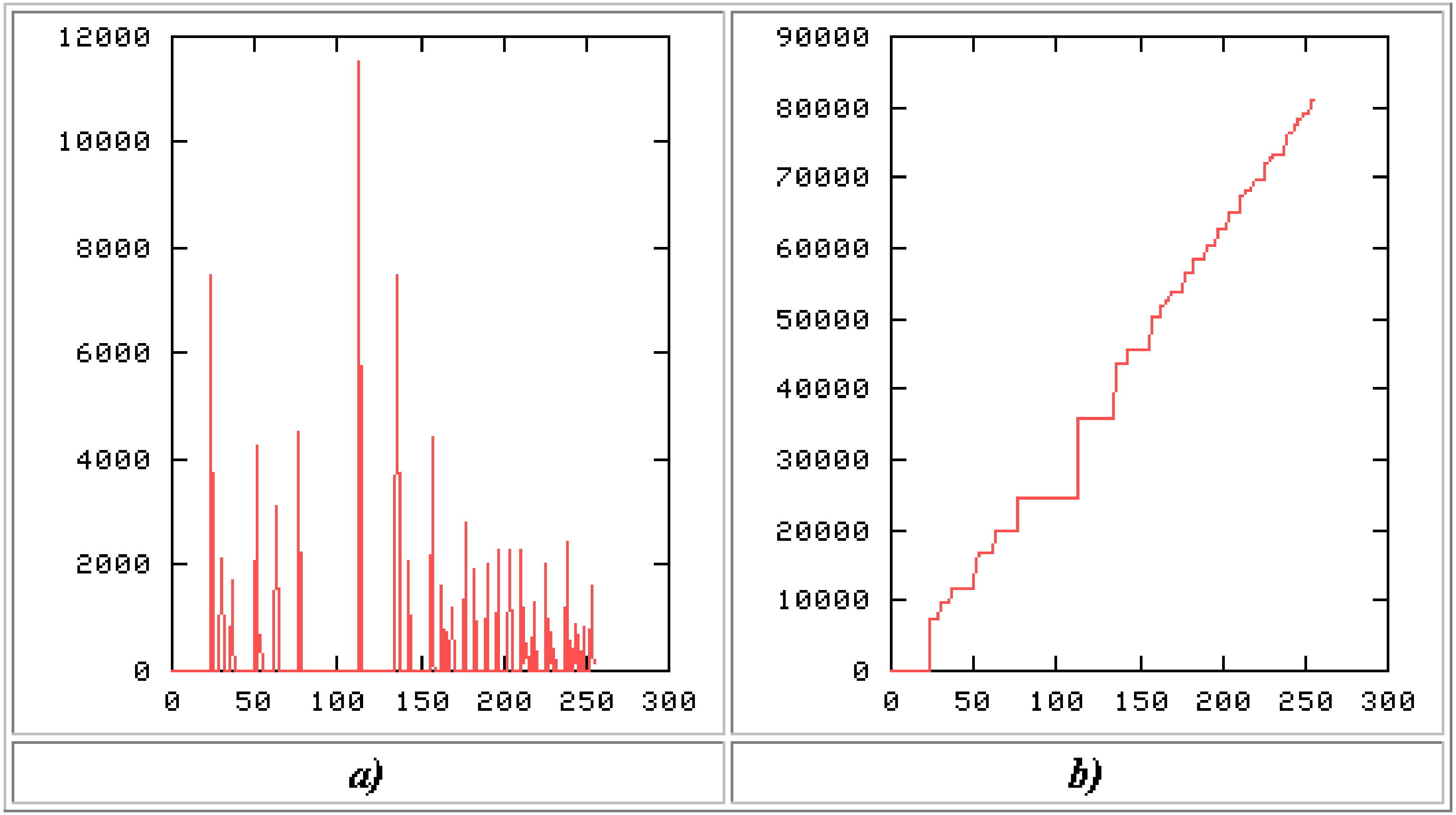
a) b)

a) Imagem Original
b) Histograma



a) b)

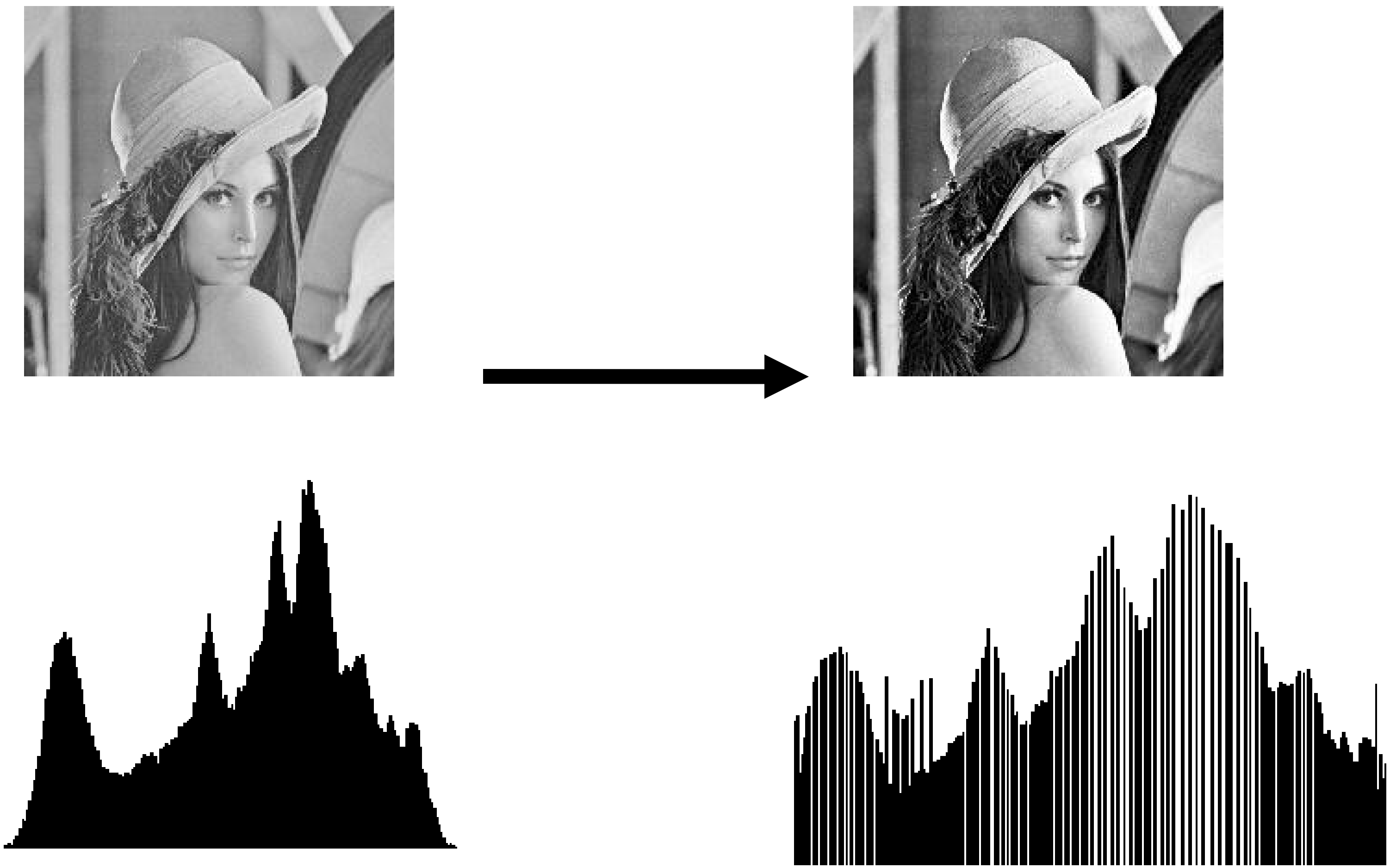
a) Histograma acumulado da imagem original
b) Imagem equalizada

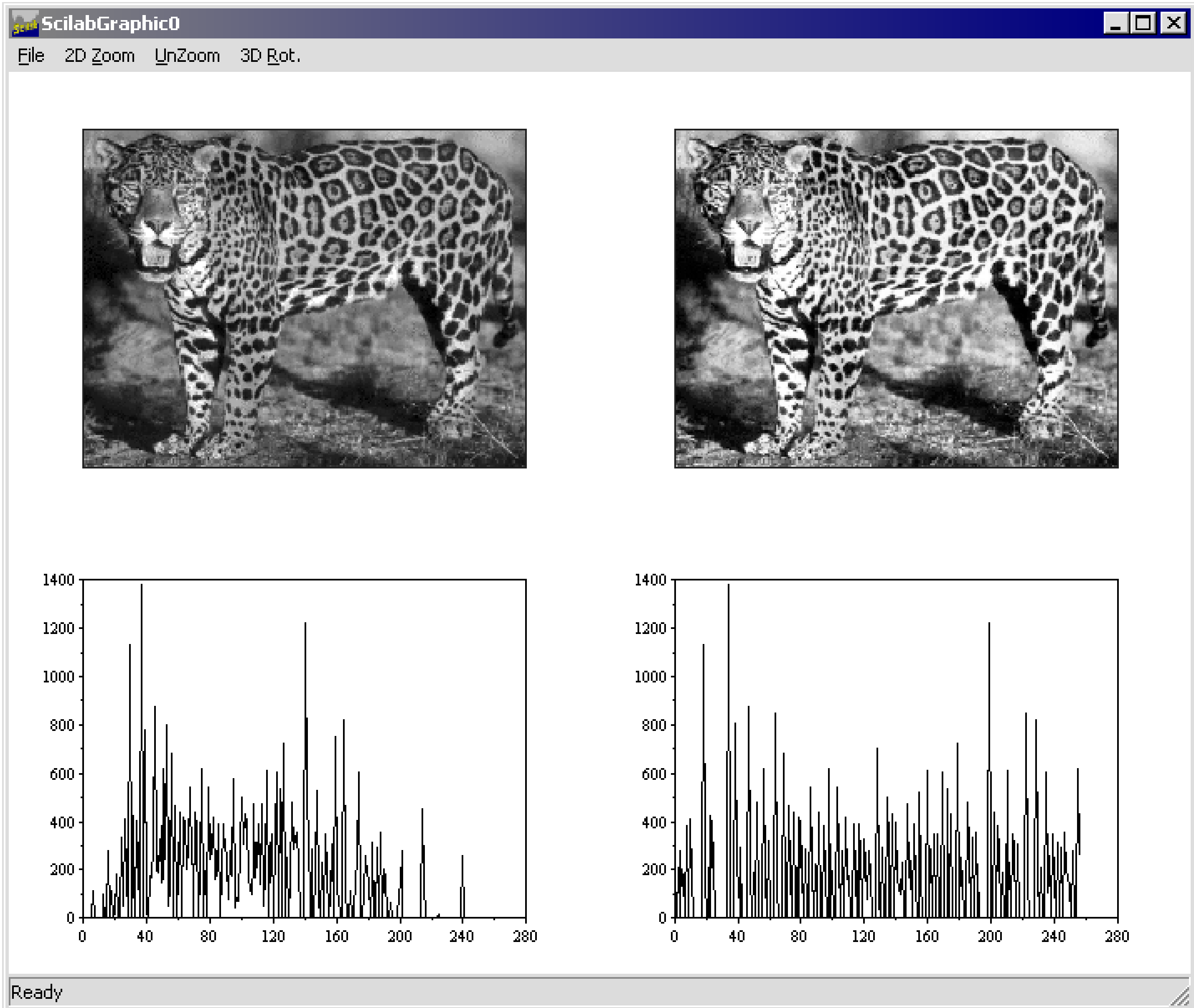


a) Histograma da imagem equalizada (devido à discretização, não gera valores constantes)

b) aproximação a uma reta

Mais exemplos




ScilabGraphic0
File 2D Zoom UnZoom 3D Rot.
1400
1200
1000
800
600
400
200
0
0 40 80 120 160 200 240 280
Ready

4 Categorias

- Transformações de brilho (histograma, etc.)
- Transformações geométricas (distorções)
- Métodos de filtragem espacial que consideram uma vizinhança local a cada pixel (suavização, filtros)
- Restauração (requer conhecimento sobre a imagem investigada)

Pré-processamento local

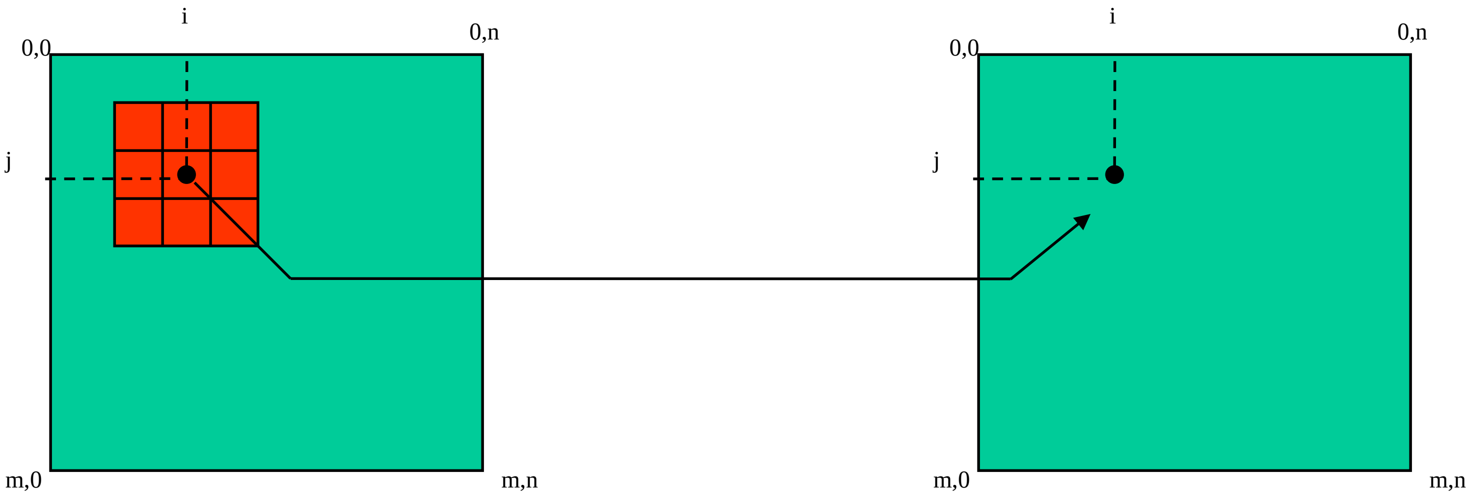
- Utiliza uma vizinhança de poucos pixels para produzir novas intensidades de pixel
- Duas categorias de técnicas
  - **suavização** (*smoothing*): suprimir ruído ou pequenas flutuações nas imagens (supressão de altas frequências)
  - **operadores de gradiente**: derivadas locais das funções de imagem. Por quê? Derivadas são maiores em regiões de rápidas mudanças (supressão de baixas frequências)
- Qual domínio? Espacial ou da Frequência

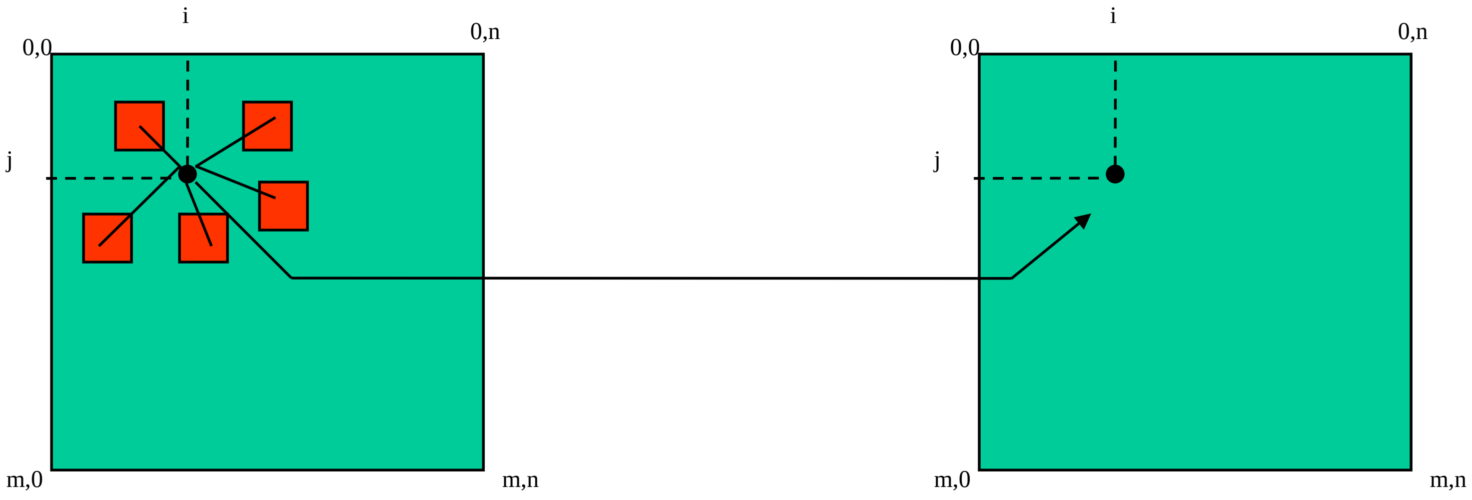
Filtros digitais no domínio do espaço

- Definição
  - Também conhecidos como operadores locais
  - Os filtros locais são técnicas baseadas na **convolução** de *templates* (janelas, matrizes, máscaras) ou tuplas (conjunto de pixels)

  - Uma grande variedade de filtros digitais pode ser implementada por meio da convolução no domínio espacial

Templates x tuplas





Convolução de gabaritos (*templates*)

- Se T(x,y) é um gabarito (*m* x *n*) e I(x,y) é a Imagem (M x N)
- A convolução de T⊗I é dada por:

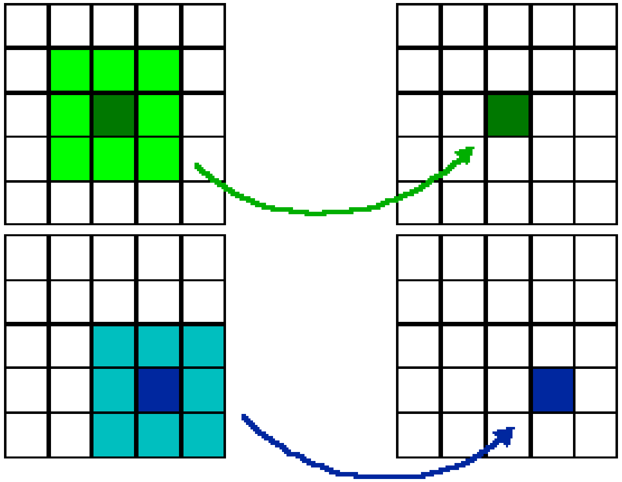
$$T \otimes I(x,y) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} T(i,j) I(x+i, y+i)$$

- Ou ainda ....

Alguns conceitos importantes

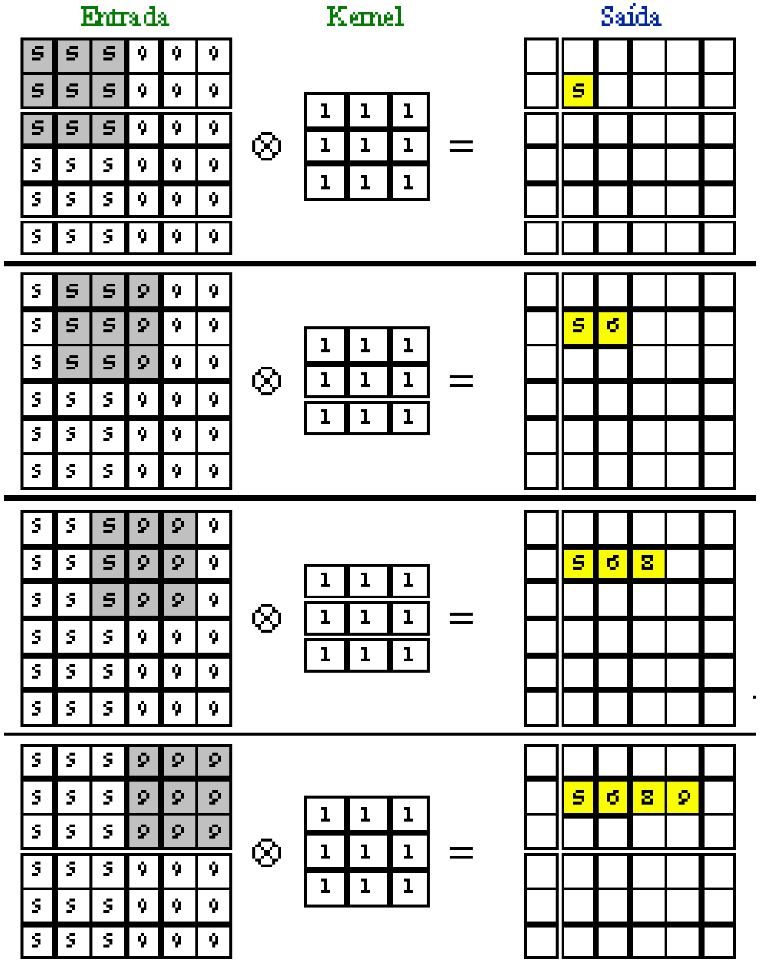
Convolução discreta $$g(x) = f(x) * h(x) = \sum_{all\ k} h(x-k) f(k)$$

h(x): resposta de impulso (*impulse response*), *kernel*, máscara, filtro ou *template*

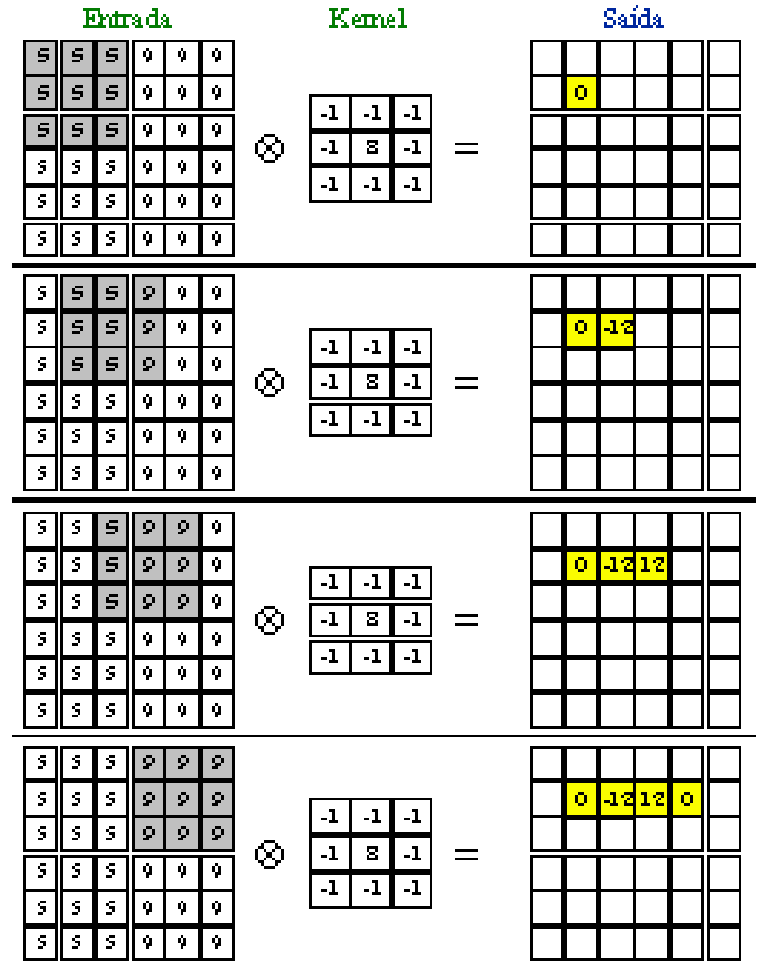


Exemplos de convolução

☞ Filtro Passa-baixa

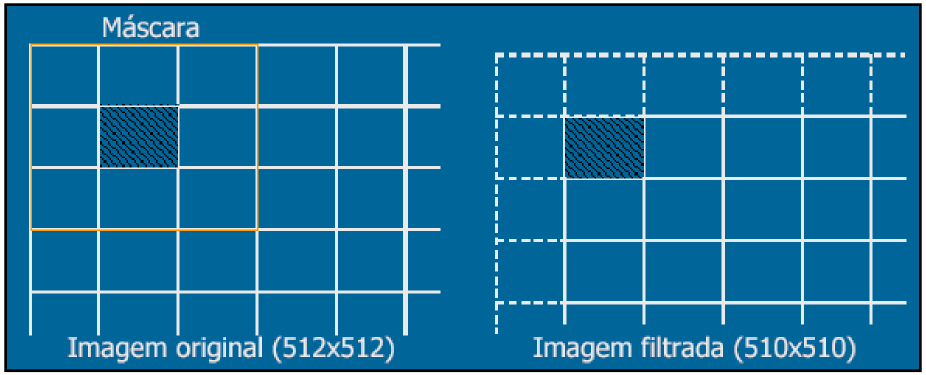


☞ Filtro Passa-alta



Tratamento das extremidades da imagem

- Convolução aperiódica
  - Não considera os pontos da imagem onde o gabarito não se encaixa na imagem



- Convolução Periódica
  - O gabarito é deslocado sobre as extremidades da Imagens como se estas fosses adjacentes

- Gabarito Truncado
  - Quando o gabarito não se encaixa na imagem o gabarito é truncado

Convolução x Custo Computacional

- O custo computacional da convolução é alto
  - Em uma imagem M x M e um *template* N x N são realizadas M^2 x $N^{2.}$
  - Se a imagem é de 512 x 512 e o gabarito é de 16 x 16, são necessárias 67.108.864 multiplicações
  - Computação paralela

Filtros Digitais – gabarito

- Passa Baixa e média espacial
- Filtragem mediana
- Passa Alta
- Passa Banda
- Gradientes (roberts, sobel, etc.)

Suavização

Detectores de Bordas

Filtro Passa-Baixa

- Suavização ("*Smoothing*") da imagem
- Redução do efeito de ruído e detalhes irrelevantes
- Maior a máscara, maior efeito de borramento

$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} \qquad \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} * 1/16 \qquad \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} * 1/10$$

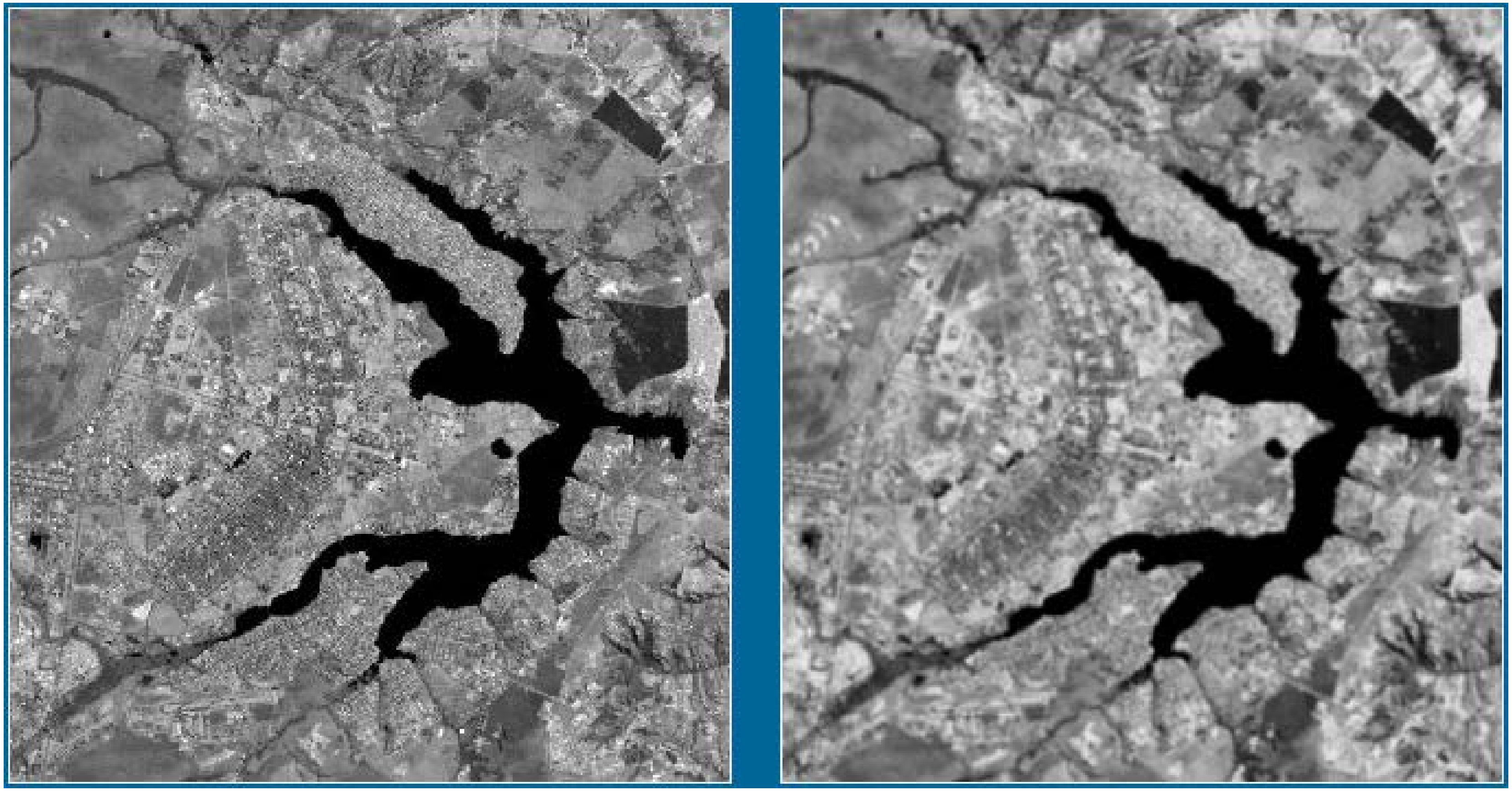
Filtro Passa-Baixa

- Filtro Gaussiano
  - "semelhante" ao filtro da média. A máscara representa o formato de uma Gaussiana.

1/256	4/256	6/256	4/245	1/256
4/256	16/256	24/256	16/256	4/256
6/256	24/256	36/256	24/256	6/256
4/256	16/256	24/256	16/256	4/256
1/256	4/256	6/256	4/245	1/256

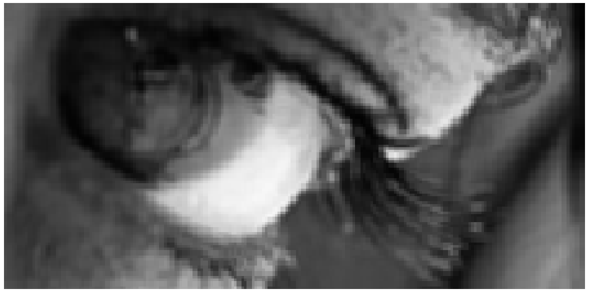
  - Curiosidade: como gerar uma máscara de convolução espacial Gaussiana? http://www.lcad.icmc.usp.br:/~jbatista/procimg/gaussian.c

Filtro Passa-Baixa



Filtro Média : Exemplo

Original



Filtragem de média 3x3

Filtragem de média 5x5

Filtro de Mediana

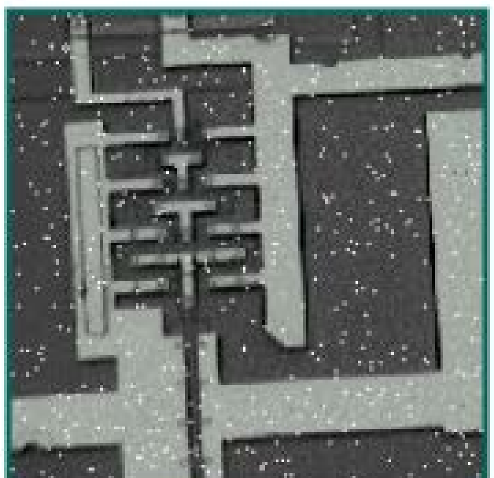
- Os filtros de mediana permitem, tal como os filtros de média, reduzir o ruído das imagens sem, no entanto, esbater as arestas e os contornos das mesmas
- O valor mediano de um conjunto é aquele para o qual existem metade dos valores menores que ele e metade maiores que ele
- Este tipo de filtro é particularmente adaptado à remoção de ruído impulsivo que aparece em regiões limitadas da imagem

Filtro de Mediana

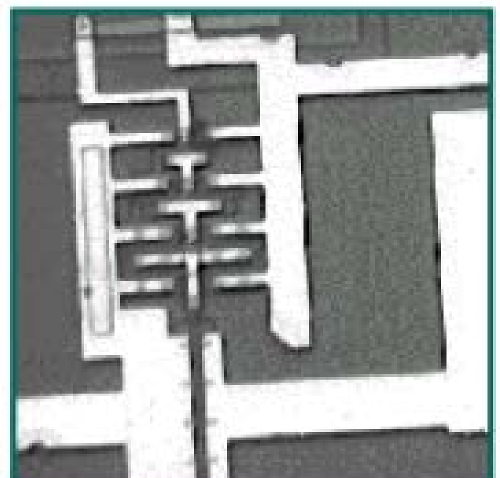
75	77	77
77	253	81
77	75	79

→ 75 75 77 77 77 77 79 81 253 →

75	77	77
77	77	81
77	75	79



Original



Filtrado com Mediana 3x3

Filtro de Mediana: Exemplo

Imagem original

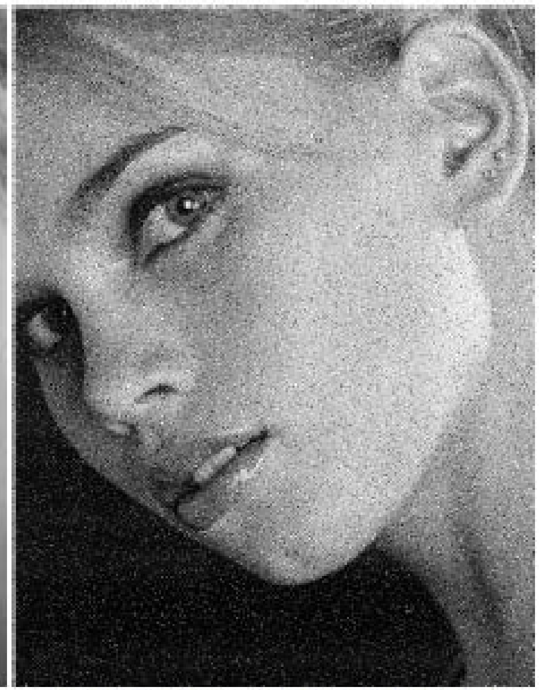


Imagem com ruído adicionado (20%)

Imagem depois de filtrada com um filtro de mediana 3x3

Comparação Mediana/Média

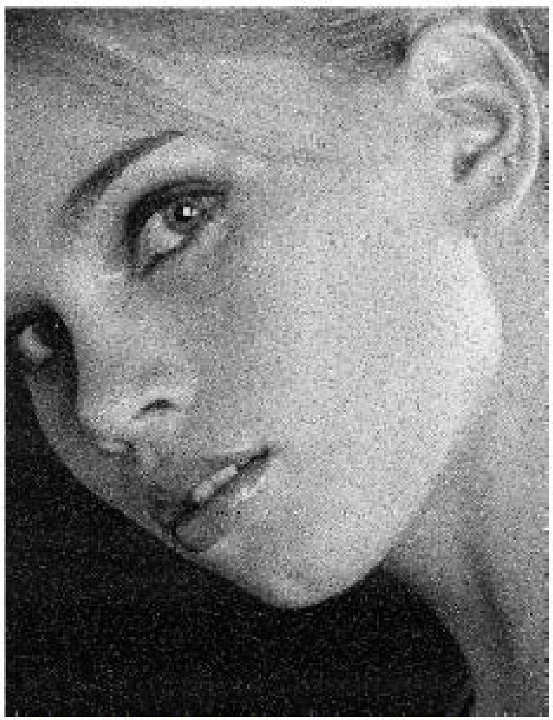


Imagem com ruído adicionado (20%)

Imagem depois de filtrada com um filtro de mediana 3x3

Imagem depois de filtrada com um filtro de média

Comparação Mediana/Média

ruído

Média 3x3

Mediana 3x3

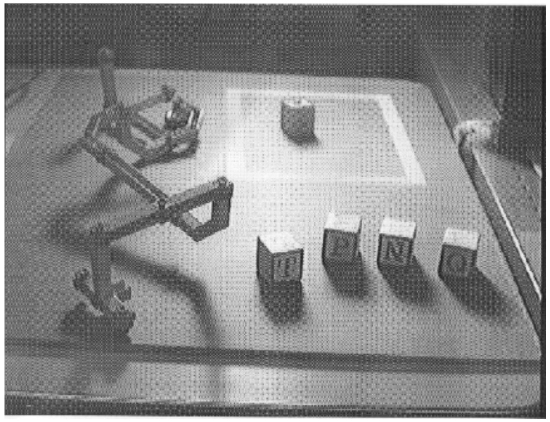
Observações

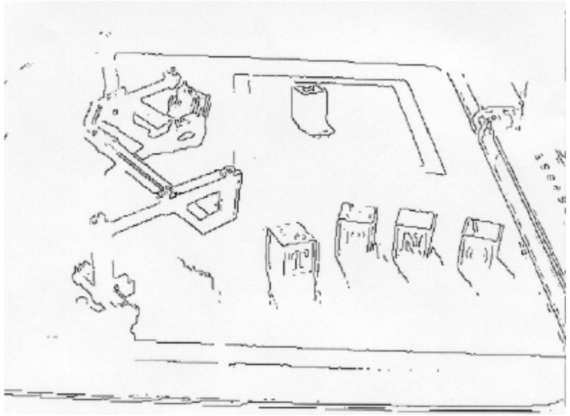
- *Edges* (arestas/bordas): pixels nos quais há uma mudança brusca na intensidade

- *Boundary* (contorno): linha fechada formada pelas bordas de um objeto

O que causa mudanças na intensidade?

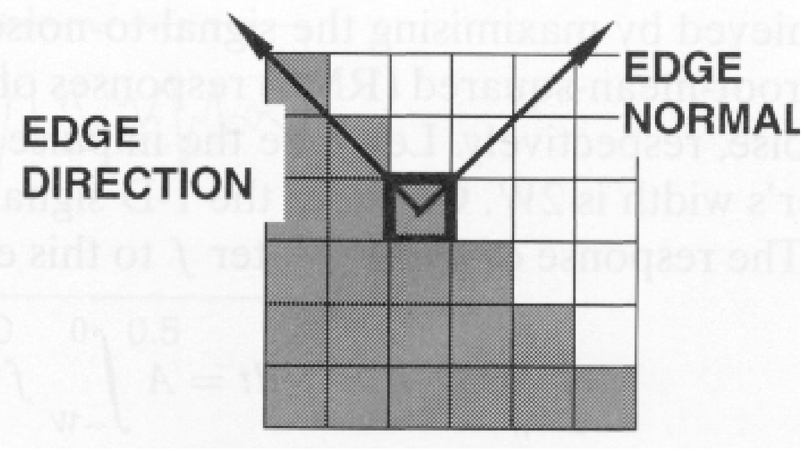
- Eventos Geométricos
  - Descontinuidades no contorno ou orientação da superfície e/ou cor da superfície e textura
- Eventos não geométricos
  - Mudança de iluminação
  - Luz especular
  - Sombras





Descritores de Borda

- **Normal:** vetor unitário na direção da mudança máxima de intensidade
- **Direção:** vetor unitário perpendicular à normal
- **Posição:** posição na imagem em que a borda está localizada
- **"Strength":** relacionado ao contraste local ao longo da normal

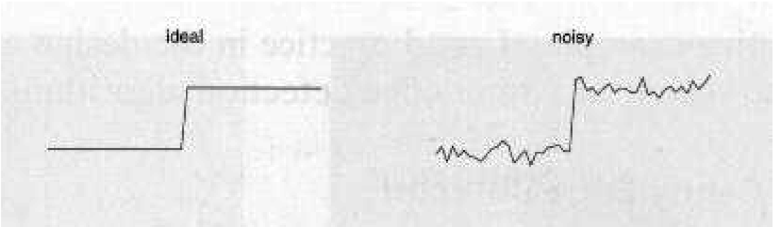


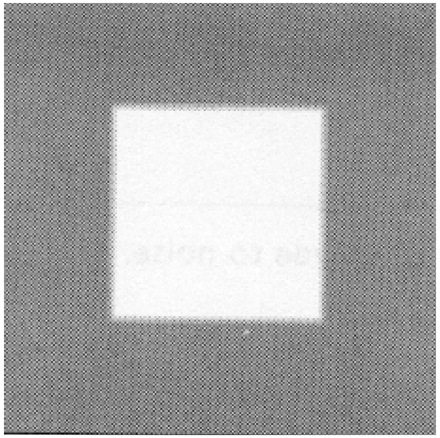
Tipos de bordas

- Degrau (*step*)
- Rampa (*ramp*)
- Platô (*ridge*)
- Telhado (*roof*)

Borda degrau (*step edge*)

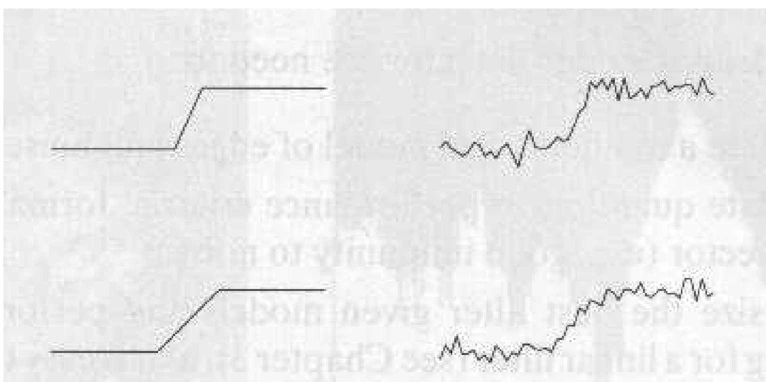
- A intensidade muda abruptamente de um valor (de um lado) para um outro valor (para o lado oposto da descontinuidade)

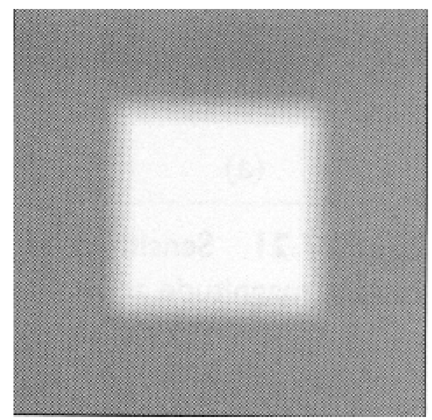




Borda rampa (*ramp edge*)

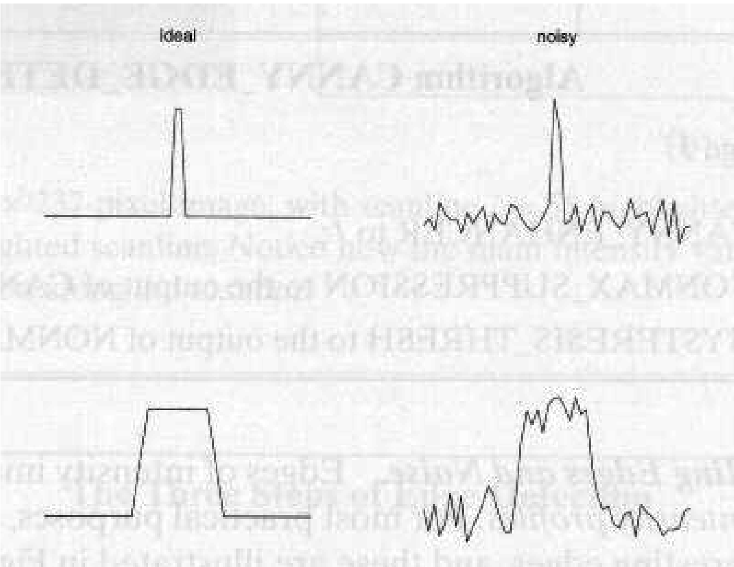
- Um degrau em que a mudança de intensidade não é instantânea, mas ocorre em uma distância finita





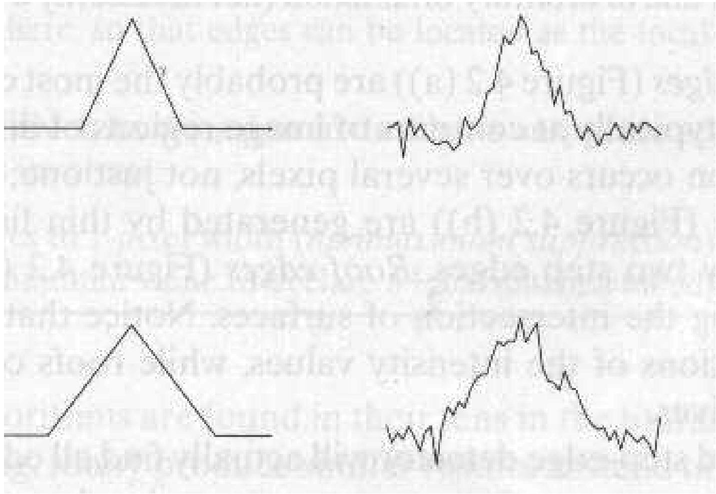
Borda platô *Ridge edge*

- Mudança abrupta de intensidade, voltando ao ponto de partida após curta distância (geralmente causada por linhas na imagem)



Borda Telhado (*roof edge*)

- Um *ridge edge* ocorre quando a mudança de intensidade não é instantânea, ocorrendo numa distância finita (geralmente causada pela intersecção de duas superfícies)



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 enfatizam o ruído existente na imagem.

Principais etapas na detecção de bordas

- Suavização: eliminar ruídos desnecessários
- Agudização (*Enhancement*): aplicar filtro para realçar a qualidade das bordas na imagem
- Detecção: determinar que bordas devem ser descartadas (ruído) e quais devem ser mantidas
  - Limiarização na “strength” da borda !
- Localização: determinar local exato da bordas

Tratamento matemático de arestas

- Derivadas: representam mudanças em funções contínuas, assim operadores que descrevem arestas podem ser expressos em termos de derivadas parciais
- Mudanças na função imagem podem ser representadas pelo Gradiente
- Uma aresta é uma variável vetorial, com magnitude e direção

Detectores de Borda

- Os detectores de bordas são baseados em diferenciação, sendo que pode-se considerar derivadas de primeira ordem ou de segunda ordem

- gradiente ->borda -> módulo do vetor

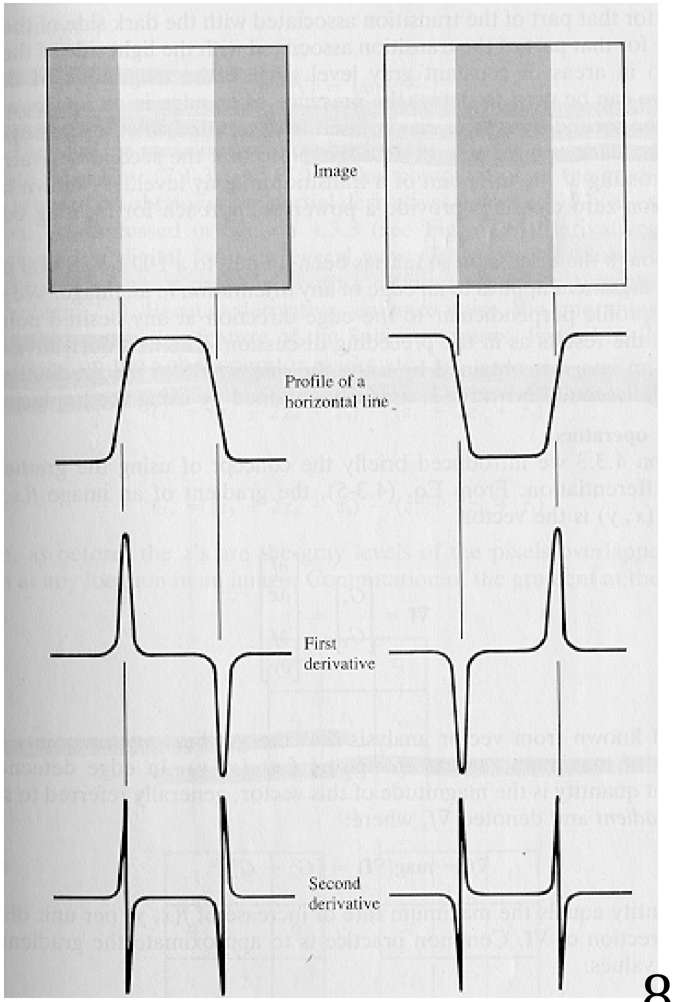
- laplaciano ->borda-> cruzamento por zero

Detecção de bordas por derivadas

- Pontos que estão nas bordas podem ser localizados

  (1) detectando-se um local **maxima** ou **minima** da primeira derivada

  (2) detectando-se os **zero-crossings** da segunda derivada



Detecção de borda usando 1ª Derivada – **1D**

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

mask: $[-1 \quad 1]$

$\textbf{mask M} = [-\mathbf{1}, \mathbf{0}, \mathbf{1}]$ (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

Detecçãode borda usando 2ª Derivada – **1D**

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

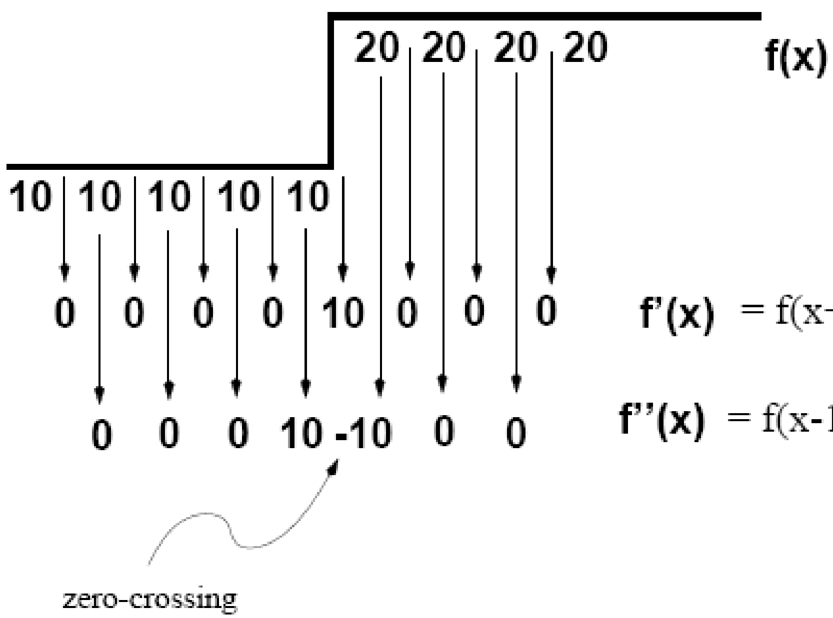
$$f(x+2) - 2f(x+1) + f(x) \quad (h=1)$$

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

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

mask: $[1 \quad -2 \quad 1]$

Detecção de borda usando 2ª Derivada – **1D**



f'(x) $= f(x+1) - f(x)$ (approximates f'() at x+1/2)

f''(x) $= f(x-1) - 2f(x) + f(x+1)$ (approximates f''() at x)

Detecção de borda usando 2ª 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

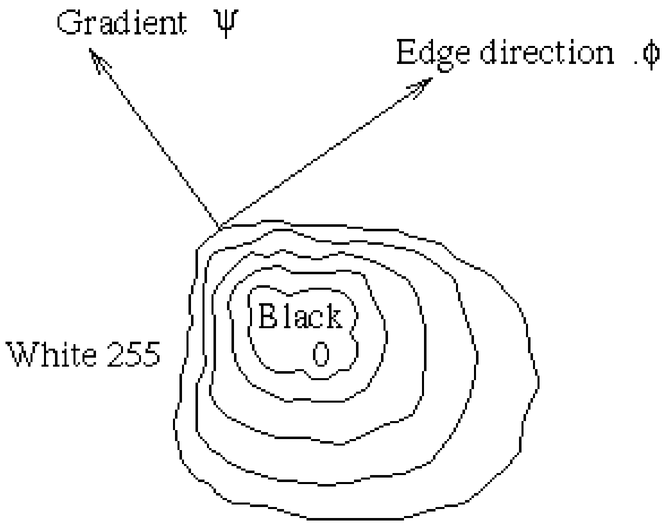
Borda telhado

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

Considerações sobre bordas e derivadas

- Primeira ordem
  - Produzem bordas mais espessas
  - A magnitude da resposta é menor, pois é menos agressiva na hora de facilitar as mudanças bruscas
- 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 (derivada positiva)

O Gradiente



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

$$\psi = arg(\frac{\partial g}{\partial x}, \frac{\partial g}{\partial y})$$

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

Detecção de Borda com 1ª Derivada – **2D** Gradiente

- O Gradiente é definido como:

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

- Aproximação do gradiente por diferenças finitas:

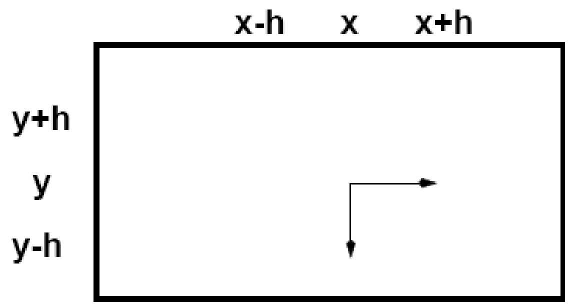
$$\frac{\partial f}{\partial x} = \frac{f(x+h_x, y) - f(x, y)}{h_y} = f(x+1, y) - f(x, y), \ (h_x = 1)$$

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

Detecção de Borda com 1ª Derivada – **2D**

Gradiente

- Na notação de coordenadas de pixel, *j* corresponde à direção *x* e *i* à direção negativa *y*



-1	1

1
-1

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

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

Observações sobre 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
- Kirsh

Operador baseado na 1ª Derivada – **2D**
Roberts

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

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

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

M_x e M_y são aproximações em i+ ½ e j + ½

Operadores de Roberts

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

1	0
0	-1

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

0	1
-1	0

0	0	50	0	0	...
0	0	50	0	0	...
0	0	50	0	0	...
0	0	50	0	0	...
0	0	50	0	0	...
0	0	50	0	0	...
0	0	50	0	0	...
0	0	50	0	0	...
0	0	50	0	0	...
...	...	...	...	...	...

$$F(i,j) = \left| g(i,j) - g(i+1, j+1) \right| + \left| g(i, j+1) - g(i+1, j) \right|$$

0	0	100	0	0	...
0	0	100	0	0	...
0	0	100	0	0	...
0	0	100	0	0	...
0	0	100	0	0	...
0	0	100	0	0	...
0	0	100	0	0	...
0	0	100	0	0	...
0	0	100	0	0	...
...	...	...	...	...	...

Operadores de Roberts

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
0	0	0	0	0	0

1	0
0	-1

0	0	0	0	0	...
0	0	0	0	0	...
0	0	0	0	0	...
0	0	0	0	0	...
0	0	0	0	0	...
0	0	0	0	0	...
0	0	0	0	0	...
0	0	0	0	0	...
0	0	0	0	0	...
...	...	...	...	...	...

0	1
-1	0

0	0	0	0	0	...
0	0	0	0	0	...
50	0	0	0	0	...
50	50	0	0	0	...
0	50	50	0	0	...
0	0	50	50	0	...
0	0	0	50	50	...
0	0	0	0	50	...
0	0	0	0	0	...
...	...	...	...	...	...

$$F(i,j) = \left|g(i,j) - g(i+1,j+1)\right| + \left|g(i,j+1) - g(i+1,j)\right|$$

0	0	0	0	0	...
0	0	0	0	0	...
50	0	0	0	0	...
50	50	0	0	0	...
0	50	50	0	0	...
0	0	50	50	0	...
0	0	0	50	50	...
0	0	0	0	50	...
0	0	0	0	0	...
...	...	...	...	...	...

Operador baseado na 1ª Derivada – **2D**
Outra aproximação

- Considere o arranjo de pixeis 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

- Setando c = 1, temos o operador Prewitt !

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

M_x e M_y são aproximações em (i ,j)

Operador baseado na 1ª Derivada – **2D** Sobel

- Fazendo c = 2, temos Sobel:

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

M_x e M_y são aproximações em (i ,j)

Sobel

a	b	c
d	e	f
g	h	i

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

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

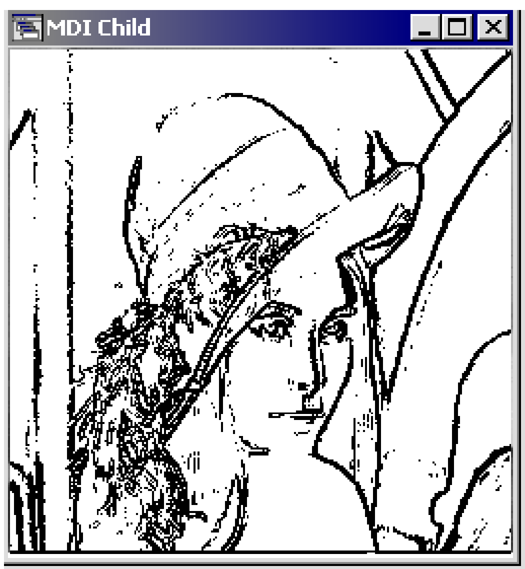
$$S = \sqrt{S_x^2 + S_y^2}$$

Ou numa versão simplificada:

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

Gradiente Sobel

Sx Sy



Bordas = thresholding $\left|\vec{\nabla} g\right|$

Operadores de Sobel

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

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

...	...	...	...	...	...
...	0	200	200	0	...
...	0	200	200	0	...
...	0	200	200	0	...
...	0	200	200	0	...
...	0	200	200	0	...
...	0	200	200	0	...
...	0	200	200	0	...
...	0	200	200	0	...
...	...	...	...	...	...

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

...	...	...	...	...	...
...	0	0	0	0	...
...	0	0	0	0	...
...	0	0	0	0	...
...	0	0	0	0	...
...	0	0	0	0	...
...	0	0	0	0	...
...	0	0	0	0	...
...	0	0	0	0	...
...	...	...	...	...	...

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

...	...	...	...	...	...
...	0	200	200	0	...
...	0	200	200	0	...
...	0	200	200	0	...
...	0	200	200	0	...
...	0	200	200	0	...
...	0	200	200	0	...
...	0	200	200	0	...
...	0	200	200	0	...
...	...	...	...	...	...

a	b	c
d	e	f
g	h	i

Operadores de Sobel

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
0	0	0	0	0	0

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

...	...	...	...	...	...
...	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	...
...	...	...	...	...	...

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

...	...	...	...	...	...
...	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	...
...	...	...	...	...	...

$S = |S_x| + |S_y|$

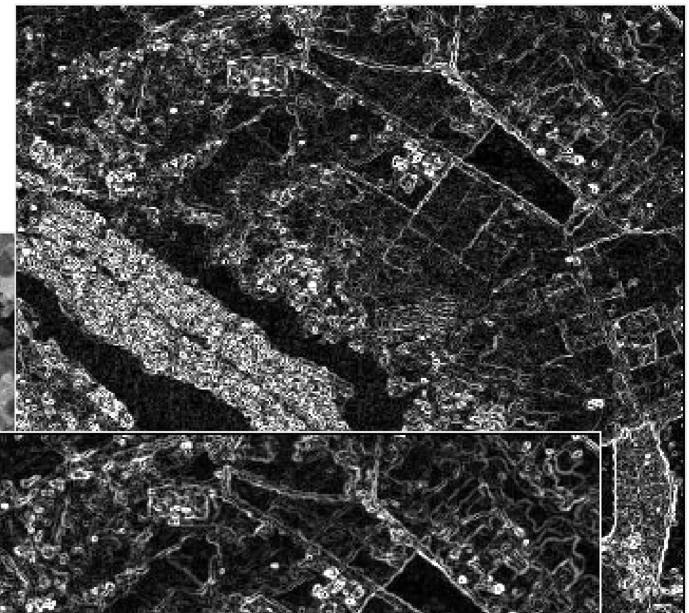
...	...	...	...	...	...
...	0	0	0	0	...
...	100	0	0	0	...
...	300	100	0	0	...
...	300	300	100	0	...
...	100	300	300	100	...
...	0	100	300	300	...
...	0	0	100	300	...
...	0	0	0	100	...
...	...	...	...	...	...

a	b	c
d	e	f
g	h	i

Operadores de Sobel

Comparação

Roberts

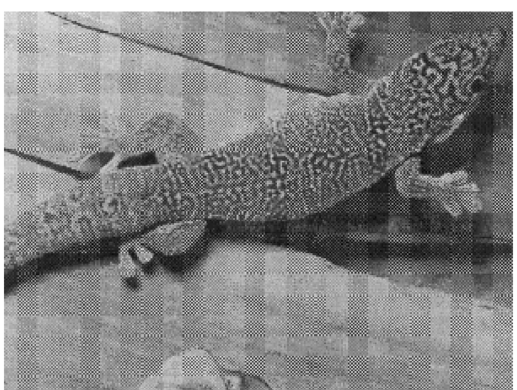


Original

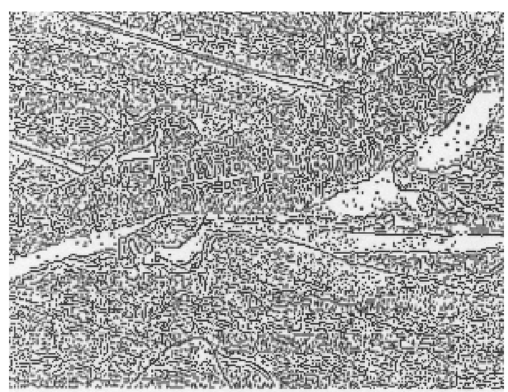
Sobel

Questões práticas sobre filtros

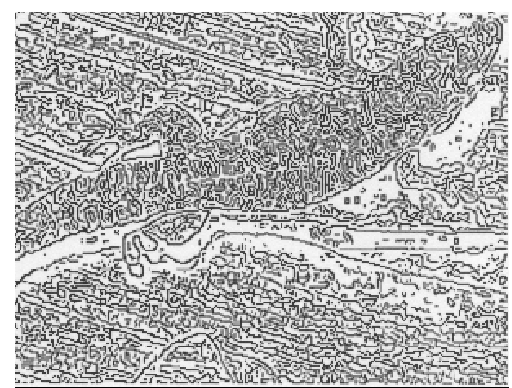
- Balanço entre supressão-localização de ruído
  - Filtros maiores reduzem ruído, mas pioram a localização (adicionam incertezas sobre a localização das arestas) e vice-versa



Filtro menor

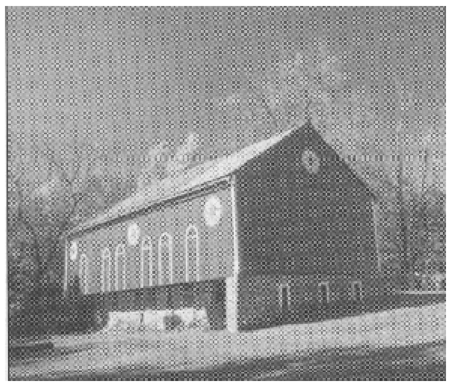


Filtro maior

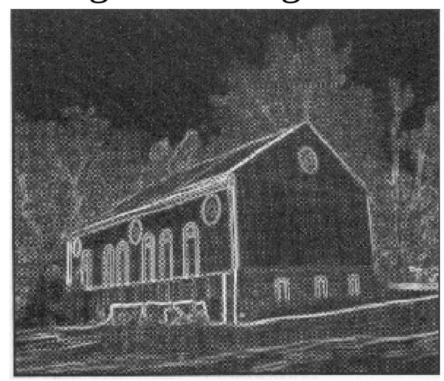


Questões práticas sobre filtros

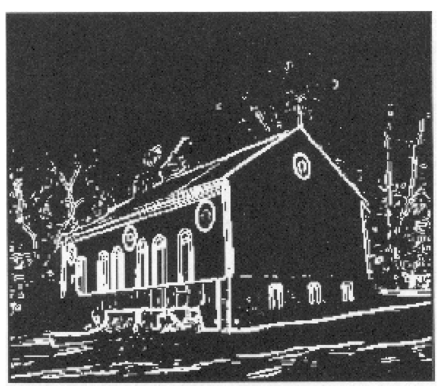
- Como escolher o limiar ?



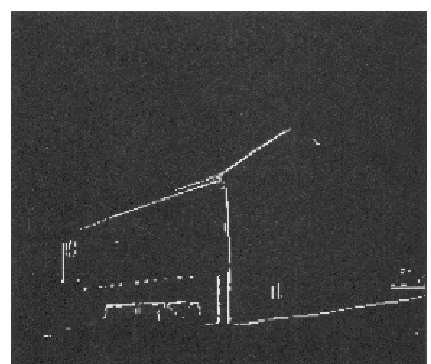
Magnitude do gradiente



Baixo threshold

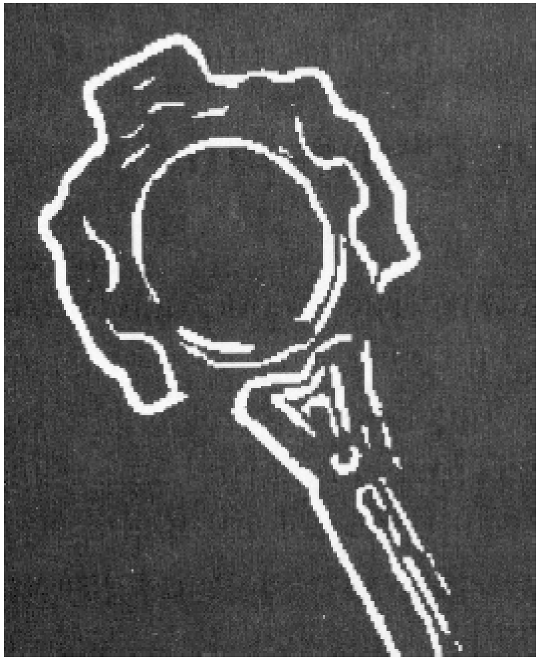


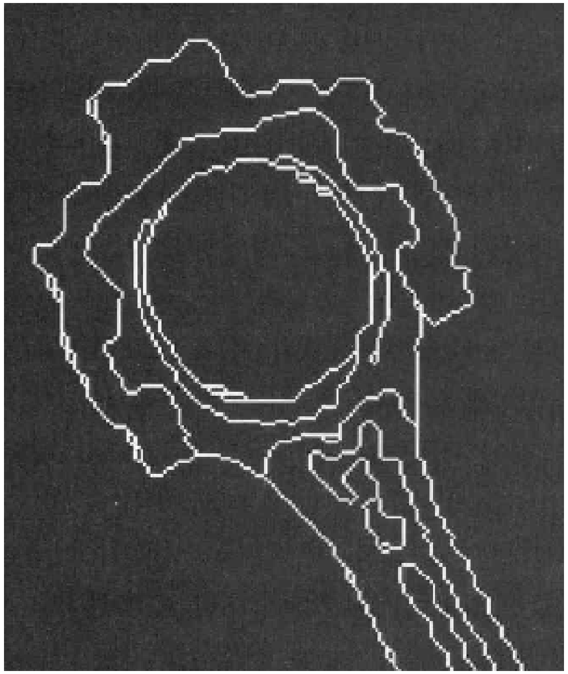
Alto threshold



Questões práticas sobre filtros

- Afinamento e "linking" normalmente são operações necessárias !





Detectores de borda com 2ª 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:

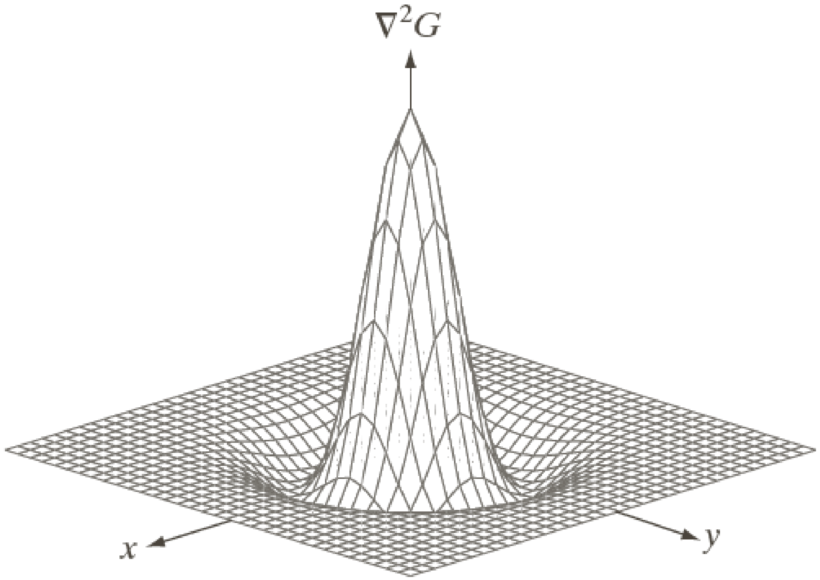
0	1	0
1	-4	1
0	1	0

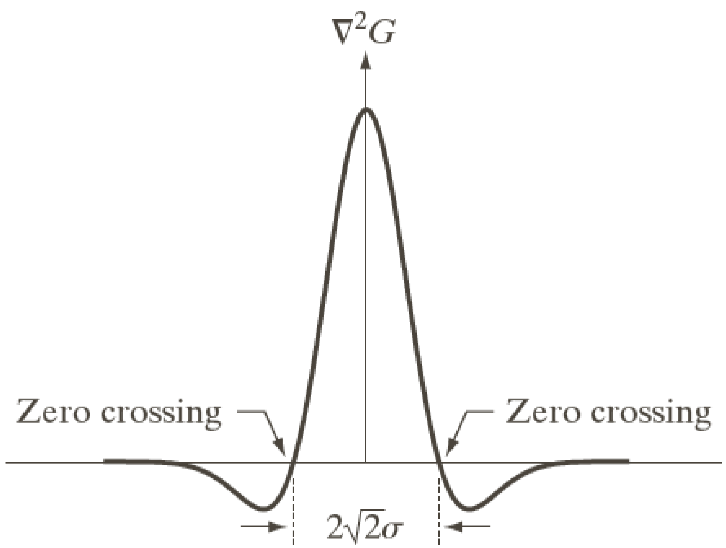
Detectores de borda com 2ª Derivada – **2D**
Exemplo Laplaciano

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	-
-	-	-	-	-	-





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

a b
c d

FIGURE 10.21
(a) Three-dimensional plot of the *negative* of the LoG. (b) Negative of the LoG displayed as an image. (c) Cross section of (a) showing zero crossings. (d) 5×5 mask approximation to the shape in (a). The negative of this mask would be used in practice.

Detectores de borda com 2ª Derivada – **2D**
Propriedades Laplaciano

- É um operador isotrópico (não diferencia direções)
- Fácil de implementar (basta uma máscara)
- Não dá informação sobre a direção da borda
- Mais sensível ao ruído (i.e., deriva duas vezes!)

Detectores de borda com 2ª Derivada – **2D Laplaciano do Gaussiano (LoG)**

- Para reduzir o efeito do ruído, a imagem é primeiramente suavizada com um filtro passa-baixa
- No caso do LoG, este filtro é uma Gaussiana

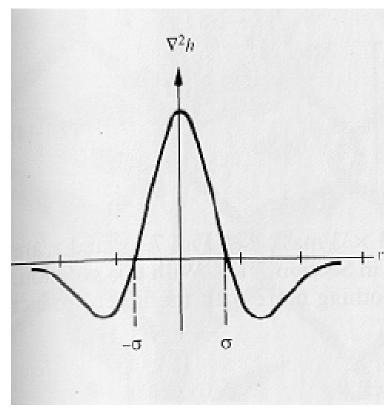
$$G(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

σ controla a suavização

- Pode ser mostrado que:

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

$$\nabla^2 G(x, y) = (\frac{r^2 - \sigma^2}{\sigma^4}) e^{-r^2/2\sigma^2}, \; (r^2 = x^2 + y^2)$$



LoG

- Aplicado o filtro LoG, deve-se encontrar o cruzamento por zero
  – Vizinhança 3 x 3 centrado em *p*. Um cruzamento por zero em *p* ocorre quando os sinais de pelo menos dois de seus pixels vizinhos opostos são diferentes! (esquerda/direita; cima/baixo e duas diagonais)
  – Pode-se ainda considerar um limiar entre a diferença absoluta dos vizinhos
- Quanto ao filtro Gaussiano:
  – 99,7% do volume sob uma superfície de uma gaussiana 2D situa-se entre +- 3σ ao redor da média.
  – O tamanho do filtro discreto LoG *n* x *n* deve ser tal que *n* seja o menor inteiro ímpar >= 6σ

Detectores de borda com 2ª Derivada – **2D**

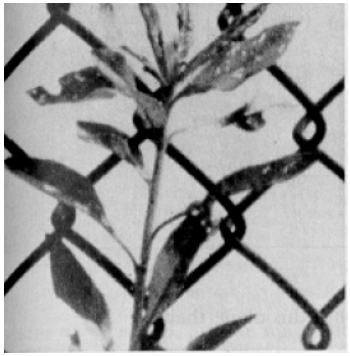
Exemplo Laplaciano of Gaussiano (LoG)

5 × 5 Laplacian of Gaussian mask

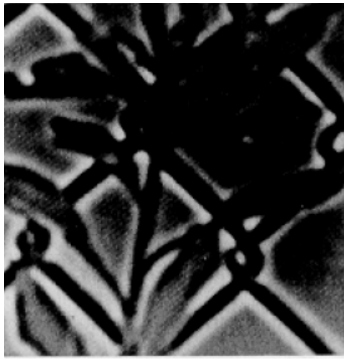
0	0	−1	0	0
0	−1	−2	−1	0
−1	−2	16	−2	−1
0	−1	−2	−1	0
0	0	−1	0	0

17 × 17 Laplacian of Gaussian mask

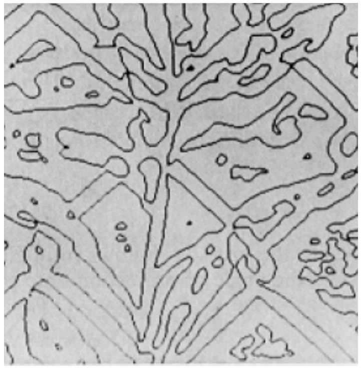
0	0	0	0	0	0	−1	−1	−1	−1	−1	0	0	0	0	0	0
0	0	0	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	0	0	0	0
0	0	−1	−1	−1	−2	−3	−3	−3	−3	−3	−2	−1	−1	−1	0	0
0	0	−1	−1	−2	−3	−3	−3	−3	−3	−3	−3	−2	−1	−1	0	0
0	−1	−1	−2	−3	−3	−3	−2	−3	−2	−3	−3	−3	−2	−1	−1	0
0	−1	−2	−3	−3	−3	0	2	4	2	0	−3	−3	−3	−2	−1	0
−1	−1	−3	−3	−3	0	4	10	12	10	4	0	−3	−3	−3	−1	−1
−1	−1	−3	−3	−2	2	10	18	21	18	10	2	−2	−3	−3	−1	−1
−1	−1	−3	−3	−3	4	12	21	24	21	12	4	−3	−3	−3	−1	−1
−1	−1	−3	−3	−2	2	10	18	21	18	10	2	−2	−3	−3	−1	−1
−1	−1	−3	−3	−3	0	4	10	12	10	4	0	−3	−3	−3	−1	−1
0	−1	−2	−3	−3	−3	0	2	4	2	0	−3	−3	−3	−2	−1	0
0	−1	−1	−2	−3	−3	−3	−2	−3	−2	−3	−3	−3	−2	−1	−1	0
0	0	−1	−1	−2	−3	−3	−3	−3	−3	−3	−3	−2	−1	−1	0	0
0	0	−1	−1	−1	−2	−3	−3	−3	−3	−3	−2	−1	−1	−1	0	0
0	0	0	0	−1	−1	−1	−1	−1	−1	−1	−1	−1	0	0	0	0
0	0	0	0	0	0	−1	−1	−1	−1	−1	0	0	0	0	0	0

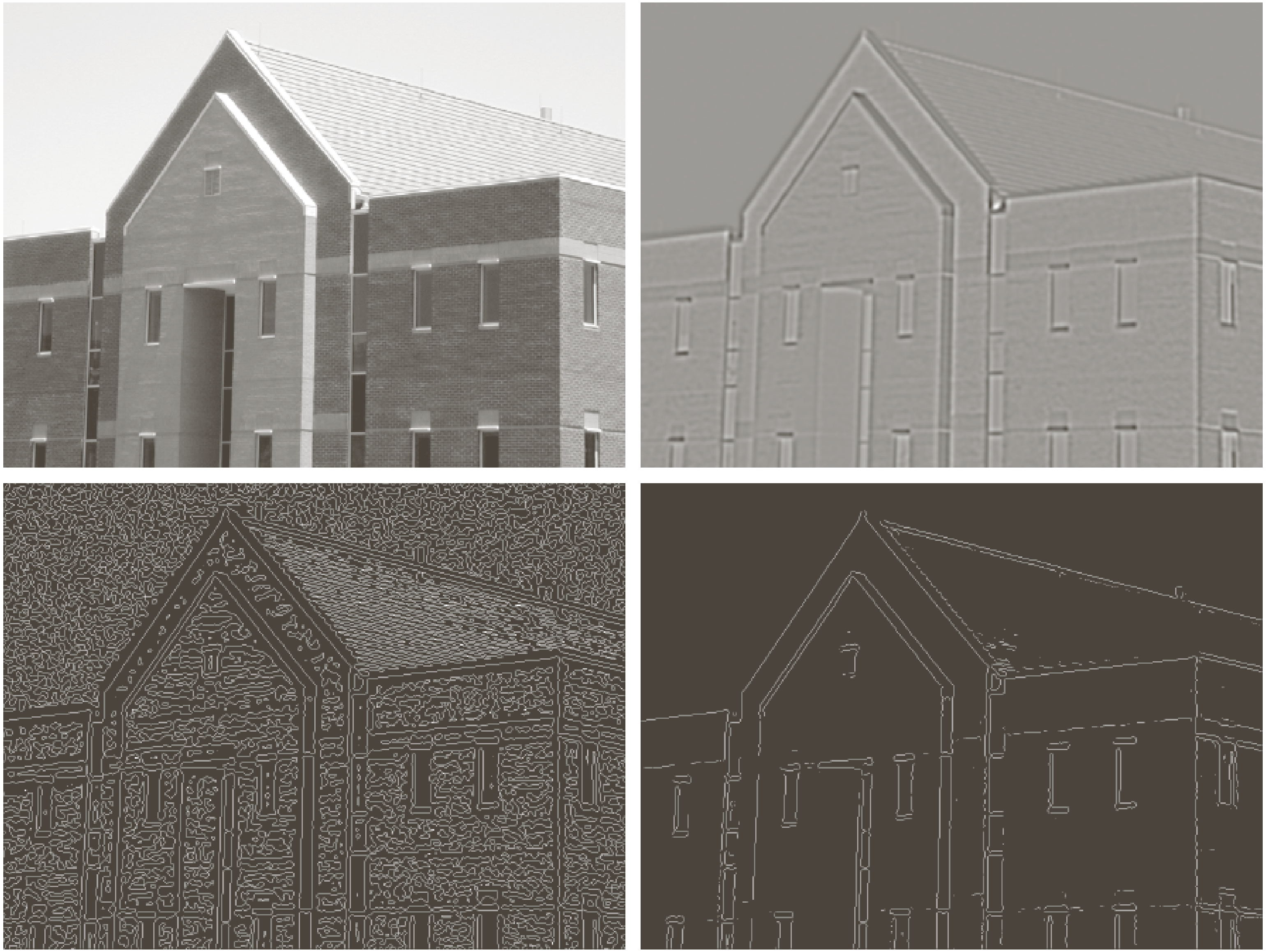


convolution result



zero-crossings





a b
c d

FIGURE 10.22 (a) Original image of size 834 × 1114 pixels, with intensity values scaled to the range [0, 1]. (b) Results of Steps 1 and 2 of the Marr-Hildreth algorithm using $\sigma = 4$ and $n = 25$. (c) Zero crossings of (b) using a threshold of 0 (note the closed-loop edges). (d) Zero crossings found using a threshold equal to 4% of the maximum value of the image in (b). Note the thin edges.

Detector de borda Canny

- "Taxonomicamente" este operador deveria estar junto aos operadores de primeira derivada, mas devido à sua "boniteza" 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: retorna um único ponto de borda (nro. de máximos locais deve ser mínimo)

Detector de borda 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
    - Análise baseada em bordas do tipo degrau (*step-edges*) corrompidas por ruído branco Gaussiano
    - (ruído branco : ruído com espectro de freqüência contínuo e uniforme sobre uma banda específica)
    - (ruído branco Gaussiano: ruído branco em que a distribuição dos valores de amplitude é Gaussiano)
- Critério única resposta
  - Aproximação numérica

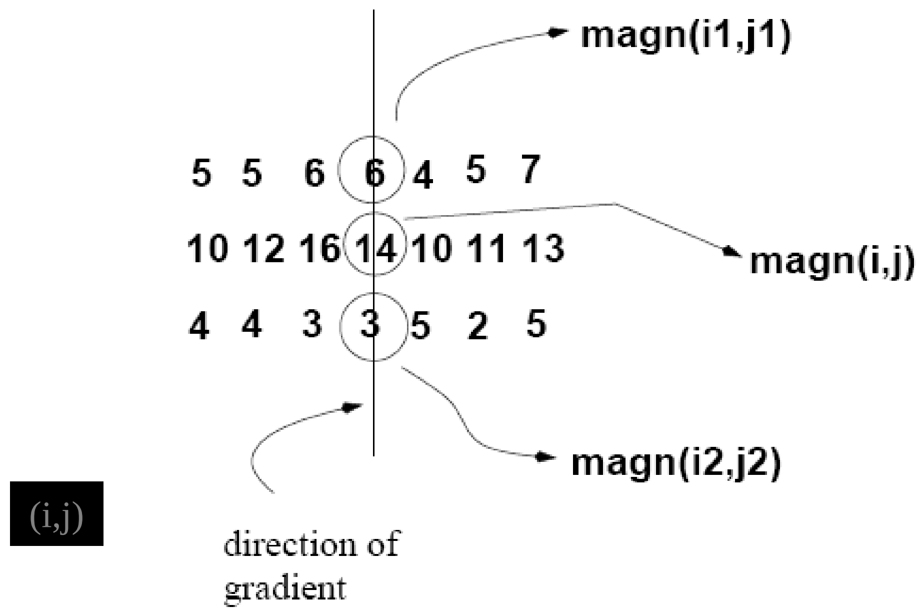
Canny - Etapas

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 qqr filtro de primeira derivada visto antes
   - $M(x,y)$ fatalmente conterá cristas largas em torno dos máximos locais. Para afinar estas cristas utiliza algoritmo de supressão dos não máximos, afinando $M(x,y)$
3. Limiarizar $M(x,y)$ e analisar conectividade

Canny

Non-maxima suppression

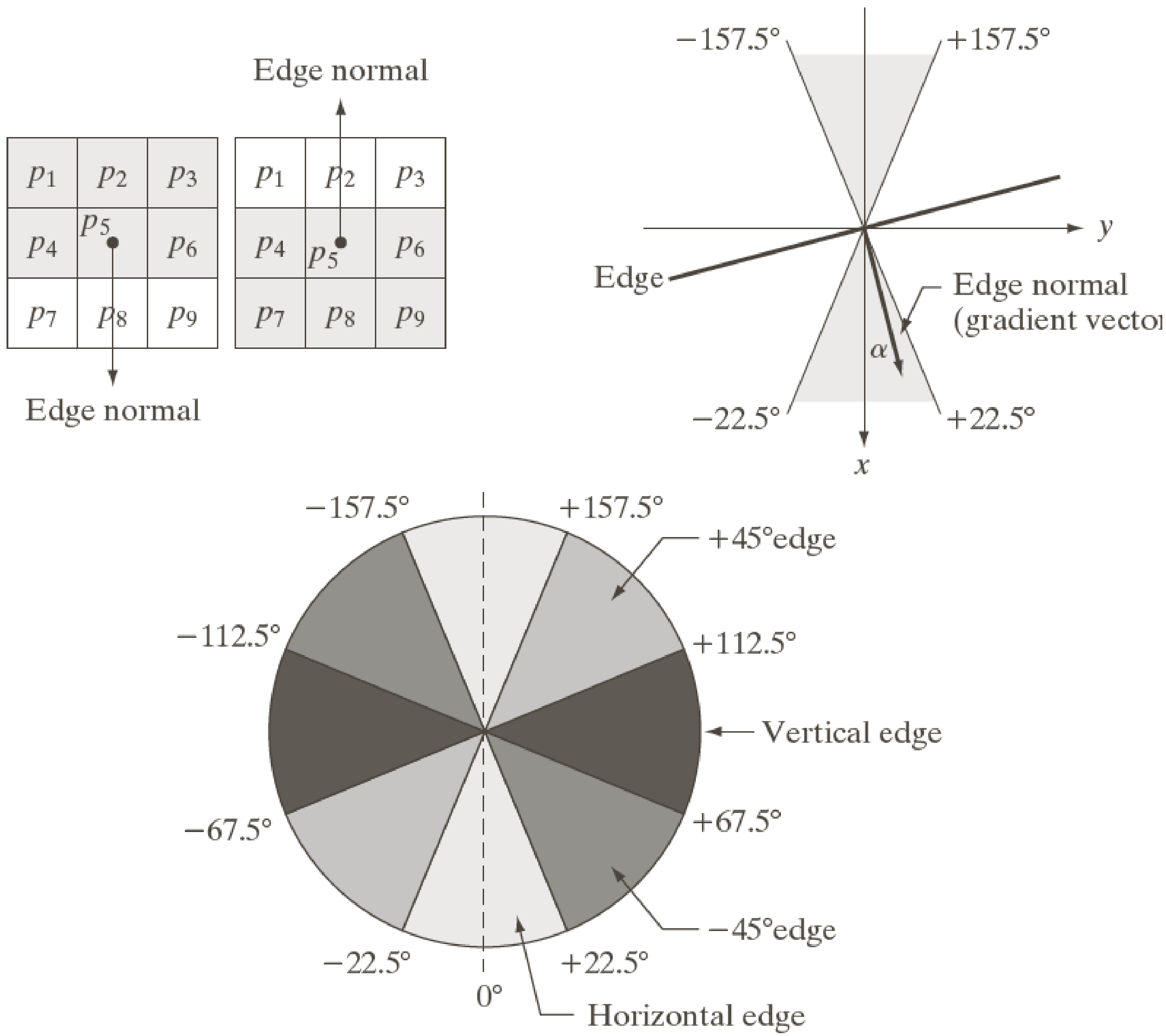
Supressão dos não máximos



Algorithm

For each pixel (x,y) do:

if $magn(i, j) < magn(i_1, j_1)$ or $magn(i, j) < magn(i_2, j_2)$
    then $I_N(i, j) = 0$
else $I_N(i, j) = magn(i, j)$



a b
c

FIGURE 10.24
(a) Two possible orientations of a horizontal edge (in gray) in a 3×3 neighborhood. (b) Range of values (in gray) of α, the direction angle of the *edge normal*, for a horizontal edge. (c) The angle ranges of the edge normals for the four types of edge directions in a 3×3 neighborhood. Each edge direction has two ranges, shown in corresponding shades of gray.

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 α (x,y)
   – Usar qqr filtro de primeira derivada visto antes
   – M(x,y) fatalmente conterá cristas largas em torno dos máximos locais. Para afinar estas cristas utiliza algoritmo de supressão dos não máximos, afinando M(x,y)
3. Limiarizar M(x,y) e analisar conectividade

Canny
Limiarização por histerese/*edge linking*

- Usar dois limiares
  - Um limiar baixo t_l (obtém I_1)
  - Um limiar alto t_h (obtém I_2 normalmente, $t_h = 2t_l$)

Algorithm

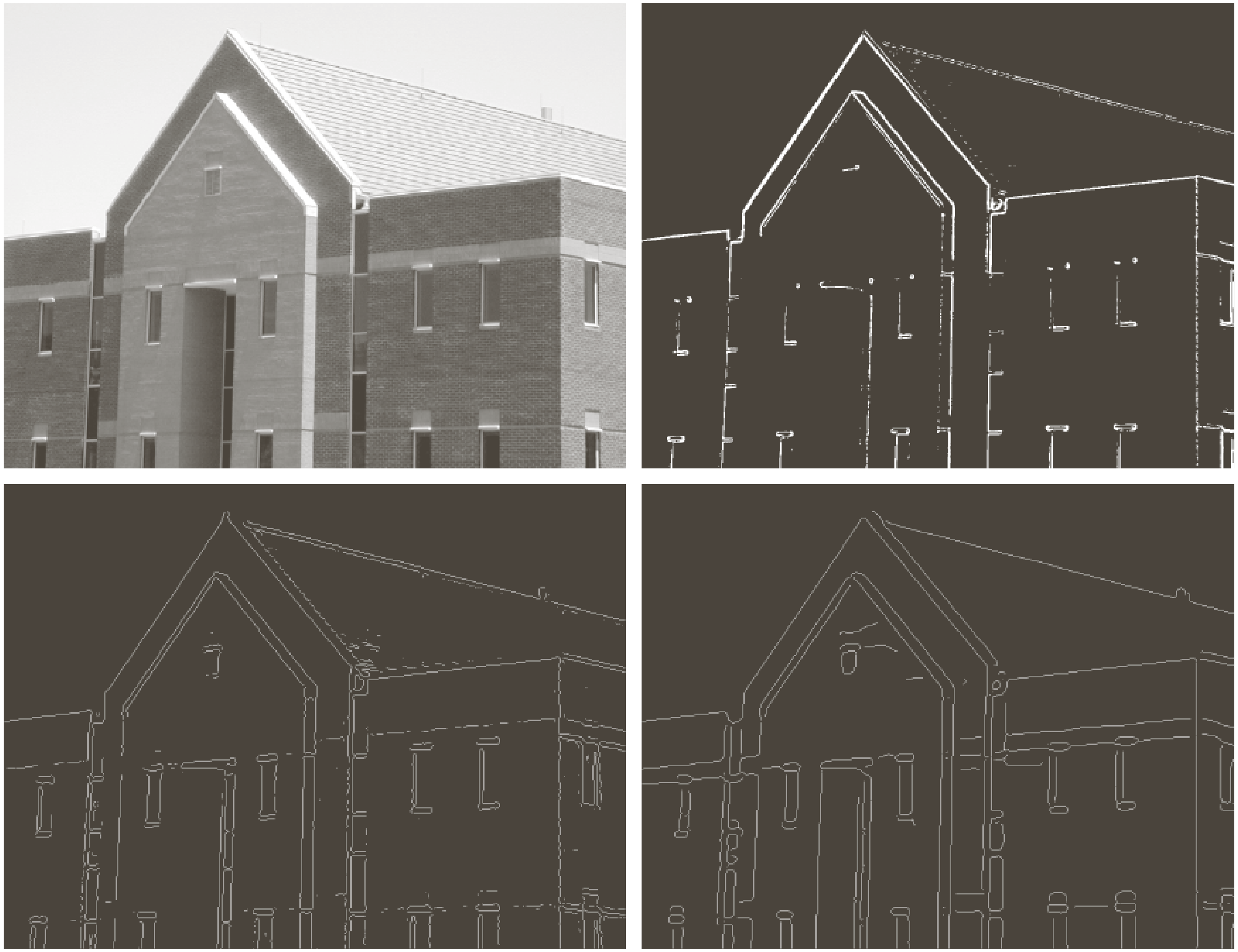
1. Produce two thresholded images $I_1(i, j)$ and $I_2(i, j)$.

(note: since $I_2(i, j)$ was formed with a high threshold, it will contain fewer false edges but there might be gaps in the contours)

2. Link the edges in $I_2(i, j)$ into contours

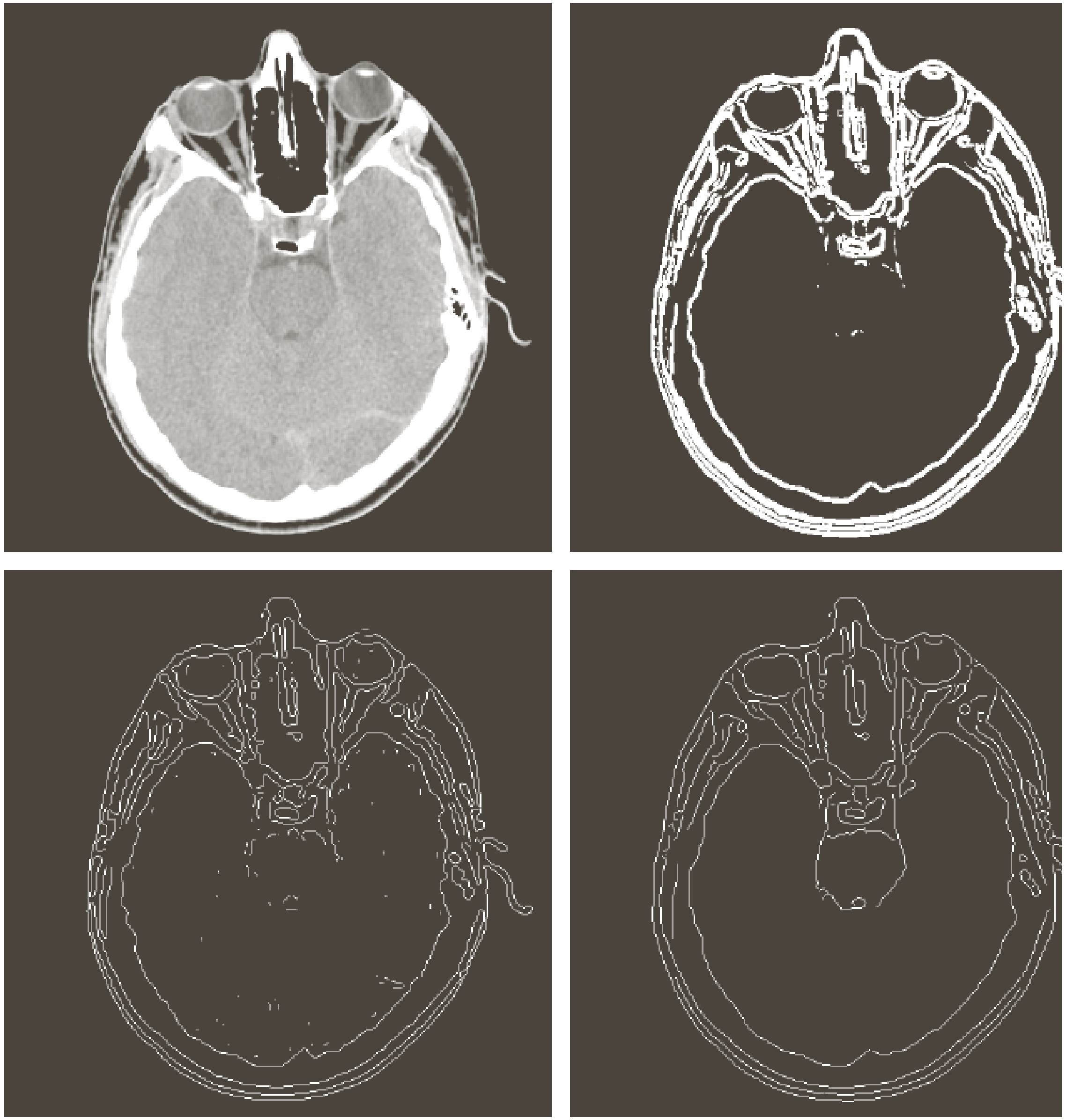
    2.1 Look in $I_1(i, j)$ when a gap is found.

    2.2 By examining the 8 neighbors in $I_1(i, j)$, gather edge points from $I_1(i, j)$ until the gap has been bridged to an edge in $I_2(i, j)$.



a	b
c	d

FIGURE 10.25 (a) Original image of size 834 × 1114 pixels, with intensity values scaled to the range [0, 1]. (b) Thresholded gradient of smoothed image. (c) Image obtained using the Marr-Hildreth algorithm. (d) Image obtained using the Canny algorithm. Note the significant improvement of the Canny image compared to the other two.

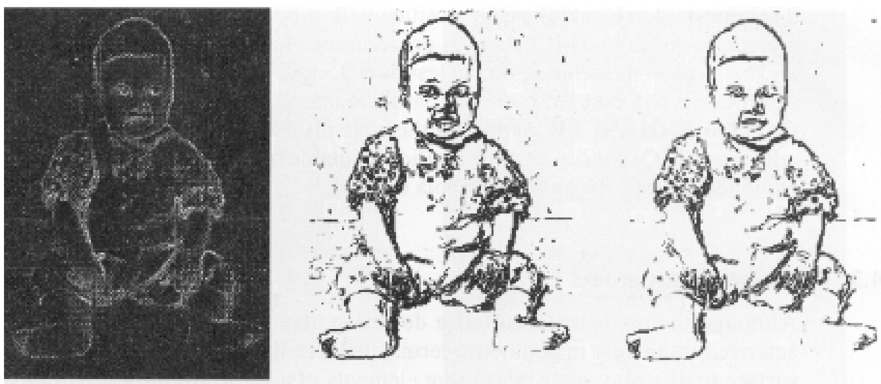


a b
c d

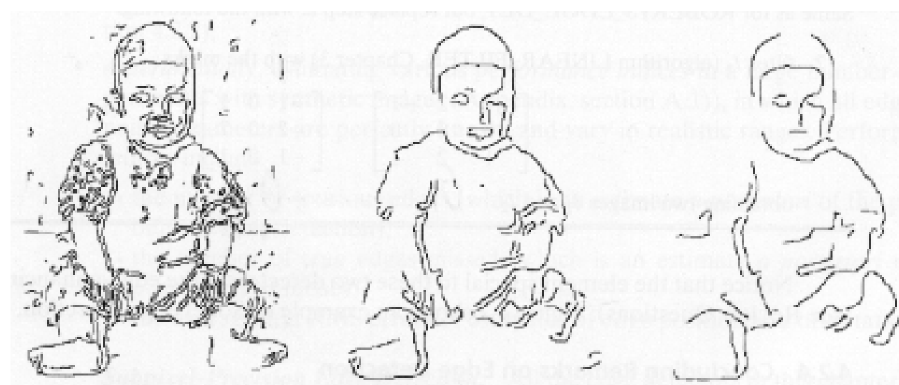
FIGURE 10.26
(a) Original head CT image of size 512×512 pixels, with intensity values scaled to the range $[0, 1]$. (b) Thresholded gradient of smoothed image. (c) Image obtained using the Marr-Hildreth algorithm. (d) Image obtained using the Canny algorithm. (Original image courtesy of Dr. David R. Pickens, Vanderbilt University.)

Exemplo – qual o efeito de σ?

(left:Sobel, middle: thresh=35, right: thersh=50)

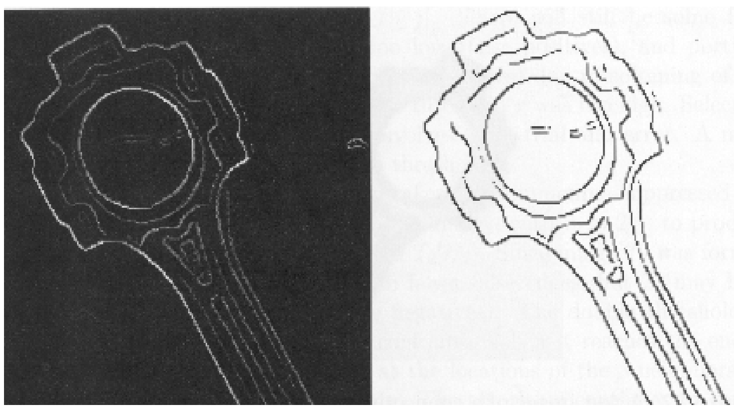


(Canny - left:σ=1, middle: σ=2, right: σ=3)

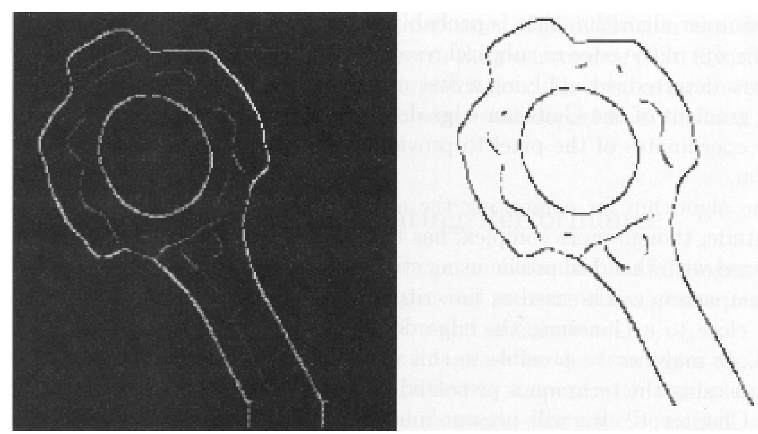


Exemplo – qual o efeito de σ?

(Canny - 7x7 Gaussian, more details)



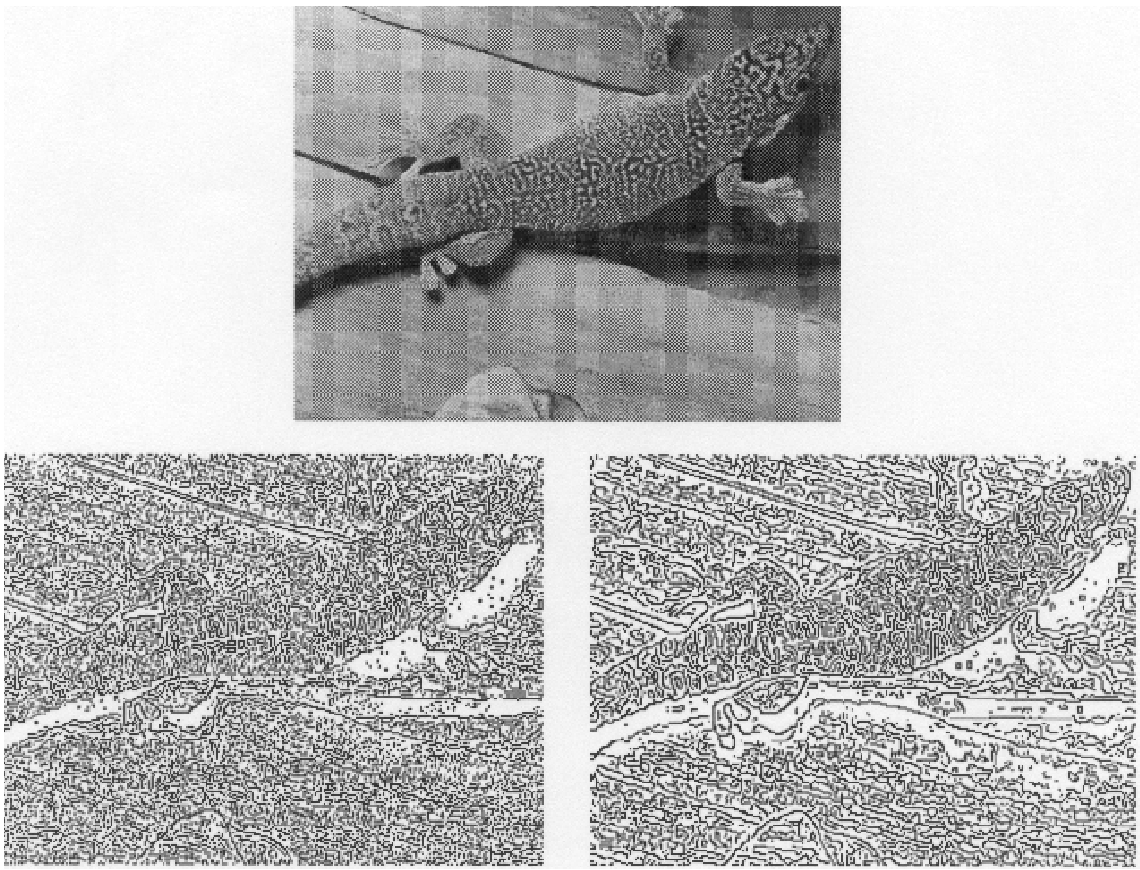
(Canny - 31x31 Gaussian, less details)



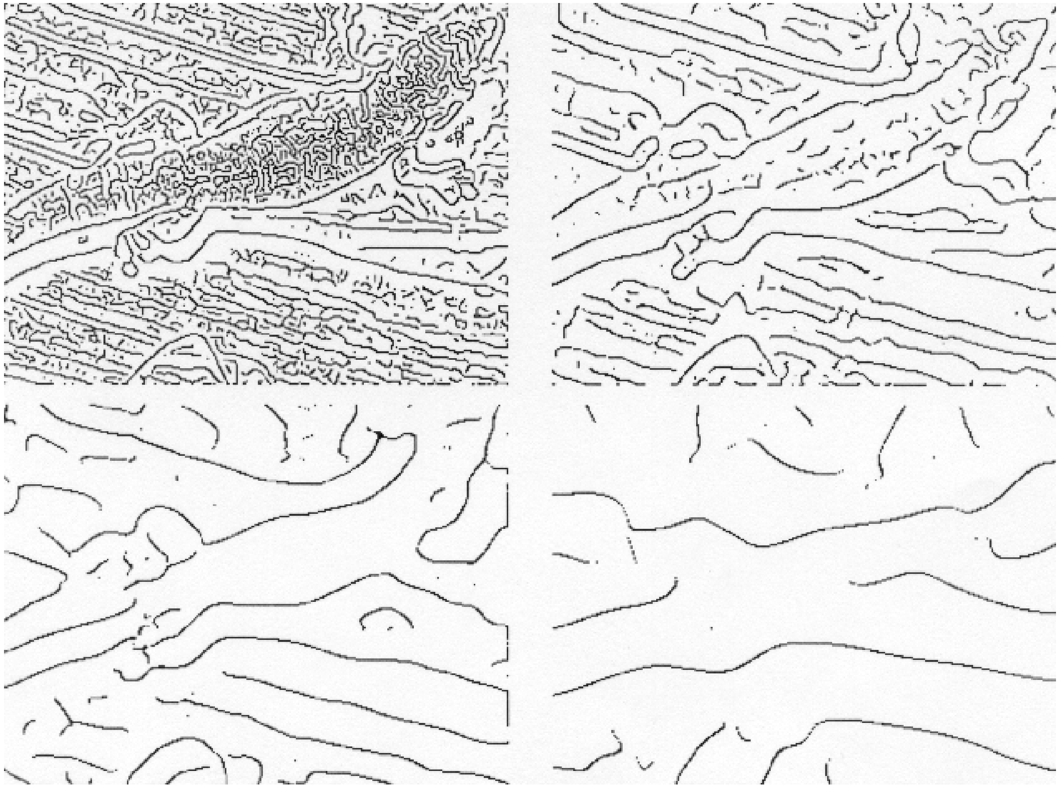
Processamento multi-escala (*scale space*)

- Um problema prático com qqr detector de borda é a questão da escolha da escala de suavização (o valor de σ na Gaussiana)
- Em muitas aplicações é desejável processar imagens em várias escalas
  - talvez combinar os resultados
- Determinamos quais bordas são mais significativas em termos da faixa de escalas sobre as quais elas serão evidentes

Processamento multi-escala



Processamento multi-escala



(Canny edges at multiple scales of smoothing, σ=0.5, 1, 2, 4, 8, 16)

Bibliografia

- Gonzalez & Woods, Processamento Digital de Imagens, 3ª. Edição (capts. 3 e 10)
- Sonka, Image Processing, Analysis and Machine Vision