TP23 – Imagerie Computationnelle

En plus des exercices ci-dessous, il vous est fortement conseillé de consulter les Numerical Tours with Matlab (https://www.numerical-tours.com/matlab/) ou with Python (https://www.numerical-tours.com/python/), et en particulier l'implémentation de toutes les méthodes de restauration d'images vues en cours, comme par exemple :

- Simple Denoising Methods / Linear Image Denoising
- Simple Denoising Methods / Denoising by Sobolev and Total Variation Regularization
- Wavelet Denoising / Image Denoising with Wavelets
- Advanced Denoising Methods / Non-Local Means
- Advanced Denoising Methods / Bilateral Filtering
- Sparsity and Redundant Representations / Compressed Sensing of Images
- Sparsity and Redundant Representations / Dictionary Learning for Denoising
- Inverse Problems / Image Deconvolution using Variational Method
- Inverse Problems / Image Deconvolution using Sparse Regularization
- Inverse Problems / Reconstruction from Partial Tomography Measurements
- Inverse Problems / Tomography Inversion using Tikhonov and Sparse Regularization

Déconvolution : calculs de base

1. Vérification des propriétés du produit de convolution vues en cours à partir des images :

$$h = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$
$$x = \mathrm{round}(5 \times \mathrm{rand}(3,3))$$

   a. Calcul du produit de convolution complet dans le domaine spatial.
   b. Calculer le produit de convolution complet dans le domaine fréquentiel.
   c. Calcul du produit de convolution circulaire en utilisant la fonction *imfilter*. Le résultat est de même taille que x.
   d. Calcul du produit de convolution circulaire en passant par le domaine de Fourier. Le résultat est de même taille que x.
   e. Obtention du résultat en c. en définissant une matrice BCCB H à partir de h et en multipliant cette matrice par la version vectorisée de l'image x.

Problèmes inverses à la déconvolution et ses solutions

Nous examinons le modèle de formation d'image suivant :

$$i(x,y) = h(x,y) * i_{ideal}(x,y) + b(x,y) \qquad eq.1$$

Avec * représente le produit de convolution circulaire et $b(x,y)$ représente un bruit gaussien. Généralement, $h(x,y)$ est un noyau qui induit un flou dans l'image. Dans ce qui suit, nous supposerons que $h(x,y)$ est connu. L'objectif est donc de retrouver $i_{ideal}(x,y)$ à partir de $i(x,y)$.

A. Modèle directe : calculer l'image floue et bruitée à partir d'une image parfaite

1. Chargez une image de votre choix, ou l'image *coat_of_arms.png* (à télécharger ici : http://www.ogemarques.com/wp-content/uploads/2014/11/Tutorial_Images.zip)
2. Créez une image constante par morceaux
3. Créez une image sparse.

Visualisez-les. Pour chaque cas, l'image constituera par la suite $i_{ideal}(x,y)$.

B. Pour chaque image, vous devez :

1. Générer et visualiser :
    a. Générer l'image $i(x,y)$, en considérant $h(x,y)$ un filtre Gaussien de taille 15x15, l'image sera bruitée avec un bruit additif Gaussien, de puissance telle que le RSB soit de 40 dB.
    b. Visualiser la transformée de Fourier de $i_{ideal}(x,y)$, de $i(x,y)$ et de $h(x,y)$. Commenter les trois résultats.
2. Estimer l'image par une régularisation Tikhonov (norme L2).
3. Faire varier le paramètre de régularisation. Commenter les résultats.
4. Comparer les résultats avec l'estimation des paramètres de régularisation par les méthodes vues en cours (GCV et ML). Commenter les résultats.
5. Faire varier des paramètres du filtre h(x,y) et le RSB entré et puis commenter les résultats.
6. Idem que la question 2.) avec une norme L1 et TV (Total Variation) par ADMM.