

Débruitage

C. Dossal

Avril 2013

Introduction

Dans cette feuille par simplicité, nous ne considérerons que des bruits blancs gaussiens (1D ou 2D). Cette hypothèse nous conduira à utiliser la norme ℓ_2 , le SNR ou le PSNR comme mesure de qualité de nos images et de nos signaux *débruités*. Nous allons essayer de voir dans différentes situations comment l'utilisation de bases orthonormées ou de *frames*, représentations redondantes en français peut permettre des débruitages efficaces.

Avant de mettre en place les différentes méthodes de débruitage, il faut rédiger une fonction *psnr* qui calcule le *psnr* d'une image ou d'un signal dégradé par rapport à une image ou un signal de référence.

1 Débruitage de signaux 1D

Nous allons utiliser dans un premier temps un extrait court de son de saxophone (100 000 points)

1. Extraire un son de saxophone de 100 000 points (à partir par exemple de *sax greff do4 3.wav*) et le bruiteur avec un bruit blanc gaussien de variance 10^{-4} (d'écart type 10^{-2}). Ecouter les deux sons.

1.1 Débruitage par convolution gaussienne.

Une méthode classique de débruitage consiste à effectuer la convolution du signal bruité par une gaussienne. Nous allons d'abord implémenter cette méthode pour la comparer à d'autres méthodes plus évoluées.

2. Ecrire un programme matlab

```
function Srec=DebruitGaussien(SB,sigma)
```

qui prend en entrée un son bruité *SB* et une variance *sigma* et qui renvoie un signal débruité obtenu par filtrage par une gaussienne de variance *sigma*.

3. Ecrire un programme qui affiche le psnr de la reconstruction en fonction de *sigma* et déterminer la valeur de *sigma* optimale.
4. Tester plusieurs valeurs d'intensité de bruit. Comment évolue la valeur de *sigma* optimale en fonction de l'intensité du bruit ?

1.2 Débruitage par seuillage dans la base de Fourier.

On peut montrer que si l'image ou le signal que l'on souhaite *débruiter* est essentiellement concentré sur un petit nombre de coefficients dans une base orthonormée donnée, un seuillage des coefficients dans cette base permet un débruitage efficace.

5. Ecrire un programme matlab

```
function Srec=DebruitSeuillageFourier(SB,T)
```

qui prend en entrée un son bruité SB et un seuil T et qui renvoie un signal débruité en seuillant les coefficients de Fourier de SB à la valeur T .

6. Faire différents tests avec différentes valeurs de T et écrire un code matlab qui affiche le psnr de la reconstruction en fonction de T et déterminer la valeur de T optimale.
7. Comparer les deux méthodes en psnr et en qualité auditive en testant différents niveaux de bruit.
8. Effectuer la même comparaison sur le son de guitare proposé sur ma page web. Que pensez vous des résultats ?

1.3 Utilisation de la transformée de Fourier à court terme.

Dans le cas d'un son non stationnaire (plusieurs notes jouées les unes après les autres par exemple), l'utilisation d'une fft globale n'est pas optimale. On peut obtenir de meilleurs résultats en utilisant une transformée de Fourier à fenêtres. Plus précisément, on peut seuiller les coefficients du spectrogramme.

9. Ecrire un programme matlab

```
function Srec=DebruitSpec(SB,N,T)
```

qui effectue un débruitage du son SB en seuillant le spectrogramme calculé avec des fenêtres de taille N avec un seuil T .

10. Déterminer une valeur optimale du seuil en terme de psnr et de qualité auditive (ce qui reste subjectif).
11. Comparer ces résultats aux précédents résultats obtenus dans le cas du son de saxophone et de la guitare.

2 Débruitage d'images

Pour les images, nous allons travailler pour commencer sur *Lenna*. Vous en trouverez une version en .raw sur ma page web. Pour l'ouvrir sous matlab, utilisez la commande suivante :

```
par = [256,256];fid=fopen('lenna.raw','r');I=fread(fid,par);fclose(fid);
```

L'image se trouve alors dans le tableau I , c'est un tableau 256×256 .

12. Visualiser l'image en niveau de gris (colormap('gray')) et construire une image bruitée en ajoutant un bruit aléatoire d'écart type 10. Visualiser l'image bruitée.

2.1 Débruitage par convolution.

13. Ecrire un programme matlab qui effectue un débruitage par convolution avec une gaussienne et qui calcule le psnr associé à la reconstruction.
14. Ecrire un programme matlab qui calcule la variance de la gaussienne assurant la meilleure reconstruction en terme de psnr et le psnr optimal.

2.2 Seuillage dans une base de DCT locale.

Comme dans le cas des signaux 1D, un seuillage dans une base bien choisie permet un débruitage efficace. Nous allons tester une base de DCT locale 2D construite sur des carrés 8×8 , celle qui est utilisée dans jpeg.

15. Ecrire un code matlab qui effectue une dct locale sur des carrés 8×8 . On utilisera la commande `dct2`.

```
function D=DctLocale(I)
```

Comparer la somme des carrés des coefficients de cette transformée avec la somme des carrés de l'image originale. Que constatez vous ?

16. Tester la fonction sur *lenna* et éventuellement sur d'autres images de référence.
17. Ecrire une fonction qui réalise l'opération inverse :

```
function Irec=IDctLocale(D)
```

qui reconstruit une image à partir des coefficients de dct locale.

18. Ecrire un programme matlab qui effectue un débruitage par seuillage dans cette base de DCT avec un seuil T .

```
function Irec=SeuillageDCTLocale(I,T)
```

19. Ecrire un programme matlab qui calcul la valeur de T optimale en terme de psnr et qui donne le psnr associé. On pourra afficher la valeur du psnr de la reconstruction en fonction de T .
20. Comment se comparent ces résultats avec l'approche par convolution ? Du point de vue du psnr et du point de vue visuel ? Les défaut sont ils les mêmes ?

2.3 Utilisation d'une union de bases orthonormées.

Il est possible d'améliorer encore la qualité de la reconstruction en utilisant une union de bases orthonormées. L'idée est de débruiter l'image dans plusieurs bases orthonormées et d'effectuer ensuite la moyenne de ces débruitages pour obtenir une meilleure qualité d'image. Ici nous allons utiliser le fait que la transformée en dct locale n'est pas invariante par translation c'est à dire que si on effectue une translation de l'image d'un pixel, les valeurs absolues des coefficients de la DCT locale vont être modifiés. Ce qui n'est pas le cas pour une fft globale sur l'image.

21. A l'aide de la commande *circshift*, écrire un programme matlab qui effectue le débruitage de la fonction dans une base de DCT locale translatée de l pixels vers la gauche et k pixels vers le bas :

```
function Irec=SeuillageDCTLocaleTrans(I,T,k,l)
```

Attention : on utilisera deux fois la commande *circshift*.

22. Ecrire un programme matlab qui effectue un débruitage en utilisant plusieurs translations et en effectuant la moyenne des reconstruction :

```
function Irec=SeuillageDCTLocaleTransTotal(I,T,k,l)
```

23. Calculer la valeur du seuil optimal et le psnr associé. Comparer cette approche aux précédentes.