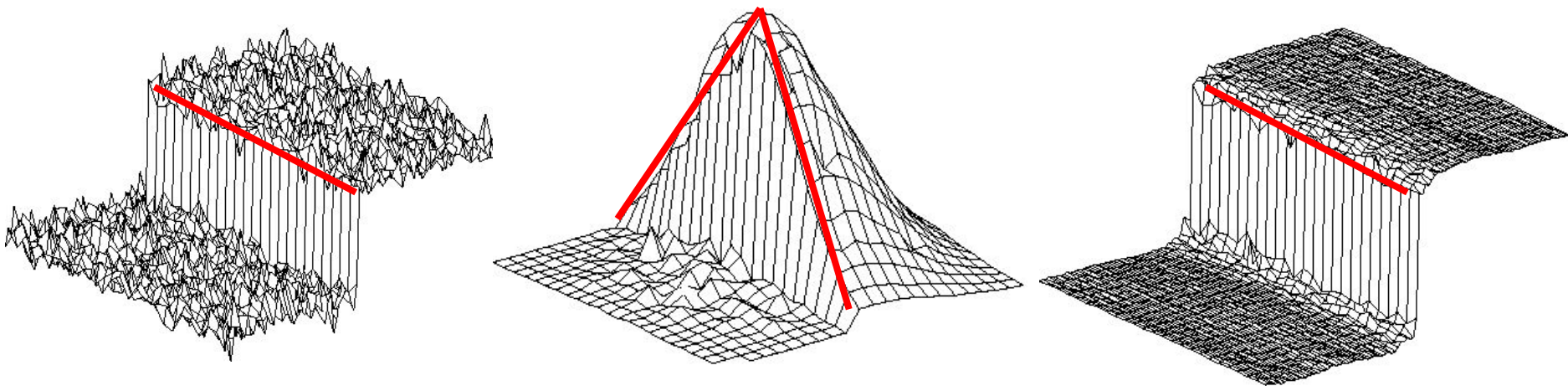


# GRADIENTES DE IMAGEM

**ES235 – Aula 06**  
**João Marcelo Teixeira**  
**Willams Costa**

# FILTROS PASSA-ALTA

- Objetivo: realçar alta-frequência, ou seja, variações bruscas nos pixels da imagem (regiões de borda)



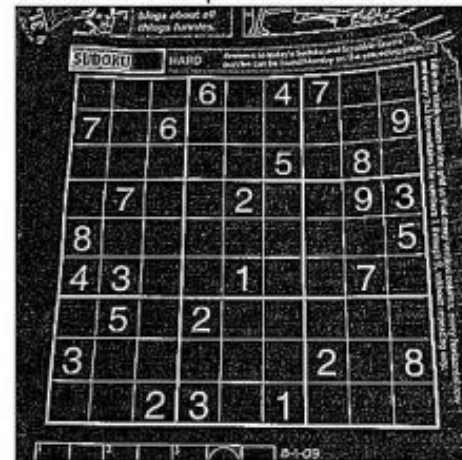
# FILTROS PASSA-ALTA

- OpenCV disponibiliza 3 filtros passa-alta:
  - Sobel
  - Scharr
  - Laplaciano

Original



Laplacian



Sobel X



Sobel Y



# SOBEL



Original



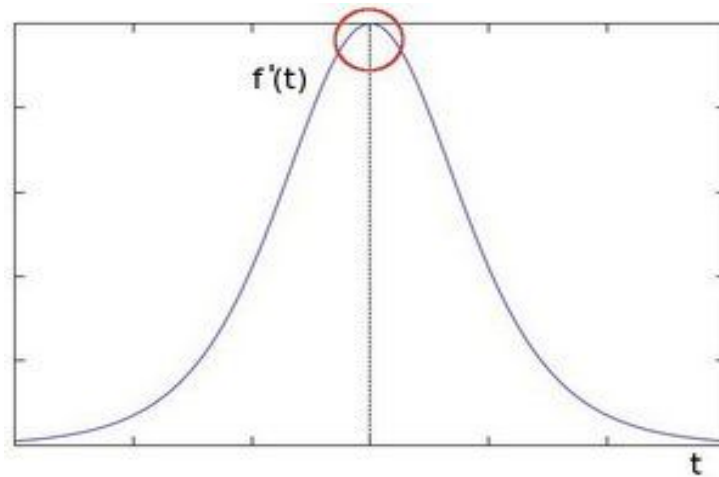
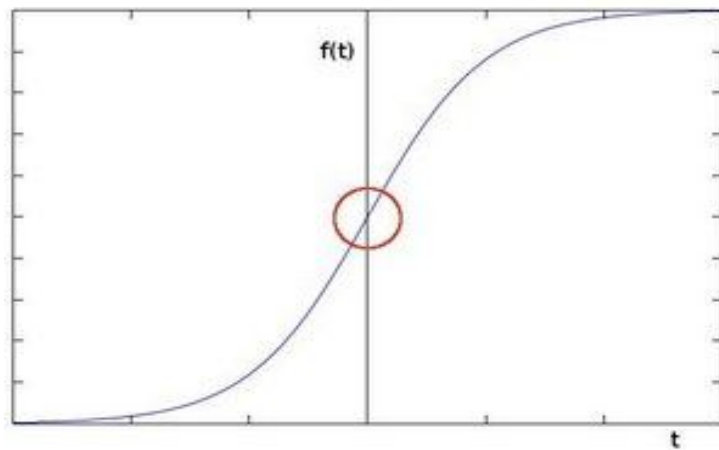
Sobel em X



Sobel em Y

# SOBEL

- 



# SOBEL

- 

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * I$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * I$$

$$G = \sqrt{G_x^2 + G_y^2}$$

ou

$$G = |G_x| + |G_y|$$

# SCHARR

- Quando o tamanho do Kernel é 3x3 (ou pequeno), Scharr fornece uma melhor aproximação da derivada da imagem em comparação ao Sobel

$$G_x = \begin{bmatrix} -3 & 0 & +3 \\ -10 & 0 & +10 \\ -3 & 0 & +3 \end{bmatrix}$$

$$G_y = \begin{bmatrix} -3 & -10 & -3 \\ 0 & 0 & 0 \\ +3 & +10 & +3 \end{bmatrix}$$

`Scharr(src, dst, ddepth, dx, dy, scale, delta, borderType)`

e

`Sobel(src, dst, ddepth, dx, dy, CV_SCHARR, scale, delta, borderType).`

são equivalentes

# LAPLACIANO

$$\Delta src = \frac{\partial^2 src}{\partial x^2} + \frac{\partial^2 src}{\partial y^2}$$

Exemplo de Laplaciano 3x3 (calculado a partir de filtros de Sobel):

$$kernel = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$



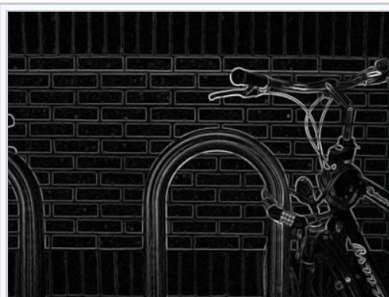
# LAPLACIANO



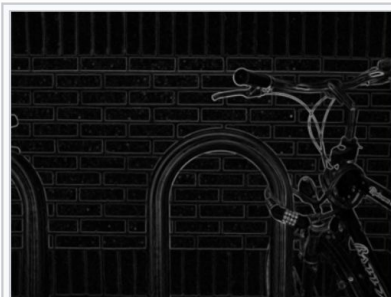
# COMPARANDO DIFERENTES GRADIENTES



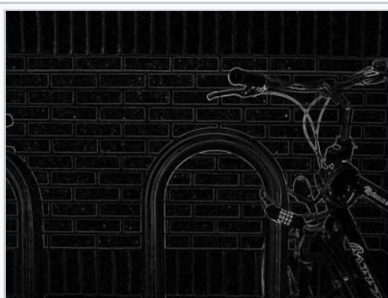
Grayscale test image of brick wall and bike rack



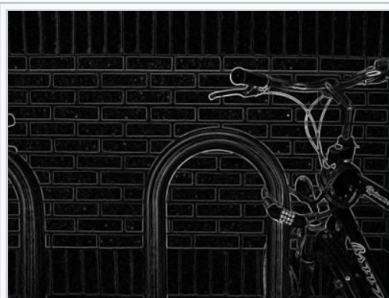
Gradient magnitude from Sobel-Feldman operator



Gradient magnitude from Scharr operator



Gradient magnitude from Roberts Cross operator



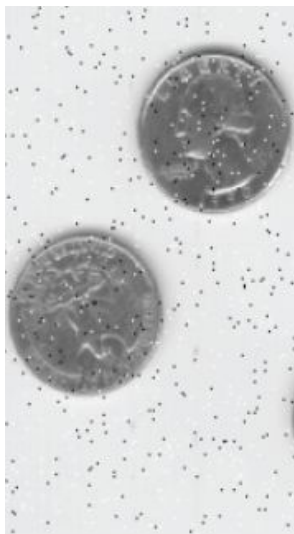
Gradient magnitude from Prewitt operator

# CANNY EDGE DETECTOR

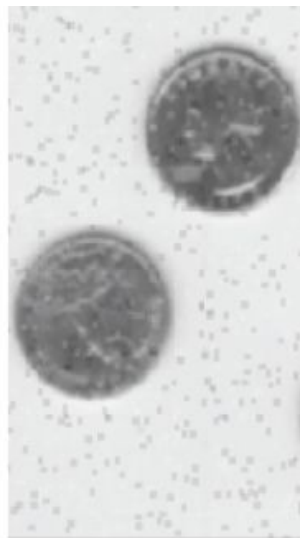
- Passo a passo do algoritmo:
  - a) Suaviza a imagem com um filtro Gaussiano 5x5 (reduz ruído)
  - b) Calcula os gradientes da imagem
  - c) Realiza a supressão de pontos de não-máximo
  - d) Aplica uma limiarização para eliminar pixels de borda falsos-positivos

# CANNY EDGE DETECTOR

- Passo a passo do algoritmo:
  - Suaviza a imagem com um filtro Gaussiano 5x5 (reduz ruído)



input



Gaussian filter

# CANNY EDGE DETECTOR

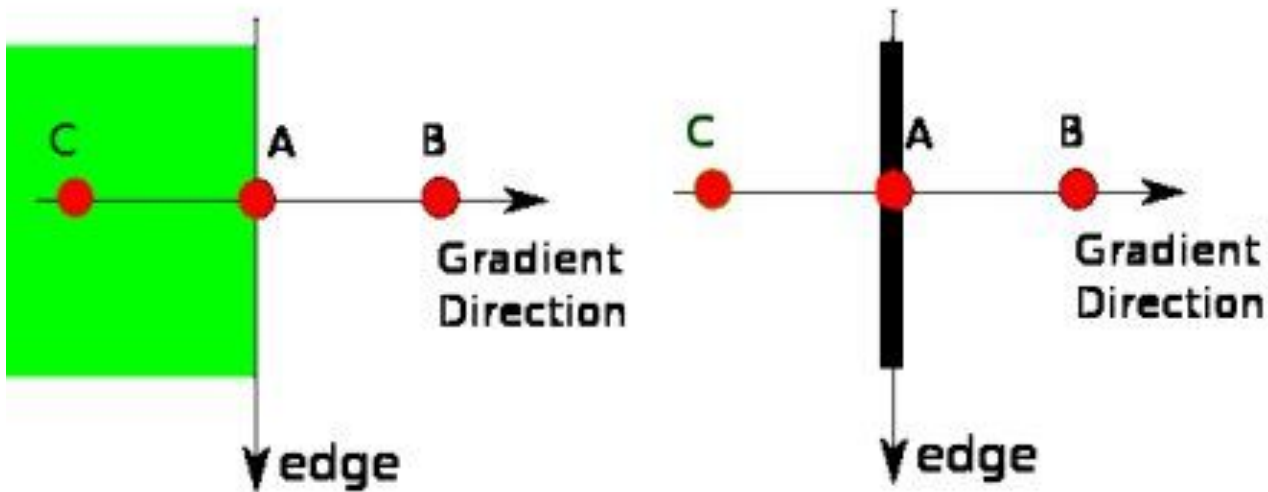
- Passo a passo do algoritmo:
  - Calcula os gradientes da imagem

$$\textit{Edge\_Gradient} (G) = \sqrt{G_x^2 + G_y^2}$$

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

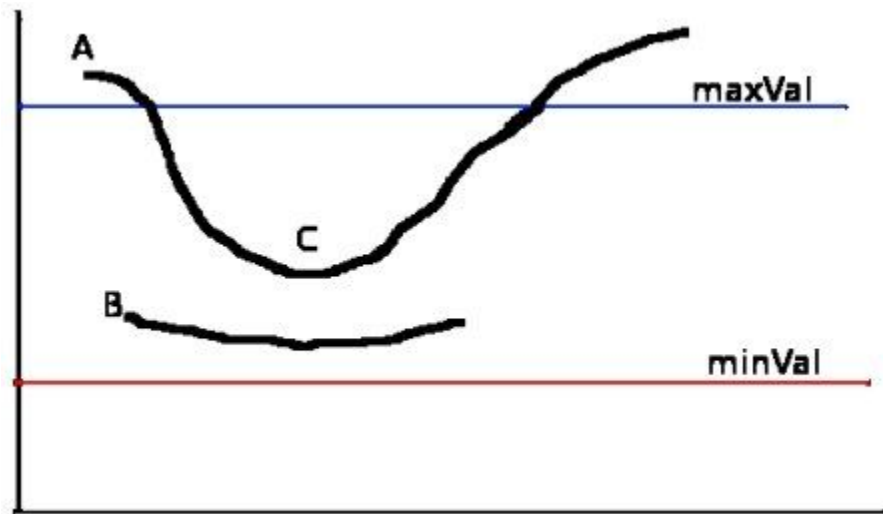
# CANNY EDGE DETECTOR

- Passo a passo do algoritmo:
  - Realiza a supressão de pontos de não-máximo



# CANNY EDGE DETECTOR

- Passo a passo do algoritmo:
  - Aplica uma limiarização para eliminar pixels de borda falsos-positivos



# CANNY EDGE DETECTOR

- Passo a passo do algoritmo:
  - Exemplo de resultado

Original Image



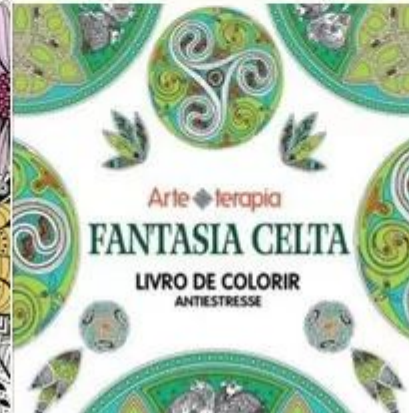
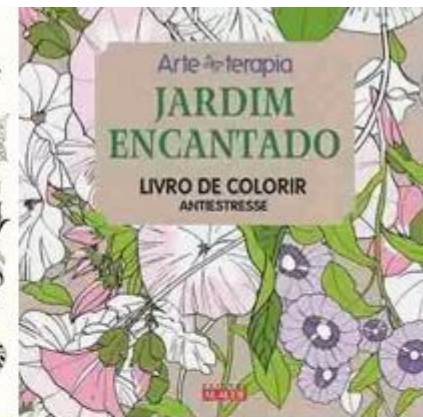
Edge Image





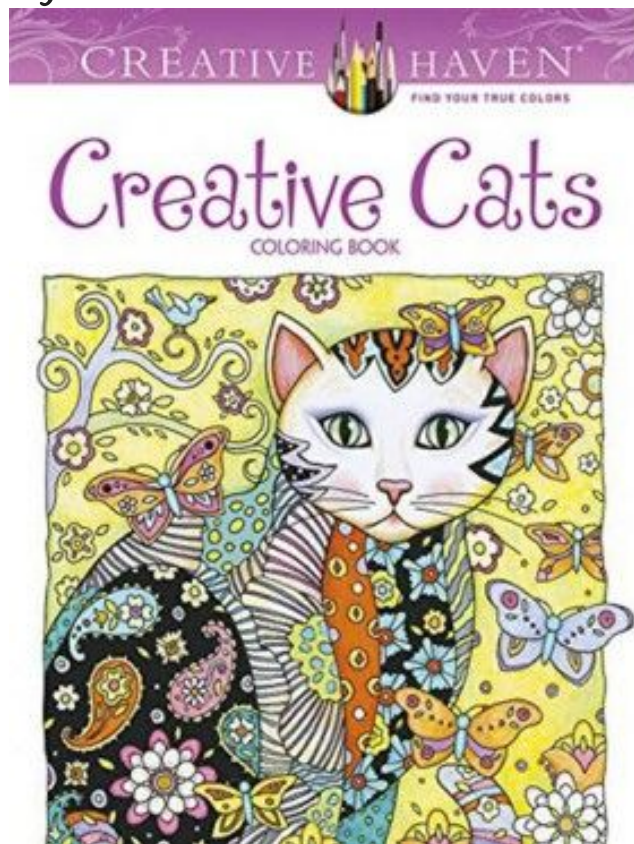
# EXEMPLO DE APLICAÇÃO

- Criação de “livros de colorir”



# EXEMPLO DE APLICAÇÃO

- Criação de “livros de colorir”





# EXEMPLO DE APLICAÇÃO



# REFERÊNCIAS

Rafael C. Gonzalez and Richard E. Woods. 2006. Digital Image Processing (3rd Edition). Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

[https://docs.opencv.org/3.2.0/d5/d0f/tutorial\\_py\\_gradients.html](https://docs.opencv.org/3.2.0/d5/d0f/tutorial_py_gradients.html)

[https://docs.opencv.org/2.4.13.7/doc/tutorials/imgproc/imgtrans/sobel\\_derivatives/sobel\\_derivatives.html](https://docs.opencv.org/2.4.13.7/doc/tutorials/imgproc/imgtrans/sobel_derivatives/sobel_derivatives.html)

[https://docs.opencv.org/3.4.3/da/d85/tutorial\\_js\\_gradients.html](https://docs.opencv.org/3.4.3/da/d85/tutorial_js_gradients.html)