

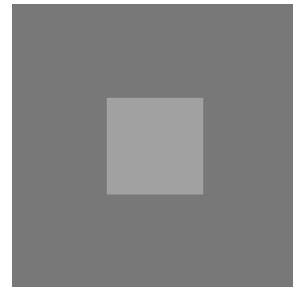
# Compte rendu du TP3

L'objectif de ce TP est faire la détection des contours dans une image en utilisant des différents filtres que ça soit **gradient**, **Roberts**, **Sobel** ou **Prewitt**.

## 1. Filtre gradient

- **Création d'une image 256 x 256 :**

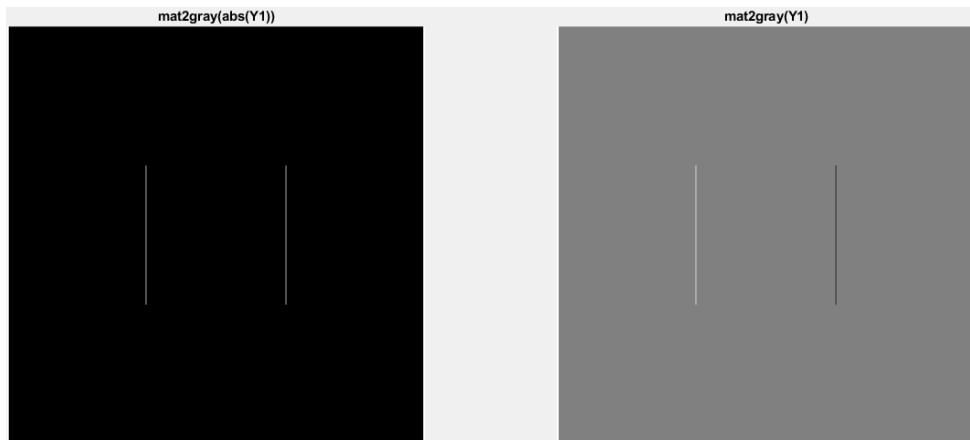
```
img = uint8(120*ones(256,256));
for I = 86:171
    for j = 86:171
        img(I,j,②)=161;
    end
end
```



Ce script permet de créer une image avec deux différents niveaux de gris : un carré gris clair sur un fond gris foncé.

- **Application du filtre h1 :**

```
h1=[1, -1];
Y1 = conv2(img,h1);
figure(2);
subplot(1,2,1); imshow(mat2gray(abs(Y1))); title("mat2gray(abs(Y1))");
subplot(1,2,2); imshow(mat2gray(Y1)); title("mat2gray(Y1)");
```



Comme nous remarquons sur la figure ci-dessus, le filtre h1 permet de faire la détection des contours verticaux en plus il est sensible aux transitions clair/foncé et aussi foncé/claire.

- **Application du filtre h2 :**

```
h2 =[1; -1];
Y2 = conv2(img,h2);
figure(3);
subplot(1,2,1); imshow(mat2gray(abs(Y2))); title("mat2gray(abs(Y2))");
subplot(1,2,2); imshow(mat2gray(Y2)); title("mat2gray(Y2)");
```



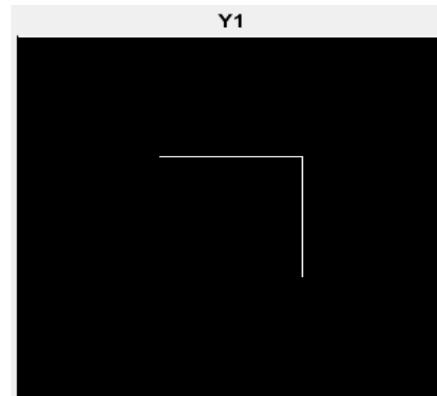
Le filtre **h2** permet de faire la détection des contours horizontaux et il est sensible aux différents types de transitions : clair/foncé et foncé/claire dans notre cas.

## 2. Méthode de Roberts

- Application du premier masque :

```
l1 = [0 1; -1 0];
l2 = [1 0; 0 -1];
Y1 = conv2(img,l1);
figure(3);
imshow(Y1);title ("Y1");
```

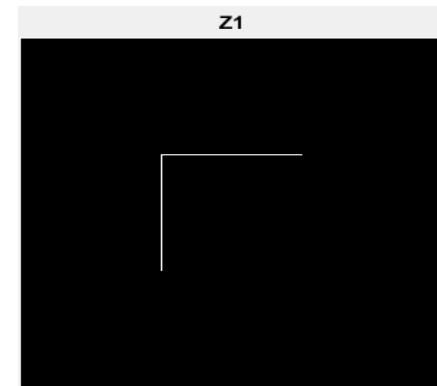
Le premier masque permet de détecter les contours dans la direction **Nord-Ouest, Sud-Est**.



- Application du deuxième masque :

```
Z1 = conv2(img, l2);
imshow(Z1); title("Z1");
```

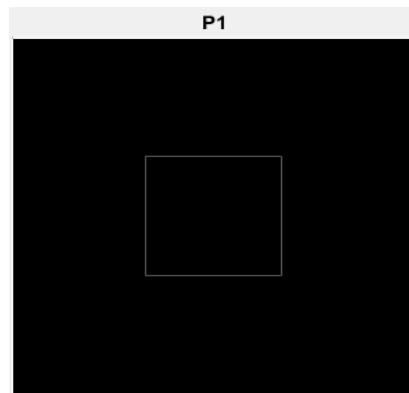
Le deuxième masque permet de détecter les contours dans la direction **Sud-Ouest, Nord-Est**.



- Calcul du module :

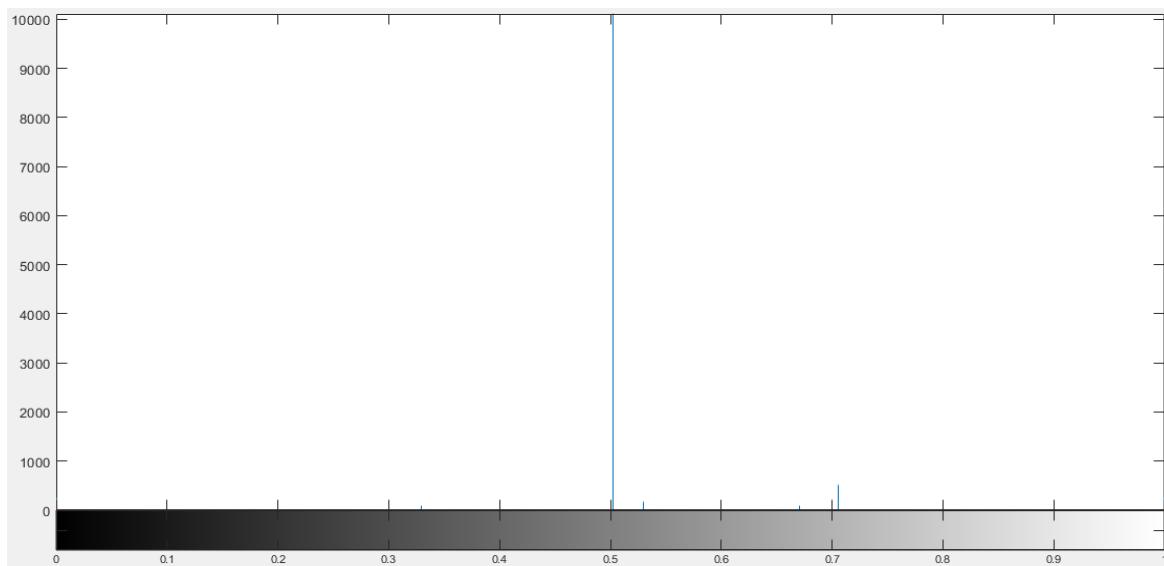
```
P1 = sqrt(Y1.^2 + Z1.^2);
P1 = mat2gray(P1);
imshow(P1); title("P1");
```

Le module permet de combiner les résultats obtenus par les deux masques.



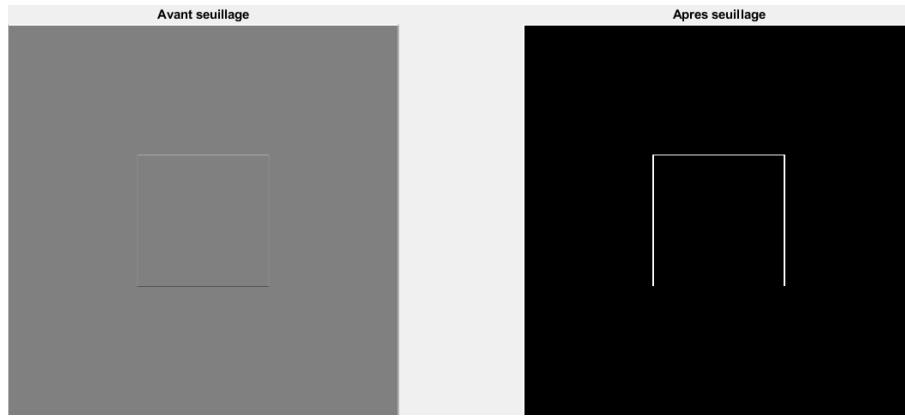
- **Application du Seuillage :**

La première chose à faire pour bien choisir le seuil c'est de visualiser de l'histogramme du module :



On constate que le seuil optimal du seuillage va être à peu près compris entre 0.5 et 0.7 : nous choisissons dans notre cas une valeur proche de 0.51 qui donne des résultats assez bien :

```
figure;
subplot(1,2,1); imshow(P1); title("Avant seuillage");
s = 0.512;
for i = 1:256
    for j = 1:256
        if (P1(i,j) >= s)
            P1(i,j) = 255;
        else
            P1(i,j) = 0;
        end
    end
end
subplot(1,2,2); imshow(P1); title("Après seuillage");
```



### 3. Méthode de Sobel

- Application des masques Gx, Gy :

```

h1 = [1 0 -1; 2 0 -2; 1 0 -1];
h2 = [1 2 1; 0 0 0; -1 -2 -1];
Gx = mat2gray(conv2(img,h1));
subplot(1,3,1); imshow(Gx); title("H1");

```

Le premier masque permet de détecter les contours dans la direction verticale.

- Application du deuxième masque :

```

Gy = mat2gray(conv2(img,h2));
subplot(1,3,2); imshow(Gy); title("H2");

```

Le deuxième masque permet de détecter les contours dans la direction horizontale.

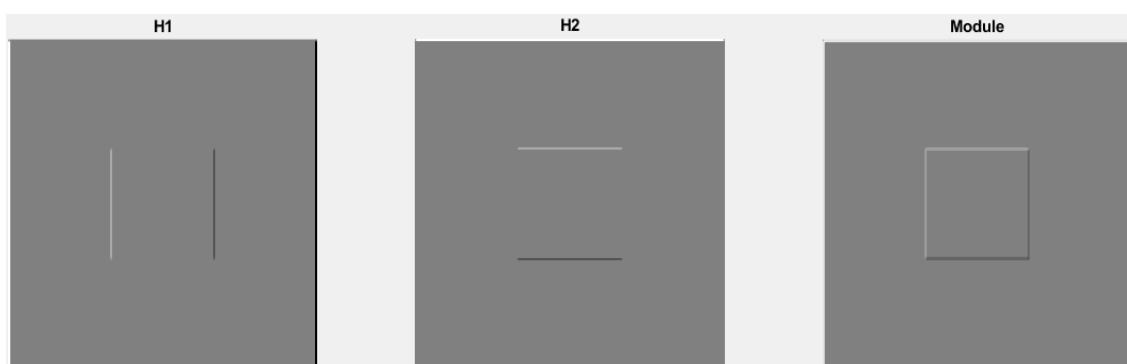
- Calcul du module :

```

P2 = mat2gray(sqrt(Gy.^2 + Gx.^2));
subplot(1,3,3); imshow(P2); title("Module");

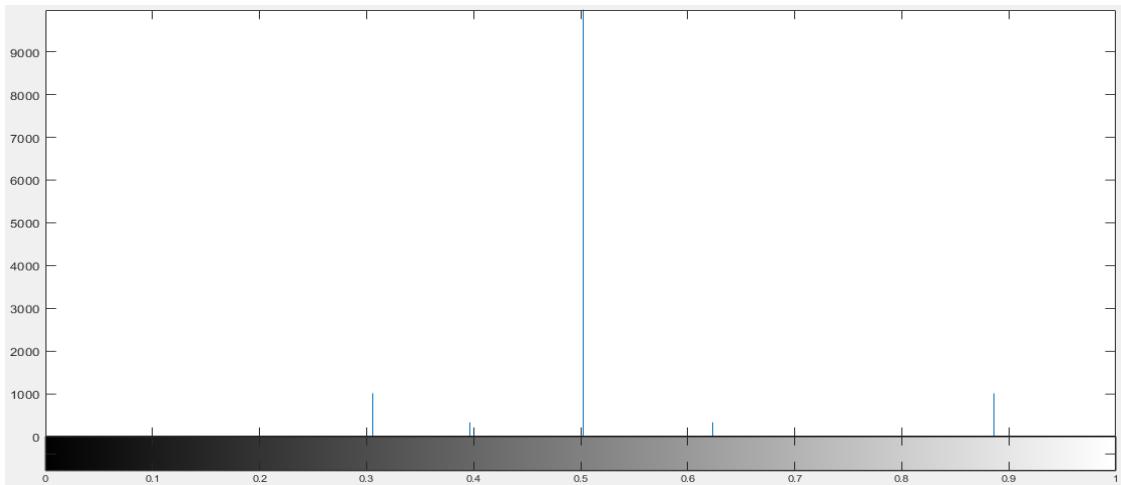
```

Le module permet de combiner les résultats obtenus par les deux masques : verticale et horizontale ce qui permet d'obtenir des contours carrés.



- Application du Seuillage :

Pareille à la section précédente, la première chose à faire pour bien choisir le seuil c'est de visualiser de l'histogramme du module :



On constate que le seuil optimal du seuillage va être à peu près compris entre 0.3 et 0.9 : nous choisissons dans notre cas une valeur proche de 0.57 qui donne des résultats meilleurs :

```
figure;
subplot(1,2,1); imshow(P2); title("Avant seuillage");

s=0.55;
for i = 1:256
    for j = 1:256
        if (P2(i,j) >= s)
            P2(i,j) = 255;
        else
            P2(i,j) = 0;
        end
    end
end

subplot(1,2,2); imshow(P2); title("Apres seuillage");
```



## 4. Méthode de Prewitt

- Application des masques  $G_x, G_y$  :

```
h1 = [1 0 -1; 1 0 -1; 1 0 -1];
h2 = [1 1 1; 0 0 0; -1 -1 -1];
Gx = mat2gray(conv2(img,h1));
subplot(1,3,1); imshow(Gx); title("H1");
```

Le premier masque permet de détecter les contours dans la direction verticale.

- **Application du deuxième masque :**

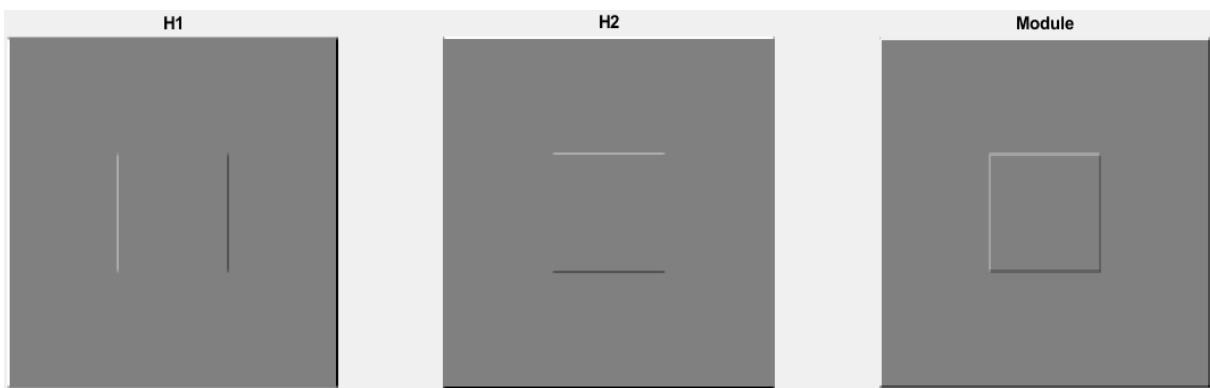
```
Gy = mat2gray(conv2(img,h2));
subplot(1,3,2); imshow(Gy); title("H2");
```

Le deuxième masque permet de détecter les contours dans la direction horizontale.

- **Calcul du module :**

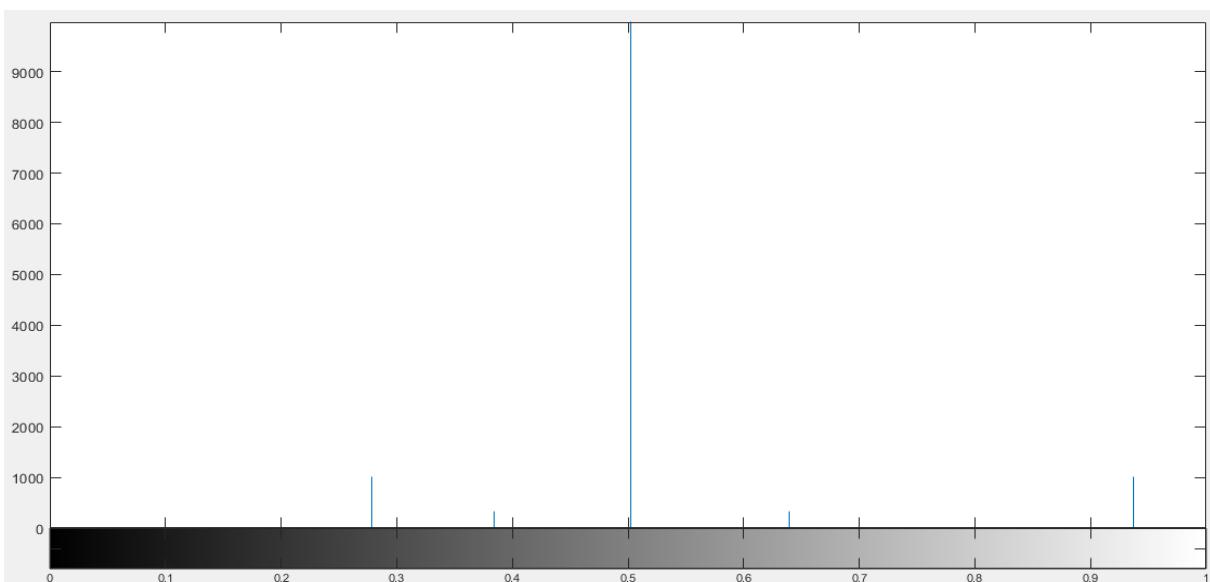
```
P2 = mat2gray(sqrt(Gy.^2 + Gx.^2));
subplot(1,3,3); imshow(P2); title("Module");
```

Le module permet de combiner les résultats obtenus par les deux masques : verticale et horizontale ce qui permet d'obtenir des contours carrés.



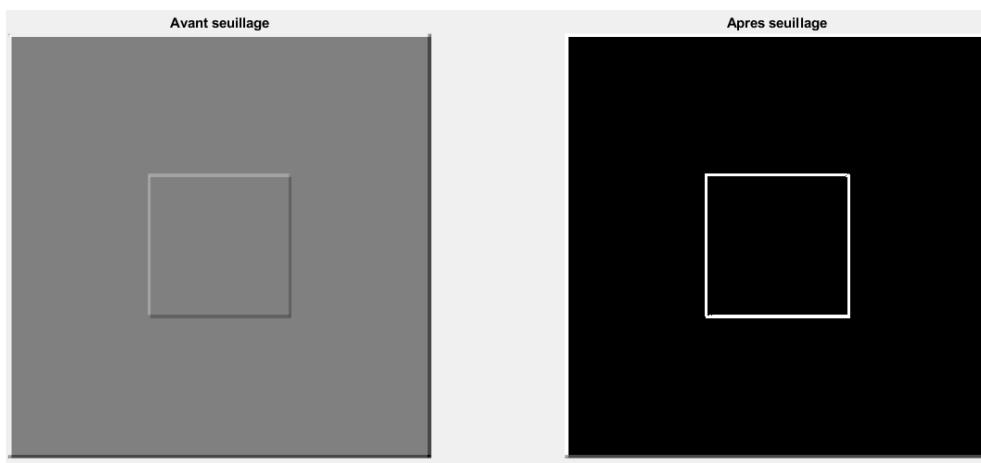
- **Application du Seuillage :**

Pareille à la section précédente, la première chose à faire pour bien choisir le seuil c'est de visualiser de l'histogramme du module :



On constate que le seuil optimal du seuillage va être à peu près compris entre 0.4 et 0.7 : nous choisissons dans notre cas une valeur proche de 0.54 qui donne des résultats meilleurs :

```
figure;
subplot(1,2,1); imshow(P2); title("Avant seuillage");
s=0.54;
for i = 1:256
    for j = 1:256
        if (P2(i,j) >= s)
            P2(i,j) = 255;
        else
            P2(i,j) = 0;
        end
    end
end
subplot(1,2,2); imshow(P2);title("Apres seuillage");
```



## Conclusion :

Au cours des différentes phases de la réalisation de ce TP, nous avons réussi à mettre en œuvre certains méthodes de détection des contours dans des images. Ces méthodes peuvent être utilisées dans d'autres tâches à savoir la détection d'objets dans une image et segmentation des images.