

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

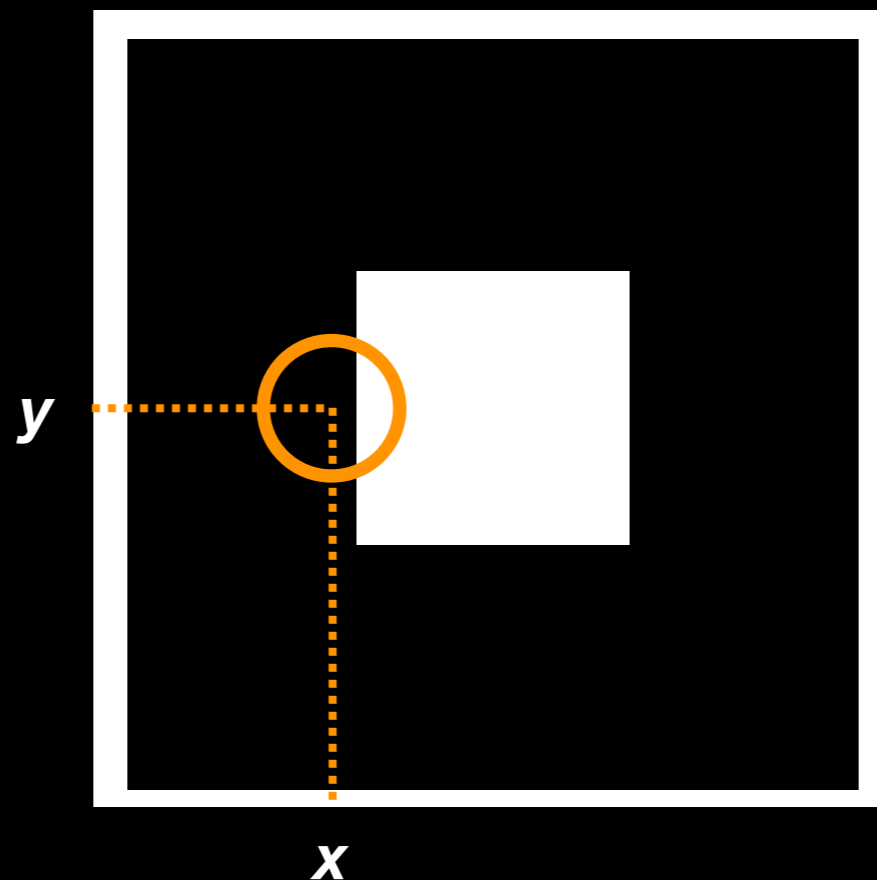
- Un élément structurant = un masque (comme pour un filtrage en passant par le plan de Fourier)...
- Dans le cas de l'érosion et de la dilatation :

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

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 maximale 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 minimale.
- Sous MATLAB/OCTAVE :

```
>> IE = imerode(I, S);  
>> ID = imdilate(I, S);
```

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

```
>> S = strel('disk', 10);
```

(ou `strel('disk', 10, n)`
si on veut une approximation \neq
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 rectangle 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

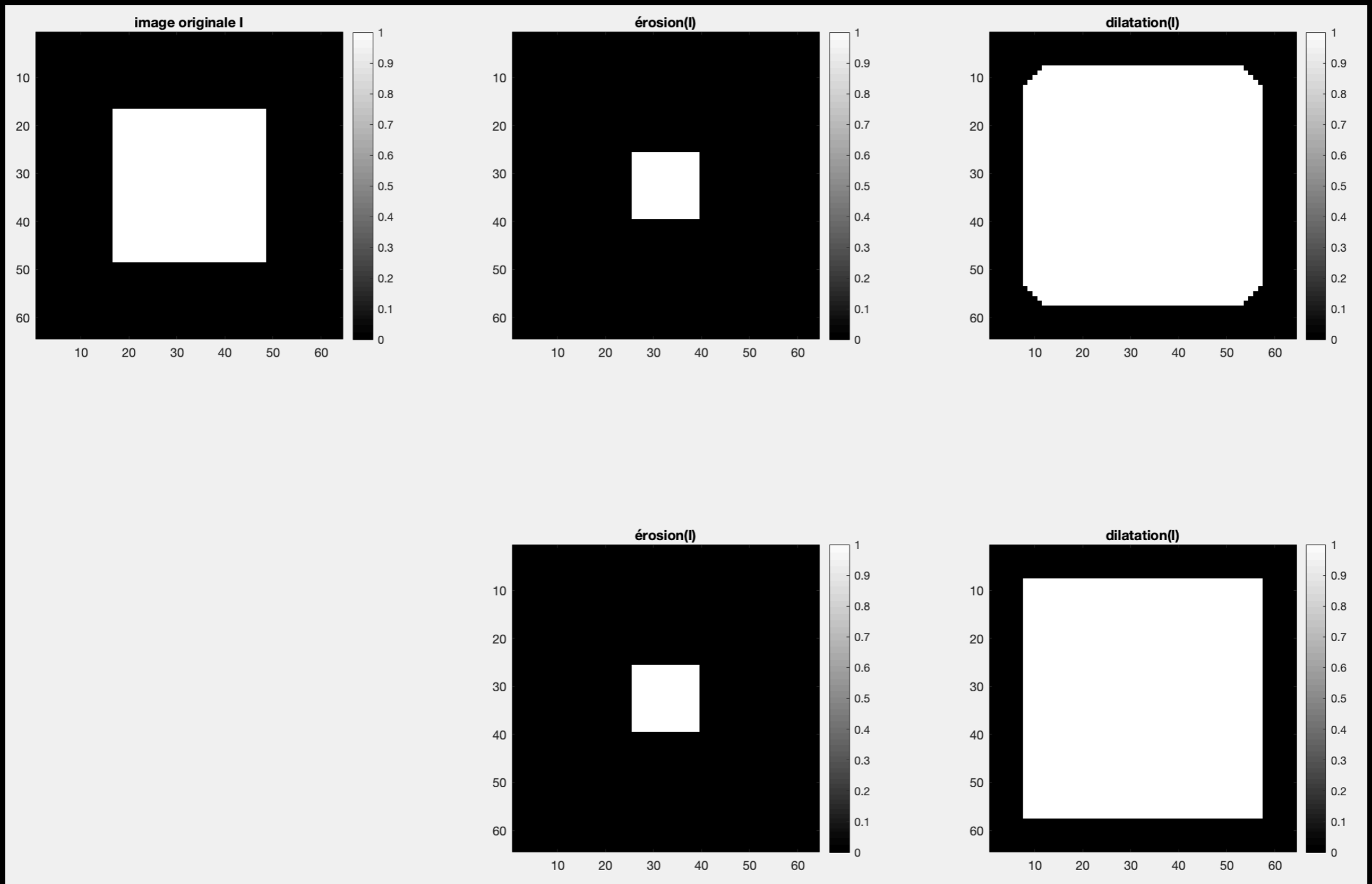
figure, colormap('gray')
subplot(1,3,1), imagesc(I), colorbar, axis('square'), title('image originale I')

% 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 I , toujours avec le même 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);
```

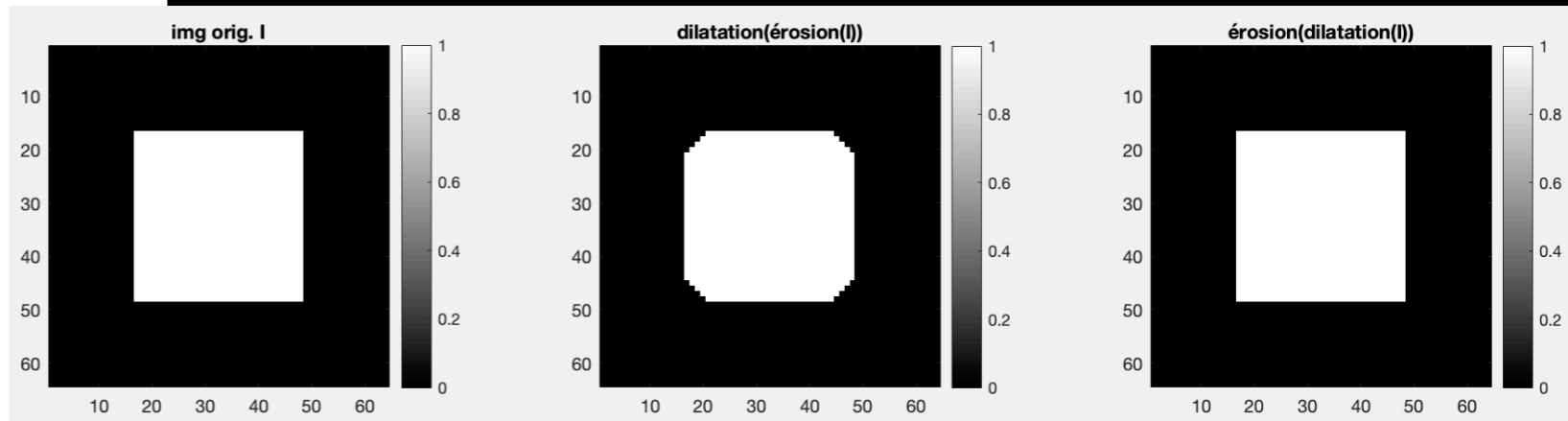
```
% dilatation(érosion(I)) vs. érosion(dilatation(I))
```

```
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')

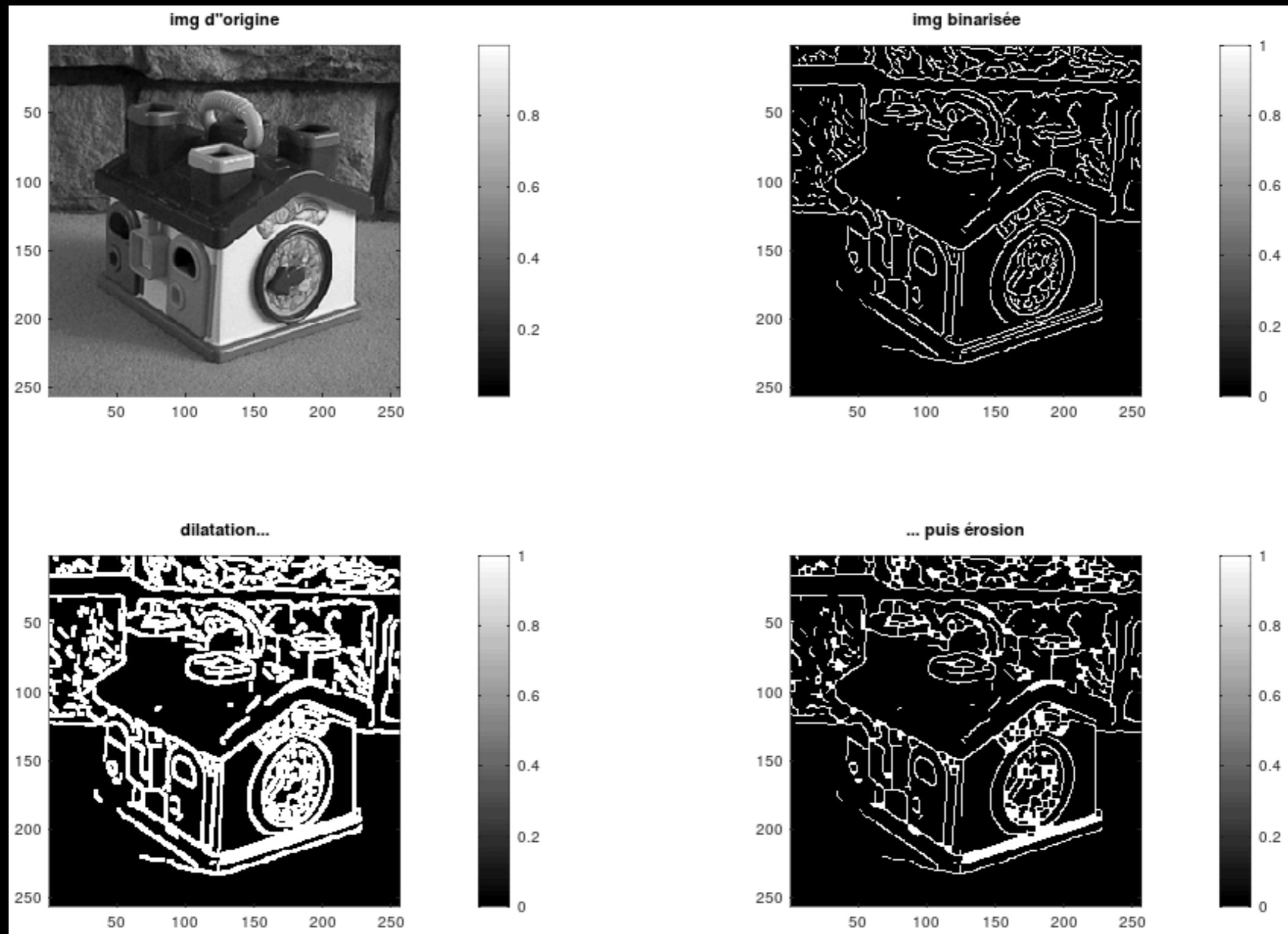
subplot(2,2,1), imagesc(house), colorbar
axis('square'), title('img d"origine')

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



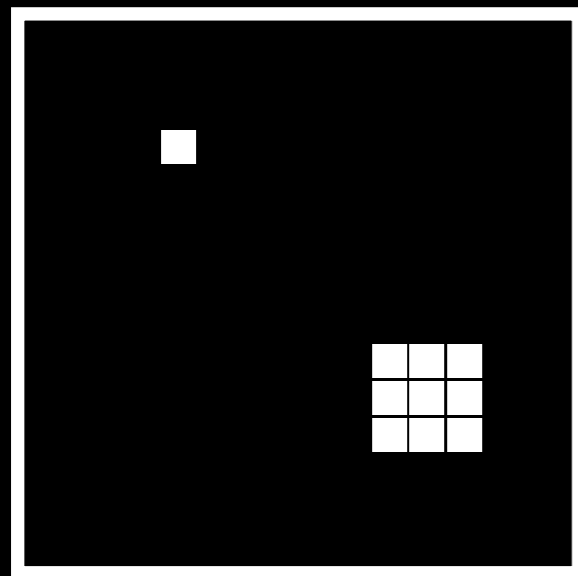
V. Morphologie mathématique

3- OUVERTURE ET FERMETURE

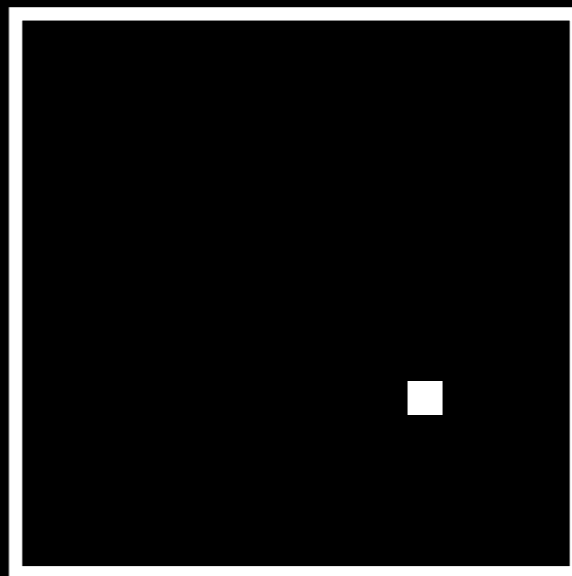
- **Ouverture** : érosion puis dilatation. Peut permettre de réduire certains bruits : les pixels blancs isolés sont éliminés par érosion et ne sont pas restitués par dilatation. Alors que des pixels blancs connectés forment une zone réduite par érosion mais reconstruite ensuite par dilatation.
- **Fermeture**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

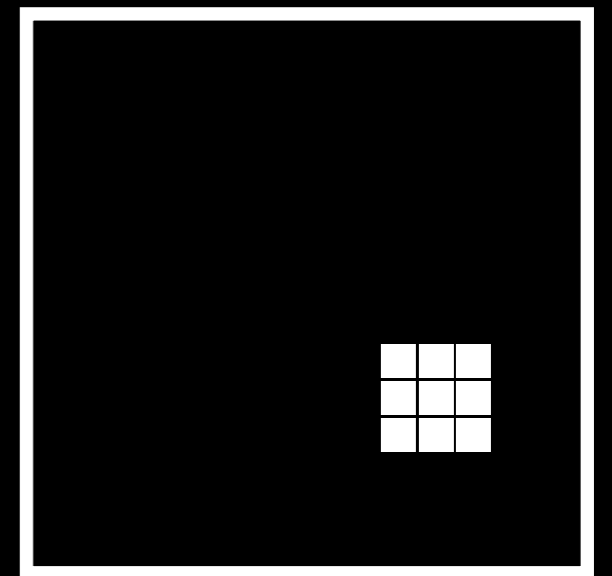
- Ouverture



→
érosion

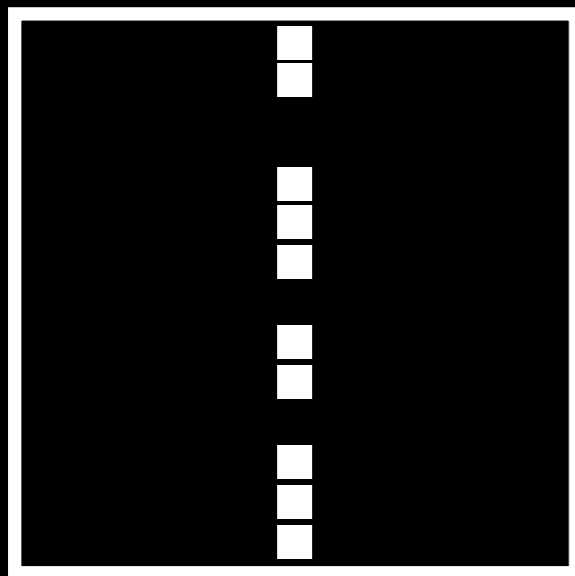


→
dilatation

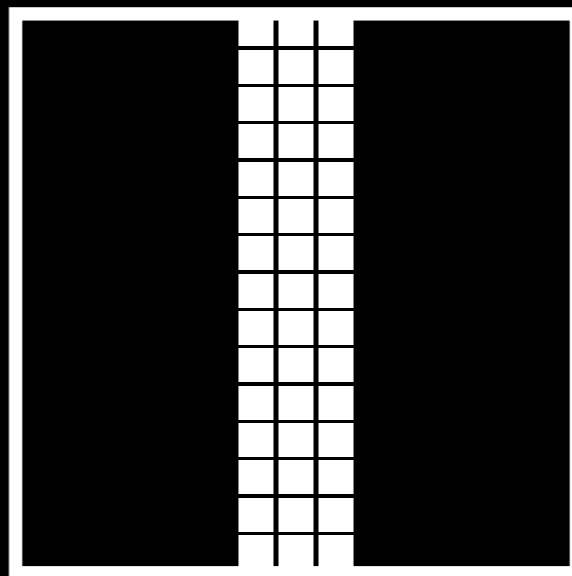


V. Morphologie mathématique

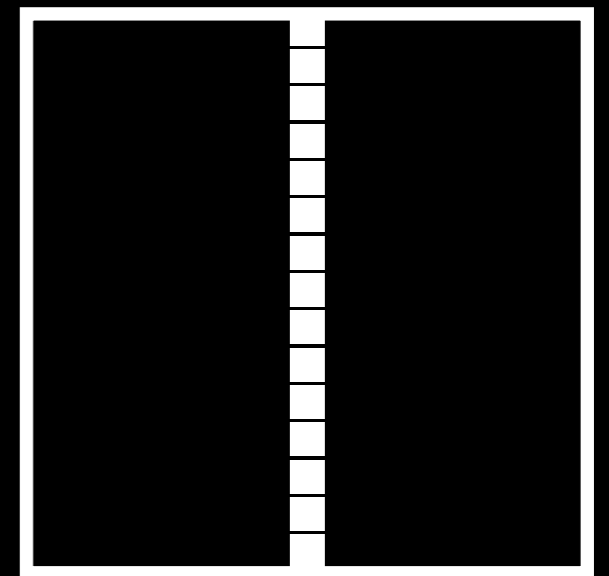
- Fermeture



→
dilatation



→
érosion



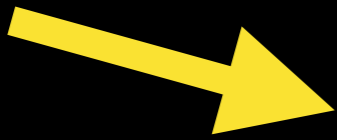
V. Morphologie mathématique

- **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

```
1 clear
2 close all
3
4
5 %---
6 % load image from disk
7 load image_bruit, whos J
8 % figure 1
9 figure(1), colormap(gray)
10 subplot(2,2,1), imagesc(J), title('image test'), colorbar, axis('square')
11 %---
12 % Prewitt
13 Ph=fspecial('prewitt'); Pv=-Ph'; JP=filter2(Pv,J, 'same');
14 subplot(2,2,2), imagesc(JP), title('Prewitt'), colorbar, axis('square')
15 %---
16 % Sobel
17 Sh=fspecial('sobel'); Sv=-Sh'; JS=filter2(Sv,J, 'same');
18 subplot(2,2,3), imagesc(JS), title('Sobel'), colorbar, axis('square')
19 %---
20 % Roberts
21 Ra=[1 0;0 -1]; Rb=rot90(Ra,-1);
22 JRa=filter2(Ra,J, 'same'); JRb=filter2(Rb,J, 'same');
23 JR=sqrt(JRa.*JRa+JRb.*JRb);
24 subplot(2,2,4), imagesc(JR), title('Roberts'), colorbar, axis('square')
25 %---
26 % comparaison des images uniques de contours seuillées
27 % seuillages
28 seuilP=.25; JPs=JP>seuilP;
29 % - ligne ajoutée pour l'exo 4 chapitre 5
30 save image_Prewitt_seuil_new.mat JPs
31 % -
32 seuils=.362; JSs=JS>seuils;
33 seuilR=.175; JRs=JR>seuilR;
34 % figure 2
35 figure(2), colormap(gray)
36 imagesc(JPs), title(['Prewitt+seuil=',num2str(seuilP)];'=> 9 FA')), colorbar
37 axis('square')
38 % figure 3
39 figure(3), colormap(gray)
40 imagesc(JSs), title(['Sobel+seuil=',num2str(seuils)] ;'=> 9 FA')), colorbar
41 axis('square')
42 % figure 4
43 figure(4), colormap(gray)
44 imagesc(JRs), title(['Roberts+seuil=',num2str(seuilR)];'=> 9 FA')), colorbar
45 axis('square')
```

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

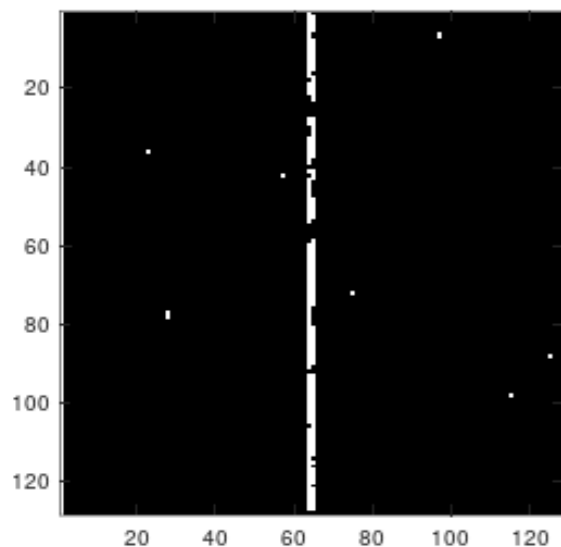


(instruction qui peut aussi
être tapée en ligne de
commande après avoir
exécuté la routine telle
qu'elle était déjà...)

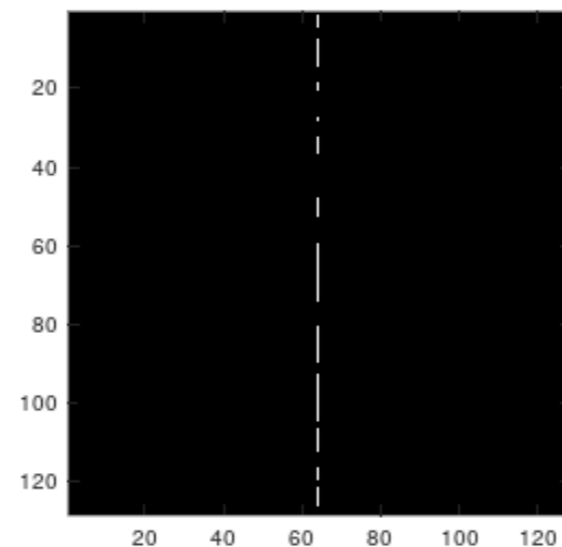
V. Morphologie mathématique

```
1 clear
2 close all
3
4
5 load /Users/marcel/Documents/MATLAB/GBM/4-contours/image_Prewitt_seuil_new.mat
6 whos JPs
7
8 figure, colormap('gray')
9 subplot(1,3,1), imagesc(JPs), colorbar, axis('square'), title('img orig.')
10
11 carre2=strel('square', 2);
12 JPsE=imerode(JPs, carre2);
13 JPsED=imdilate(JPsE, carre2);
14
15 subplot(1,3,2), imagesc(JPsE), colorbar, axis('square'),
16 title('érosion carre 2x2')
17
18 subplot(1,3,3), imagesc(JPsED), colorbar, axis('square'),
19 title('ouverture carre 2x2')
```

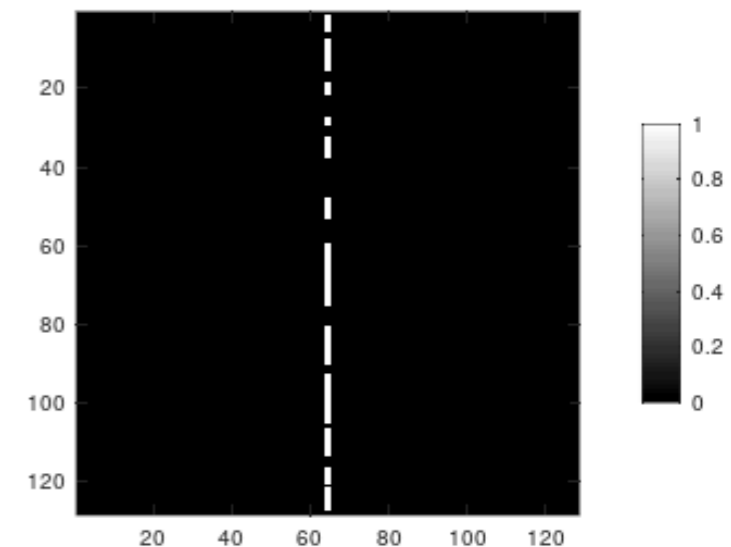
img orig.



érosion carre 2x2



ouverture carre 2x2



V. Morphologie mathématique

- Ouverture puis fermeture : 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

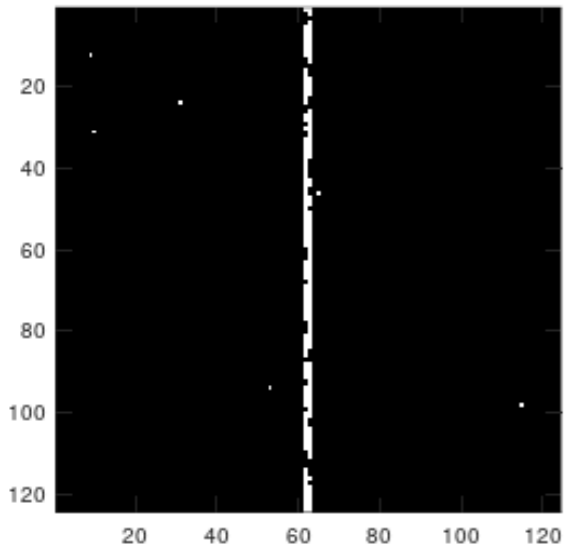
- **EXERCICE 5** : Quel est l'effet d'une fermeture sur l'image « ouverte » de l'exercice 4 ? (En fermant avec un carré assez grand pour refermer le contour cassé...)

V. Morphologie mathématique

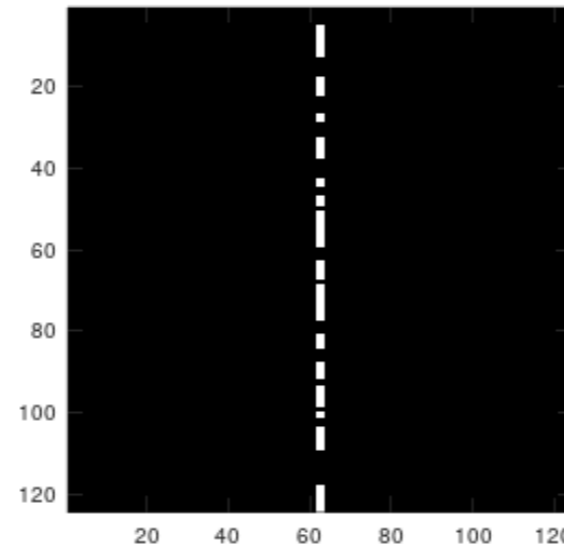
```
1 clear
2 close all
3
4
5 load /Users/marcel/Documents/MATLAB/GBM/4-contours/img_Prewitt_seuil.mat
6 whos JPs
7
8 figure, colormap(gray)
9
10 subplot(1,3,1), imagesc(JPs), colorbar, axis('square')
11 title('image d"origine')
12
13 carre2=strel('square', 2);
14 JPsE=imerode(JPs, carre2);
15 JPsED=imdilate(JPsE, carre2);
16
17 subplot(1,3,2), imagesc(JPsED), colorbar, axis('square')
18 title('ouverture (carré 2x2)...')
19
20 carrebig=strel('square', 9);
21 JPsEDD=imdilate(JPsED, carrebig);
22 JPsEDDE=imerode(JPsEDD, carrebig);
23
24 subplot(1,3,3), imagesc(JPsEDDE), colorbar, axis('square')
25 title('... puis fermeture (carré 9x9)')
```

V. Morphologie mathématique

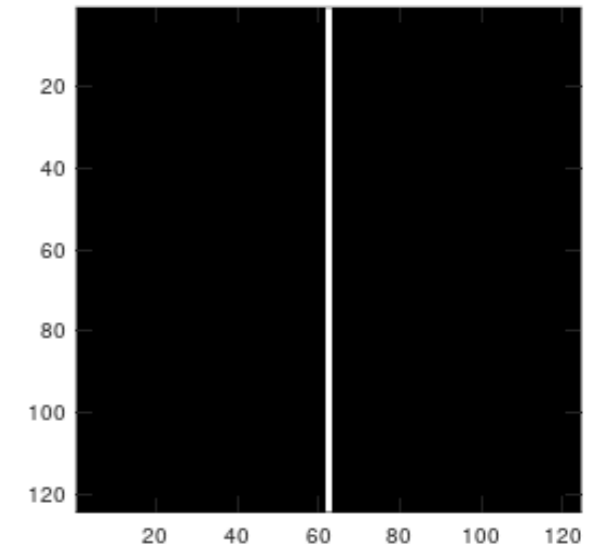
image d'origine



ouverture (carré 2x2)...



... puis fermeture (carré 9x9)



V. Morphologie mathématique

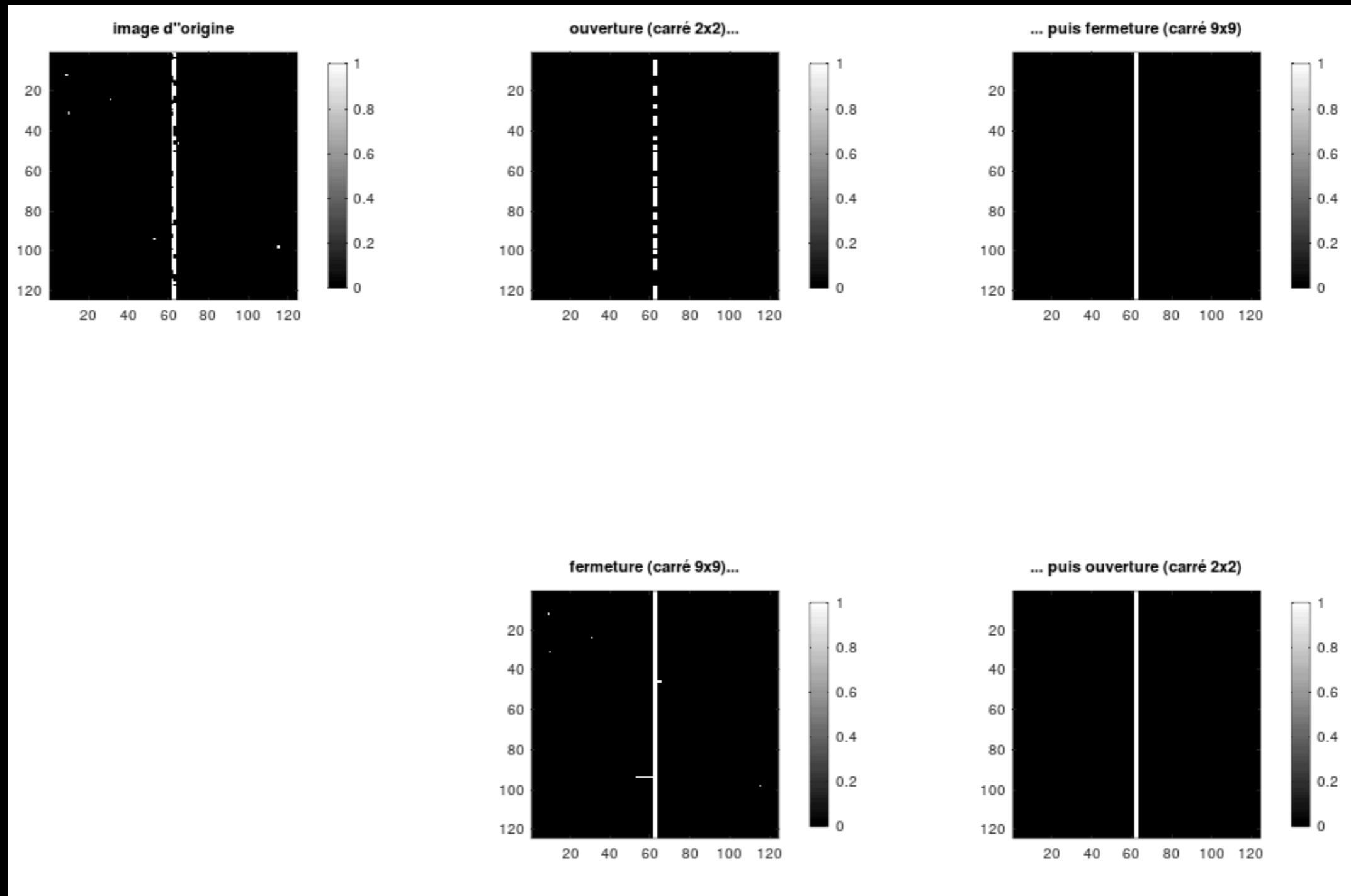
- **EXERCICE 6** : Comparer avec fermeture (la même que précédemment) puis ouverture (la même aussi).

V. Morphologie mathématique

```
1 clear
2 close all
3 f
4
5 load /Users/marcel/Documents/MATLAB/GBM/4-contours/img_Prewitt_seuil.mat
6 whos JPs
7
8 figure, colormap(gray)
9
10 subplot(2,3,1), imagesc(JPs), colorbar, axis('square')
11 title('image d"origine')
12
13 carre2=strel('square', 2);
14 JPsE=imerode(JPs, carre2);
15 JPsED=imdilate(JPsE, carre2);
16
17 subplot(2,3,2), imagesc(JPsED), colorbar, axis('square')
18 title('ouverture (carré 2x2)...')
19
20 carrebig=strel('square', 9);
21 % voir 9 et 21...
22
23 JPsEDD=imdilate(JPsED, carrebig);
24 JPsEDDE=imerode(JPsEDD, carrebig);
25
26 subplot(2,3,3), imagesc(JPsEDDE), colorbar, axis('square')
27 title('... puis fermeture (carré 9x9)')
28
29 JPsD=imdilate(JPs, carrebig);
30 JPsDE=imerode(JPsD, carrebig);
31
32 subplot(2,3,5)
33 imagesc(JPsDE), colorbar, axis('square')
34 title('fermeture (carré 9x9)...')
35
36 JPsDEE=imerode(JPsDE, carre2);
37 JPsDEED=imdilate(JPsDEE, carre2);
38
39 subplot(2,3,6)
40 imagesc(JPsDEED), colorbar, axis('square')
41 title('... puis ouverture (carré 2x2)')
```

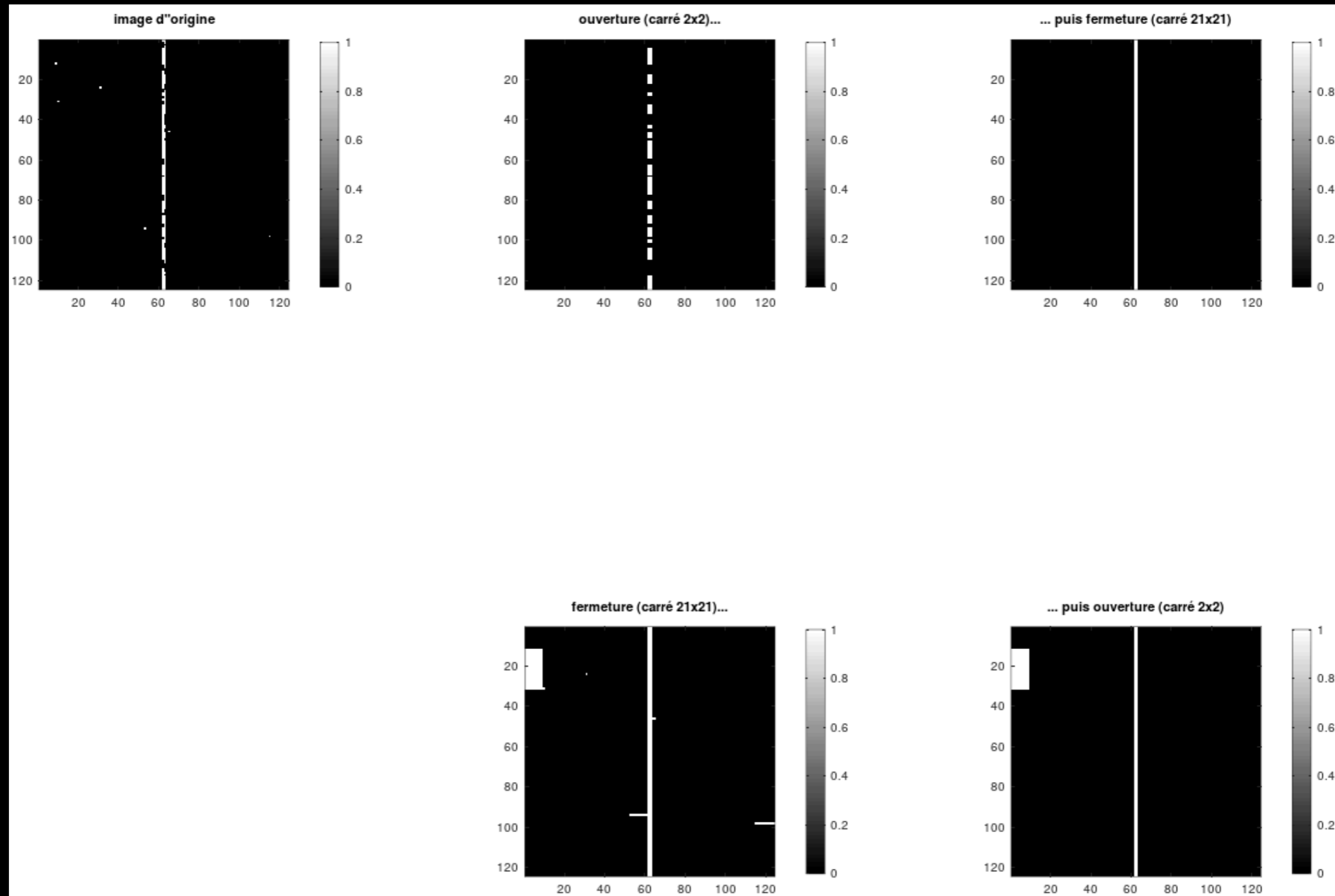

V. Morphologie mathématique

Si tout va bien :



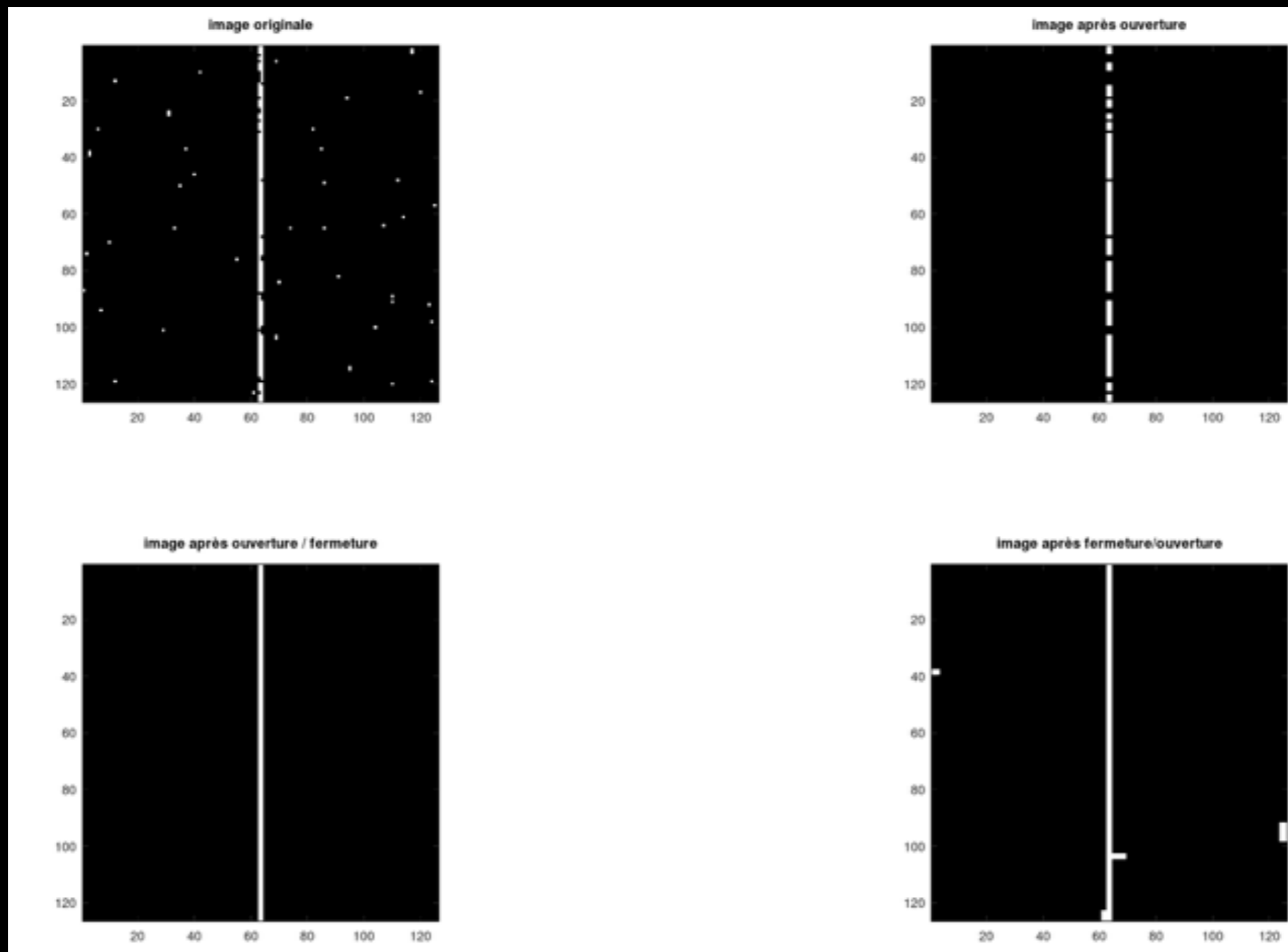
V. Morphologie mathématique

Mais le risque est le suivant :



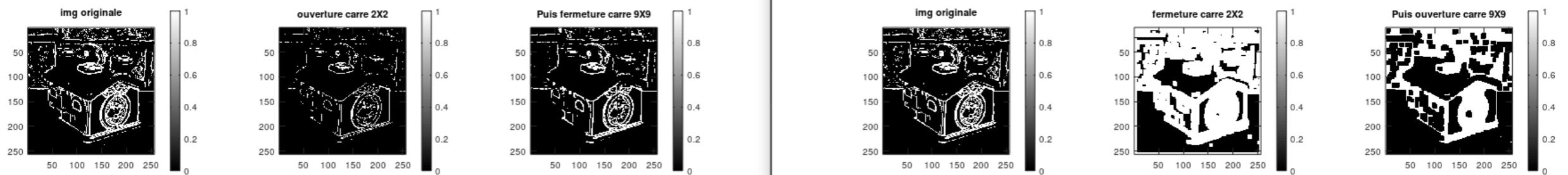
V. Morphologie mathématique

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,NH00D)
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,NH00D)` performs opening with the structuring element `strel(NH00D)`, where `NH00D` 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(NH00D)`, if `NH00D` is an array of 0s and 1s that specifies the structuring element neighborhood, or `strel(gather(NH00D))` if `NH00D` 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,NH00D)
gpuarrayIM2 = imclose(gpuarrayIM, __)
```

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,NH00D)` performs closing with the structuring element `strel(NH00D)`, where `NH00D` is an array of 0's and 1's that specifies the structuring element neighborhood.

`gpuarrayIM2 = imclose(gpuarrayIM, __)` 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

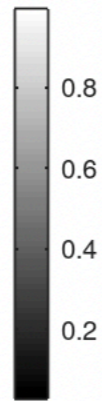
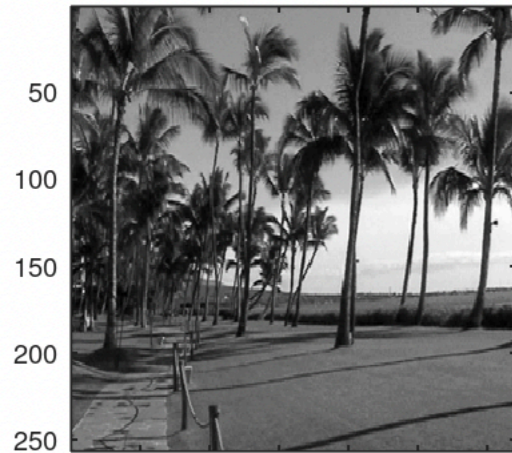
- **EXERCICE 7** : Prendre l'image 'trees.jpg', la transformer en niveaux de gris, bruitez 5% des pixels par un bruit « poivre et sel », réduire ce bruit par ouverture puis fermeture. Avec *imerode* et *imdilate* d'une part et avec *imopen* et *imclose* d'autre part. (Élément structurant = carré 2x2.)

V. Morphologie mathématique

```
5 % lire image
6 im='/Users/marcel/Documents/MATLAB/GBM/0-images/trees.jpg';
7 trees=imread(im);
8 trees=rgb2gray(trees); trees=double(trees)/255.;
9 whos trees
10
11 % bruiteur
12 t_snp=imnoise(trees, 'salt & pepper', 0.05);
13 figure, colormap(gray)
14 subplot(2,3,1)
15 imagesc(trees), colorbar, axis('square'), title('image d'origine')
16 subplot(2,3,2)
17 imagesc(t_snp), colorbar, axis('square'), title('image bruitée')
18
19 % élément structurant
20 carre=strel('square', 2);
21
22 % érosion puis dilatation = ouverture
23 tE=imerode(t_snp, carre);
24 tED=imdilate(tE, carre);
25 subplot(2,3,3)
26 imagesc(tED), colorbar, axis('square'), title('érosion puis dilatation...')
27
28 % dilatation puis érosion = fermeture
29 tEDD=imdilate(tED, carre);
30 tEDDE=imerode(tEDD, carre);
31 subplot(2,3,4)
32 imagesc(tEDDE), colorbar, axis('square'), title('... puis re-dilat. puis re-éros.')
33
34 % ouverture avec imopen
35 t_op = imopen(t_snp, carre);
36 subplot(2,3,5)
37 imagesc(t_op), colorbar, axis('square'), title('ouverture avec imopen...')
38
39 % fermeture avec imclose
40 t_cl = imclose(t_op, carre);
41 subplot(2,3,6)
42 imagesc(t_cl), colorbar, axis('square'), title('... puis fermeture avec imclose.')
```

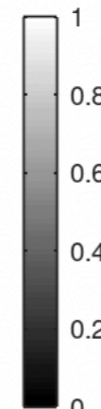
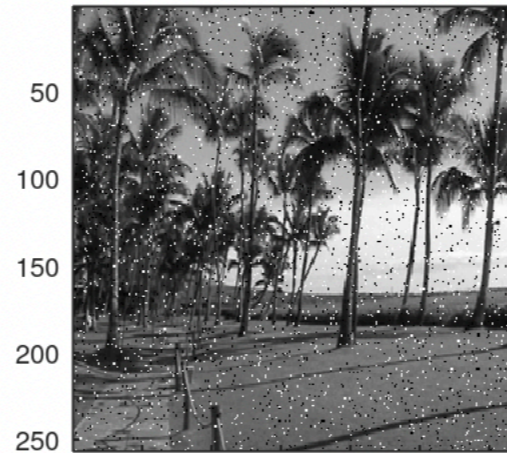
V. Morphologie mathématique

image d'origine



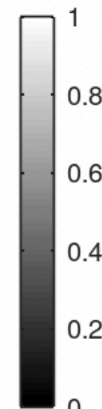
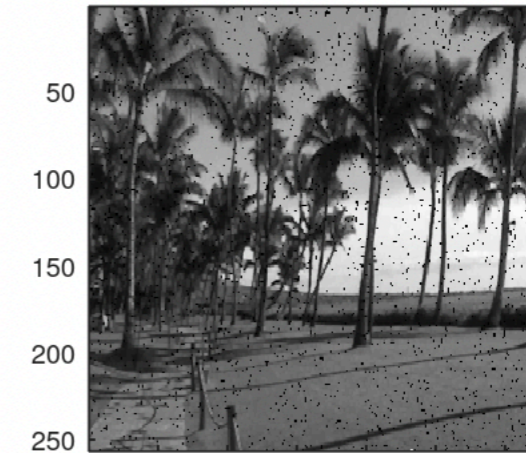
50 100 150 200 250 300

image bruitée



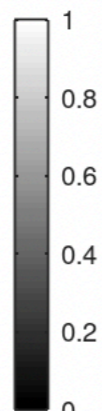
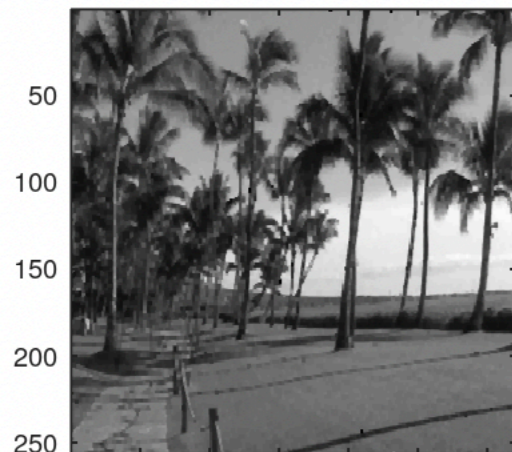
50 100 150 200 250 300

erosion puis dilatation...



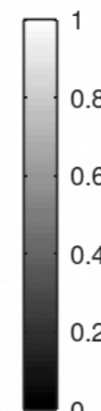
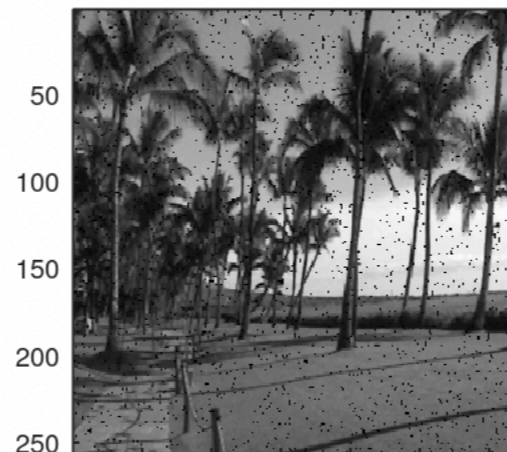
50 100 150 200 250 300

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



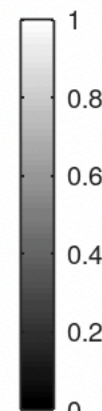
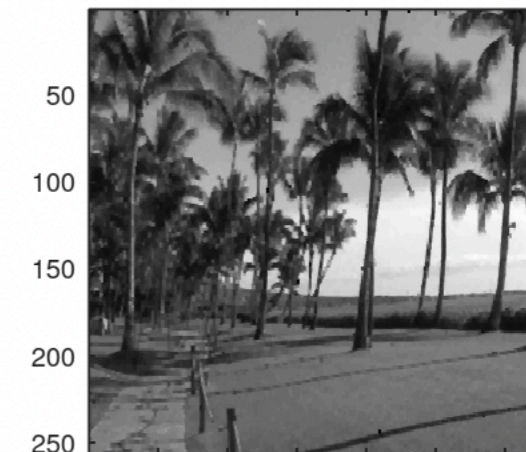
50 100 150 200 250 300

ouverture avec imopen...



50 100 150 200 250 300

... puis fermeture avec imclose.



50 100 150 200 250 300