UNIVERSIDADE FEDERAL DE OURO PRETO - UFOP
DEPARTAMENTO DE COMPUTAÇÃO - DECOM

Disciplina: PROCESSAMENTO DE IMAGENS - BCC 326
Prof.: Guillermo Cámara Chávez
Curso: Ciência da Computação

LISTA 07

Gustavo Lucas Moreira[1]
gustavo.lucas@aluno.ufop.edu.br

17.2.4289

1. A função *edge* do MATLAB encontra as bordas de uma imagem. A função vem implementada com vários tipos de máscaras (*Sobel*, *Prewitt*, *Roberts*, *Canny*, Laplaciano, Zero Crossing). Carregar uma imagem e encontrar as bordas usando os diferentes filtros. Qual obtém melhor resultado?

Após a aplicação das várias máscaras utilizando a função *edge,* nota-se que a máscara *Canny* obteve o melhor resultado em destacar as bordas da imagem.

Execução:

- Sobel :

```
>> imgSobel = imread('lennagray.png');
>> imgSobel = edge(imgSobel, 'Sobel');
```

[1] Graduando do Curso de Ciência da Computação da Universidade Federal de Ouro Preto

```
>> imshow(imgSobel);
```

- Prewitt:

```
>> imgPrewitt = imread('lennagray.png');
>> imgPrewitt = edge(imgPrewitt, 'Prewitt');
>> imshow(imgPrewitt);
```

- Roberts:

```
>> imgRoberts = imread('lennagray.png');
>> imgRoberts = rgb2gray(imgRoberts);
```

```
>> imgRoberts = edge(imgRoberts, 'Roberts');
>> imshow(imgRoberts);
```

- Canny :

```
>> imgCanny = imread('lennagray.png');
>> imgCanny = edge(imgCanny, 'Canny');
>> imshow(imgCanny);
```

- Laplaciano:

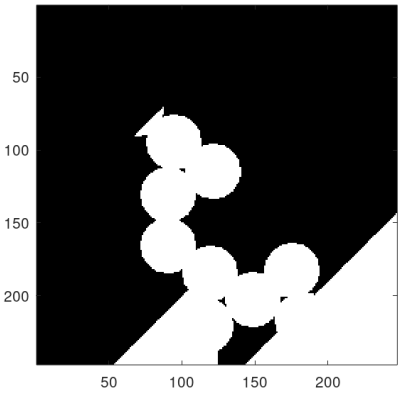
```
>> imgLaplaciano = imread('lennagray.png');
>> imgLaplaciano = edge(imgLaplaciano, 'log');
>> imshow(imgLaplaciano);
```

- Zero Crossiing:

2. Implementar a função que calcula o *threshold* global (ver slides da aula).

```
function threshold = thresholdGlobal(image)
  threshold    = 127;
  thresholdOld = 0;

  while abs (threshold - thresholdOld) > 0.1
    grup1 = image(image <= threshold);
    grup2 = image(image > threshold);

    average1 = mean(grup1);
    average2 = mean(grup2);

    thresholdOld = threshold;

    threshold = (average1 + average2)/2;
  endwhile

endfunction
```

3. Crie uma versão da imagem cırculo com
   - Quanto menor o bloco, melhor o resultado.



4. Através do método de Otsu é possível encontrar um “melhor” limiar para binarizar uma imagem. Carregue uma imagem e binarize ela utilizando a função *graythresh*().

```
function grayThresh()
  img = imread('lennagray.png');
  otsu = graythresh(img);
  imwrite(img > 255*otsu, 'otsu.png');

endfunction
```