Analyse hilbertienne et analyse de Fourier
ENS de Cachan
Département de Mathématiques Vendredi 20 novembre 2009

Feuille de TP numéro 2

1 Rappels

- **- Consignes pour le rapport de TP :** cette feuille de TP comporte des questions auxquelles il est demandé de répondre dans un rapport de TP.

- **- Barème :** la moyenne des notes obtenues pour les rapports des quatre TP contribuera à la note de partiel à hauteur de 6 points sur 20.

- **- Format du rapport :** les rapports de TP sont à rendre sous forme de fichier .pdf rédigés avec LaTeX.

- **- Remise du rapport :** le rapport de ce TP doit être impérativement rendu avant le **vendredi 4 novembre 2009**. Les rapports, au format .pdf, doivent être envoyés par email à l'adresse **sabater@cmla.ens-cachan.fr**.

2 Du bruit dans les images : le modèle additif

Dans ce TP, on s'intéressera à la réduction du bruit dans les images. Le bruit n'est pas qu'un phénomène acoustique, on peut aussi en trouver dans le domaine visuel. Par exemple, quand l'antenne d'un poste de télévision est mal orientée, l'image à l'écran est de très mauvaise qualité et contient des parasites, très désagréables à l'oeil, que l'on appelle du bruit. Le bruit peut donc être naturellement présent dans la vidéo, mais aussi dans les images fixes, prises par exemple avec un appareil photographique.

Ici nous considérons un modèle additif de bruit :

$$v = u + n \quad avec \quad n \sim N(0, \sigma^2 Id)$$

où u est l'image originale, v est l'image bruitée et n est le bruit i.i.d. (indépendant et identiquement distribué, c'est-à-dire que le bruit en un pixel est indépendant et a la même loi de probabilité que le bruit en n'importe quel autre pixel), gaussien (la loi du bruit le chaque pixel est gaussienne) et indépendant de l'image originale. Nous supposerons que la moyenne du bruit est nulle et de variance σ^2 (ce qui représente en quelque sorte sa puissance). Ce modèle est illustré dans la figure 1.

► **Question 1 :** A l'aide de la commande `randn` créez une image d'un bruit gaussien de moyenne nulle et de variance 2.5. Quel allure a le spectre de ce bruit ? Bruitez une image avec ce bruit et visualisez son spectre. Qu'observez-vous ?

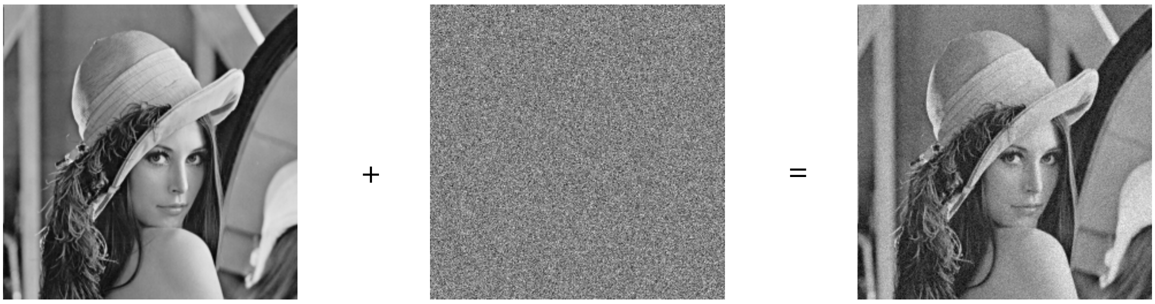


Figure 1: Ajout artifiel de bruit blanc gaussien.

3 Déconvolution

Maintenant on s'intéresse à une dégradation de l'image de type convolutive $v = u * K$

$$v(x) \; = \sum_{x \in supp(K)} u(t-x)K(x) \; , \tag{1}$$

et on cherche à retrouver u à partir de v.

Pour la mise en pratique, considérons le cas en une dimension (le cas des signaux). Soit u le signal créneau qui se définit sous Matlab comme

```
t = -15:0.1:14.9;
u = (abs(t)<5)*1;
```

► **Question 2 :** En utilisant la fonction `convol.m` étudiez la dégradation convolutive du signal u en considérant le noyaux K constant égal à 1

```
K = ones(1,10);
```

► **Question 3 :** Ecrivez l'équation (1) dans le domaine fréquentiel et exprimez $F(v)$ en fonction de $F(u)$ et $F(K)$. Créez une fonction `function [res] = Fconvol(signal,K)` qui réalise la dégradation convolutive dans le domaine fréquentiel. Observez le résultat de la convolution du signal u précédent avec un noyau Gaussien

```
s = -15:0.1:14.9;
K= exp(-(s).^2./10);
```

► **Question 4 :** Créez une fonction `function [res] = iconvol(signal,K)` qui réalise une déconvolution (opération inverse à la convolution), dans le domaine de Fourier. Vérifiez que, après la composition des deux, vous obtenez bien le signal d'origine.

3.1 Influence du bruit sur la déconvolution et filtre de Wiener

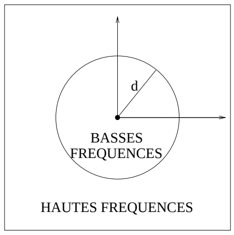
La dégradation convolutive s'accompagne en général de bruit : $v = u * K + n$.

▶ **Question 5 :** Simulez un bruit avec la commande `randn` et ajoutez-le au signal convolé $u*K$. Maintenant, que se passe-t'il si on réalise une déconvolution ?

▶ **Question 6 :** A l'aide de la fonction `compar_deconvol_wiener.m` comparez le filtre de Wiener par rapport à la déconvolution. Regardez en détail cette fonction et expliquez ce qu'elle fait.

4 Les filtres passe-bas et passe-haut

Nous allons étudier deux types de filtre, le filtre passe-bas et le filtre passe-haut. Le filtre passe-bas n'affecte pas les composantes de basse fréquence dans les données d'une image, mais doit atténuer les composantes de haute fréquence. Le filtre passe-haut a les caractéristiques inverses, il n'affecte pas les composantes de haute fréquence mais atténue les basses fréquences.



Domaine fréquentiel

▶ **Question 7 :** Construisez une fonction `filtre_passe_bas(I,d)` qui prend en entrée une image I et une amplitude d et qui affiche l'image d'entrée et l'surimage filtrée par un filtre passe-bas sur la même figure (utilisez `subplot`et la fonction `vis_img_mat.m` pour l'affichage.) Aide : Vous pouvez créer le filtre directement dans le domaine fréquentiel avec des valeurs 0 où 1 uniquement.

▶ **Question 8 :** Construisez une fonction `filtre_passe_haut(I,d)` qui prend en entrée une image I et une amplitude d et qui affiche l'image d'entrée et l'image filtrée par un filtre passe-haut sur la même figure.

▶ **Question 9 :** Quel est l'effet de ses deux filtres sur l'image ? Faites une discusion dans les deux cas suivant les valeurs du paramètre d.

5 Transformée de cosinus discrète (TCD)

Souvent il est intéressant de faire une symétrisation d'un signal discret $u(k), k = 0,\cdots,N-1$ avant de faire sa transformée de Fourier pour éviter les effets de bords. Le nouveau signal est donc $h(k) = h(2N-1-k) = u(k) pour 0 \leqslant N-1$.
é Si la transformée de cosinus discrète de u se définit comme

$$\forall p \in Z \qquad C_u(p) = \sum_{k=0}^{N-1} u(k) cos\left(\frac{\pi p}{N}(k+\frac{1}{2})\right),$$

alors $\hat{h} = 2 * e^{\frac{i\pi p}{2N}} C_u(p)$.

▶ **Question 10 :** En regardant les fonctions `my_dct.m` et `my_dct2.m` expliquez comment la TCD est calculée en pratique en 1 et 2 dimensions. Pourquoi `my_dct2.m` utilise uniquement la fonction `my_dct.m` ? Quels avantages voyez-vous à la TCD par rapport à la TFD ?

Considérez l'image de matrice A définie de la façon suivante et faites sa TCD.

```
 N = 256;
for i=1:N
    for j=1:N
        A(i,j) = 255/(3*N)*(2*i+j);
    end
end
A = 255*(A>128);
```

► **Question 11 :** Que se passe-t'il si vous gardez uniquement les $j \times j$ coefficients de la TCD avant de refaire la transformation inverse ? Vous pouvez construire une boucle `for` où l'on ferait ce calcul pour différents valeurs de j. Note : Utilisez la fonction `my_idct2.m` pour réaliser la TCD inverse.

6 Algorithme JPEG

JPEG est un format de représentation des images. L'algorithme sépare l'image en blocs 8×8 et calcule la TCD sur chacun de ces blocs. Après, les hautes fréquences, celles auxquelles l'oeil humain est très peu sensible, sont ramenées à 0. L'intérêt est qu'une longue suite de zéros nécessite très peu de place pour être stoquée dans un fichier.

La fonction `[code,sizes] = jpegenc(I)` permet de coder en JPEG l'image I et la fonction `[I_decode] =jpegdec(code,sizes)` fait l'opération de décodage.

► **Question 12 :** Etudiez l'effet de ce codage sur plusieurs images. Ce type de codage produit-il des pertes sur l'image ? Quels artefacts apparaissent en restituant une image codée ?