

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 (en mode TP) :**

Comparer quantitativement les filtrages de Prewitt, Sobel et Roberts sur une image 128x128 composée d'une première moitié grise légèrement foncée, à 45% de luminosité, et d'une autre moitié grise plus claire, 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...)

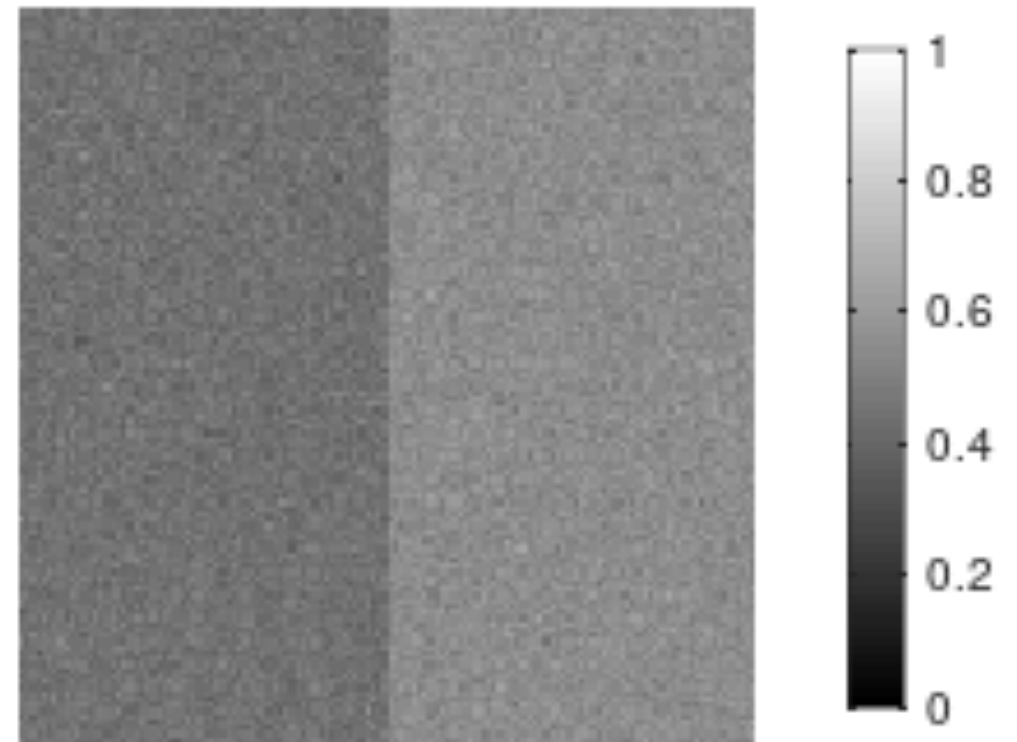
IV. Détection de contours

```
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

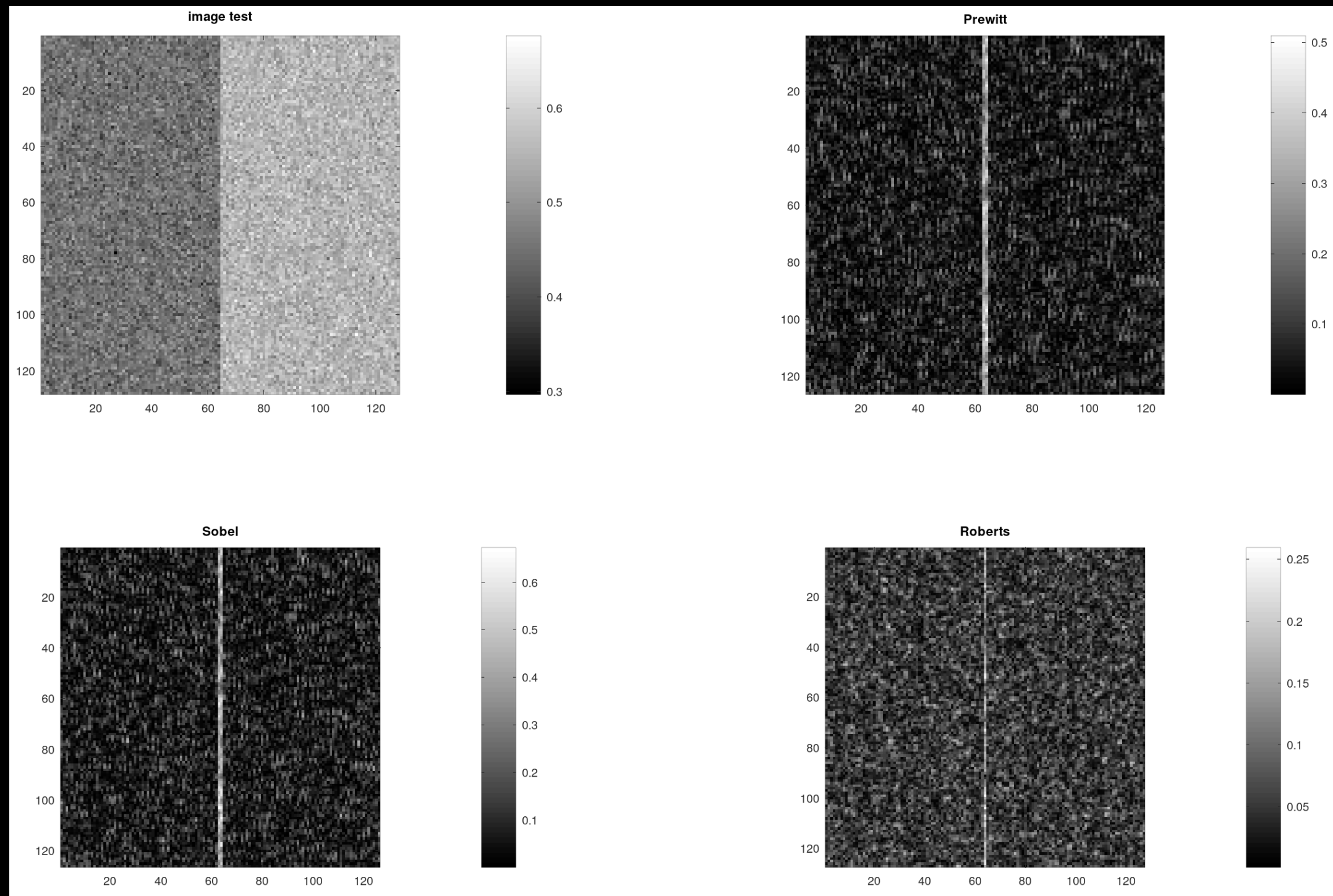


IV. Détection de contours

```
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')
```

IV. Détection de contours

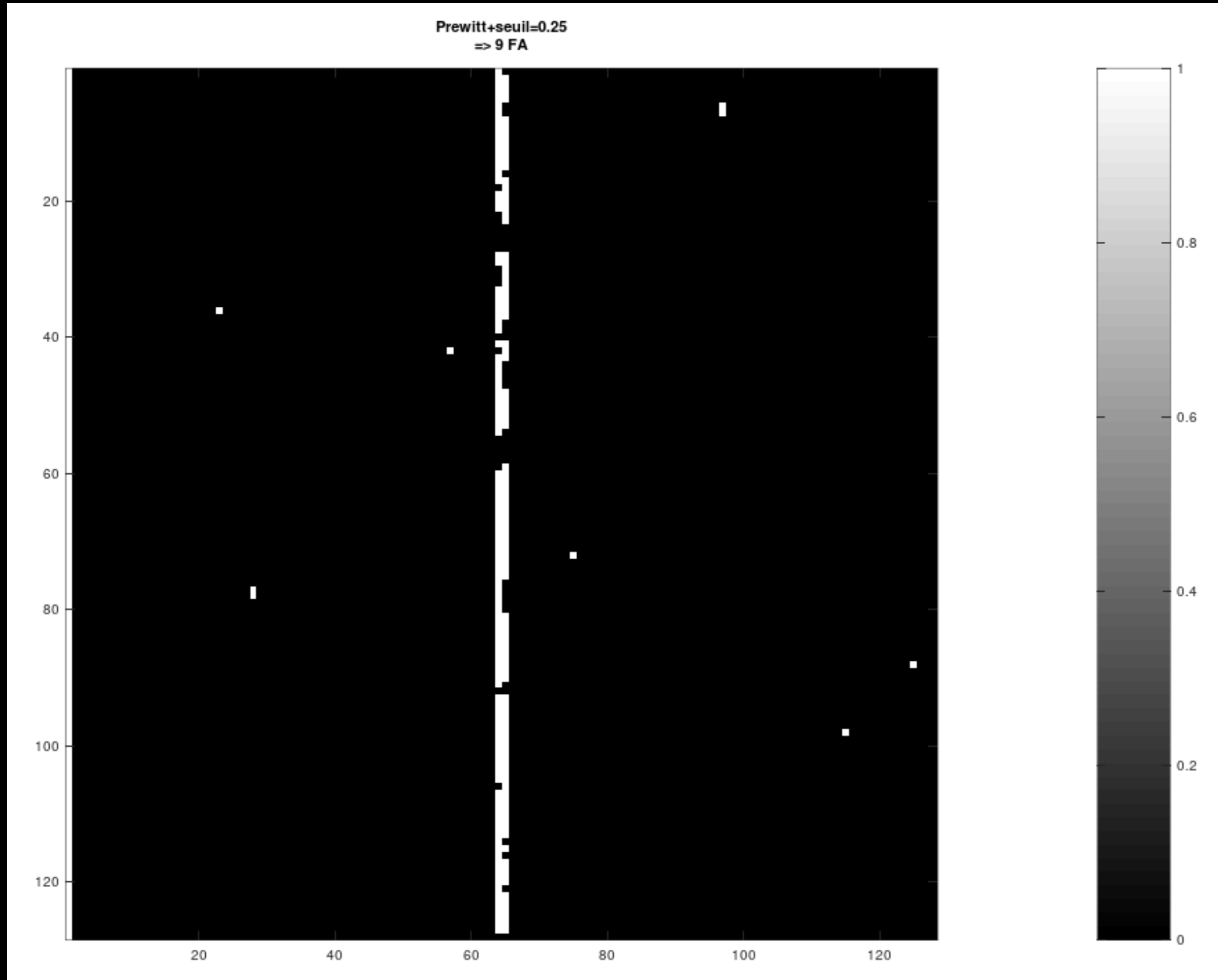


IV. Détection de contours

```
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
% -
```

IV. Détection de contours

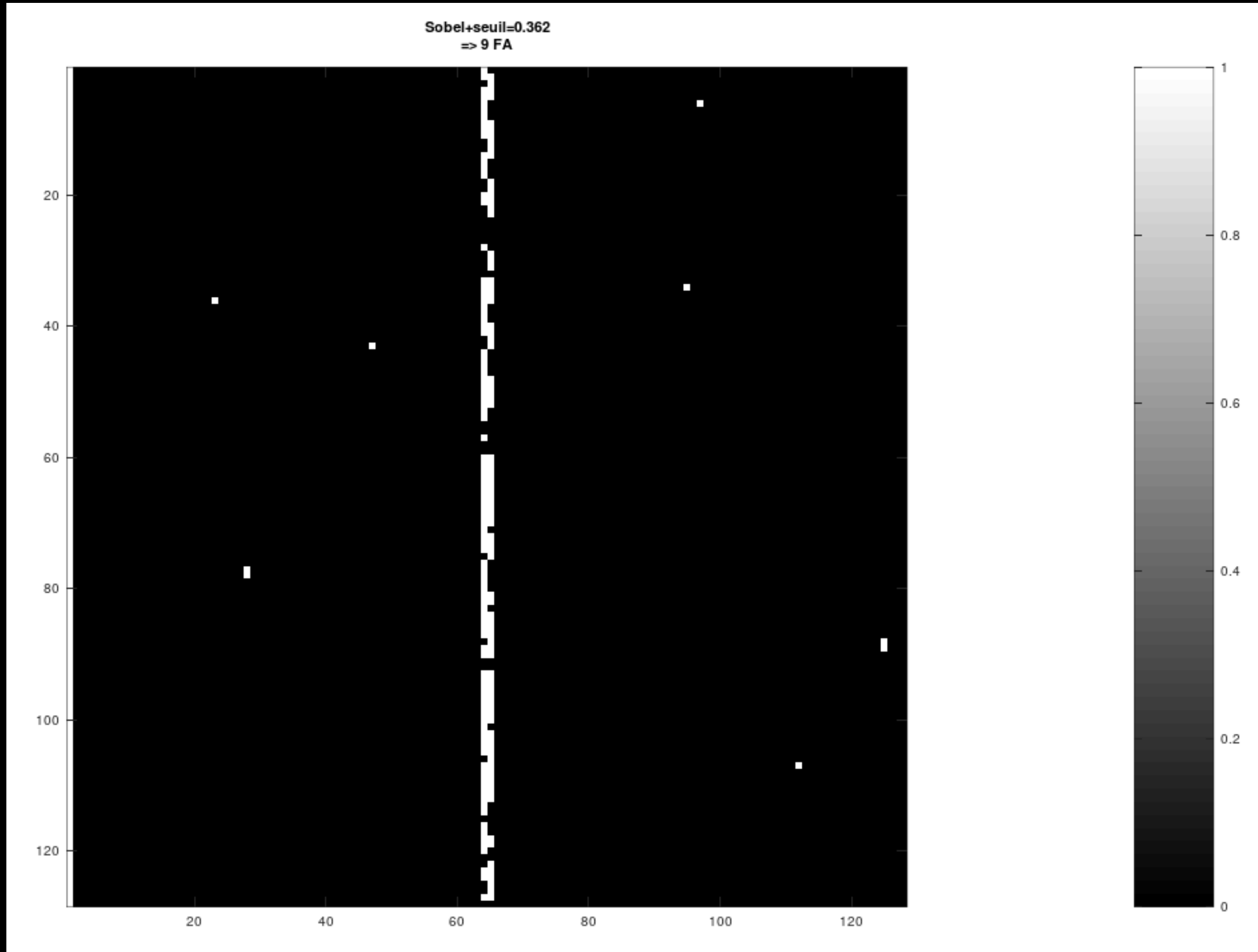


IV. Détection de contours

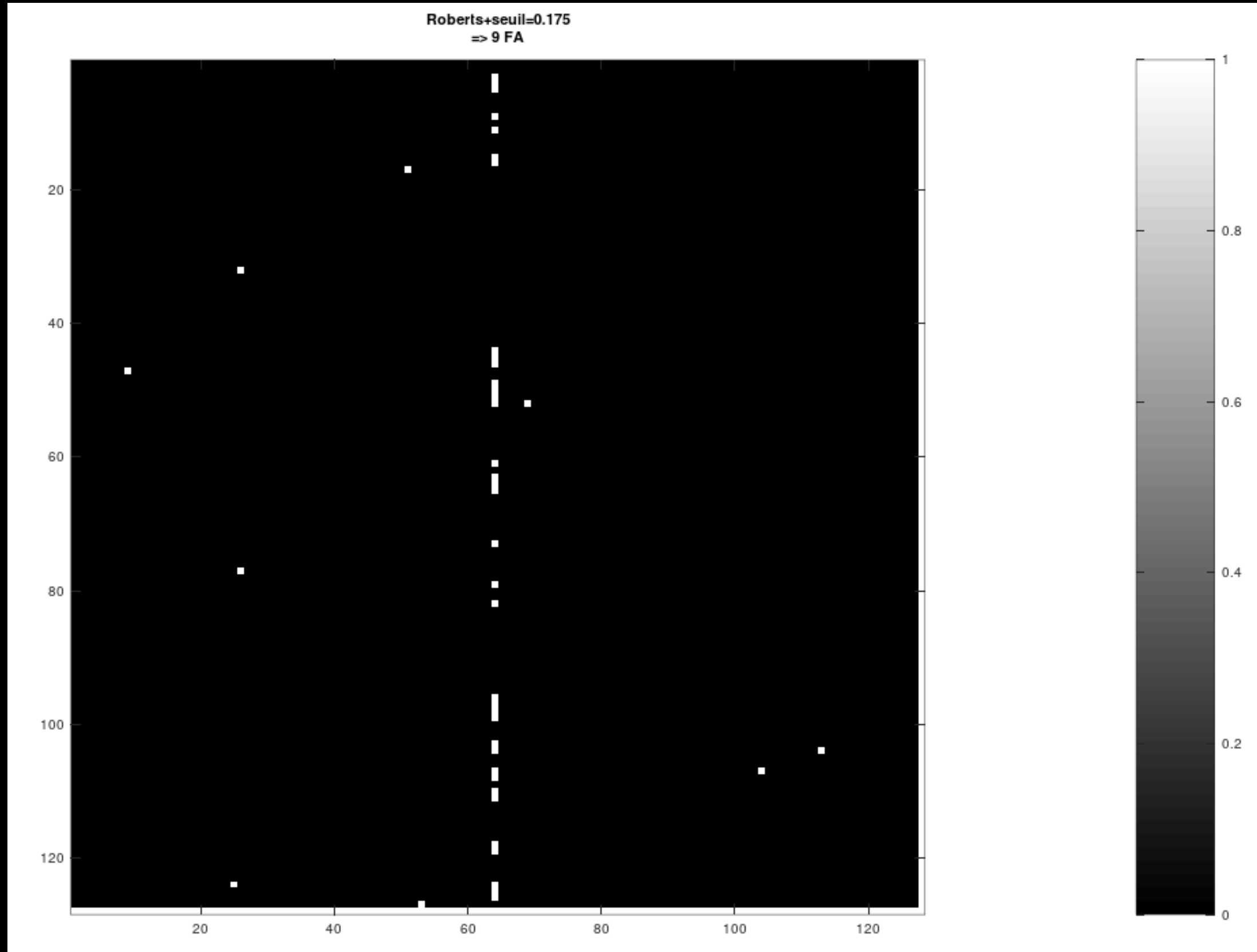
```
close all
pkg load image

%---
% 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;
```

IV. Détection de contours



IV. Détection de contours



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

```
43 %---
44 % mise en évidence du contour
45 % somme sur les colonnes
46 sumP=sum(JPs); sumS=sum(JSs); sumR=sum(JRs);
47 % détection du contour
48 dimP = (size(JPs))(1);
49 dimS = (size(JSs))(1);
50 dimR = (size(JRs))(1);
51 gooP = sumP(dimP/2:dimP/2+1);
52 gooS = sumS(dimS/2:dimS/2+1);
53 gooR = sumR((dimR+1)/2);
54
55 {'-----' ;
56 'Détection du contour...' ;
57 '-----' ;
58 ['Prewitt : ', num2str(dimP), ' ', num2str(gooP)] ;
59 ['Sobel   : ', num2str(dimS), ' ', num2str(gooS)] ;
60 ['Roberts : ', num2str(dimR), ' ', num2str(gooR)] ;
61 '-----' }
62 % => probabilités de détection
63 {'-----' ;
64 '=> Probabilités de détection...' ;
65 '-----' ;
66 ['Prewitt : ', num2str(sum(gooP)/(2*dimP))] ;
67 ['Sobel   : ', num2str(sum(gooS)/(2*dimS))] ;
68 ['Roberts : ', num2str(sum(gooR)/dimR)] ;
69 '-----' }
```

```
dummy=size(JPs), dimP = dummy(1);
dummy=size(JSs), dimS = dummy(1);
dummy=size(JRs), dimR = dummy(1);
```

```
ans =
{
  [1,1] = -----
  [2,1] = Détection du contour...
  [3,1] = -----
  [4,1] = Prewitt : 126  107  97
  [5,1] = Sobel   : 126  95  83
  [6,1] = Roberts : 127  36
  [7,1] = -----
}
```

```
ans =
{
  [1,1] = -----
  [2,1] = => Probabilités de détection...
  [3,1] = -----
  [4,1] = Prewitt : 0.80952
  [5,1] = Sobel   : 0.70635
  [6,1] = Roberts : 0.28346
  [7,1] = -----
}
```

- Conclusion : Prewitt > Sobel >> Roberts
- Mais : Roberts détecte un « vrai » bord, de 1 pixel de large (pas 2)...

IV. Détection de contours

\	nb de FA	seuil Prewitt	seuil Sobel	seuil Roberts	Pdét Prewitt	Pdét Sobel	Pdét Roberts
Jérémy Julien	5	0,305	0,429	0,15	0,51?	0,47?	0,07?
Alexis	9	0,25	0,333	0,14	0,79	0,67	0,15
Inès	10	0,2685	0,37	0,17			
Yumna	2	0,28	0,39	0,185			
Yasmine	2	0,28	0,42	0,17			
Annabelle	15	0,26	0,3733	0,14453			
Nathan Soufyan	18	0,29	0,33	0,19			
Bouchra	6	0,33	0,376	0,145	0,76	0,62	0,29

IV. Détection de contours

- **EXERCICE 3 (suite) : Calculer les P_{fa} correspondantes « à la main ». Puis coder ce calcul de P_{fa} de manière à le rendre automatique.**

Théoriquement, on doit avoir :

$$P_{fa}^P = P_{fa}^S = 9/(126 \times 124) \simeq 5.7604 \cdot 10^{-4}$$

$$P_{fa}^R = 9/(127 \times 126) \simeq 5.6243 \cdot 10^{-4}$$

Et en effet :

```
71 %---
72 % probabilité de fausse alarme
73 FA_ga_P = sum(sumP(1:dimP/2-1));
74 FA_dr_P = sum(sumP(dimP/2+2:dimP));
75 FA_ga_S = sum(sumS(1:dimS/2-1));
76 FA_dr_S = sum(sumS(dimS/2+2:dimS));
77 FA_ga_R = sum(sumR(1:(dimR+1)/2-1));
78 FA_dr_R = sum(sumR((dimR+1)/2+1:dimR));
79 {'-----' ;
80  'Probabilité de fausse alarme' ;
81  '-----' ;
82  ['Prewitt : ', num2str((FA_ga_P+FA_dr_P)/((dimP-2)*dimP))] ;
83  ['Sobel   : ', num2str((FA_ga_S+FA_dr_S)/((dimS-2)*dimS))] ;
84  ['Roberts : ', num2str((FA_ga_R+FA_dr_R)/((dimR-1)*dimR))] ;
85  '-----' }
```

```
ans =
{
    [1,1] = -----
    [2,1] = Probabilité de fausse alarme
    [3,1] = -----
    [4,1] = Prewitt : 0.00057604
    [5,1] = Sobel   : 0.00057604
    [6,1] = Roberts : 0.00056243
    [7,1] = -----
}
```

IV. Détection de contours

- **EXERCICE 3bis** : Partir de $P_{fa} \approx 1\%$, en déduire le nombre de fausses alarmes (N_{fa}) correspondant (pour chaque cas : Prewitt, Sobel, Roberts), en déduire alors (pour chaque cas aussi) le seuil nécessaire (en comptant automatiquement N_{fa}), puis calculer les $P_{dét.}$ résultantes à partir des images de contour unique seuillées correspondantes.

$P_{fa}=1\% \Rightarrow N_{fa} = 156$ pour Prewitt et Sobel (1% de 124x126) et $N_{fa} = 160$ pour Roberts (1% de 126x127).