IV. Détection de contours

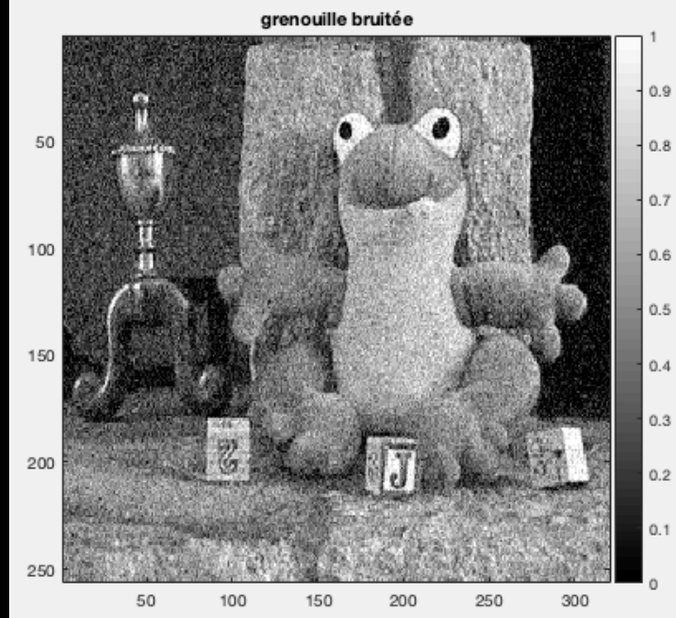
- **EXERCICE 4bis : Même chose avec une image significativement bruitée (bruit gaussien additif, variance de 0.01).**

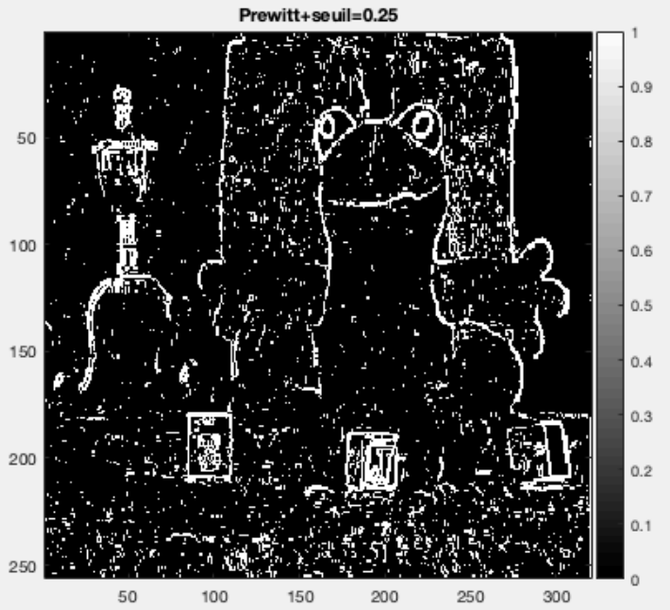
(Bien écrêter les valeurs de l'image bruitée qui pourraient descendre en-dessous de 0 ou dépasser 1, pour être compatible avec le fonctionnement de edge qui réclame des images entre 0 et 1.)

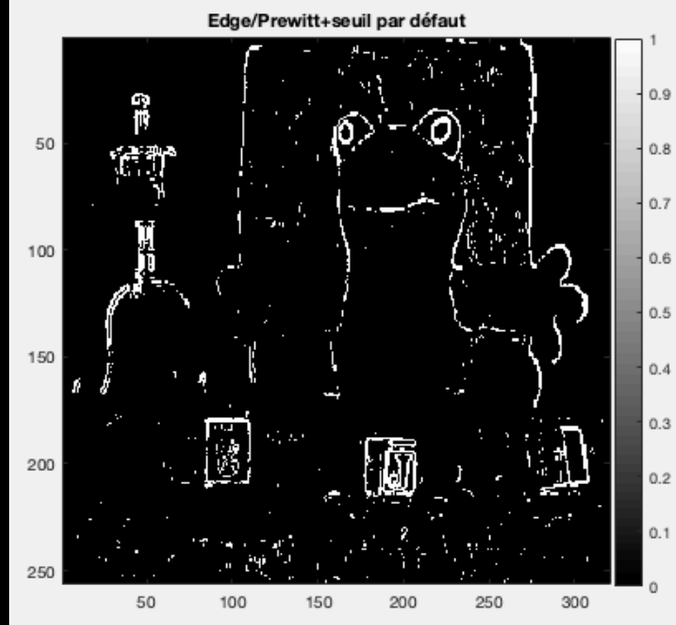
IV. Détection de contours

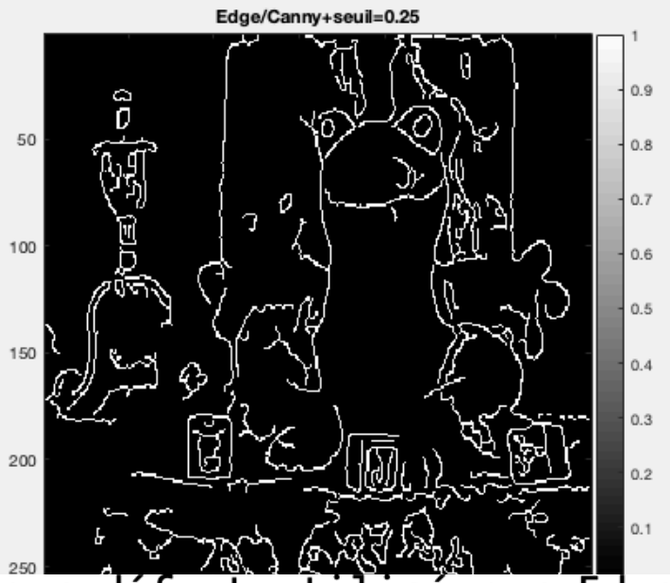
```
%---
% image de la grenouille bruitée
J=imnoise(J, 'gaussian', 0., .01);
% écrétage de l'image entre 0 et 1
idx=find(J<0); J(idx)=0.;
idx=find(J>1); J(idx)=1.;
% Prewitt
Ph=fspecial('prewitt'); Pv=-Ph';
IPh=filter2(Ph,J,'same'); IPv=filter2(Pv,J,'same');
IPvh=sqrt(IPv.^2+IPh.^2);
IPvh=IPvh-min(min(IPvh)); IPvh=IPvh/max(max(IPvh));
seuilP=.25; IPs=IPvh>seuilP;
% Edge/Prewitt
%'seuil par défaut utilisé par Edge/Prewitt', 2*sqrt(mean(mean(IPvh.^2)))
[JE, seuilP2]=edge(J, 'prewitt', 'nothinning');
'seuil par défaut utilisé par Edge/Prewitt (avec bruit)', seuilP2
% Edge/Canny
JC=edge(J, 'canny', seuilC);
% figure
figure(2), colormap(gray)
subplot(2,2,1), imagesc(J), title('grenouille bruitée'), colorbar, axis('square')
subplot(2,2,2), imagesc(IPs), colorbar, axis('square')
   title(['Prewitt+seuil=',num2str(seuilP)])
subplot(2,2,3), imagesc(JE), colorbar, axis('square')
   title('Edge/Prewitt+seuil par défaut')
subplot(2,2,4), imagesc(JC), colorbar, axis('square')
   title(['Edge/Canny+seuil=',num2str(seuilC)])
```

IV. Détection de contours










```
ans = seuil par défaut utilisé par Edge/Prewitt (sans bruit)
seuilP1 = 0.1276
ans = seuil par défaut utilisé par Edge/Prewitt (avec bruit)
seuilP2 = 0.1685
```

V. Morphologie mathématique

1- INTRODUCTION

- **Un opérateur de morphologie mathématique reçoit une image en entrée et fournit une image en sortie.**

- **On définit cet opérateur par l'intermédiaire d'un élément structurant. (On peut aussi le définir par l'intermédiaire d'une table de correspondance.)**

- **En général : opération sur des images binaires.**

V. Morphologie mathématique

2- DÉFINITION PAR UN ÉLÉMENT STRUCTURANT

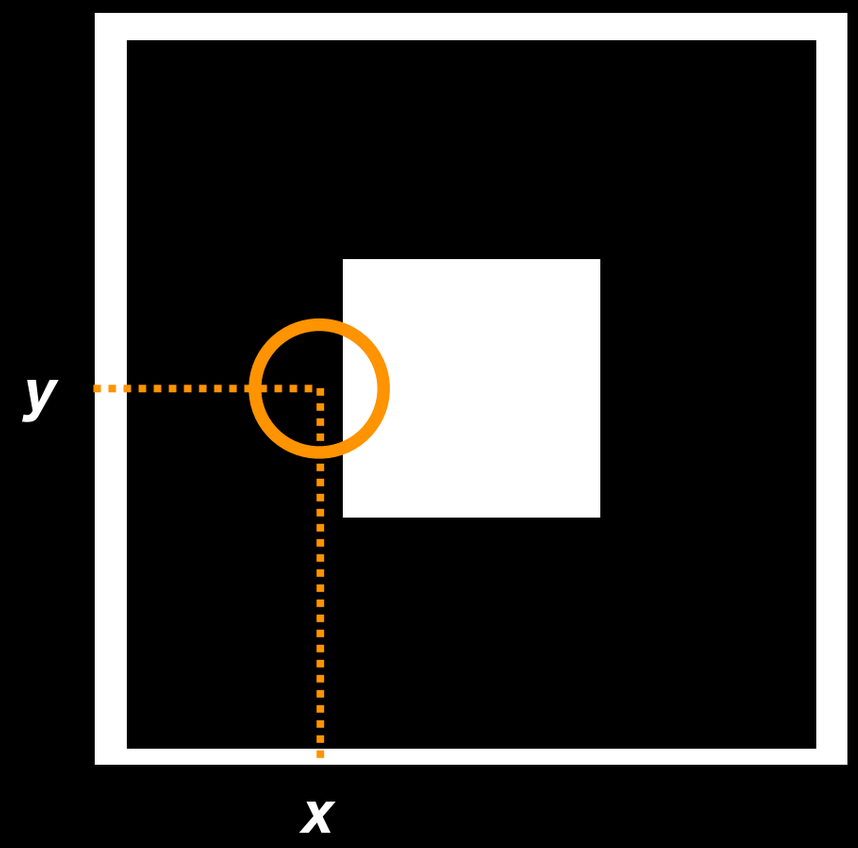
- **Un élément structurant = un masque (comme pour un filtrage en passant par le plan de Fourier)…**
- **Dans le cas de l'érosion et de la dilatation :**

$$\text{Erosion} : I'(x,y) = min_{(\mathrm{a,b})\in \mathrm{S_{xy}}}[I(a,b)]$$
$$\text{Dilatation} : I'(x,y) = max_{(\mathrm{a,b})\in \mathrm{S_{xy}}}[I(a,b)]$$

où S_{xy} est l'élément structurant placé en (*x*, *y*), *I* est l'image initiale, et *I'* est l'image résultante de l'application de l'opérateur morphologique.

V. Morphologie mathématique

- **Par exemple : on veut dilater *I*, un rectangle blanc sur fond noir, avec comme élément structurant un disque…**



- **Pour connaître *I'(x,y)*, on place le centre du disque en (*x*, *y*), on recherche la valeur <u>maximale</u> des pixels se trouvant « sous » le disque, et on remplace par cette valeur.**

V. Morphologie mathématique

- **Pour l'érosion : idem, mais en cherchant la valeur <u>minimale</u>.**
- **Sous MATLAB/OCTAVE :**

>> IE = imerode(I, S);

>> ID = imdilate(I, S);

où *n* permet de répéter éventuellement l'opération plusieurs fois, et *S* est définit par exemple par :

>> S = strel('disk', 10); **(ou *strel('disk', 10, n)***
si on veut une approximation ≠ que 'n=4' (défaut) -> voir help)

10 étant le « *RAYON* » du disque.

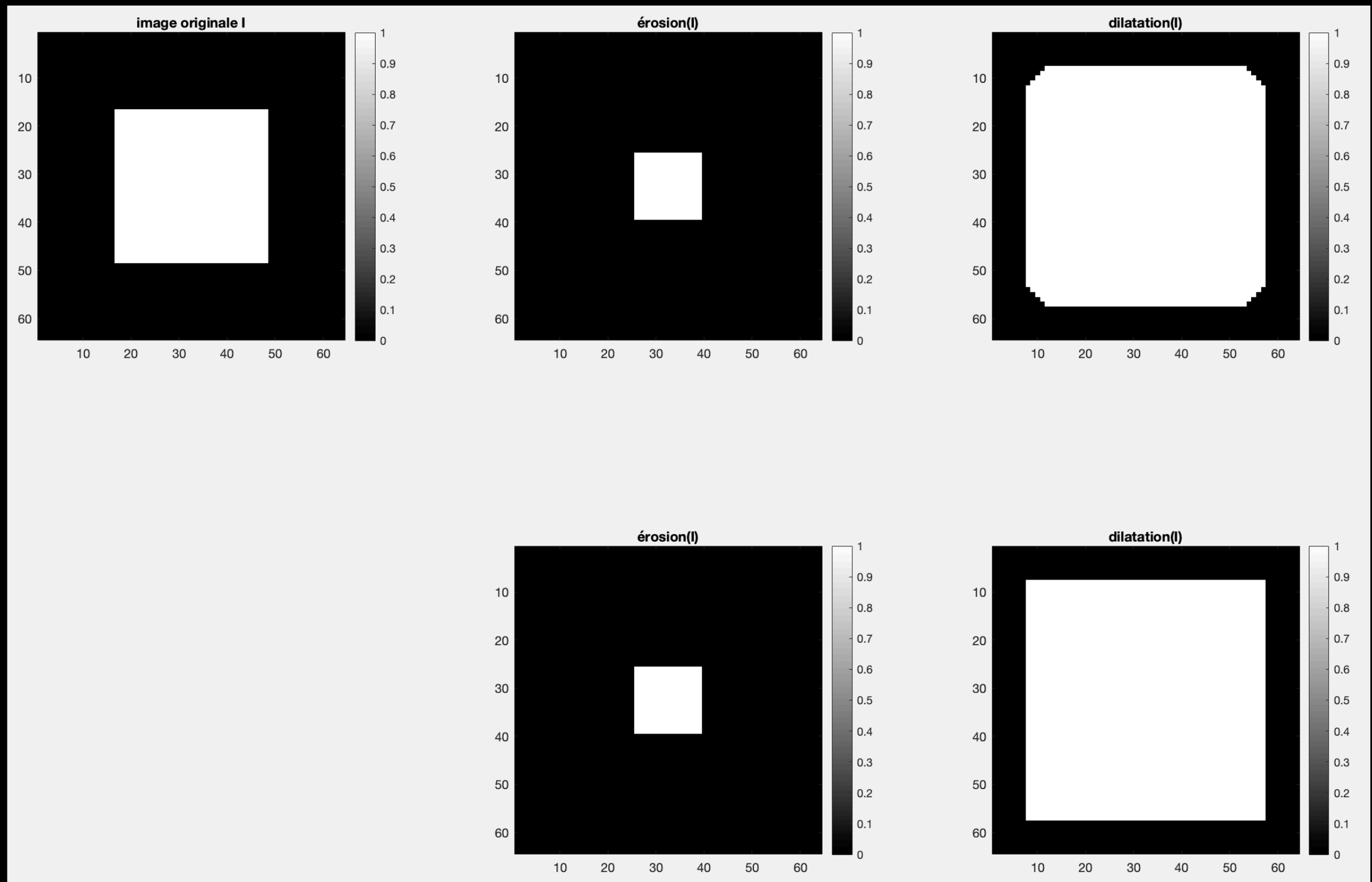
V. Morphologie mathématique

- **EXERCICE 1 : Éroder d'une part, et dilater d'autre part, un carré blanc 32x32 (qu'on appellera *I* par la suite), centré, sur fond noir 64x64, à l'aide d'un élément structurant 'disk' de « rayon » 10 pixels. Que se passe-t-il si on utilise un masque carré (option 'square' dans strel) ?**

V. Morphologie mathématique

```
clear
close all
%pkg load image

% carré blanc sur fond noir
dim=64; I=zeros(dim,dim);
% pour être général (n'importe quel carré blanc sur fond noir) :
nn=32; I(dim/2-nn/2+1:dim/2+nn/2, dim/2-nn/2+1:dim/2+nn/2)=1;
% ou pour faire simple ici :
% I=zeros(64,64);
% I(17:48,17:48)=1;
whos I

figure, colormap('gray')
subplot(1,3,1), imagesc(I), colorbar, axis('square'), title('image originale I')

% érosion(I) vs. dilatation(I)
disque=strel('disk',10);
carre =strel('square',19);

%IE=imerode(I, disque); % Octave
IE=imerode(I, disque);
IE2=imerode(I, carre);
subplot(2,3,2), imagesc(IE), colorbar, axis('square'), title('érosion(I)')
subplot(2,3,5), imagesc(IE2), colorbar, axis('square'), title('érosion(I)')

%ID=imdilate(I, disque); % Octave
ID=imdilate(I, disque);
ID2=imdilate(I, carre);
subplot(2,3,3), imagesc(ID), colorbar, axis('square'), title('dilatation(I)')
subplot(2,3,6), imagesc(ID2), colorbar, axis('square'), title('dilatation(I)')
```

V. Morphologie mathématique

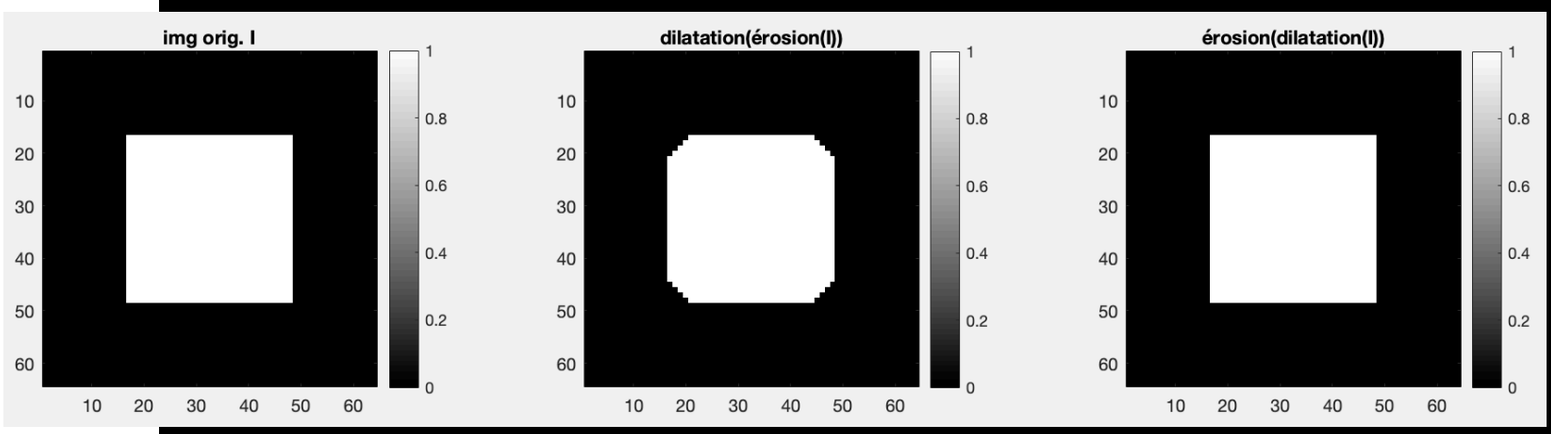


V. Morphologie mathématique

- **EXERCICE 2 : Éroder *puis* dilater *I* d'une part, et d'autre part dilater puis éroder le même carré blanc sur fond noir, avec le même disque comme élément structurant. Comparer le résultat des deux opérations, visuellement et *quantitativement*.**

V. Morphologie mathématique

```
clear
close all
%pkg load image % Octave

% carré blanc sur fond noir
dim=64; I=zeros(dim,dim);
nn=32; I(dim/2-nn/2+1:dim/2+nn/2, dim/2-nn/2+1:dim/2+nn/2)=1;
whos I

% érosion(I) vs. dilatation(I)
disque=strel('disk',10);
IE=imerode(I, disque);
ID=imdilate(I, disque);

% dilatation(érosion(I)) vs. érosion(dilatation(I))
figure, colormap('gray')
subplot(1,3,1), imagesc(I), colorbar, axis('square')
title('img orig. I')

IED=imdilate(IE, disque);
subplot(1,3,2), imagesc(IED), colorbar, axis('square')
title('dilatation(érosion(I))')

IDE=imerode(ID, disque);
subplot(1,3,3), imagesc(IDE), colorbar, axis('square')
title('érosion(dilatation(I))')

{'----------------------------------' ;
 'comparaison en termes d"intégrale :' ;
 '----------------------------------' ;
 ['I : ', num2str(sum(sum(I)))] ;
 ['dilatation(érosion(I)) : ', num2str(sum(sum(IED)))] ;
 ['érosion(dilatation(I)) : ', num2str(sum(sum(IDE)))] ;
 '----------------------------------' }
```




```
{'----------------------------------'}
{'comparaison en termes d"intégrale :'}
{'----------------------------------'}
{'I : 1024'                         }
{'dilatation(érosion(I)) : 984'     }
{'érosion(dilatation(I)) : 1024'    }
{'----------------------------------'}
```

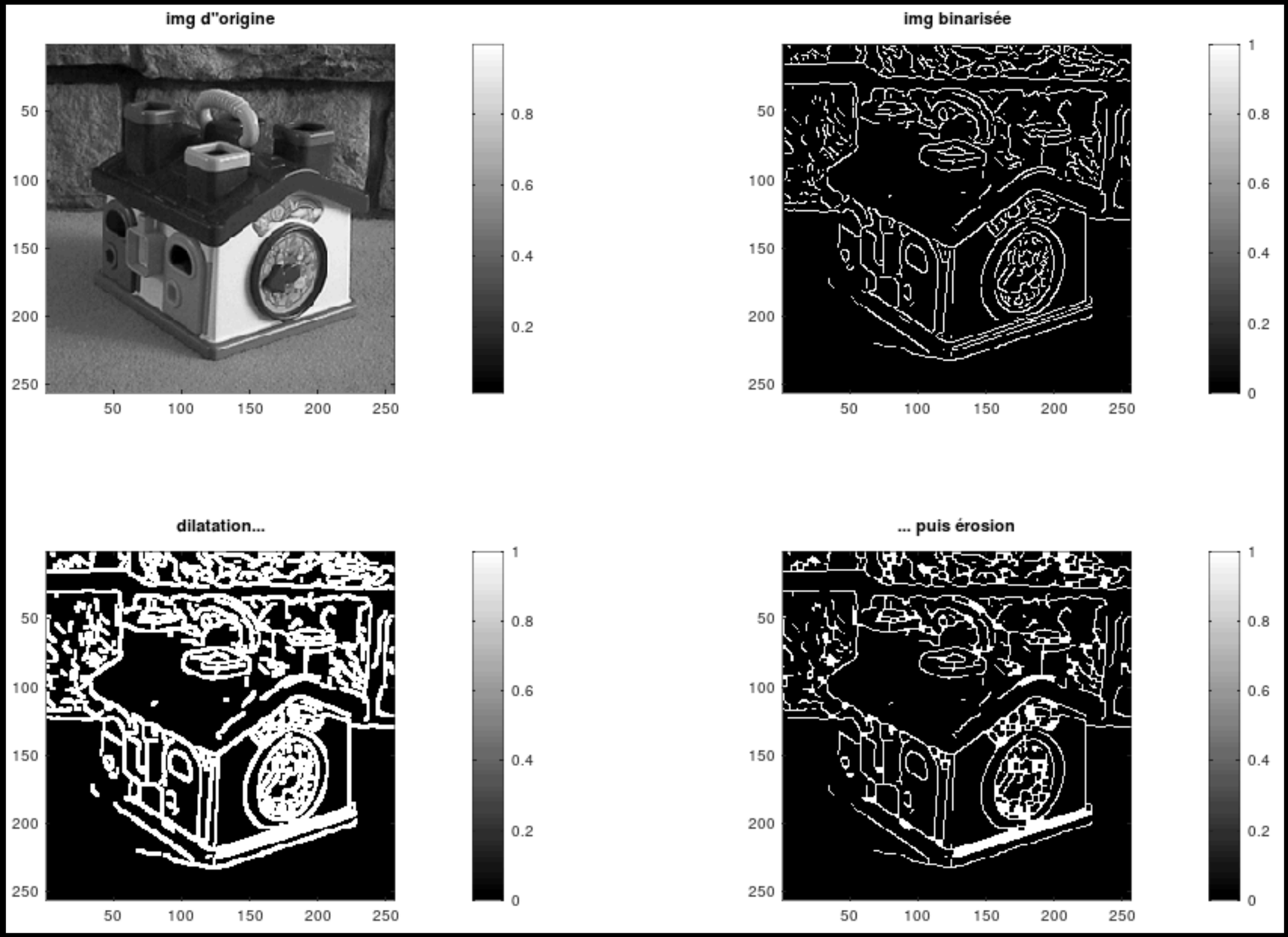
V. Morphologie mathématique

- **EXERCICE 3 : Reprendre l'image 'house', la binariser avec la fonction 'edge' (avec par exemple les paramètres par défaut, ou mieux l'algorithme de Canny), dilater puis éroder le résultat avec un élément structurant constitué d'un carré 3x3. Commenter.**

V. Morphologie mathématique

```
clear
close all
%pkg load image

im='/Users/marcel/Documents/MATLAB/0-images/house.jpg';
house=imread(im);
house=rgb2gray(house);
house=double(house)/255.;

hb=edge(house, 'canny');

carre=strel('square', 3);
% ou : ones(3,3);

hbD=imdilate(hb, carre);
hbDE=imerode(hbD,carre);

figure, colormap('gray')

subplot(2,2,1), imagesc(house), colorbar
axis('square'), title('img d"origine')

subplot(2,2,2), imagesc(hb), colorbar
axis('square'), title('img binarisée')

subplot(2,2,3), imagesc(hbD), colorbar
axis('square'), title('dilatation...')

subplot(2,2,4), imagesc(hbDE), colorbar
axis('square'), title('... puis érosion')
```

V. Morphologie mathématique



V. Morphologie mathématique

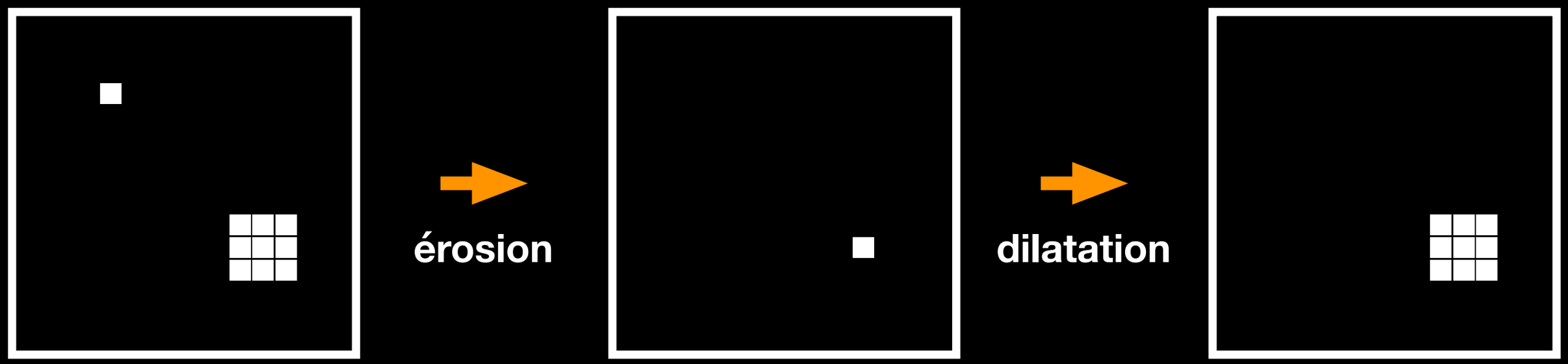
3- OUVERTURE ET FERMETURE

- **<u>Ouverture</u> : érosion puis dilatation. Peut permettre de réduire certains bruits : les pixels blancs isolés sont éliminés par érosion et ne sont pas restitués par dilatation. Alors que des pixels blancs connectés forment une zone réduite par érosion mais reconstruite ensuite par dilatation.**

- **<u>Fermeture</u> : dilatation puis érosion. Peut permettre de reconnecter des parties d'un objet ou d'un contour qui ont été déconnectés par exemple par un seuillage trop brusque. Peut aussi permettre de réduire certains bruits (pixels noirs isolés).**

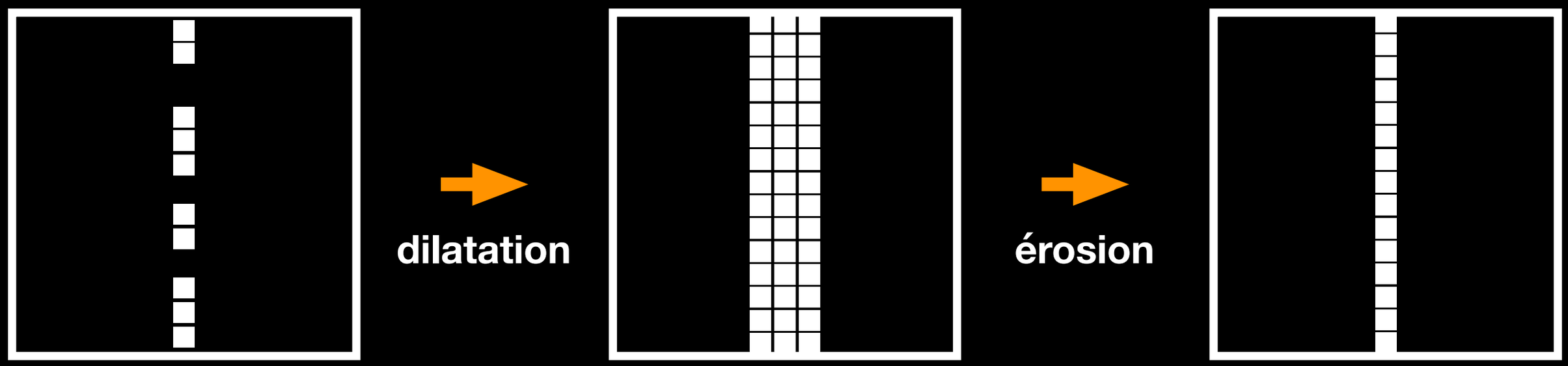
V. Morphologie mathématique

- **Ouverture**



V. Morphologie mathématique

- **Fermeture**



V. Morphologie mathématique

- **EXERCICE 4 : Reprendre l'image seuillée après filtrage de Prewitt de l'exercice 3 du Chapitre IV, puis éliminer les fausses alarmes par ouverture. Commenter sur les paramètres utilisés et les résultats obtenus.**

V. Morphologie mathématique

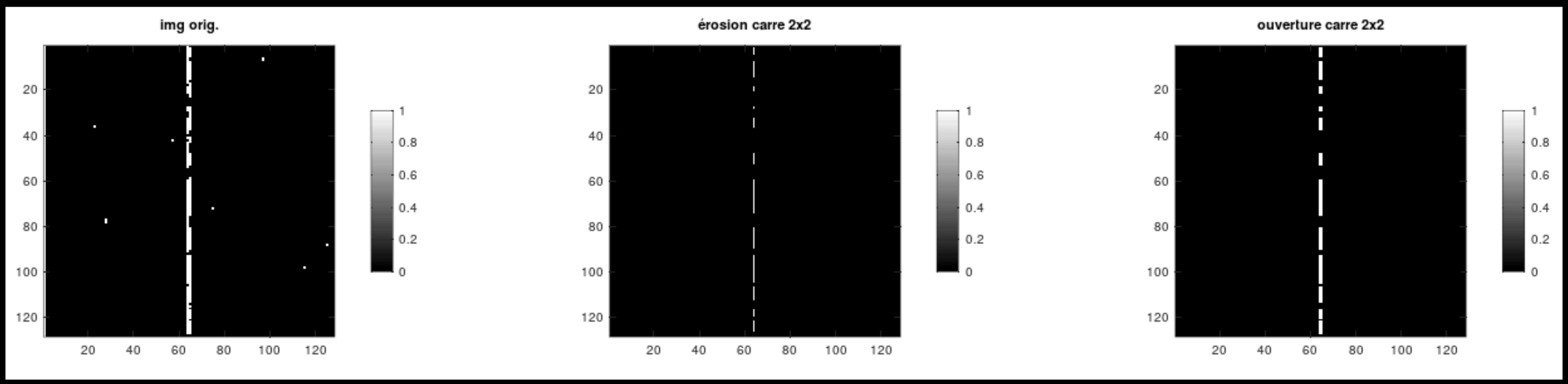
```
clear
close all


%---
% 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=filter2(Pv,J, 'same');
subplot(2,2,2), imagesc(JP), title('Prewitt'), colorbar, axis('square')
%---
% Sobel
Sh=fspecial('sobel'); Sv=-Sh'; JS=filter2(Sv,J, 'same');
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, 'same'); JRb=filter2(Rb,J, 'same');
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')
```

ligne de code ajoutée dans l'exo 3 du chap. IV

(instruction qui peut aussi être tapée en ligne de commande après avoir exécuté la routine telle qu'elle était déjà…)

V. Morphologie mathématique

```
clear
close all


load /Users/marcel/Documents/MATLAB/GBM/4-contours/image_Prewitt_seuil_new.mat
whos JPs

figure, colormap('gray')
subplot(1,3,1), imagesc(JPs), colorbar, axis('square'), title('img orig.')

carre2=strel('square', 2);
JPsE=imerode(JPs, carre2);
JPsED=imdilate(JPsE, carre2);

subplot(1,3,2), imagesc(JPsE), colorbar, axis('square'),
title('érosion carre 2x2')

subplot(1,3,3), imagesc(JPsED), colorbar, axis('square'),
title('ouverture carre 2x2')
```



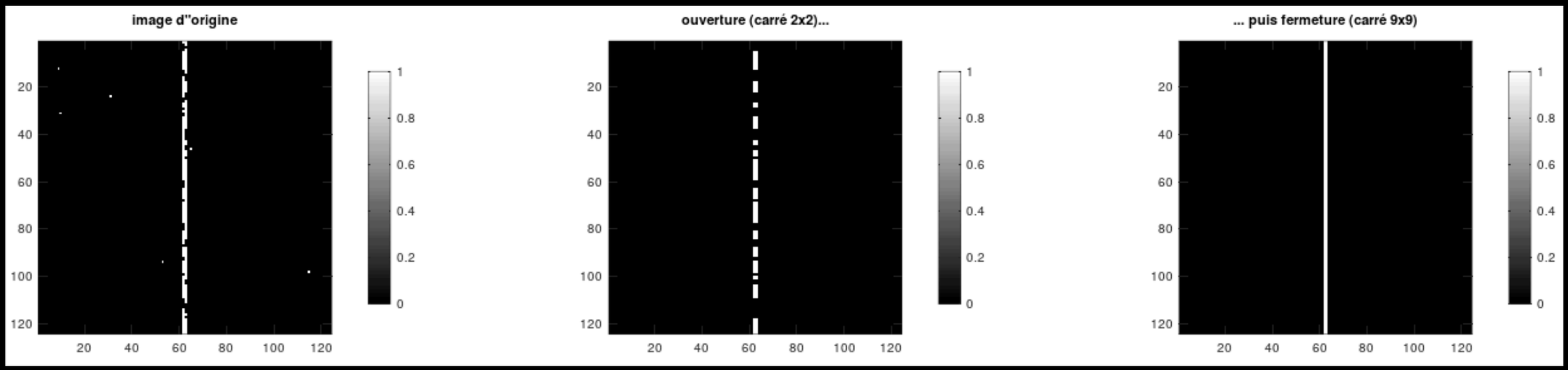
V. Morphologie mathématique

- **EXERCICE 5 : Quel est l'effet d'une fermeture sur l'image « ouverte » de l'exercice 4 ? (En fermant avec un carré assez grand pour refermer le contour cassé…)**

V. Morphologie mathématique

```
clear
close all

load /Users/marcel/Documents/MATLAB/GBM/4-contours/img_Prewitt_seuil.mat
whos JPs

figure, colormap(gray)

subplot(1,3,1), imagesc(JPs), colorbar, axis('square')
title('image d"origine')

carre2=strel('square', 2);
JPsE=imerode(JPs, carre2);
JPsED=imdilate(JPsE, carre2);

subplot(1,3,2), imagesc(JPsED), colorbar, axis('square')
title('ouverture (carré 2x2)...')

carrebig=strel('square', 9);
JPsEDD=imdilate(JPsED, carrebig);
JPsEDDE=imerode(JPsEDD, carrebig);

subplot(1,3,3), imagesc(JPsEDDE), colorbar, axis('square')
title('... puis fermeture (carré 9x9)')
```

V. Morphologie mathématique



V. Morphologie mathématique

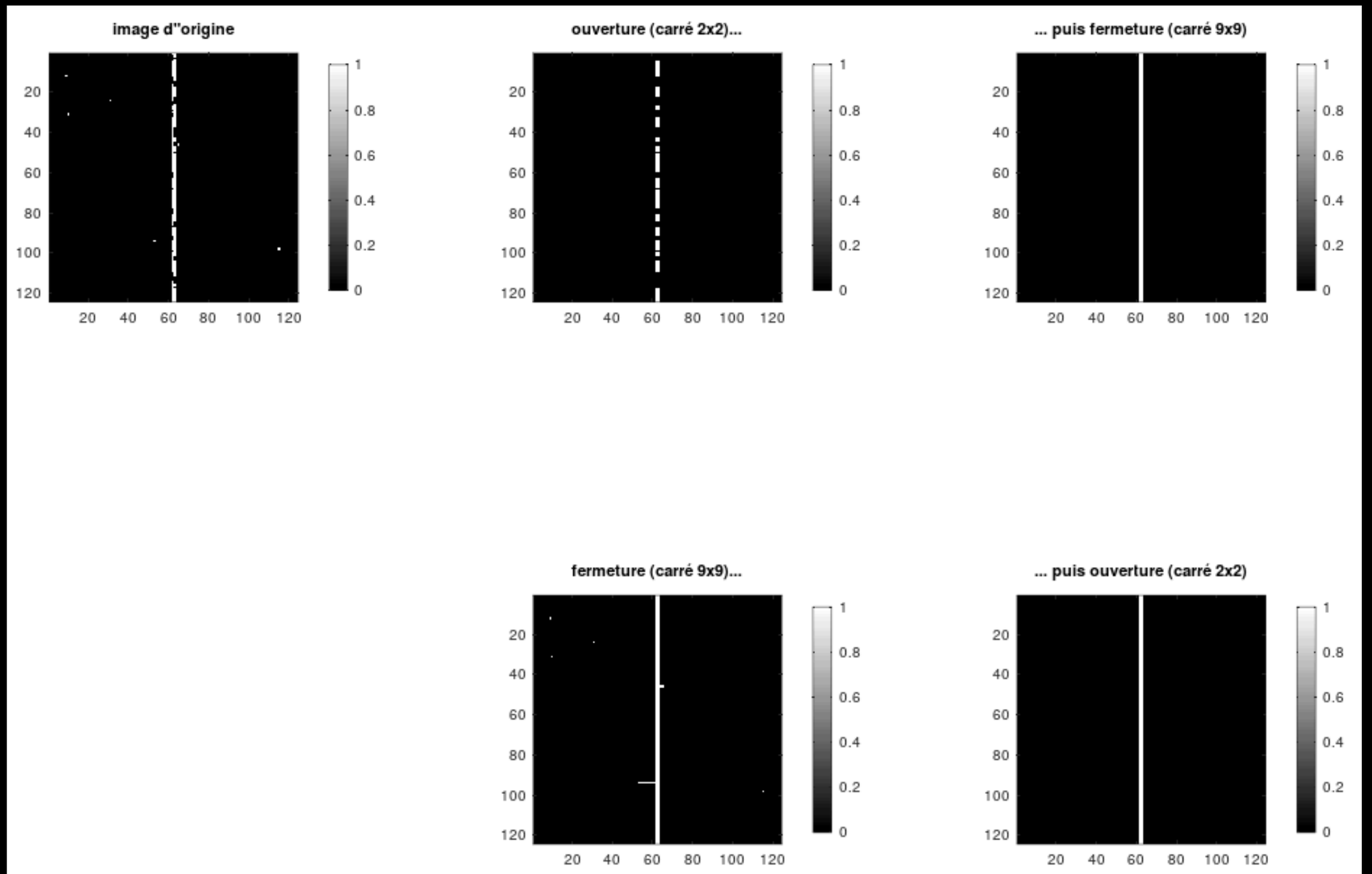
- **EXERCICE 6 : Comparer avec fermeture (la même que précédemment) puis ouverture (la même aussi).**

V. Morphologie mathématique

```
clear
close all

load /Users/marcel/Documents/MATLAB/GBM/4-contours/img_Prewitt_seuil.mat
whos JPs

figure, colormap(gray)

subplot(2,3,1), imagesc(JPs), colorbar, axis('square')
title('image d"origine')

carre2=strel('square', 2);
JPsE=imerode(JPs, carre2);
JPsED=imdilate(JPsE, carre2);

subplot(2,3,2), imagesc(JPsED), colorbar, axis('square')
title('ouverture (carré 2x2)...')

carrebig=strel('square', 9);
% voir 9 et 21...

JPsEDD=imdilate(JPsED, carrebig);
JPsEDDE=imerode(JPsEDD, carrebig);

subplot(2,3,3), imagesc(JPsEDDE), colorbar, axis('square')
title('... puis fermeture (carré 9x9)')

JPsD=imdilate(JPs, carrebig);
JPsDE=imerode(JPsD, carrebig);

subplot(2,3,5)
imagesc(JPsDE), colorbar, axis('square')
title('fermeture (carré 9x9)...')

JPsDEE=imerode(JPsDE, carre2);
JPsDEED=imdilate(JPsDEE, carre2);

subplot(2,3,6)
imagesc(JPsDEED), colorbar, axis('square')
title('... puis ouverture (carré 2x2)')
```

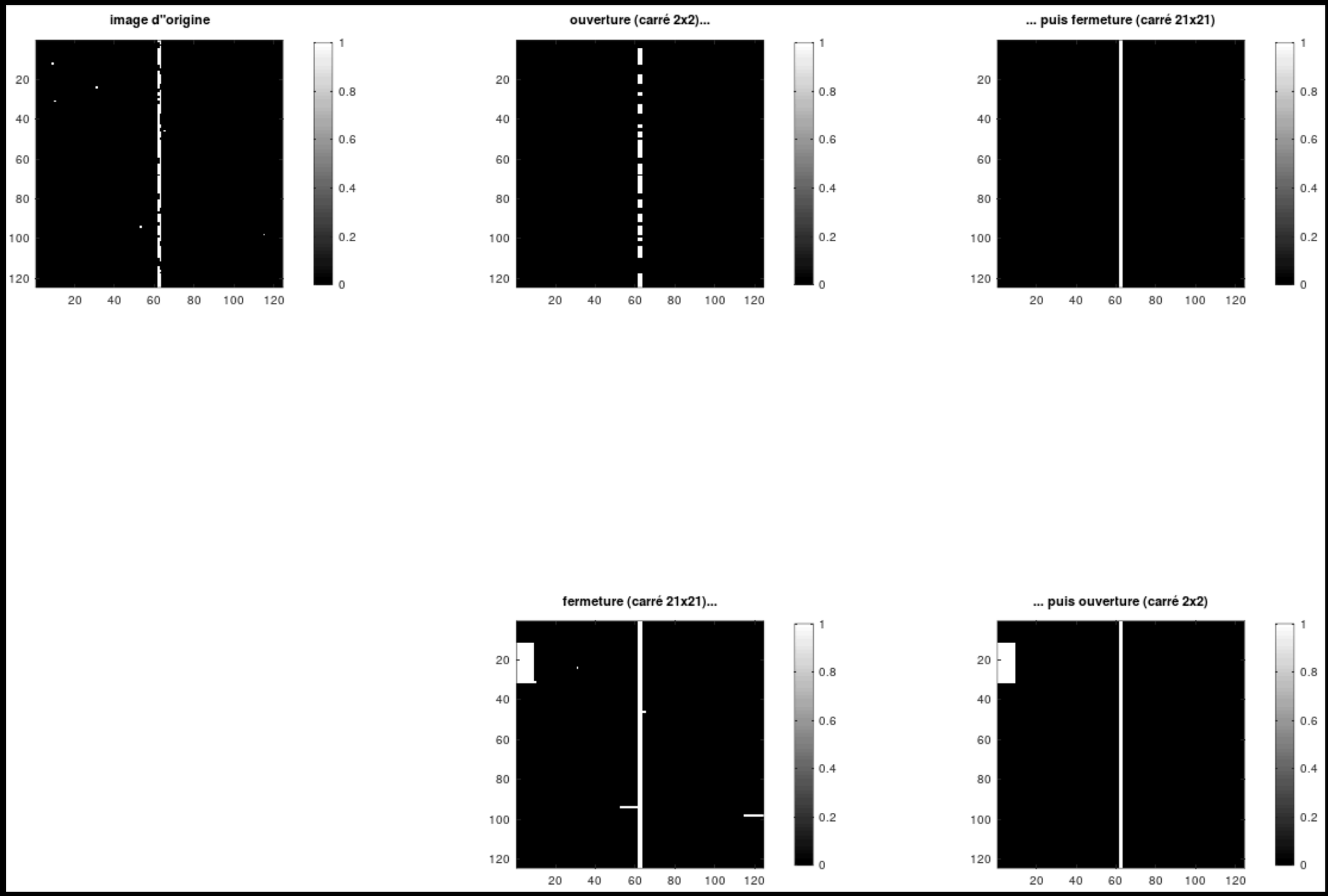
V. Morphologie mathématique

Si tout va bien :



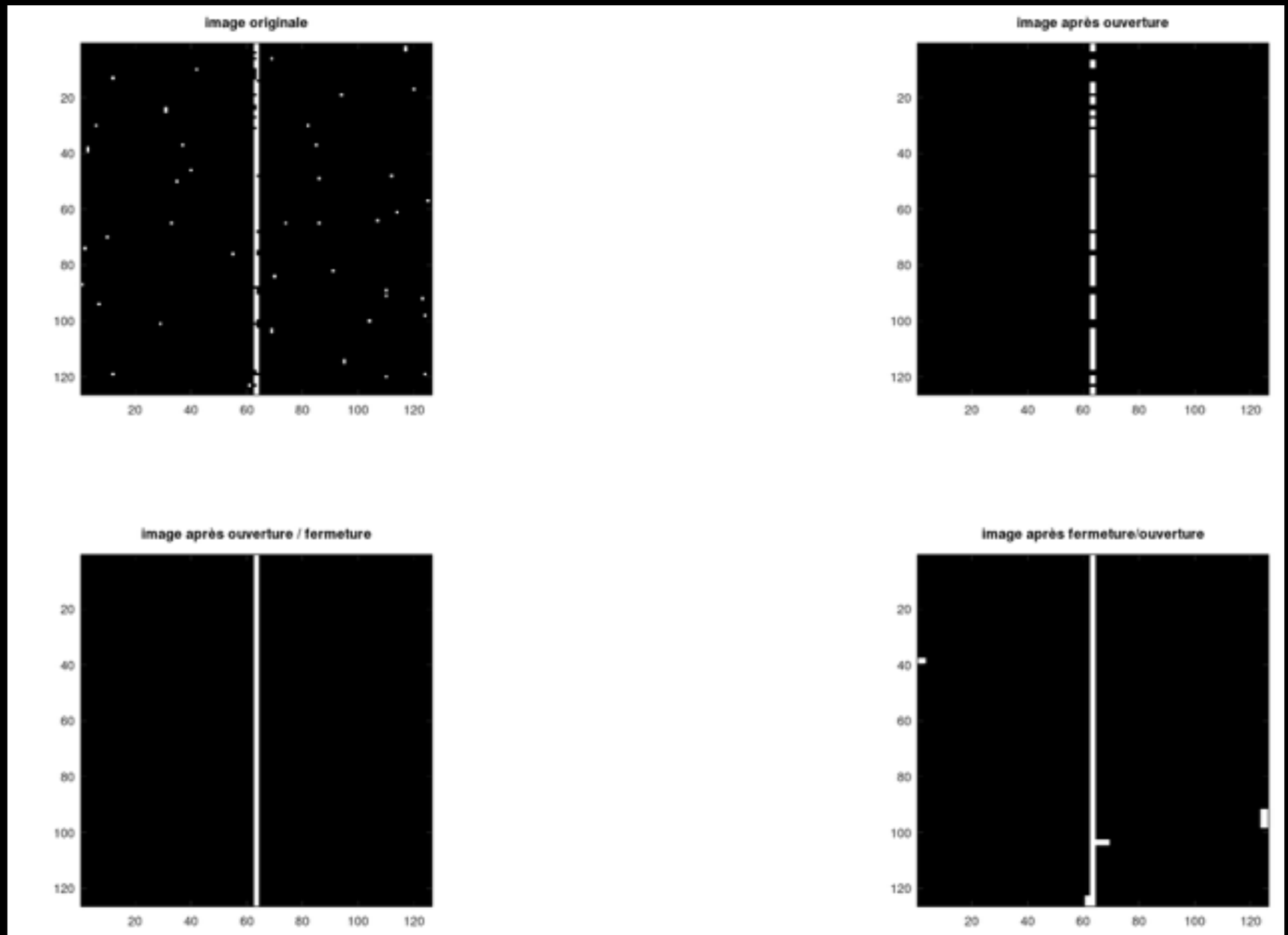
V. Morphologie mathématique

Mais le risque est le suivant :



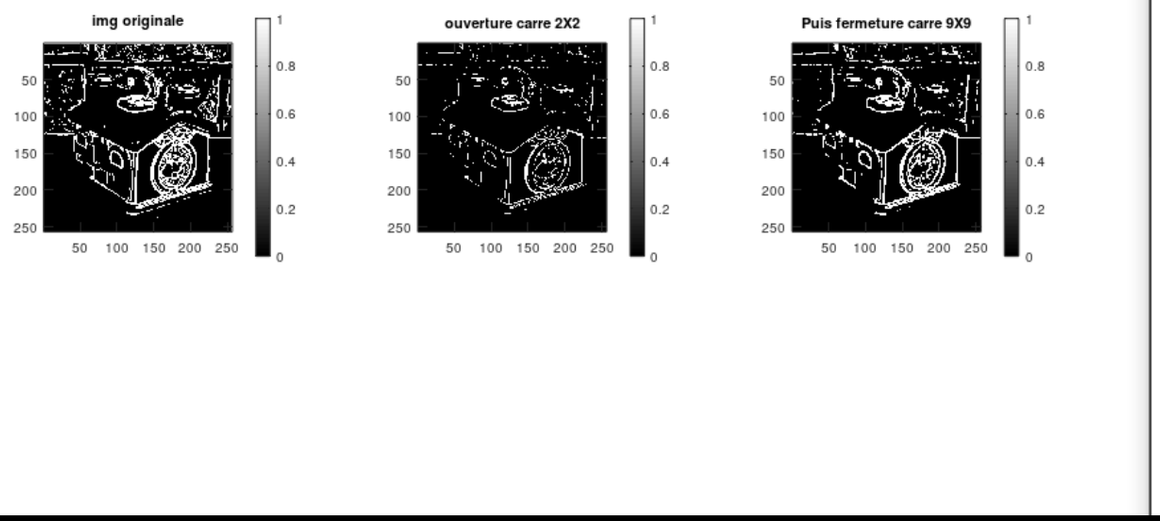
V. Morphologie mathématique

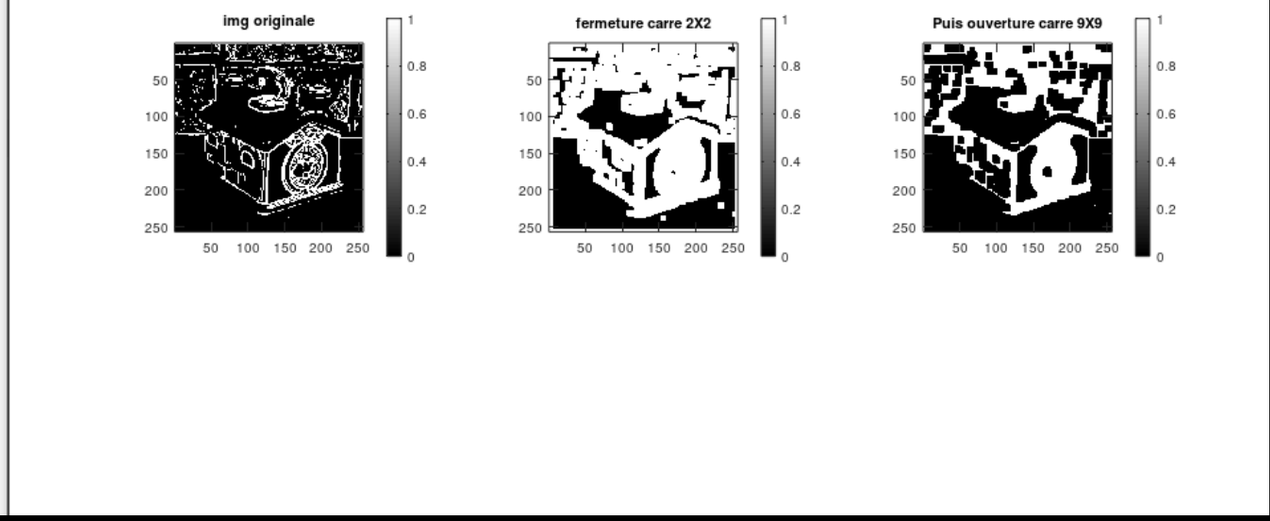
Autre exemple de problème possible :



V. Morphologie mathématique

Et ça peut être pire avec un objet plus complexe :





(ouverture puis fermeture à gauche, fermeture puis ouverture à droite)

V. Morphologie mathématique

4- REMARQUE

- **Il existe aussi les routines *imopen* et *imclose*… (-> help de Matlab/Octave).**

imopen

Morphologically open image

collapse all in page

Syntax

```
IM2 = imopen(IM,SE)
IM2 = imopen(IM,NHOOD)
gpuarrayIM2 = imopen(gpuarrayIM,__)
```

Description

`IM2 = imopen(IM,SE)` performs morphological opening on the grayscale or binary image `IM` with the structuring element `SE`. The argument `SE` must be a single structuring element object, as opposed to an array of objects. The morphological open operation is an erosion followed by a dilation, using the same structuring element for both operations.

`IM2 = imopen(IM,NHOOD)` performs opening with the structuring element `strel(NHOOD)`, where `NHOOD` is an array of 0s and 1s that specifies the structuring element neighborhood.

`gpuarrayIM2 = imopen(gpuarrayIM,__)` performs the operation on a graphics processing unit (GPU) with the structuring element `strel(NHOOD)`, if `NHOOD` is an array of 0s and 1s that specifies the structuring element neighborhood, or `strel(gather(NHOOD))` if `NHOOD` is a `gpuArray` object that specifies the structuring element neighborhood. This syntax requires the Parallel Computing Toolbox™.

Class Support

`IM` can be any numeric or logical class and any dimension, and must be nonsparse. If `IM` is logical, then `SE` must be flat.

`gpuarrayIM` must be a `gpuArray` of type `uint8` or `logical`. When used with a `gpuarray`, the structuring element must be flat and two-dimensional.

The output has the same class as the input.

imclose

Morphologically close image

collapse all in page

Syntax

```
IM2 = imclose(IM,SE)
IM2 = imclose(IM,NHOOD)
gpuarrayIM2 = imclose(gpuarraryIM,__)
```

Description

`IM2 = imclose(IM,SE)` performs morphological closing on the grayscale or binary image `IM`, returning the closed image, `IM2`. The structuring element, `SE`, must be a single structuring element object, as opposed to an array of objects. The morphological close operation is a dilation followed by an erosion, using the same structuring element for both operations.

`IM2 = imclose(IM,NHOOD)` performs closing with the structuring element `strel(NHOOD)`, where `NHOOD` is an array of 0's and 1's that specifies the structuring element neighborhood.

`gpuarrayIM2 = imclose(gpuarraryIM,__)` performs the operation on a graphics processing unit (GPU), where `gpuarrayIM` is a `gpuArray` containing the grayscale or binary image. `gpuarrayIM2` is a `gpuArray` of the same class as the input image. This syntax requires the Parallel Computing Toolbox™.

Class Support

`IM` can be any numeric or logical class and any dimension, and must be nonsparse. If `IM` is logical, then `SE` must be flat.

`gpuarrayIM` must be a `gpuArray` of type `uint8` or `logical`. When used with a `gpuarray`, the structuring element must be flat and two-dimensional.

The output has the same class as the input.

V. Morphologie mathématique

- **<u>Ouverture puis fermeture</u> : érosion puis dilatation, puis dilatation puis érosion. Permet la destruction des pixels isolés, même pour des images en niveaux de gris…**

  => réduction du bruit de type « poivre et sel » par morphologie mathématique !

V. Morphologie mathématique

- **EXERCICE 7 : Prendre l'image 'trees.jpg', la transformer en niveaux de gris, bruiter 5% des pixels par un bruit « poivre et sel », réduire ce bruit par ouverture puis fermeture. Avec *imerode* et *imdilate* d'une part et avec *imopen* et *imclose* d'autre part. (Élément structurant = carré 2x2.)**

```
clear
close all
pkg load image % Octave

% lire image
im='/Users/marcel/Documents/MATLAB/0-images/trees.jpg';
trees=imread(im);
trees=rgb2gray(trees); trees=double(trees)/255.;
whos trees

% bruiter
t_snp=imnoise(trees, 'salt & pepper', 0.05);
figure, colormap(gray)
subplot(3,2,1)
imagesc(trees), colorbar, axis('square'), title('image d"origine')
subplot(3,2,2)
imagesc(t_snp), colorbar, axis('square'), title('image bruitée')

% élément structurant
carre=strel('square', 2);

% érosion puis dilatation = ouverture
tE =imerode(t_snp, carre);
tED=imdilate(tE, carre);
subplot(3,2,3)
imagesc(tED), colorbar, axis('square'), title('erosion puis dilatation...')

% dilatation puis érosion = fermeture
tEDD =imdilate(tED, carre);
tEDDE=imerode(tEDD, carre);
subplot(3,2,4)
imagesc(tEDDE),colorbar,axis('square'),title('... puis re-dilat. puis re-eros.')

% ouverture avec imopen
t_op = imopen(t_snp, carre);
subplot(3,2,5)
imagesc(t_op), colorbar, axis('square'), title('ouverture avec imopen...')

% fermetrure avec imclose
t_cl = imclose(t_op, carre);
subplot(3,2,6)
imagesc(t_cl),colorbar,axis('square'),title('... puis fermeture avec imclose.')
```

image d''origine

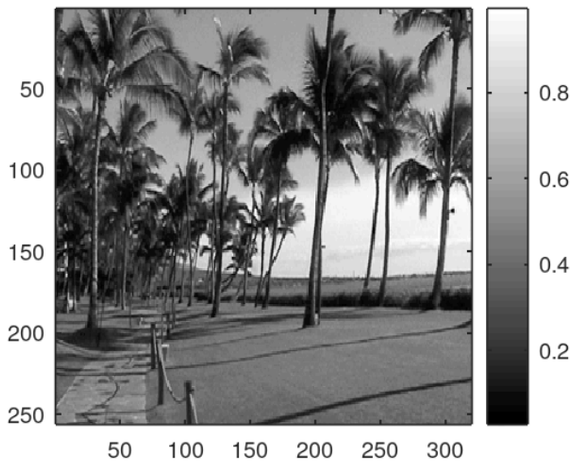
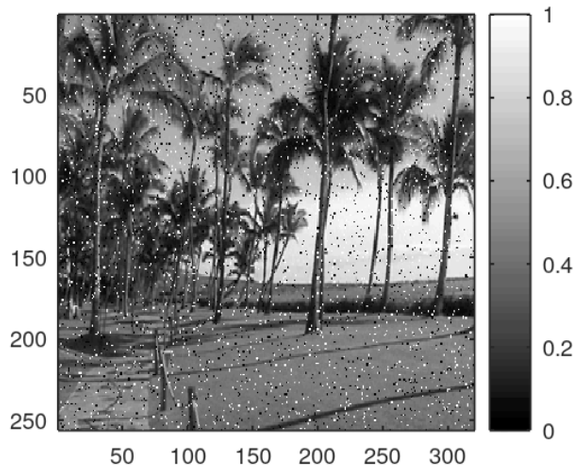
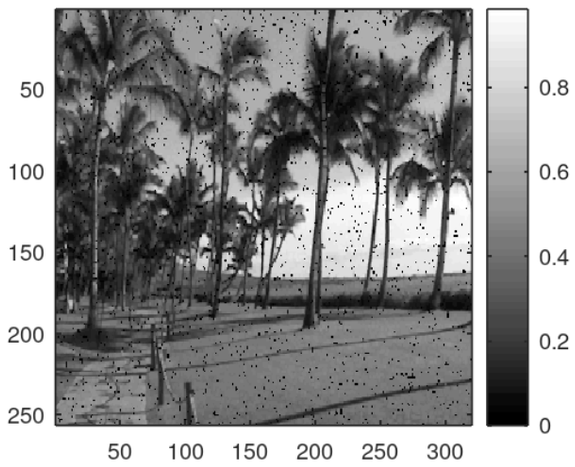


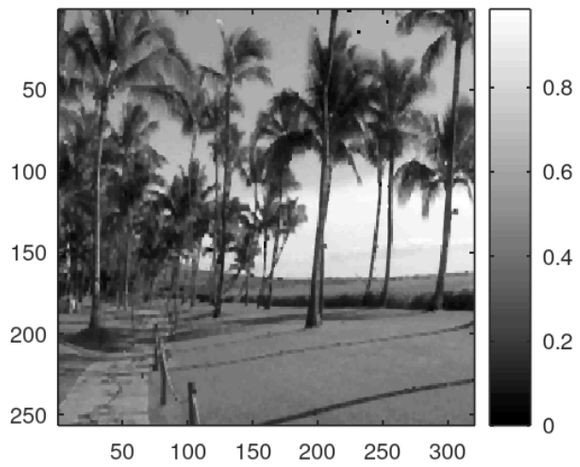
image bruitée



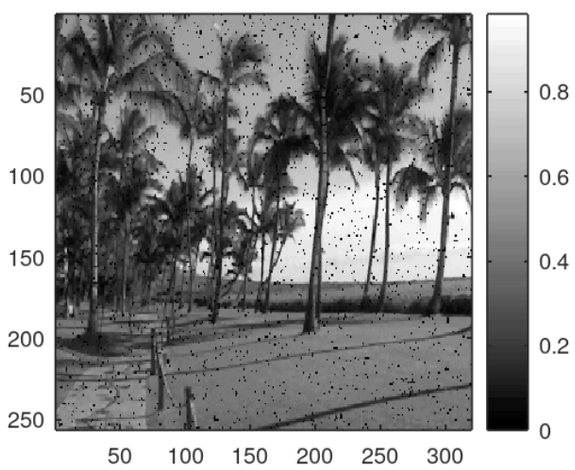
erosion puis dilatation...



... puis re-dilat. puis re-eros.



ouverture avec imopen...



... puis fermeture avec imclose.