Lissage et filtrage linéaire

Un système d'enregistrement d'image ne restitue pas l'image de manière parfaite : des informations parasites apparaissent et viennent s'ajouter de manière aléatoire aux détails de la scène d'origine. Cet effet est désigné sous le terme général de "bruit". Pour un appareil photo numérique, le bruit est principalement dû à la physique des capteurs et aux composant électroniques qui exploitent le signal provenant des capteurs. L'objectif du lissage est de réduire l'influence du bruit, afin d'obtenir une meilleure restitution de l'image pour son analyse.

Lissage et filtrage linéaire

Un filtre F(x) est linéaire lorsqu'il répond à la condition suivante : $F(\alpha x) = \alpha F(x)$. Il est implanté » par une convolution. Cette catégorie de filtres correspond à des filtres passe-bas, éliminant les hautes fréquences qui représentent généralement le bruit. Le problème est de déterminer la fréquence de coupure exacte du filtre car les contours des objets sont eux aussi représentés par des fréquences relativement élevées, qu'il faut conserver. Nous pouvons classer ces filtres dans trois catégories principales : filtres linéaires, filtres non linéaires et filtres adaptatifs. Le rôle d'un filtre de lissage peut être résumé en trois actions :

- Tous les pixels d'une zone relativement homogène doivent être à une même valeur : cela revient à réduire la variance de cette zone homogène.
- Tous les pixels proches d'un contour doivent appartenir à une zone homogène s'ils n'appartiennent pas à ce contour. Cela revient à ne pas créer de nouvelles zones ou valeurs à proximité des points de contours.
- Aucun point de contour ne doit disparaître : le filtre ne doit pas entraîner de perte de résolution sur l'image, en particulier sur les zones de 1 ou 2 pixels d'épaisseurs (Images de caractères ou d'empreintes digitales). De plus, le filtre ne doit pas entraîner de déformations de ces contours (en particulier pour les coins et les intersections de contours) et ne doit pas déplacer ces contours (en particulier lorsque des mesures métriques sont envisagées).

Les filtres linéaires permettent une réduction du bruit dans les images, mais ont l'inconvénient de traiter de manière identique le bruit et les signaux utiles (contour), et donc de dégrader l'information pertinente dans le cas d'une détection de contours. Par contre, les filtres linéaires ont prennent en compte de la hauteur des bruits qu'ils ont à traiter. Généralement, leur action est proportionnelle à la valeur du bruit à éliminer.

FFT et Filtrage fréquentiel

La transformée de Fourier discrète d'une image de taille N × M s'écrit :

$$TF(I(u,v)) = \frac{1}{\sqrt{N.M}} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} I(x,y) e^{-i2\pi \left(\frac{ux}{N} + \frac{yv}{M}\right)}$$

Pour une image à valeurs réelles, les parties réelle et imaginaire de la transformée de Fourier présentent une symétrie en N/2 et M/2, et par conséquent la disposition des fréquences obtenue est la suivante (figure 1 a):

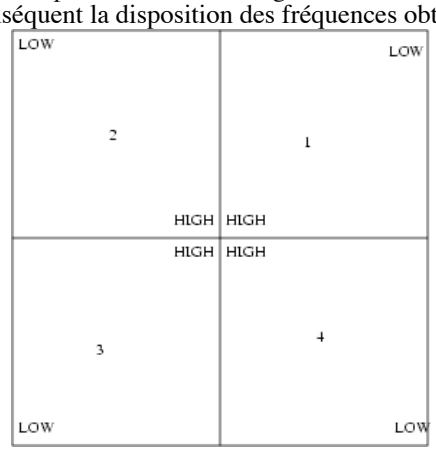


Figure 1 a Figure1 b

En replaçant les basses fréquences au centre de l'image, on obtient une représentation fréquentielle plus conforme, avec l'origine des coordonnées fréquentielles en (N/2, M/2) (figure 1b) (fonction fftshift)

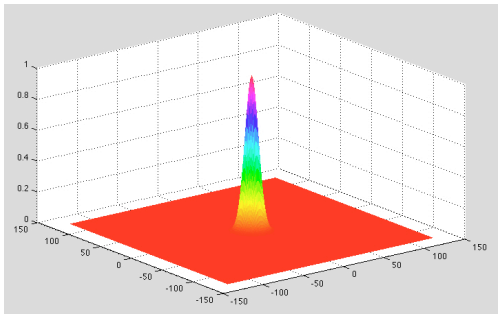
Un des intérêt de la transformée de Fourier pour le filtrage est que celle ci transforme la convolution en produit dans l'espace fréquentiel : $TF(I * H) = TF(I) \cdot TF(H)$.
On peut donc appliquer un filtre linéaire H en multipliant la transformée de Fourier d'une image par la transformée de Fourier du filtre, puis en effectuant la transformée de Fourier inverse du résultat. Pour une image de taille n^2 et un masque de taille m^2 , la multiplication nécessite n^2 opérations, et l'algorithme FFT a une

complexité en $O(n^2 \log n^2)$, ce qui donne une complexité totale en $O(n^2 \log n^2)$. Pour un filtrage spatial, la complexité est en $O(n^2 m)$ si le filtre est séparable et de taille m, et en $O(n^2 m^2)$ si le filtre n'est pas séparable et de taille m. Le choix entre les 2 techniques de lissage se fera donc en comparant m et log(n).

Filtrage gaussien par FFT

La FFT d'une gaussienne est une gaussienne :

$$G(x,y) = \frac{1}{\sqrt{2\pi}\,\sigma} . \; e^{-\frac{x^2+y^2}{2\sigma^2}} \quad et \quad TF(G(u,v)) = e^{-2\pi^2\sigma^2(u^2+v^2)}$$



On peut donc facilement réaliser un filtrage gaussien en définissant directement le filtre dans l'espace de Fourrier (sans faire de FFT). Il suffit ensuite de faire le produit de la TF de l'image avec ce filtre puis de prendre la TF^{-1} pour réaliser un filtrage gaussien. Le paramètre σ est lié à la taille des objets présents dans l'image. Les avantages sont :

- Le filtrage est presque exact, car les coefficients sont rapidement presque nuls (figure ci contre, pour σ=6)
- La taille de la gaussienne n'intervient pas dans le design du filtre
- La complexité et le temps de calcul sont constants pour des σ différents
- Il existe des bibliothèques particulièrement optimisées pour le calcul de la fft (voir FFTW).

Remarques importantes pour la mise en œuvre :

- Le filtre est centré sur le milieu de l'image. Il faut donc penser à « shifter » la TF de l'image avant de faire le produit, puis penser à « shifter »le produit avant de faire la TF^{-1}.
- Les fréquences sont normalisées. Dans le domaine discret, la transformée de Fourrier de la gaussienne s'exprime par $TF(G)(u,v) = e^{-2\pi^2\sigma^2\left(\left(\frac{u-N/2}{N}\right)^2+\left(\frac{v-M/2}{M}\right)^2\right)} \; pour \;\; 0 <= u < N \;\; et \;\; 0 <= v < M$ avec N et M dimensions de l'image

Convolution spatiale

Dans le domaine continu, une convolution entre une image I et un filtre F est définie par :

$$I * F = R(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I(x-x',y-y').F(x',y')dx'dy'$$

Dans le domaine discret, une convolution entre une image I et un filtre F est définie par :

$$R(x,y) = \sum_{i=-N}^{N}\sum_{j=-M}^{M} I(x+i,y+j).F(i,j)$$

Filtrage linéaire gaussien

Les coefficients du filtre sont les valeurs d'une gaussienne à deux dimensions, normalisés de telle sorte que leur somme soit égale à 1. Ce type de filtre est efficace et trouve un écho dans le système visuel humain : c'est en effet le type de filtrage réalisé par les premières cellules du processus visuel humain situées après les cellules photo-réceptrices de l'œil, avec plusieurs tailles de filtres sur des canaux différents. La gaussienne s'écrit:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} . \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{y^2}{2\sigma^2}}$$

Dans le domaine discret, la convolution s'ecrit :

$$R(x,y) = \sum_{i=-N}^{N}\sum_{j=-M}^{M} \frac{1}{2\pi\sigma^2} e^{-\frac{i^2+j^2}{2\sigma^2}} I(x+i,y+j)$$

Pour un filtre isotrope en x,y, les bornes N et M sont égales à la largeur W du filtre. Cette valeur doit permettre de bien approximer ce filtre dans le domaine discret ie les derniers coefficients du filtre sont pratiquement nuls. Pour $W=3\sigma$, la courbe est décrite à 99,5 %. Dans ce cas, il faut $49\sigma^2$multiplications et $49\sigma^2$-1 additions par pixel, ce qui est rapidement trop couteux. On vérifiera l'écart en fonction de W, entre le résultat de la version FFT et la version spatiale pour des valeurs σ de différentes.

Filtrage linéaire séparable

Un filtre séparable est une filtre tel que H(x,y)=H1(x).H2(y).

Dans ce cas, la convolution 2D s'écrit comme une convolution 1D par H1, suivi par une convolution 1D de H2.

$$R(x,y) = \int_{x'=-\infty}^{\infty}\int_{y'=-\infty}^{\infty} I(x-x',y-y')H(x',y')dx'dy' = \int_{x'=-\infty}^{\infty}\left[\int_{y'=-\infty}^{\infty} I(x-x',y-y')H_2(y')dy'\right]H_1(x')dx$$

La fonction gaussienne G(x,y) étant séparable, il convient de transformer la convolution bidimensionnelle en 2 convolutions monodimensionnelles, une selon l'axe horizontal et une selon l'axe vertical. Nous réduisons le nombre d'opérations à 12σ multiplications et 12σ-2 additions par pixel.

$$R(x,y) = \int_{x'=-\infty}^{\infty}\int_{y'=-\infty}^{\infty} I(x-x',y-y')e^{-\frac{x'^2+y'^2}{2\sigma^2}}dx'dy' = \int_{x'=-\infty}^{\infty}\left[\int_{y'=-\infty}^{\infty} I(x-x',y-y')e^{-\frac{y'^2}{2\sigma^2}}dy'\right]e^{-\frac{x'^2}{2\sigma^2}}dx'$$

Les coefficients du filtre doivent être normalisés, leur somme étant égale à 1. Trois étapes sont donc nécessaires pour effectuer ce calcul:

- calcul des coefficients normalisés du masque monodimensionnel.
- filtrage horizontal avec ce masque.
- filtrage vertical avec ce masque, en utilisant les valeurs obtenues par le filtrage horizontal.

Remarque1: Ce type de filtre, surtout dans sa version séparable, impose des calculs en réels. Dans le cas contraire, la troncature effectuée lors du 1er passage provoque des erreurs importantes lors du 2ième passage

Remarque2: Problème des bords : dans le domaine numérique, l'image est de taille finie. La convolution des pixels aux bord de l'image (i<W/2) fait appel à des pixels situés en dehors de l'image. Ces pixels peuvent prendre les valeurs suivantes :

- Zéro
- La valeur du pixels du bords : on prolonge l'image par continuité
- La valeur des pixels du bord opposé de l'image : on périodise l'image, C'est cette version qui sera utilisée ici pour permettre une comparaison avec la lissage par FFT

En appliquant de gros masques de convolution, on peut constater que le temps de calcul devient très important. En effet, pour une image de taille n^2 et un masque de taille m^2 le nombre de calcul à réaliser pour appliquer le masque est de n^2 m^2 et la taille maximum du masque est n^2. Pour y remédier, on peut utiliser la transformée de Fourier de l'image, en utilisant une simple multiplication ponctuelle et le lien entre convolution et Transformée de Fourrier. Cela rend les calculs beaucoup plus rapides car il existe un algorithme de calcul de la transformée très rapide, la Fast Fourier Transform (FFT).

Filtrage non linéaire

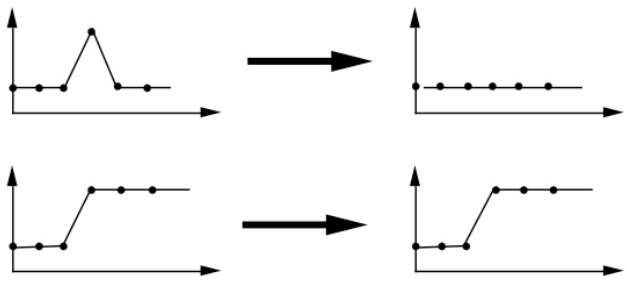
Pour palier le problème soulevé par les filtres linéaires (filtrage identique sur le bruit et sur les contours), on utilise des filtres non linéaires, destinés à supprimer le bruit sans détériorer les contours. Le filtre le plus simple de cette catégorie est le filtre médian.

Filtre médian

Il consiste à remplacer le pixel central par la valeur médiane du voisinage (taille impaire) dans lequel il se trouve. Ce filtre ne tient pas compte de la hauteur du bruit, au contraire des filtres linéaires, puisqu'il opère un classement relatif. Tout point hors de la norme est donc rejeté, quelque soit sa valeur. Il n'introduit donc pas de nouvelles valeurs d'intensité et réduit la variance des intensités. Il permet d'éliminer les valeurs incorrectes isolées. Il permet d'éliminer les oscillations d'intensité de période inférieure à la taille de la fenêtre ainsi que les bruits d'impulsion. La fenêtre utilisée pour le filtrage peut être une fenêtre carrée NxN, une croix NxN, une ligne ou une colonne. Les effets de ce filtre varient significativement en fonction de cette fenêtre. Pour des grandes tailles, le filtre utilisant une fenêtre carrée a tendance à éliminer (en les arrondissant) les angles et les coins des contours, car toute impulsion inférieure à la taille du filtre est éliminée, quelque soit sa direction (un angle est une impulsion dans une direction donnée).

$$\begin{bmatrix} & \cdots\cdots & \\ \cdots & 40\ 50\ 60 & \cdots \\ \cdots & 45\ 82\ 75 & \cdots \\ \cdots & 80\ 90\ 95 & \cdots \\ & \cdots\cdots & \end{bmatrix}$$

Dans ce cas, le pixel central valant 82 est remplacé par 75



Le temps d'exécution de ce filtre est long car il impose un tri sur (2xN+1)x(2xN+1) valeurs, et augmente rapidement avec la taille du voisinage. Une manière élégante de réduire les temps d'exécution d'un médian de grande taille consiste à utiliser l'histogramme cumulé qui s'obtient facilement entre 2 positions voisines de la fenêtre de traitement:

Médian de taille 2N+1

Pour chaque ligne i de l'image (à partir de la ligne N)

- Calculer l'histogramme d'une fenêtre 2N+1 * 2N+1 autour du point d'indices i,N
- La valeur du nouveau point est déterminé par la valeur k telle que :

$$\sum_{i=0}^{k} h(i) \geq 2N^2 + 2N$$

- Pour chaque point j de la ligne i
  - Supprimer de l'histogramme les valeurs des points de la colonne j-N-1
  - Ajouter à l'histogramme les valeurs des points de la colonne j+N
  - La nouvelle valeur k du point est telle que :

$$\sum_{i=0}^{k} h(i) \geq 2N^2 + 2N$$

Cet algorithme présente un temps d'exécution linéaire en fonction de la taille du filtre. Il a la particularité de n'effectuer aucun tri.

Filtres adaptatifs :

Ces filtres sont des opérateurs de lissage qui modifient leur comportement ou leurs coefficients en fonction du contexte local. Ils comportent généralement des seuils qui leur permettent de s'adapter aux valeurs de bruits à éliminer. Leur inconvénient est la connaissance requise à priori du type de bruit et des valeurs du bruit.

Filtre adaptatif récursif (St Marc, Chen, Medioni)

L'idée suivi par ces auteurs est la suivante : le filtrage consiste d'une part à réduire la variance du bruit dans les zones homogènes, d'autre part à conserver les fronts correspondant à bords des objets. Il s'agit donc de filtrer uniquement les zones homogènes pour éviter l'adoucissement des contours. Malheureusement, l'emplacement de ces contours est inconnu, auquel cas notre problème serait résolu. Ils proposent donc de filtrer une image par une fonction dépendant du gradient de l'image, de telle sorte que le lissage soit fort pour les gradients faibles (zones relativement homogènes) et faible pour les gradients forts.

Le filtrage gaussien peut en fait être approximé par une succession de filtres moyenneurs de taille 1, mais passés successivement plusieurs fois. Si $S^0(x)$ est le signal à filtrer :

$$S^{t+1}(x,y) = \frac{\sum_{i=-1}^{1}\sum_{j=-1}^{1} w^t(x+i,y+j).S^t(x+i,y+j)}{\sum_{i=-1}^{1}\sum_{j=-1}^{1} w^t(x+i,y+j)} \quad avec \quad w^t(x,y) = e^{-\left(\frac{G_x^{t\,2}+G_y^{t\,2}}{2k^2}\right)}$$

$$et \quad G_x^{t\,2}(x,y) + G_y^{t\,2}(x,y) = (S^t(x+1,y) - S^t(x-1,y))^2 + (S^t(x,y+1) - S^t(x,y-1))^2$$

Ce filtrage est répété itérativement jusqu'à ce que l'image filtrée soit stationnaire. Il y a deux opérations distinctes réalisées par le filtrage : d'une part, le rehaussement des contours qui doivent rester présents dans l'image finale et d'autre part le lissage des régions homogènes. Le premier effet est obtenu juste après quelques itérations tandis que le second peut être très long (100 à 200 itérations au moins). La valeur du paramètre k correspond à peu près au seuil pour lequel les contours seront conservés et accentués si leur gradient est supérieur à ce seuil k.

Remarque 1 : on peut utiliser une version dégradée en considérant $w^t=w^0$. Le résultat est légèrement moins bon avec cette version, mais beaucoup plus rapide.

Remarque 2: ce filtre est très proche du filtrage par diffusion isotrope non linéaire. En 1D, il s'écrit en

$$S^{t+1}(x) = c^t(x+1).S^t(x+1) + c^t(x).S^t(x) + c^t(x-1).S^t(x-1)$$

$$avec \quad c^t(x+1) + c^t(x) + c^t(x-1) = 1$$

$$S^{t+1}(x) = c^t(x+1).S^t(x+1) + (1 - c^t(x+1) - c^t(x-1)).S^t(x) + c^t(x-1).S^t(x-1)$$

soit

$$S^{t+1}(x) - S^t(x) = c^t(x+1).(S^t(x+1) - S^t(x)) - c^t(x-1).(S^t(x) - S^t(x-1))$$

$$qui \ est \ l'équation \ de \ la \ chaleur \ \frac{\partial S}{\partial t} = \nabla(c\nabla I)$$

Si c est une constante, la solution de cette équation est le lissage gaussien, et le temps t est lié à la largeur du lissage par $\sigma^2=2t$. Un nombre d'itérations infini conduit à une image lissée constante.

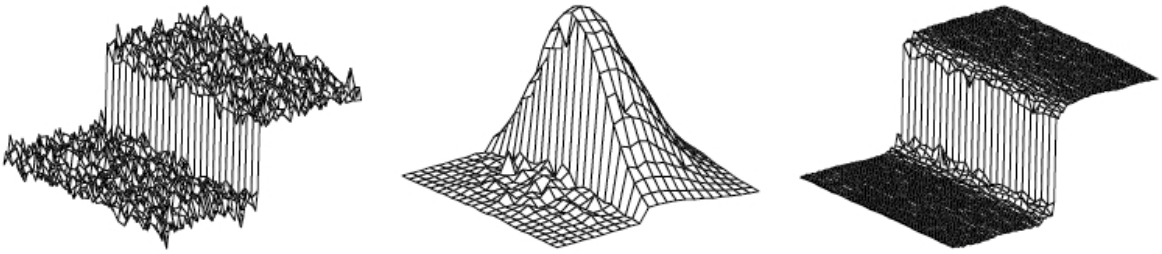
Dans notre cas, c est un scalaire qui dépend des caractéristiques locales de l'image et le lissage converge vers une image constante par morceaux.

Filtre bilatéral

Les coefficients de ce filtre dépendent non seulement de la distance euclidienne au point à lisser (comme pour la gaussienne), mais aussi de la ressemblance des intensités entre le pixel central et le pixel du masque.

$$S(x,y) = \frac{\sum_{i=-3\sigma_1}^{3\sigma_1}\sum_{j=-3\sigma_1}^{3\sigma_1} e^{-\frac{i^2+j^2}{2\sigma_1^2}} e^{-\frac{(I(x+i,y+j)-I(x,y))^2}{2\sigma_2^2}} I(x+i,y+j)}{\sum_{i=-3\sigma_1}^{3\sigma_1}\sum_{j=-3\sigma_1}^{3\sigma_1} e^{-\frac{i^2+j^2}{2\sigma_1^2}} e^{-\frac{(I(x+i,y+j)-I(x,y))^2}{2\sigma_2^2}}}$$

De ce fait, les pixels se ressemblant provoqueront un fort lissage (ie des zones homogènes) alors que des pixels ne se ressemblant pas provoqueront de faibles lissages. L'exemple ci dessous montre l'image à gauche, les coefficients du masque situé à 2 pixels sur la droite de la frontière, et l'image lissée. Ce filtre est lui



aussi itéré plusieurs fois afin d'obtenir un effet maximum. Si la valeur de σ_2 est liée au bruit de l'image, celle de σ_1 est comme pour la gaussienne classique liée à la taille des détails que l'on souhaite préserver.

Travail à réaliser

L'objectif est de comparer les différents filtrages sur quelques images. Les différentes images formes1bbxx.pgm sont des images synthétiques qui ont été lissées (gaussienne σ=2) puis auxquelles on a ajouté un bruit blanc. L'image formes1sp.pgm simule un bruit multiplicatif type speckle. Les images formes1pets1.pgm , formes1pets5.pgm, formes1pets10.pgm contiennent un bruit de type poivre et sel. L'image initiale étant connue, on peut calculer le PSNR entre l'image lissée et l'image parfaite pour quantifier la qualité du lissage.

Exercice 1 : FFT et gaussienne.

Appliquer le filtre gaussien en passant par la FFT. Le code de la FFT et un exemple d'utilisation sont disponibles sur le kiosk. Attention lors de la multiplication des transformées de Fourier à bien considérer les mêmes coordonnées fréquentielles : il est recommandé de replacer les basses fréquences au centre en utilisant fftshift.
Comparer sur les images bruitées disponibles sur le kiosk, qualitativement et quantitativement (PSNR) les résultats obtenus pour les différents types de bruit (gaussien, speeckle, poivre et sel) en fonction de la taille σ.

Exercice 2 : Convolution par une gaussienne

Implémenter la convolution gaussienne d'une image par un masque séparable. La taille du masque et la valeur du lissage doivent être paramétrable. Comparer les valeurs obtenues entre la convolution par FFT et celle obtenue par un masque en fonction de la taille du masque (pour des valeurs fixées de σ : 3 puis 7 par exemple). Pour quelle taille de masque retrouve t on des résultats identiques par FFT et par un masque ?
En utilisant cette taille de masque, comparer les temps de calcul des deux implémentations pour des gaussiennes de variance croissantes. Retrouve t on le résultat théorique attendu ?

Exercice 3 : Lissage Médian

Programmer le médian et utiliser différentes tailles pour observer l'évolution du lissage pour les différents types de bruit (gaussien, speeckle, poivre et sel). Tracer le PSNR en fonction de la taille.
Quelles remarques qualitatives pouvez vous faire avec les images tangram.pgm, globules.pgm et lacornou.pgm

Exercice 4 : Filtre adaptatif

Programmer le filtre et mêmes questions qu'avec le médian en fonction de k

Exercice 5 : Filtre bilatéral

Programmer le filtre et mêmes questions qu'avec le médian en fonction de σ_1.

Compte rendu: Comparaison

Que peut on dire qualitativement et quantitativement des avantages et inconvénients de ces méthodes ?

Mesurer le temps

```
#include <time.h>
main() { clock_t debut, fin ;
  debut=clock();
  fonction_dont_on_veut_mesurer_le_temps();
  fin=clock();
```

```
  printf("durée=%f\n",((double)fin-debut)/CLOCKS_PER_SEC);
}
```