Traitement des images numériques

TP 3 : Filtrage et débruitage

Université Paris 13, Institut Galilée
Master Ingénierie et Innovations en Images et Réseaux - 1ère année

2017-2018

Consignes

- Récupérer le fichier `TP3.zip` sur le site

  `http://www.laurentoudre.fr/tin.html`

- Ouvrir MATLAB et créer un répertoire de travail. Dézipper le fichier `TP3.zip` dans ce répertoire.
- A la fin de la séance, récupérer les scripts que vous avez écrits et les envoyer par e-mail au chargé de TP ainsi qu'à vous même afin de les conserver pour la prochaine séance.

Rendu

- Trois fichiers : `TP3Partie1.m`, `TP3Partie2.m` et `TP3Partie3.m` Chaque fichier doit contenir votre nom, votre prénom et la date.
- Compte-rendu succinct à rendre à la fin de la séance, contenant les observations, commentaires et réponses aux questions. Le compte rendu doit contenir votre nom et votre prénom.

Plan de l'étude

1 Dégradations dans une image

1.1 Rappels de cours

Lors de l'acquisition, de la transmission ou de la compression d'une image, il peut apparaître de nombreuses dégradations. Un des domaines principaux en traitement d'image consiste à traiter et corriger ces dégradations pour obtenir une image de meilleure qualité. On s'interesse ici à deux types de dégradations fréquemment rencontrées dans les images :

- Le **bruit additif**, qui affecte tous les pixels de l'image. Dans ce TP, nous considérerons un **bruit blanc additif Gaussien**, de moyenne nulle et de variance σ^2. Il s'agit d'un modèle fréquemment utilisé en première approximation pour modéliser le bruit d'acquisition et de lecture (si l'on ne dispose pas d'un modèle plus raffiné). Le bruit Gaussien affecte à la fois les basses et les hautes fréquences. Il est caractérisé par sa variance σ^2 : plus σ^2 est élevé, plus l'image est dégradée.
- Le **bruit impulsionnel**, n'affecte que certains pixels de l'image. Dans ce TP, nous considérerons un bruit **sel et poivre**, qui est une dégradation de l'image sous la forme de pixels noirs et blancs répartis au hasard. Ce bruit est dû soit à des erreurs de transmission de données, soit à la défaillance d'éléments du capteur CCD, soit à la présence de particules fines sur le capteur d'images. On le caractérise par le pourcentage p de pixels modifiés : plus p est élevé, plus l'image est dégradée.

Pour ajouter du bruit à une image sous MATLAB, on utilise la commande `imnoise`

```
% X : image renormalisee (valeurs entre 0 et 1)
Y = imnoise(X,'gaussian',m,v) % Applique un bruit additif gaussien
                              % de moyenne m et de variance v
Y = imnoise(X,'salt & pepper',p) % Applique un bruit poivre et sel de pourcentage p
```

1.2 Etude sous MATLAB

1. Créer sous MATLAB un script vide nommé `TP3Partie1.m`
2. Ouvrir l'image `cameraman.tif`, la stocker dans une matrice `X1` et la renormaliser.
3. Appliquer sur l'image `X1` un bruit blanc Gaussien de variance $\sigma^2 = 0.01$ et stocker le résultat dans une matrice `X2`. Afficher sur la même figure l'image originelle et l'image bruitée. Faire varier σ^2 et commenter.
4. Appliquer sur l'image `X1` un bruit poivre et sel avec un pourcentage $p = 0.05$ de pixels modifiés et stocker le résultat dans une matrice `X3`. Afficher sur la même figure l'image originelle et l'image bruitée. Faire varier p et commenter.
5. Afficher sur une même figure les images `X1`, `X2` et `X3`. Comparer les effets des deux dégradations et commenter.
6. Tracer sur la même figure la ligne numéro 128 des images `X1`, `X2` et `X3` et commenter.

Pour tracer plusieurs signaux sur la même figure (chacun dans une couleur différente), on peut soit utiliser la commande `hold on`, soit rajouter des instructions dans la fonction `plot`

```
% x1, x2, x3 : trois signaux de taille N
% Facon 1
figure
plot(1:N,x1)
hold on
plot(1:N,x2,'g')
plot(1:N,x3,'r')
% Facon 2
figure
plot(1:N,x1,1:N,x2,1:N,x3)
```

2 Filtrage d'une image : domaine spatial

2.1 Rappels de cours

Le filtrage peut être vu comme une opération transformant une image en une autre image ayant des propriétés spatiales et fréquentielles différentes. On distingue deux types de filtrage :

- Le **filtrage linéaire** est une opération de convolution en 2D transformant une image en une autre en général de même taille. Il est défini par une matrice $h(m,n)$ de taille $M_h \times N_h$ appelée **masque de convolution** (en général $M_h = N_h$). Le filtrage linéaire revient à remplacer la valeur de chaque pixel par une moyenne pondérée calculée avec les pixels voisins. Le masque contient les coefficients de pondérations de chacun des pixels. Dans le domaine fréquentiel, par opposition au filtrage non-linéaire, le filtrage linéaire ne fait pas apparaître de puissance sur une fréquence là où il n'y en avait pas. En revanche, il permet d'augmenter ou de diminuer l'énergie sur telle ou telle fréquence.
- Il existe également des **filtres non-linéaires** utilisés par exemple pour diminuer un bruit spécifique. Il s'agit encore une fois de remplacer la valeur de chaque pixel à partir des pixels voisins. En revanche, contrairement au filtrage linéaire, l'opération réalisée sur les pixels voisins est cette fois ci non-linéaire (par exemple une médiane ou une opération *ad hoc*).

- Pour réaliser un filtrage linéaire, il faut d'abord définir le masque `h` à utiliser. Pour cela, soit on le définit de façon analytique, soit on utilise la fonction `fspecial` de MATLAB

```
% Methodes analytiques
h = ones(3,3)/9; % Filtre moyenneur de taille 3 x 3
h = [1 0 1 ; 0 2 0 ; 1 0 1]/6; % Filtre ad hoc
% Methodes en utilisant fspecial
h = fspecial('average',[3 3]); % Filtre moyenneur de taille 3 x 3
h = fspecial('gaussian',[15 15],1); % Filtre gaussien de taille 15 x 15
                                    % et d'ecart type 1
```

- On réalise ensuite le filtrage de l'image grâce à la commande `imfilter` :

```
% X : image renormalisee (valeurs entre 0 et 1)
% h : masque de convolution
Y = imfilter(X,h,'replicate');
```

- Le filtrage non-linéaire que nous allons considérer ici est le filtrage médian, qui peut être réalisé grâce à la commande `medfilt2` :

```
% X : image renormalisee (valeurs entre 0 et 1)
Y = medfilt2(X,[3 3]); % Filtrage median de taille 3 x 3
```

2.2 Etude sous MATLAB

1. Créer sous MATLAB un script vide nommé `TP3Partie2.m`
2. Reprendre les images `X1`, `X2` et `X3` précédemment définies. Pour l'image `X2` on prendra $\sigma^2 = 0.01$, et pour `X3`, on prendra $p = 0.05$.
3. Appliquer un filtre moyenneur de taille 3×3 sur l'image `X2` et stocker le résultat dans `Y2`. Afficher sur la même figure `X1`, `X2` et `Y2`. Le bruit a-t-il été atténué ?
4. Appliquer un filtre médian de taille 3×3 sur l'image `X3` et stocker le résultat dans `Y3`. Afficher sur la même figure `X1`, `X3` et `Y3`. Le bruit a-t-il été atténué ?
5. Afin de pouvoir quantifier la qualité du débruitage, on va utiliser une mesure objective appelée Peak Signal to Noise Ratio (PSNR) et définie par :

$$PSNR = 10 \log_{10} \left(\frac{R^2}{\frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} \left[x^o(m,n) - x^d(m,n) \right]^2} \right)$$

où x^o et x^d sont respectivement les images originelles et débruitées et où R est la dynamique du signal (valeur maximale possible pour un pixel). Cette métrique est très largement utilisée pour évaluer les méthodes de compression et de débruitage d'images. Si le PSNR est utile pour mesurer la proximité de l'image débruitée par rapport à l'original au niveau du signal, il ne prend pas en compte la qualité visuelle de reconstruction et ne peut être considéré comme une mesure objective de la qualité visuelle d'une image.

Si l'on travaille sur des images renormalisées, on a $R = 1$ et le PSNR peut se calculer facilement grâce à l'instruction :

```
% X : image originelle, Y : image debruitee
PSNR=-10*log10(std2(X-Y));
```

6. Calculer le PSNR pour les deux simulations précédemment réalisées. Sachant qu'on considère en général qu'un excellent débruitage offre un PSNR d'au moins 20 dB, les résultats vous semblent-ils logiques ?

7. Tester les 10 filtres suivants sur `X2`, puis sur `X3`. Lequel donne les meilleures performances sur `X2` ? sur `X3` ?

   (a) Filtre moyenneur : 3×3, 5×5 et 7×7

   (b) Filtre Gaussien de taille 15×15 : $\sigma_h = 2$, $\sigma_h = 1.5$, $\sigma_h = 1$ et $\sigma_h = 0.5$

   (c) Filtre médian : 3×3, 5×5 et 7×7

3 Filtrage d'une image : domaine fréquentiel

3.1 Rappels de cours

Le filtrage linéaire consiste en un produit de convolution dans le domaine spatial, ce qui correspond à une multiplication dans le domaine spectral. On s'interesse donc souvent à la réponse fréquentielle d'un filtre pour savoir notamment quelles fréquences il va amplifier, quelles directions privilégiées il va mettre en évidence, etc... En particulier, en observant la transformée de Fourier du masque de convolution (éventuellement complété par des zéros), on arrive à observer le comportement fréquentiel du filtre. Tout comme la transformée de Fourier d'une image classique, on peut représenter la réponse fréquentielle en échelle linéaire ou en échelle logarithmique.

3.2 Etude sous MATLAB

1. Créer sous MATLAB un script vide nommé `TP3Partie3.m`

2. Ouvrir l'image `cameraman.tif`, la stocker dans une matrice `X1` et la renormaliser. Générer un masque `h1` correspondant à un filtre moyenneur de taille 3×3.

3. Utiliser la fonction `AffichageFiltrage(X,h)` pour afficher les spectres de l'image originelle, de l'image filtrée ainsi que la réponse en fréquence du filtre.

La fonction `AffichageFiltrage(X,h)` fournie réalise les étapes suivantes :

- Prend en entrée une image renormalisée `X`, et un masque de convolution `h`
- Calcule l'image filtrée `Y`
- Affiche sur la même image, de gauche à droite et de haut en bas :
  - L'image originelle *(en haut à gauche)*
  - Le spectre de l'image originelle en échelle linéaire *(en haut au milieu gauche)*
  - Le spectre de l'image filtrée en échelle linéaire *(en haut au milieu droit)*
  - La réponse en fréquence du filtre en échelle linéaire *(en haut à droite)*
  - L'image filtrée *(en bas à gauche)*
  - Le spectre de l'image originelle en échelle logarithmique *(en bas au milieu gauche)*
  - Le spectre de l'image filtrée en échelle logarithmique *(en bas au milieu droit)*
  - La réponse en fréquence du filtre en échelle logarithmique *(en bas à droite)*

4. Quel effet le filtre a-t-il sur le spectre ? S'agit-il d'un filtre passe-bas, passe-haut, etc... ? Met-il en évidence des directions particulières ?

5. Refaire la même expérience avec un filtre moyenneur de taille 5×5 et de taille 7×7 et commenter.

6. Refaire la même expérience avec un filtre Gaussien de taille 15×15 et d'écart type $\sigma_h = 2$, $\sigma_h = 1.5$ et $\sigma_h = 1$. Commenter.