

Module VIS - Master MITIC TP3 : Manipulation d'images

O. LE MEUR
olemeur@irisa.fr

16 janvier 2017

But du TP : FILTRAGE SPATIAL D'IMAGES

Travail à rendre :

Le travail est à rendre pour la semaine suivante. Pour chaque question, on donnera les commandes et traitements effectués, les images et valeurs obtenues et les commentaires correspondants.

L'ensemble doit être rendu *sous forme électronique* (au format PDF, Word ou Open Office). Ce compte rendu est à envoyer à votre encadrant : olemeur@irisa.fr. Si vous travaillez à deux, le nom du fichier doit inclure les noms des binômes.

Vous trouverez le sujet et les ressources associées à ce tp sur le lien suivant :

http://people.irisa.fr/Olivier.Le_Meur/teaching/MITIC/TP3_ressources.rar

http://people.irisa.fr/Olivier.Le_Meur/teaching/MITIC/TP3.pdf

Rappel

La convolution discrète est telle que, pour un pixel $g(i, j)$ de l'image filtrée :

$$g(i, j) = (f * h)(i, j) = \sum_{n=-\infty}^{+\infty} \sum_{m=-\infty}^{+\infty} f(n, m) \times h(i - n, j - m)$$

où f est l'image initiale et h le filtre, appelé aussi masque de convolution.

1 Filtrage passe-bas d'une image couleur

1.1 Filtrage moyenneur

On se propose de filtrer l'image source avec un noyau de convolution de taille 3×3 . Les coefficients de ce noyau sont tous égaux à 1 (utiliser la fonction *ones* pour créer cette matrice). Ne pas oublier le facteur de normalisation. En utilisant la fonction *imfilter*, vous pourrez réaliser le filtrage. Visualiser le résultat.

Si on augmente la taille de la fenêtre de filtrage, que se passe-t-il ?

1.2 Filtrage Gaussien

Pour ce type de filtrage, vous pouvez utiliser la fonction *fspecial*. Réaliser plusieurs tests sur des images différentes et commenter. Faire varier la variance du filtre Gaussien. Commenter.

2 Détection de contours

2.1 Gradient

On définit les opérateurs gradient suivant (attention, ils sont différents) :

$dx = [1; -1];$

$dy = [1, -1];$

- Appliquer ces opérateurs à une image en niveau de gris et à une image synthétique 20×20 (une ligne noire sur un fond blanc). Pour réaliser la convolution, vous utiliserez la fonction *conv2*.

Pour faciliter la visualisation, redimensionnez l'image source comme indiqué ci-dessous :

```
AFull=imread('mandrill_lumi.bmp');
```

```
A = imresize(AFull, 0.25);
```

- Calculer le module du gradient et conserver uniquement les valeurs supérieures à un seuil prédéfini par exemple 50 (les boucles sont toujours interdites!).

Vous visualiserez l'orientation du gradient ainsi que l'isophote en utilisant le code suivant (annuler les valeurs des gradients sur les bords de l'image pour la visualisation) :

```
figure;clf;
imagesc(A);colormap(gray);axis image; % A image source
hold on;
scale_quiver = 10; % 10 for natural image; 2 for the synthetic black and white image
% AA_x image gradient dx, AA_y image gradient dy
quiver(AA_x,AA_y,scale_quiver,'b'); % gradient vector orthogonal to isophote
quiver(-AA_y,AA_x,scale_quiver,'r'); % isophote (orthogonal to gradient vector) ;
% linespec . draws a line instead of arrow
quiver(AA_y,-AA_x,scale_quiver,'r');
```

2.2 Laplacien

On se propose de détecter les contours d'une image en utilisant l'opérateur Laplacien. Appliquer les filtres suivants sur l'image originale en utilisant la commande *imfilter* :

- Laplacien anisotrope 4-connexité :
edgeFilter = [0,-1,0;-1,4,-1;0,-1,0]
- Laplacien isotrope 8-connexité :
edgeFilter = [-1,-1,-1;-1,8,-1;-1,-1,-1]

Vous visualiserez les résultats et vous les discuterez.

3 Sharpening d'une image

L'accentuation des fréquences hautes d'une image (en anglais *Sharpening*) permet dans bien des cas d'améliorer la qualité de l'image. On se propose de tester une méthode de *sharpening* basée sur la détection de contours.

Cette méthode est composée des étapes suivantes :

- Filtrer l'image avec un filtre passe-bas. Vous utiliserez le filtre suivant (ne pas oublier de le normaliser) :
gaussianFilter = [1,4,7,4,1;4,20,33,20,4;7,33,55,33,7;4,20,33,20,4;1,4,7,4,1];
- Détecter les contours de l'image filtrée (vous pouvez utiliser le filtre Laplacien défini dans la section précédente)
- Ajouter ces contours à l'image filtrée passe-bas

A noter qu'il est possible d'obtenir un résultat similaire en appelant la fonction *fspecial*, de la façon suivante :

```
sharpFilter = fspecial('unsharp');
```

4 Débruitage : filtre médian pour un bruit impulsionnel

Le filtre médian est un cas particulier des filtres de rang. Il est souvent utilisé pour débruiter des données corrompues par un bruit impulsionnel. On se propose ici de le tester sur une image bruitée.

- Simuler sur l'image *lena.pgm* un bruit "poivre et sel" d'intensité 0.05 (utiliser *imnoise*). Afficher l'image bruitée résultante.
- Débruiter l'image en utilisant *medfilt2*. Effectuer ainsi le filtrage de l'image bruitée par un filtre médian de taille 3×3 et visualiser le résultat.
- Convoluer l'image bruitée avec un filtre moyenneur de taille 3×3 . Comparer le résultat obtenu avec celui de la question précédente.