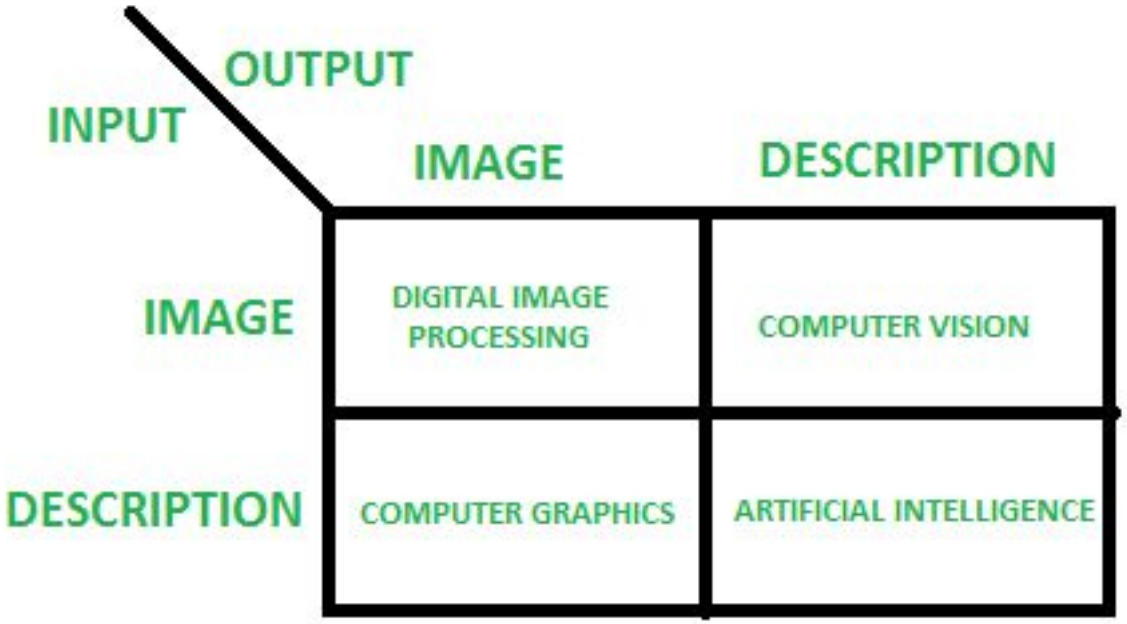
Computação Gráfica
Processamento de Imagem

www.dca.ufrn.br/~lmarcos/courses/compgraf

Processamento de Imagens

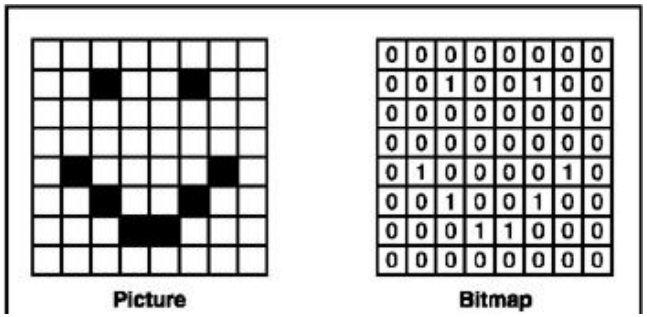
– Operações com imagens
– Processamento de pontos
– Filtragem (processamento de imagens)



Operações com imagens

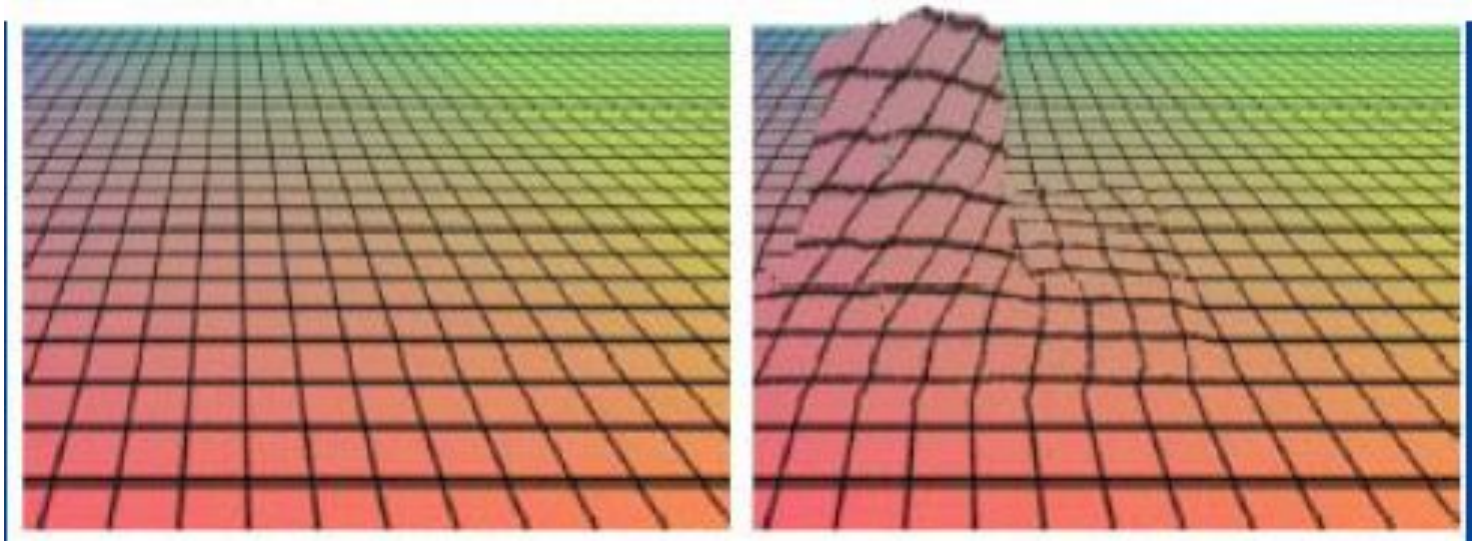
- Warping
- Morphing
- Composição de imagens

- Estrutura da imagem: matriz (2D) de pixels, uma cópia do frame-buffer em memória (R,G,B,α)



Warping (“entortamento”)

Move pixels na imagem (segundo alguma transformação)



Warping

- Truque da sala dos espelhos (lentes)

Morphing

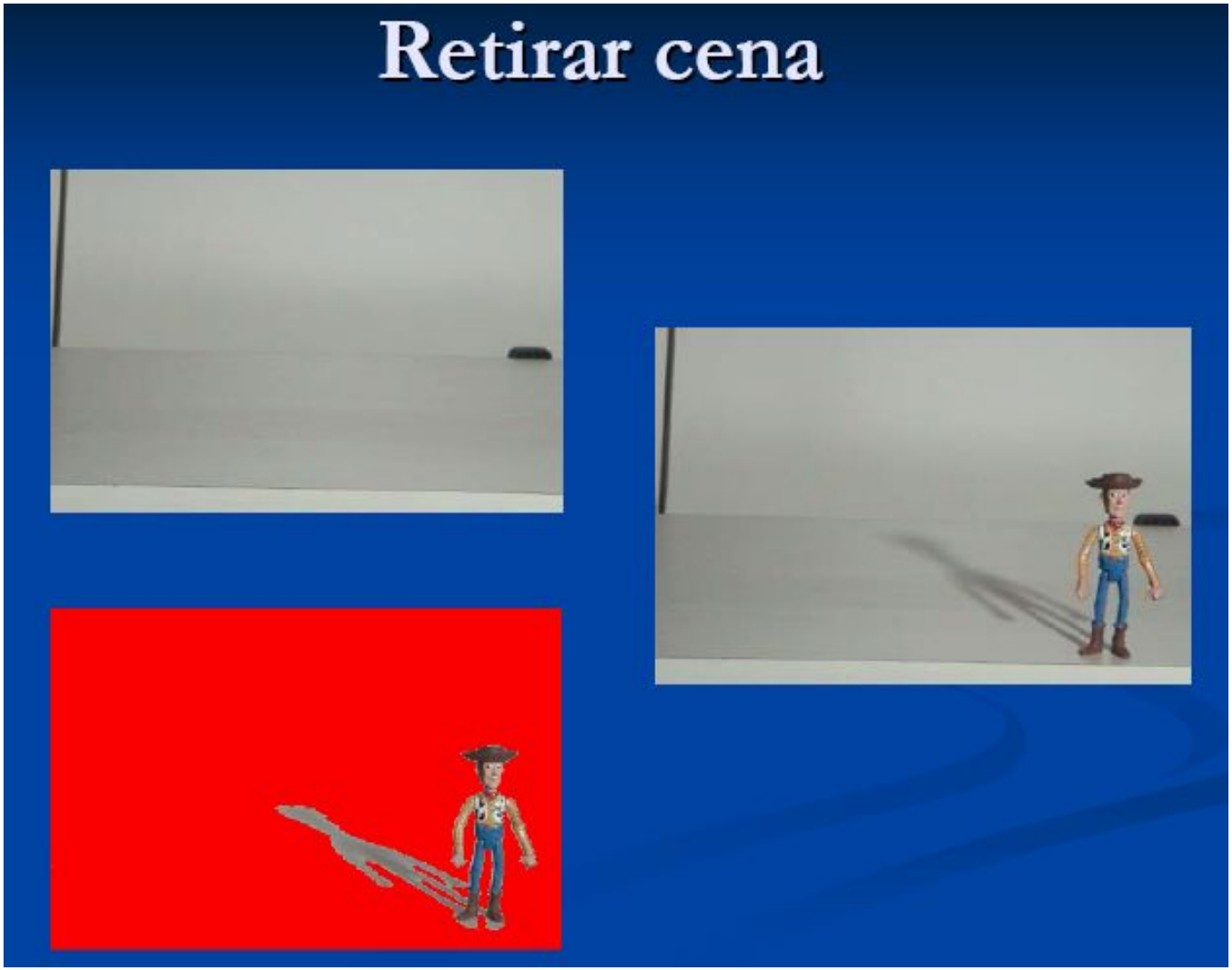
- Transformar uma imagem em outra, de forma suave


MakeAGIF.com

Composição de imagens

- O canal α adicional é usado para isso
  - Opacidade: 0 é transparente e 1 opaco
- Matematicamente:
- $I_r(x,y) = (1- \alpha)I_1(x,y) + \alpha\, I_2(x,y)$
  - $\alpha = 0$ ou 1: uma ou outra vence
  - $0 < \alpha < 1$: cor resultante é composição das duas
- Efeitos especiais (inserção de caracteres ou objetos artificiais em filmes), retoques em fotos etc.

Composição



Composição

- Soma ponderada das imagens

Composição

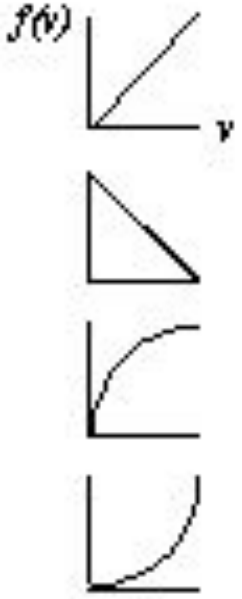
Inserindo a sombra

Processamento de Imagens

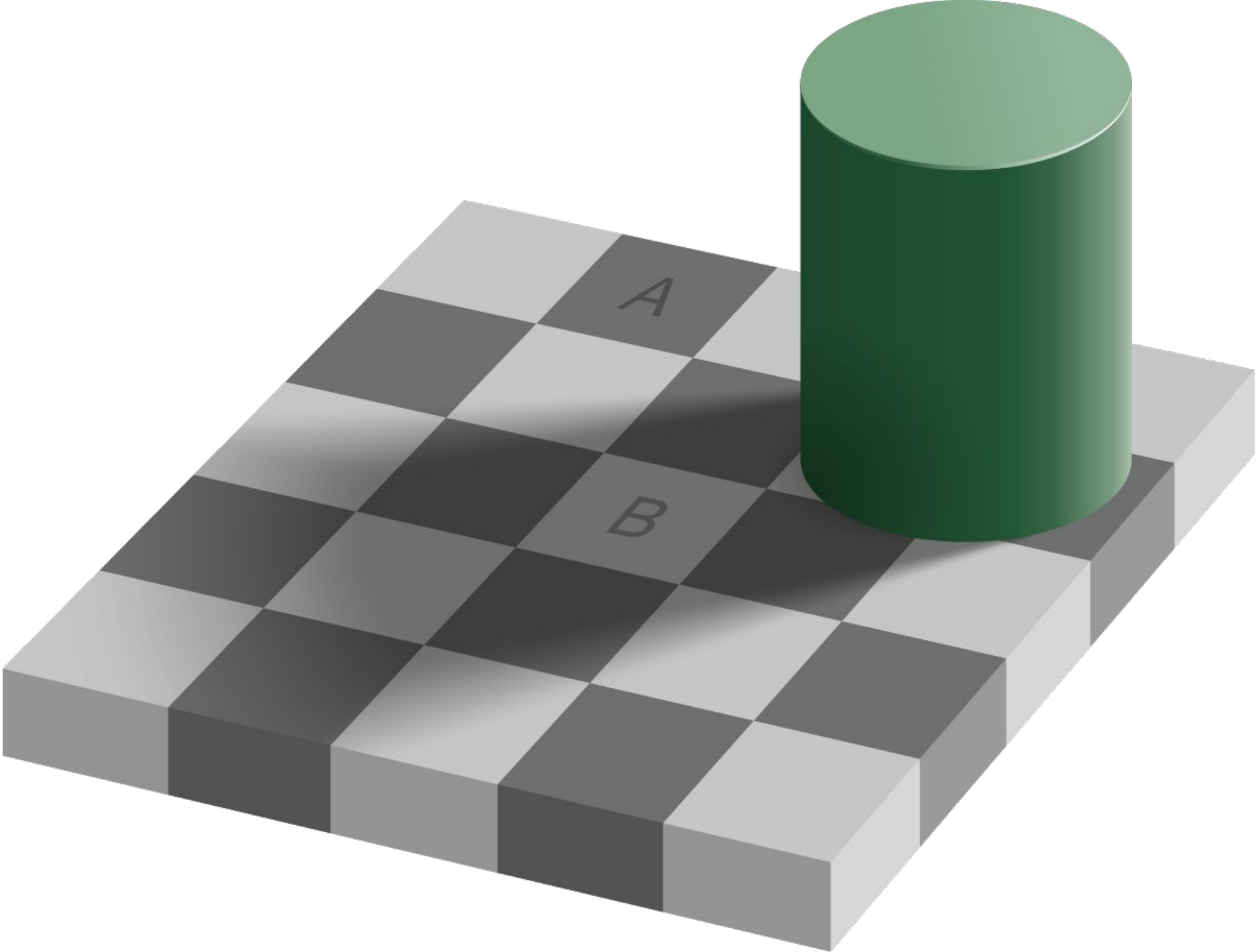
- PI: generalização 2D, discreta, de processamento de sinais (Engenharia Elétrica, Áudio, Sismologia, etc).
  - Referimo-nos a uma imagem como um sinal bi-dimensional
- Processamento de Ponto
  - Modifica valor em função do valor de entrada
- Filtragem de Imagens
  - Modifica o valor do ponto em função dos vizinhos

Processamento de ponto

- Entrada: $I(x,y) \in [0,1]$ Saída: I'(x,y)=$f(I(x,y))$
- Transforma cada pixel separadamente
- Considerando v no intervalo [0,1]:
- $f(v) = v$ (sem mudança, Idt)
- $f(v) = 1 - v$ (nega a imagem)
- $f(v) = v^p$ (p<1) Aumenta brilho
- $f(v) = v^p$ (p>1) Diminui brilho



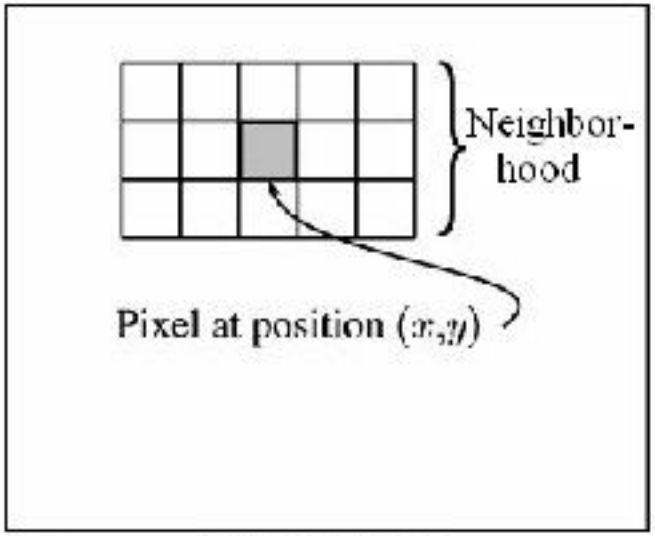
Thresholding (binarização)


A
B

Chapter 5: Neighborhood Processing

Point processing: applies a function to each pixel

Neighborhood processing: applies a function to a neighborhood of each pixel



Original image

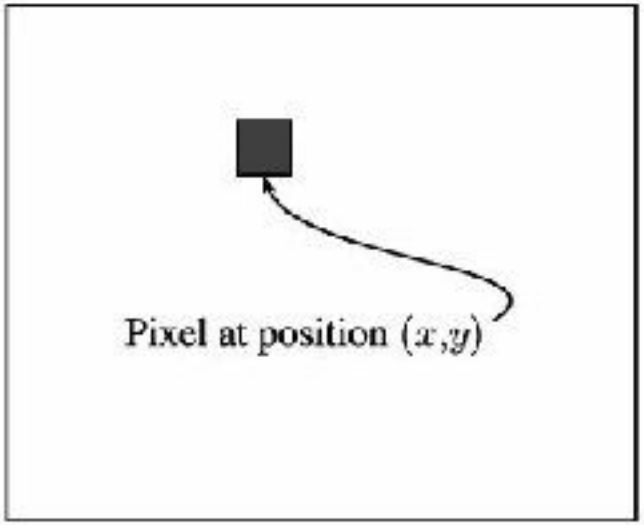


Image after processing

Filtragem

- Recebe como entrada uma imagem
- Realiza alguma operação
- Resultado é uma imagem

Aplicações de filtragem

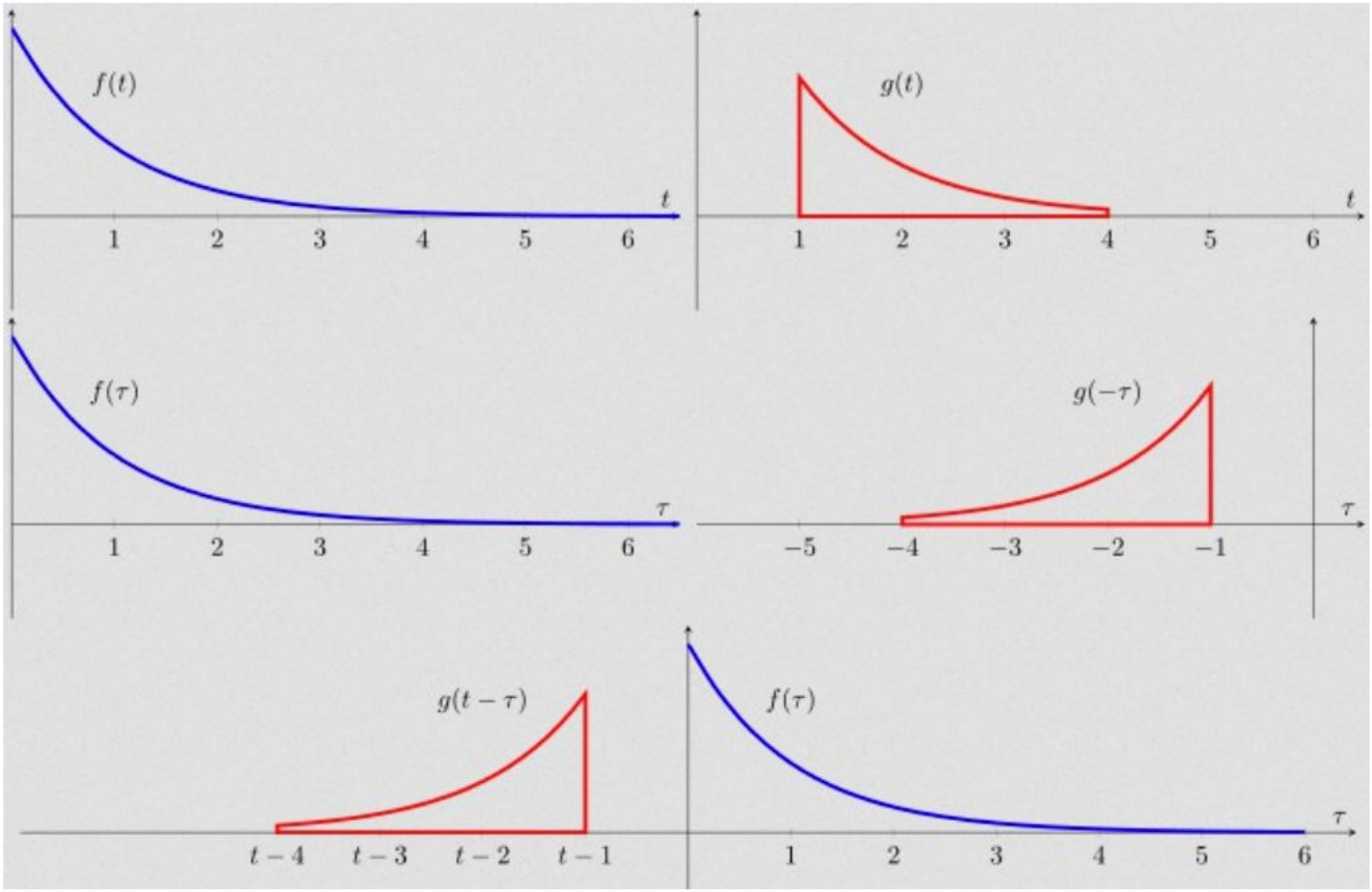
- Restauração
- Melhoria de Imagens
- Extração de features (características)
- Atenuação
- Compressão de imagens
- Pré-processamento para segmentação

Sinais e Filtragem

- Áudio gravado é um sinal 1D: amplitude(*t*)
- Imagem é um sinal 2D: *int*(*x*,*y*) ou *cor*(*x*,*y*)
- Sinais podem ser contínuos (analógicos) ou discretos (digitais)
- Tratamos imagens (raster) que são sinais discretos no espaço (*x*,*y*), em intensidade (valor quantizado) e no tempo (*t*)

Convolução

$$(f * g)(t) \overset{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau)\, g(t-\tau)\, d\tau$$



Convolução

Early Processing - Convolution

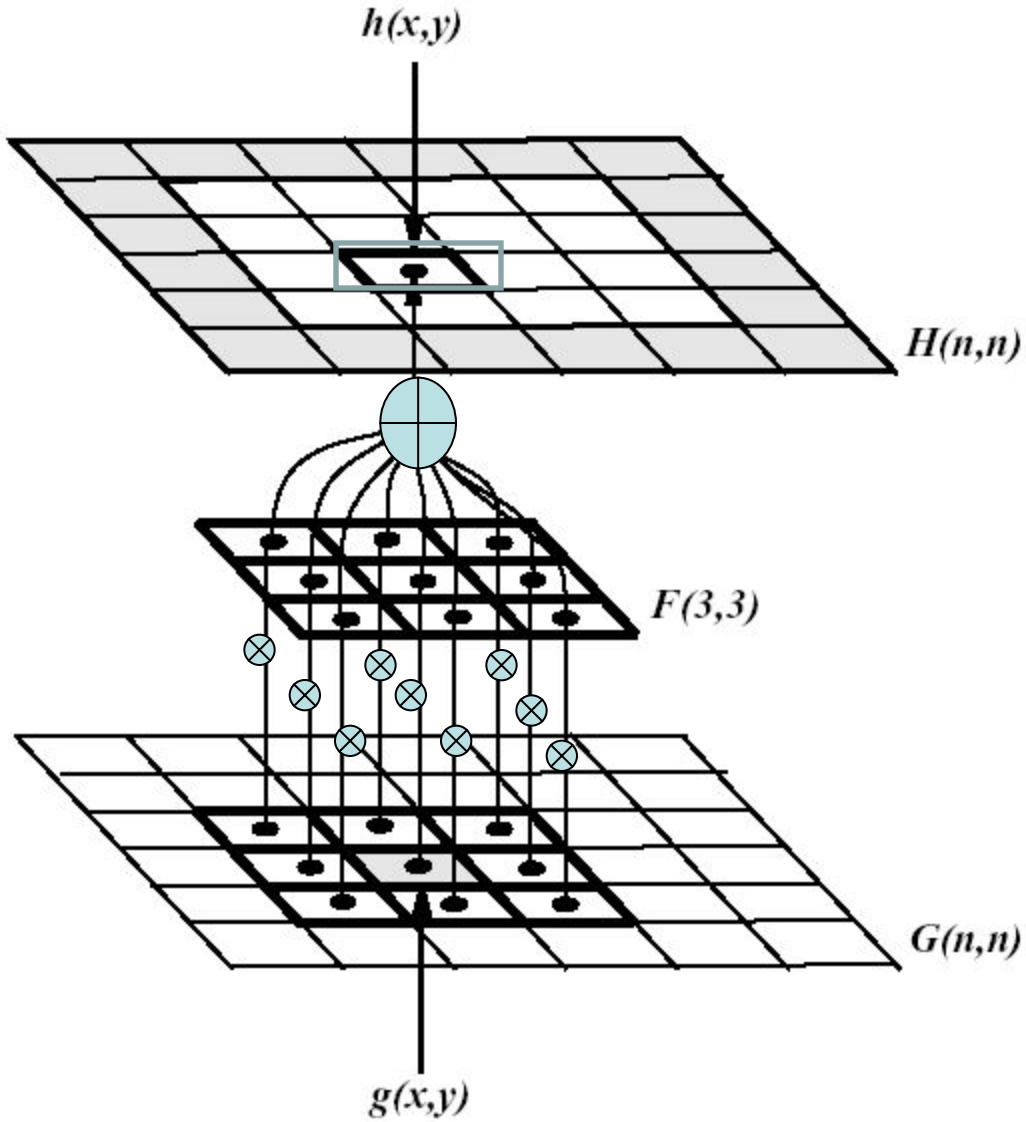
The convolution of two functions $f(x)$ and $g(x)$:

$$f(x,y) * g(x,y) = h(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(u,v) g(x-u, y-v) du dv$$

or

$$f(x,y) * g(x,y) = h(x,y) = \sum_{-\infty}^{\infty} \sum_{-\infty}^{\infty} f(u,v) g(x-u, y-v)$$

Convolução



Linear, Shift-invariant Filters

$$b[x,y] = \sum_{u=-1}^{+1} \sum_{v=-1}^{+1} h[u,v]\,a[x-u, y-v]$$

$a[x]$ = input signal

$b[x]$ = output signal

$h[x]$ = 3x3 filter

x takes on only integer values

$\frac{1}{4}$

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

h

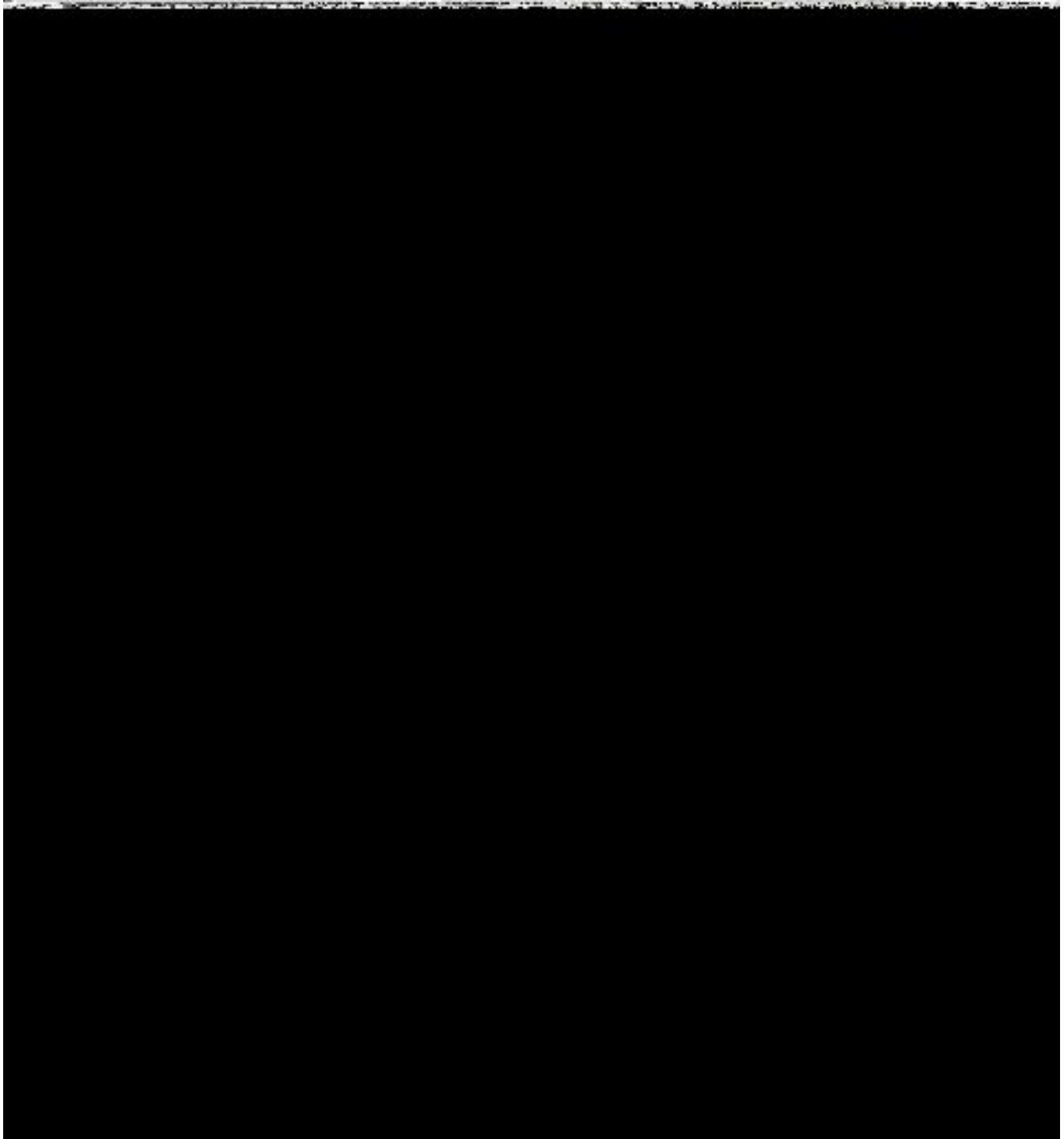
(0,0) at center

0	0	0	0	0	25	25	25	25	25
0	0	0	0	0	25	25	25	25	25
0	0	0	0	0	25	25	25	25	25
0	0	0	0	0	25	25	25	25	25
0	0	0	0	0	25	25	25	25	25
0	0	0	0	0	25	25	25	25	25
0	0	0	0	0	25	25	25	25	25
0	0	0	0	0	25	25	25	25	25
0	0	0	0	0	25	25	25	25	25
0	0	0	0	0	25	25	25	25	25

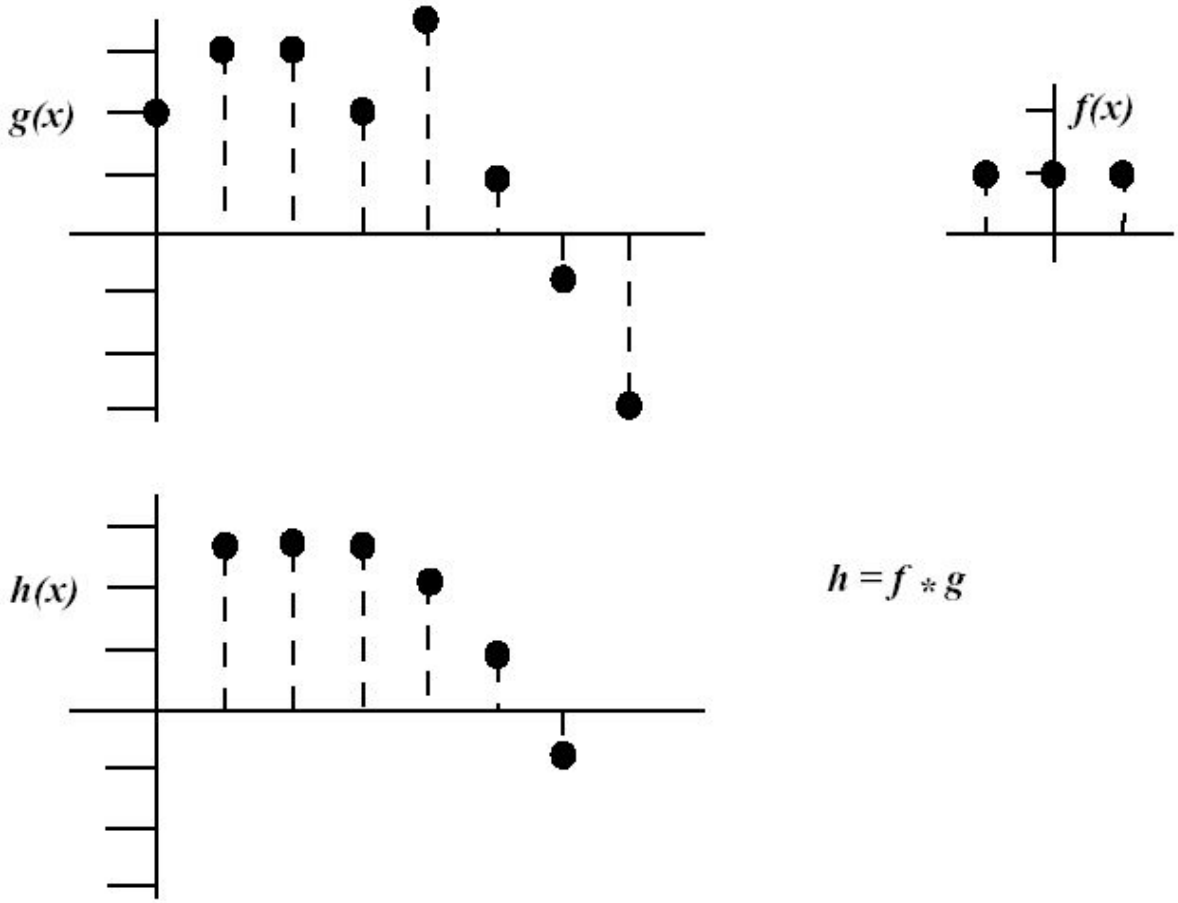
a

0	0	0	0	25	25	0	0	0	0
0	0	0	0	25	25	0	0	0	0
0	0	0	0	25	25	0	0	0	0
0	0	0	0	25	25	0	0	0	0
0	0	0	0	25	25	0	0	0	0
0	0	0	0	25	25	0	0	0	0
0	0	0	0	25	25	0	0	0	0
0	0	0	0	25	25	0	0	0	0
0	0	0	0	25	25	0	0	0	0
0	0	0	0	25	25	0	0	0	0

b



Filtro passa-baixa (atenuação)



Blurring Filters

A simple blurring effect can be achieved with a 3x3 filter centered around a pixel,

written explicitly:

$$b[x,y] = (a[x-1,y-1] + a[x,y-1] + a[x+1,y-1] + a[x-1,y] + a[x,y] + a[x+1,y] + a[x-1,y+1] + a[x,y+1] + a[x+1,y+1]) / 9$$

or as coefficient matrix:

$$\frac{1}{9}\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

More blurring is achieved with a wider *n* x *n* filter:

$$\frac{1}{n*n}\begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix}$$

Image Filtering: Blurring

original, 64x64 pixels

3x3 blur

5x5 blur

Detecção de arestas

- Gradiente (2D):

$$\nabla g = \frac{dg}{dx}\hat{x} + \frac{dg}{dy}\hat{y}$$

$$\frac{dg(x,y)}{dx} \approx \frac{g(x+1,y) - g(x-1,y)}{2}$$

$$\Rightarrow f_x = \left[-\frac{1}{2},\ 0,\ \frac{1}{2}\right]_{1x3}$$

Detecção de arestas

$$\frac{dg(x,y)}{dy} \approx \frac{g(x,y+1) - g(x,y-1)}{2}$$

$$\Rightarrow f_y = \begin{bmatrix} -\frac{1}{2} \\ 0 \\ \frac{1}{2} \end{bmatrix}_{3x1}$$

Detecção de arestas

- Magnitude do gradiente

$$|\nabla g| = \left[\left(\frac{dg}{dx}\right)^2 + \left(\frac{dg}{dy}\right)^2\right]^{\frac{1}{2}}$$

- Direção de mudança da intensidade

$$\phi = tan^{-1}\left(\frac{dg/dy}{dg/dx}\right)$$

Edge Filter

To find edges, use approximation to the magnitude of the gradient of the image.

Gradient and its magnitude:

$$\nabla a = \left(\frac{\partial a}{\partial x} \quad \frac{\partial a}{\partial y} \right), \qquad \nabla a = \sqrt{\left(\frac{\partial a}{\partial x} \right)^2 + \left(\frac{\partial a}{\partial y} \right)^2}$$

Sobel edge filter uses these weights:

$$\frac{\partial}{\partial x} \Rightarrow \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, \qquad \frac{\partial}{\partial y} \Rightarrow \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

This is a nonlinear filter because of the sqrt and square operations.

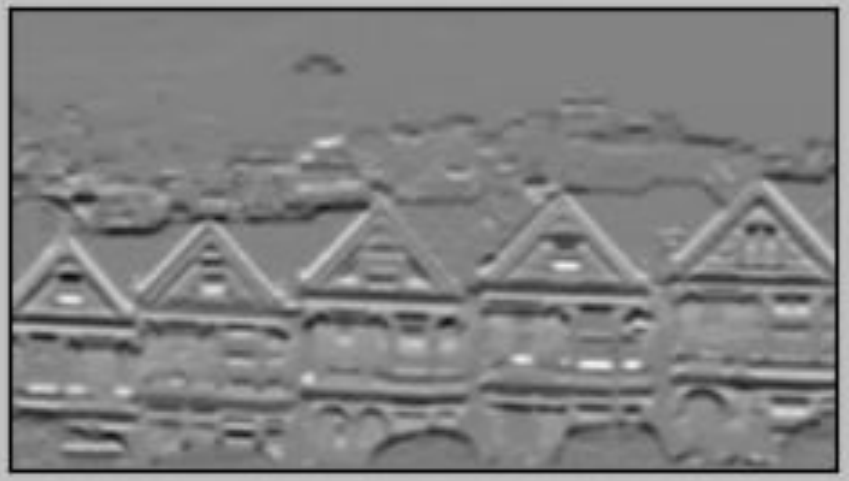
Original

$$\frac{\partial}{\partial x} \Rightarrow \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix},$$

$$\frac{\partial}{\partial y} \Rightarrow \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

Sobel X

Sobel Y

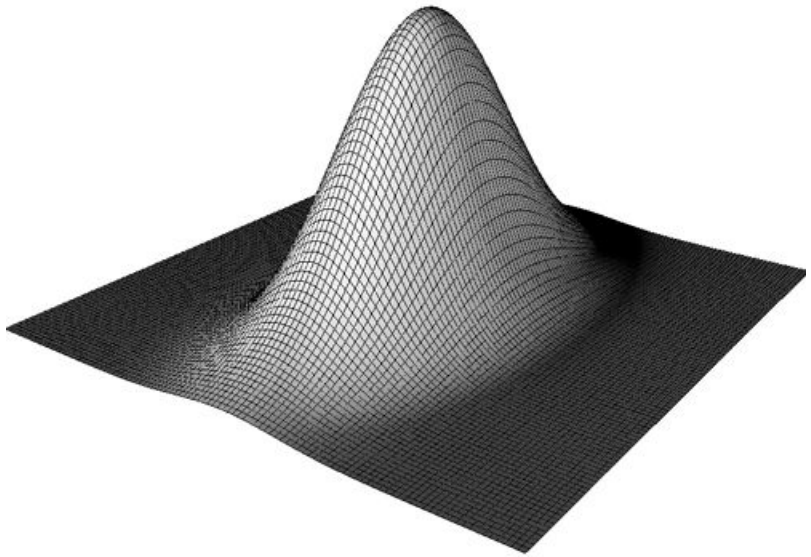


Outros detetores de bordas

operator	∇_1	∇_2
Roberts	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Prewit	$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$
Sobel	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$

Suavizando

- Filtro média
- Filtro Gaussiano (operadores gaussianos)
  - Construir a máscara a partir da função Gaussiana



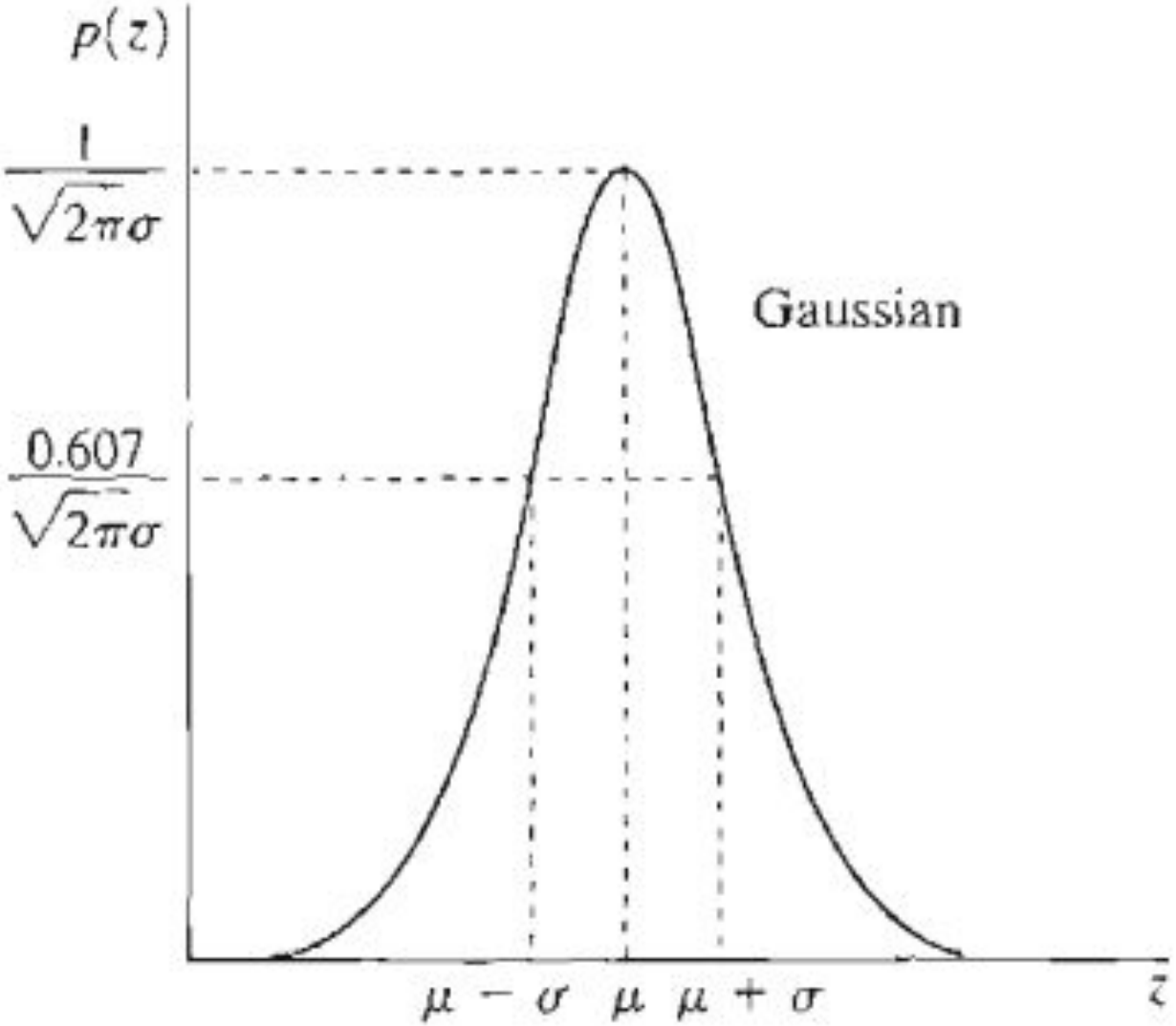
Operador Gaussiano

$$g_\sigma(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2}$$

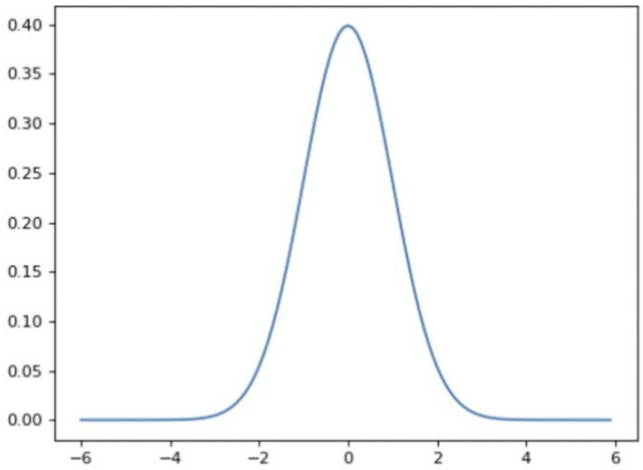
$$g'_\sigma(x) = \frac{-x}{\sqrt{2\pi}\sigma^3} e^{-x^2/2\sigma^2}$$

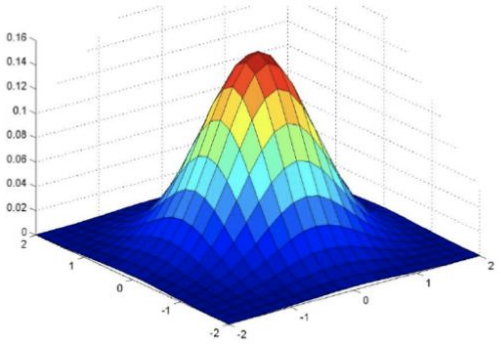
$$g''_\sigma(x) = \frac{1}{\sqrt{2\pi}} \left[\frac{x^2}{\sigma^5} - \frac{1}{\sigma^3} \right] e^{-x^2/2\sigma^2}$$

Significado das variáveis



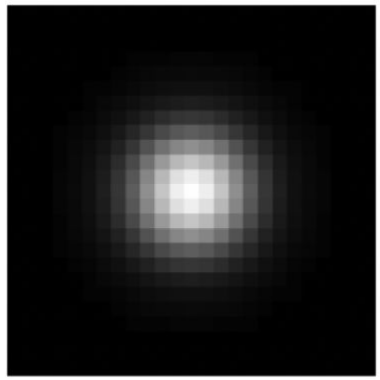
Filtro Gaussiano





Gaussian kernel

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

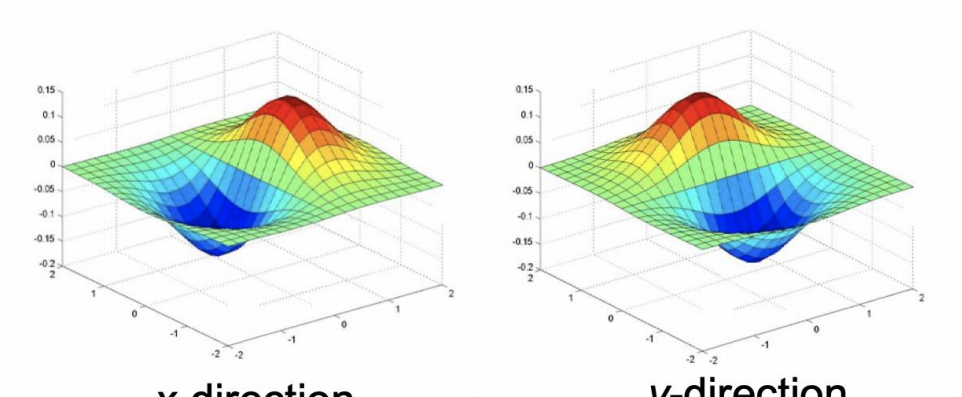


Máscara 3x3

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

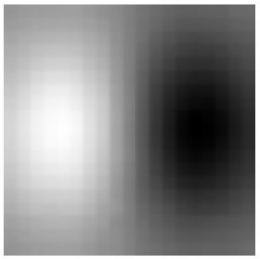
Efeito: suavização

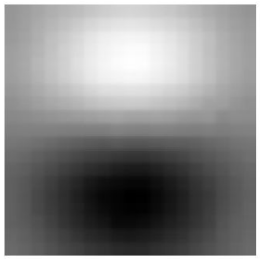
Gradiente do gaussiano



x-direction

y-direction



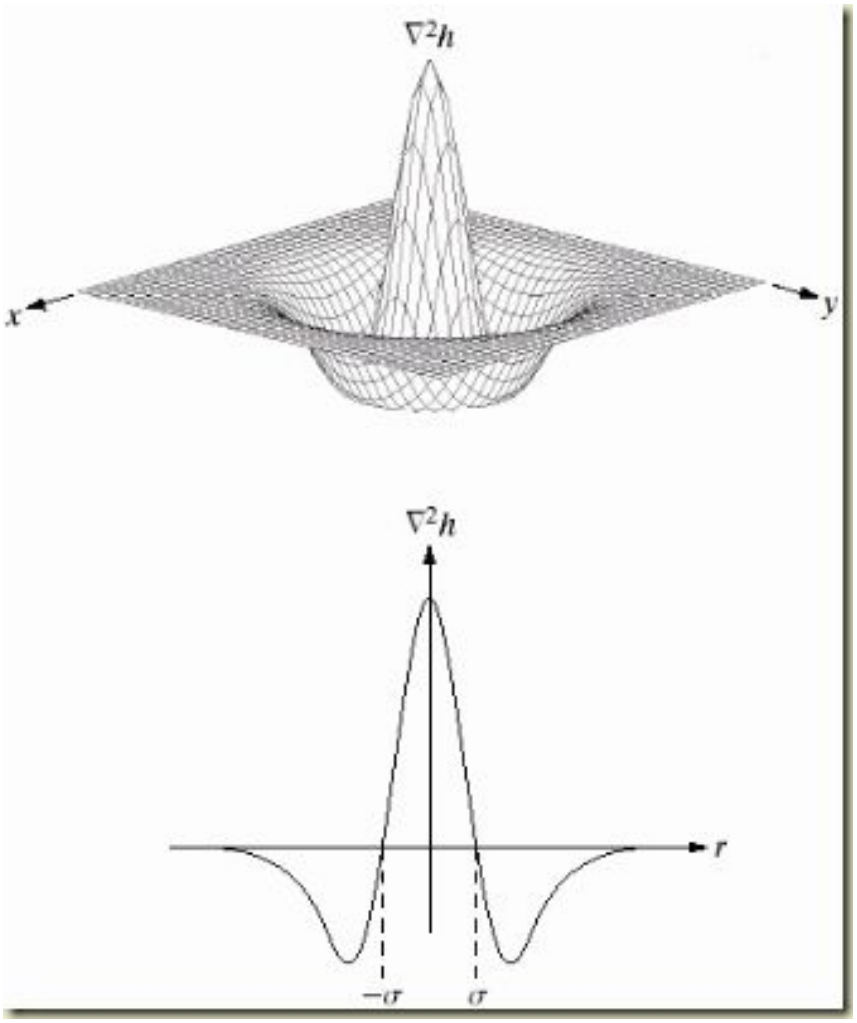


Efeito: detecção de bordas

Máscara 3x3

$$\frac{\partial}{\partial x} \Rightarrow \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, \qquad \frac{\partial}{\partial y} \Rightarrow \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

Laplaciano do gaussiano



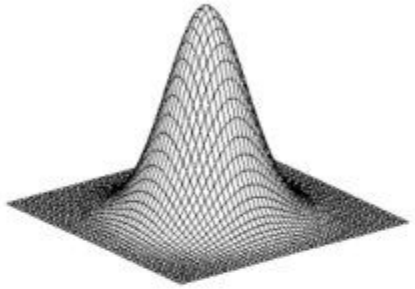
Máscara 3x3

$$\nabla^2 g = \frac{d^2 g}{dx^2} + \frac{d^2 g}{dy^2} \Rightarrow f = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Efeito: detecção de texturas

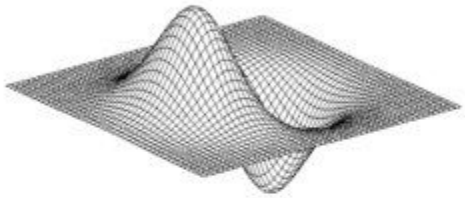
Filtros 2d para detecção de bordas

2D edge detection filters



Gaussian

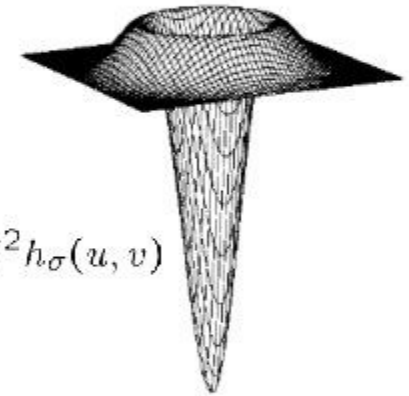
$$h_\sigma(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$



derivative of Gaussian

$$\frac{\partial}{\partial x} h_\sigma(u,v)$$

Laplacian of Gaussian



$$\nabla^2 h_\sigma(u,v)$$

∇^2 is the **Laplacian** operator:

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

Lena original

Gaussiano

Gradiente do Gaussiano

Laplaciano do Gaussiano

Original

Laplacian

Sobel X

Sobel Y