Universidade do Estado de Santa Catarina – Campus CCT

Disciplina: Processamento de Imagens

Docente: Dr. Gilmario Barbosa dos Santos

Discentes: Adilson Krischanski e Brian Laus Bertemes

DETECÇÃO DE BORDA

Joinville 24 de setembro de 2022

Objetivo:

Aplicar os conceitos trabalhando de detecção de borda trabalhados em sala de aula, utilizando as imagens Lua1_gray.jpg, chessboard_inv.png e img2.jpg disponibilizadas pelo professor e em seguida implementar e aplicar nas imagens o filtro de detecção de borda com os operadores de Sobel, Prewitt e Scharr.

Procedimento

De forma similar a atividade anterior (segmentação por área) continua utilizando os conceitos de orientação objeto para o desenvolvimento, ultimamos a classe imagem já implementada na primeira atividade e realizamos algumas alterações, a classe main é a responsável por executar o código e gerar o resultado.

A classe imagem possui como atributos a imagem a ser filtrada, e seu tamanho. Também é nessa classe que ficam os métodos necessários para a manipulação da imagem.

A classe main e a classe que instancia a classe imagem, dado que o enunciado pede para que os testes sejam realizados com três imagens diferentes, instanciamos 3 objetos do tipo Imagem, sendo um para cada imagem e em seguida clamamos o para cada imagem os a aplicação do filtro com os três parâmetros 1 (sobel) 2 (Prewitt) 3 (Scharr).

Código Desenvolvido

O código foi desenvolvido na linguagem de script python com auxílio das bibliotecas OpenCV (cv2), matplotlb.pyplot (plt), math, skimage.exposure e numpy (pd).

Para que pudéssemos iniciar a codificação criamos as matrizes com os valores de x e y para cada um dos filtros solicitados no enunciado.

- scharr_x = np.array(([[-3, 0, 3], [-10, 0, 10], [-3, 0, 3]]),dtype="int")
- scharr_y = np.array(([[-3, -10, -3], [0, 0, 0], [3, 10, 3]]),dtype="int")
- prewitt_x = np.array(([[-1, 0, 1], [-1, 0, 1], [-1, 0, 1]]),dtype="int")
- prewitt_y = np.array(([[-1, -1, -1], [0, 0, 0], [1, 1, 1]]),dtype="int")
- sobel_x = np.array(([[-1, 0, 1], [-2, 0, 2], [-1, 0, 1]]), dtype="int")
- sobel_y = np.array(([[-1, -2, -1], [0, 0, 0], [1, 2, 1]]), dtype="int")
-

Convolve: E a função responsável por aplicar o a relação de vizinhança a imagem e, e retornar os valores dos pixeis devidamente alterados de acordo com kernel passado, seja para o eixo x ou para o eixo y para que em seguida possa ser aplicado dentro a função View_img a relação gradiente.

View_img: recebe como parâmetro o filtro a ser utilizado na aplicação sendo 1 (sobel) 2 (Prewitt) 3 (Scharr), a função chama incialmente passa a imagem original para tons de cinza em seguida chama a função convolve passando os parâmetros do filtro uma vez para x e outra para y colocando os resultados nas variáveis out_x out_y, em seguida são feitas as operações do gradiente.

O código completo pode ser encontrado em GitHub: https://github.com/adilsonkrischanski/PIM/tree/main/trabalho2 .

Resultados

Quando aplicamos o kernel para X temos uma relação de todas as bordas verticais da imagem, ao aplicar os o kernel para Y temos a relação de todas as bordas horizontais das, depois aplicar a relação de gradiente entre ambos os resultados temos uma imagem com todas as bordas presentes na imagem que foi analisada.

Imagens após a aplicacao dos seus respectivos filtros

Imagem chessboard_inv.png com kernel Sobel

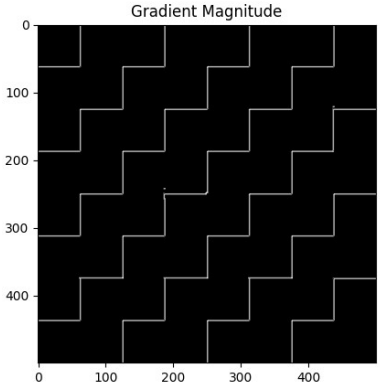


Imagem chessboard_inv.png com kernel prewitt

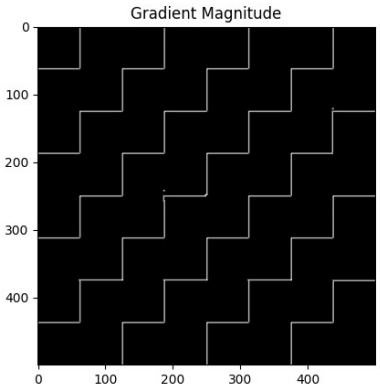


Imagem chessboard_inv.png com kernel scharr

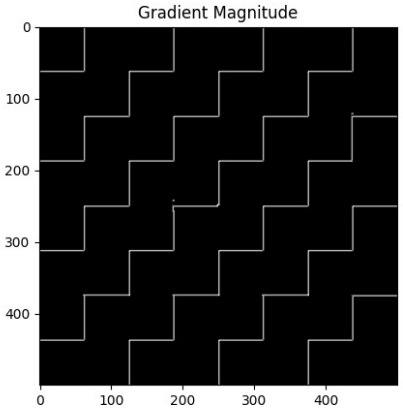


Imagem Lua1_gray.jpg com kernel Sobel

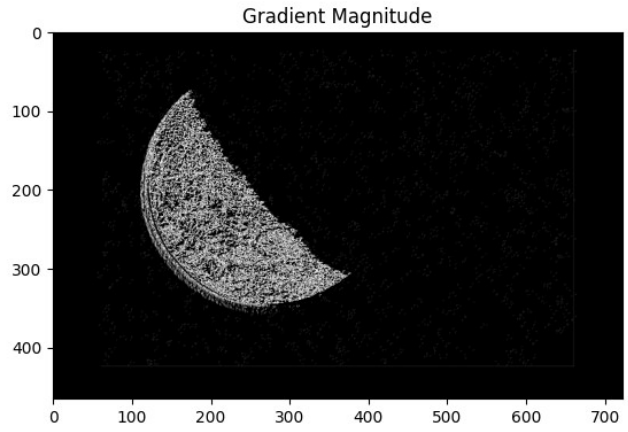


Imagem Lua1_gray.jpg com kernel prewitt

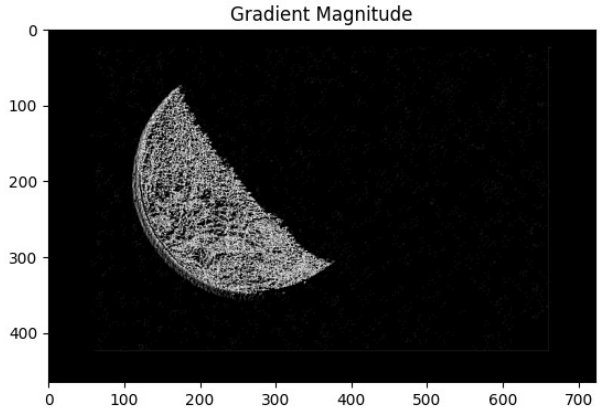


Imagem Lua1_gray.jpg com kernel scharr

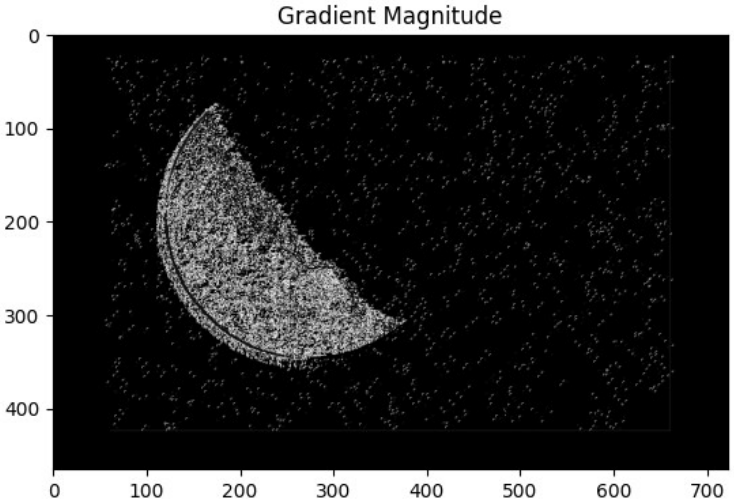


Imagem img02.jpg kernel Sobel

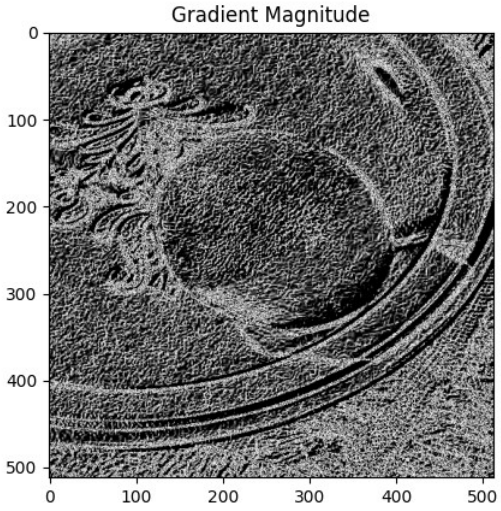


Imagem img02.jpg com kernel prewitt

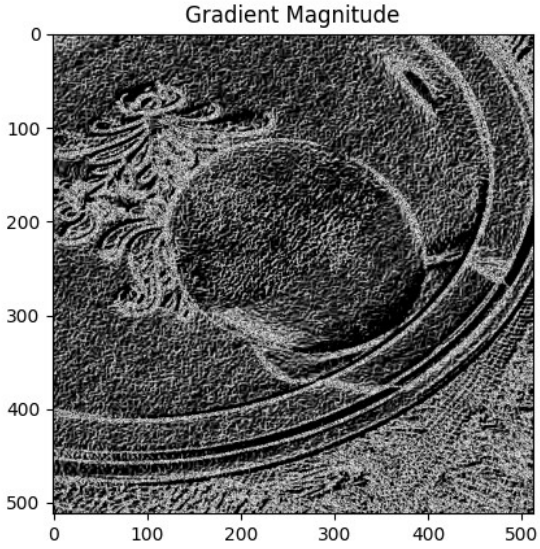
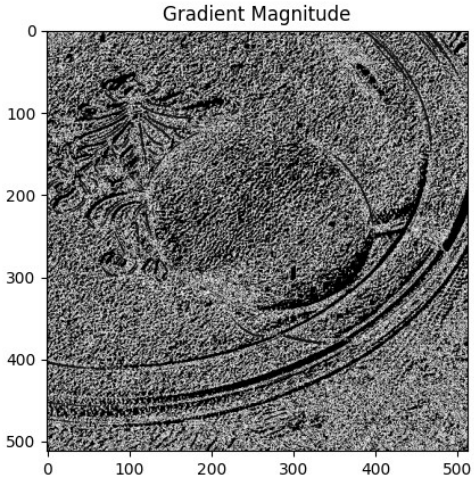


Imagem img02.jpg com kernel scharr



Referencias de Pesquisa

Moodle, Realce/Filtragem de imagens, Processamento de Imagem Disponivel em: <https://moodle.joinville.udesc.br/pluginfile.php/220431/mod_resource/content/4/filtragem.pdf> Acesso em 20 de outubro de 2022.

Moodle, OPERADOR GRADIENTE, Processamento de Imagem Disponivel em:<https://moodle.joinville.udesc.br/pluginfile.php/220432/mod_resource/content/3/gradiente_2022_1_B.pdf> Acesso em 20 de outubro de 2022.

KRISCHANSKI, Adilson. PIM. Github, 2022. Disponível em: <https://github.com/adilsonkrischanski/PIM/tree/main/trabalho1> . Acesso em: 21 de outubro de 2022.