
Table of Contents

BAIM Mohamed : TD Super Resolution Numerique	1
I.Simulation d'une pile d'images translatées et sous échantillonnées	2
II.Super-résolution de la pile d images, connaissant les décalages entre images	3
Estimation de l'Erreur Quadratique Moyenne	4
III. Super-résolution d une pile d'images bruitées, connaissant les décalages exacts entre les images	5
IV. Super-résolution de la pile d images, connaissant les décalages approximatifs entre les images	7
V. Influence de l'intégration sur le pixel sur la Super-résolution	10
VI.Estimation des décalages entre images sous-résolues	14

BAIM Mohamed : TD Super Resolution Numerique

```
clear;close all;clc
```

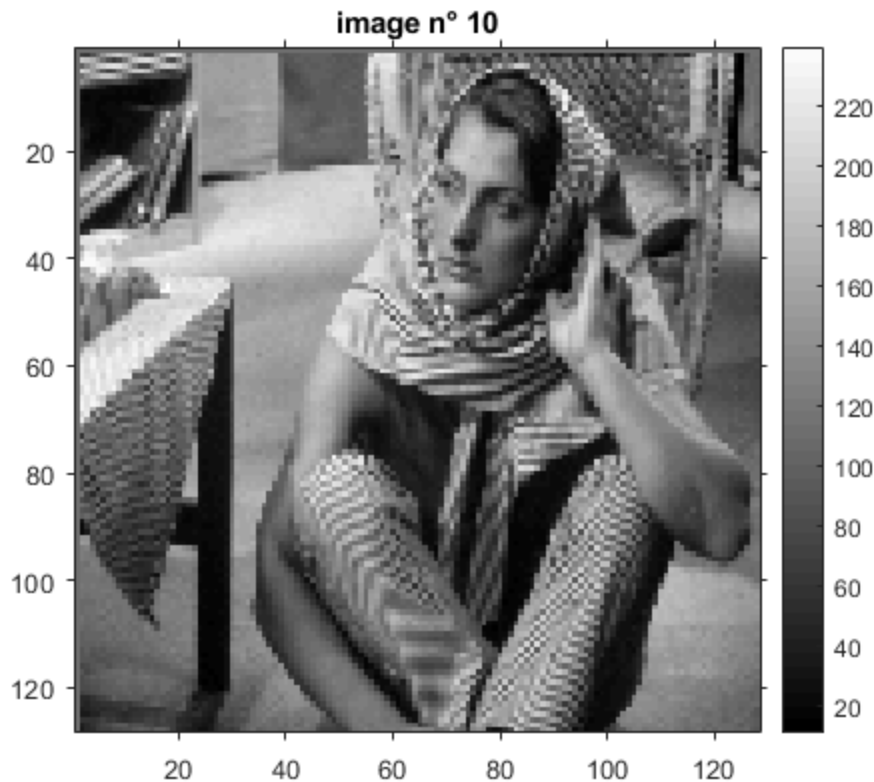
```
I=double(imread('barbara.png'));  
figure('Name','I');imshow(I,  
[],'InitialMagnification','fit');colorbar;axis on;title('I');  
f=4; % doit etre paire % f représente le facteur de résolution  
aff=1; % booleen d'affichage si aff=1 alors affichage des images de la pile
```



I.Simulation d'une pile d'images translatées et sous échantillonnées

```
Nim=10; % Nim représente le nbr d'image de la pile
I1=zeros(size(I,1),size(I,2),Nim);
I2=zeros(round(size(I,1)/f),round(size(I,2)/f),Nim);
txty=f*rand(2,Nim)-(f/2); %% txty représente le décalage ajouter à chaque
    image
txty(:,1)=[0,0];
if aff==1 ,figure('Name','pile d''images'),end
for c=1:Nim
    xform = [ 1  0  0;          0  1  0;   txty(1,c) txty(2,c)  1 ]; %Matrice de
    translation
    tform_translate = maketform('affine',xform);
    I1(:,:,c)= imtransform(I, tform_translate, 'XData',[1 size(I,2)], 'YData',[1
size(I,1)], 'FillValues',mean(I(:)));%??
    I2(:,:,c)=I1(1:f:end,1:f:end,c);%on tranque les pixels de bords
    if aff==1,      imshow(I2(:,:,c),
[, 'InitialMagnification','fit'];colorbar;axis on;title(sprintf('image n°
%d',c));pause(0.1),end
    save('pile.mat','I2','txty');

end
% Les artefacts qui apparaissent sur le pontalon sont l'origine du
% phénomène de moiré
```

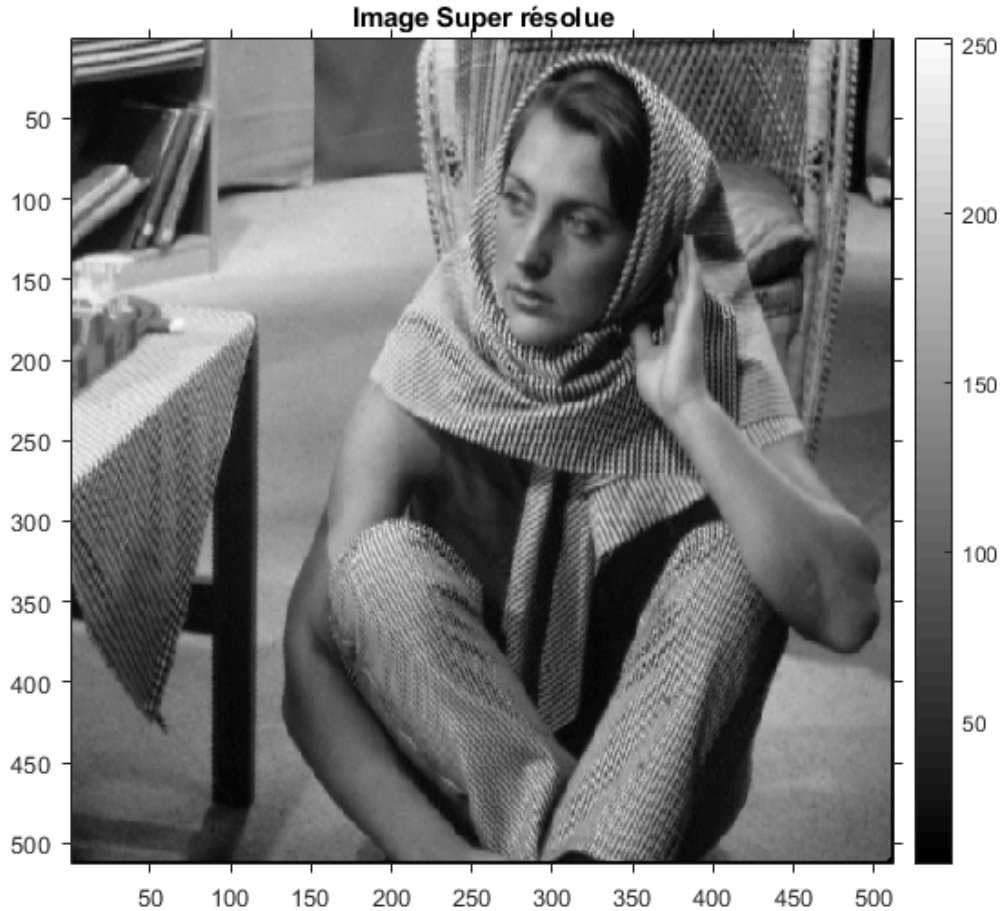


II. Super-résolution de la pile d images, connaissant les décalages entre images

```

NimSR = Nim;
% Creation d'une grille d'interpolation
[X,Y]=meshgrid(1:f:size(I,2),1:f:size(I,1));
[Xi,Yi]=meshgrid(1:size(I,2),1:size(I,1));
% Recalage
Xt=zeros(size(X,1),size(X,2)*NimSR);
Yt=Xt;
datat=Xt;
% les lignes suivantes servent à faire le recalage de la pile d'image
for c=1:NimSR % NimSR est le nombre d'images de la pile que l'on traite
    % pour obtenir l'image super-résolue
    Xt(:,(size(X,2)*(c-1)+1):(size(X,2)*(c)))=X-txty(1,c);
    Yt(:,(size(Y,2)*(c-1)+1):(size(Y,2)*(c)))=Y-txty(2,c);
    datat(:,(size(Y,2)*(c-1)+1):(size(Y,2)*(c)))=I2(:, :, c);
end
% la ligne suivante sert à faire l'interpolation
ISR = griddata(Xt,Yt,datat,Xi,Yi,'cubic');
figure;
imshow(ISR,[]);title("Image Super résolue");colorbar;axis on;

```



Estimation de l'Erreur Quadratique Moyenne

```
Err = I(5:end-4,5:end-4)-ISR(5:end-4,5:end-4);  
MSE = norm(Err.^2)/(504*504);  
disp(MSE)  
% Pour Nim = 10: MSE = 0.81  
% Pour Nim = 20: MSE = 0.42  
% Pour Nim = 40: MSE = 0.18  
% Pour Nim = 50: MSE = 0.15
```

0.4123

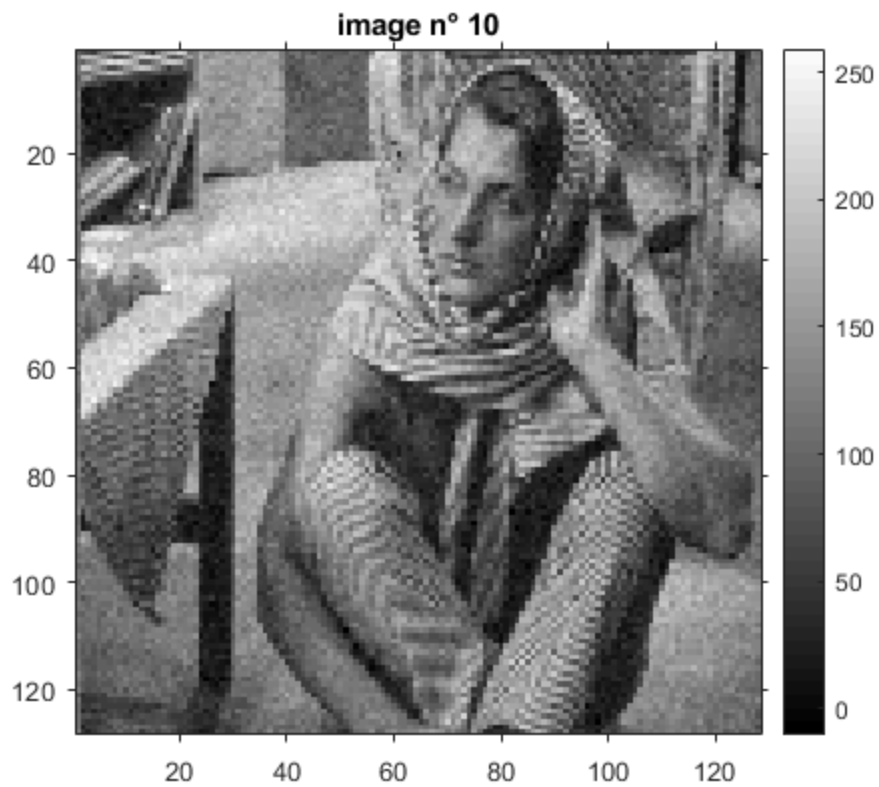
III. Super-résolution d une pile d'images bruitées, connaissant les décalages exacts entre les images

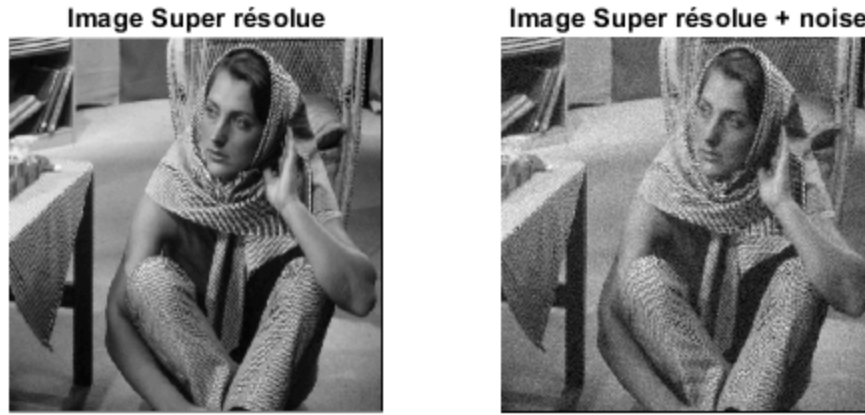
Ss Echantillonnage

```
Nim=10; % Nim représente le nbr d'image de la pile
I1=zeros(size(I,1),size(I,2),Nim);
I2=zeros(round(size(I,1)/f),round(size(I,2)/f),Nim);
txty=f*rand(2,Nim)-(f/2); %% txty représente le décalage ajouter à chaque
    image
txty(:,1)=[0,0];
if aff==1 ,figure('Name','pile d'images'),end
for c=1:Nim
    xform = [ 1  0  0;          0  1  0;   txty(1,c) txty(2,c)  1 ]; %Matrice de
    translation
    tform_translate = maketform('affine',xform);
    I1(:, :,c)= imtransform(I, tform_translate, 'XData',[1 size(I,2)], 'YData',[1
size(I,1)], 'FillValues',mean(I(:)));
    noise = 10*randn(128);
    I2(:, :,c)=I1(1:f:end,1:f:end,c)+noise; %on tranque les pixels de bords +
    noise
    if aff==1,      imshow(I2(:, :,c),
[, 'InitialMagnification','fit'];colorbar;axis on;title(sprintf('image n°
%d',c));pause(0.1),end
    save('pile.mat','I2','txty');
end
% Interpolation
NimSR = Nim;
% Creation d'une grille d'interpolation
[X,Y]=meshgrid(1:f:size(I,2),1:f:size(I,1));
[Xi,Yi]=meshgrid(1:size(I,2),1:size(I,1));
% Recalage
Xt=zeros(size(X,1),size(X,2)*NimSR);
Yt=Xt;
datat=Xt;
% les lignes suivantes servent à faire le recalage de la pile d'image
for c=1:NimSR % NimSR est le nombre d'images de la pile que l'on traite
    % pour obtenir l'image super-résolue
    Xt(:,(size(X,2)*(c-1)+1):(size(X,2)*(c)))=X-txty(1,c);
    Yt(:,(size(Y,2)*(c-1)+1):(size(Y,2)*(c)))=Y-txty(2,c);
    datat(:,(size(Y,2)*(c-1)+1):(size(Y,2)*(c)))=I2(:, :,c);
end
% la ligne suivante sert à faire l'interpolation
ISRnoise = griddata(Xt,Yt,datat,Xi,Yi,'cubic');
% Affichage
figure; colorbar; axis on;
subplot(1,2,1);
imshow(ISR,[,]);title("Image Super résolue");
subplot(1,2,2);
imshow(ISRnoise,[,]);title("Image Super résolue + noise");
```

```
% Err quadratique
Errnoise = I(5:end-4,5:end-4)-ISRnoise(5:end-4,5:end-4);
MSE_noise = norm(Errnoise.^2)/(504*504);
disp(MSE_noise)
% L'erreur quadratique a diminuée
```

0.9288





IV. Super-résolution de la pile d images, connaissant les décalages approximatifs entre les images

SS ech et pile

```

Nim=10; % Nim représente le nbr d'image de la pile
alpha = 10;
I1=zeros(size(I,1),size(I,2),Nim);
I2=zeros(round(size(I,1)/f),round(size(I,2)/f),Nim);
txty=f*rand(2,Nim)-(f/2)+alpha*randn(size(txty)); %% txty représente le
    décalage ajouter à chaque image
txty(:,1)=[0,0];
if aff==1 ,figure('Name','pile d''images'),end
for c=1:Nim
    xform = [ 1  0  0;          0  1  0;   txty(1,c) txty(2,c)  1 ]; %Matrice de
    translation
    tform_translate = maketform('affine',xform);
    I1(:, :,c)= imtransform(I, tform_translate,'XData',[1 size(I,2)],'YData',[1
size(I,1)],'FillValues',mean(I(:)));%??
    I2(:, :,c)=I1(1:f:end,1:f:end,c);%on tranque les pixels de bords
    if aff==1,      imshow(I2(:, :,c),
[, 'InitialMagnification','fit');colorbar;axis on;title(sprintf('image n°
%d',c));pause(0.1),end

```

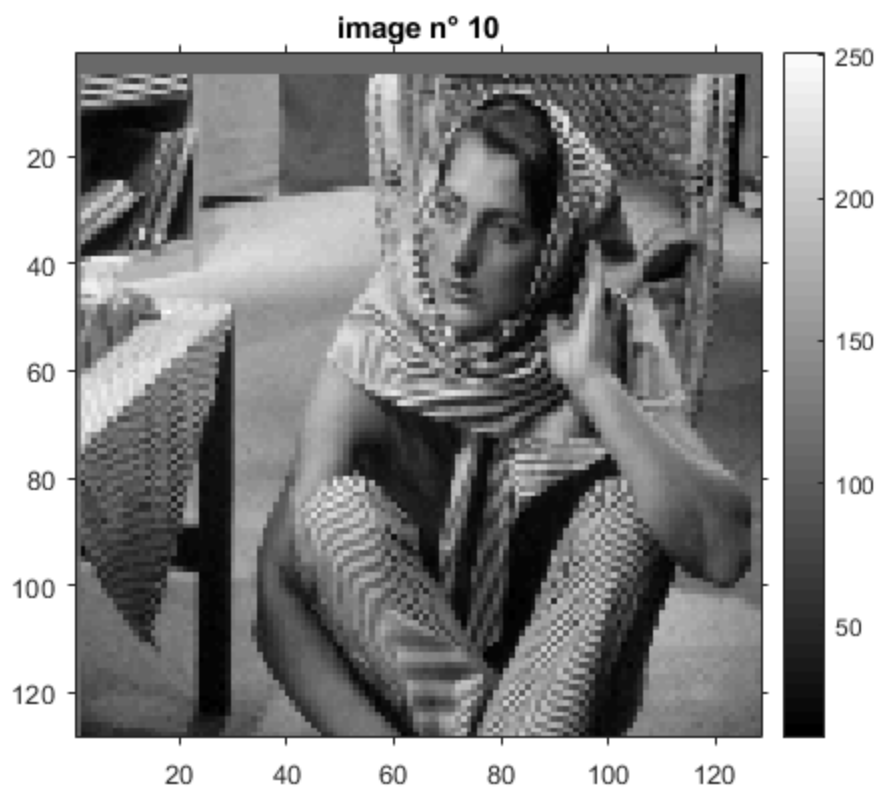
```

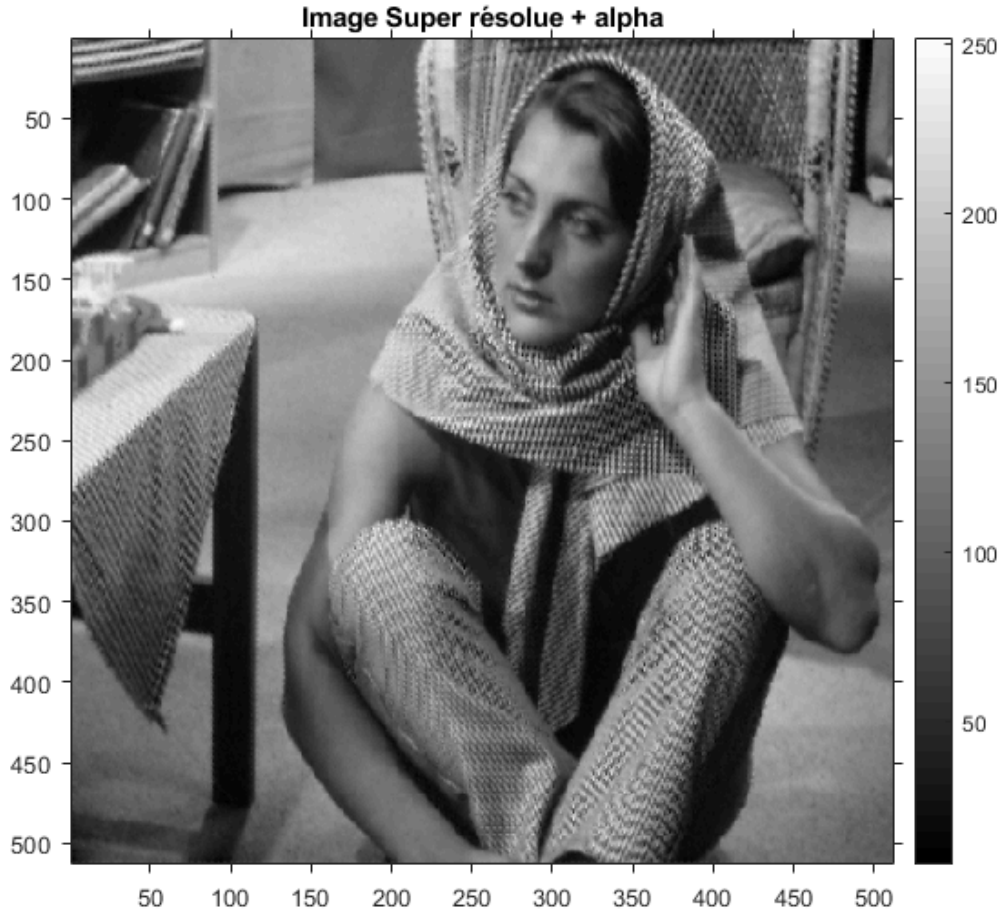
save('pile.mat','I2','txty');

end
% Interpolation et recalage
NimSR = Nim;
% Creation d'une grille d'interpolation
[X,Y]=meshgrid(1:f:size(I,2),1:f:size(I,1));
[Xi,Yi]=meshgrid(1:size(I,2),1:size(I,1));
% Recalage
Xt=zeros(size(X,1),size(X,2)*NimSR);
Yt=Xt;
datat=Xt;
% les lignes suivantes servent à faire le recalage de la pile d'image
for c=1:NimSR % NimSR est le nombre d'images de la pile que l'on traite
    % pour obtenir l'image super-résolue
    Xt(:,(size(X,2)*(c-1)+1):(size(X,2)*(c)))=X-txty(1,c);
    Yt(:,(size(Y,2)*(c-1)+1):(size(Y,2)*(c)))=Y-txty(2,c);
    datat(:,(size(Y,2)*(c-1)+1):(size(Y,2)*(c)))=I2(:, :, c);
end
% la ligne suivante sert à faire l'interpolation
ISR_alpha = griddata(Xt,Yt,datat,Xi,Yi,'cubic');
figure;
imshow(ISR_alpha,[]);title("Image Super résolue + alpha");colorbar;axis on;
% Err quadratique
Erralpha = I(5:end-4,5:end-4)-ISR_alpha(5:end-4,5:end-4);
MSE_alpha = norm(Erralpha.^2)/(504*504);
disp(MSE_alpha)

0.7253

```



V. Influence de l'intégration sur le pixel sur la Super-résolution

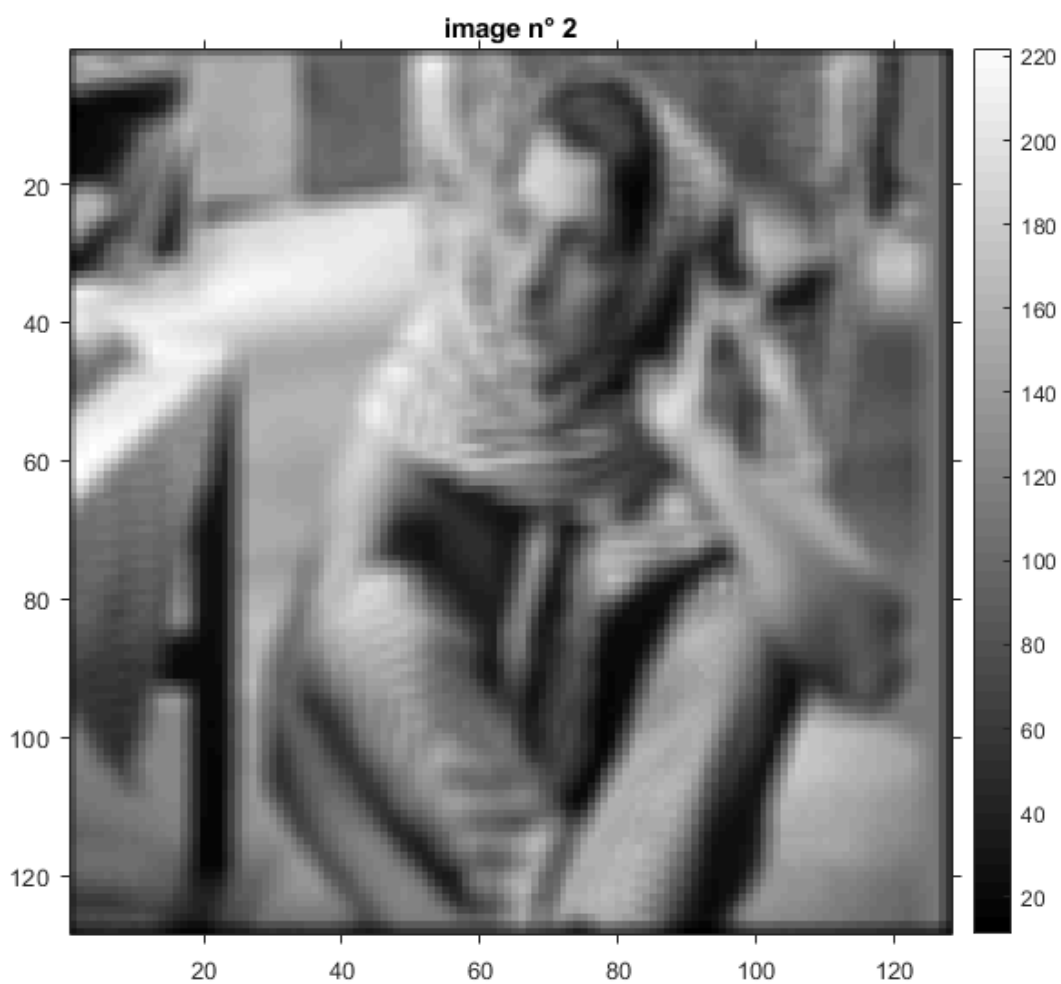
SS ech+ integration sur pixel

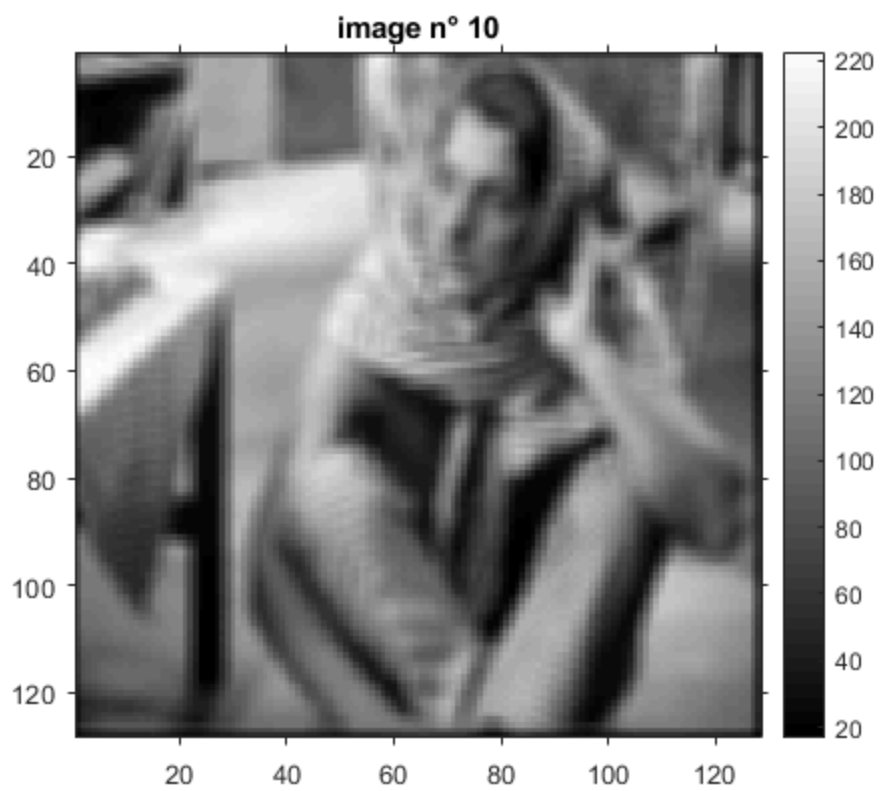
```
Nim=10; % Nim représente le nbr d'image de la pile
alpha = 10;
I1=zeros(size(I,1),size(I,2),Nim);
I2=zeros(round(size(I,1)/f),round(size(I,2)/f),Nim);
txty=f*rand(2,Nim)-(f/2)+alpha*randn(size(txty)); %% txty représente le
    décalage ajouter à chaque image
txty(:,1)=[0,0];
h = fspecial("average",[4 4]);
if aff==1 ,figure('Name','pile d'images'),end
for c=1:Nim
    xform = [ 1 0 0;          0 1 0;    txty(1,c) txty(2,c) 1 ]; %Matrice de
    translation
```

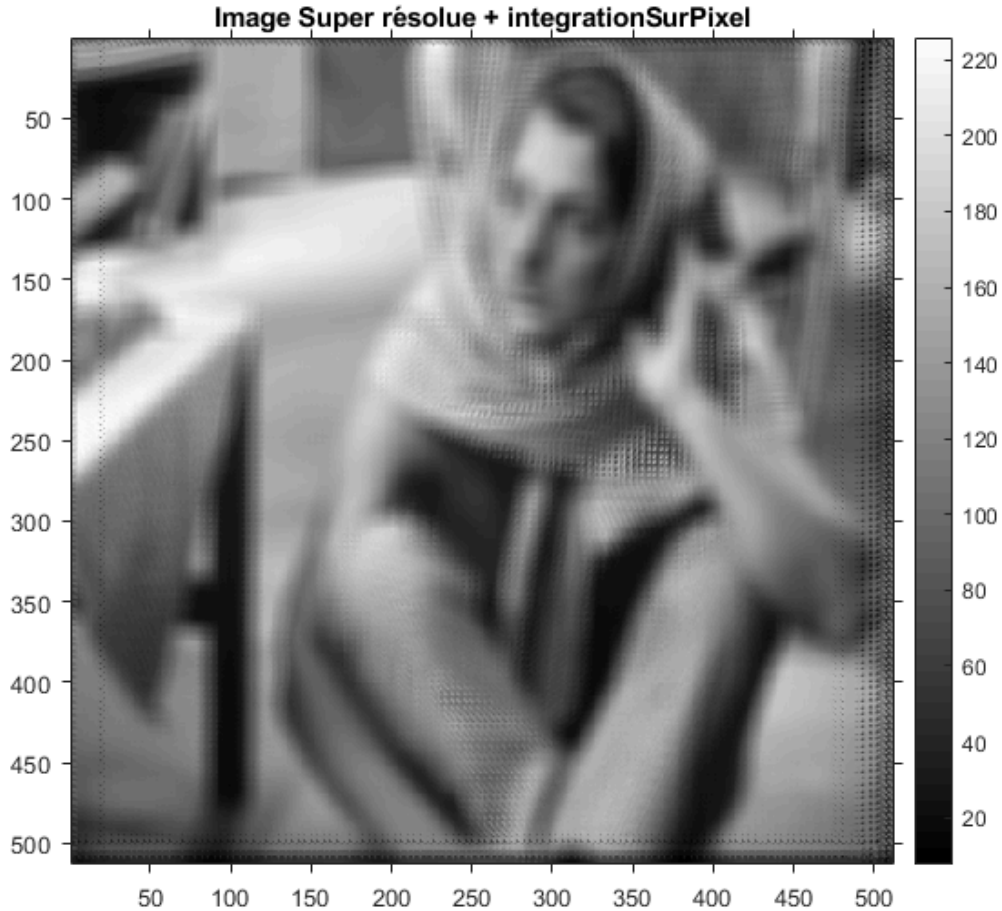
```

    tform_translate = maketform('affine',xform);
    I1(:,:,c)= imtransform(I, tform_translate,'XData',[1 size(I,2)],'YData',[1
size(I,1)], 'FillValues',mean(I(:)));%??
    I2(:,:,c)=I1(1:f:end,1:f:end,c);%on tranque les pixels de bords
    I2(:,:,c)= imfilter(I2(:,:,c),h);
    if aff==1,        imshow(I2(:,:,c),
[], 'InitialMagnification','fit');colorbar;axis on;title(sprintf('image n°
%d',c));pause(0.1),end
    save('pile.mat','I2','txty');
end
% Super res
% Interpolation et recalage
NimSR = Nim;
% Creation d'une grille d'interpolation
[X,Y]=meshgrid(1:f:size(I,2),1:f:size(I,1));
[Xi,Yi]=meshgrid(1:size(I,2),1:size(I,1));
% Recalage
Xt=zeros(size(X,1),size(X,2)*NimSR);
Yt=Xt;
datat=Xt;
% les lignes suivantes servent à faire le recalage de la pile d'image
for c=1:NimSR % NimSR est le nombre d'images de la pile que l'on traite
    % pour obtenir l'image super-résolue
    Xt(:,(size(X,2)*(c-1)+1):(size(X,2)*(c)))=X-txty(1,c);
    Yt(:,(size(Y,2)*(c-1)+1):(size(Y,2)*(c)))=Y-txty(2,c);
    datat(:,(size(Y,2)*(c-1)+1):(size(Y,2)*(c)))=I2(:,:,c);
end
% la ligne suivante sert à faire l'interpolation
ISR_integ = griddata(Xt,Yt,datat,Xi,Yi,'cubic');
figure;
imshow(ISR_integ,[]);title("Image Super résolue +
integrationSurPixel");colorbar;axis on;
% L'integration sur le pixel réduit la qualité de l'image

```







VI. Estimation des décalages entre images sous-résolues

```
dbtype("MaxSubPixel2.m");
```

```
1    function [xs,ys,val]=MaxSubPixel2(imS,v);
2    % MaxSubPixel2 .m : Calcul des coordonnées "subpixel" du maximum d'une
   image et de l'amplitude de ce maximum "subpixel" par ajustement polynomial
   1D.
3    % Usage : [xs,ys,val]=MaxSubPixel2(imS,v);
4    % avec pour paramètres d'entrée :
5    %         im : image d'entrée
6    %         v : voisinage utilisé pour ajuster le polynome
7    % avec pour paramètres de sortie :
8    %         xs,ys : coordonnées "subpixel" du maximum de l'image
9    %         val : valeur de ce maximum
```

```
10    % Créé par Corinne Fournier, 11/2011
11
12    [vx,x]=max(max(imS)); % Max au pixel pres
13    [vy,y]=max(max(imS')) ;
14    vali(1)=vx;
15
16    [P,S]=polyfit(x-v:x+v,imS(y,x-v:x+v),2);
17    xs=-P(2)/2/P(1);% Max Subpixel
18    vali(2)=P(1)*xs*xs+P(2)*xs+P(3);
19
20    [P,S]=polyfit(y-v:y+v,imS(y-v:y+v,x)',2);
21    ys=-P(2)/2/P(1);
22
23    vali(3)=P(1)*ys*ys+P(2)*ys+P(3);
24    val=max(vali);
```

Published with MATLAB® R2023a