

Chapitre 3

Débruitage par filtrage linéaire

L'étude d'un signal 1D ou 2D nécessite de supprimer au maximum le bruit parasite dû aux conditions d'acquisition. L'un des buts du filtrage est de « nettoyer » le signal en éliminant le plus de bruit possible tout en préservant le maximum d'informations. En outre, l'information contenue dans un signal n'est pas forcément entièrement pertinente : il faut « sélectionner » l'information utile suivant l'usage que l'on veut en faire. Par exemple, à l'écoute d'un morceau de musique, on peut vouloir un renforcement des sons graves. Une autre finalité du filtrage est donc de sélectionner et renforcer certaines bandes de fréquences porteuses de l'information intéressante.

Le filtrage des images a la même finalité que celui des signaux 1D. Il s'agit essentiellement d'enlever le bruit (parasite) ou de sélectionner certaines fréquences. Si la notion de haute fréquence ou basse fréquence est naturelle en signal 1D (son aigu ou grave), la **fréquence spatiale** est un concept plus délicat qui découle du fait que les images appartiennent au domaine spatial. La fréquence temporelle est une grandeur qui caractérise le nombre de phénomènes qui se déroulent au cours d'un temps donné. Si en voiture, le long d'une route, on voit 2 bandes blanches PAR seconde : c'est une fréquence temporelle. Il est ensuite facile de comprendre que ce concept de fréquence « temporelle » peut aussi se traduire en fréquence spatiale en disant qu'il y a 200 bandes blanches PAR kilomètre.

Dans une image, les détails se répètent fréquemment sur un petit nombre de pixels, on dit qu'ils ont une fréquence élevée : c'est le cas pour les textures fines (comme les feuilles d'un arbre ou de l'herbe) et certains contours de l'image. Au contraire, les fréquences basses correspondent à de faibles variations diluées sur de grandes parties de l'image, par exemple des variations de fond de ciel.



Fig. 3.1 Hautes et basses fréquences d'une image

Nous verrons dans la suite que la plupart des filtres agissent sélectivement sur ces fréquences pour les sélectionner, en vue de les amplifier ou de les réduire tout comme dans le cas unidimensionnel.

3.1 Le « bruit »

Dans toute image numérique, les valeurs de gris ou de couleur observées présentent une incertitude. Cette incertitude est due aux aléas du comptage des photons arrivant sur chaque capteur. Les valeurs de couleur mesurées sont perturbées car les capteurs reçoivent des photons parasites et subissent des fluctuations électrostatiques lors de leurs charges et décharges. Quand un capteur reçoit beaucoup de photons venant d'une scène bien éclairée, les parasites sont négligeables par rapport au flux de vrais photons. Mais, même dans une photo d'exposition suffisante, les pixels sombres reçoivent très peu de photons et sont donc « bruités ». Visuellement, on distingue en général deux types de bruit d'image qui s'accumulent :

- le bruit de *chrominance*, qui est la composante colorée des pixels bruités : il est visible sous la forme de taches de couleurs aléatoires,
- le bruit de *luminance*, qui est la composante lumineuse des pixels bruités : il est visible sous la forme de taches plus foncées ou plus claires donnant un aspect granuleux à l'image.

3.1.1 *Bruit thermique*

Le bruit *thermique* est dû à l'agitation naturelle des électrons, qui augmente avec la température du capteur. Ce phénomène est appelé courant d'obscurité. Les fabricants de caméras le quantifient par le nombre d'électrons. Sur des appareils fixes de laboratoire ou de studios professionnels, ce bruit peut être réduit efficacement par refroidissement du capteur. Sur les appareils photo grand public, les fabricants réduisent l'impact de ce bruit en ajoutant un filtre infrarouge juste devant le capteur, ce qui limite l'agitation électronique.

3.1.2 *Bruit « Poivre et sel »*

Le bruit « Poivre et sel » ou bruit *impulsionnel* est un bruit qui assigne à une certain nombre de pixels de l'image une valeur 0 ou 255 aléatoirement. Ce bruit est dû soit à des erreurs de transmission de données, soit à la défaillance d'éléments de capteur, soit à la présence de particules fines sur le capteur d'images.

3.1.3 *Bruit de grenaille*

Le bruit de *grenaille* ou bruit *quantique* est un bruit électronique. Il se produit lorsque le nombre fini de particules transportant l'énergie (électrons dans un circuit électronique, ou photons dans un dispositif optique) est suffisamment faible pour donner lieu à des fluctuations statistiques perceptibles. Le bruit des photons est la principale source de bruit dans les images prises par les appareils photo numériques actuels.

3.1.4 *Le flou*

En photographie le flou de *bougé* peut avoir plusieurs origines :

- le déplacement du photographe s'il est en mouvement ;
- l'instabilité du photographe sur ses appuis s'il est par ailleurs immobile dans l'espace ;
- les tremblements naturels de la main qui soutient l'appareil si le photographe est en revanche complètement stable sur ses appuis ;
- le mouvement de l'appareil du fait de la pression sur le déclencheur s'il n'est pas tenu à la main mais installé sur un support fixe tel qu'un trépied ;

- en photographie numérique, les vibrations engendrées par le déplacement du miroir si le photographe utilise une télécommande pour éviter le mouvement lié au déclenchement.

Le terme flou cinétique (ou *flou de mouvement*) désigne le flou visible sur une photographie ou dans une animation, dû au mouvement rapide du sujet photographié pendant l'enregistrement ou à un long temps de pose. Il se distingue du flou de bougé, qui est dû à l'instabilité de l'appareil-photo pendant la prise de vue.

Enfin le flou est souvent dû à une mauvaise focalisation. Dans le cas de l'œil (myopie) on parle plutôt d'accommodation mais le phénomène est similaire pour un appareil d'acquisition. Lorsque notre regard se porte sur un objet lointain, nous le voyons net. Simultanément, un objet se trouvant plus près de nous dans l'axe de notre regard nous paraît flou. Si au contraire nous regardons cet objet rapproché, notre œil accommode sur celui-ci (nous le voyons net) et l'objet lointain devient flou.

3.1.5 Modélisation du bruit et du flou

En conclusion, les images obtenues par un procédé d'acquisition artificiel dont souvent dégradées par des perturbations appelées génériquement « bruit ». Ce bruit est dans la majorité des cas dû aux conditions d'acquisition :

- étalonnage de l'appareil : malgré des technologies de plus en plus pointues, il existe toujours des erreurs minimales de mesure ou de manipulation,
- perturbations par le milieu ambiant : c'est le cas des images médicales par exemple (radiographie, échographie) où l'information est bruitée par la traversée des tissus observés,
- mauvaises conditions d'acquisition : nuages pour les photos aériennes, manque de luminosité etc ...

Les dégradations peuvent être aussi le fait du vieillissement des supports (photographies ou films anciens) ou le résultat d'artéfacts numériques lors de reconstructions à partir des données physiques (comme en tomographie par exemple). Une notion utile pour estimer le bruit dans une image (et plus généralement dans un signal) et le rapport signal sur bruit (SNR : *Signal to Noise Ratio*). Ce rapport est souvent obtenu (expérimentalement) grâce à l'étalonnage des instruments de mesure.

Définition 3.1.1 On appelle « rapport signal sur bruit » (*Signal to Noise Ratio* : *SNR*), d'un signal donné x entaché d'un bruit b sur un intervalle de temps fini I la quantité :

$$SNR = 20 \cdot \log_{10} \left(\frac{\|x\|}{\|b\|} \right) ,$$

où $\|\cdot\|$ désigne la norme euclidienne du signal discret. Il est exprimé en décibels (dB) qui est donc une échelle logarithmique.

On rappelle que la norme euclidienne d'un signal discret $x = (x_i)_{1 \leq i \leq N} \in \mathbb{R}^N$ est définie par

$$\|x\| = \sqrt{\sum_{i=1}^N x_i^2}.$$

Cette notion est reprise dans le cas des images dans le chapitre 4, section 4.3.3. On peut implémenter le SNR de plusieurs façons avec MATLAB[®] en utilisant par exemple la formule $\text{SNR} = 20 \cdot \log_{10}(\text{norm}(u(:))/\text{norm}(u(:)-v(:)))$ où u est l'image observée et v l'image restaurée (la différence $u - v$ représentant alors le bruit).

Le bruit est de nature et d'origine variées et donc se modélise de différentes façons :

1. Bruit additif gaussien (centré) d'écart-type σ souvent égal au SNR. Nous utiliserons cette dénomination (standard) qui est un peu abusive toutefois. En effet, si u représente l'image originale, l'image dégradée u_b s'obtient par

$$u_b = u + b_\sigma,$$

où b_σ est la réalisation d'une variable aléatoire B_σ de loi normale centrée, d'écart-type σ . C'est le modèle le plus courant.

2. Bruit multiplicatif. L'image dégradée u_b s'obtient par

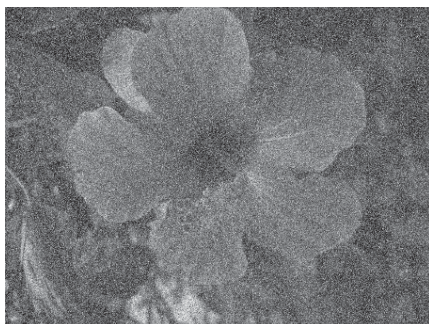
$$u_b = u \cdot (1 + b_\sigma).$$

Le bruit est donc proportionnel à l'intensité lumineuse. Le chatoiement (« speckle » en anglais) est l'ensemble de petites taches rapidement fluctuantes qui apparaissent dans la texture instantanée d'une image et qui lui donnent un aspect granuleux. Elles sont dues soit à la diffusion des ondes d'un faisceau cohérent spatialement de lumière, un laser par exemple, par une cible présentant des irrégularités à l'échelle de la longueur d'onde, soit à la propagation d'un faisceau cohérent dans une atmosphère caractérisée par des variations aléatoires d'indice de réfraction.

3. Le flou est souvent modélisé par un opérateur de convolution qui a pour effet de régulariser fortement les images et donc d'adoucir les détails. L'opération de défloutage s'appelle donc logiquement une *déconvolution*.



(a) Image originale

(b) Bruit blanc gaussien $\sigma = 0.2$ ($\simeq 50$ niveaux de gris)

(c) Bruit poivre et sel



(d) Bruit multiplicatif

Fig. 3.2 Exemples de bruits

Les filtres linéaires d'un signal 1D sont et ne sont que des filtres de convolution : nous allons expliciter cela dans la section suivante. Il est donc naturel de commencer par ce type de filtre pour le filtrage spatial.

3.2 Filtrage spatial (bruit additif)

3.2.1 Filtrage unidimensionnel

Avant de présenter les principales techniques de base pour le filtrage des images, nous rappelons brièvement le principe du filtrage unidimensionnel (pour plus de détails on peut se référer à [15]).

Pour définir un *filtre* linéaire mathématiquement, on se donne deux espaces vectoriels $\mathcal{X}$ (entrée) et $\mathcal{Y}$ (sortie) munis d'une topologie (par exemple des espaces normés) et un opérateur $\mathcal{A}$ linéaire qui, à un signal $e \in \mathcal{X}$ dit signal

d'entrée (*input*), associe un signal $s \in \mathcal{Y}$ appelé signal de sortie (*output*) :

$$\mathcal{A} : e \mapsto s := \mathcal{A}(e) .$$

En pratique, les espaces utilisés pour l'étude des signaux 1D sont les espaces $L^1(\mathbb{R})$, $L^2(\mathbb{R})$ ou $L^\infty(\mathbb{R})$ si le signal est défini sur $\mathbb{R}$ et les espaces $L_p^1(I)$, $L_p^2(I)$ ou $L_p^\infty(I)$ si les signaux sont définis sur un intervalle fini I de $\mathbb{R}$ et périodisés. Ces espaces sont définis dans l'annexe A.1 (on peut voir aussi [15, 47] par exemple).

Définition 3.2.1 (Filtre) *Un filtre linéaire est formé des espaces d'entrée et de sortie et d'un opérateur linéaire continu qui vérifie les deux propriétés suivantes :*

1. *Invariance temporelle : si $\mathcal{T}_a : x \mapsto x(\cdot - a)$ est l'opérateur de translation alors*

$$\mathcal{T}_a \mathcal{A} = \mathcal{A} \mathcal{T}_a .$$

2. *Causalité (ou réalisabilité) : si pour tout t_o donné, on a la propriété suivante :*

$$\forall t < t_o \quad x_1(t) = x_2(t) \implies \forall t < t_o \quad \mathcal{A}x_1(t) = \mathcal{A}x_2(t) .$$

Les espaces peuvent être de dimension infinie (signaux *analogiques*) ou finie (signaux *discrets ou numériques*). On définit un signal *monochromatique* de la manière suivante

$$w_\lambda : t \mapsto \exp(2i\pi\lambda t) , \quad (3.1)$$

de sorte que $w_{n\lambda} = (w_\lambda)^n$.

On peut montrer ([15]) qu'un filtre linéaire associe à tout signal d'entrée monochromatique le même signal multiplié par un facteur indépendant du temps, généralement complexe, appelé *fonction de transfert* ou *gain complexe* du filtre. De plus, un système linéaire continu est un filtre linéaire si et seulement si la relation entre l'entrée e et la sortie s est une convolution :

$$s(t) = (h * e)(t) = \int_{-\infty}^{+\infty} h(\theta) e(t - \theta) d\theta .$$

Pour les filtres à temps discret on a

$$s_n = \sum_{k \in \mathbb{Z}} h_k e_{n-k} .$$

En d'autres termes les filtres linéaires continus unidimensionnels sont et ne sont que des filtres de convolution (où la convolution est soit continue, soit discrète).

Le noyau de convolution h est la *réponse impulsionnelle* du filtre. La transformée de Fourier continue (annexe A.1.3) ou discrète (annexe A.1.2) suivant les cas, $H := \hat{h}$ de h est la *fonction de transfert* du filtre.

Généralement, on distingue les filtres suivant l'action qu'ils ont sur le spectre (c'est-à-dire par la forme de leur fonction de transfert) :

- Définition 3.2.2** – un filtre passe-bas va éliminer ou atténuer fortement l'énergie des hautes fréquences d'un spectre en ne laissant « passer » que les basses fréquences ;
- un filtre passe-haut va éliminer ou atténuer fortement l'énergie des basses fréquences d'un spectre ;
 - un filtre passe-bande ne conservera que l'énergie concentrée dans une bande de fréquences.
 - un filtre coupe-bande qui est le complémentaire du précédent.

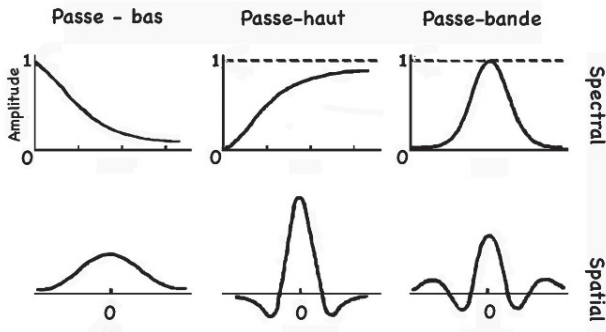


Fig. 3.3 Filtres passe-bas et passe-haut

3.2.2 Filtres de convolution

Comme nous l'avons déjà évoqué, les images peuvent être dégradées par différentes perturbations, dont nous donnons un aperçu dans la figure 3.2 p. 36. Le filtrage a pour but d'éliminer l'effet de ces perturbations en essayant de ne pas toucher aux informations essentielles de l'image (contours, dynamique, textures etc.)

Dans cette section, nous allons présenter les techniques linéaires de base pour effectuer un filtrage permettant de supprimer les effets d'un *bruit additif* que l'on supposera gaussien (centré).

Comme dans le cas 1D, le filtrage linéaire spatial des images est aussi une opération de convolution (2D). Si f est l'image à filtrer (ou à *rehausser*) et κ la réponse impulsionnelle du filtre (spatial) on a :

$$(f * \kappa)(x, y) = \mathcal{F}^{-1} \left\{ \mathcal{F}(f) \cdot \underbrace{\mathcal{F}(\kappa)}_{K(u, v)} \right\} (x, y).$$

où $\mathcal{F}$ est la transformation de Fourier 2D (définie annexe A.1.3.3) et K la fonction de transfert du filtre. Dans le contexte du traitement d'image le noyau κ est appelé **PSF** (*Point Spread Function*) ou *masque*.

Comme les images numériques sont des objets de dimension finie nous allons présenter les filtres dans le cas discret.

Dans tout ce qui suit x et y sont des entiers (coordonnées des pixels) et f est à valeurs entières (dans $\{0, \dots, 255\}$). Comme dans le cas unidimensionnel, on peut distinguer trois types de filtrage :

- le filtre *pas-se-bas* diminue le bruit mais atténue les détails de l'image (flou plus prononcé),
- le filtre *pas-se-haut* accentue les contours et les détails de l'image mais amplifie le bruit,
- le filtre *pas-se-bande* élimine certaines fréquences indésirables présentes dans l'image.

On ne fait pas en général une convolution globale qui serait d'une part coûteuse à implémenter, d'autre part régulariserait l'image dans sa globalité de manière trop prononcée. On choisit des noyaux dont le support est petit de façon à effectuer une convolution locale au voisinage de chaque pixel. Concrètement, cela revient à utiliser des masques (discrets) dont le support ne contient que quelques pixels au voisinage d'un pixel (x, y) :

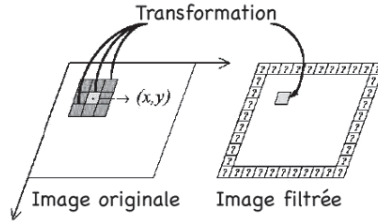


Fig. 3.4 Convolution locale

Le masque du filtre κ est à support inclus dans $[x_1, x_2] \times [y_1, y_2]$:

$$g_{x,y} = (f * \kappa)_{x,y} = \sum_{i=x_1}^{x_2} \sum_{j=y_1}^{y_2} f_{x-i, y-j} \kappa_{i,j}. \quad (3.2)$$

Généralement, le filtre est de dimensions d_1 , d_2 impaires et est symétrique. En effet, l'effet d'un filtre doit être (a priori) isotrope et il ne faut pas privilégier des directions par rapport à d'autres. Dans ce cas

$$[x_1, x_2] = \left[-\frac{d_1-1}{2}, \frac{d_1-1}{2}\right] \quad \text{et} \quad [y_1, y_2] = \left[-\frac{d_2-1}{2}, \frac{d_2-1}{2}\right],$$

$$(f * \kappa)_{x,y} = \sum_{i=-(d_1-1)/2}^{(d_1-1)/2} \sum_{j=-(d_2-1)/2}^{(d_2-1)/2} f_{x+i,y+j} \kappa_{i,j}. \quad (3.3)$$

Donnons un exemple avec $d_1 = d_2 = d = 3$. Ici $x_1 = y_1 = -1$ et $x_2 = y_2 = 1$:

w_1	w_2	w_3	$\leftarrow y - 1$
w_4	w_5	w_6	$\leftarrow y$
w_7	w_8	w_9	$\leftarrow y + 1$
$\uparrow$	$\uparrow$	$\uparrow$	
$x - 1$	x	$x + 1$	

Tableau 3.1 Représentation matricielle du filtre en (x, y) - $d_1 = d_2 = 3$

Ici $\kappa(0,0) = w_5$. Sur cet exemple on a précisément

$$\begin{aligned} g_{x,y} = & w_1 f_{x-1,y-1} + w_2 f_{x,y-1} + w_3 f_{x+1,y-1} \\ & + w_4 f_{x-1,y} + w_5 f_{x,y} + w_6 f_{x+1,y} \\ & + w_7 f_{x-1,y+1} + w_8 f_{x,y+1} + w_9 f_{x+1,y+1}. \end{aligned}$$

Afin de conserver la moyenne de l'image f (et donc sa luminosité), la somme des éléments du filtre est normalisée à 1 : $\sum_{\ell} w_{\ell} = 1$.

On constate tout de suite que la formule précédente ne permet pas de filtrer correctement le bord : en effet les bandes verticales de taille $(d_1 - 1)/2$ sur les bords gauche et droit de l'image ne sont pas « atteintes ». De la même façon les bandes horizontales supérieures et inférieures de taille $(d_1 - 1)/2$ ne sont pas traitées non plus. Pour remédier à cet inconvénient, on effectue souvent une réflexion de l'image autour de ses bords ce qui permet de filtrer l'image réfléchie avant de la redimensionner.

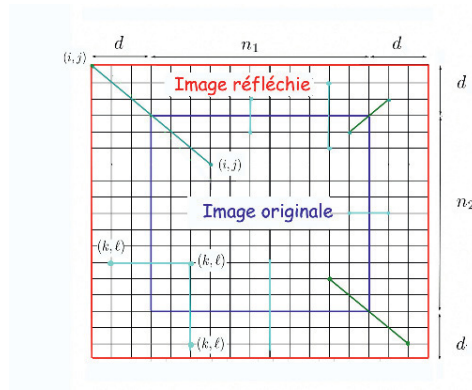


Fig. 3.5 Réflexion d'une image par rapport à ses bords. Ici $(d_1 - 1)/2 = (d_2 - 1)/2 := d$ et la taille de l'image et $n_1 \times n_2$.

Définition 3.2.3 Un filtre 2D est dit séparable s'il est possible de décomposer le noyau de convolution h_{2D} en deux filtres 1D appliqués successivement en horizontal puis en vertical (ou inversement) :

$$h_{2D} = h_{1D}^V \otimes h_{1D}^H,$$

où le symbole $\otimes$ désigne le produit tensoriel. Plus précisément, si $U = (u_i)_{i=1,N}$ et $V = (v_i)_{i=1,N}$ sont deux vecteurs lignes de $\mathbb{R}^N$ et U^t est le vecteur (colonne) transposé de U , alors $U \otimes V$ est le produit matriciel $U^t V$:

$$\forall i, j = 1, \dots, N, \quad (U \otimes V)_{i,j} = U_i V_j.$$

On peut alors traiter séparément les lignes et les colonnes de l'image ce qui est un gros avantage pour l'implémentation.

Proposition 3.2.1 Pour qu'un filtre 2D soit séparable, il faut et il suffit que les coefficients de ses lignes et de ses colonnes soient proportionnels. Plus précisément, un filtre donné par un tableau $N \times M$ dont les coefficients sont $a_{i,j}$, $1 \leq i \leq N$, $1 \leq j \leq M$, est séparable si et seulement si

$$\forall i = 1, \dots, N, \quad \exists C_i^\ell \in \mathbb{R} \text{ tel que } \forall j = 1, \dots, M, \quad a_{i,j} = C_i^\ell a_{1,j},$$

(toutes les lignes sont proportionnelles à la première et donc entre elles) et

$$\forall j = 1, \dots, M, \quad \exists C_j^c \in \mathbb{R} \text{ tel que } \forall i = 1, \dots, N, \quad a_{i,j} = C_j^c a_{i,1},$$

(même chose pour les colonnes).

Pour une dimension 3×3 , les filtres séparables sont obtenus comme suit :

$$\begin{bmatrix} a & b & c \end{bmatrix} \otimes \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} \times \begin{bmatrix} a & b & c \end{bmatrix} = \begin{bmatrix} a\alpha & b\alpha & c\alpha \\ a\beta & b\beta & c\beta \\ a\gamma & b\gamma & c\gamma \end{bmatrix}.$$

Exemple 3.2.1 (Filtre de moyenne passe-bas)

$$\frac{1}{9} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \frac{1}{25} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Tableau 3.2 Filtres de moyenne 3×3 et 5×5 respectivement

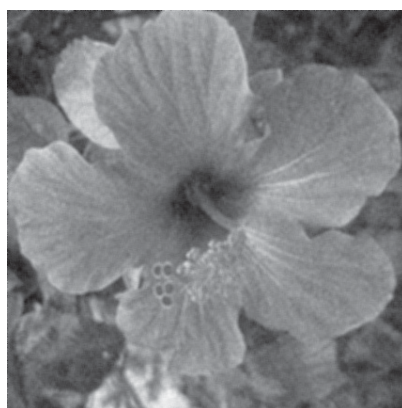
Nous présentons dans la figure 3.6 l'effet des filtres de moyenne sur une image bruitée par un bruit gaussien additif, de moyenne nulle et d'écart-type σ . Ce sont des filtres séparables.



(a) Image originale

(b) Image bruitée (bruit blanc gaussien $\sigma = 0.1 \simeq 25$ niveaux de gris)

(c) Filtre de moyenne 3 x 3



(d) Filtre de moyenne 5 x 5

Fig. 3.6 Effet d'un filtre de moyenne.

On remarque aussi l'effet de lissage induit par l'opération de convolution qui régularise les contours en supprimant les sauts de la fonction. En effet, le contour d'une image est un endroit où le niveau de gris, c'est-à-dire la valeur de la fonction image, varie brutalement (par exemple de 0 à 255 pour un contour noir sur blanc). Cela correspond à une discontinuité de la fonction. Une opération de convolution par un noyau régulier (par exemple continu) renvoie une fonction continue et donc supprime les discontinuités de l'image

d'origine. Si la valeur de la fonction varie rapidement au voisinage d'un point, le contour qui était très marqué auparavant sera remplacé par un contour plus épais où le passage d'un niveau de gris à l'autre (par exemple du blanc au noir) se fera sur plusieurs pixels avec des niveaux de gris intermédiaires. Cela entraîne un floutage du contour. La figure 3.7 illustre le phénomène.

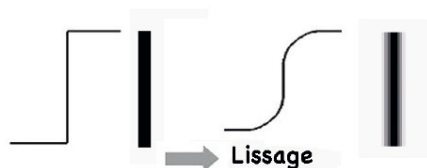


Fig. 3.7 Effet de lissage

Exemple 3.2.2 (Filtre gaussien) *Un filtre gaussien est donné par discrétisation de la fonction gaussienne*

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

sur un voisinage de $(0, 0)$. Ici σ est l'écart-type et la moyenne est nulle.

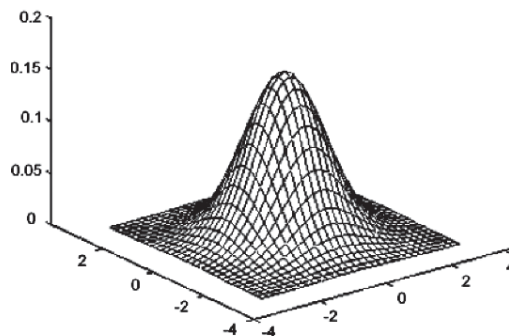


Fig. 3.8 Fonction gaussienne (2D).

Si par exemple $\sigma = 0.8$ on a le filtre 3×3 suivant

$$\begin{array}{|c|c|c|} \hline G(-1, -1) & G(0, -1) & G(1, -1) \\ \hline G(-1, 0) & G(0, 0) & G(1, 0) \\ \hline G(-1, 1) & G(0, 1) & G(1, 1) \\ \hline \end{array} \simeq \frac{1}{16} \cdot \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

et $\sigma = 1$ pour un filtre 5×5 donne environ

$$\frac{1}{300} \cdot \begin{array}{|c|c|c|c|c|} \hline 1 & 4 & 6 & 4 & 1 \\ \hline 4 & 18 & 30 & 18 & 4 \\ \hline 6 & 30 & 48 & 30 & 6 \\ \hline 4 & 18 & 30 & 18 & 4 \\ \hline 1 & 4 & 6 & 4 & 1 \\ \hline \end{array}.$$

La taille du filtre gaussien est gouvernée par σ . En général un filtre gaussien avec $\sigma < 1$ est utilisé pour réduire le bruit. Plus σ est grand, plus le flou appliqué à l'image sera important

Les filtres présentés sont des filtres passe-bas : ils atténuent les détails de l'image (et donc le bruit additif) mais, en érodant les contours, ils ajoutent du flou à l'image. Nous verrons dans une section suivante comment atténuer le flou.

3.3 Filtrage fréquentiel (bruit additif)

3.3.1 Filtre passe-bas

On peut définir un filtre linéaire par sa formulation spatiale (comme convolution) mais aussi par sa formulation fréquentielle, c'est-à-dire par la façon dont il modifie les fréquences de l'image d'entrée, via sa fonction de transfert H . En effet les fréquences de l'entrée f et de la sortie g sont liées par

$$\mathcal{F}(g) = H \mathcal{F}(f),$$

où $\mathcal{F}$ est la transformation de Fourier 2D (voir Annexe A.1.3.3).

Le passe-bas *idéal* est le filtre qui ne modifie pas les fréquences (λ, μ) telles que $\|(\lambda, \mu)\| \leq \delta_c$ (fréquence de coupure) et supprime les autres. Ici $\|\cdot\|$ désigne une norme quelconque de $\mathbb{R}^2$. En pratique, on peut choisir la norme euclidienne :

$$\|(\lambda, \mu)\|_2 = \sqrt{\lambda^2 + \mu^2}$$

comme dans la figure 3.9, mais aussi

$$\|(\lambda, \mu)\|_1 = |\lambda| + |\mu| \text{ ou } \|(\lambda, \mu)\|_\infty = \max(|\lambda|, |\mu|).$$

En d'autres termes

$$H(\lambda, \mu) = \begin{cases} 1 & \text{si } \|(\lambda, \mu)\| \leq \delta_c \\ 0 & \text{sinon} \end{cases}$$

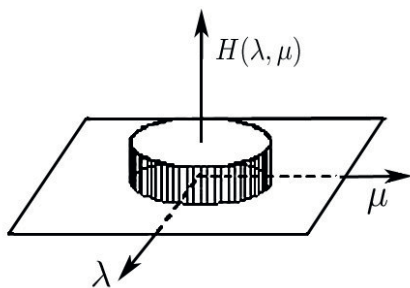


Fig. 3.9 Fonction de transfert *idéale*

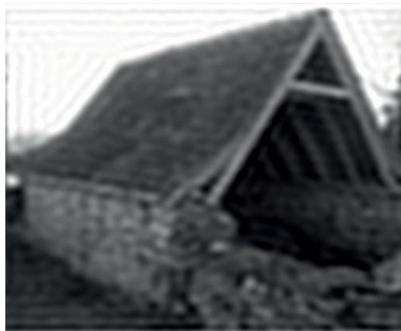
Pour des signaux unidimensionnels, la réponse impulsionnelle $h \in L^2(\mathbb{R})$ d'un tel filtre est un sinus cardinal

$$t \mapsto h(t) = \frac{\sin 2\pi\delta_c t}{\pi t} .$$

Un tel filtre est dit *idéal*. En pratique, la convolution par le sinus cardinal provoque des ondulations (effet *Gibbs*) sur l'image filtrée (Voir les figures 3.10 et 3.11)



(a) Image originale

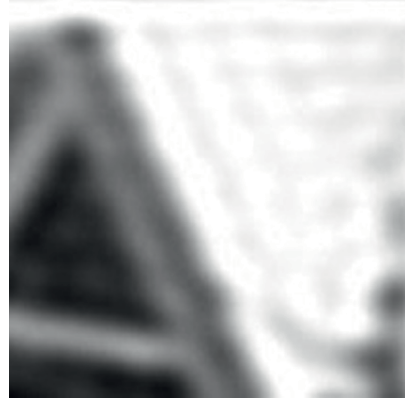


(b) Image filtrée

Fig. 3.10 Application d'un créneau « idéal » ($\delta_c \simeq 15\%$ de la taille de l'image) : on voit les ondulations dues à l'effet Gibbs.



(a) Zoom (coin supérieur droit)- Original



(b) Zoom de l'image filtrée

Fig. 3.11 Zoom sur le coin supérieur droit de la figure précédente : on voit les ondulations et un flou très important

Le filtre passe-bas idéal n'est pas utilisable du fait de la discontinuité de la fonction de transfert qui induit un effet Gibbs important. On fait donc une régularisation de la fonction H précédente qui aura pour effet de supprimer les discontinuités en atténuant (fortement) les hautes fréquences au lieu de les supprimer. Le filtre suivant est le *filtre passe-bas de Butterworth*. La fonction de transfert est alors

$$H(\lambda, \mu) = \frac{1}{1 + \left(\frac{\|(\lambda, \mu)\|}{\delta_c} \right)^{2n}}$$

où δ_c est encore la fréquence de coupure et n un paramètre qui sert à régler l'approximation.

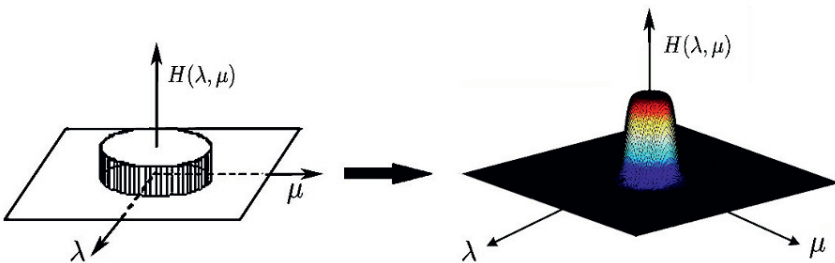


Fig. 3.12 Fonction de transfert de Butterworth ($\delta_c = 0.2$, $n = 5$) - (à gauche la fonction de transfert idéale

En traitement d'image, un filtre passe-bas atténue les hautes fréquences : le résultat obtenu après un tel filtrage est un adoucissement des détails et une réduction du bruit granuleux.

3.3.2 Filtres passe-haut

Le filtre passe-haut idéal est obtenu de manière symétrique au passe-bas comme par exemple dans la figure 3.13

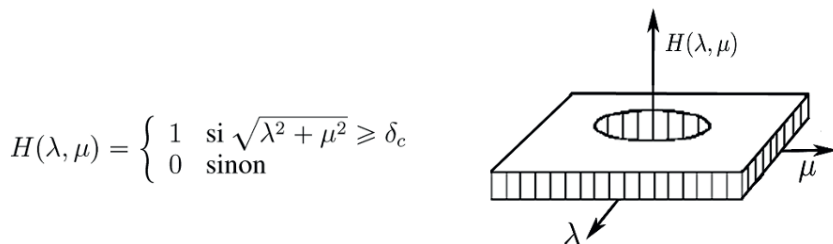


Fig. 3.13 Fonction de transfert du filtre passe-haut idéal avec la norme euclidienne

Le filtre *passe-haut de Butterworth* est donné par

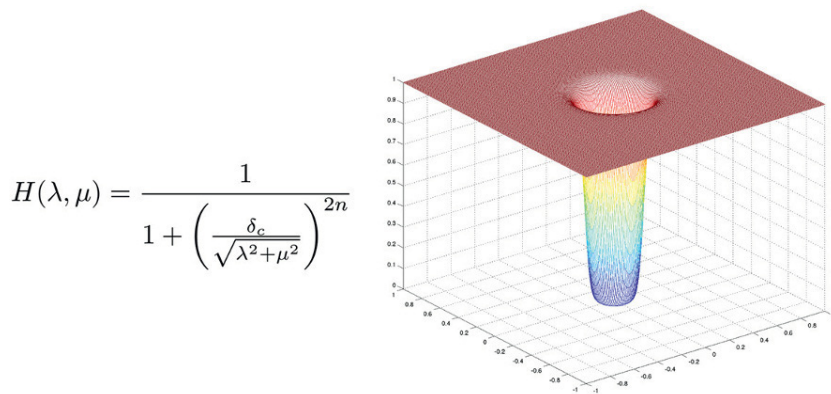


Fig. 3.14 Fonction de transfert du filtre passe-haut de Butterworth ($\delta_c = 0.2$, $n = 5$)

Un filtre passe-haut favorise les hautes fréquences spatiales, comme les détails, et de ce fait, il améliore le contraste. Toutefois, il produit des effets secondaires :

- *augmentation du bruit* : dans les images avec un rapport Signal/ Bruit faible, le filtre augmente le bruit granuleux dans l'image,
- *effet de bord* : il est possible que sur les bords de l'image apparaisse un cadre qui correspond aux effets de bord observés lors de la convolution spatiale. Cet effet peut s'éliminer en faisant une réflexion de quelques pixels de l'image autour de son cadre (Figure 3.5).

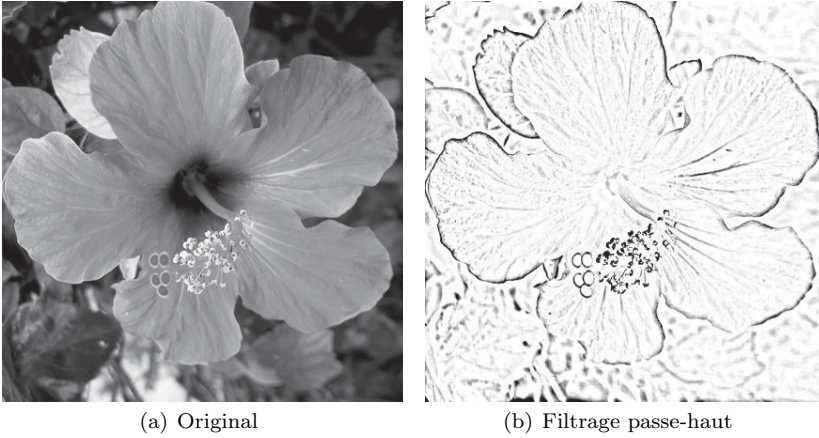


Fig. 3.15 Filtrage passe-haut avec un filtre de Butterworth ($n = 2$ et $\delta_c \simeq 0.1 \times$ taille de l'image) sur une image non bruitée (l'image est recadrée pour l'affichage - voir chapitre 2)

Si on filtre une image avec un filtre passe-bas, l'image obtenue par différence entre l'originale et l'image filtrée par le passe-bas correspond à un filtrage passe-haut.

3.3.2.1 Filtres passe-bande

Mentionnons pour finir les filtres passe-bande (et coupe -bande) moins pertinents dans le cadre 2D que le cadre 1D. Ils permettent de ne garder que les fréquences comprises dans un certain intervalle :

$$H(\lambda, \mu) = \begin{cases} 1 & \text{si } \delta_c - \frac{\varepsilon}{2} \leq \|(\lambda, \mu)\| \leq \delta_c + \frac{\varepsilon}{2} \\ 0 & \text{sinon} \end{cases}$$

où ε est la largeur de bande et δ_c la fréquence de coupure.

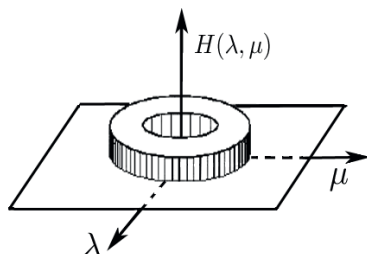


Fig. 3.16 Fonction de transfert d'un filtre passe-bande idéal.

3.4 Filtrage différentiel

Dans les modèles différentiels, on considère l'image comme une fonction $f : \Omega \rightarrow [0, 255]$ où Ω est un ouvert de $\mathbb{R}^2$, dont on étudie le comportement local à l'aide de ses dérivées. Une telle étude n'a de sens que si la fonction f est assez régulière. Ce n'est pas toujours le cas ! Par exemple, une image peut être continue par morceaux (comme un damier) et les points de discontinuités sont les points de contours.

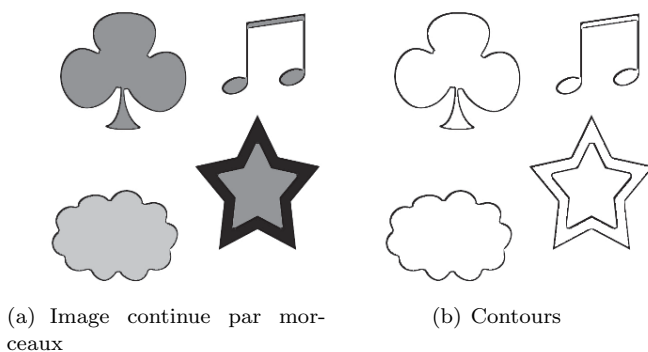


Fig. 3.17 Exemple d'images constantes par morceaux

Il y a bien entendu beaucoup d'autres non-régularités dans une image, dues par exemple aux *textures*, au bruit etc.



Fig. 3.18 Exemple d'images texturées

Grâce au plongement dans l'espace « continu » (de dimension infinie), un grand nombre d'opérations d'analyse peuvent s'exprimer en termes d'équations aux dérivées partielles. Ceci permet de donner un fondement mathématique satisfaisant aux traitements et aussi de fournir des méthodes pour les calculer, par des schémas numériques de résolution avant ou après discrétisation.

Les filtres *différentiels* permettent de mettre en évidence certaines variations spatiales de l'image. Ils sont utilisés comme traitements de base dans de nombreuses opérations comme le *rehaussement de contraste* ou la *détection de contours*. Les filtres présentés dans cette section sont essentiellement des filtres passe-haut. Ils permettent d'isoler les détails d'une image (les contours et les textures sur une image non bruitée et le bruit en plus sur une image bruitée).

3.4.1 Calcul des gradients discrets

Le gradient de l'image f en un point (pixel) $M(x, y)$, s'il existe est égal à :

$$\nabla f(x, y) = \left(\frac{\partial f}{\partial x}(x, y), \frac{\partial f}{\partial y}(x, y) \right).$$

En pratique, il faut approcher ces gradients pour travailler avec des *gradients discrets* qui correspondent à des taux de variation, calculables même si l'image est discontinue (en particulier au voisinage d'un contour). Les approximations les plus simples des dérivées directionnelles se font par différences finies.

Nous allons préciser ce point dans le cas de la dérivée d'une fonction d'une seule variable : on se donne une fonction f continue sur $[0, L]$ et dérivable sur $]0, L[$. On peut approcher la dérivée de la fonction en des points $x_1, \dots, x_{N-1}$ d'une subdivision (uniforme pour simplifier) de $[0, L]$ de la forme : $x_i = ih$, $i = 0, \dots, N$ où $h = \frac{L}{N}$ est le pas de la grille. Si N est assez grand h est petit et on peut approcher la dérivée par le taux de variation :

$$\forall i = 1, \dots, N-1 \quad f'(x_i) \simeq \frac{f(x_i + h) - f(x_i)}{h} = \frac{N}{L}(f(x_{i+1}) - f(x_i)) .$$

C'est ce que nous allons faire en dimension 2 pour calculer les gradients discrets. Toutefois la notion de *taille* de l'image est très relative. En effet, l'image est essentiellement déterminée par le nombre de pixels. La *taille physique*, dépend du support sur lequel on va lire l'image. Une image numérique apparaissant sur un smartphone n'aura pas la même taille, ni la même résolution que la même image projetée sur un écran de cinéma.

Si on considère, par exemple, une image carrée de côté $L = 1(\text{cm})$, qui comporte $N \times N$ pixels, le pas de discrétisation est $h = 1/N$: il est donc petit (si N est grand) et la formule d'approximation par le taux de variation est mathématiquement justifiée (car elle est locale). Si on considère la même image sur un support carré de côté $L = N$ (cm), le pas h vaut 1 mais le gradient de l'image n'a pas changé. Comme on ne sait pas quelle sera la taille du support a priori, nous adopterons la convention suivante : l'image est de « taille » $N \times N$ et le pas $h = 1$. Ainsi, on peut définir le gradient discret de l'image par

$$f_x(x, y) = f(x+1, y) - f(x, y) , \quad f_y(x, y) = f(x, y+1) - f(x, y) ,$$

où (x, y) désigne les coordonnées (entières) du pixel. On peut aussi envisager des différences finies à gauche, centrées, etc. Pour plus de détails on peut se référer à la section 3.4.4 ou à [84].

On peut alors calculer ces gradients à l'aide de convolutions avec des noyaux très simples : par exemple, l'approximation f_x de $\frac{\partial f}{\partial x}$ se fait par convolution avec $[0 \ -1 \ 1]$. Ce noyau n'est pas symétrique mais la formule générale de convolution discrète (3.2) donne :

$$f_x(x, y) = \sum_{i=0}^1 \sum_{j=0}^1 f(x+i, y+j) \kappa_{i,j} = -f(x, y) + f(x+1, y).$$

De même l'approximation f_y de $\frac{\partial f}{\partial y}$ se fait par convolution avec $\begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}$:

$$f_y(x, y) = -f(x, y) + f(x, y+1).$$

0	-1	1	← y
↑	↑	↑	
$x - 1$	x	$x + 1$	

0	← $y - 1$
-1	← y
1	← $y + 1$
↑	
x	

Tableau 3.3 Masques des gradients par rapport à x (gauche) et y (droite)

On utilise plus souvent $[-1 \ 0 \ 1]$ et $\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$ qui produisent des frontières plus épaisses mais qui sont bien centrées. Ces opérations sont très sensibles au bruit et on les combine généralement avec un filtre lisseur dans la direction orthogonale à celle de dérivation, par exemple par le noyau suivant (ou sa transposée) : $[1 \ 2 \ 1]$. On obtient alors des filtres séparables. Le calcul des dérivées directionnelles en x et y revient finalement à la convolution avec les noyaux suivants :

$$f_x(x, y) = (f * h^x)(x, y) \text{ et } f_y(x, y) = (f * h^y)(x, y)$$

avec

$$h^x = [-1 \ 0 \ 1] \otimes [1 \ 2 \ 1]^t = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

et

$$h^y = [1 \ 2 \ 1] \otimes [-1 \ 0 \ 1]^t = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}.$$

Ce sont les *masques de Sobel*. On peut constater dans la figure 3.19 que le gradient discret en x correspond aux contours verticaux et le gradient en y aux contours horizontaux. Cela vient du fait que la première composante x du pixel (x, y) correspond aux colonnes de la matrice représentant l'image et l'indice y aux lignes (voir la figure 1.2 du chapitre 1).

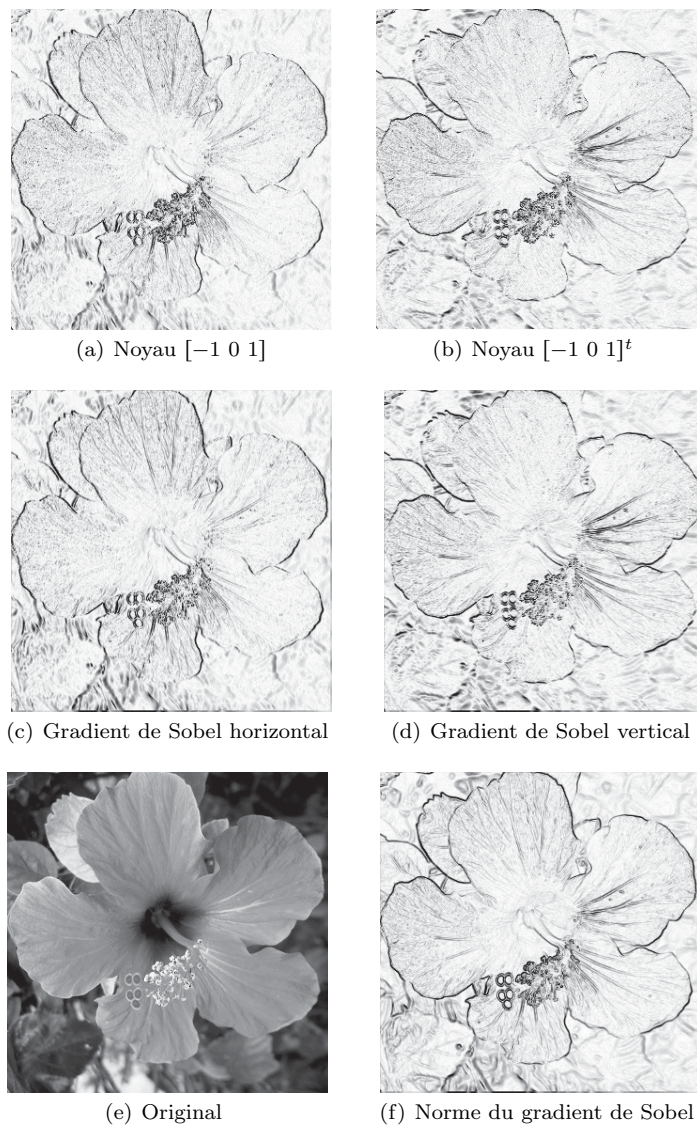


Fig. 3.19 Valeur absolue des gradients et des gradients de Sobel (après passage en négatif puis contraste avec la fonction `imadjust` de MATLAB)

TYPES DE MASQUE	NORME DU GRADIENT	DIRECTION
<i>Masques de Roberts</i> -10 01 0-1 10 G_1, G_2	$A = \sqrt{G_1^2 + G_2^2}$	$\theta = \frac{\pi}{4}$ + $\arctan\left(\frac{G_2}{G_1}\right)$
<i>Masques de Sobel</i> -101 -202 -101 -1-2-1 000 121 G_x, G_y	$A = \sqrt{G_x^2 + G_y^2}$	$\theta = \arctan\left(\frac{G_y}{G_x}\right)$
<i>Masques de Prewitt</i> 10-1 10-1 10-1 111 000 -1-1-1 G_x, G_y	$A = \sqrt{G_x^2 + G_y^2}$	$\theta = \arctan\left(\frac{G_y}{G_x}\right)$
<i>Masques de Kirsh</i> 555 -30-3 -3-3-3 + les 7 autres masques obtenus par permutation circulaire des coefficients G_i pour i de 1 à 8	maximum des $ G_i $	Direction correspondant au G_i sélectionné
<i>Masques de Robinson</i> 111 1-21 -1-1-1 + les 7 autres masques obtenus par permutation circulaire des coefficients G_i pour i de 1 à 8	maximum des $ G_i $	Idem

Tableau 3.4 Différents types de masques pour le gradient, sa norme et sa direction



Fig. 3.20 Gradients de Robinson (valeur absolue contrastée) dans 3 directions différentes (voir tableau 3.4)

On remarque que la norme du gradient est un bon détecteur de contour : en effet un contour est un endroit où l'on observe de fortes variations de niveaux de gris, c'est-à-dire des gradients importants. Les points correspondant aux maxima de la norme du gradient sont donc des points appartenant à des contours. Toute la difficulté de la segmentation (c'est-à-dire de la recherche des contours dans une image) est alors de trouver des critères permettant de faire une sélection pertinente parmi ces points.

3.4.2 Approximation de la dérivée seconde

De la même façon, l'approximation par différences finies la plus simple de la dérivée seconde est la convolution par le noyau $[1 \ -2 \ 1]$ pour l'approximation f_{xx} de $\frac{\partial^2 f}{\partial x^2}$ et $\begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$ pour l'approximation f_{yy} de $\frac{\partial^2 f}{\partial y^2}$.

Le laplacien $\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$ peut être approché par l'un opérateurs linéaires suivants :

<i>Laplacien discret - 4</i>	<i>Laplacien discret - 8</i>																		
<table><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>-4</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr></table>	0	1	0	1	-4	1	0	1	0	<table><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>-8</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	1	1	1	1	-8	1	1	1	1
0	1	0																	
1	-4	1																	
0	1	0																	
1	1	1																	
1	-8	1																	
1	1	1																	

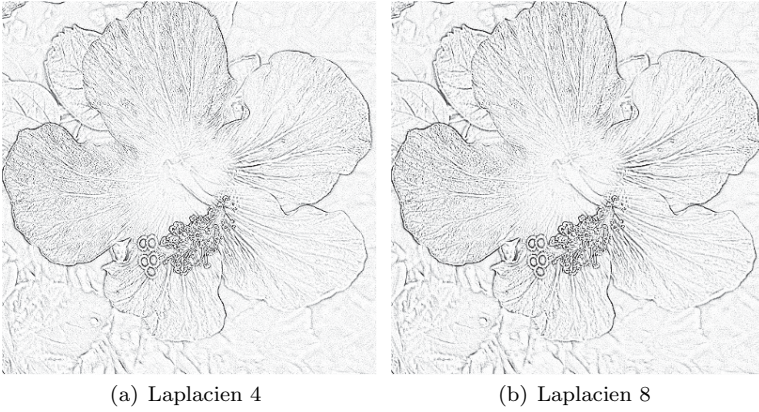


Fig. 3.21 Valeur absolue du laplacien (avec rehaussement de contraste)

3.4.3 Filtrage par équations aux dérivées partielles

3.4.3.1 Equation de la chaleur

Considérons un filtrage gaussien dans le cadre continu. On sait que si l'image de départ est une fonction $u_o \in L^\infty(\Omega)$ à support compact inclus dans $\Omega \subset \mathbb{R}^2$, l'image filtrée est la convolée de u_o avec un noyau gaussien

$$G_\sigma(x) = G_\sigma(x_1, x_2) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x_1^2 + x_2^2}{2\sigma^2}\right) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{\|x\|^2}{2\sigma^2}\right).$$

On pose $u(t, x) = (h(t, \cdot) * u_o)(x)$ où $h(t, x) = G_{\sqrt{2t}}(x) = \frac{1}{4\pi t} \exp\left(-\frac{\|x\|^2}{4t}\right)$. Comme $h(t, \cdot) \in \mathcal{C}^\infty(\mathbb{R}^2)$ a ses dérivées bornées et $u_o \in L^1(\mathbb{R}^2)$, la convolée $u(t, \cdot)$ est aussi $\mathcal{C}^\infty$ et on peut calculer Δu :

$$\forall t > 0, \forall x \in \mathbb{R}^2 \quad \Delta u(t, x) = \frac{\partial^2 u}{\partial x_1^2}(t, x) + \frac{\partial^2 u}{\partial x_2^2}(t, x) = (\Delta h(t, \cdot) * u_o)(x).$$

Un rapide calcul montre que

$$\Delta h(t, x) = \left(-\frac{1}{4\pi t^2} + \frac{\|x\|^2}{16\pi t^3}\right) \exp\left(-\frac{\|x\|^2}{4t}\right) = \left(-\frac{1}{t} + \frac{\|x\|^2}{4t^2}\right) h(t, x),$$

et on obtient

$$\Delta u(t, x) = \left(-\frac{1}{t} + \frac{\|x\|^2}{4t^2}\right) u(t, x).$$

D'autre part, pour $t > 0$ on peut dériver directement u par rapport à t :

$$\frac{\partial u}{\partial t}(t, x) = \iint_{\mathbb{R}^2} \frac{\partial h}{\partial t}(t, y) u_o(x - y) dy$$

ce qui donne :

$$\frac{\partial u}{\partial t}(t, x) - \Delta u(t, x) = 0 \text{ sur }]0, t[\times \mathbb{R}^2.$$

D'autre part, avec

$$u(t, x) = \iint_{\mathbb{R}^2} \frac{1}{4\pi t} \exp\left(-\frac{\|y\|^2}{4t}\right) u_o(x - y) dy$$

on obtient

$$\lim_{t \rightarrow 0} u(t, x) = \langle \delta_x, u_o \rangle = u_o(x),$$

car la famille de fonctions gaussiennes converge au sens des distributions vers la mesure de Dirac. On obtient alors le résultat suivant

Proposition 3.4.1 *La fonction filtrée $u(t, \cdot) = h(t, \cdot) * u_o$ vérifie l'équation aux dérivées partielles suivante (équation de la chaleur) :*

$$\begin{cases} \frac{\partial u}{\partial t}(t, x) - \Delta u(t, x) = 0 & \text{dans }]0, T[\times \mathbb{R}^2 \\ u(0, x) = u_o(x) & x \in \mathbb{R}^2, \end{cases} \quad (3.4)$$

où u_o (qui représente l'image) est une fonction à support compact dans $\overline{\Omega}$ et Ω est un ouvert de $\mathbb{R}^2$.

On peut étendre cette formulation en imposant que le support de la fonction $u(t, \cdot)$ reste dans Ω . On considère alors l'équation dans $]0, T[\times \Omega$ et on fixe soit des conditions aux limites au bord de Ω , soit des conditions aux limites périodiques en périodisant la fonction u (si le cadre est rectangle par exemple). Habituellement, on impose une condition frontière de Neumann, ce qui donne

$$\begin{cases} \frac{\partial u}{\partial t}(t, x) - \Delta u(t, x) = 0 & \text{dans }]0, T[\times \Omega \\ \frac{\partial u}{\partial n} = 0 & \text{sur }]0, T[\times \partial\Omega \\ u(0, x) = u_o(x) & x \in \Omega. \end{cases} \quad (3.5)$$

On rappelle que la dérivée normale de u est obtenue en $(t, x) \in]0, T[\times \partial\Omega$ par

$$\frac{\partial u}{\partial n}(t, x) = (\nabla u(t, x), \mathbf{n}_x),$$

où $\mathbf{n}_x$ est le vecteur normal (sortant) à $\partial\Omega$ en x

Suivant le temps d'évolution, on obtient une version plus ou moins lissée de l'image de départ. On peut alors faire une convolution par le noyau $G_{\sqrt{2t}}$ en utilisant une FFT (voir annexe, section A.1.2) pour calculer la solution de l'EDP (3.4). On peut aussi utiliser un schéma aux différences finies pour le calcul de celle de l'EDP (3.5). Nous détaillons cela dans la section 3.4.4 suivante.

3.4.4 Mise en œuvre numérique

La mise en œuvre numérique se fait avec une discrétisation en différences finies, la plupart du temps explicite en raison de la très grande taille des images (et donc des matrices associées). La condition de Neumann est assurée grâce à une réflexion de l'image par rapport à ses bords.

Dans ce qui suit, l'image (discrète) est notée u et sa valeur au pixel (i, j) : $u_{i,j}$. Comme mentionné précédemment, on choisit le pas de discrétisation égal à $h = 1$. On peut discrétiser le gradient de différentes manières (centrée, à droite, à gauche)

$$\delta_x u_{i,j} = \frac{u_{i+1,j} - u_{i-1,j}}{2}, \quad \delta_y u_{i,j} = \frac{u_{i,j+1} - u_{i,j-1}}{2}, \quad (3.6)$$

$$\delta_x^+ u_{i,j} = u_{i+1,j} - u_{i,j}, \quad \delta_y^+ u_{i,j} = u_{i,j+1} - u_{i,j},$$

$$\delta_x^- u_{i,j} = u_{i,j} - u_{i-1,j}, \quad \delta_y^- u_{i,j} = u_{i,j} - u_{i,j-1}.$$

La norme du gradient peut se calculer par exemple par

$$|\nabla u|_{i,j} = \sqrt{(\delta_x^+ u_{i,j})^2 + (\delta_y^+ u_{i,j})^2}.$$

On rappelle que l'opérateur de divergence est défini par

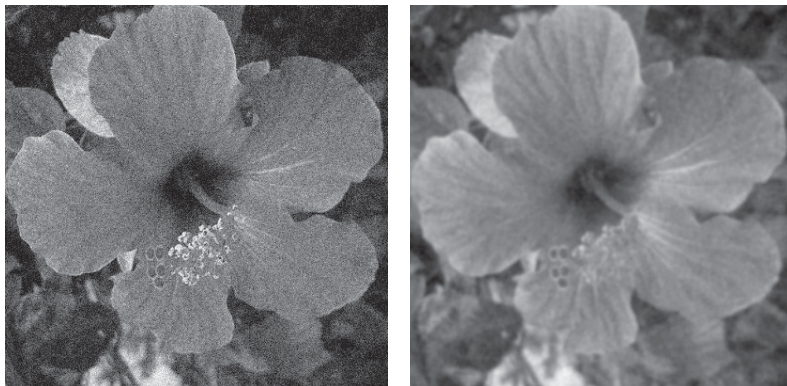
$$\operatorname{div} \mathbf{p}(x, y) = \frac{\partial p_1}{\partial x}(x, y) + \frac{\partial p_2}{\partial y}(x, y),$$

pour tout champ $\mathbf{p} = (p_1, p_2)$ de $\mathbb{R}^2$ dans $\mathbb{R}^2$ dérivable. Nous verrons dans la section 4.3.2 du chapitre 4 que l'opérateur de divergence est aussi l'opérateur adjoint du gradient (au signe près).

Si l'opérateur gradient est discrétisé par différences finies à droite (avec $\delta_x^+ u$ et $\delta_y^+ u$), alors une discrétisation possible de la divergence d'un couple $\mathbf{p} = (p^1, p^2)$ est donnée par

$$(\operatorname{div} p)_{i,j} = \begin{cases} p_{i,j}^1 - p_{i-1,j}^1 & \text{si } 1 < i < N \\ p_{i,j}^1 & \text{si } i = 1 \\ -p_{i-1,j}^1 & \text{si } i = N \end{cases} + \begin{cases} p_{i,j}^2 - p_{i,j-1}^2 & \text{si } 1 < j < M \\ p_{i,j}^2 & \text{si } j = 1 \\ -p_{i,j-1}^2 & \text{si } j = M \end{cases} \quad (3.7)$$

avec (N, M) est la taille des images p^1 et p^2 . Comme $\Delta f = \operatorname{div} \nabla f$ pour toute fonction deux fois dérivable de $\mathbb{R}^2$ dans $\mathbb{R}$, on peut alors calculer une version discrète du laplacien (ou utiliser les masques de la section 3.4.2).



(a) Image bruitée - bruit gaussien $\sigma = 0.1$ ($\simeq 25$ niveaux de gris)

(b) Image filtrée

Fig. 3.22 Filtrage par EDP de la chaleur avec pas de temps $dt = 0.2$ et 10 itérations

3.5 Déconvolution (cas d'un flou)

Une autre source de perturbation d'une image est le *flou*. Nous avons vu qu'un filtre de convolution passe-bas permettait d'enlever le bruit additif mais que l'image filtrée était floutée. Cela s'explique par le fait que l'opérateur de convolution est régularisant. L'opérateur de flou est donc souvent modélisé par un produit de convolution, de noyau positif symétrique h (qui est la plupart du temps gaussien). Il n'est pas nécessairement inversible (et même lorsqu'il est inversible, son inverse est souvent numériquement difficile à calculer).

3.5.1 Approche « spatiale » : équation de la chaleur rétrograde (inverse)

Nous avons constaté précédemment que faire une convolution par un noyau gaussien revient à résoudre une équation de la chaleur. Le temps final joue le rôle de l'écart type de la gaussienne. Pour faire l'opération inverse, la déconvolution on peut donc imaginer de résoudre une équation de la chaleur

« rétrograde » en partant de l'état « final » qui est l'image floutée et en ajustant le temps final au rapport signal sur bruit.

$$\begin{cases} \frac{\partial u}{\partial t}(t, x) + \Delta u(t, x) = 0 & \text{dans }]0, T[\times \Omega \\ u(T, x) = u_o(x) & \forall x \in \Omega \end{cases} \quad (3.8)$$



(a) Original



(b) Image floutée



(c) Itération 5



(d) Itération 6

Fig. 3.23 Déconvolution par équation de la chaleur inverse : $dt = 0.5$

Cette équation est notoirement mal posée (on ne peut assurer ni l'existence d'une solution, ni la stabilité d'un schéma numérique) et il convient de ne faire qu'un petit nombre d'itérations.

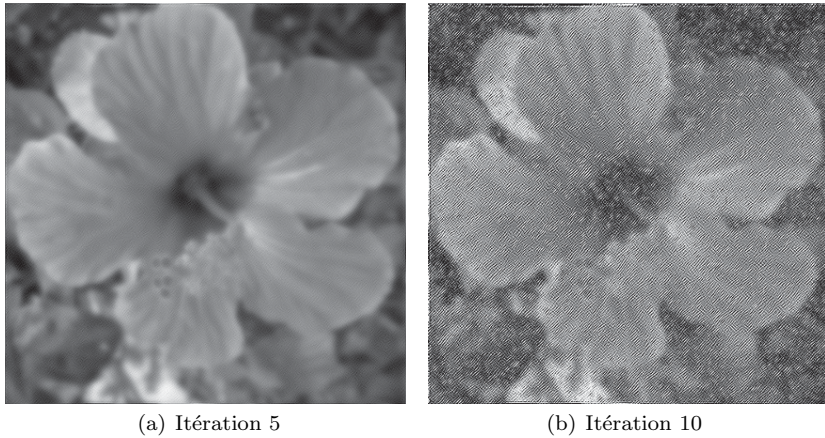


Fig. 3.24 Déconvolution par équation de la chaleur inverse : $dt = 0.5$

3.5.2 Filtre inverse et algorithme de Van Cittert

L'algorithme de Van Cittert repose sur la formulation fréquentielle d'une convolution. Supposons que h soit l'opérateur de flou (inconnu ou donné par étalonnage des appareils de mesure) et qu'on ne prenne pas en compte le bruit. L'image floutée f vérifie $f = h * u$, où u est l'image originale (qu'on veut retrouver). Une formulation équivalente est $\hat{f} = \hat{h}\hat{u}$ c'est-à-dire $\hat{u} = \frac{\hat{f}}{\hat{h}}$ où $\hat{f}$ désigne la transformée de Fourier de f (Annexe A.1.3). Le filtre *inverse* est le plus simple des filtres. Dans certaines conditions, il peut donner de très bons résultats. Il consiste donc à calculer $1/\hat{h}$ et à l'appliquer à l'image floutée. C'est le meilleur filtre pour déconvoluer une image non bruitée. Toutefois il n'est pas toujours possible de calculer $1/\hat{h}$ car $\hat{h}$ peut s'annuler. Une alternative est l'algorithme de Van Cittert.

Posons $\hat{g} = 1 - \hat{h}$ de sorte que *formellement* on obtient

$$\hat{u} = \frac{\hat{f}}{1 - \hat{g}} = \left(\sum_{k=0}^{+\infty} \hat{g}^k \right) \hat{f}.$$

Si on pose $u_o = f$ et $\hat{u}_n = \left(\sum_{k=0}^n \hat{g}^k \right) \hat{f}$ pour tout $n \geq 1$ on obtient

$$\hat{u}_{n+1} = \hat{f} + \hat{g}\hat{u}_n = \hat{f} + (1 - \hat{h})\hat{u}_n,$$

ou de manière équivalente

$$u_{n+1} = f + u_n - h * u_n.$$

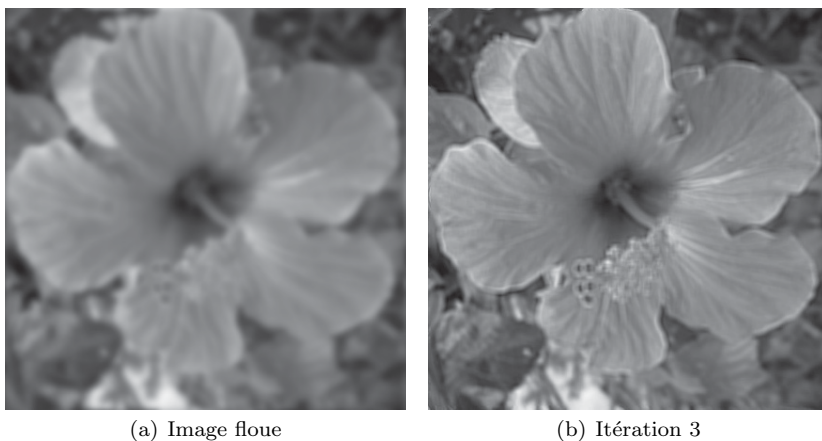


Fig. 3.25 Déconvolution par algorithme de Van Cittert : h est un masque gaussien de taille 9 et d'écart-type $\sigma = 4$ (connu)

La principale difficulté dans l'utilisation de cet algorithme est le choix a priori du filtre h .

Le filtre inverse est une technique très utile pour la déconvolution mais il ne prend pas en compte le bruit. Si f est une image floutée **et** bruitée : $f = h * u + b$, où u est l'image originale à restaurer et b un bruit blanc gaussien, le passage à un filtre inverse donne $\hat{u} = \frac{\hat{f}}{\hat{h}} - \frac{\hat{b}}{\hat{h}}$. Le bruit blanc chargeant uniformément les fréquences on a $\hat{b} \simeq 1$ et si h est un filtre passe-bas $\hat{h} \simeq 0$ au voisinage de l'infini (c'est-à-dire au dessus d'une fréquence de coupure λ_c). Il s'ensuit que le filtrage est efficace dans une bande de fréquences inférieures à λ_c mais que le bruit est amplifié au delà.

Pour traiter des images à la fois floutées et bruitées on préfère utiliser le filtre de Wiener (voir section 4.5.1).

Introduction au traitement mathématique des images -
méthodes déterministes

Bergounioux, M.

2015, XVII, 241 p. 168 ill., Softcover

ISBN: 978-3-662-46538-7