



MA323 : Techniques avancées en Signal et Image

Traitement d'images par approches variationnelles et équations aux dérivées partielles

Auteur:
Gweltaz LEVER

Responsable du cours:
P. WEISS

ISAE - Supaero
10 Avenue Edouard Belin, 31400 TOULOUSE

mars 2015

Sommaire

Sommaire	i
Table des figures	ii
Nomenclature	ii
1 TP 2 et 4	1
1.1 TP2 / Equation de la chaleur : filtrage linéaire et non linéaire	1
1.1.1 Discrétisation et opérateurs de base	1
1.1.2 Equation de la chaleur	3
1.1.3 Détecteur de Hildrett-Marr	4
1.1.4 Perona-Malik	7
1.2 TP4 / Restauration d'images	9
1.2.1 Régularisation de Tychonov	9
1.2.2 Modèle de Rudin-Osher-Fatemi	10
1.2.3 Déconvolution	16
1.2.4 Seuillage en ondelettes	16
1.3 Conclusion	17

Table des figures

1.1	Application du gradient	2
1.2	Laplacien de Lena	3
1.3	Calcul de la courbure	4
1.4	Application de l'équation de la chaleur avec un pas de temps $dt = 0.1$ et un nombre d'itérations $it = 500$	5
1.5	Comparaison de l'image obtenue par l'équation de la chaleur et par convolution gaussienne	6
1.6	Comparaison des contours extraits par méthode de Hildrett sur l'image originale, lissée et avec la fonction <i>edge</i>	6
1.7	Comparaison des résultats de l'algorithme de l'équation de la chaleur et de Perona-Malik dans le cas d'une image bruitée, $it = 300$	7
1.8	Comparaison des algorithmes de régularisation de Tychonov, $dt = 0.05$, $it = 300$	10
1.9	Restauration (ROF - descente de gradient), $SNR = 22.68$, $\lambda = 1.5$, $it = 500$, $\varepsilon = 1$	11
1.10	Images restaurées par ROF - projection de Chambolle, $\tau = 1/8$, $it = 100$	12
1.11	Images restaurées par ROF - gradient projeté, $\tau = 1/8$, $it = 200$	13
1.12	Restauration par algorithme de Nesterov, $SNR = 23.75$, $\lambda = 0.1$, $it = 200$	13
1.13	Images restaurées par ROF	15
1.14	Images déconvoluées, $\tau = 1/8$, $\lambda = 0.02$, $\varepsilon = 0.1$	16
1.15	Images restaurées par transformée en ondelettes non invariantes par translation	18
1.16	Images restaurées par transformée en ondelettes invariantes par translation	19

BE 1

TP 2 et 4

L'objectif de ce cours est de se familiariser avec des techniques de traitement d'image basées sur des approches variationnelles et des équations aux dérivées partielles. Dans un premier volet, nous nous intéresserons à du filtrage linéaire et non linéaire utilisant l'équation de la chaleur et des processus de convolution. Dans un second temps, nous focaliserons nos efforts sur la restauration d'images, notamment en considérant la régularisation de Tychonov et le modèle de Rudin-Osher-Fatemi. L'image considérée au cours de ce BE est l'image classique de Lena.

Pour exécuter les scripts, il suffit de lancer les scripts TP2 et TP4.

1.1 TP2 / Equation de la chaleur : filtrage linéaire et non linéaire

1.1.1 Discrétisation et opérateurs de base

Les opérateurs

Afin d'appliquer l'équation de la chaleur à l'image de notre choix, il est nécessaire d'implémenter les fonctions de base des équations aux dérivées partielles (EDP), à savoir les opérateurs de gradient, divergence, laplacien et la discrétisation de la courbure (cf. p. 24 - Section 4.2 du cours). Une image numérique étant discrète, elle peut être décrite par un vecteur à deux dimensions de taille $N \times N$. On note X l'espace euclidien $\mathbb{R}^{N \times N}$ et $Y = X \times X$ que l'on munit du produit scalaire :

$$(u, v)_X = \sum_{1 \leq i, j \leq N} u_{i,j} v_{i,j} \quad (1.1)$$

On peut alors définir le gradient d'un élément $u \in X$ comme le vecteur $\nabla u \in Y$ donné par :

$$(\nabla u)_{i,j} = ((\nabla u)_{i,j}^1, (\nabla u)_{i,j}^2) \quad (1.2)$$

avec

$$(\nabla u)_{i,j}^1 = \begin{cases} u_{i+1,j} - u_{i,j} & \text{si } i < N, \\ 0 & \text{if } i = N. \end{cases} \quad (1.3)$$

et

$$(\nabla u)_{i,j}^2 = \begin{cases} u_{i,j+1} - u_{i,j} & \text{si } j < N, \\ 0 & \text{if } j = N. \end{cases} \quad (1.4)$$

De la même façon, on définit la divergence sur Y comme $\text{div} = -\nabla^*$. La version discrète de l'opérateur Laplacien s'écrit alors $\Delta = \text{div}(\nabla u)$.

Enfin, on définit la courbure de la façon suivante :

$$\text{curv}(u)(x) = \text{div}\left(\frac{\nabla u}{\|\nabla u\|_2}\right)(x) \quad (1.5)$$



(a) Représentation du gradient



(b) Detail du gradient

FIGURE 1.1 Application du gradient

Application à une image

Le gradient, le laplacien et la courbure ont alors été appliqués à une image de niveaux de gris.

Le gradient Le gradient nous donne une indication sur les variations d'intensités au sein de l'image. Celui-ci a été représenté sur une grille de points pour l'image Lena, Figure 1.1. On constate donc que les zones de contour sont des zones de fort gradient, l'intensité y étant très variable. En revanche les zones uniformes vont présenter un faible gradient.

Le laplacien Le laplacien permet de mettre en valeur les régions à fort changement d'intensité, ce qui permet d'extraire les contours d'une image, Figure 1.2. Les frontières entre le chapeau et l'arrière plan ou entre les cheveux de Lena et l'arrière plan apparaissent donc clairement.

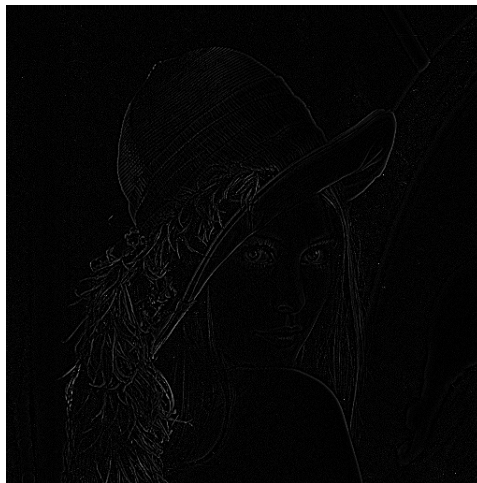


FIGURE 1.2 Laplacien de Lena

La courbure La courbure représente les lignes de niveau de l'intensité. Il est assez dur de cerner ce qu'elle met en valeur sur une image de Lena qui reste naturellement bruitée Figure 1.3a. Cet opérateur a donc aussi été appliqué à un carré Figure 1.3b. Cette image permet de mieux cerner sa signification : la courbure est forte au niveau des forts changements le long d'une ligne de niveau. Ainsi, la courbure est élevée au niveau des coins du carré.

1.1.2 Equation de la chaleur

La première équation aux dérivées partielles qui a été utilisée en traitement d'images est l'équation de la chaleur. Une équation que l'on retrouve dans de nombreux domaines de la

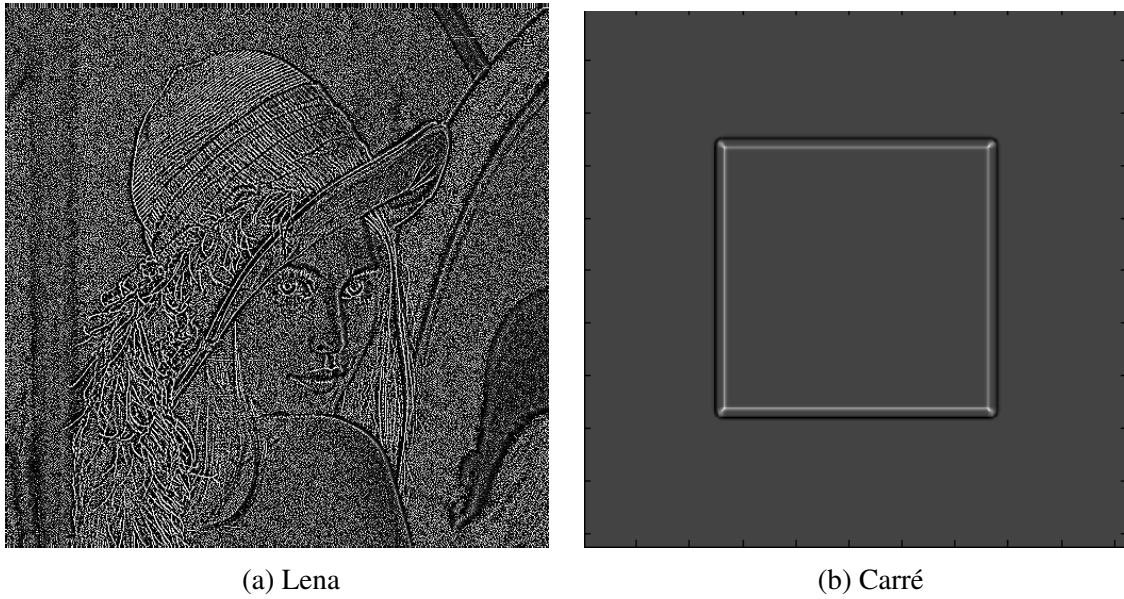


FIGURE 1.3 Calcul de la courbure

physique. Cette équation parabolique s'écrit :

$$\begin{cases} \frac{\partial u}{\partial t} - \Delta u(t, x) = 0 & \text{si } t \leq 0 \text{ et } x \in \mathbb{R}^2 \\ u(0, x) = u_0(x)S. \end{cases} \quad (1.6)$$

En appliquant cette équation à une image non bruitée et une image bruitée de Lena, on obtient les images ci-dessous, Figure 1.4.

L'équation de la chaleur rend l'image de Lena floue. En effet, cette équation effectue un lissage gaussien. Ceci conduit à une perte des contours, c'est-à-dire à un filtrage passe-bas de l'image. Dans le cas de l'image bruitée, le bruit *poivre et sel* est gommé de l'image. Les pixels bruités présentant des zones à forte variation d'intensité, ils sont lissés. On obtient donc une image comparable à l'image non bruitée.

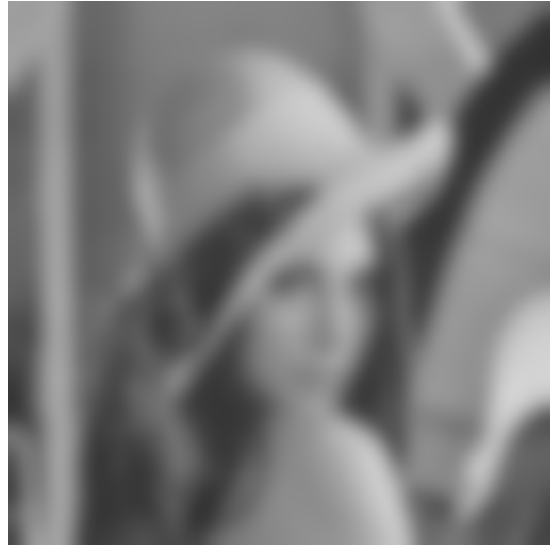
De plus, afin de vérifier que l'EDP de la chaleur équivaut à effectuer un lissage gaussien, on peut comparer notre résultat à une convolution par un noyau gaussien comme cela a été effectué Figure 1.5. On constate en effet un résultat similaire, bien que les paramètres diffèrent légèrement, l'image obtenue par convolution étant moins lissée.

1.1.3 Détecteur de Hildrett-Marr

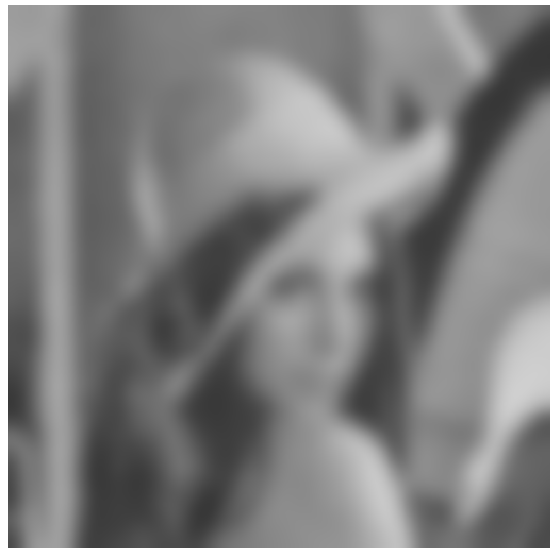
A partir d'une image lissée, il est possible de détecter sans trop de difficulté les bords d'une image, celle-ci ayant été débarrassée des impuretés et du bruit qu'elle contenait. Le détecteur de Hildrett-Marr consiste à chercher les zéros du Laplacien d'une image pour lesquels le



(a) Image originale



(b) Après application de l'équation de la chaleur

(c) Image bruitée à l'aide d'un bruit *poivre et sel*

(d) Après application de l'équation de la chaleur

FIGURE 1.4 Application de l'équation de la chaleur avec un pas de temps $dt = 0.1$ et un nombre d'itérations $it = 500$



FIGURE 1.5 Comparaison de l'image obtenue par l'équation de la chaleur et par convolution gaussienne

gradient est non nul. On extrait ainsi les bords de notre image.

Appliqué à l'image de Lena non lissée, un tel algorithme n'est pas efficace, Figure 1.6a, en revanche, lorsqu'on lisse suffisamment notre image, les contours sont extraits de manière correcte, Figure 1.6b.

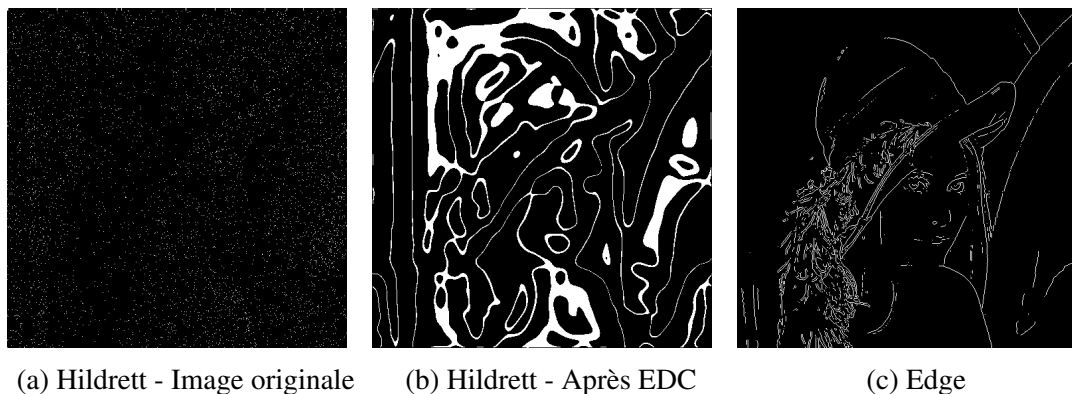


FIGURE 1.6 Comparaison des contours extraits par méthode de Hildrett sur l'image originale, lissée et avec la fonction *edge*

Si on compare cette méthode avec la fonction *edge* qui se base sur l'approximation de Sobel de la dérivée, on remarque qu'avec la méthode de Hildrett tous les contours ont été extraits mais certaines parties sont difficiles à interpréter (régions toutes blanches) tandis que la fonction *edge* ne permet pas d'obtenir un contour complet : on observe des discontinuités

et certains contours n'ont pas été détectés, Figure 1.6c .

1.1.4 Perona-Malik

Afin d'améliorer les résultats obtenus lors de l'application de l'EDP de la chaleur sur les bords, Perona et Malik ont proposé un nouveau modèle numérique dirigé par l'équation suivante :

$$\begin{cases} \frac{\partial u}{\partial t} = \operatorname{div}(c(|\nabla u|)\nabla u) & \text{dans } \Omega \times (0, T) \\ \frac{\partial u}{\partial N} = 0 & \text{sur } \partial\Omega \times (0, T) \\ u(0, x) = u_0(x) & \text{dans } \Omega. \end{cases} \quad (1.7)$$

Avec c une fonction décroissante de \mathbb{R}_+ dans \mathbb{R}_+ . On constate que le modèle est bien similaire à celui de l'équation de la chaleur car en prenant $c = 1$ on retombe sur cette même équation.

Deux fonctions c_i ont été testées :

$$c_1(t) = \frac{1}{\sqrt{1 + (t/\alpha)}} \quad (1.8)$$

$$c_2(t) = \frac{1}{1 + (t/\alpha)} \quad (1.9)$$

Les résultats de l'algorithme appliqué à l'image bruitée précédente sont présentés, Figure 1.7.



(a) Résultat de l'équation de la chaleur, $dt = 0.1$

(b) Perona, fonction c_1 , $dt = 0.1$, $\alpha = 0.1$

(c) Perona, fonction c_2 , $dt = 0.2$, $\alpha = 0.3$

FIGURE 1.7 Comparaison des résultats de l'algorithme de l'équation de la chaleur et de Perona-Malik dans le cas d'une image bruitée, $it = 300$

Dans le cas de la fonction c_1 on constate clairement que les bords sont mieux conservés que dans le cas de l'équation de la chaleur. Cette remarque est un peu moins vérifiée dans le cas de la fonction c_2 .

1.2 TP4 / Restauration d'images

Dans ce second TP, l'objectif est de restaurer une image bruitée à l'aide de différents algorithmes. L'image originale u est dégradée par un bruit additif v considéré comme gaussien. Ainsi, on observe une image $f = u + v$. Le but est de reconstruire l'image u à partir de cette observation f . Notre observation est une image de Lena à laquelle nous avons ajouté un bruit gaussien de variance 0.01.

La méthode du maximum de vraisemblance nous conduit à chercher une solution au problème de minimisation,

$$\inf_u \|f - u\|_2^2 \quad (1.10)$$

Pour le résoudre numériquement, il est nécessaire d'introduire un terme de régularisation,

$$\inf_u \|f - u\|_2^2 + L(u) \quad (1.11)$$

1.2.1 Régularisation de Tychonov

La régularisation de Tychonov consiste à considérer le problème basé sur l'équation précédente avec $L(u)$, tel que

$$\inf_u \|f - u\|_2^2 + 2\lambda \|\nabla u\|_X^2 \quad (1.12)$$

Résolution par EDP

On peut alors déduire l'équation d'Euler associée à ce problème. Ceci nous amène à un premier algorithme de descente de gradient. Celui-ci se base sur l'EDP suivante :

$$\frac{\partial u}{\partial t} = \lambda \Delta u - u + f \quad (1.13)$$

Cet algorithme a été appliqué à l'image de Lena, Figure 1.8. Le rapport signal à bruit (SNR) qui était de 14.36 dB pour l'image de base passe à 22.56 dB. L'algorithme est donc assez efficace malgré l'aspect encore légèrement granuleux de l'image.

Résolution utilisant la transformée de Fourier

Un algorithme plus rapide que le précédent consiste à utiliser la transformée de Fourier. En effet, on démontre que

$$\mathcal{F}(u)(p, q) = \frac{\mathcal{F}(f)(p, q)}{1 + 8\lambda (\sin^2 \frac{\pi p}{N} + \sin^2 \frac{\pi q}{N})} \quad (1.14)$$

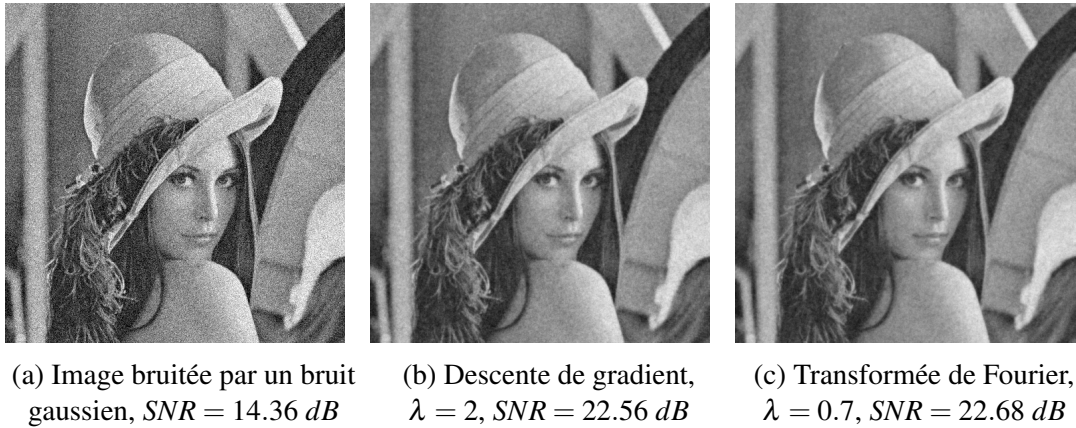


FIGURE 1.8 Comparaison des algorithmes de régularisation de Tychonov, $dt = 0.05$, $it = 300$

Il suffit donc de calculer des transformées de Fourier, moins coûteuses en temps de calcul. On prend soin de prolonger l'image par symétrie avant d'appliquer la transformée de Fourier.

Ainsi, on constate que le résultat est légèrement supérieur à celui obtenu par descente de gradient en ce qui concerne le $SNR = 22.68 \text{ dB}$. En revanche, le temps de calcul passe de 5.2s à 0.08s pour un résultat comparable (cf. Figure 1.8). Cette méthode est donc effectivement plus rapide que la descente de gradient. En ce qui concerne le lissage excessif attribué à la régularisation de Tykonov, les images obtenues semblent encore acceptables de ce point de vu.

1.2.2 Modèle de Rudin-Osher-Fatemi

Dans l'approche de Rudin- Osher-Fatemi, l'énergie à minimiser est la suivante, avec u l'image sans bruit, v le bruit et f l'image observée :

$$\inf_{(u,v)/f=u+v} \left(J(u) + \frac{1}{2\lambda} \|v\|_X^2 \right) \quad (1.15)$$

Ce problème a été résolu à l'aide de quatre algorithmes différents : un algorithme de descente de gradient, un algorithme de point fixe, un algorithme de gradient projeté et enfin l'algorithme de Nesterov. Ces quatre algorithmes sont présentés et comparés dans la suite de cette partie, en utilisant toujours l'image de Lena à laquelle on additionne un bruit gaussien.

Descente de gradient

L'algorithme du gradient s'écrit de façon similaire aux algorithmes du gradient abordés précédemment. Ainsi, il se base sur l'équation :

$$\frac{\partial u}{\partial t} = \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right) - \frac{1}{\lambda}(u - f) \quad (1.16)$$

Il est nécessaire d'ajouter un terme *epsilon* au dénominateur de notre premier terme de droite. En effet, dans le cas où $|\nabla u|$ tend vers zéro, l'équation peut devenir instable. L'ajout d'un terme résiduel permet d'assurer la convergence de l'algorithme. De plus on choisit dt de à assurer la convergence, i.e.

$$dt = \frac{2}{1 + 8\lambda/\varepsilon} \quad (1.17)$$

Après optimisation du choix de la valeur de λ , soit celle maximisant notre *SNR*, on obtient la Figure 1.9. Le résultat est comparable au niveau du *SNR* à la régularisation de Tychonov. Par contre, l'aspect n'est pas le même : les contours sont mieux définis.



FIGURE 1.9 Restauration (ROF - descente de gradient), $SNR = 22.68$, $\lambda = 1.5$, $it = 500$, $\varepsilon = 1$

Algorithme de projection de Chambolle

Une autre manière de résoudre le problème de Rudin-Osher-Fatemi consiste à appliquer l'algorithme de projection, proposé par Chambolle. Celui-ci consiste à calculer le projecteur de f sur λK , où

$$K = \{\operatorname{div}(g)/g \in Y, |g_{i,j}| \leq 1 \forall i, j\} \quad (1.18)$$

u est alors donnée par $u = f - P_{\lambda K}(f)$. Cette projection peut se résoudre en appliquant un algorithme du point fixe, à savoir

$$p^0 = 0 \quad (1.19)$$

$$p_{i,j}^{n+1} = \frac{p_{i,j}^n + \tau(\nabla(\operatorname{div}(p^n) - f/\lambda))_{i,j}}{1 + \tau|\nabla(\operatorname{div}(p^n) - f/\lambda))_{i,j}|} \quad (1.20)$$

et en calculant $u = f - \lambda \operatorname{div}(p^\infty)$.

L'application à notre image bruitée nous fournit les Figures 1.10.



(a) $\lambda = 0.1$, $SNR = 23.85 \text{ dB}$

(b) $\lambda = 0.25$, $SNR = 21.57 \text{ dB}$

FIGURE 1.10 Images restaurées par ROF - projection de Chambolle,
 $\tau = 1/8$, $it = 100$

Les images possèdent les caractéristiques des images restaurées par variation totale. En effet, elles sont fidèles en ce qui concerne les contours. Par contre les textures sont modifiées : elles apparaissent plus uniformes, pouvant donner l'impression d'une image de dessin animé (surtout dans le cas $\lambda = 0.25$).

Algorithme du gradient projeté

Une variante de l'algorithme de projection de Chambolle se base sur un algorithme de gradient projeté pour calculer la projection de f sur $P_{\lambda K}$. Ainsi,

$$\begin{cases} v^n &= f/\lambda - \operatorname{div} p^n \\ p_{i,j}^{n+1} &= \frac{p_{i,j}^n + \tau(\nabla(\operatorname{div}(p^n) - f/\lambda))_{i,j}}{\max\{1, |p_{i,j}^n + \tau(\nabla(\operatorname{div}(p^n) - f/\lambda))_{i,j}|\}} \end{cases} \quad (1.21)$$

Pour que l'algorithme converge, il faut prendre $\tau < 1/4$.

Le résultat de cet algorithme est similaire au précédent en terme de SNR et en terme d'aspect visuel, Figure 1.11.



FIGURE 1.11 Images restaurées par ROF - gradient projeté, $\tau = 1/8$, $it = 200$

Algorithme de Nesterov

L'algorithme de Nesterov est sensé procurer une solution plus rapide à notre problème ROF. La description de cet algorithme peut être retrouvée p.29 du poly sur les approches variationnelles. La Figure 1.12 présente l'image restaurée obtenue.



FIGURE 1.12 Restauration par algorithme de Nesterov, $SNR = 23.75$, $\lambda = 0.1$, $it = 200$

Pour résumer, voici Figure 1.13, l'ensemble des images restaurées à partir des algorithmes précédents. En ce qui concerne la vitesse de calcul de chaque algorithme, on peut dresser le tableau suivant :

Algorithme	n° d'itérations	Temps (s)	SNR (dB)
Descente de gradient	500	18.29	22.68
	200	8.0	22.67
Algorithme de Chambolle	100	2.53	23.85
	50	1.41	22.03
Gradient projeté	200	5.05	23.75
	100	2.56	23.76
	50	1.25	23.79
Algorithme de Nesterov	200	6.5	23.75
	100	2.90	23.72
	50	1.50	23.75

TABLE 1.1 Performances des algorithmes de résolution du problème ROF

L'algorithme le plus efficace avec des résultats très bons est l'algorithme du gradient projeté. Étonnamment, l'algorithme de Nesterov ne délivre pas de meilleures performances que les autres algorithmes. Si l'accent est mis sur la vitesse de calcul, l'algorithme de Chambolle est légèrement moins bon que l'algorithme du gradient projeté et de Nesterov. Ces derniers procurent de bons résultats en seulement 50 itérations et un temps de calcul de environ 1.5s



(a) Image bruitée



(b) Descente de gradient



(c) Projection de Chambolle



(d) Gradient projeté



(e) algorithme de Nesterov

FIGURE 1.13 Images restaurées par ROF

1.2.3 Déconvolution

Un autre problème qu'il est possible de résoudre est le suivant :

$$\inf_u \frac{1}{2\lambda} \|f - Au\|^2 + \int \sqrt{\varepsilon^2 + |\nabla u|^2} \quad (1.22)$$

Pour résoudre ce problème de déconvolution, il est possible de procéder à une descente de gradient, avec

$$u_{n+1} = u_n - \tau \left(1/\lambda A^*(Au - f) - \operatorname{div} \left(\frac{\nabla u}{\varepsilon^2 + |\nabla u|^2} \right) \right) \quad (1.23)$$

où A est un noyau gaussien.

En choisissant λ et σ de façon à maximiser le SNR , on arrive à une valeur $\lambda = 0.02$ et $\sigma = 0.5$. En ne procédant qu'à 50 itérations on obtient un SNR_{50} de 23.6 dB qui est légèrement plus élevé que le $SNR_{150} = 23.57$ dB, Figure 1.14.



FIGURE 1.14 Images déconvoluées, $\tau = 1/8, \lambda = 0.02, \varepsilon = 0.1$

L'image obtenue par déconvolution présente des courbures très marquées. Les différents éléments de l'image sont clairement séparés : Lena semble presque avoir été découpée de l'image. Plus σ est grand plus ce phénomène est prononcé.

1.2.4 Seuillage en ondelettes

La dernière méthode de restauration présentée dans ce TP se base sur l'utilisation d'ondelettes. En décomposant notre image en ondelettes et en effectuant un seuillage sur les coefficients d'ondelettes obtenus, il est possible de restaurer une image dégradée par un bruit blanc

gaussien. pour cela, notre seuil *thres* doit vérifier

$$thres \leq \sigma \sqrt{2 \log(N^2)} \quad (1.24)$$

avec σ la variance du bruit blanc gaussien, N la taille de l'image.

Nous considérons ici, un seuillage en ondelette doux, aussi appelé *wavelet soft thresholding*. Les ondelettes qui ont été testées sont les ondelettes de Daubechies (*db1*, *db2* et *db10*) et les Symlets (*sym2*, *sym10*). Les ondelettes n'étant pas invariantes par translation, des algorithmes invariants par translation ont été testés. Ceux-ci se basent sur le principe suivant : on décale l'image et on calcule les coefficients successifs des ondelettes. Ces coefficients sont alors moyennés afin de rendre le coefficient final invariant par translation et donc plus pertinent. Par contre, cet algorithme est bien entendu plus coûteux en temps. Les résultats sont exposés Figure 1.15 et Figure 1.16

On constate que la transformée en ondelette non invariante par translation introduit du bruit, notamment dans le cas de *sym10*. Ceci n'est plus le cas lorsqu'on s'assure que la transformée en ondelettes est invariante par translation. L'image apparaît alors plus nette, moins brouillée.

1.3 Conclusion

En définitive, nous avons pu découvrir l'impact de différents opérateurs lorsqu'ils sont appliqués à une image. Ce traitement d'images par approches variationnelles permet notamment d'extraire les contours d'une image, comme c'est le cas pour le détecteur de Hildrett-Marr, mais aussi de restaurer une image bruitée. La première équation qui peut être utilisée est l'équation de la chaleur, qui a le même effet que la convolution gaussienne. Pour restaurer une image, l'algorithme de régularisation de Tychonov est peut adapté car il lisse les contours. Le modèle de Rudin-Osher-Fatemi qui peut être résolu à l'aide de plusieurs algorithmes permet d'obtenir une restauration qui conserve mieux les contours, tout comme c'est le cas lorsqu'on applique une déconvolution. Enfin, il est possible de recourir à un seuillage doux basé sur différents types d'ondelettes.



(a) db1, $thres = 0.16$, $SNR = 20.97\text{ dB}$



(b) db2, $thres = 0.16$, $SNR = 21.76\text{ dB}$



(c) db10, $thres = 0.16$, $SNR = 21.78\text{ dB}$

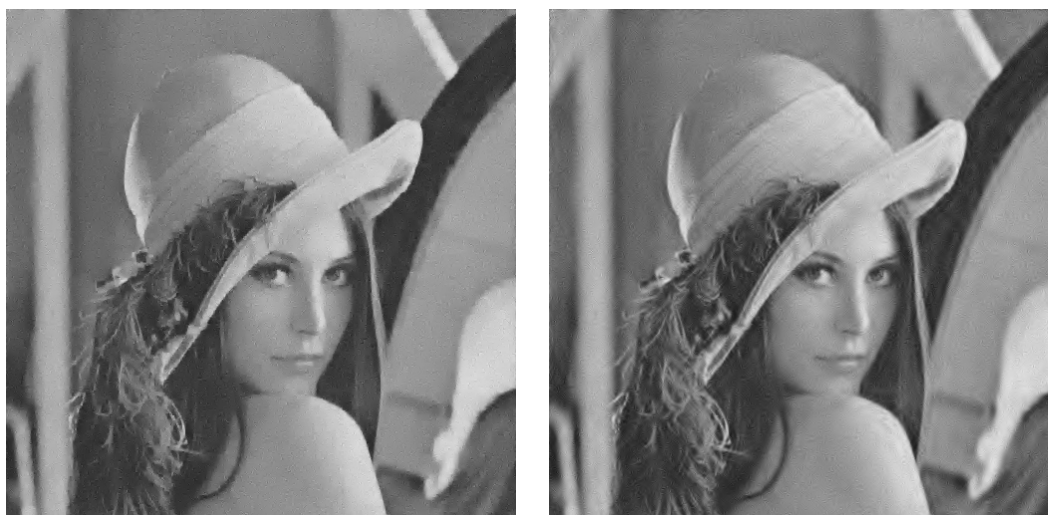


(d) sym2, $thres = 0.16$, $SNR = 21.76\text{ dB}$



(e) sym10, $thres = 0.17$, $SNR = 22.09\text{ dB}$

FIGURE 1.15 Images restaurées par transformée en ondelettes non invariantes par translation



(a) $db1, thres = 0.16, SNR = 22.39 \text{ dB}$

(b) $sym10, thres = 0.16, SNR = 22.77 \text{ dB}$

FIGURE 1.16 Images restaurées par transformée en ondelettes invariante par translation