Exercice 1

1-

Le premier pixel et le dernier pixel (//) ne sont pas traités.

i1 : /255/,255,255,255,2,2,255,255,255,/255/

i2 : /255/,255,1,0,0,0,0,0,255,/255/

Exercice 2

Dilatation = OU

Erosion = ET

Transposé = symétrie centrale de l’élément structurant.

On cherche la transposée, et on applique dilatation/érosion.

Exercice 4

[Lena,L] = imread('Images/lena.tif');

Lena\_ps = imnoise(Lena,"salt & pepper",0.1);

imshow(Lena)

imshow(Lena\_ps)

Lena\_M = Median(Lena);

imshow(Lena\_M)

Lena\_ps\_M = Median(Lena\_ps);

imshow(Lena\_ps\_M)

%imshow(Median(imnoise(Median(imnoise(Median(imnoise(Median(imnoise(Median(Lena),'salt & pepper',0.1)),'salt & pepper',0.1)),'salt & pepper',0.1)),'salt & pepper',0.1)))

%imshow(Median(Median(Median(Median(Lena)))))

Exercice 5

[Morpho,L\_m] = imread('Images\Morpho.tif');

imshow(Morpho)

Morpho\_dilat = Dilat(Morpho);

Morpho\_erod = Erod(Morpho);

imshow(Morpho\_dilat)

imshow(Morpho\_erod)

Morpho\_dilat\_5 = DilatN(Morpho,5);

Morpho\_erod\_5 = ErodN(Morpho,5);

imshow(Morpho\_dilat\_5)

imshow(Morpho\_erod\_5)

Exercice 6

Morpho\_ouvert = Ouverture(Morpho);

imshow(Morpho\_ouvert)

Morpho\_ferme = Fermeture(Morpho);

imshow(Morpho\_ferme)

Morpho\_ouvert\_5 = OuvertureN(Morpho,5);

imshow(Morpho\_ouvert\_5)

Morpho\_ferme\_5 = FermetureN(Morpho,5);

imshow(Morpho\_ferme\_5)

Morpho\_ouver\_ferme\_3 = FermetureN(OuvertureN(Morpho,3),3);

imshow(Morpho\_ouver\_ferme\_3)

Exercice 7

%Gradiant Interne = Origine - Erode

%Gradiant Externe = Dilat - Origine

%Gradiant Morphologique = GradInt + GradExt

Morpho\_GradInt = Morpho - Erod(Morpho);

Morpho\_GradExt = Dilat(Morpho) - Morpho;

Morpho\_GradMorphologique = Morpho\_GradInt + Morpho\_GradExt;

imshow(Morpho\_GradMorphologique)

Exercice 8

function J = Median(I) %EX4

[nb\_lig, nb\_col] = size(I);

voisin=zeros(3,3);

J = zeros(nb\_lig,nb\_col,'uint8');

for i = 2:nb\_lig-1

for j=2:nb\_col-1

voisin = [I(i-1,j-1) I(i-1,j) I(i-1,j+1);

I(i,j-1) I(i,j) I(i,j+1);

I(i+1,j-1) I(i+1,j) I(i+1,j+1)];

Res = sort(double(voisin(:)));

J(i,j) = Res(5);

end

end

J( :,1)=J( :,2);

J( :,nb\_col)=J(:,nb\_col-1);

J(1, :)=J(2, :);

J(nb\_lig, :)=J(nb\_lig-1, :);

end

function J = Dilat(I) %EX5

[nb\_lig, nb\_col] = size(I);

voisin=zeros(3,3);

J = zeros(nb\_lig,nb\_col,'uint8');

for i = 2:nb\_lig-1

for j=2:nb\_col-1

voisin = [I(i-1,j-1) I(i-1,j) I(i-1,j+1);

I(i,j-1) I(i,j) I(i,j+1);

I(i+1,j-1) I(i+1,j) I(i+1,j+1)];

J(i,j) = max(double(voisin(:)));

end

end

J( :,1)=J( :,2);

J( :,nb\_col)=J(:,nb\_col-1);

J(1, :)=J(2, :);

J(nb\_lig, :)=J(nb\_lig-1, :);

end

function J = DilatN(I,n) %EX5

I\_tmp = I;

for i=1:n

I\_tmp = Dilat(I\_tmp);

end

J = I\_tmp;

end

function J = Erod(I) %EX5

[nb\_lig, nb\_col] = size(I);

voisin=zeros(3,3);

J = zeros(nb\_lig,nb\_col,'uint8');

for i = 2:nb\_lig-1

for j=2:nb\_col-1

voisin = [I(i-1,j-1) I(i-1,j) I(i-1,j+1);

I(i,j-1) I(i,j) I(i,j+1);

I(i+1,j-1) I(i+1,j) I(i+1,j+1)];

J(i,j) = min(double(voisin(:)));

end

end

J( :,1)=J( :,2);

J( :,nb\_col)=J(:,nb\_col-1);

J(1, :)=J(2, :);

J(nb\_lig, :)=J(nb\_lig-1, :);

end

function J = ErodN(I,n) %EX5

I\_tmp = I;

for i=1:n

I\_tmp = Erod(I\_tmp);

end

J = I\_tmp;

end

function J = Fermeture(I) %EX6

J = Erod(Dilat(I));

end

function J = FermetureN(I,n) %EX6

I\_tmp = I;

for i=1:n

I\_tmp = Dilat(I\_tmp);

end

for i=1:n

I\_tmp = Erod(I\_tmp);

end

J = I\_tmp;

end

function J = Ouverture(I) %EX6

J = Dilat(Erod(I));

end

function J = OuvertureN(I,n) %EX6

I\_tmp = I;

for i=1:n

I\_tmp = Erod(I\_tmp);

end

for i=1:n

I\_tmp = Dilat(I\_tmp);

end

J = I\_tmp;

end