

TD5 Image & Signal

Super-résolution numérique

Bloc Image 2

C. Fournier

Les fonctions à utiliser sont en partie données dans l'énoncé. Vous devrez faire appel à l'aide de Matlab pour leur syntaxe.

On réalisera un programme (.m). Des fonctions intermédiaires pourront être programmées. Il est indispensable d'annoter votre programme pour le rendre plus lisible pour une utilisation ultérieure et pour l'enseignant. Un compte rendu de TD devra être déposé sur le portail en fin de séance (programmes, fonctions, images résultats, **commentaires**).

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

Dans cette première partie du TD vous générerez des images sous échantillonnées et légèrement translatées les unes par rapport aux autres.

1. Générez cette pile d'images en vous aidant du script « TDSRnum_main.m » que vous commenterez (compléter les commentaires comportant des « ? »).



(a)



(b)

Figure 1 : à gauche image originale, à droite image sous résolue

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

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

Le but de cette partie est de restaurer l'image initiale (Fig.1.a) à partir de la pile d'images en connaissant les décalages inter-images. Cette opération de super-résolution consiste (dans ce cas simple) à (1) recaler les images de la pile, (2) interpoler les niveaux de gris aux nœuds d'une grille super-résolue (cf cours).

1. Pour réaliser ces deux étapes, utilisez le code suivant que vous annoterez.

```
[X,Y]=meshgrid(1:f:size(I,2),1:f:size(I,1));
[Xi,Yi]=meshgrid(1:size(I,2),1:size(I,1));
Xt=zeros(size(X,1),size(X,2)*NimSR);Yt=Xt;datat=Xt;
% les lignes suivantes servent à ???
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 ???
ISR = griddata(Xt,Yt,datat,Xi,Yi, 'cubic');
```

2. Commentez le résultat obtenu.
3. Etudiez quantitativement l'influence du nombre d'images utilisées sur la qualité de l'image super-résolue. Vous pourrez pour cela calculer l'Ecart Quadratique Moyen (EQM) entre chaque image super-résolue et l'image initiale ('barbara.png'). Visualisez l'évolution de la racine carrée de l'EQM pour un nombre d'images allant de 10 à 50.

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

1. Ajoutez un bruit additif blanc gaussien (fonction "randn", voir l'aide) sur les images sous-résolues.
2. Calculez l'image super-résolue correspondante.
3. Commentez le résultat obtenu.

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

1. Pour étudier l'influence d'une erreur de l'estimation du décalage entre images sur l'image super-résolue, introduisez un bruit dans le tableau de valeurs des décalages (txty= txty+alpha*randn(size(txty)) vous pourrez tester différentes valeurs de α), puis reconstruire.
2. Comparez les EQM entre les images restaurées et l'image initiale. Commentez.

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

1. Modifiez le code de génération des images sous résolues simulées dans la partie I, pour prendre en compte une intégration sur le pixel (vous considérerez un facteur de remplissage du pixel de 100%, c'est-à-dire une zone photosensible de taille égale à la taille du pixel). Pour ceci, vous utiliserez un filtre moyennneur avant de sous échantillonner.
2. Appliquer ensuite l'algorithme de super-résolution. Quel est l'impact de l'intégration sur le pixel sur l'image super-résolue ?

VI. Estimation des décalages entre images sous-résolues

L'estimation des décalages entre images peut être réalisé par corrélation (cf TD précédents) ou par ajustement d'une rampe sur la différence de phase des spectres des deux images.

1. Implémentez une fonction d'estimation « subpixel » du décalage (horizontal et vertical) entre deux images par corrélation.
2. Implémentez une fonction d'estimation subpixel du décalage (horizontal et vertical) entre deux images par ajustement d'une rampe sur la différence de phase entre les spectres des deux images.
 - a. Les hautes fréquences des spectres des images sont très bruitées expliquez pourquoi.
 - b. Implémentez la fonction d'estimation subpixel du décalage par ajustement d'une rampe sur la différence de phase entre les spectres des deux images en ne considérant que les basses fréquences des spectres.
3. Testez les deux algorithmes précédents sur des images translatées et sous-échantillonnées d'un facteur 2 puis d'un facteur 4.
4. Commentez.