



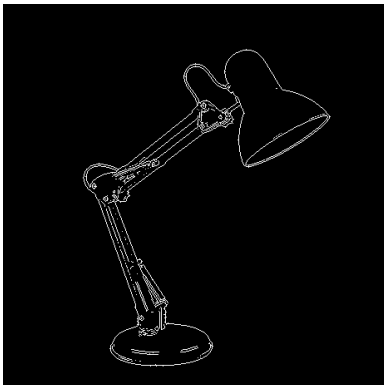
Lista Segmentação

Aluno: Felipe Augusto Vasconcelos e Silva
Matrícula: 1624358

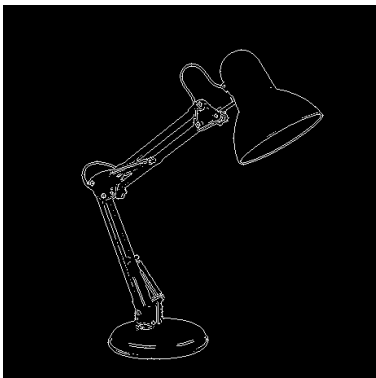
Questão 1) A função `edge` do matlab encontra as bordas de uma imagem. A função vem implementada com vários tipos de máscaras (Sobel, Prewitt, Roberts, Canny, Laplaciano, Zero Crossing). Carregar uma imagem e encontrar as bordas usando os diferentes filtros. Qual obtém melhor resultado?

```
img = imread('luminaria.jpg');  
img = rgb2gray(img);  
mascaras = {"Sobel"; "Prewitt"; "Roberts"; "Canny"; "log"};  
  
for i = 1 : size(mascaras)(1)  
    nimg = edge(img,mascaras{i});  
    imwrite(nimg, strcat(mascaras{i}, '.png'));  
    figure; imshow(nimg);  
endfor
```

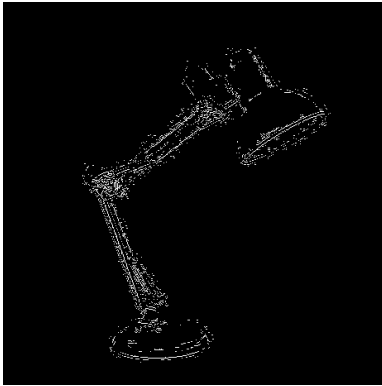
Sobel:



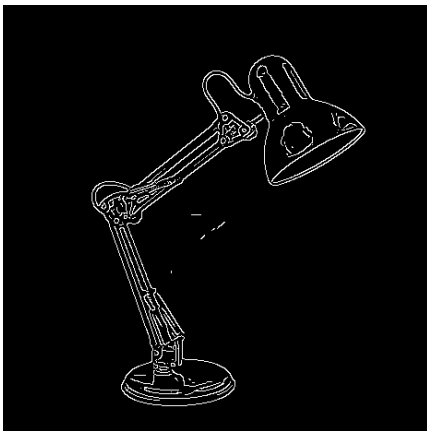
Prewitt:



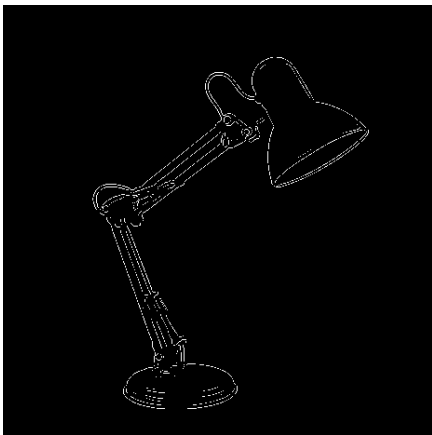
Roberts:



Canny:

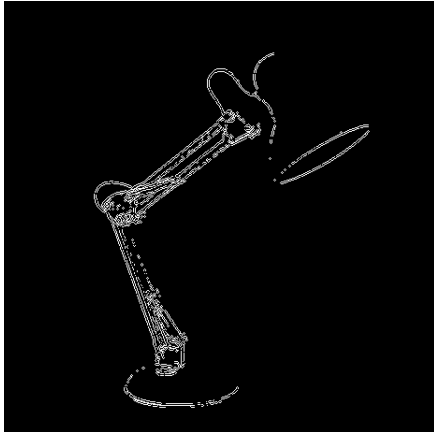


Laplaciano(log):



Zero Crossing:

```
nimg = edge(img,'zerocross', 0.5 , [0 -1 0; 1 4 -1; 0 -1 0]);  
imwrite(nimg, 'zerocross.png');  
figure; imshow(nimg);
```



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

```
function nimg = global_threshold(img, filter)
    [row, col] = size(img);
    mean_t = filter;
    last_mean_t = 0;
    nimg = zeros(row, col);
    while mean_t > last_mean_t
        g1 = 0;
        size_g1 = 0;
        g2 = 0;
        size_g2 = 0;
        for i = 1 : row
            for j = 1 : col
                if img(i,j) <= mean_t
                    g2 += img(i,j);
                    size_g2 += 1;
                else
                    g1 += img(i,j);
                    size_g1 += 1;
                    nimg(i,j) = 255;
                endif
            endfor
        endfor

        micro_1 = g1/size_g1;
        micro_2 = g2/size_g2;

        last_mean_t = mean_t
        mean_t = (micro_1 + micro_2)/2
    endwhile
```

Filter = 110



Questão 3) Crie uma versão da imagem círculo com

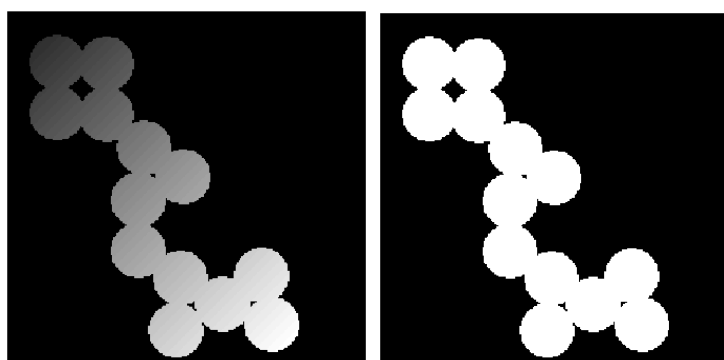
```
t=imread('coins.png');
t = rgb2gray(t); % transformar para escala de cinza
[row,col] = size(t);
[x,y]=meshgrid(1:row,1:col);
t2=double(t).*((x+y)/2+64)+x+y;
t3=uint8(255*mat2gray(t2));
```

Binarize a imagem t3 a fim de obter somente os círculos, use um filtro adaptativo. Qual tamanho de bloco produz um melhor resultado? Use a função threshold global implementada na questão 2.

```
t = imread('coins.png');
t = rgb2gray(double(t));
[row,col] = size(t);
[x,y]= meshgrid(1:row,1:col);
t2 = t.*((x+y)/2+64)+x+y ;
t3 = uint8(255*mat2gray(t2));
t3 = t.*t3;
figure; imshow(t3);

t3(1: 1 + row/3, 1:end) = global_threshold(t3(1: 1 + row/3, 1:end), 10);
t3(floor(row/3): floor(2*row/3), 1:end) = global_threshold(t3(floor(row/3): floor(2*row/3), 1:end), 30);
t3(floor(2*row/3)+1:end, 1:end) = global_threshold(t3(floor(2*row/3)+1:end, 1:end), 10);

figure; imshow(t3);
```



Questão 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()`.

```
img = imread('luminaria.jpg');  
t = graythresh(img);  
bw = im2bw(img,t);  
figure; imshow(bw);
```

