
# VISÃO COMPUTACIONAL

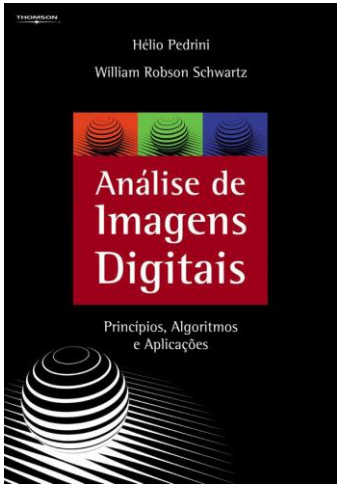
## DETECÇÃO DE DESCONTINUIDADES

Prof. Msc. Giovanni Lucca França da Silva
E-mail: giovanni-lucca@live.com

# SOBRE A DISCIPLINA

- Bibliografia principal:
  - GONZALEZ, Rafael C.; WOODS, Richard C. **Processamento digital de imagens**. Pearson, 2011.


- Bibliografia complementar:
  - PEDRINI, Hélio; SCHWARTZ, William Robson. **Análise de imagens digitais: princípios, algoritmos e aplicações**. Thomson Learning, 2008.



# NA AULA PASSADA...

- Morfologia Matemática.

# ROTEIRO

- Detecção de descontinuidades.
  - Detecção de pontos.
  - Detecção de linhas.
  - Detecção de bordas.

# DETECÇÃO DE DESCONTINUIDADES

- O que significa descontinuidade?

- Quais vantagens a detecção de descontinuidades possibilita para a análise de imagens?

# DETECÇÃO DE DESCONTINUIDADES

- O que significa descontinuidade?
  - Interrupção de continuidade.
  - Mudança de um comportamento.

- Quais vantagens a detecção de descontinuidades possibilita para a análise de imagens?
  - Possuem a maioria da informação semântica presente em uma imagem.
  - Redução dos dados.

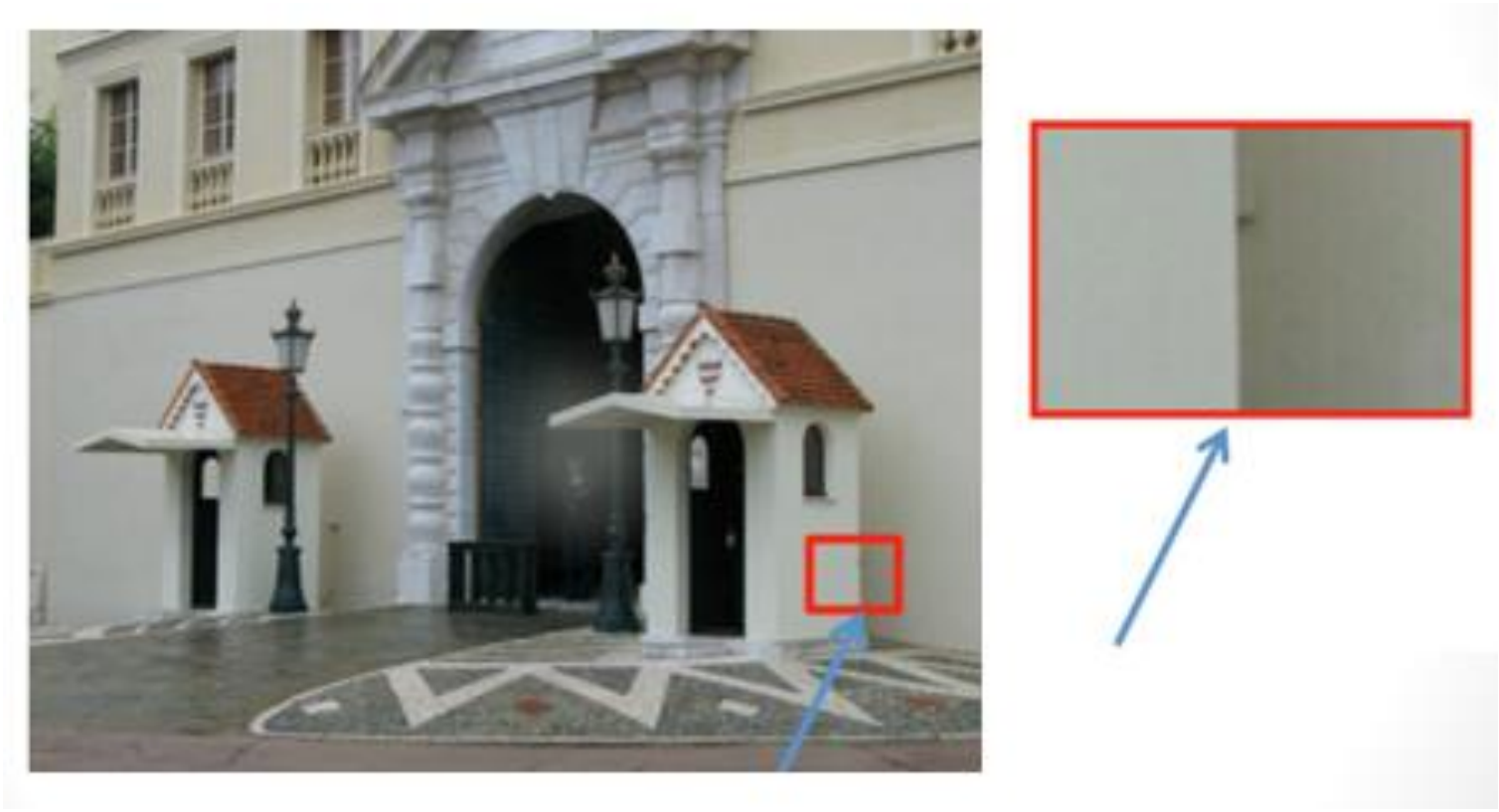
# DETECÇÃO DE DESCONTINUIDADES

# DETECÇÃO DE DESCONTINUIDADES

- Consiste em determinar pontos de uma imagem digital em que a intensidade muda repentinamente.

- Mudanças repentinas em imagens geralmente refletem eventos importantes:
  - Noção de profundidade.
  - Mudanças das propriedades do material.
  - Variações na iluminação da cena.

# DETECÇÃO DE DESCONTINUIDADES



# DETECÇÃO DE DESCONTINUIDADES

# DETECÇÃO DE DESCONTINUIDADES

# DETECÇÃO DE DESCONTINUIDADES

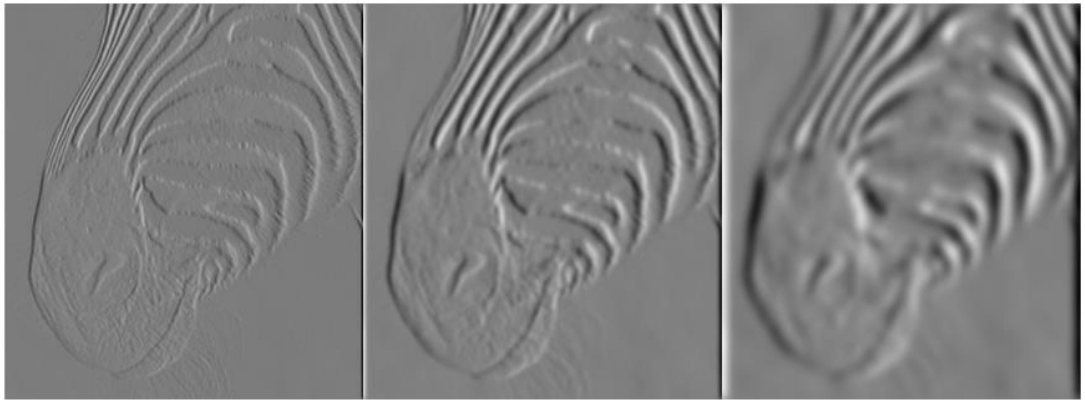
- Aplicações da detecção de descontinuidades:
  - Segmentação baseada em bordas.
  - Extração de características.

- Reduz significativamente a quantidade de dados a serem processados.
  - Descartando informação considerada menos relevante.
  - Preservando importantes propriedades estruturais da imagem.

# DETECÇÃO DE DESCONTINUIDADES

- Problema: a detecção pode ser prejudicada por bordas falsas criadas por ruídos na imagem (digitalização ou compressão).

- Solução: utilizar alguma técnica de redução de ruídos antes da detecção.



# DETECÇÃO DE DESCONTINUIDADES

- Existem três tipos básicos de descontinuidades em imagens digitais:
  - Pontos isolados.
  - Segmentos de retas.
  - Bordas.

- A maneira mais comum para detecção de descontinuidades é por meio da varredura da imagem por uma máscara.

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

| | | |
|---|---|---|
| $w_1$ | $w_2$ | $w_3$ |
| $w_4$ | $w_5$ | $w_6$ |
| $w_7$ | $w_8$ | $w_9$ |

# DETECÇÃO DE DESCONTINUIDADES

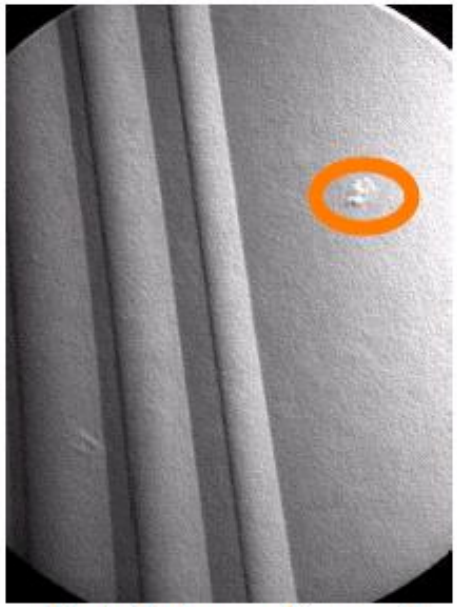
- Detecção de pontos isolados.
  - Pode ser realizada pela aplicação direta da máscara na imagem.
  - Um ponto é detectado na posição central da máscara se |R| > limiar.
    - Limiar não-negativo.

| | | |
|---|---|---|
| –1 | –1 | –1 |
| –1 | 8 | –1 |
| –1 | –1 | –1 |

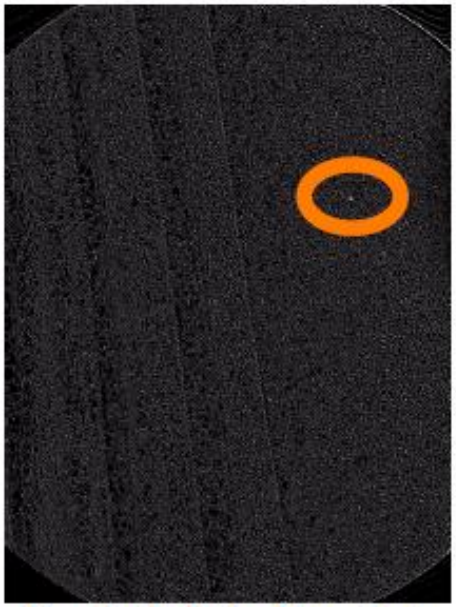
$$R = w_1 z_1 + w_2 z_2 + \cdots + w_9 z_9 = \sum_{i=1}^{9} w_i z_i$$

# DETECÇÃO DE DESCONTINUIDADES

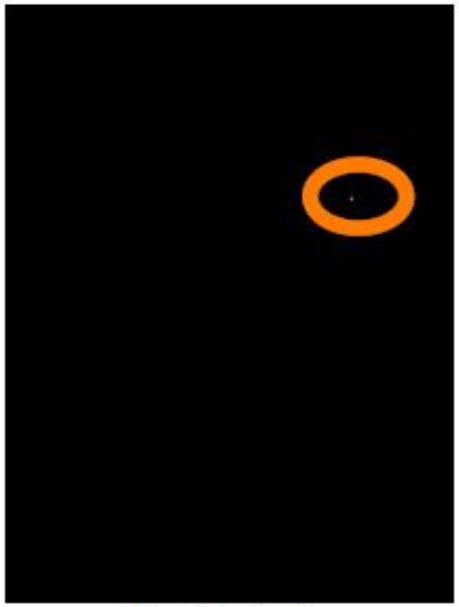
- Detecção de pontos isolados.



Raio X de uma lâmina de turbina



Resultado R da detecção de ponto



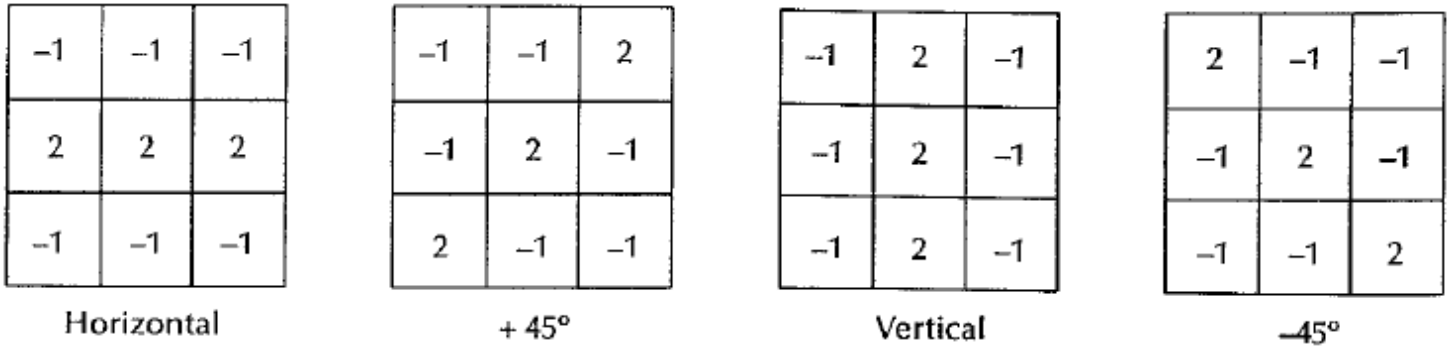
Resultado do thresholding

# DETECÇÃO DE DESCONTINUIDADES

- Detecção de pontos isolados.

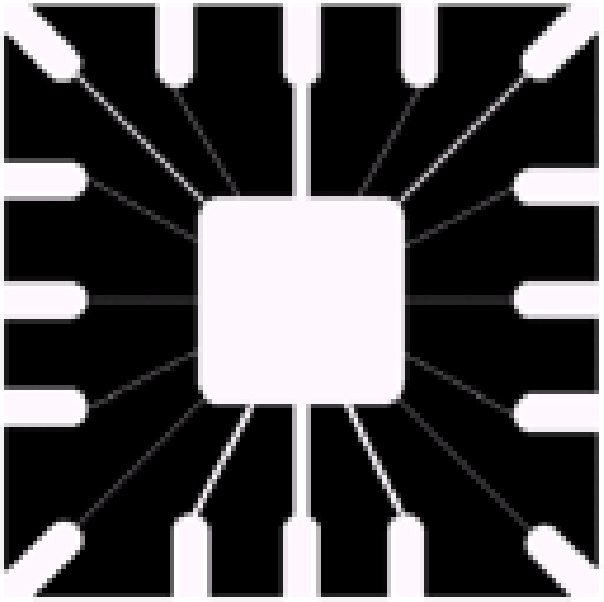
# DETECÇÃO DE DESCONTINUIDADES

- Detecção de retas.
  - Segmentos de retas também podem ser detectados pelo uso de máscaras.
  - Suponha que todas as máscaras são aplicadas em uma imagem. Se em um certo ponto da imagem, |Ri| > |Rj| para todos os j != i, então diz-se que esse ponto está provavelmente mais associado com uma linha na direção da máscara i.



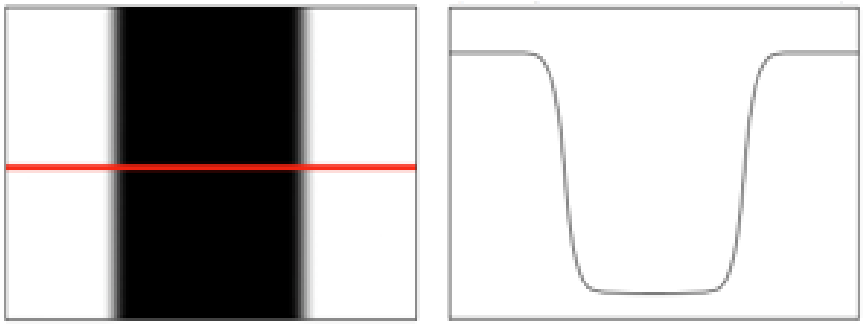
# DETECÇÃO DE DESCONTINUIDADES

- Detecção de retas.



# DETECÇÃO DE DESCONTINUIDADES

- Detecção de bordas.
  - Bordas: limite entre duas regiões com propriedades relativamente distintas de níveis de cinza.
  - Método usado com frequência na segmentação de imagens com base nas variações abruptas (local) de intensidade.

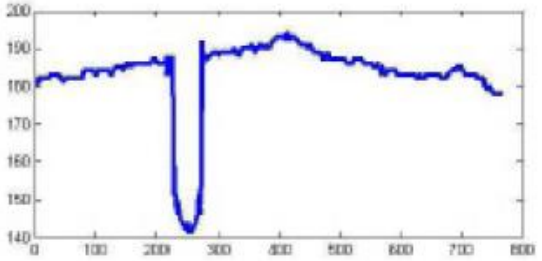


# DETECÇÃO DE DESCONTINUIDADES

- Detecção de bordas.
  - Borda é o tipo de característica mais importante presente em uma imagem.
  - Bordas são independente da iluminação.
  - Bordas são fáceis de detectar computacionalmente.
  - Bordas são usadas para determinar características de nível e complexidade maiores (linhas, curvas, cantos).

# DETECÇÃO DE DESCONTINUIDADES

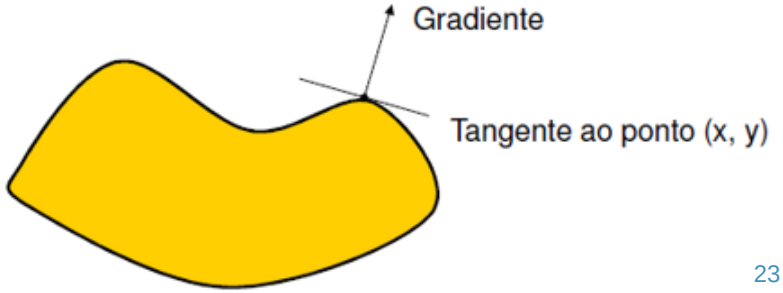
- Detecção de bordas.
  - Causas de bordas nas imagens:
    - Descontinuidade de profundidade.
    - Descontinuidade na orientação de uma superfície.
    - Descontinuidade na mudança das propriedades da superfície do material.
    - Descontinuidade na iluminação. Ex.: sombras.



# DETECÇÃO DE DESCONTINUIDADES

- Detecção de bordas.
  - Como fazer para detectar esses tipos de descontinuidades na imagem?
    - Usando o gradiente da imagem.
  - O gradiente é um vetor com magnitude nas direções x e y iguais às correspondentes derivadas parciais.
    - A direção do gradiente indica os locais nos quais os níveis de cinza sofrem maior variação.

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



# DETECÇÃO DE DESCONTINUIDADES

- Detecção de bordas.
  - Para funções discretas usa-se uma aproximação de 1ª ordem do gradiente.

$$\frac{df(x)}{dx} = \frac{f(x+h) - f(x-h)}{2h}$$

Onde h corresponde ao incremento adotado na coordenada x.

  - No caso de imagens, h corresponde a um pixel.

$$\frac{\partial \mathbf{I}(x,y)}{\partial x} = \frac{\mathbf{I}(x+1,y) - \mathbf{I}(x-1,y)}{2} \qquad \frac{\partial \mathbf{I}(x,y)}{\partial y} = \frac{\mathbf{I}(x,y+1) - \mathbf{I}(x,y-1)}{2}$$

# DETECÇÃO DE DESCONTINUIDADES

- Detecção de bordas.
  - Como fazer para calcular o gradiente de uma imagem de forma eficiente?
    - Usando a convolução da imagem com um filtro.

$$\frac{\partial \mathbf{I}(x,y)}{\partial x} = \frac{\mathbf{I}(x+1,y) - \mathbf{I}(x-1,y)}{2}$$

$$\frac{\partial \mathbf{I}(x,y)}{\partial y} = \frac{\mathbf{I}(x,y+1) - \mathbf{I}(x,y-1)}{2}$$

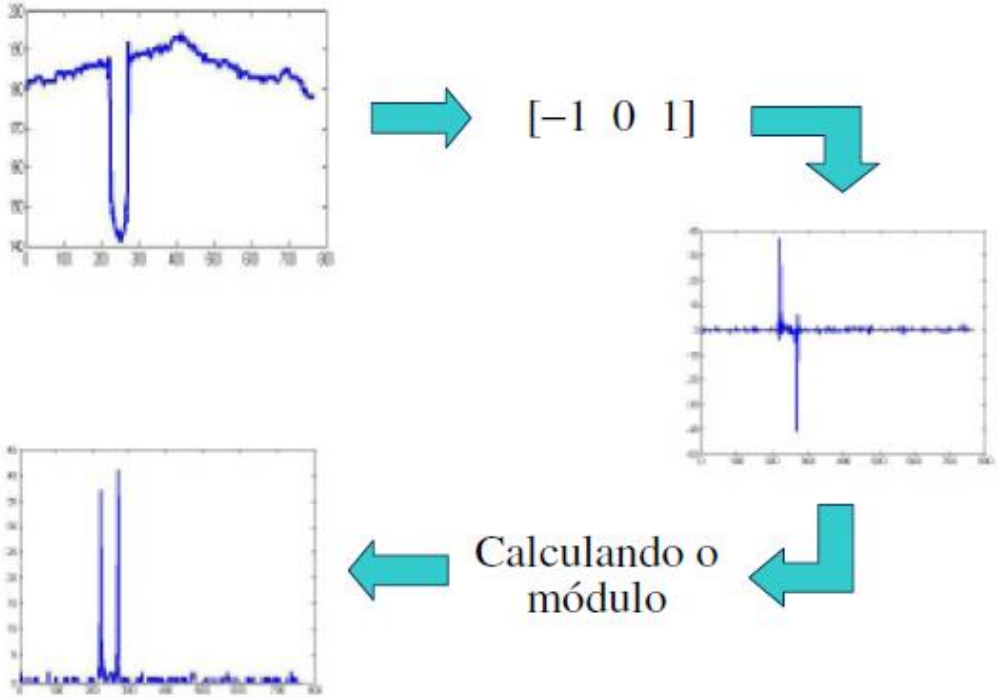
$$\frac{\partial \mathbf{I}}{\partial x} = \mathbf{I} \otimes \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

$$\frac{\partial \mathbf{I}}{\partial y} = \mathbf{I} \otimes \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

    - Note que a divisão por 2 foi eliminada. Isso é feito para aumentar a velocidade do cálculo.

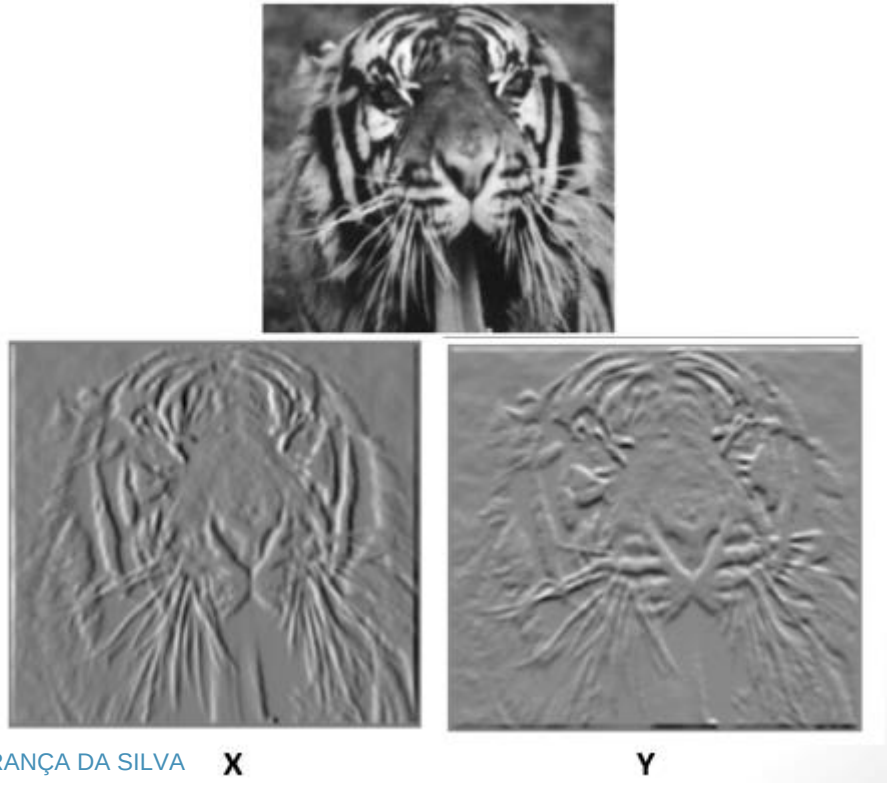
# DETECÇÃO DE DESCONTINUIDADES

- Detecção de bordas.



# DETECÇÃO DE DESCONTINUIDADES

- Detecção de bordas.

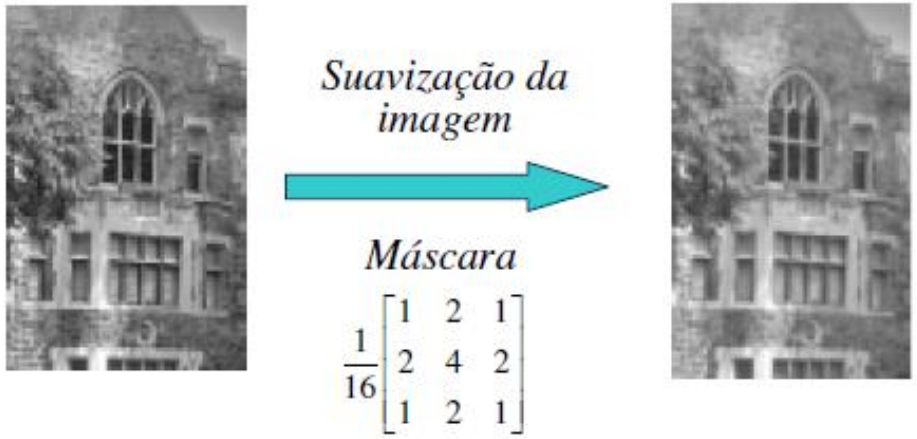


# DETECÇÃO DE DESCONTINUIDADES

- Detecção de bordas.
  - Etapas básicas do processo de detecção de bordas:
    - Filtrar: suavizar a imagem.
    - Ressaltar: calcular as derivadas parciais nas direções horizontal e vertical.
    - Calcular magnitude do gradiente: calcular o módulo do vetor gradiente.
    - Detectar: limiarizar a imagem ressaltada para achar bordas mais fortes.
    - Analisar: rejeitar ou incluir bordas.

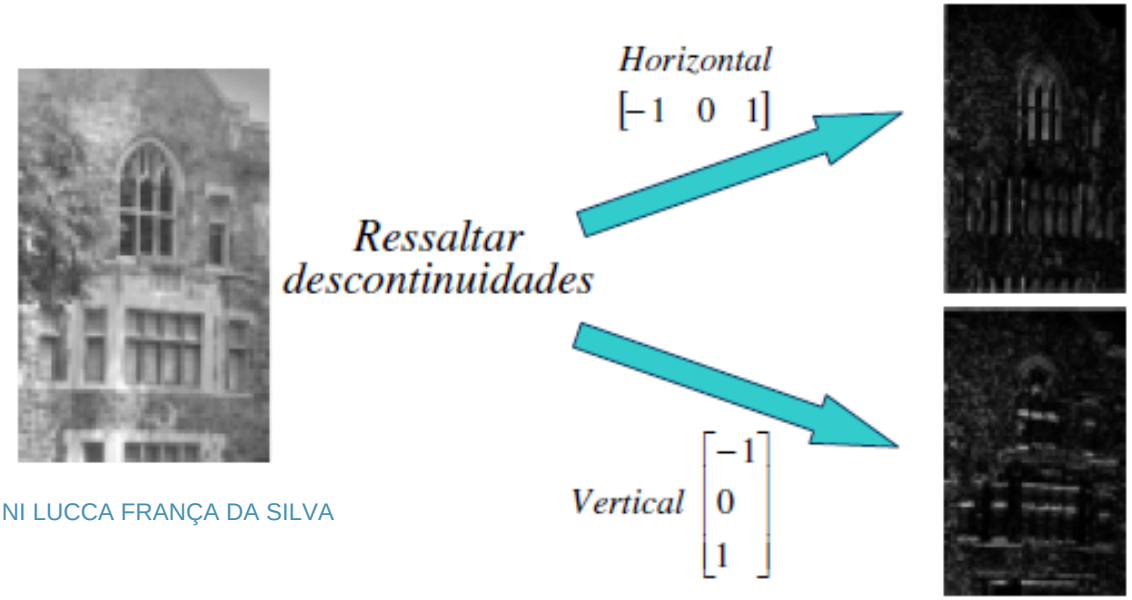
# DETECÇÃO DE DESCONTINUIDADES

- Detecção de bordas.
  - Primeiro passo: suavizar.
    - Cálculo do gradiente é muito suscetível a ruídos.
    - Suavização é importante para a eliminação de ruídos (pequenas bordas).



# DETECÇÃO DE DESCONTINUIDADES

- Detecção de bordas.
  - Segundo passo: ressaltar.
    - Cálculo do gradiente (derivadas parciais nas direções horizontal e vertical).
    - Ressalta altas frequências (descontinuidades).



# DETECÇÃO DE DESCONTINUIDADES

- Detecção de bordas.
  - Terceiro passo: calcular o módulo do gradiente.
    - No passo anterior foi calculado o gradiente nas direções x e y para cada ponto.
    - O cálculo do módulo do gradiente para cada pixel da imagem é dado por:

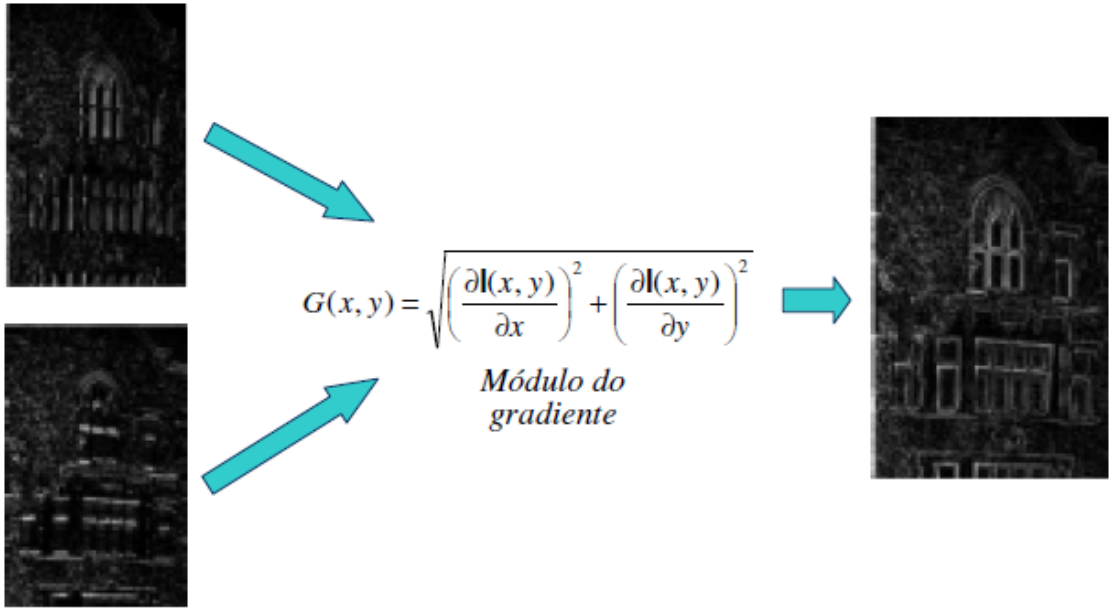
$$G(x,y) = \sqrt{\left(\frac{\partial \mathbf{I}(x,y)}{\partial x}\right)^2 + \left(\frac{\partial \mathbf{I}(x,y)}{\partial y}\right)^2}$$

    - Para diminuir o esforço computacional, calcula-se a soma dos valores absolutos das duas componentes do gradiente.

$$G(x,y) = \left|\frac{\partial \mathbf{I}(x,y)}{\partial x}\right| + \left|\frac{\partial \mathbf{I}(x,y)}{\partial y}\right|$$

# DETECÇÃO DE DESCONTINUIDADES

- Detecção de bordas.
  - Terceiro passo: calcular o módulo do gradiente.



# DETECÇÃO DE DESCONTINUIDADES

- Detecção de bordas.
  - Quarto passo: detectar bordas fortes.
    - Utilização de um limiar.

# DETECÇÃO DE DESCONTINUIDADES

- Detecção de bordas.
  - Quarto passo: detectar bordas fortes.

Limiar = 20

Limiar = 50

# DETECÇÃO DE DESCONTINUIDADES

- Detecção de bordas.
  - Operador gradiente:

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial x} \end{bmatrix}$$

$$G = [G_x G_y]$$

  - Magnitude do gradiente:

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

$$G_m = \sqrt{G_x^2 + G_y^2} \quad \text{ou} \quad G_m = |G_x| + |G_y|$$

  - Direção do gradiente:

$$\theta = \tan^{-1}\left( \frac{G_y}{G_x} \right)$$

# DETECÇÃO DE DESCONTINUIDADES

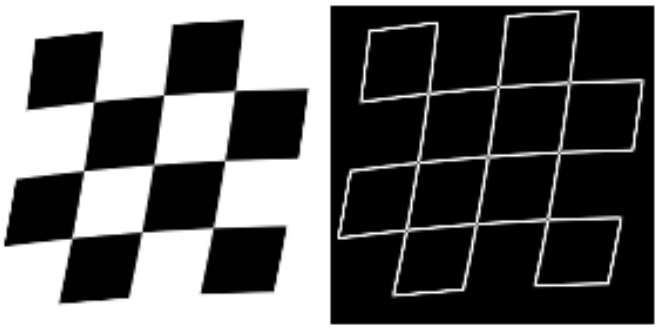
- Detecção de bordas.
  - Métodos usuais para detecção de bordas:
    - Roberts (1965): primeiro método desenvolvido.
    - Sobel (1990): provavelmente o mais utilizado.
    - Prewitt (1970): similar ao Sobel.
    - Laplaciano.

# DETECÇÃO DE DESCONTINUIDADES

- Detecção de bordas.
  - Roberts (1965).
    - Método mais simples e sensível a ruídos.
    - Melhor aplicado em imagens binárias.

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

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



# DETECÇÃO DE DESCONTINUIDADES

- Detecção de bordas.
  - Roberts (1965).

# DETECÇÃO DE DESCONTINUIDADES

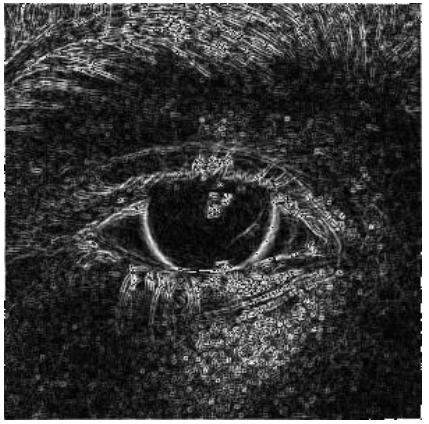
- Detecção de bordas.
  - Prewitt (1970).

Horizontal = $G_x$

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

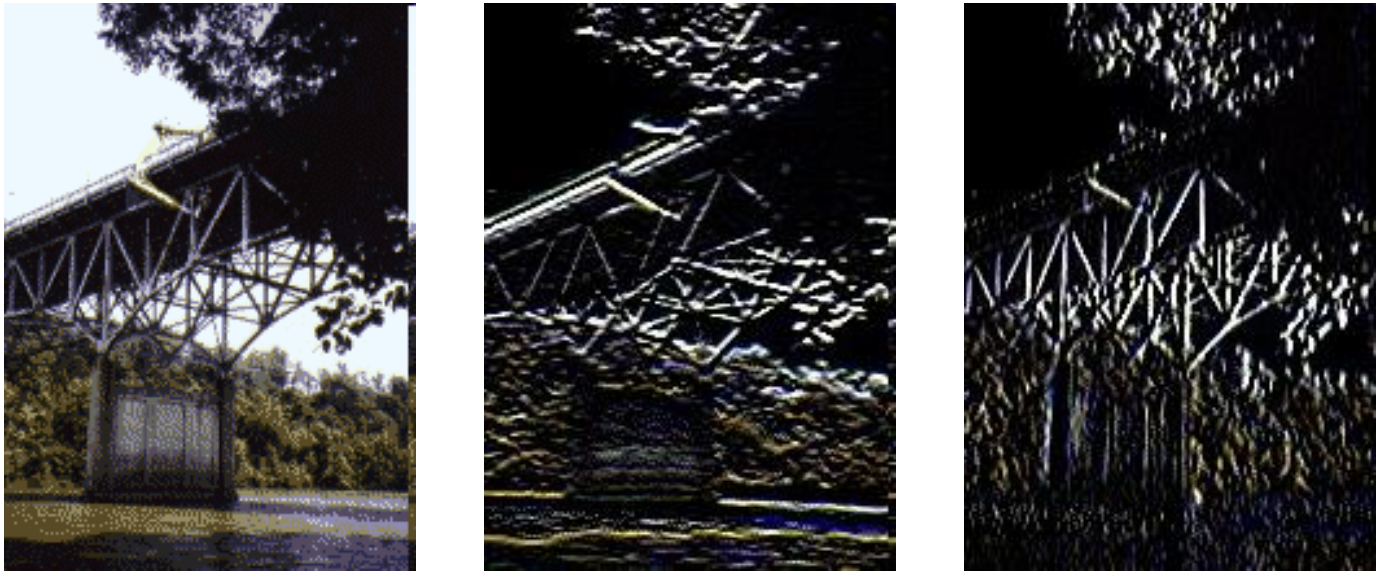
Vertical = $G_y$

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



# DETECÇÃO DE DESCONTINUIDADES

- Detecção de bordas.
  - Prewitt (1970).



# DETECÇÃO DE DESCONTINUIDADES

- Detecção de bordas.
  - Sobel (1990).
    - Ponderado.

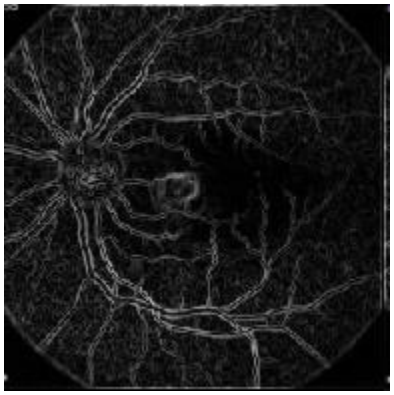
Mudanças na horizontal= $G_x$

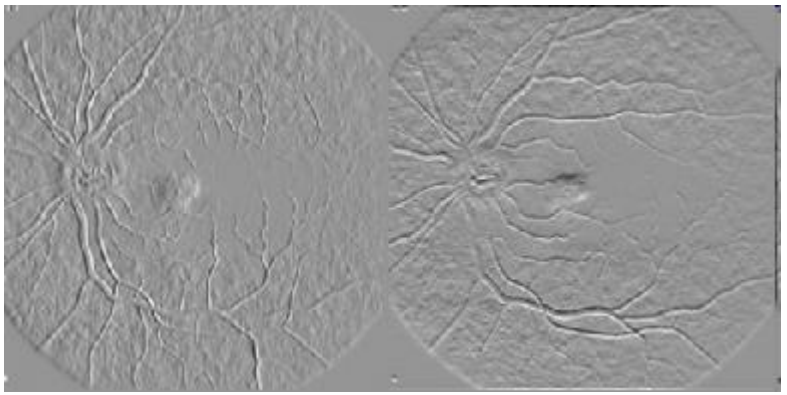
$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Mudanças na vertical= $G_y$

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

G





# DETECÇÃO DE DESCONTINUIDADES

- Detecção de bordas.
  - Sobel (1990).

| | | |
|---|---|---|
| $a$ | $b$ | $c$ |
| $d$ | $e$ | $f$ |
| $g$ | $h$ | $i$ |

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

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

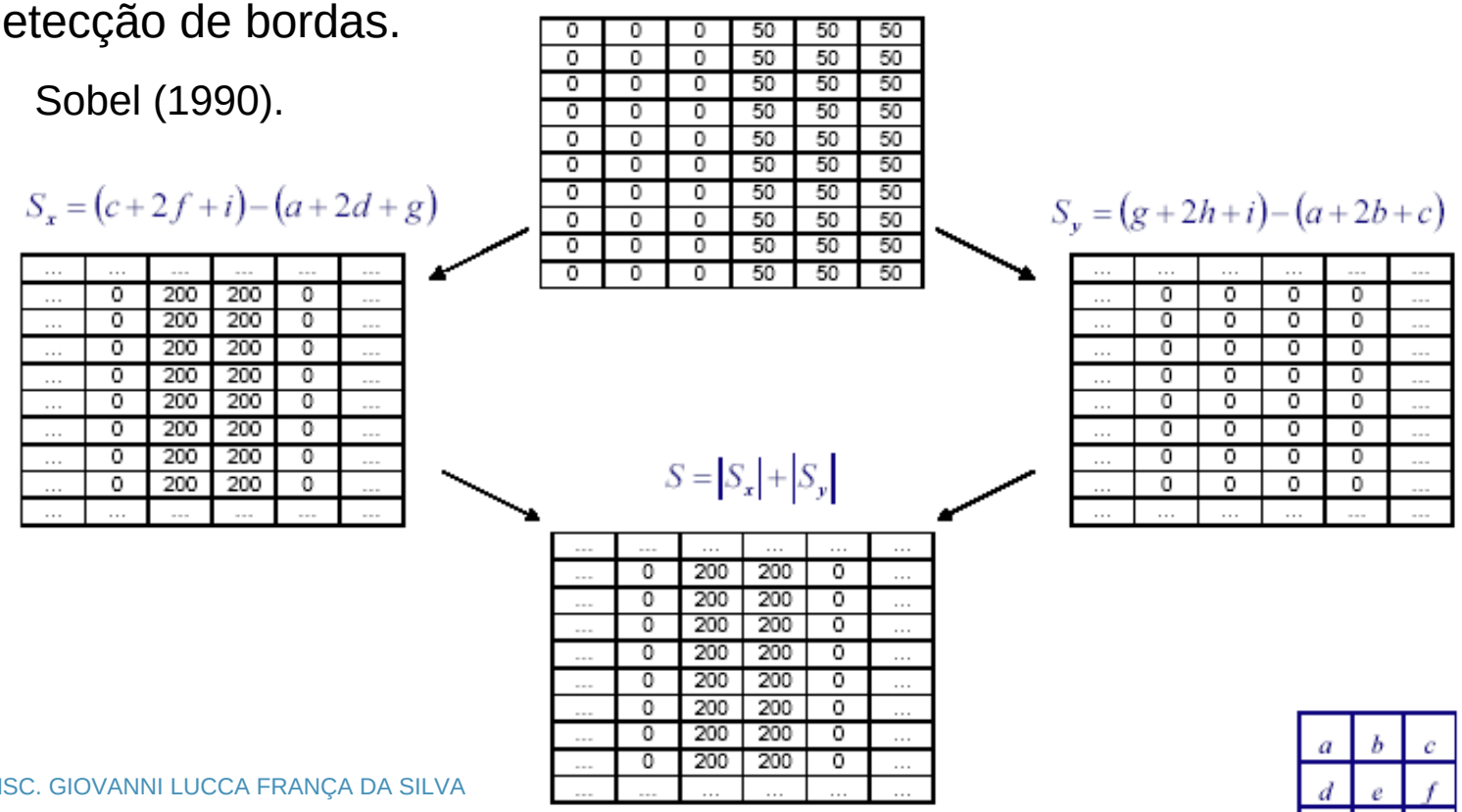
$$S = \sqrt{S_x^2 + S_y^2}$$

Ou numa versão simplificada:

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

# DETECÇÃO DE DESCONTINUIDADES

- Detecção de bordas.
  - Sobel (1990).




| a | b | c |
|---|---|---|
| d | e | f |
| g | h | i |

# DETECÇÃO DE DESCONTINUIDADES

- Detecção de bordas.
  - Sobel (1990).

# DETECÇÃO DE DESCONTINUIDADES

- Detecção de bordas.
  - Código do Sobel no OpenCV.

**C++:** void **Sobel**(InputArray **src**, OutputArray **dst**, int **ddepth**, int **dx**, int **dy**, int **ksize**=3, double **scale**=1, double **delta**=0, int **borderType**=BORDER_DEFAULT )

**Python:** `cv2.Sobel`(src, ddepth, dx, dy[, dst[, ksize[, scale[, delta[, borderType]]]]]) → dst

**Parameters:**
- **src** – input image.
- **dst** – output image of the same size and the same number of channels as `src` .
- **ddepth** –

  output image depth; the following combinations of `src.depth()` and `ddepth` are supported:

  - `src.depth()` = `CV_8U`, `ddepth` = -1/`CV_16S`/`CV_32F`/`CV_64F`
  - `src.depth()` = `CV_16U`/`CV_16S`, `ddepth` = -1/`CV_32F`/`CV_64F`
  - `src.depth()` = `CV_32F`, `ddepth` = -1/`CV_32F`/`CV_64F`
  - `src.depth()` = `CV_64F`, `ddepth` = -1/`CV_64F`

- **xorder** – order of the derivative x.
- **yorder** – order of the derivative y.

# DETECÇÃO DE DESCONTINUIDADES

- Detecção de bordas.

Roberts

Sobel

Prewitt

Isotrópico

# DETECÇÃO DE DESCONTINUIDADES

- Detecção de bordas.
  - Laplaciano.
    - Não é muito utilizado por ser uma derivada de segunda ordem.
    - Muito sensível ao ruído.
    - Produz bordas duplas.

$$h(i,j) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad h(i,j) = \begin{bmatrix} 2 & -1 & 2 \\ -1 & -4 & -1 \\ 2 & -1 & 2 \end{bmatrix}$$

# DETECÇÃO DE DESCONTINUIDADES

- Detecção de bordas.
  - Código do Laplaciano no OpenCV.

**C++:** void **Laplacian**(InputArray **src**, OutputArray **dst**, int **ddepth**, int **ksize**=1, double **scale**=1, double **delta**=0, int **borderType**=BORDER_DEFAULT )

**Python:** cv2.**Laplacian**(src, ddepth[, dst[, ksize[, scale[, delta[, borderType]]]]]) → dst

**Parameters:**
- **src** – Source image.
- **dst** – Destination image of the same size and the same number of channels as `src` .
- **ddepth** – Desired depth of the destination image.
- **ksize** – Aperture size used to compute the second-derivative filters. See `getDerivKernels()` for details. The size must be positive and odd.
- **scale** – Optional scale factor for the computed Laplacian values. By default, no scaling is applied. See `getDerivKernels()` for details.
- **delta** – Optional delta value that is added to the results prior to storing them in `dst` .
- **borderType** – Pixel extrapolation method. See `borderInterpolate()` for details.

# DETECÇÃO DE DESCONTINUIDADES

- Detecção de bordas.
  - Exemplo no OpenCV.

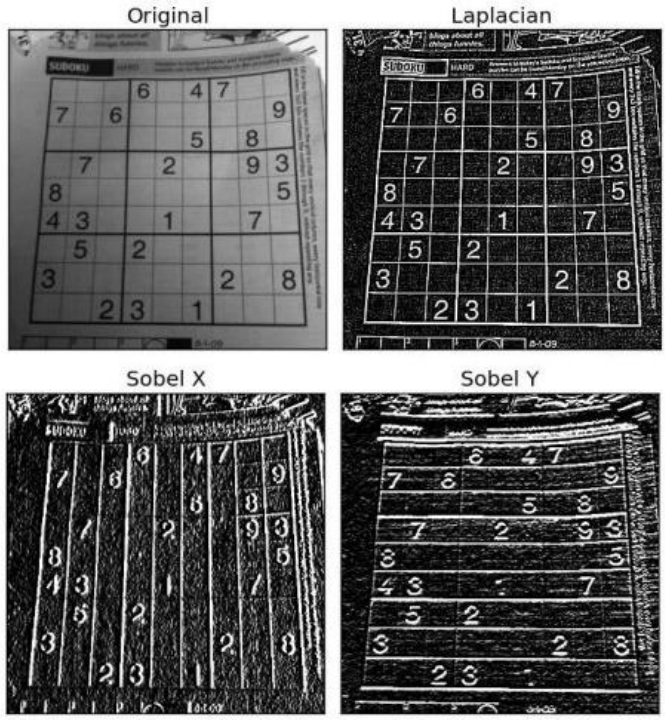
```
import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('dave.jpg',0)

laplacian = cv2.Laplacian(img,cv2.CV_64F)
sobelx = cv2.Sobel(img,cv2.CV_64F,1,0,ksize=5)
sobely = cv2.Sobel(img,cv2.CV_64F,0,1,ksize=5)

plt.subplot(2,2,1),plt.imshow(img,cmap = 'gray')
plt.title('Original'), plt.xticks([]), plt.yticks([])
plt.subplot(2,2,2),plt.imshow(laplacian,cmap = 'gray')
plt.title('Laplacian'), plt.xticks([]), plt.yticks([])
plt.subplot(2,2,3),plt.imshow(sobelx,cmap = 'gray')
plt.title('Sobel X'), plt.xticks([]), plt.yticks([])
plt.subplot(2,2,4),plt.imshow(sobely,cmap = 'gray')
plt.title('Sobel Y'), plt.xticks([]), plt.yticks([])

plt.show()
```



# REFERÊNCIAS


- GONZALEZ, Rafael C.; WOODS, Richard C. **Processamento digital de imagens**. Pearson, 2011.

- PEDRINI, Hélio; SCHWARTZ, William Robson. **Análise de imagens digitais: princípios, algoritmos e aplicações**. Thomson Learning, 2008.

- SILVA, Aristófanes. **Notas de aula da disciplina Processamento de Imagens da Universidade Federal do Maranhão**. 2018.

- BRAZ Jr, Geraldo. **Notas de aula da disciplina Visão Computacional da Universidade Federal do Maranhão**. 2018.