Table of Contents

E6

```
im = imread('rabbit.jpg');
imshow(im)
lap = [0,1,0;1,-4,1;0,1,0];
```

Laplacià

```
im = double(im);
im_lap=imfilter(im, lap);
figure,imshow(im_lap,[]),title('laplacià')
```

imatge binària

```
zero_crossing = zeros(size(im_lap));
for i = 2:size(im_lap, 1)-1
    for j = 2:size(im_lap, 2)-1
        neighbors = [im_lap(i-1, j) im_lap(i, j-1) im_lap(i, j+1) im_lap(i+1,
j)];
        if any(neighbors > 0) && any(neighbors < 0)
        zero_crossing(i, j) = 1;
        end
    end
end

figure,imshow(zero_crossing,[]),title('zero crossing')
```

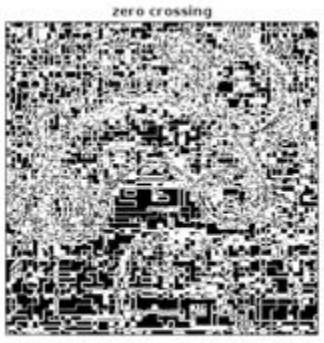
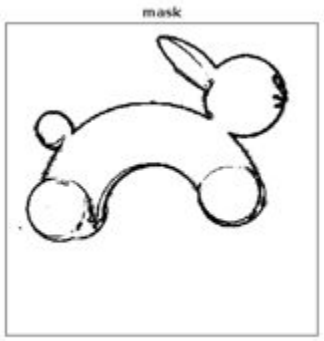
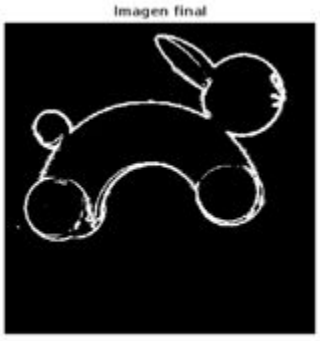


Image Final

Calcular el gradient de Sobel

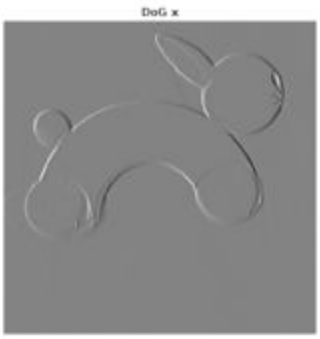
```
Sy = fspecial('sobel')/4;
Sx = Sy';
im = double(im);
imYSobel = imfilter(im, Sy);
imXSobel = imfilter(im, Sx);
mod= sqrt(imXSobel.^2+imYSobel.^2);
mask = (mod<15);
figure,imshow(mask,[]),title('mask')

% Obtener las dos mascaras, la primera para reforzar el imagen con  ruido.
% La segunda mascara filtra los ruidos no deseados obteniendo solo el
% contorno definido.
mask_2= mask ~= 1;
enforce_image = (mask.*zero_crossing) ~= 1;
final_img = mask_2 .* enforce_image;

figure,imshow(final_img,[]),title('Imagen final')
```





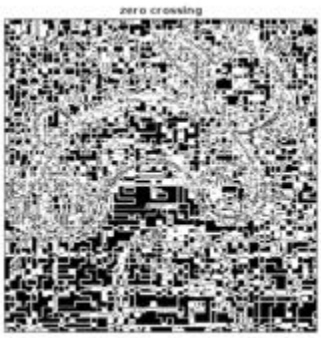
Derivada Gausiana

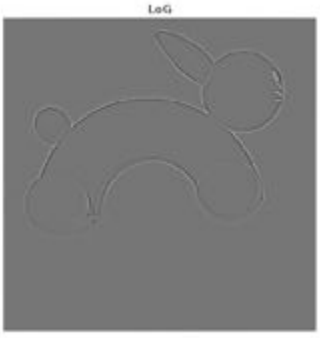
```
aux = fspecial('gaussian', 5, 1);
dog=aux(1:3,:)-aux(3:5,:);
im_dogy = imfilter(im, dog);
figure,imshow(im_dogy,[]),title('DoG y')
im_dogx = imfilter(im, dog');
figure,imshow(im_dogx,[]),title('DoG x')
```



Derivada Gausiana Laplacià

```
w = fspecial('log',5);
im_log = imfilter(im,w);
figure,imshow(im_log,[]),title('LoG')
```





Canny

```
res = edge(im,'canny',[0.25,0.9],4);
figure,imshow(res,[]),title('Canny')

im_gaus = imfilter(im,aux);
figure,imshow(im_gaus,[]),title('filtrat gausia')
figure,imshow(im-im_gaus,[]),title('residuo')
```

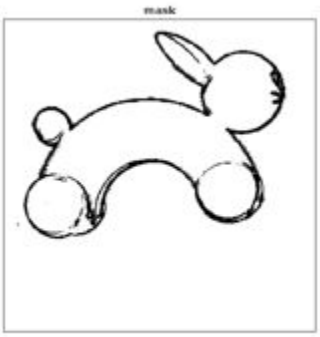
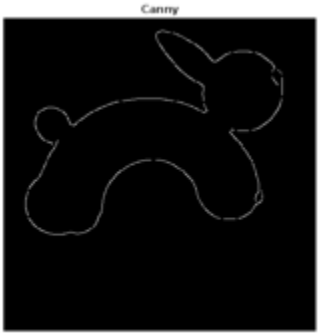

mask


Imagen final


Canny


filtrat gaussia

Published with MATLAB® R2023b