
MARTI RAMON ROS, AINA GARCIA ESPRIU

Table of Contents

Filtres passa-baixos	1
Filtres passa-alts (realçat)	4
Exercici final	8

Filtres passa-baixos

```
w = ones(7);      % kernel de conv
w = w/49;         % matriu modul 1

im = imread('gull.tif');
im2 = imnoise(im, 'gaussian'); % afegir soroll gaussia
figure, imshow(im2), title('soroll gaussia');

resMean = imfilter(im2, w, 'conv'); % aplicar un filtre mitjana soroll
gauss
figure, imshow(resMean), title('filtre mitjana');

w2 = fspecial('gaussian', 7, 2); %Crea matriu amb distribucio
gaussiana
resGauss = imfilter(im2, w2, 'conv'); % aplicar filtre gaussia soroll
gauss
figure, imshow(resGauss), title('filtre gaussia soroll gaussia');

im3 = imnoise(im, 'salt & pepper', 0.2); % Soroll salt and pepper
figure, imshow(im3), title('soroll impulsional');

resGauss2 = imfilter(im3, w2, 'conv'); % Aplicar filtre gaussia soroll
s&p
figure, imshow(resGauss2), title('filtre gaussia soroll s&p');

resMed = medfilt2(im3, [5 5]); % Aplicar filtre mediana 5x5
figure, imshow(resMed), title('filtre mediana soroll s&p');
```

soroll gaussia



filtre mitjana



filtre gaussia soroll gaussia



soroll impulsional



filtre gaussià soroll s&p



filtre mediana soroll s&p



Fitlres passa-alts (realçat)

```
im = imread('rabbit.jpg');  
  
wx = [-1 0 1; -1 0 1; -1 0 1]; % filtre de prewitt  
wx = wx/2;  
  
wy = [-1 -1 -1; 0 0 0; 1 1 1];  
wy = wy/2;  
  
[x, y] = size(im);  
im2 = im;  
  
im = double(im);  
wx = double(wx);  
wy = double(wy);
```

```
for i = 2:x-1 %implementacio de la convolucio amb fors
    for j = 2:y-1
        im2(i,j) = abs(sum(sum(wx.*im(i-1:i+1, j-1:j+1)))));
    end
end

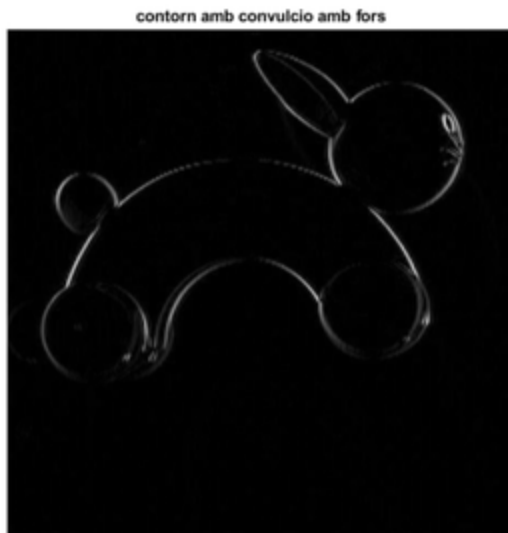
figure, imshow(im2, []), title('contorn amb convulcio amb fors');

Gx = imfilter(double(im), wx, 'conv');
figure, imshow(Gx, []), title('prewitt horitzontal');

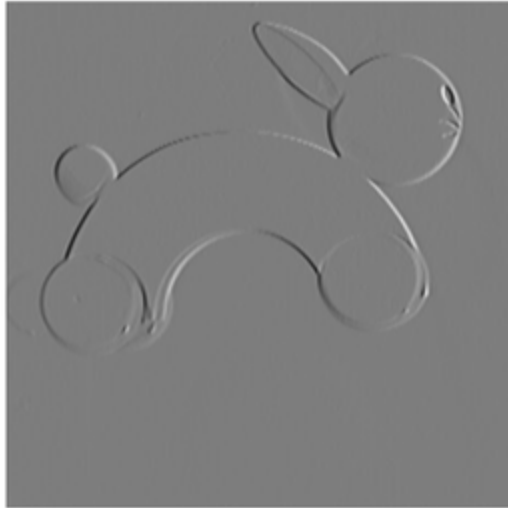
Gy = imfilter(double(im), wy, 'conv');
figure, imshow(Gy, []), title('prewitt vertical');

mod = sqrt(Gx.^2+Gy.^2);
mod = uint8(mod);
figure, imshow(mod), title('modul'); % pretwitt fusionat

alfa = atan2(Gy, Gx); % direccio
alfa = uint8((alfa + pi)*(255/2)/pi);
figure, imshow(alfa), title('orientacio'); % representacio de l'angle
del gradient
mask = mod<4; % mascara per eliminar el que tingui gradient <4
figure, imshow(mask), title('mascara');
alfa2 = alfa;
alfa2(mask)=0;
figure, imshow(alfa2), title ('orientacio amb mascara de gradient');
```



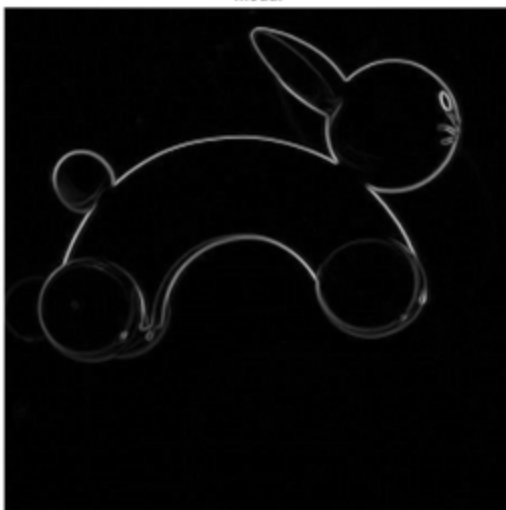
prewitt horizontal



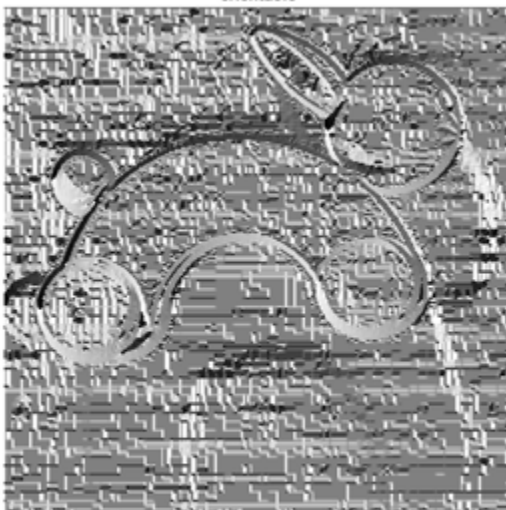
prewitt vertical

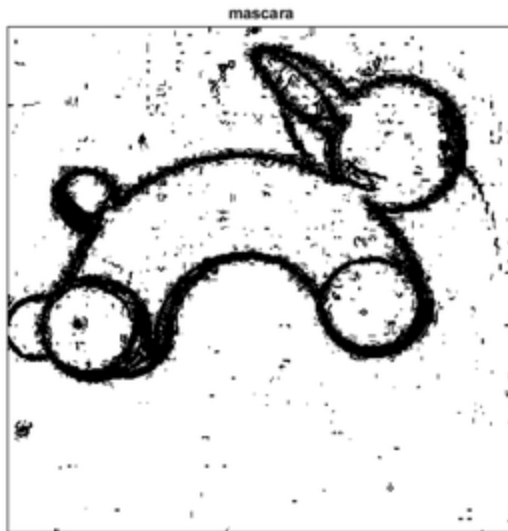


modul



orientacio





Exercici final

```
mod2 = imbinarize(mod); %imatge binaritzada :)  
imshow(mod2);
```




Published with MATLAB® R2018b