

---

# BIOENGENHARIA - LABORATORIO 02

## Table of Contents

Ex. 1. ....	1
Ex. 2. ....	5
3. ....	8
4. ....	13

Aluno: Pedro Henrique Garcia Macedo R.A.: 1829696

## Ex. 1.

```
load("p_4_1.mat")
I = y;
figure(1)
imshow(y);

% Ex. 1a.
Sh = [ -1 -2 -1; 0 0 0; 1 2 1];
Sv = [ -1 0 1; -2 0 2; -1 0 1];

figure(2)
Ih = imfilter(y, Sh);
imshow(Ih)
title('Sobel Horizontal')

% Ex. 1b.
figure(3)
Iv = imfilter(y, Sv);
imshow(Iv)
title('Sobel Vertical')

% Comparação
J = edge(y, 'Sobel', [], 'horizontal');
figure(4)
imshow(J)
title('Sobel Horizontal Edge')

J = edge(y, 'sobel', [], 'vertical');
figure(5)
imshow(J)
title('Sobel Vertical Edge')

% Ex. 1c.
figure(6)
imshow(I)
title('Imagem original')
J = edge(I, 'log');
figure(7)
```

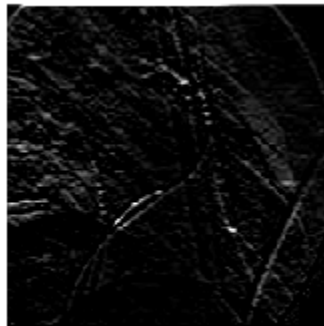
```
imshow(J)
title('Filtro Laplaciana da Gaussiana')

% Ex. 1d.
figure(8)
imshow(I)
title('Imagem original')
J = edge(I, 'canny');
figure(9)
imshow(J)
title('Canny')

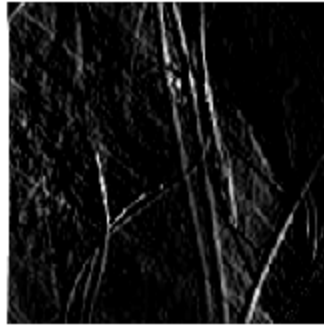
% Ex 1e.
% O método canny apresenta melhores resultados para extrair as bordas
% da imagem.
```



**Sobel Horizontal**



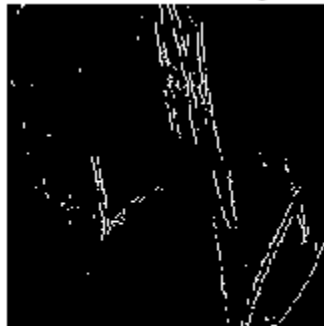
**Sobel Vertical**



**Sobel Horizontal Edge**



**Sobel Vertical Edge**



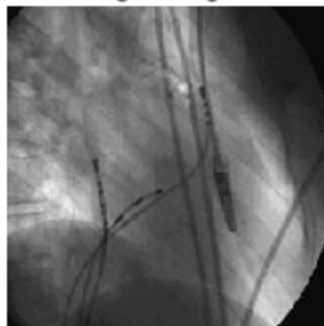
**Imagem original**

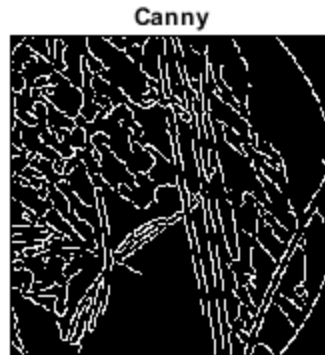


**Filtro Laplaciana da Gaussiana**



**Imagem original**





## Ex. 2.

```
clear; clc;
load("p_4_2.mat")
I = y;
figure(1)
imshow(I);
title('Imagem Original')

Hh = [-1 -1 -1; 2 2 2; -1 -1 -1];
hld = imfilter(I, Hh);
figure(2)
imshow(hld);
title('Máscara Horizontal')

Hv = [-1 2 -1; -1 2 -1; -1 2 -1];
vld = imfilter(I, Hv);
figure(3)
imshow(vld);
title('Máscara Vertical')

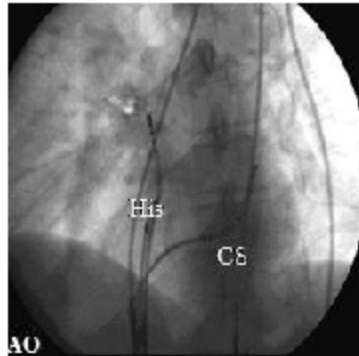
H45 = [2 -1 -1; -1 2 -1; -1 -1 2];
l45d = imfilter(I, H45);
figure(4)
imshow(l45d);
title('Máscara 45°')

Hn45 = [-1 -1 2; -1 2 -1; 2 -1 -1];
ln45d = imfilter(I, Hn45);
figure(5)
imshow(ln45d);
title('Máscara -45°')

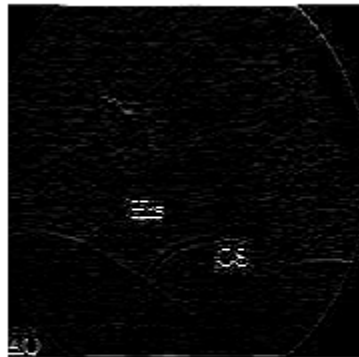
% 2e.
% As máscaras de linha detectam objetos na mesma orientação do filtro.
```

% A máscara na vertical destaca o catéter e portanto é a mais apropriada.

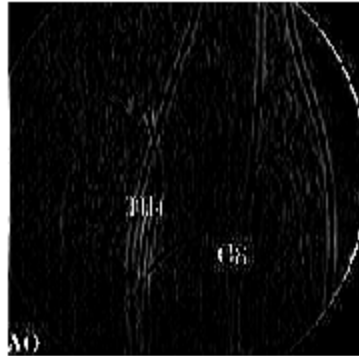
**Imagem Original**



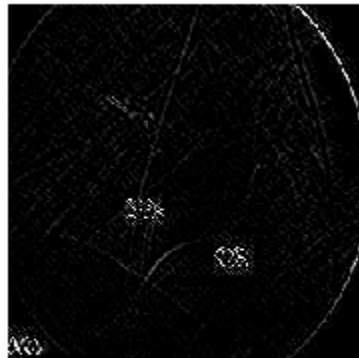
**Máscara Horizontal**



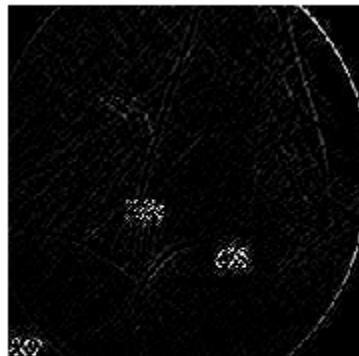
**Máscara Vertical**



**Máscara 45°**



**Máscara -45°**



### 3.

```
clear; clc;
load('p_4_4.mat');
I = x;
figure(1)
imshow(I);
title('Imagem Original')

Hh = [-1 -1 -1; 2 2 2; -1 -1 -1];
hld = imfilter(I, Hh);
figure(2)
imshow(hld);
title('Máscara Horizontal')

Hv = [-1 2 -1; -1 2 -1; -1 2 -1];
vld = imfilter(I, Hv);
figure(3)
imshow(vld);
title('Máscara Vertical')

H45 = [2 -1 -1; -1 2 -1; -1 -1 2];
l45d = imfilter(I, H45);
figure(4)
imshow(l45d);
title('Máscara 45°')

Hn45 = [-1 -1 2; -1 2 -1; 2 -1 -1];
ln45d = imfilter(I, Hn45);
figure(5)
imshow(ln45d);
title('Máscara -45°')

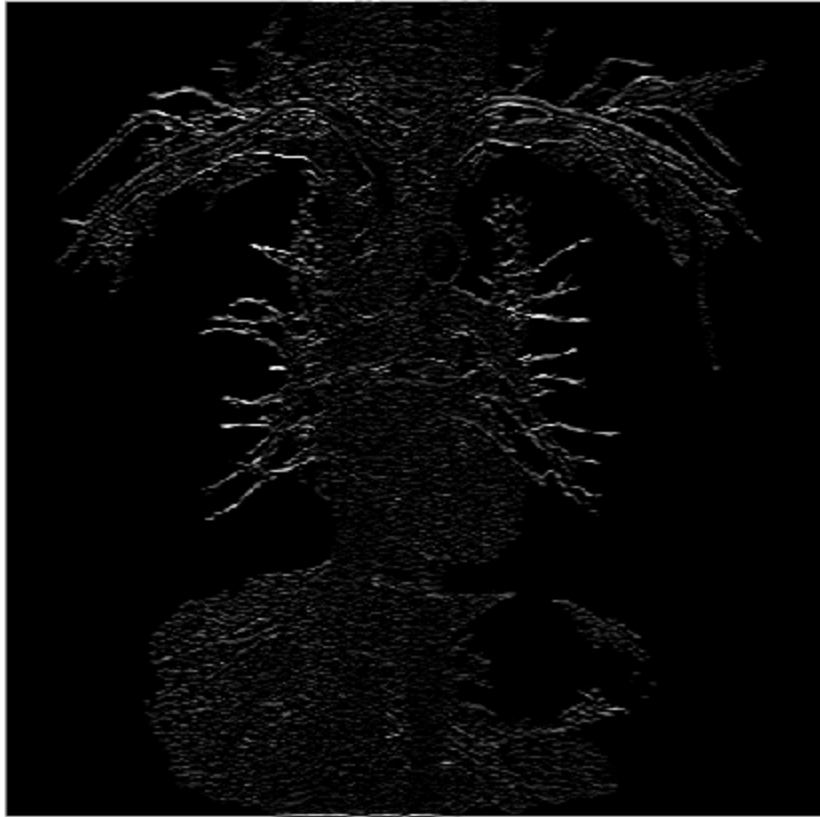
% O filtro horizontal detecta características na horizontal, e o
% vertical
% na vertical. O filtro de 45° evidencia muito mais as bordas.
```



**Imagem Original**



**Máscara Horizontal**



**Máscara Vertical**



**Máscara 45°**



Máscara -45°



4.

```
clear; clc;
load('p_4_5.mat');
x = rgb2gray(x);
nova = x;

figure(1)
imshow(x)
title("Imagem original")

figure(2)
A = imread("pixels_marcados", "png");
image(A);

% pixels selecionados utilizando instrumento olhometro
p1 = x(248, 260);
p2 = x(306, 345);
p3 = x(156, 435);
```

```
p4 = x(279, 468);
p5 = x(183, 619);
p6 = x(301, 776);
p7 = x(339, 613);
p8 = x(511, 405);

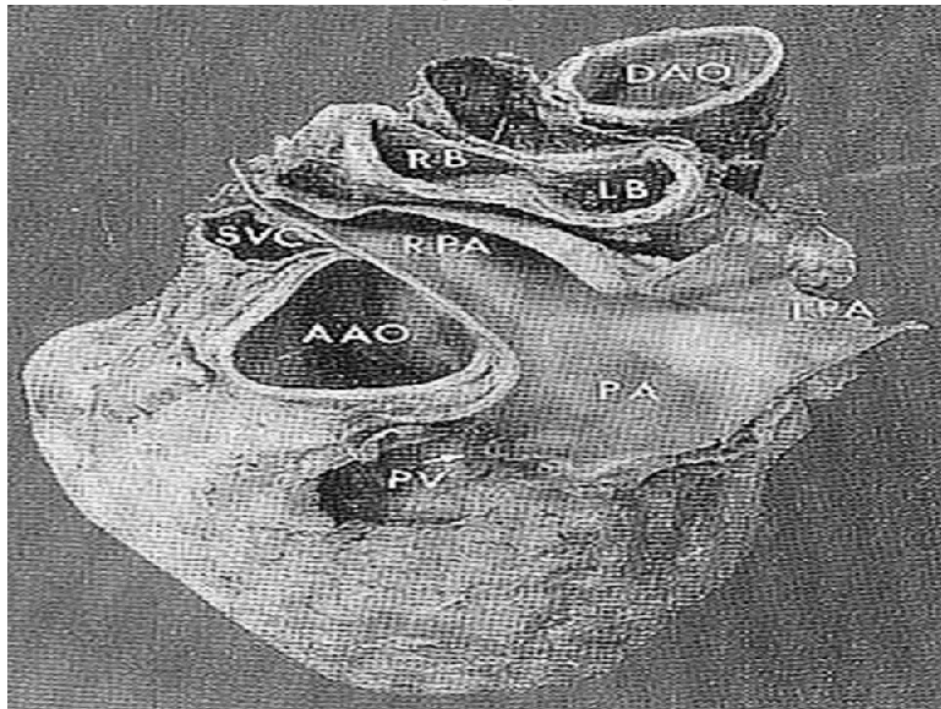
T = 15;
[l, c] = size(x);

% percorre a imagem
% compara com o threshold T;
% se a diferença for menor que o threshold, o pixel é preto
% senão, é branco
for ll=1:l
    for cc=1:c
        if abs(x(ll,cc) - p1) < T
            nova(ll,cc) = 255;
        elseif abs(x(ll,cc) - p2) < T
            nova(ll,cc) = 255;
        elseif abs(x(ll,cc) - p3) < T
            nova(ll,cc) = 255;
        elseif abs(x(ll,cc) - p4) < T
            nova(ll,cc) = 255;
        elseif abs(x(ll,cc) - p5) < T
            nova(ll,cc) = 255;
        elseif abs(x(ll,cc) - p6) < T
            nova(ll,cc) = 255;
        elseif abs(x(ll,cc) - p7) < T
            nova(ll,cc) = 255;
        elseif abs(x(ll,cc) - p8) < T
            nova(ll,cc) = 255;
        else
            nova(ll,cc) = 0;
        end
    end
end

figure(3)
imshow(nova)
title("Imagem segmentada")

% Com um valor adequado, podemos segmentar diferentes regiões da
% imagem
% que possuam níveis de cinza similar.
```

Imagem original



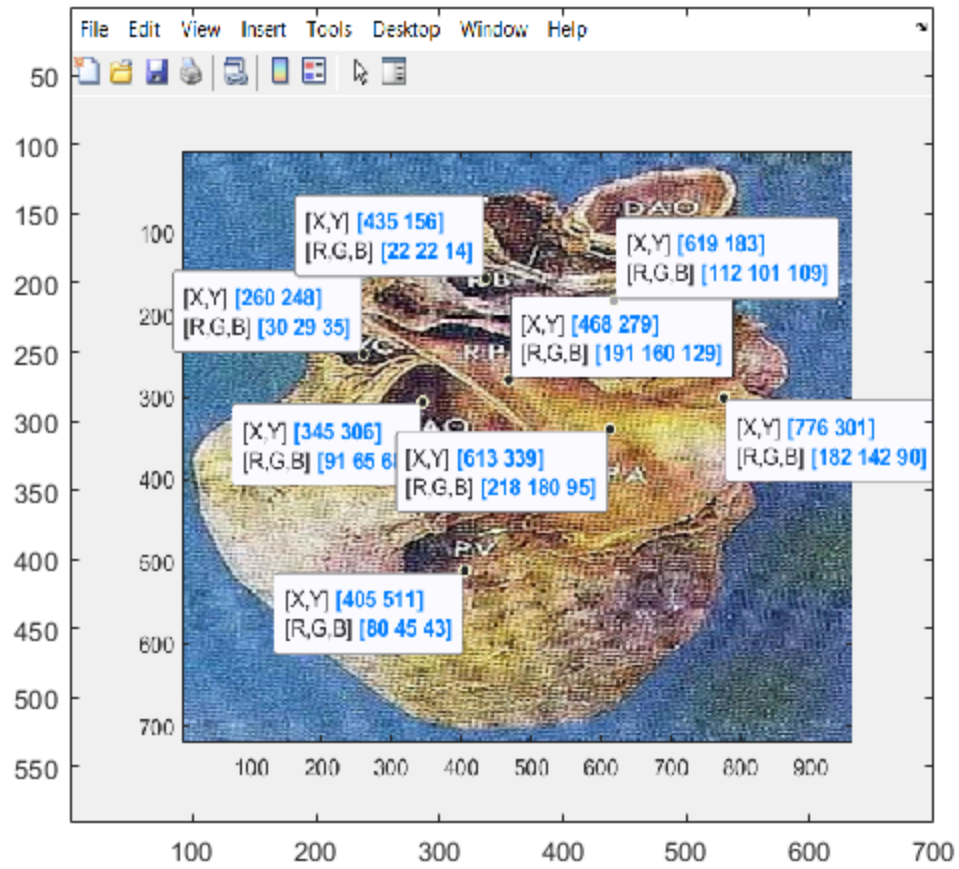
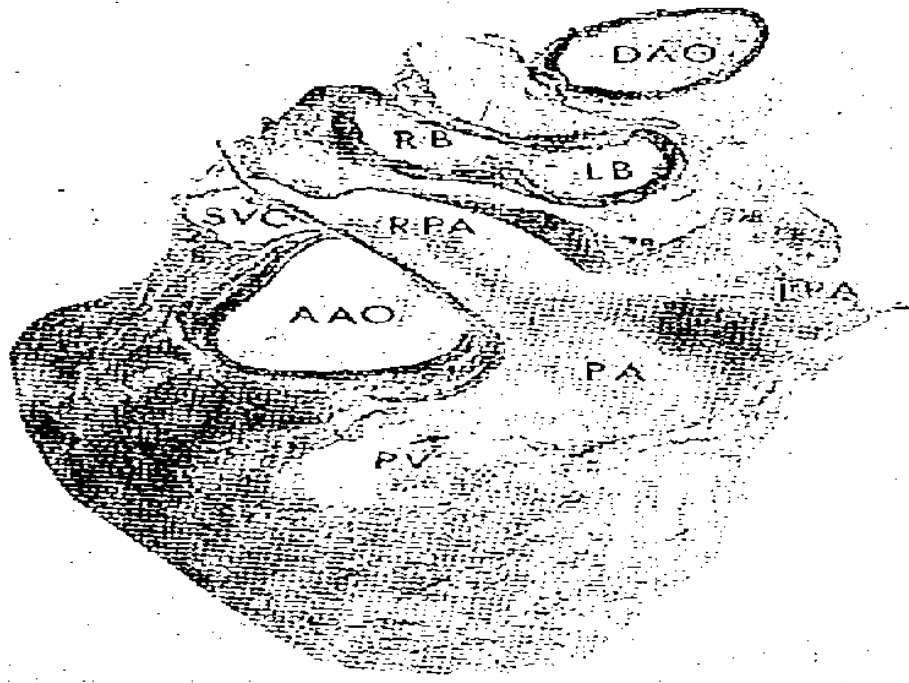




Imagem segmentada



*Published with MATLAB® R2018b*