
Table of Contents

càlcul dels gradients	1
filtre especial	2
mòdul	3
direcció	5
laplacia	6

càlcul dels gradients

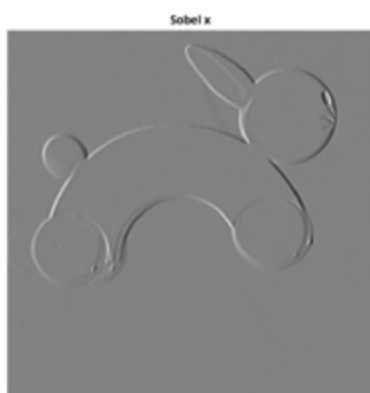
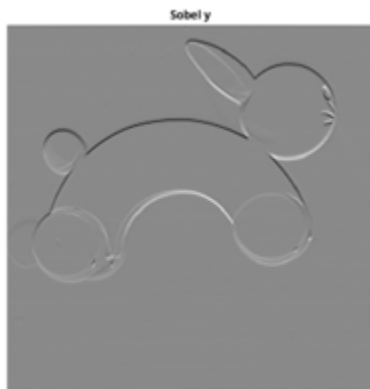
```
k = [-1, 0, 1];  
im = imread('rabbit.jpg');  
imshow(im);  
  
Gx = imfilter(im, k, 'conv');  
figure, imshow(Gx), title('Gradient x');  
  
im = double(im);  
Gx = imfilter(im, k, 'conv');  
figure, imshow(Gx, []), title('Gradient x');  
  
Gy = imfilter(im, k', 'conv');  
figure, imshow(Gy, []), title('Gradient y');
```





filtre especial

```
Sy = fspecial('sobel')/4;  
Gy = imfilter(im, Sy, 'conv');  
figure,imshow(Gy, []),title('Sobel y');  
  
Gx = imfilter(im, Sy', 'conv');  
figure,imshow(Gx, []),title('Sobel x');
```

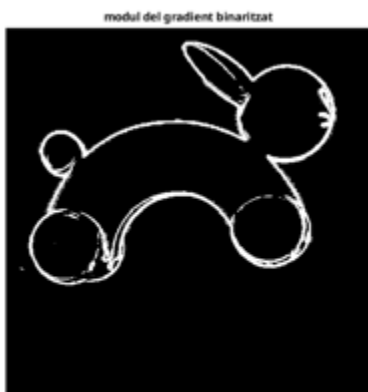
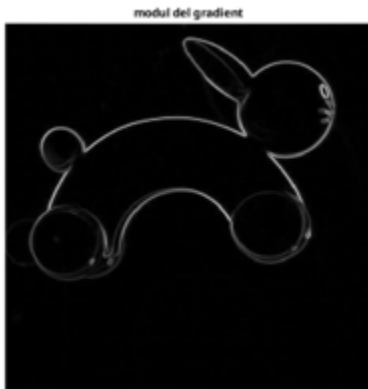


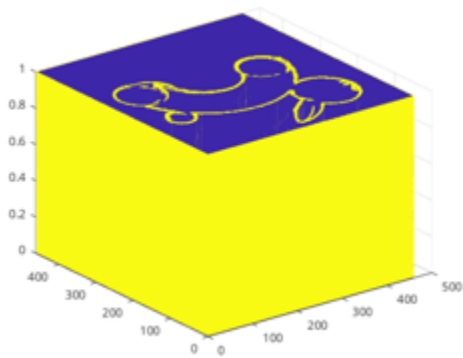
mòdul

```
mod = sqrt(Gx.^2+Gy.^2);  
figure,imshow(mod, []),title('modul del gradient');  
%figure, mesh(mod);  
  
% binaritzar la imatge del modul del gradient (trobar un llindar optim) per  
% trobar un contron fi i tancat (bwareaopen)
```

```
im_binaritzada = mod > 15;
figure,imshow(im_binaritzada, []),title('modul del gradient binaritzat');

% apliquem un minim de bwareaopen per eliminar soroll
im_neta = bwareaopen(im_binaritzada, 20);
figure,imshow(im_neta, []),title('modul del gradient binaritzat i net');
figure, mesh(im_neta);
```



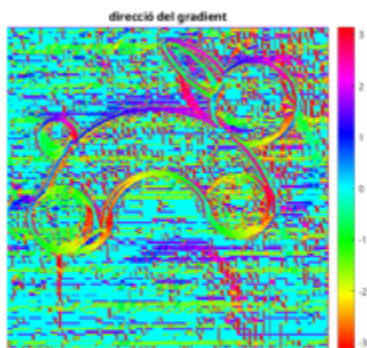


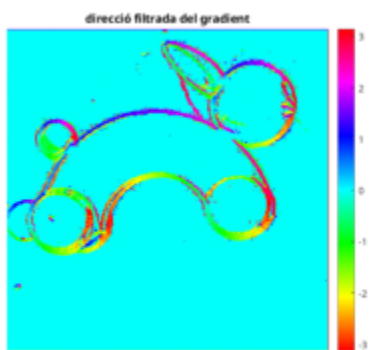
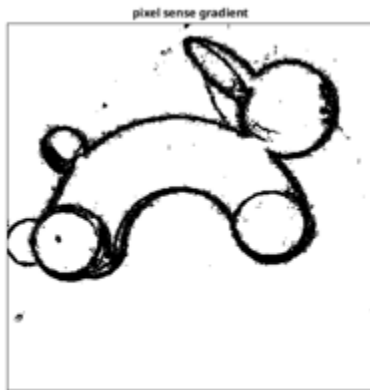
direcció

```
dir = atan2(Gy,Gx);
figure,imshow(dir, []),title('direcció del gradient');
colormap hsv, colorbar;

mask = mod < 4; %% threshold
figure, imshow(mask), title('pixel sense gradient');

dir2 = dir;
dir2(mask) = 0;
figure,imshow(dir2, []),title('direcció filtrada del gradient');
colormap hsv, colorbar;
```





laplacia

1. Aplicar el filtre Laplacianà a la imatge

```
k = fspecial('laplacian');
```

```
im(im_neta ~= 1) = 0;
```

```
lap = imfilter(im, k, 'conv');
```

% 2. Inicialitzar una imatge binària per als creuaments per zero

```
[files, columnes] = size(lap);
```

```
creuamentsPerZero = false(files, columnes);
```

% 3. Detectar els creuaments per zero a la imatge Laplaciana

```
for i = 2:files-1
```

```
    for j = 2:columnes-1
```

```
        % Examinar els veïns (8 connectats) per detectar canvis de signe
```

```
        veïns = [lap(i-1,j), lap(i+1,j), lap(i,j-1), lap(i,j+1), ...
```

```
                lap(i-1,j-1), lap(i-1,j+1), lap(i+1,j-1), lap(i+1,j+1)];
```

```
        % Si hi ha un canvi de signe entre el píxel actual i algun dels seus
```

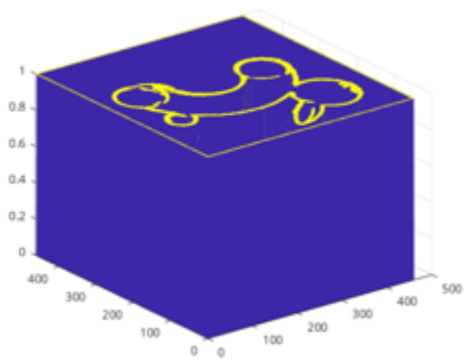
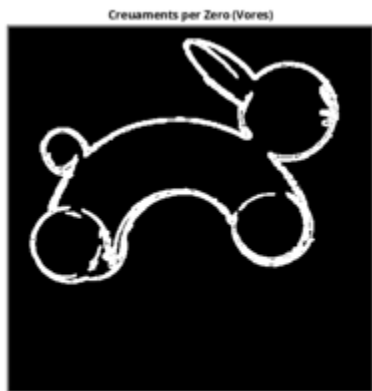
```
veïns
```

```
        if any(veïns .* lap(i,j) < 0)
```

```
            creuamentsPerZero(i,j) = true;
```

```
        end
    end
end

% 4. Mostrar la imatge amb els creuaments per zero
figure, imshow(creuamentsPerZero); title('Creuaments per Zero (Vores)');
figure, mesh(creuamentsPerZero);
```



Published with MATLAB® R2024b