

---

# Exemple de correction des parties I et II du TD sur la Super Resolution Numerique

## Table of Contents

I. Images tests : simulation d'une pile d'images sous-échantillonnées et traduites .....	1
I.1. Générez cette pile d'images en vous aidant du script : TDSRnum_main.m .....	1
I.2. D'où proviennent les artefacts observés sur le pantalon et le foulard de la personne ? .....	3
II. Super-résolution de la pile d'images, connaissant les décalages entre images .....	3
II.1. Etapes de (1) recalage et (2) interpolation, .....	3
II.2. Commentaire sur l'image obtenue .....	5
II.3. Etude quantitative de l'influence du nombre d'images utilisées sur la qualité de l'image super-résolue .....	6

## I. Images tests : simulation d'une pile d'images sous-échantillonnées et traduites

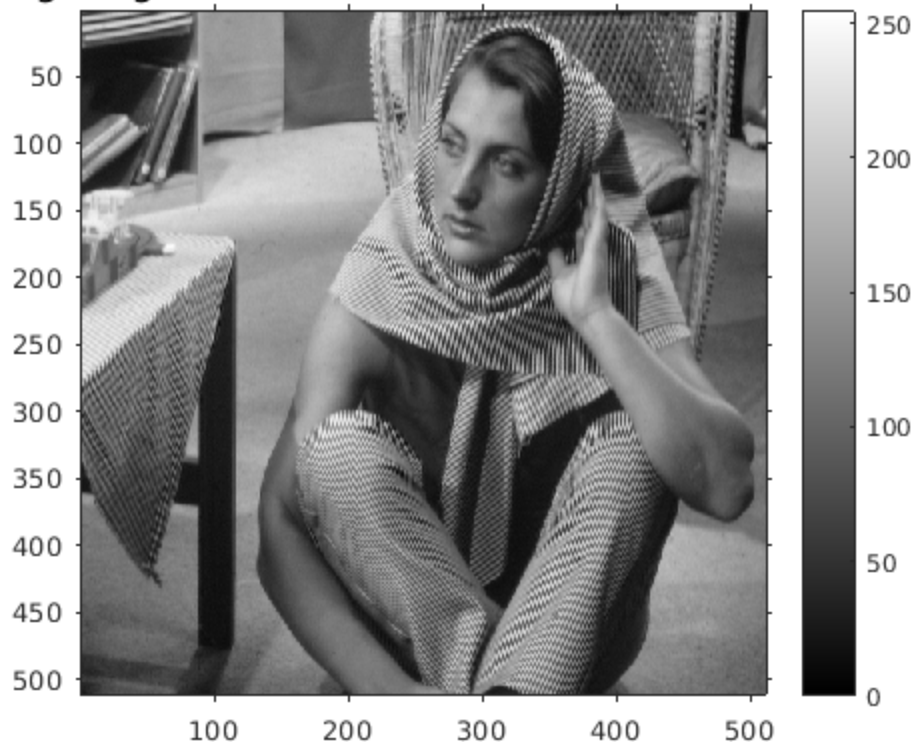
### I.1. Générez cette pile d'images en vous aidant du script : TDSRnum\_main.m

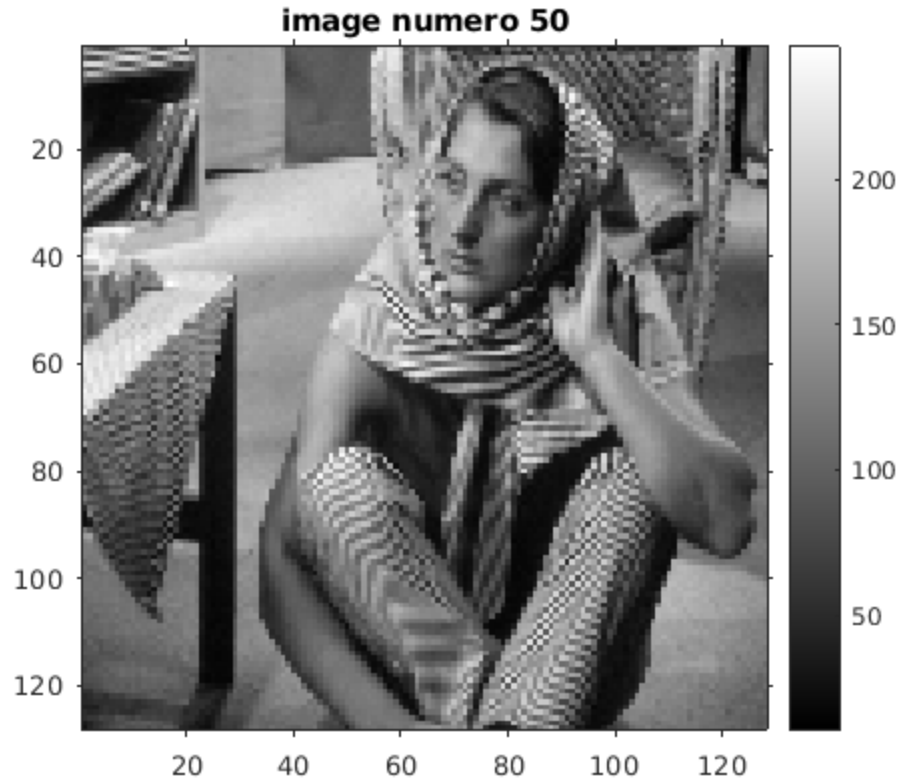
```
clear;close all;clc;format compact
I=double(imread('barbara.png'));
figure('Name','I');imshow(I,
[],'InitialMagnification','fit');colorbar;axis on;title('I image
originale dans ce TD a un role de "verite terrain"');
f=4; % doit etre en puissance de 2 % f représente le facteur de sous
échantillonnage
aff=1; % boolean d'affichage si aff=1 alors affichage des iamges
intermédiiaires
calc0=0; % calcul de la pile d'images
calc1=1;% etude quantitative de la qualité de la reconstruction en
fonction du nb d'images de la pile
Nim=50; % nombres d'images de la pile
NimSR=16; % nombres d'images utilisées pour le calcul de l'image
super-résolue
I1=zeros(size(I,1),size(I,2),Nim);
I2=zeros([round(size(I,1)/f),round(size(I,2)/f),Nim]);
if calc0==1
    txtxy=f*rand(2,Nim)-(f/2); %% chaque image de la pile sera
    traduites d'un vecteur txtxy par rapport # l'image 1
    txtxy(:,1)=[0,0]; %% la premiere image sera prise comme référence
    (translation=0)
```

```
if aff==1 ,figure('Name','pile d''images'),end

for c=1:Nim;
    xform = [ 1 0 0;          0 1 0;  txtx(1,c) txtx(2,c)  1 ];
    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);
    if aff==1,    imshow(I2(:,:,c),
[],'InitialMagnification','fit');colorbar;axis on;title(sprintf('image
num %d',c));pause(0.1),end
    %    imwritesc(I2(:,:,c),sprintf('barbara_%03d.png',c))
end
save('pile.mat','I2','txtx');
else
    load('pile.mat','I2','txtx');
end
figure;imshow(I2(:,:,end),
[],'InitialMagnification','fit');colorbar;axis on;title(sprintf('image
numero %d',Nim));pause(0.1),
```

**I image originale dans ce TD a un role de "verite terrain"**





## I.2. D'où proviennent les artéfacts observés sur le pantalon et le foulard de la personne ?

Les artefacts sont des figures de Moirés. Ils sont dus au fait qu'on sous échantillonne des fréquences qui sont supérieures à la fréquence de Nyquist-Shannon (après sous échantillonnage, c'est à dire  $f_{NS}=1/(2 \cdot (f \cdot p))$ ,  $p$  étant le pas d'échantillonnage de l'image originale/"Verite Terrain").

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

### II.1. Etapes de (1) recalage et (2) interpolation,

```
[X,Y]=meshgrid(1:f:size(I,2),1:f:size(I,1)); % grille de coordonnées
des pixels de l'image initiale
[Xi,Yi]=meshgrid(1:size(I,2),1:size(I,1));% grille de coordonnées des
pixels de la premiere image de la pile
% generation des grilles recalées et des données
Xt=zeros(size(X,1),size(Y,2)*Nim) ;Yt=Xt;datat=Xt;
for c=1:NimSR
    Xt(:,size(X,2)*(c-1)+1:size(X,2)*c)=X-txty(1,c);
    Yt(:,size(X,2)*(c-1)+1:size(X,2)*c)=Y-txty(2,c);
```

```
    datat(:,size(X,2)*(c-1)+1:size(X,2)*c)=I2(:, :, c);
end
tic; ISR = griddata(Xt,Yt,datat,Xi,Yi,'cubic');tgriddata=toc % (2)
    interpolation

dbtype super_resolution.m
figure('Name','I super-résolue');
imshow(ISR,[],'InitialMagnification','fit');colorbar;axis on;title('I
SR');

Warning: Duplicate x-y data points detected: using average values for
duplicate points.
tgriddata =
    1.7093

1    function [ISR,tgriddata]=super_resolution(I2,f,txtty)
2    %% Fonction permettant d'augmenter le nombre de pixels d'une
    image par un facteur f^2
3    % input :
4    % I2 : pile d'images =donnees (LxCxNimSR)
5    % f : facteur de SR
6    % txtty : tableau des valeurs des déplacements transversaux dans
    la pile
7    % output :
8    % ISR : image super-resolue (L.fx C.f)
9    % tgriddata : duree du traitement
10
11    [X,Y]=meshgrid(1:f:size(I2,2)*f,1:f:size(I2,1)*f); % grille de
    coordonnées des pixels de l'image initiale
12    [Xi,Yi]=meshgrid(1:size(I2,2)*f,1:size(I2,1)*f);% grille de
    coordonnées des pixels de la premiere image de la pile
13    NimSR=size(I2,3);
14    % generation des grilles recalées et des données
15    Xt=zeros(size(X,1),size(Y,2)*NimSR) ;Yt=Xt;datat=Xt;
16    for c=1:NimSR
17        Xt(:,size(X,2)*(c-1)+1:size(X,2)*c)=X-txtty(1,c);
18        Yt(:,size(X,2)*(c-1)+1:size(X,2)*c)=Y-txtty(2,c);
19        datat(:,size(X,2)*(c-1)+1:size(X,2)*c)=I2(:, :, c);
20    end
21    tic; ISR = griddata(Xt,Yt,datat,Xi,Yi,'cubic');tgriddata=toc; %
    (2) interpolation
```



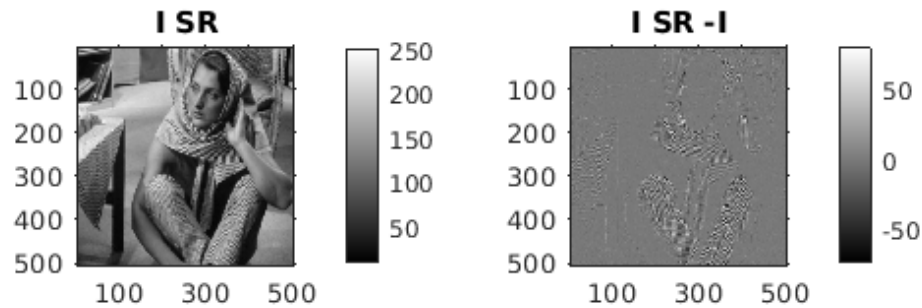
## II.2. Commentaire sur l'image obtenue

Les Moirés présents dans la pile d'image ne sont plus présents sur l'image super-résolue.

L'image SR présente des pixels qui n'ont pas de valeurs numériques (NaN). Ceci provient de l'étape d'interpolation qui ne peut pas être effectuée sur les bords de l'image.

Pour s'affranchir de ces problèmes situés au bord de l'image, nous tronquons l'image SR.

```
ISR_tr=ISR(f+1:end-f,f+1:end-f);  
I_tr=I(f+1:end-f,f+1:end-f,:);  
figure('Name','I super-résolue');  
subplot(1,2,1);imshow(ISR_tr,  
[],'InitialMagnification','fit');colorbar;axis on;title('I SR');  
subplot(1,2,2);imshow(ISR_tr-I_tr,  
[],'InitialMagnification','fit');colorbar;axis on;title('I SR -I');
```



## II.3. Etude quantitative de l'influence du nombre d'images utilisées sur la qualité de l'image super-résolue

comparaison de l'image SR avec l'image initiale, évolution en fonction de NimSR

```
NimSR=2:2:Nim;
if calc1==1
    If=zeros(size(I,1),size(I,2),size(NimSR,2));
    for c0=1:size(NimSR,2)
        %           c0

        [If(:,:,c0),tgriddata(c0)]=super_resolution(I2(:,:,1:NimSR(c0)),f,txty);
    end
    save('If.mat','If','NimSR')
else
    load('If.mat','If','NimSR')
end
If_tr=If(f+1:end-f,f+1:end-f,:); % Image SR tronquee pour s'affranchir
des bords
figure;
subplot(1,2,1)
```

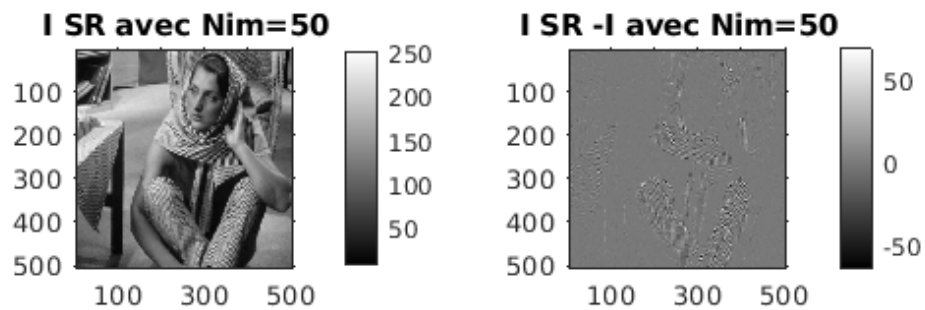
---

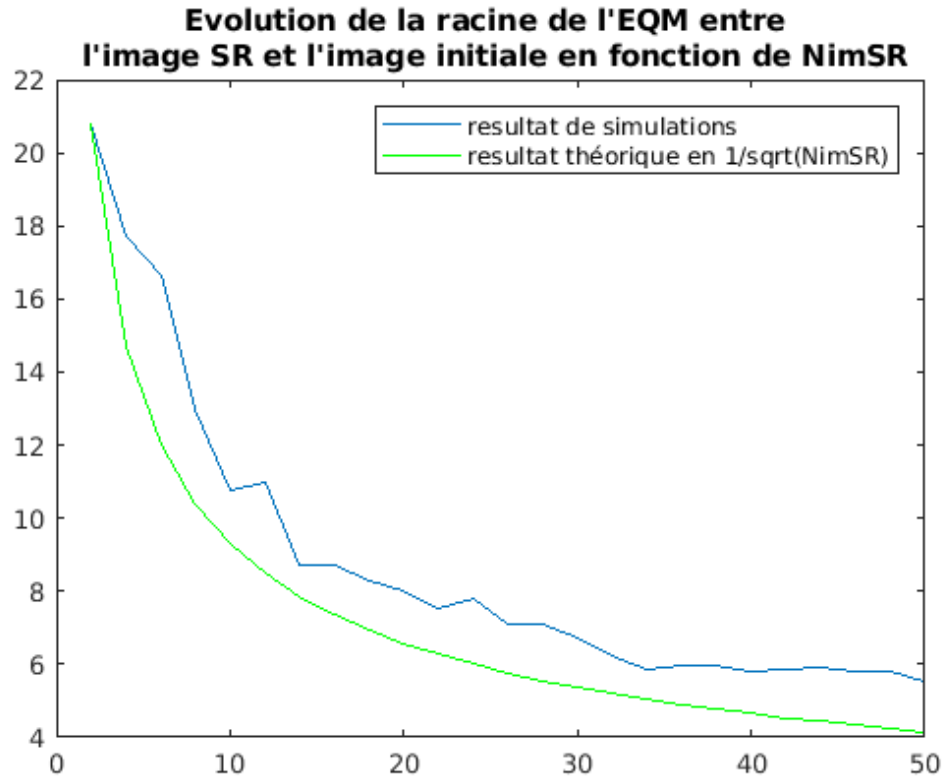
```

imshow(If_tr(:,:,end),
[], 'InitialMagnification','fit');colorbar;axis on;title(sprintf('I SR
avec Nim=%d',NimSR(end)));pause(0.5)
for c0=1:size(NimSR,2)
    teqm(c0)=immse(If_tr(:,:,c0),I_tr);
end
subplot(1,2,2);
imshow(If_tr(:,:,end)-I_tr,
[], 'InitialMagnification','fit');colorbar;axis on;title(sprintf('I SR
-I avec Nim=%d',NimSR(c0)));pause(0.5)
r_teqm=sqrt(teqm);

figure; plot(NimSR,r_teqm)
hold on; plot(NimSR,r_teqm(1)*sqrt(NimSR(1))./sqrt(NimSR),'g')
title(sprintf('Evolution de la racine de l''EQM entre \n l''image SR
et l''image initiale en fonction de NimSR'))
legend('resultat de simulations','resultat théorique en 1/
sqrt(NimSR)')

```





On observe une diminution de l'écart quadratique moyen entre l'image SR et l'image initiale avec le nombre d'images de la pile. Cette évolution est en  $\sqrt{\text{NimSR}}$

Notons qu'en n'utilisant que 10 images dans la pile, l'image SR est déjà visuellement correcte.

*Published with MATLAB® R2017b*