

CES8AH - École des Mines de Nancy

# De la photographie numérique à la photographie computationnelle

## Séance 8

Frédéric Sur

Le but de ce TP est d'expérimenter le débruitage d'images par diverses méthodes.

### 1 Filtre médian et bruit impulsif

Expérimentez le filtre médian sur une image affectée d'un bruit impulsif. Vous utiliserez la fonction Matlab `medfilt2` et vous chargerez l'image disponible ici :

[http://en.wikipedia.org/wiki/Salt-and-pepper\\_noise](http://en.wikipedia.org/wiki/Salt-and-pepper_noise)

Vous pouvez aussi générer une image bruitée à l'aide de `imnoise` comme expliqué dans l'aide de `medfilt2`.

Vous expérimenterez différentes tailles du voisinage.

### 2 Filtre linéaire

Expérimentez le filtrage gaussien sur une image affectée par un bruit blanc gaussien généré par vous-même (toujours avec `imnoise`).

Vous utiliserez la fonction `imfilter` et créez un filtre gaussien d'écart-type `sigma` par :

```
>> h=fspecial('gaussian',[6*sigma,6*sigma],sigma);
```

Essayez des valeurs de `sigma` entre 1 et 10, et constatez qu'un compromis doit être réalisé entre débruitage et préservation des contours de l'image.

Le filtre de Wiener adaptif fourni par `wiener2` améliore-t-il la situation ? La description de l'algorithme est disponible à l'aide de `doc wiener2`. Comprenez-vous comment l'algorithme permet de préserver les contours ?

### 3 Diffusion anisotrope

Une implantation Matlab de la diffusion anisotrope est disponible ici :

<http://www.peterkovesi.com/matlabfns/Spatial/anisodiff.m>

Regardez dans l'aide de cette fonction le rôle des différents paramètres. Constatez que les deux choix en cours pour la fonction `c` sont disponibles. À quoi correspond le paramètre `lambda` ? Vous pouvez fixer `lambda=0.1`. Le nombre d'itérations va fixer l'« instant » d'arrêt de la diffusion. Vous pouvez essayer des valeurs de `niter` entre 20 et 50.

Vous pouvez aussi décommenter les deux dernières lignes comme suggéré dans le code source.

*Pour ceux qui avancent vite* : adaptez le code pour qu'il implante la version *régularisée* du modèle de Perona-Malik, qui consiste à calculer le gradient non pas comme une différence finie simple, mais comme la convoluée de l'image avec la dérivée d'un noyau Gaussien.

## 4 Débruitage TV-L2

La fonction `TVdenoising.m` fournie implémente le débruitage de la fonctionnelle TV-L2 vue en cours.

Vous pouvez fixer `Niter` à 5000, et `lambda` à 10. Faites des essais aussi avec des valeurs bien plus grandes ou plus petites de `lambda`.

## 5 Algorithme des moyennes non locales

Une démonstration en ligne du code des moyennes non-locales, par ses auteurs, est disponible ici : [http://www.ipol.im/pub/art/2011/bcm\\_nlm/](http://www.ipol.im/pub/art/2011/bcm_nlm/)

Comparez aux résultats des algorithmes précédents. Attention, les images que vous soumettez à la démonstration en ligne sur IPOL sont enregistrées et publiquement disponibles.

### Références

[1] P.D. Kovesi, MATLAB and Octave Functions for Computer Vision and Image Processing. Available from : <http://www.peterkovesi.com/matlabfns/>.

[2] A. Buades, B. Coll, and J.-M. Morel, Non-local means denoising. *Image Processing On Line*, vol. 1, 2011. Available from : [http://www.ipol.im/pub/art/2011/bcm\\_nlm/](http://www.ipol.im/pub/art/2011/bcm_nlm/).