TP4 - Le filtrage de diffusion

Vincent Barra - Christophe Tilmant

2011-2012

Dans les deux séances de travail précédentes, nous nous sommes intéressés à des filtres qui agissent de façon *homogène* sur l'image, c'est-à-dire que le traitement s'applique de façon identique à l'ensemble des pixels. Si l'on prend le cas du filtre Gaussien il est largement utilisé pour réduire le bruit, cependant il comporte des inconvénients : il altère les contours et les délocalise. Ce filtre peut également être vu comme un processus de diffusion, où le niveau de gris de l'image diffuse de façon homogène et *isotrope*. Le qualificatif isotrope précise que la diffusion se réalise dans toutes les directions spatiales sans préférences.
L'idée du **filtre de diffusion** consiste à appliquer le processus physique de diffusion à une image. Il existe différentes approches qui diffèrent par leurs caractères homogène/inhomogène, linéaire/non linéaire et isotrope/anisotrope. Ce filtrage est un procédé visant à éliminer le bruit d'une image, tout en préservant les informations importantes, en particulier les contours.

1 Fondement physique de la diffusion

Cette méthode est fondée sur les principes physiques de la diffusion entre fluides : l'équation de diffusion est similaire à celle des concentrations locales d'un fluide qui s'équilibrent sous la condition de conservation de la matière. Le transfert pour atteindre l'équilibre des concentrations s'exprime avec la première loi empirique de Fick :

$$\mathbf{j} = -\mathbf{D}.\nabla u \tag{1}$$

avec,

- $\mathbf{D}$: tenseur de diffusion, symétrique défini positif,
- $u((x, y, z), t)$: concentration de matière au point (x, y, z) à l'instant t, $u : R^3 \times [0; +\infty[\to R$,
- ∇u : gradient spatial de la concentration de matière,
- $\mathbf{j}$: flux de matière.

Si ∇u et $\mathbf{j}$ sont parallèles, c'est une diffusion isotrope, et D peut être remplacé par un scalaire g nommé diffusivité. Le cas général est appelé anisotrope. La propriété de transport de matiére sous la condition de conservation s'exprime avec l'équation de continuité :

$$\partial_t u = -div\mathbf{j} \tag{2}$$

où t représente le temps et $\partial_t u$ la dérivée temporelle de la concentration de matière. En combinant ces deux équations, on obtient l'équation de diffusion (seconde loi empirique de Fick) :

$$\partial_t u = div\,(\mathbf{D}.\nabla u) \tag{3}$$

Dans le cas où le tenseur de diffusion D est constant on parle de diffusion homogène ; et dans le cas où il dépend de la structure différentielle de u, la diffusion est inhomogène.
Cette équation appara̍ît dans beaucoup de domaines. Dans le domaine du transfert de chaleur, c'est l'équation de la chaleur. En traitement d'images, on peut assimiler la concentration u au niveau de gris et les conditions initiales à l'image de départ. Dans ce cas le tenseur de diffusion est le degré de liberté de la méthode et n'est pas obligatoirement constant. Fréquemment il y a un avantage à choisir ce tenseur comme une fonction des caractéristiques locales de l'image. Trois cas sont intéressants en traitement d'images (cf. figure 1) :

- le *filtre de diffusion isotropique linèaire* utilisant donc une diffusivité constante ;
- le *filtre de diffusion isotropique non-linèaire* (ou simplement filtre de diffusion non lineaire) utilisant une diffusivité s'adaptant aux caractéristiques locales de l'image ;
- le *filtre de diffusion anisotropique non-linèaire* utilisant un tenseur de diffusion s'adaptant aux caractéristiques locales de l'image.

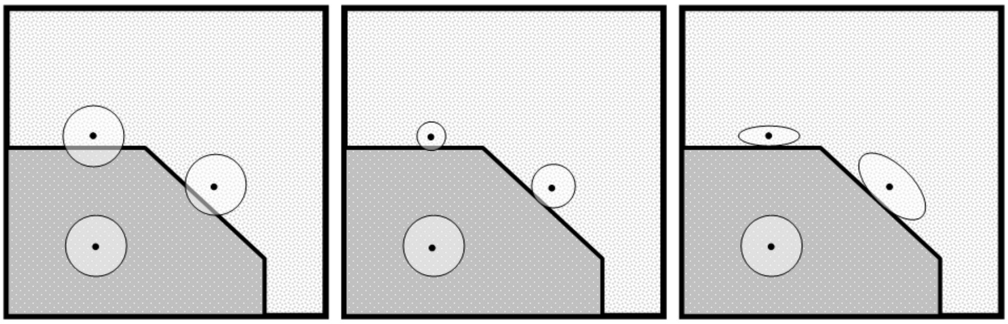

Fig. 1 – Illustration des différents filtres de diffusion. Dans le cas linéaire homogène on a une diffusion constante et identique en tous points de l'image. Dans le cas isotrope non-linéaire, la "taille" de diffusion s'adapte à l'information locale (norme du gradient spatial par exemple). Enfin, dans le cas anisotrope la diffusion se fait parallélement aux contours, en exploitant par exemple l'orientation du gradient spatial.

2 Filtre de diffusion linéaire

Dans le cas de la diffusion linéaire homogène, l'équation de diffusion devient :

$$\partial_t u = \Delta u$$

et a pour solution :

$$u\left((i,j),t\right) = \left(K_{\sqrt{2t}} * u_0\right)(i,j)\,,\ \text{pour t} > 0$$

avec K_σ, une Gaussienne d'écart type σ . On obtient donc un lissage Gaussien de l'image. Historiquement, c'est le premier filtrage multiéchelle qui ait été étudié. Cependant, si ce filtrage réduit le bruit, il rend floue l'image, atténue les contours et on perd la localisation précise des éléments de l'image. Sur la figure 2 est illustrée l'application d'un filtre Gaussien avec des temps de diffusion croissants (c'est-à-dire des valeurs d'écart-type croissantes).

Fig. 2 – Filtrage de diffusion linéaire

Pour éviter cette altération des contours, l'idée est d'adapter la diffusivité à une "mesure" de contours (par exemple le gradient spatial). Dans ce cas on rentre dans le cadre du filtrage de diffusion non-linéaire.

3 Filtre de diffusion non-linéaire

L'introduction des équations de diffusion non-linéaires dans le domaine du traitement des images remonte à un article de Malik et Perona de 1990 [4]. La faiblesse de l'équation de la chaleur est que la diffusion est identique en tout point de l'image (cette équation modélise initialement la diffusion de la chaleur dans un milieu isotrope). En particulier, comme nous l'avons vu au paragraphe précédent, l'image est lissée aussi bien dans les zones homogènes que le long des contours. L'idée de Malik et Perona est de lisser l'image dans les zones homogènes, et de ne pas faire évoluer l'image le long des contours, voire de rehausser ces derniers, comme nous allons le voir plus précisément. L'équation correspondante s'écrit :

$$\partial_t u = div\left(g\left(|\nabla u|\right)\nabla u\right)$$

avec g une fonction décroissante, valant 1 en zéro, et tendant vers 0 en l'infini. L'équation se rapproche donc de l'équation de la chaleur aux points où $|\nabla u|$ est proche de 0. A titre d'exemple, nous considérerons $g\left(s\right) = \frac{1}{1+(\lambda s)^2}$, un des choix de fonction proposé par l'article original de Malik et Perona. Cette métrique impose une diffusion plus faible pour des valeurs de gradients (en norme) d'autant plus élevées. Donc les contours sont préservés. Le paramètre λ permet de régler une diffusion plus ou moins importante par rapport à la valeur de la norme du gradient. En effet un paramétre λ plus important préservera des contours avec des gradients d'autant plus faibles en norme. Sur la figure 3 on peut observer le résultat d'un tel filtrage pour des temps de diffusion croissants.

Fig. 3 – Filtrage de diffusion non-linéaire par la méthode de Perona-Malik

L'expression de diffusivité présentait précédemment n'est pas la seule. Une seconde expression fut, par exemple, introduite par Perona et Malik :

$$g\left(s\right) = e^{-(\lambda s)^2}$$

4 Partie pratique

4.1 Travail demandé

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

– Il est demandé de mettre en oeuvre le modéle de Perona et Malik. Le schéma de discrétisation explicite en tous points (i, j) et à l'instant $t + \Delta t$ est le suivant :

$$u\left((i,j), t+\Delta t\right) = u_{i,j}^{t+\Delta t} = u_{i,j}^t + \Delta t\left(c_{E_{i,j}}^t \nabla_E u_{i,j}^t + c_{W_{i,j}}^t \nabla_W u_{i,j}^t + c_{N_{i,j}}^t \nabla_N u_{i,j}^t + c_{S_{i,j}}^t \nabla_S u_{i,j}^t\right)$$

où, N,S,E et W représentent les directions spatiales nord, sud, est et ouest, le symbole ∇ dénote le gradient spatial dans la direction indiquée par l'indice, enfin les coefficients c sont définis par :

$$\begin{array}{ll} c_{N_{i,j}}^t = g\left(\left|\nabla_N u_{i,j}^t\right|\right) & c_{S_{i,j}}^t = g\left(\left|\nabla_S u_{i,j}^t\right|\right) \\ c_{E_{i,j}}^t = g\left(\left|\nabla_E u_{i,j}^t\right|\right) & c_{W_{i,j}}^t = g\left(\left|\nabla_W u_{i,j}^t\right|\right) \end{array}$$

- Vous testerez votre algorithme sur des images que vous bruiterez. Comparez les résultats avec les filtres étudiés durant les séances précédentes. Comparez également les deux expressions de diffusivité proposées par Perona et Malik.

4.2 Quelques fonctions utiles

Vous trouverez dans la référence de Cimg la description complète des fonctions suivantes :

- get_gradient : qui permet de calculer des dérivées par différences finies
  `CImgList< Tfloat >  get_gradient (const char *const axes=0, const int scheme=3) const`
- get_noise : qui permet de bruiter une image
  `CImg  get_noise (const double sigma=-20, const unsigned int ntype=0) const`

5 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] P. Perona and J. Malik., Scale-space and edge detection using ansotropic diffusion, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 12(7), pp. 629-639, July 1990.