

TP2 - Filtrage spatial

Vincent Barra - Christophe Tilmant

2011-2012

1 Partie théorique

1.1 Introduction

Le terme spatial se réfère à l'ensemble des pixels composant l'image. Tout traitement spatial peut s'exprimer comme

$$J(x, y) = T[I(x, y)]$$

où I est l'image originale, J l'image traitée, et T est un opérateur sur I , défini sur un voisinage de (x, y) . Si ce voisinage est de taille 1, T agit pixel par pixel, et on parle alors d'analyse point à point. On récupère alors entre autres les méthodes d'analyse d'histogramme, auxquelles on adjoint toute transformation mathématique $i \mapsto T(i)$ (logarithmique, exponentielle, ...).

Nous envisageons dans la suite un voisinage de taille strictement supérieure à 1, centré en (x, y) . On parlera alors de filtrage spatial (par opposition au filtrage fréquentiel, que nous aborderons lors du prochain TP), et nous nous intéresserons dans ce TP aux filtres linéaires et non linéaires.

Les filtres linéaires sont fondés sur l'hypothèse de linéarité du système d'acquisition. Les filtres dits passe-bas atténuent ou éliminent les hautes fréquences spatiales dans l'image, en laissant les basses fréquences intactes. Au contraire, un filtre passe haut laisse intactes les hautes fréquences, et atténue les basses. Enfin, les filtres passe-bande atténuent ou éliminent une bande de fréquences donnée.

Quel que soit le type de filtrage spatial linéaire envisagé, la démarche est identique. Il s'agit de définir autour d'un pixel (x, y) un voisinage W de taille N impaire, centré en (x, y) , dont les coefficients sont par exemple pour $N = 3$:

$w_{-1,-1}$	$w_{-1,0}$	$w_{-1,1}$
$w_{0,-1}$	$w_{0,0}$	$w_{0,1}$
$w_{1,-1}$	$w_{1,0}$	$w_{1,1}$

Le filtrage consiste alors simplement en le balayage de l'image par le masque, et en l'affectation au pixel (x, y) du niveau de gris résultant de la combinaison linéaire des pixels voisins pondérés par les coefficients du filtre :

$$\sum_{i=-1}^1 \sum_{j=-1}^1 w_{i,j} I(x-i, y-j)$$

Le filtrage non linéaire, quant à lui, est également réalisé à partir de voisinages centrés, mais agit directement à partir des niveaux de gris $I(x+i, y+j)$ par une transformation non linéaire (max, min, médiane...)

1.2 Filtres lissants

1.2.1 Filtrage linéaire

Un masque de lissage a ses coefficients non-négatifs, et leur somme vaut 1. En général, les coefficients peuvent être donnés manuellement, ou par une fonction mathématique (ex : Gaussienne). Parmi les nombreux masques qui peuvent être utilisés, nous donnons ici deux exemples caractéristiques :

L'effet du lissage est de "mélanger" les niveaux de gris dans un voisinage, donc d'engendrer du flou sur l'image. La somme des coefficients du masque valant 1, le lissage préservera toute zone de l'image où le niveau de gris est constant. Par conséquent, ce type de filtrage conserve les zones à faibles variations

1/9	1/9	1/9	1/16	1/8	1/16
1/9	1/9	1/9	1/8	1/4	1/8
1/9	1/9	1/9	1/16	1/8	1/16

(a)
(b)

FIGURE 1 – Deux exemples de masques en filtrage linéaire passe-bas

de niveaux de gris et supprime les variations rapides.

La taille du filtre a bien sur une importance considérable dans le filtrage. La figure 2 présente les résultats d'un filtrage d'une coupe IRM sagittale de cerveau pour des masques de lissage $n \times n$ à coefficients constants (tous égaux à $1/n^2$), pour $n \in \{3, 5, 7\}$:

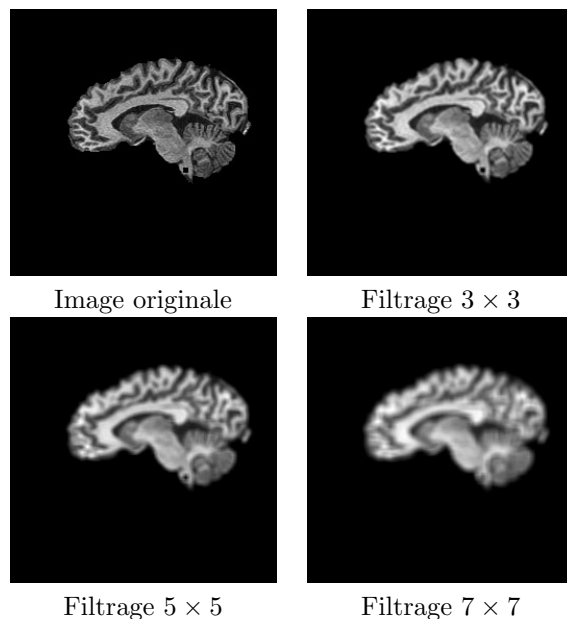


FIGURE 2 – Effet de la taille d'un masque de lissage

Outre l'élimination de petits détails, le lissage linéaire peut s'avérer utile dans certaines circonstances, par exemple :

- Pour filtrer une image bruitée.
- Pour créer des dégradés dans une image dont les niveaux de gris ont été sous-quantifiés
- Avant une détection de contours, pour améliorer le résultat de celle-ci
- Pour restaurer une image en demi-ton

La figure 3 donne un exemple de réduction de bruit (image corrompue par un bruit gaussien) par le filtre (a) de la figure 1.

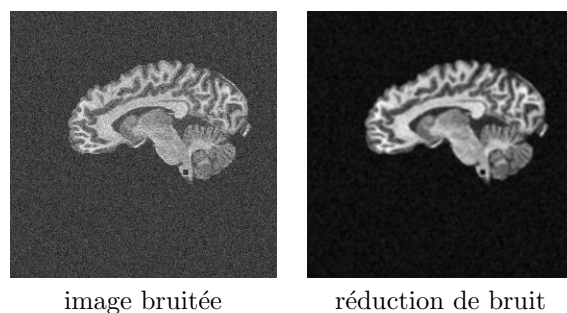


FIGURE 3 – Exemple en réduction de bruit

La figure 4 donne un exemple en correction de sous-quantification.

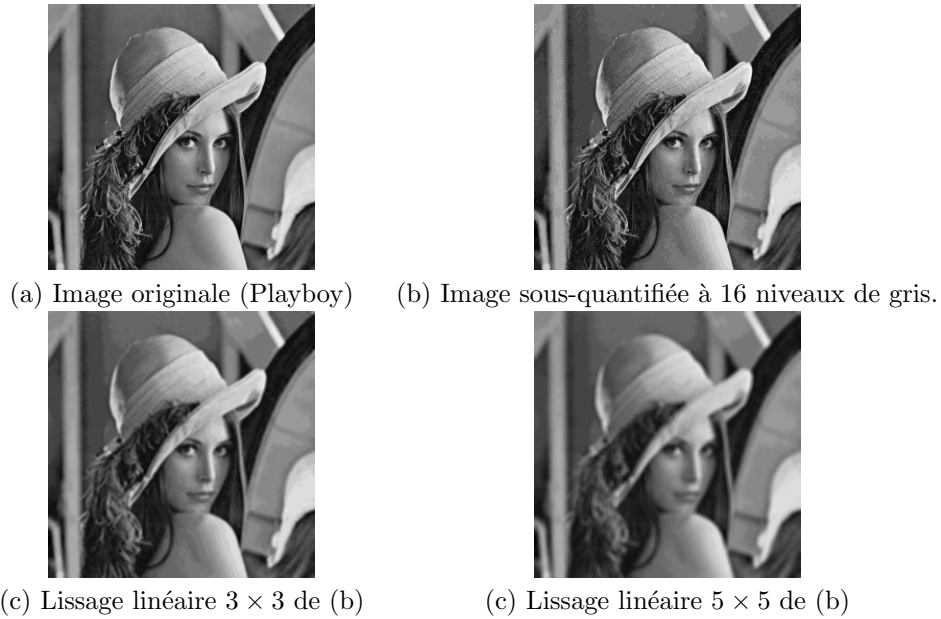


FIGURE 4 – Exemple en correction de sous-quantification

1.2.2 Filtrage non linéaire

Un des principaux inconvénients des méthodes précédentes est que le flou s'applique à la fois aux frontières et aux détails haute fréquence. Si l'objectif du traitement est la réduction du bruit, on aimerait le plus possible préserver les bords naturels dans l'image. La technique du filtrage médian répond à cette problématique, lorsque le bruit est impulsif. Ce filtre a de plus la bonne propriété de laisser inchangées les transitions monotones entre régions (figure 5).

Le principe est simple : il suffit de remplacer la valeur de $I(x, y)$ par la médiane des niveaux de gris dans un voisinage de (x, y)

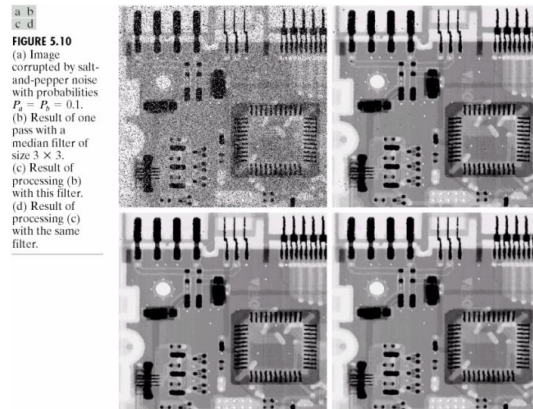


FIGURE 5 – Exemple de filtrage médian (tiré de [2])

Le filtrage médian présente néanmoins quelques inconvénients. Pour des bruits à distribution peu concentrées (gaussien, uniforme), ses performances sont faibles par rapport au filtre linéaire optimal. Par ailleurs, il peut affecter la géométrie des régions de l'image. Les zones présentant un angle aigu ont tendance à être arrondies par le filtrage. On perd ainsi l'information sur les points anguleux.

Une généralisation immédiate du principe du filtre médian, est le filtre d'ordre. Il s'agit d'ordonner de manière croissante les niveaux de gris des pixels contenus dans le voisinage de (x, y) , pour obtenir un vecteur (a_k) , et de retenir pour $I(x, y)$ une fonction de la $j^{\text{ème}}$ valeur après tri a_j ($j^{\text{ème}}$ statistique d'ordre). A partir de ces valeurs, on peut définir un L-filtrage de l'image, qui consiste à appliquer une

combinaison linéaire des statistiques d'ordre $\sum_{k=1}^N c_k a_k$ en remplacement de $I(x, y)$. Afin de ne pas modifier

l'intensité des zones homogènes de l'image, les coefficients doivent satisfaire $\sum_{k=1}^N c_k = 1$.

Dans le cas où l'image est homogène sans transition et localement modélisable par un bruit blanc de densité f et de fonction de répartition F , il est possible d'optimiser le choix des coefficients c_k au sens d'un critère d'erreur quadratique. On peut montrer en effet que la puissance du bruit en sortie d'un L-filtre optimal est toujours inférieure ou au pire égale à celle du meilleur filtre linéaire (filtre moyennneur). Les valeurs des coefficients optimaux dépendent de la forme de la densité f :

- Pour un bruit gaussien de variance σ^2 , $\forall k \quad c_k = \frac{1}{N}$ (filtre moyennneur, de variance $\frac{\sigma^2}{N}$)
- Pour un bruit uniforme, $c_k = \frac{1}{2}$ si $k \in \{1, N\}$, 0 sinon (filtre milieu)

De plus, si $c_k = \chi_{\{k=1\}}$, le filtrage est une érosion, et si $c_k = \chi_{\{k=N\}}$ c'est une dilatation. On retrouve donc les deux opérations basiques de la morphologie mathématique, à partir desquelles on peut également effectuer des filtrages non linéaires (ouverture et fermeture).

1.3 Filtres réhausseurs de contours

1.3.1 Filtrage linéaire

L'accentuation (sharpening) consiste à soustraire l'image lissée de l'image originale. On calcule donc, pour une image I et un masque de lissage W

$$I - W \circ I = (Id - W) \circ I$$

Cela revient donc à appliquer à I le masque $W' = Id - W$, qui possède alors les propriétés suivantes :

- la valeur centrale est $1 - w_{0,0}$
- $-1 \leq w'_{ij} \leq 0$, pour tout $(i, j) \neq (0, 0)$
- $\sum_{i,j=-1}^1 w_{ij} = 0$

La figure 6 donne les filtres correspondants aux filtres lissants de la figure 1

-1/9	-1/9	-1/9	-1/16	-1/8	-1/16
-1/9	8/9	-1/9	-1/8	3/4	-1/8
-1/9	-1/9	-1/9	-1/16	-1/8	-1/16
(a)			(b)		

FIGURE 6 – Deux exemples de masques en filtrage linéaire passe-haut

Puisque la combinaison linéaire peut amener à des valeurs d'intensité négatives, il est nécessaire ici de ramener l'ensemble des intensités calculées sur un intervalle inclus dans \mathbb{R}^+ .

L'accentuation est apparentée à la version discrète du Laplacien (approximation en 4 ou 8-connexité par différences finies des dérivées partielles), ce qui nous amène tout naturellement à la notion de filtrage dérivatif.

1.3.2 Filtrage dérivatif

Le moyennage, modélisé par le filtre (a) de la figure 1 tend à rendre floue l'image. Ce moyennage est l'approximation d'une intégration, et on s'attend donc à ce que la différentiation ait l'effet contraire, à savoir un renforcement des contours de l'image.

L'opérateur de différentiation le plus simple à mettre en place en traitement d'images est le gradient. En effet, une simple approximation par différences finies des dérivées partielles permet, par exemple, de définir une approximation de la norme du gradient au pixel (x, y)

$$\sqrt{(I(x, y) - I(x + 1, y))^2 + (I(x, y) - I(x, y + 1))^2}$$

ou

$$|I(x, y) - I(x + 1, y)| + |I(x, y) - I(x, y + 1)|$$

Il est également possible d'approcher cette norme par différences croisées :

$$\sqrt{(I(x, y) - I(x + 1, y + 1))^2 + (I(x + 1, y) - I(x, y + 1))^2}$$

ou

$$|I(x, y) - I(x + 1, y + 1)| + |I(x + 1, y) - I(x, y + 1)|$$

auquel cas on aboutit aux filtres de Roberts. Enfin, en utilisant un voisinage de taille 3×3 , il est possible d'approcher la norme du gradient par les opérateurs de Prewitt en 4 ou 8-connexité (fig 7(a)) ou Sobel (fig 7(b))

<table><tr><td>-1</td><td>-1</td><td>-1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	-1	-1	-1	0	0	0	1	1	1	<table><tr><td>-1</td><td>0</td><td>1</td></tr><tr><td>-1</td><td>0</td><td>1</td></tr><tr><td>-1</td><td>0</td><td>1</td></tr></table>	-1	0	1	-1	0	1	-1	0	1
-1	-1	-1																	
0	0	0																	
1	1	1																	
-1	0	1																	
-1	0	1																	
-1	0	1																	
Deux des 8 Filtres de Prewitt	(à rotation près)																		
<table><tr><td>-1</td><td>-2</td><td>-1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>2</td><td>1</td></tr></table>	-1	-2	-1	0	0	0	1	2	1	<table><tr><td>-1</td><td>0</td><td>1</td></tr><tr><td>-2</td><td>0</td><td>2</td></tr><tr><td>-1</td><td>0</td><td>1</td></tr></table>	-1	0	1	-2	0	2	-1	0	1
-1	-2	-1																	
0	0	0																	
1	2	1																	
-1	0	1																	
-2	0	2																	
-1	0	1																	
Deux des 8 Filtres de Sobel	(à rotation près)																		

FIGURE 7 – Masques de Prewitt et de Sobel

1.3.3 Quelques filtres non linéaires

Terminons en vrac par la définition de quelques filtres non linéaires :

- Filtres adaptatifs : ces filtres s'inspirent des filtres moyenneurs, mais chaque terme de la moyenne est pondéré par un coefficient qui décroît avec la similarité entre le pixel considéré et le pixel du centre de la fenêtre. Parmi ceux ci, on trouve le filtre- σ , pour lequel le coefficient de pondération du pixel est l'inverse de son gradient, de manière à ce que les points trop différents du point traité ne contribuent que peu, ou le toboggan de Fairfield, qui creuse l'histogramme de l'image en attribuant à chaque pixel le niveau de gris du point de minimum de gradient le plus proche dans un voisinage. Tous les points du chemin reliant ce pixel au pixel de minimum de gradient se voient de plus affecter cette valeur, sont marqués pour ne plus être traités, et le processus est itéré jusqu'à ce que tous les points soient marqués.
- Filtres à fenêtres adaptatives : on recherche autour de chaque pixel la fenêtre la plus adaptée au filtrage, soit en sélectionnant parmi une famille de fenêtres celle qui convient le mieux (e.g. filtrage de Nagao), soit en faisant croître une fenêtre et en contrôlant sa croissance (e.g. filtrage de Wu).

2 Partie pratique

2.1 Travail demandé

Vous trouverez sur <http://ent.univ-bpclermont.fr/> dans la rubrique "Ressources Pédagogiques" ce sujet, ainsi qu'un squelette du TP.

- Ecrire une fonction qui affiche la phase du gradient d'une image, ce dernier étant approché par les opérateurs de Prewitt.
- Ecrire la fonction qui affiche le résultat d'un filtrage par un filtre- σ . Là encore, le gradient sera approché par les opérateurs de Prewitt.

2.2 Quelques fonctions utiles

Vous trouverez dans la [référence](#) de Cimg la description complète des fonctions suivantes :

- `get_convolve` : retourne la convolution de l'image par un masque

```
CImg<typename cimg::largest<T,t>::type> get_convolve( const CImg< t > &mask,
const unsigned int cond = 1,
const bool weighted_convolve = false) const
```

Par ailleurs `CImg` dispose de macros pour les opérations sur les voisinages (cf. doc et TP précédent). En particulier, chaque élément du masque peut être directement accédé : pour un voisinage `CImg_3 x 3 x 1(W, float)`, les éléments w_{ij} s'écrivent $Wpp, Wcp, \dots Wnn$, où l'indice p signifie previous, c current et n

next. Ainsi, par exemple $w_{0,-1} = Wcp$. L'exemple suivant montre comment calculer simplement la norme du gradient par différences finies :

```
// Chargement d'une image
CImg<float> image("im.bmp");

// Voisinage 3x3 I
CImg_3x3(I,float);

// Image du gradient (m^emes dimension que l'image initiale)
CImg<float> gradnorm(image.dimx(),image.dimy());

// Boucle sur les dimensions en utilisant le voisinage I
cimg_for3x3(image,x,y,0,0,I)
{
// d\eriv\ee suivant x
const float ix = 0.5f*(Inc-Ipc);
// d\eriv\ee suivant y
const float iy = 0.5f*(Icn-Icp);
// norme du gradient en (x,y)
gradnorm(x,y) = std::sqrt(ix*ix+iy*iy);
}
// Affichage
gradnorm.display("norme du gradient");
```

Références

- [1] J.P. Cocquerez and S. Philipp, *Analyse d'images : Filtrage et segmentation*, Masson, 1995
- [2] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison-Wesley, 1992
- [3] W. K. Pratt, *Digital Image Processing*, Wiley Interscience, 2001
- [4] The Quantitative Imaging Group, *An interactive image processing course*