

De la photographie numérique à la photographie computationnelle

CES8AH - École des Mines de Nancy

Séance 9

Frédéric Sur

Le but de ce TP est d'expérimenter différentes méthodes de la restauration non-aveugle d'images, et d'établir leurs performances respectives.

Rappel. Vous pouvez appliquer un filtre (de dégradation) à une image par `imfilter`, filtre que vous aurez défini préalablement par `fspecial`. Notez que `fspecial` permet en particulier de générer des filtres gaussiens, ou des filtres simulant un flou de bougé selon une direction. Vous pouvez aussi ajouter un bruit (ici Gaussien) à une image à l'aide de `imnoise`. Notez la définition de la variance à passer en paramètre à cette fonction.

Pour éviter les différentes plaisanteries dues à la gestion des types entier / double, convertissez vos images à l'aide de `im2double` (attention à la normalisation effectuée) au moment de leur lecture, et affichez les par `imshow(I)`.

Pour chaque expérience, il faut essayer différentes images non dégradées, différents paramètres, et différentes valeurs de l'amplitude du bruit, afin de comprendre le comportement des algorithmes de restauration.

1 Déconvolution directe

Dégradez une image de votre choix par convolution et ajoutez un bruit blanc gaussien d'amplitude croissante. Expérimentez la déconvolution directe, par exemple :

```
>> I = im2double(imread('cameraman.tif'));
>> figure, imshow(I);
>> H = fspecial('motion',20,40); % essayez d'autres filtres
>> fftH=psf2otf(H, size(I));
>> I_blur= imfilter(I,H,'conv','circular');
>> I_blurnoise= imnoise(I_blur,'gaussian',0,0); % changez la variance (ici 0)
>> figure, imshow(I_blurnoise);
>> I_deconvdirect = ifft2(fft2(I_blurnoise)./fftH);
>> figure, imshow(I_deconvdirect);
```

2 Filtre de Wiener

Expérimentez ce qui est décrit sur ce blog officiel Matlab :
<http://blogs.mathworks.com/steve/2007/11/02/image-deblurring-wiener-filter/>
et adaptez ce qui est proposé à l'exemple précédent.

3 Approche par régularisation

Expérimentez ce qui est décrit ici :

<http://blogs.mathworks.com/steve/2008/07/21/image-deblurring-using-regularization/>

4 Approche par variation totale

Testez le code `deconvtv` (voir [1]) disponible sur la page web suivante :

<http://scholar.harvard.edu/stanleychan/software/deconvtv-fast-numerical-solver-total-variation-minimization>

5 Algorithme de Richardson-Lucy

Expérimentez cet algorithme, en particulier l'évolution selon le nombre d'itération : voir `doc deconvlucy`, et :

<http://fr.mathworks.com/help/images/deblurring-with-the-lucy-richardson-algorithm.html>

6 Déconvolution aveugle

Vous pouvez expérimenter la fonction Matlab `deconvblind` :

`doc deconvblind`

<http://fr.mathworks.com/help/images/deblurring-with-the-blind-deconvolution-algorithm.html>

Références

[1] S. H. Chan, R. Khoshabeh, K. B. Gibson, P. E. Gill, and T. Q. Nguyen, "An augmented Lagrangian method for total variation video restoration", IEEE Transactions on Image Processing, vol. 20, no. 11, p. 3097-3111, 2011