

# IV. Détection de contours

- Exemple :

```
>> frog=imread('.../frog.jpg') ;
```

```
>> I = rgb2gray(frog) ;
```

```
>> I = double(I)/255 ;
```

```
>> Ph = fspecial('prewitt')
```

```
>> Pv = -Ph'
```

```
>> lh = filter2(Ph, I) ;
```

```
>> lv = filter2(Pv, I) ;
```

```
>> lvh = sqrt(lv.^2+lh.^2) ;
```

```
>> figure, colormap(gray)
```

```
>> subplot(2,2,1), imagesc(lv), colorbar, title('Prewitt vertical'), axis('image')
```

```
>> subplot(2,2,2), imagesc(lh), colorbar, title('Prewitt horizontal'), axis('image')
```

```
>> subplot(2,2,3), imagesc(lvh), colorbar, title('Prewitt combiné'), axis('image')
```

```
>> seuil = 0.5 ;
```

(bouger le seuil : x2 ou /2...)

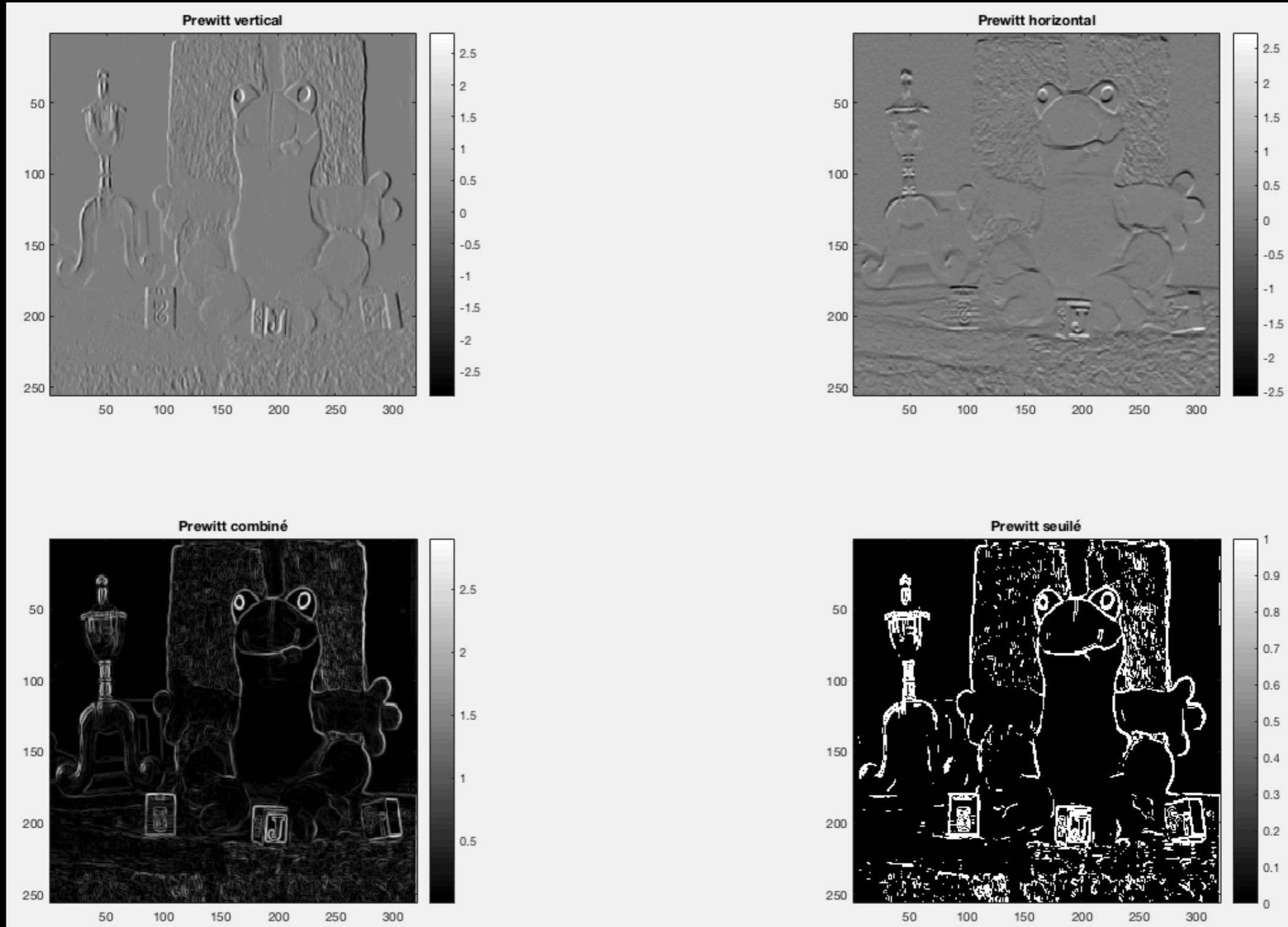
```
>> ls = (lvh > seuil) ;
```

```
>> subplot(2,2,4), imagesc(ls), colorbar, title('Prewitt seuillé'), axis('image')
```

# IV. Détection de contours

```
1 clear
2 close all
3 %pkg load image %Octave (pas Matlab)|
4
5 % image de la grenouille
6 img='/Users/marcel/Documents/MATLAB/GBM/0-images/frog.jpg';
7 frog=imread(img);
8 I=rgb2gray(frog);
9 I=double(I)/255.;
10
11 % filtre de Prewitt
12 Ph=fspecial('prewitt');
13 Pv=-Ph';
14 IPh=filter2(Ph,I);
15 IPv=filter2(Pv,I);
16 IPvh=sqrt(IPv.^2+IPh.^2);
17
18 figure, colormap('gray')
19 subplot(2,2,1), imagesc(IPv), colorbar, title('Prewitt vertical')
20 axis('image')
21 subplot(2,2,2), imagesc(IPh), colorbar, title('Prewitt horizontal')
22 axis('image')
23 subplot(2,2,3), imagesc(IPvh),colorbar, title('Prewitt combiné')
24 axis('image')
25
26 % seuillage
27 'min(IPvh)', min(min(IPvh))
28 'max(IPvh)', max(max(IPvh))
29 seuil=0.5;
30 IPs=(IPvh>seuil);
31
32 subplot(2,2,4), imagesc(IPs), colorbar, title('Prewitt seuillé')
33 axis('image')
```

# IV. Détection de contours



# IV. Détection de contours

## 3- FILTRES DE SOBEL

- Filtre horizontal  $S_h$  :

$$\begin{array}{cccc} & 1 & 2 & 1 \\ 1 & 1 & 2 & 1 \\ 0 & 0 & 0 & 0 \\ -1 & -1 & -2 & -1 \end{array} \quad \begin{array}{l} \\ : \text{variation des lignes mais avec une plus grande} \\ \text{pondération de la colonne centrale} \end{array}$$

- Filtre vertical  $S_v$  :

$$\begin{array}{cccc} & -1 & 0 & 1 \\ 1 & -1 & 0 & 1 \\ 2 & -2 & 0 & 2 \\ 1 & -1 & 0 & 1 \end{array} \quad \begin{array}{l} \\ : \text{variation des colonnes mais avec une plus grande} \\ \text{pondération de la ligne centrale} \end{array}$$

- sous MATLAB/OCTAVE :  $\gg Sh = fspecial('sobel');$   
 $\gg Sv = -Sh';$

# IV. Détection de contours

- **EXERCICE 1** : Comparer les effets respectifs des paires de filtres de Prewitt et de Sobel sur l'image de votre choix (p.ex. la grenouille). Considérer un seuil qui paraît adapté pour l'image filtrée par Prewitt (par ex. 0.659), puis ajuster le seuil de l'image filtrée par Sobel de façon à obtenir des contours équivalents.

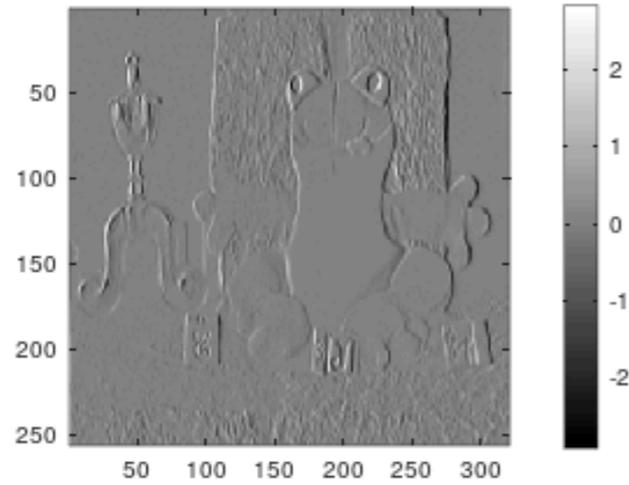
(Pouvait-on prévoir le seuil de l'image filtrée par les filtres de Sobel en considérant la dynamique sur les deux images de contours uniques ?...)

# IV. Détection de contours

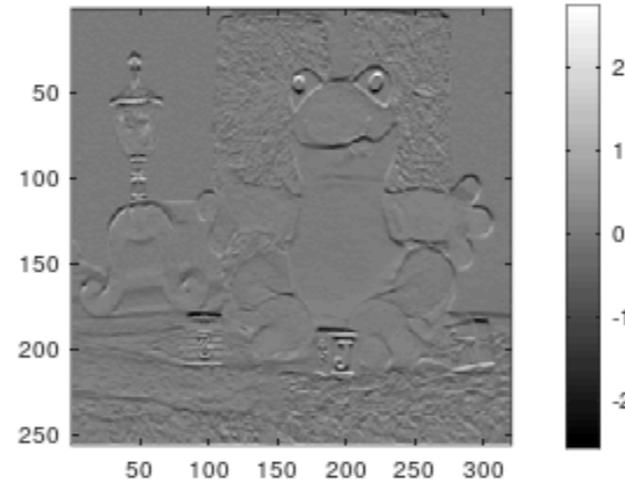
```
1 clear
2 close all
3 %pkg load image %Octave (pas Matlab)
4
5 % image de la grenouille
6 img='/Users/marcel/Documents/MATLAB/GBM/0-images/frog.jpg';
7 frog=imread(img);
8 I=rgb2gray(frog);
9 I=double(I)/255.;
10
11 % filtre de Prewitt
12 Ph=fspecial('prewitt');
13 Pv=-Ph';
14 IPh=filter2(Ph,I);
15 IPv=filter2(Pv,I);
16 IPvh=sqrt(IPv.^2+IPh.^2);
17
18 figure(1)
19 colormap('gray')
20 subplot(2,3,1), imagesc(IPv), colorbar, title('Prewitt vertical')
21 axis('square')
22 subplot(2,3,2), imagesc(IPh), colorbar, title('Prewitt horizontal')
23 axis('square')
24 subplot(2,3,3), imagesc(IPvh),colorbar, title('Prewitt complet')
25 axis('square')
26
27 % filtre de Sobel
28 Sh=fspecial('sobel');
29 Sv=-Sh';
30 ISh=filter2(Sh,I);
31 ISv=filter2(Sv,I);
32 ISvh=sqrt(ISv.^2+ISh.^2);
33
34 subplot(2,3,4), imagesc(ISv), colorbar, title('Sobel vertical')
35 axis('square')
36 subplot(2,3,5), imagesc(ISh), colorbar, title('Sobel horizontal')
37 axis('square')
38 subplot(2,3,6), imagesc(ISvh),colorbar, title('Sobel complet')
39 axis('square')
```

# IV. Détection de contours

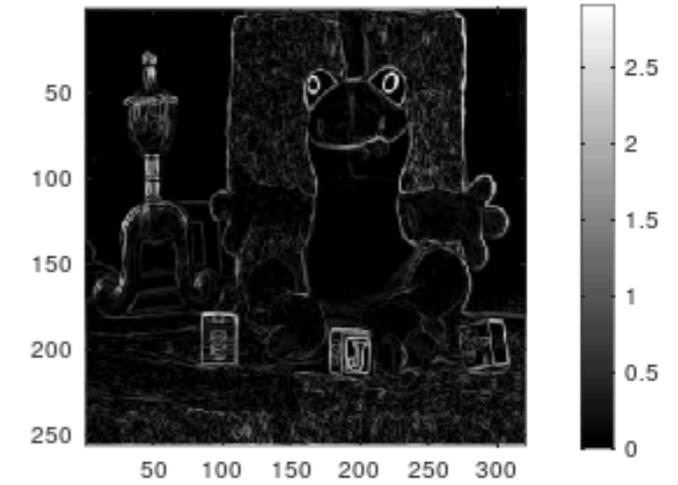
Prewitt vertical



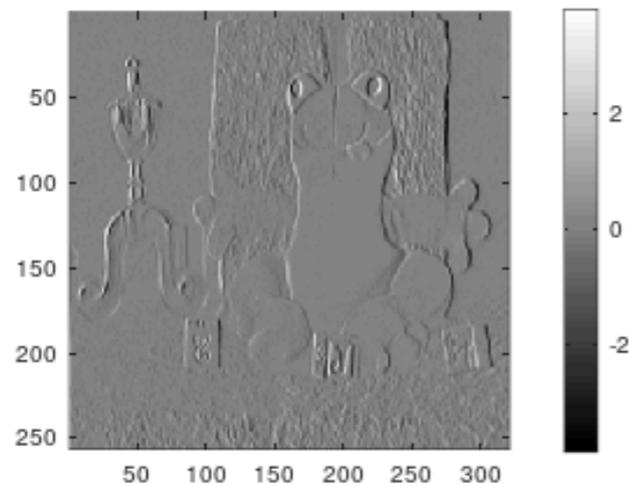
Prewitt horizontal



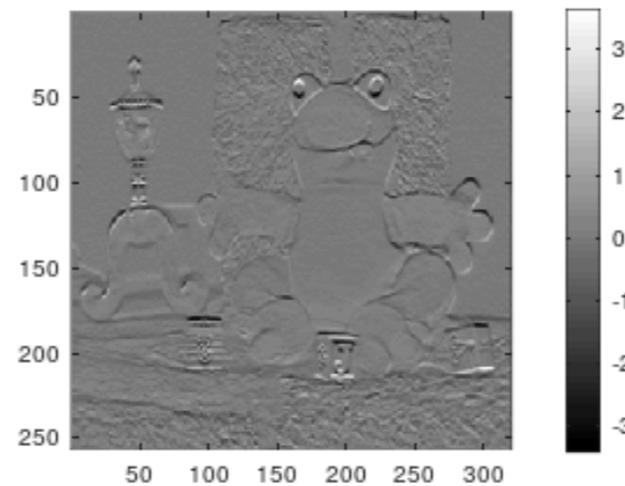
Prewitt complet



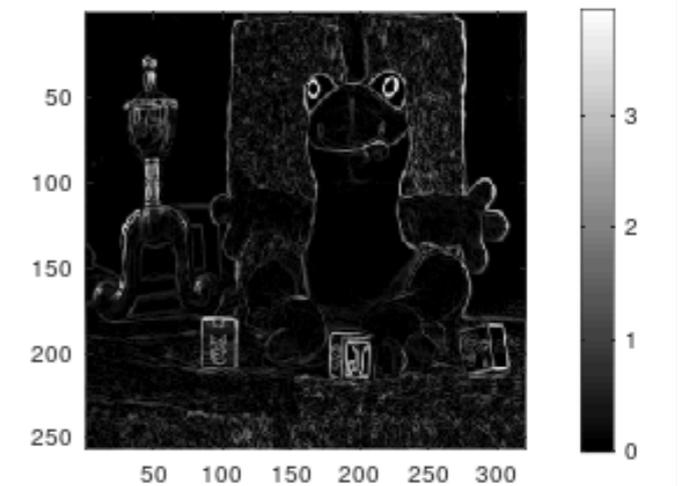
Sobel vertical



Sobel horizontal



Sobel complet

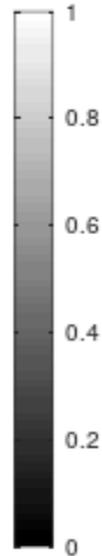


# IV. Détection de contours

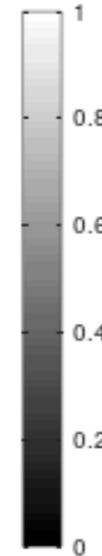
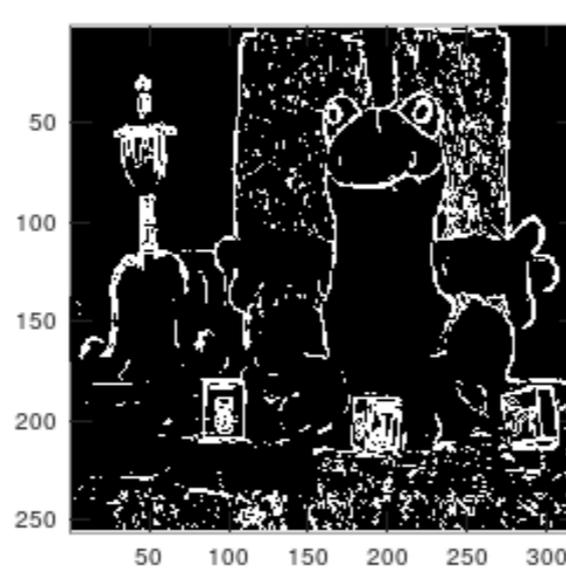
```
41 % seuillage Prewitt et comparaison avec ajustement seuil Sobel
42 seuil=0.659;
43 'seuil=', seuil
44 IPs=(IPvh>seuil);
45 ISs=(ISvh>seuil);
46
47 dynamique_ISvh = max(max(ISvh))-min(min(ISvh))
48 dynamique_IPvh = max(max(IPvh))-min(min(IPvh))
49 coeff = dynamique_ISvh/dynamique_IPvh
50 'rapport des dynamiques = ', coeff
51 seuil_Sobel=coeff*seuil;
52 'seuil adapté=', seuil_Sobel
53 ISs2=(ISvh>seuil_Sobel);
54
55 figure(2)
56 colormap('gray')
57 subplot(1,3,1), imagesc(IPs), colorbar, title('Prewitt avec seuil')
58 axis('square')
59 subplot(1,3,2), imagesc(ISs), colorbar, title('Sobel avec même seuil')
60 axis('square')
61 subplot(1,3,3), imagesc(ISs2), colorbar, title('Sobel avec seuil adapté')
62 axis('square')
```

# IV. Détection de contours

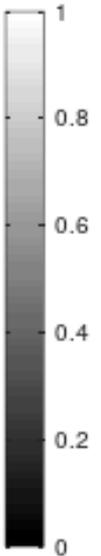
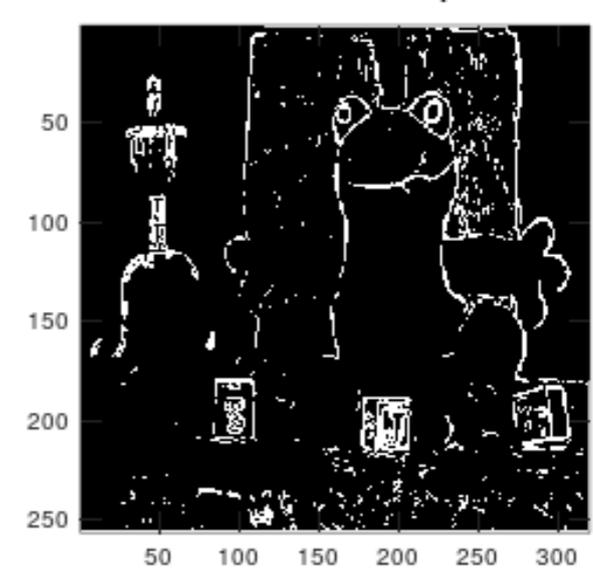
Prewitt avec seuil



Sobel avec même seuil



Sobel avec seuil adapté



```
>> exo1_Prewitt_Sobel  
  
ans = seuil=  
seuil = 0.6590  
dynamique_ISvh = 3.9417  
dynamique_IPvh = 2.9089  
coeff = 1.3550  
ans = rapport des dynamiques =  
coeff = 1.3550  
ans = seuil adapté=  
seuil_Sobel = 0.8930
```

# IV. Détection de contours

## 4- FILTRES DE ROBERTS

- Filtre diagonal a :

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} : \text{variation des anti-diagonales}$$

- Filtre anti-diagonal b :

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} : \text{variation des diagonales}$$

- sous MATLAB/OCTAVE : pas de FSPECIAL ici...

```
>> a = [1 0; 0 -1];  
>> b = [0 1; -1 0];
```

```
ou : >> b = rot90(a, -1);
```

# IV. Détection de contours

- **EXERCICE 2** : Comparer les effets respectifs de la paire de filtres de Roberts aux deux précédentes, avec même seuil puis avec des seuils adaptés. Commenter sur la différence entre le filtrage de Roberts et les deux autres, puis sur le changement de seuil en général. Pourquoi trouve-t-on que, grosso modo :
  - $\text{seuil}_{\text{Sobel}} \approx 4/3 \text{ seuil}_{\text{Prewitt}}$ ,
  - $\text{seuil}_{\text{Roberts}} \approx 1/3 \text{ seuil}_{\text{Prewitt}}$  ?

(Cette fois-ci avec, par exemple, l'image 'grillage.jpeg'...)

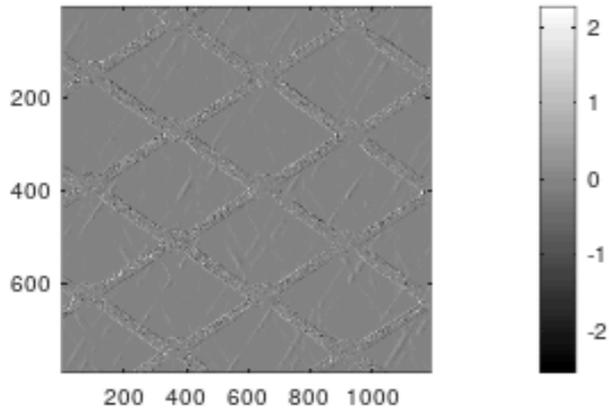
# IV. Détection de contours

```
1 clear
2 close all
3 pkg load image
4
5 % image de la grenouille
6 img='/Users/marcel/Documents/MATLAB/GBM/0-images/grillage.jpeg';
7 frog=imread(img);
8 I=rgb2gray(frog);
9 I=double(I)/255.;
10
11 % filtre de Prewitt
12 Ph=fspecial('prewitt');
13 Pv=-Ph';
14 IPh=filter2(Ph,I);
15 IPv=filter2(Pv,I);
16 IPvh=sqrt(IPv.^2+IPh.^2);
17
18 figure(1)
19 colormap('gray')
20 subplot(3,3,1), imagesc(IPv), colorbar, title('Prewitt vertical')
21 axis('square')
22 subplot(3,3,2), imagesc(IPh), colorbar, title('Prewitt horizontal')
23 axis('square')
24 subplot(3,3,3), imagesc(IPvh), colorbar, title('Prewitt complet')
25 axis('square')
26
27 % filtre de Sobel
28 Sh=fspecial('sobel');
29 Sv=-Sh';
30 ISh=filter2(Sh,I);
31 ISv=filter2(Sv,I);
32 ISvh=sqrt(ISv.^2+ISh.^2);
```

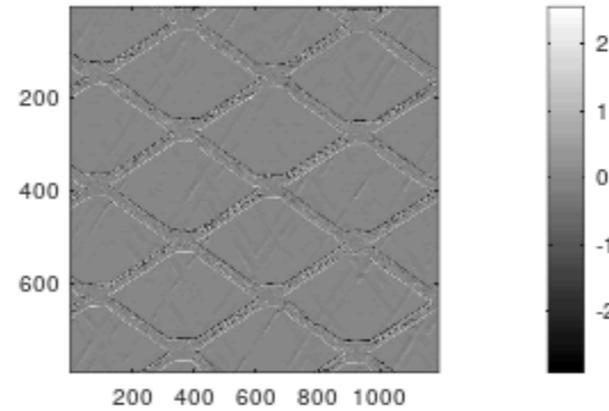
```
34 subplot(3,3,4), imagesc(ISv), colorbar, title('Sobel vertical')
35 axis('square')
36 subplot(3,3,5), imagesc(ISh), colorbar, title('Sobel horizontal')
37 axis('square')
38 subplot(3,3,6), imagesc(ISvh), colorbar, title('Sobel complet')
39 axis('square')
40
41 % filtres de Roberts
42 Ra=[1 0;0 -1];
43 Rb=[0 1;-1 0];
44 IRa=filter2(Ra,I);
45 IRb=filter2(Rb,I);
46 IRab=sqrt(IRa.^2+IRb.^2);
47
48 subplot(3,3,7), imagesc(IRa), colorbar, title('Roberts diagonal')
49 axis('square')
50 subplot(3,3,8), imagesc(IRb), colorbar, title('Roberts anti-diagonal')
51 axis('square')
52 subplot(3,3,9), imagesc(IRab), colorbar, title('Roberts complet')
53 axis('square')
```

# IV. Détection de contours

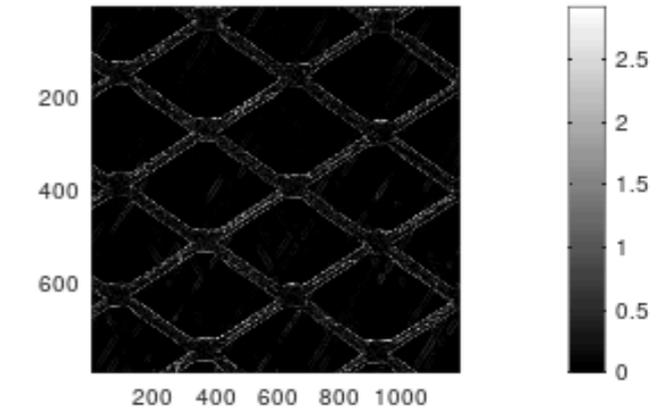
Prewitt vertical



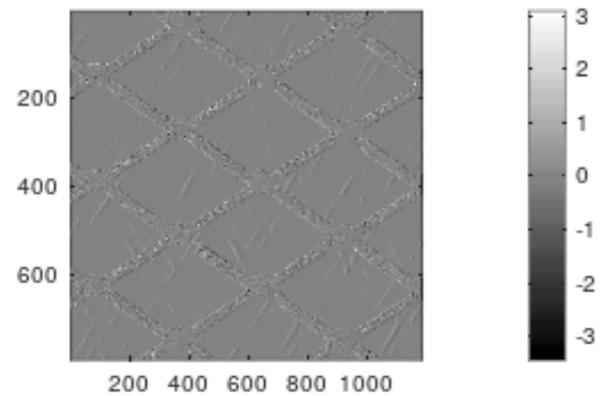
Prewitt horizontal



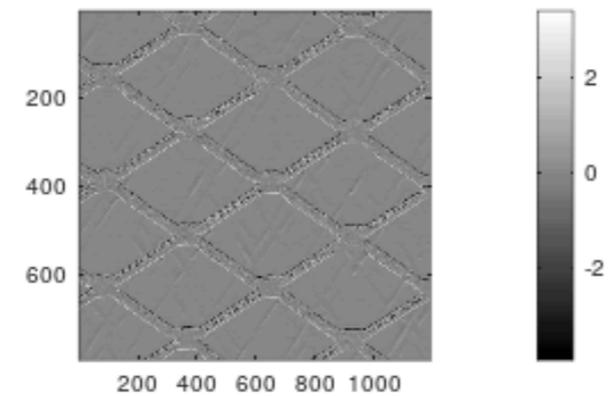
Prewitt complet



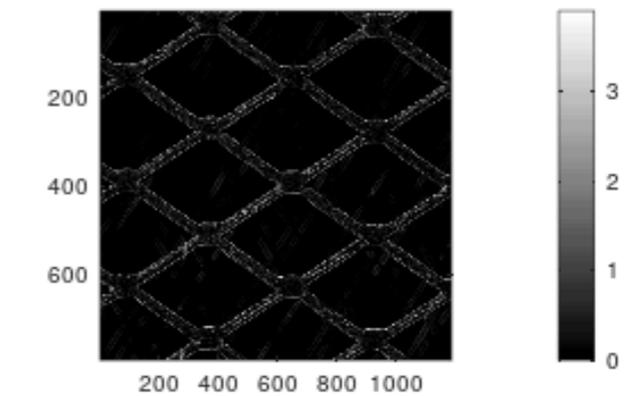
Sobel vertical



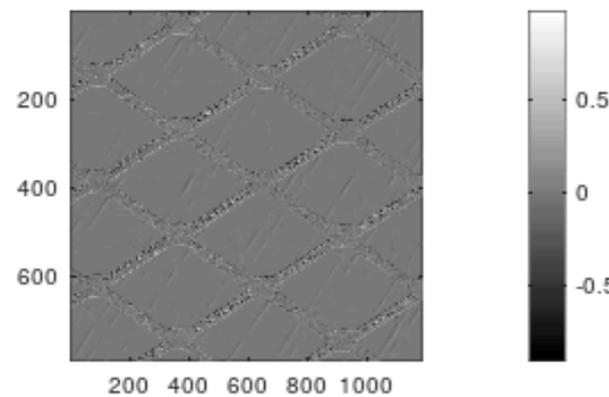
Sobel horizontal



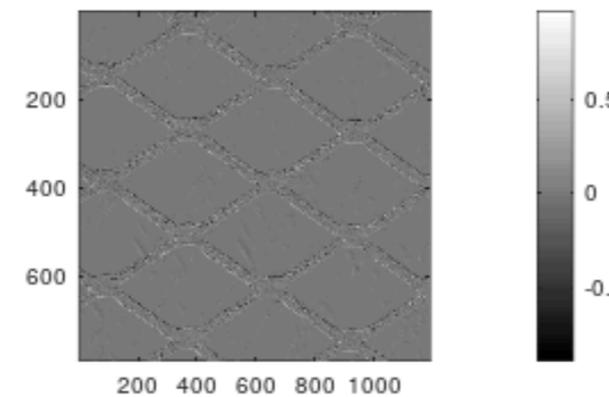
Sobel complet



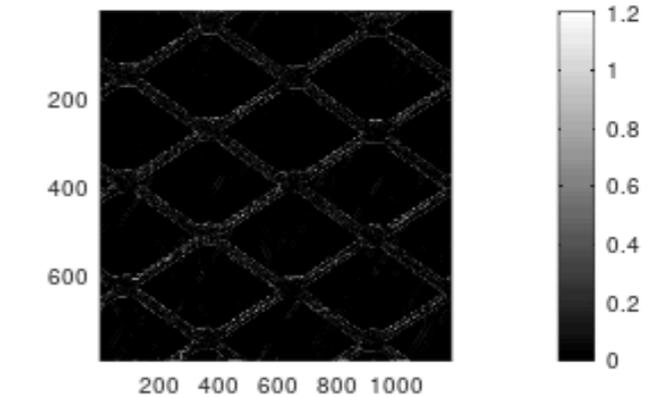
Roberts diagonal



Roberts anti-diagonal



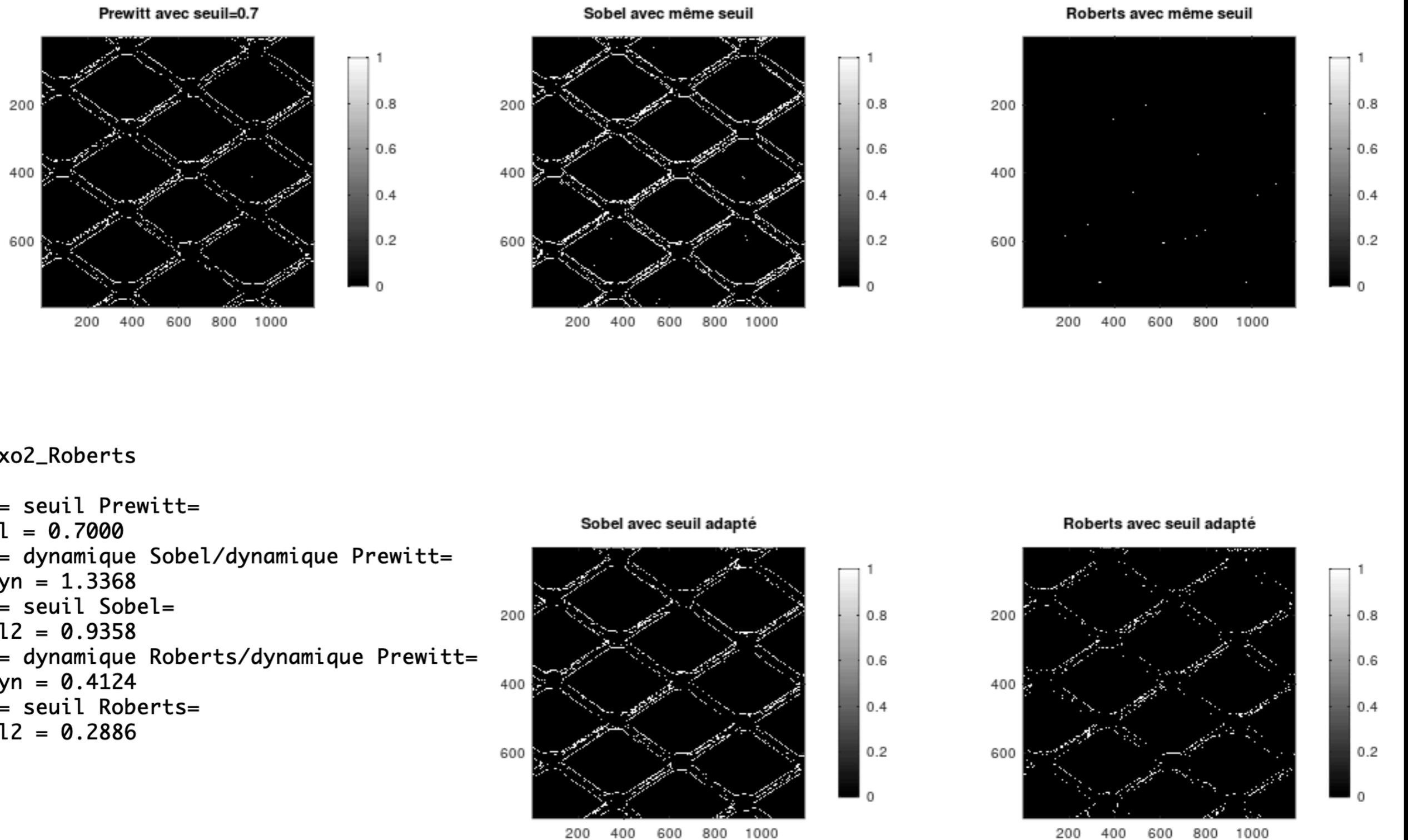
Roberts complet



# IV. Détection de contours

```
55 % seuillages/comparaison
56 seuil=0.7;
57 IPs=(IPvh>seuil);
58 'seuil Prewitt=', seuil
59
60 ISs=(ISvh>seuil);
61 dyn_Prewitt=max(max(IPvh))-min(min(IPvh));
62 dyn_Sobel=max(max(ISvh))-min(min(ISvh));
63 dyndyn=dyn_Sobel/dyn_Prewitt;
64 'dynamique Sobel/dynamique Prewitt=', dyndyn
65 seuil2=seuil*dyndyn;
66 'seuil Sobel=', seuil2
67 ISs2=(ISvh>seuil2);
68
69 IRs=(IRab>seuil);
70 dyn_Roberts=max(max(IRab))-min(min(IRab));
71 dyndyn=dyn_Roberts/dyn_Prewitt;
72 'dynamique Roberts/dynamique Prewitt=', dyndyn
73 seuil2=seuil*dyndyn;
74 'seuil Roberts=', seuil2
75 IRs2=(IRab>seuil2);
76
77 figure(2), colormap(gray)
78 subplot(2,3,1), imagesc(IPs), colorbar
79     title(['Prewitt avec seuil=', num2str(seuil)]), axis('square')
80 subplot(2,3,2), imagesc(ISs), colorbar
81     title('Sobel avec même seuil'), axis('square')
82 subplot(2,3,3), imagesc(IRs), colorbar
83     title('Roberts avec même seuil'), axis('square')
84 subplot(2,3,5), imagesc(ISs2), colorbar
85     title('Sobel avec seuil adapté'), axis('square')
86 subplot(2,3,6), imagesc(IRs2), colorbar
87     title('Roberts avec seuil adapté'), axis('square')
```

# IV. Détection de contours



```
>> exo2_Roberts
```

```
ans = seuil Prewitt=
```

```
seuil = 0.7000
```

```
ans = dynamique Sobel/dynamique Prewitt=
```

```
dyndyn = 1.3368
```

```
ans = seuil Sobel=
```

```
seuil2 = 0.9358
```

```
ans = dynamique Roberts/dynamique Prewitt=
```

```
dyndyn = 0.4124
```

```
ans = seuil Roberts=
```

```
seuil2 = 0.2886
```

# IV. Détection de contours

- Visuellement, Prewitt et Sobel sont similaires.
- Roberts : contours plus fins (à cause de la plus faible dimension du filtre), mais contours plus cassés et plus de bruit.
- Seuil décroissant => contours moins cassés (=contours plus « fermés »), mais plus de bruit (=plus de « faux contours »)
- Seuil croissant => moins de bruit (=moins de « faux contours »), mais contours pas tous détectés (=contours plus cassés, plus « ouverts »).

# IV. Détection de contours

- Les coefficients appliqués au seuil correspondent en fait aux rapports entre les sommes des valeurs absolues des coefficients des filtres eux-mêmes. En effet, pour un contour d'amplitude donnée, la réponse du filtre est proportionnelle à cette somme.

Par exemple, ici :

$$\begin{aligned} \sum_{|\text{coeff. de Prewitt}|} &= 6 ; \quad \sum_{|\text{coeff. de Roberts}|} = 2 \\ \Rightarrow \text{seuil}_{\text{Roberts}} &= \frac{1}{3} \text{seuil}_{\text{Prewitt}} \end{aligned}$$

Et :

$$\begin{aligned} \sum_{|\text{coeff. de Prewitt}|} &= 6 ; \quad \sum_{|\text{coeff. de Sobel}|} = 8 \\ \Rightarrow \text{seuil}_{\text{Sobel}} &= \frac{4}{3} \text{seuil}_{\text{Prewitt}} \end{aligned}$$

# IV. Détection de contours

## 5- COMPARAISON QUANTITATIVE SUR LA DÉTECTION DE CONTOURS

- Nous avons pour ceci besoin de métriques pertinentes :
  - > probabilité de détection (d'un contour) : proportion de points de contours correctement détectés.
  - > probabilité de fausse alarme : proportion de points de contours détectés à tort.
- Mais aussi d'une image simple avec un contour simple (par exemple un contour vertical).
- Et enfin d'une stratégie de comparaison : par exemple régler les seuils des 3 filtrages de manière à avoir la même probabilité de fausse alarme ( $P_{fa}$ ), puis comparer ensuite les probabilités de détection ( $P_d$ ) obtenues (le contraire eût été tout aussi pertinent).

# IV. Détection de contours

- **EXERCICE 3 : Comparer quantitativement les filtrages de Prewitt, Sobel et Roberts sur une image 128x128 composée d'une première moitié grise légèrement foncée (à gauche), à 45% de luminosité, et d'une autre moitié grise plus claire à droite (à 55% de luminosité), puis bruitée (bruit Gaussien additif de moyenne nulle et de variance 0.001).**

**(Commencer par un seuil de, par exemple, 0.25 pour Prewitt, sauver l'image bruitée afin de ne pas la changer à chaque exécution du programme => faire donc deux routines : une pour créer l'image bruitée, l'autre pour la traiter.)**

**(Faire ça en 3 étapes : (a) générer l'image bruitée (1ère routine) ; (b) en déduire les images de contour binaires pour un nombre fixé de fausses alarmes (2ème routine) ; (c) compléter la 2ème routine par le calcul de la probabilité de détection pour chacun des 3 cas de paire de filtres.)**

**(Discuter la taille des images filtrées « valides » avec Prewitt ou Sobel d'une part et Roberts d'autre part...)**

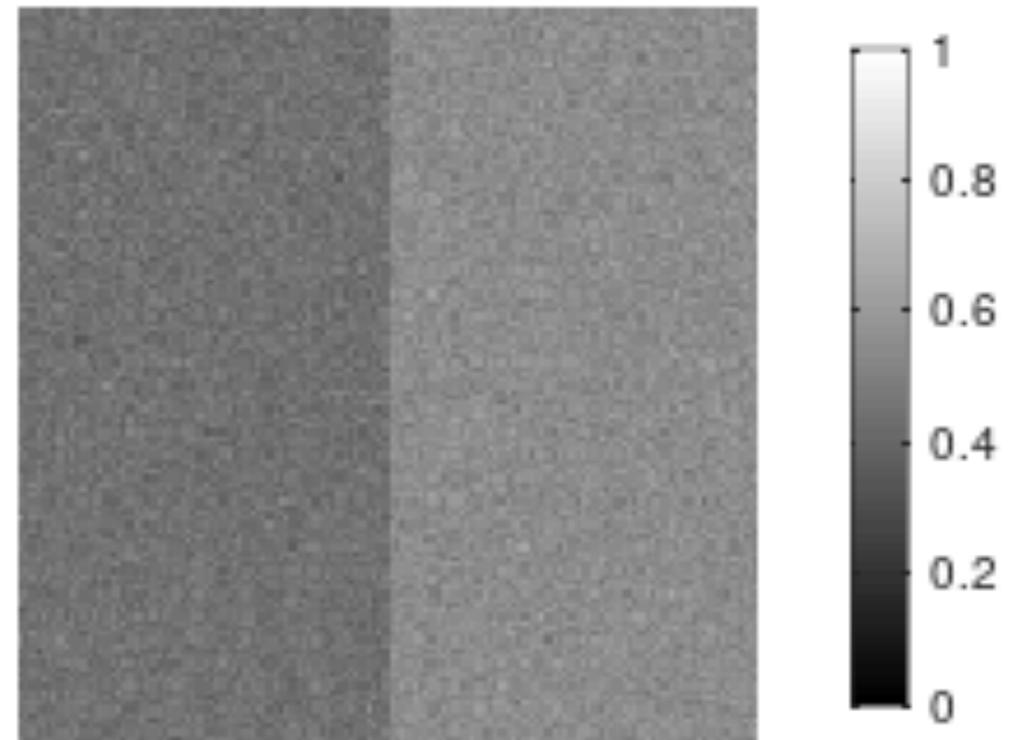
# IV. Détection de contours

```
1 clear
2 close all
3
4 % image test
5 dim=128; I=.55*ones(dim,dim); I(:,1:dim/2)=.45;
6 figure, colormap(gray)
7 subplot(1,2,1), imshow(I), title('image'), colorbar
8
9 % bruit gaussien additif
10 J=imnoise(I, 'gaussian', 0., 1e-3);
11 subplot(1,2,2), imshow(J), title('image+bruit'), colorbar
12
13 % sauver l'image de test bruitée
14 save image_bruit J
```

image



image+bruit

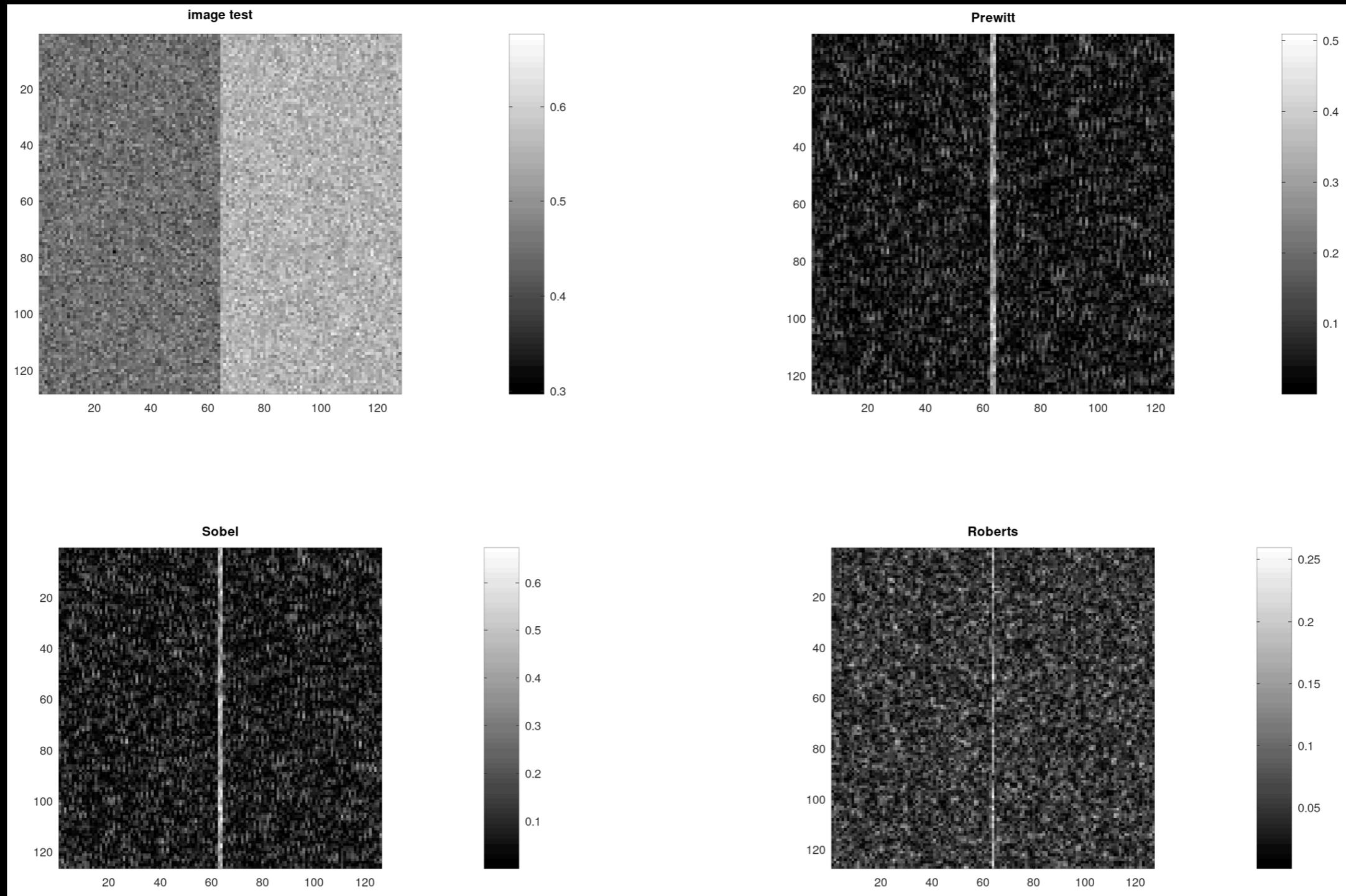


# IV. Détection de contours

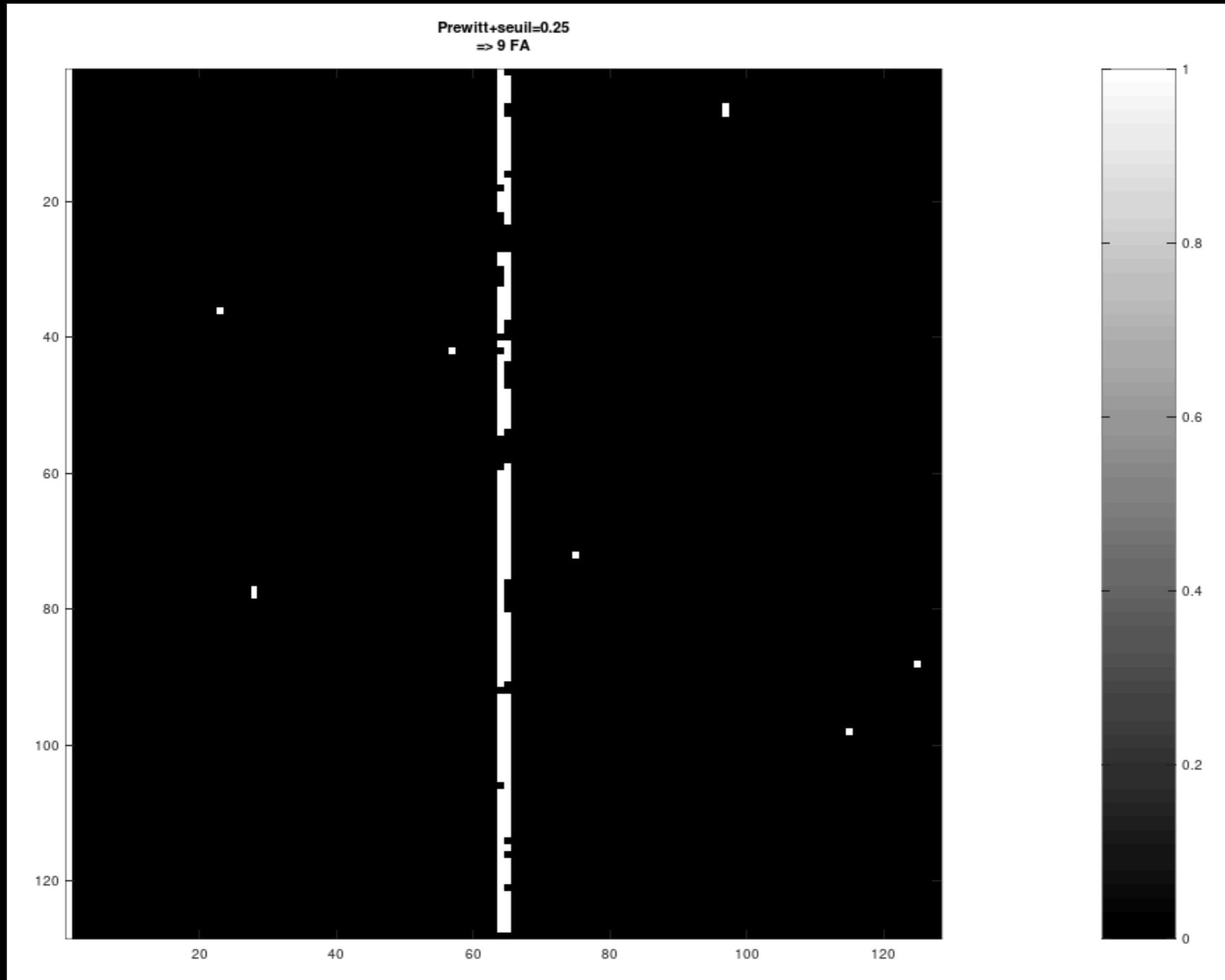
```
close all
pkg load image

%---
% load image from disk
load image_bruit, whos J
% figure 1
figure(1), colormap(gray)
subplot(2,2,1), imagesc(J), title('image test'), colorbar, axis('square')
%---
% Prewitt
Ph=fspecial('prewitt'); Pv=-Ph'; JP=abs(filter2(Pv,J, 'valid'));
subplot(2,2,2), imagesc(JP), title('Prewitt'), colorbar, axis('square')
%---
% Sobel
Sh=fspecial('sobel'); Sv=-Sh'; JS=abs(filter2(Sv,J, 'valid'));
subplot(2,2,3), imagesc(JS), title('Sobel'), colorbar, axis('square')
%---
% Roberts
Ra=[1 0;0 -1]; Rb=rot90(Ra,-1);
JRa=filter2(Ra,J, 'valid'); JRb=filter2(Rb,J, 'valid');
JR=sqrt(JRa.*JRa+JRb.*JRb);
subplot(2,2,4), imagesc(JR), title('Roberts'), colorbar, axis('square')
%---
% comparaison des images uniques de contours seuillées
% seuillages
seuilP=.25; JPs=JP>seuilP;
% - ligne ajoutée pour l'exo 4 chapitre 5
save image_Prewitt_seuil_new.mat JPs
% -
seuilS=.362; JSs=JS>seuilS;
seuilR=.175; JRs=JR>seuilR;
% figure 2
figure(2), colormap(gray)
imagesc(JPs), title(['Prewitt+seuil=', num2str(seuilP)]; '=> 9 FA')), colorbar
axis('square')
% figure 3
figure(3), colormap(gray)
imagesc(JSs), title(['Sobel+seuil=', num2str(seuilS)] ; '=> 9 FA')), colorbar
axis('square')
% figure 4
figure(4), colormap(gray)
imagesc(JRs), title(['Roberts+seuil=', num2str(seuilR)]; '=> 9 FA')), colorbar
axis('square')
```

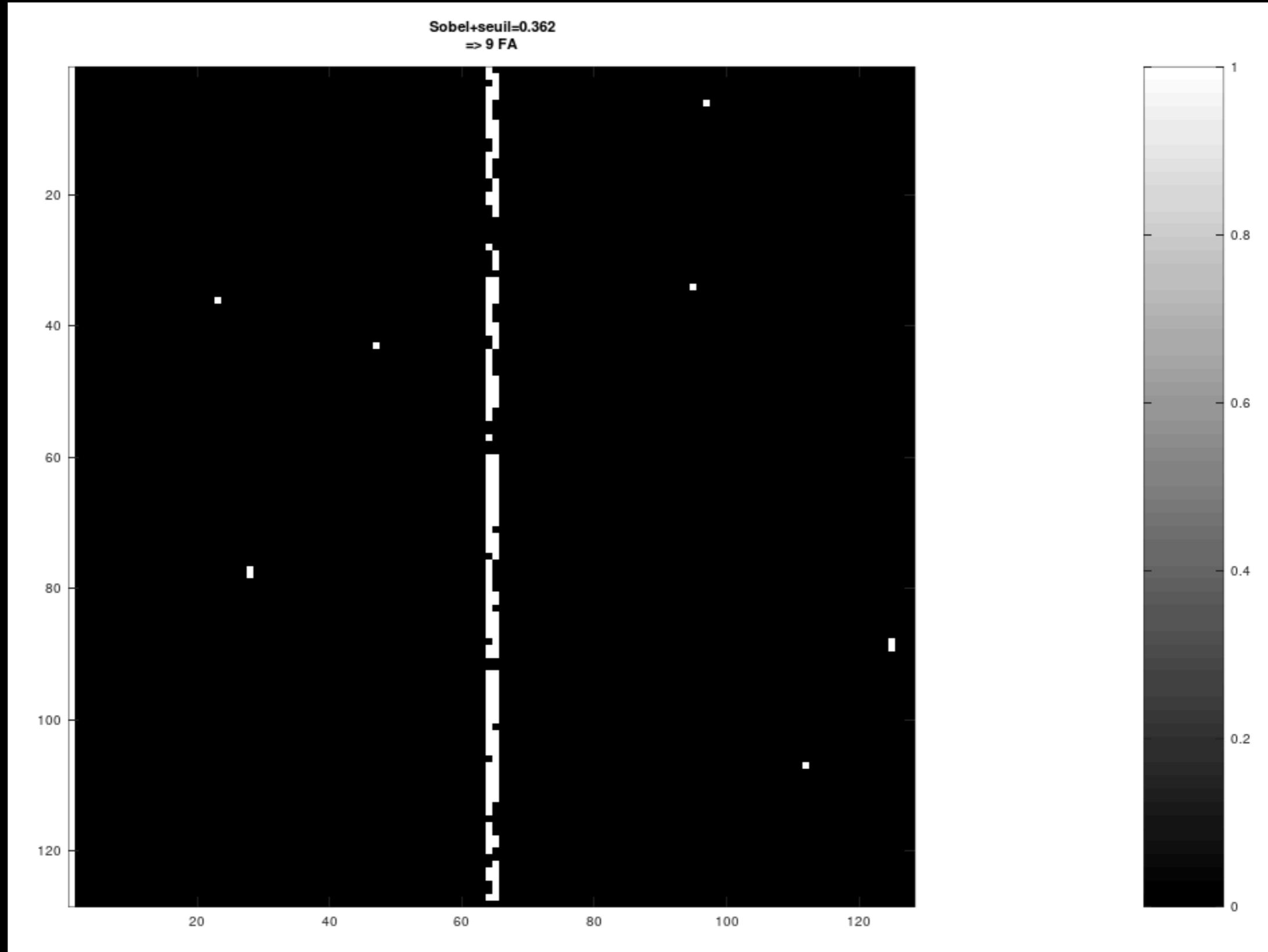
# IV. Détection de contours



# IV. Détection de contours



# IV. Détection de contours



# IV. Détection de contours

