

TP2 – EIEAR3B1 : Vision et traitement d'images 2D

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 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

Débruitage avancé

1. Soit une image en niveau de gris de votre choix, appelée i (par exemple Lena 512x512 pixels). Bruiter cette image par un bruit Gaussien additif, dont la puissance est fixée afin d'obtenir un rapport signal sur bruit de 20 dB. L'image résultante sera appelée ib .
2. L'objectif des questions suivantes est de débruiter l'image ib par différentes méthodes vues en cours. Pour chacune des méthodes mises en œuvre, en plus du critère visuel, deux critères quantitatifs seront évalués : l'erreur quadratique moyenne et le PSNR (peak signal to noise ratio) entre l'image estimée et l'image i . Pour chaque méthode, essayer de fixer les hyperparamètres respectifs à leur meilleure valeur.
 - a. Filtre de Wiener, appliqué par blocs en utilisant la fonction Matlab *wiener2*.
 - b. Seuillage dur et doux dans le domaine des ondelettes. Vous pouvez utiliser une décomposition à l'ordre 2 avec des ondelettes de Haar, ou, en absence de la wavelet toolbox, la décomposition dyadique à télécharger ici : <https://cloud.irit.fr/index.php/s/7W2kXMxo8LpVmOI>. Ensuite, vous devez faire une comparaison avec les fonctions de Matlab associées.
 - c. Filtrage moyen non local.

Déconvolution

3. Vérifier les propriétés du produit de convolution vues en cours, à partir d'une image $h = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$ et une image $x = \text{round}(5 \cdot \text{rand}(3,3))$.
 - a. Calculer le produit de convolution complet dans le domaine spatial.
 - b. Calculer le produit de convolution complet dans le domaine fréquentiel.
 - c. Calculer le produit de convolution circulaire en utilisant la fonction *imfilter*. Le résultat est de même taille que x .
 - d. Calculer le produit de convolution circulaire en passant par le domaine de Fourier. Le résultat est de même taille que x .
 - e. Retrouver le 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 .

Nous considérons le modèle de formation d'image suivant :

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

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

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

4. Charger

- 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) et visualiser la. Cette image constituera par la suite $i_{ideal}(x, y)$.
 - ou créer une image constante par morceaux
 - ou créer une image sparse
- a. Générer l'image $i(x, y)$, en considérant $h(x, y)$ un filtre moyennneur de taille 20x20, 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.

Modèle inverse : estimer l'image parfaite à partir de l'image floue et bruitée

- c. Une méthode qui permet de récupérer une estimation de l'image $i_{ideal}(x, y)$ est la déconvolution de Wiener. Avec la déconvolution de Wiener, la transformée de Fourier de $i_{ideal}(x, y)$ est obtenue par :

$$\left[\frac{1}{H(u, v) |H(u, v)|^2 + K} \right] I(u, v),$$
 avec K une constante qui approxime le rapport bruit sur signal. La constante K sera choisie comme étant 10^{-3} . La faire varier et conclure.

- d. Sous Matlab, le filtre de Wiener est implémenté par la fonction *deconvwnr*. Appliquer le filtre de Wiener à l'image générée en 4.a. Pour le cas sans bruit, des petites valeurs de K (10^{-5}) donnent les meilleurs résultats.
- e. Mettre en œuvre une déconvolution par MAP avec des régularisations sparse, TV et de Tikhonov.
- f. Comparer les résultats en utilisant les images précédentes.