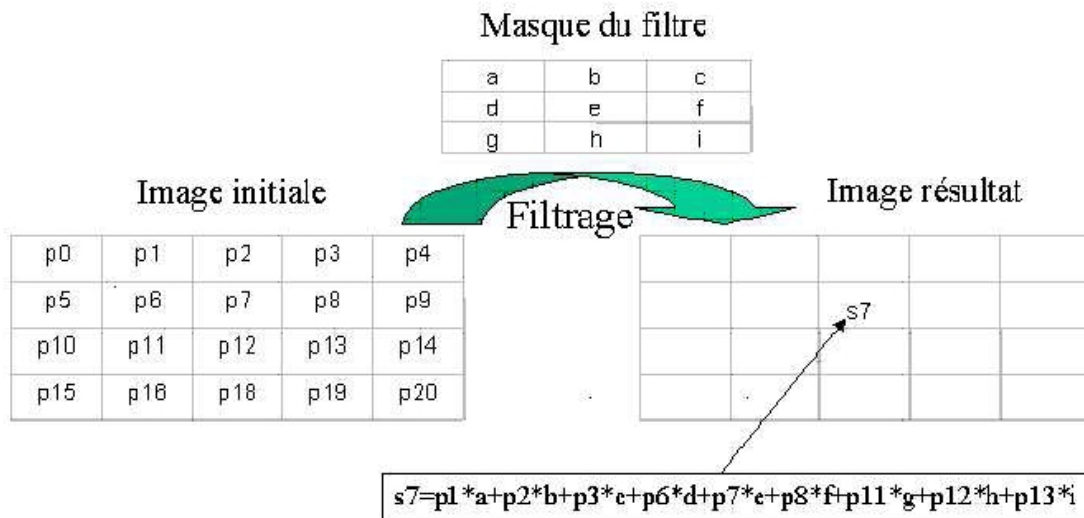


# Traitement de l'image : Filtrage

## Introduction

Le principal objectif du filtrage est de pré-traiter une image afin que l'image filtrée soit plus adaptée que l'image originale pour une application spécifique. Nous allons étudier dans ce cadre les méthodes s'appliquant dans le domaine spatial (tout en illustrant, lorsque c'est possible, le lien avec le domaine fréquentiel). Les méthodes spatiales sont des procédures qui opèrent directement sur les pixels de l'image. D'une manière générale ces techniques mettent en oeuvre une convolution discrète 2D par un masque  $h$  de l'image de départ  $I$ , comme indiqué sur la figure ci-dessous.



L'opération qui, dans le domaine temporel, se traduit par un produit de convolution, s'exprime aussi par un simple produit entre le gain fréquentiel du filtre  $h$  et la transformée de Fourier de l'image  $e$  :

$$s(i, j) = e(i, j) * h(i, j) = \sum_{n=-\infty}^{\infty} \sum_{p=-\infty}^{\infty} e(n, p)h(i - n, j - p) \tag{1}$$

$$= \mathcal{F}^{-1} \left( \widehat{e(i, j)} \widehat{h(i, j)} \right) \tag{2}$$

C'est pourquoi l'étude du gain fréquentiel du masque utilisé (correspondant à la transformée de Fourier de  $h$ ) permet souvent de caractériser la nature du filtre (filtre passe-bas, filtre passe-haut ). Nous allons maintenant étudier différents filtres.

## 1 Filtre « lisseur »

### 1.1 Moyenne locale

On se place dans le cas où un bruit additif se superpose à l'image. Dans ce cas, une méthode simple pour « lisser » l'image consiste à calculer une moyenne locale en  $(n, m)$  : chaque pixel est remplacé par la moyenne pondérée de ses voisins. Un masque typique correspondant à cette opération se définit par :

$$h = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Etudions tout d'abord l'influence de tels filtres sur des images non bruitées.

- Charger l'image *bois.jpg* . Visualiser son spectre (utiliser help pour trouver la fonction nécessaire).
- Créer deux masques h1 et h2 effectuant une moyenne locale sur un voisinage  $3 \times 3$  et  $9 \times 9$  (utiliser la commande ones).
- Appliquer ces filtres sur l'image bois en tapant  

$$\text{res}=\text{corr2d}(\text{bois},\text{h});$$
- Visualiser les résultats et comparer à l'image originale (vous pourrez pour cela afficher une « vue en coupe » de l'image en tapant par exemple :  

$$\text{plot}(\text{bois}(70, :)); \text{plot}(\text{res}(70, :));$$

Nous appliquons maintenant ces 2 filtres sur une image bruitée par un bruit gaussien que vous allez fabriquer.

- Créer l'image *boisbruit* avec la commande `imnoise`, paramètres : moyenne nulle et variance à 0,01. Visualiser l'image et son spectre.
- Appliquer les 2 filtres sur l'image *boisbruit*.
- Visualiser les résultats.

## 2 Filtre non linéaire

Le principal inconvénient des filtres linéaires (basés sur le produit de convolution) est que la réduction de bruit s'accompagne d'un étalement des transitions entre régions. Cette difficulté peut être surmontée par l'utilisation des filtres non linéaires, comme par exemple le filtre médian. Comme son nom l'indique, le filtre médian sélectionne le pixel de la fenêtre d'analyse ayant la valeur médiane. Ce filtre va être insensible à la présence de valeurs aberrantes si elles sont en nombre raisonnable, ce qui est le cas d'un bruit impulsif (de densité de probabilité  $f(a) = Ce^{-\alpha|a|}$  avec  $\alpha$  petit).

- Créer l'image *boisbruit2*. Cette image correspond à l'image *bois* corrompue par un bruit impulsif (`speckle`) avec  $\alpha = 0.001$ .
- Visualiser l'image.
- Appliquer un filtre médian en tapant

$$\text{res}=\text{fmed}(\text{boisbruit2});$$

- Appliquer le filtre  $h = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$  sur l'image *boisbruit 2*.
- Visualiser et comparer les 2 résultats.

- Commenter les principales différences de ces filtres face au bruit impulsif.

Notons que pour des bruits à distributions assez concentrées (Gaussien) les performances du filtre médian sont assez faibles.

- Recharger l'image *boisbruit* (bruit gaussien). Appliquer le filtre médian.
- Commenter le résultat.

## 3 Renforcement de contours

Dans une image, les variations d'intensité représentent des changements de propriétés physiques ou géométriques de la scène ou de l'objet observé. Dans un certain nombre de cas, ces variations d'intensité sont des informations importantes pour les opérations situées en aval de la segmentation. Les approches basées sur le filtrage sont les plus immédiates pour renforcer ou détecter les variations du signal.

### 3.1 Filtrage linéaire

#### 3.1.1 Généralités

Si l'on considère que la transition d'un signal est repérée par son point d'inflexion, sa localisation peut se faire par recherche du maximum local de la valeur absolue de la dérivée première ou par recherche du passage par 0 de la dérivée seconde. On doit donc pour cela calculer la dérivée première ou seconde de l'image. Dans le cas discret, les dérivées directionnelles suivant une direction sont approchées par de simples différences

finies :

$$\frac{\partial I}{\partial n}(n_0, m_0) \approx I_1(n_0, m_0) = I(n_0 + 1, m_0) - I(n_0 - 1, m_0) \quad (3)$$

$$\frac{\partial I}{\partial m}(n_0, m_0) \approx I_2(n_0, m_0) = I(n_0, m_0 + 1) - I(n_0, m_0 - 1) \quad (4)$$

$$(5)$$

La norme du gradient est alors donnée par :

$$|\nabla(n_0, m_0)| = \sqrt{(I_1(n_0, m_0))^2 + (I_2(n_0, m_0))^2}$$

### 3.1.2 Opérateurs de Prewitt et Sobel

Pour ces opérateurs, les dérivées directionnelles horizontale et verticale se calculent à partir des masques :

$$h_1 = \begin{pmatrix} 1 & 0 & -1 \\ c & 0 & -c \\ 1 & 0 & -1 \end{pmatrix} \text{ et } h_2 = \begin{pmatrix} -1 & -c & -1 \\ 0 & 0 & 0 \\ 1 & c & 1 \end{pmatrix}$$

(pour Prewitt  $c = 1$ , pour Sobel  $c = 2$ )

- **Montrer que l'application de l'un de ces filtres permet d'approximer la dérivée première selon une certaine direction.**

Etudions par exemple les opérateurs Sobel ( $c = 2$ ).

- **Définir  $h_1$  et  $h_2$  en tapant par exemple**

$$h_1 = [1 \ 0 \ -1; \ 2 \ 0 \ -2; \ 1 \ 0 \ 1]$$

- **Appliquer ces filtres sur l'image *bois*. Visualiser les résultats.**
- **Appliquer ces filtres sur l'image *boisbruit*. Visualiser les résultats.**
- **A partir de cette étude, commenter la nature de ces filtrages et les principales caractéristiques (par exemple face au bruit).**

Une fois le calcul du gradient effectué, on cherche en général à détecter les principales discontinuités du signal. Pour cela,

1. **On calcule la norme du gradient en tapant (si *res1* et *res2* sont les résultats des 2 filtrages précédents).**

$$D = \text{round}(\text{sqrt}(\text{res1}.^2 + \text{res2}.^2));$$

2. **Puis on seuille l'image de la norme (pour fixer le seuil on examine l'histogramme de la norme du gradient avec la fonction `histplot(n_classes, image)`)**

- **Charger l'image *objets.jpg*.**
- **Appliquer la procédure proposée ci-dessus en explicitant vos choix afin d'extraire les contours les plus importants de l'image.**
- **Commenter le résultat.**

Toutefois, nous remarquons que ces masques n'approximent la dérivée que selon une direction horizontale ou verticale. On peut étudier le gradient selon des directions diagonales en utilisant, en plus des précédents, les

$$\text{masques : } h_3 = \begin{pmatrix} c & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -c \end{pmatrix} \text{ et } h_4 = \begin{pmatrix} 0 & 1 & c \\ -1 & 0 & 1 \\ -c & -1 & 0 \end{pmatrix}$$

- **Définir  $h_3$  et  $h_4$  (avec  $c = 1$ ).**
- **Etudier les gains fréquentiels des filtres.**
- **Appliquer ces filtres sur l'image *bois*. Commenter.**

## 3.2 Filtrage non-linéaire

### 3.2.1 Filtre de Kramer et Bruckner

Le filtre de Kramer et Bruckner  $F$  [1] est défini en 1975 par :

On note  $J = F(I)$ . Soit  $\min_p = \min(I(q) | q \in W(p))$ ,  $\max_p = \max(I(q) | q \in W(p))$ ; alors  $J(p)$  est donné par

$$J(p) = \begin{cases} \max_p & \text{si } I(p) \geq (\max_p + \min_p)/2, \\ \min_p & \text{si } I(p) < (\max_p + \min_p)/2. \end{cases}$$

- Programmer le filtre de Kramer et Bruckner
- Le tester sur les images bois, et ses versions bruitées
- Réhaussement de contraste :
  - appliquer le filtre lisseur l'image bois
  - appliquer le filtre de Kramer et Bruckner à l'image lissée
  - tester plusieurs valeurs des paramètres.

### 3.2.2 Filtre de Pérona-Malik

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 [2]. 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 :  $\frac{\partial u}{\partial t} = \text{div}(g(|\nabla u|)\nabla u)$  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  $g(s) = \frac{1}{1+(\lambda s)^2}$ , 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  $\lambda$  permet de régler une diffusion plus ou moins importante par rapport à la valeur de la norme du gradient. En effet un paramètre  $\lambda$  plus important préservera des contours avec des gradients d'autant plus faibles en norme.

### 3.3 travail demandé

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

$$u((i, j), t + \Delta t) = u_{i,j}^{t+\Delta t} = u_{i,j}^t + \Delta t \left( c_{E_{i,j}}^t \nabla E u_{i,j}^t + c_{W_{i,j}}^t \nabla W u_{i,j}^t + c_{N_{i,j}}^t \nabla N u_{i,j}^t + c_{S_{i,j}}^t \nabla S u_{i,j}^t \right)$$

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

$$c_{N_{i,j}}^t = g(|\nabla N u_{i,j}^t|), c_{S_{i,j}}^t = g(|\nabla S u_{i,j}^t|), c_{E_{i,j}}^t = g(|\nabla E u_{i,j}^t|), c_{W_{i,j}}^t = g(|\nabla W u_{i,j}^t|),$$

Valeurs par défaut des paramètres :

- nombre d'itérations = 15 ;
- $\Delta t = 1/7$ ;
- $\lambda = 30$ ;

## Références

- [1] H.P. Kramer et J.B. Bruckner, *Iterations of a non-linear transformation for enhancement of digital images*, Pattern Recognition, 1975, **7**, pp 53–58,
- [2] P. Perona et J. Malik, *Scale-space and edge detection using anisotropic diffusion*, IEEE Trans. Pat. Anal. and Machine Intel., 1990, **12**, pp. 629–639,