

30/03/21

Maxine Annel Pacheco Ramírez A01551933

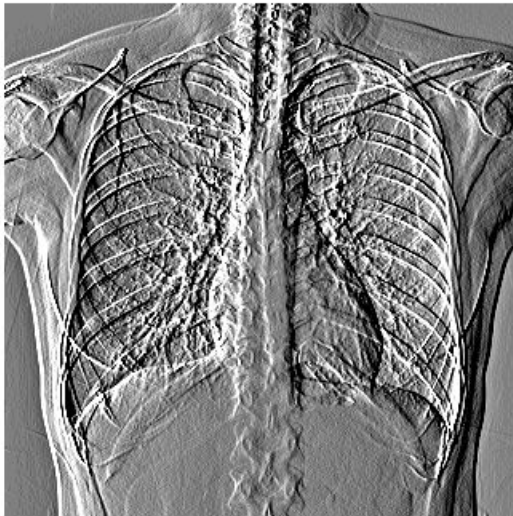
```
f=imread('radiograph1.jpg');  
f=imresize(f,0.25);  
f=double(f(:,:,1));  
imshow(f,[])
```



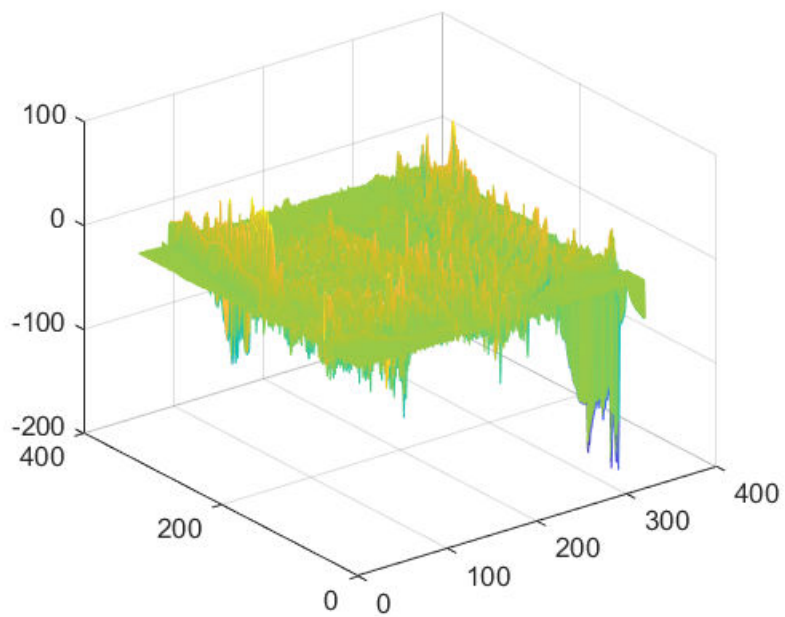
```
edgex=[1,-1] %AQUI HACEMOS CONVOLUCION
```

```
edgex = 1x2  
      1  -1
```

```
%edgex=[-1,1] %Derivadas al revés. Lo negativo es positivo AQUI HACEMOS  
%CORRELACION  
g1=conv2(f,edgex,'same');  
imshow(g1,[-10,10]);
```



```
mesh(g1)
```

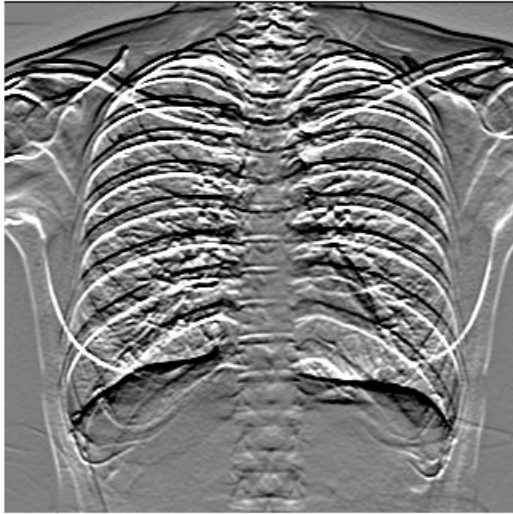


```
%Gris representa 0  
%brillante-> oscuro = negativo
```

```
edgey=[-1 -2 -1;0,0,0;1,2,1]/8 %EN Y FILTRO DE SOVEN
```

```
edgey = 3x3  
-0.1250 -0.2500 -0.1250  
0 0 0  
0.1250 0.2500 0.1250
```

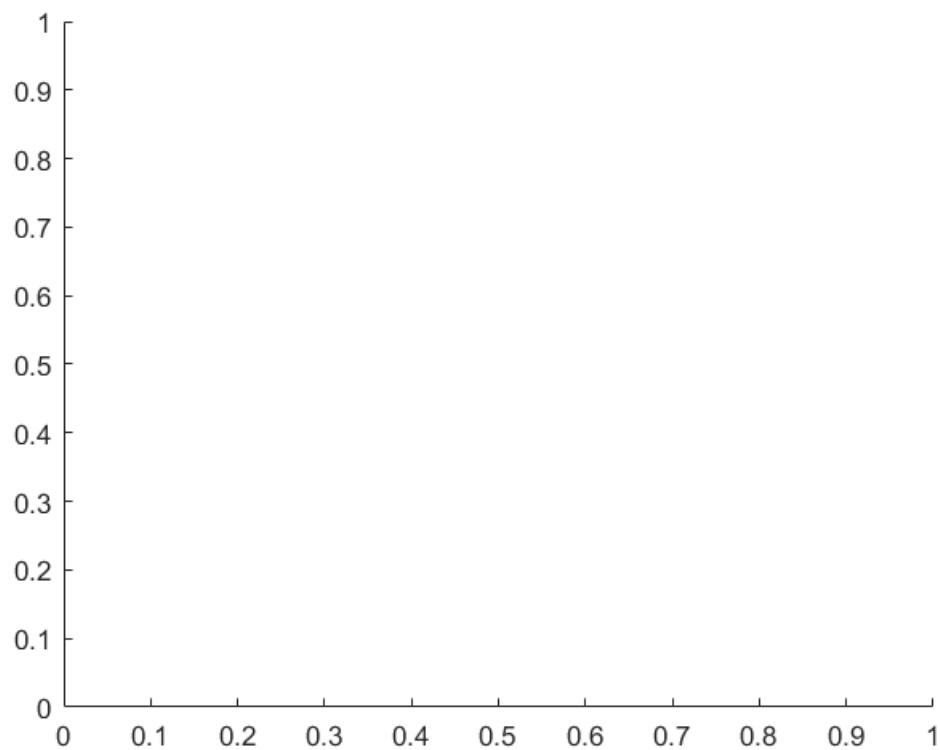
```
% edgey=[-1 -2 -1;0,0,0;1,2,1] %LE QUITE EL 8, Sin corregir por ganancia, esta mucho más fuerte
g2=conv2(f,edgey,'same');
imshow(g2,[-10,10])
```



```
%Costillas, se ven mejor en la y
%en la x se ve mejor el corazón
figure(2)
subplot(1,2,1)
imshow(g1,[-10,10])
subplot(1,2,2)
imshow(g2,[-10,10])
```



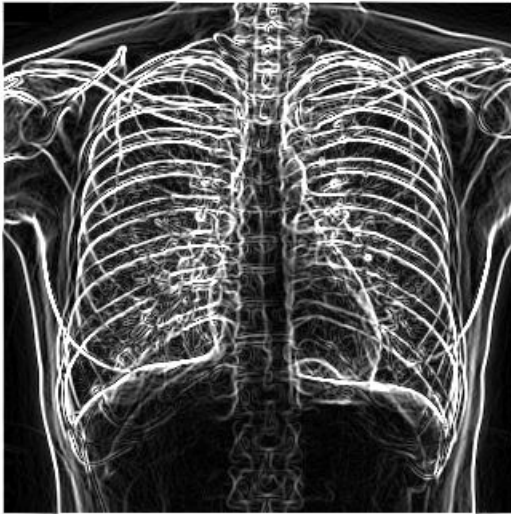
```
figure(3)  
subplot(1,1,1)
```



```
edgex=[1,0,-1;2,0,-2;1,0,-1]/8 %derividad en x con sove
```

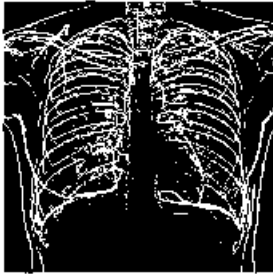
```
edgex = 3x3
    0.1250    0    -0.1250
    0.2500    0    -0.2500
    0.1250    0    -0.1250
```

```
gx=conv2(f,edgex,'same'); %correcciones de imagen
gy=conv2(f,edgey,'same');
% mag=(abs(gx)+abs(gy))/2; %magnitud del gradiente entre dos
mag=sqrt((gx).^2+(gy).^2); %este es el mas correcto
%mag=(abs(gx)+abs(gy)); %magnitud del gradiente entre dos
imshow(mag,[0,20]);
```



%interpretacion isotropica, se ve der en x y en y.
%Se puede calcular el ruido en la imagen con una máscara

```
noisemask = [-1, 0 1]; %MASCARA  
noiseimage = conv2(f,noisemask,'same'); %Convolucion con mascara para tener diferencia entre p  
noisevariance = mean2(noiseimage.^2); %varianza sacando la media  
noisestd = sqrt(noisevariance/2);  
edgedetection1 = mag > noisestd;% orilla con un umbral  
edgedetection2 = mag > 2*noisestd; %otra con otro estándar  
subplot(1,2,1)  
imshow(edgedetection1,[]);  
subplot(1,2,2)  
imshow(edgedetection2,[]);
```



```
%Con deconv, muchos falsos positivos
```

```
%Dos desviaciones estándar, falsos positivos disminuyen.
```

```
%Al calcular la magnitud se debe dividir entre 2 pq el ruido que se calcula
```

```
%esta en funcion
```

```
%como se tienen dos gradientes, el de x y el de y. si es muy malo es candidato a orilla y esto
```

```
%Esto es para mantener la misma escala
```

```
%Al dividir entre 2 el ruido estimado de igual manera toma en cuenta las
```

```
%orillas. El ruido que tenemos 1.5
```

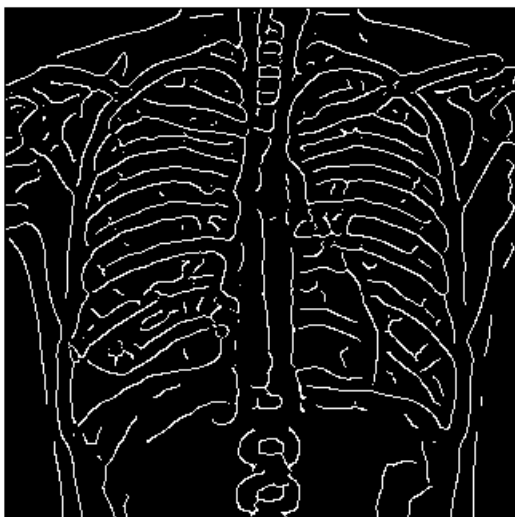
```
figure(4)  
subplot(1,1,1)  
angle=atan2(gy,gx);  
imshow(angle,[]);
```



%tenemos la imagen de los angulos, y los valores que se ven están en
%radianes, hay radianes positivos, negativos y desde 0 a -2π .

edgcany=edge(f, 'Canny',0.1,3); %DETECCION POR CANNY se junta como se tiene por gradiente
%Depende de los parámetros, las orillas que tiene CANNY

imshow(edgcany,[]);

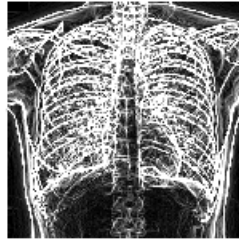
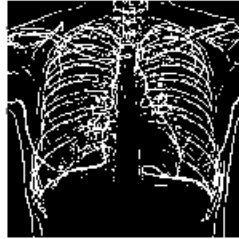


subplot(2,2,1)


```

imshow(edgcany,[])
subplot(2,2,2)
imshow( edgedetection1,[])
subplot(2,2,3)
imshow( edgedetection2,[])
subplot(2,2,4)
imshow(mag,[0,10])

```



%mascaras de convolucion veo derivadas, magnitudes, convoluciones. en cany
 %orillas conectadas.
 %CANY mejores algoritmos para detectar orillas. Orillas fuertes [aveces hay
 %rompimientos]