



# Reconstruction d'images satellitaires à partir d'un échantillonnage irrégulier

Mikael Carlavan, Pierre Weiss, Laure Blanc-Féraud, Josiane Zerubia

## ► To cite this version:

Mikael Carlavan, Pierre Weiss, Laure Blanc-Féraud, Josiane Zerubia. Reconstruction d'images satellitaires à partir d'un échantillonnage irrégulier. [Rapport de recherche] 2008. inria-00340975v1

**HAL Id: inria-00340975**

**<https://inria.hal.science/inria-00340975v1>**

Submitted on 24 Nov 2008 (v1), last revised 28 Nov 2008 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# *Reconstruction d'images satellitaires à partir d'un échantillonnage irrégulier*

Mikael Carlván — Pierre Weiss — Laure Blanc-Féraud — Josiane Zerubia

**N° 6732**

Novembre 2008

Thème COG

 *apport  
de recherche*



## Reconstruction d'images satellitaires à partir d'un échantillonnage irrégulier

Mikael Carlván, Pierre Weiss, Laure Blanc-Féraud, Josiane Zerubia \*

Thème COG — Systèmes cognitifs  
Projet Ariana

Rapport de recherche n° 6732 — Novembre 2008 — 60 pages

**Résumé :** Le problème de la reconstruction d'images est celui d'estimer une image à variable continue à partir des observations discrètes. Depuis les travaux pionniers de Shannon, de nombreuses méthodes ont été proposées dans la littérature pour la reconstruction d'images à partir de données observées avec un échantillonnage choisi, qu'il soit régulier ou irrégulier. Nous nous sommes focalisés sur le problème global de la restauration d'une image satellitaire comprenant : la reconstruction d'une image régulièrement échantillonnée à partir d'une image irrégulièrement échantillonnée (les positions des échantillons sont supposées connues); la prise en compte de la fonction de flou de l'appareil optique (supposée connue); la prise en compte du bruit. Nous proposons une méthode par minimisation de fonctionnelle, dans laquelle nous utilisons une norme  $l^1$  pour le terme d'attache aux données, et la variation totale (VT) ou une transformée en ondelettes (Dual-Tree Complex Wavelet Transform) comme terme de lissage. Cette méthode est très rapide et converge en  $O\left(\frac{1}{k}\right)$  où  $k$  est le nombre d'itérations alors que la plupart des méthodes de minimisation existantes convergent  $O\left(\frac{1}{\sqrt{k}}\right)$ .

**Mots-clés :** Optimisation convexe, méthodes variationnelles, norme  $l^1$ , ondelettes, restauration d'image, déconvolution, imagerie satellitaire.

\* Les auteurs remercient l'entreprise CS-SI pour le financement partiel de ce travail de recherche, en particulier Anne Chanié et Rosario Ruiloba de CS-SI pour cette collaboration, et également le CNES pour toutes les données fournies.

# Satellite image reconstruction from irregular sampling

**Abstract:** The problem of image reconstruction consists in estimating an image in a continuous setting from discrete observations. Beginning with the pioneering work of Shannon, several methods have been proposed for image reconstruction from regularly or irregularly spaced discrete observations. We focused on irregular sampling image reconstruction which includes: regular sampling image reconstruction from irregularly spaced samples (we assume that the locations of samples are known); image deconvolution by taking into account the Point Spread Function of the optics (supposed to be known); restoration of the noise due to acquisition by the sensors. We propose a method minimizing a functional for which we use an  $l^1$ -norm for the data term, and Total Variation (TV) or a wavelet transform (Dual-Tree Complex Wavelet Transform), for the smoothing term. This method is very fast and has a convergence rate in  $O\left(\frac{1}{k}\right)$  where  $k$  is the iteration number while most minimization methods converge in  $O\left(\frac{1}{\sqrt{k}}\right)$ .

**Key-words:** Convex optimization, variational methods,  $l^1$ -norm, wavelets, image restoration, deconvolution, satellite imaging.

# Table des matières

<b>1</b>	<b>Présentation</b>	<b>5</b>
1.1	Position du problème . . . . .	5
1.2	Images utilisées . . . . .	6
1.2.1	Construction des images . . . . .	6
1.3	Approches variationnelles . . . . .	8
<b>2</b>	<b>Etat de l'art : algorithmes rapides</b>	<b>11</b>
2.1	Gradient composite . . . . .	11
2.2	Méthode à base de gradient . . . . .	12
2.3	Méthode accélérée à base de gradient . . . . .	14
<b>3</b>	<b>Solution proposée</b>	<b>17</b>
3.1	Résolution . . . . .	17
3.2	Réalisation . . . . .	19
3.3	Résultats . . . . .	21
<b>4</b>	<b>Régularisation dans le domaine des ondelettes</b>	<b>27</b>
4.1	Transformée en ondelettes complexes . . . . .	28
4.2	Mise en oeuvre . . . . .	31
4.3	Résultats . . . . .	32
<b>5</b>	<b>Comparaison dual/primal</b>	<b>33</b>
5.1	Modèle $BV - l_1$ . . . . .	33
5.2	Algorithme primal . . . . .	33
5.3	Algorithme dual . . . . .	35
5.4	Application au problème de reconstruction . . . . .	38
<b>6</b>	<b>Comparaison régularité/parcimonie</b>	<b>41</b>
6.1	Modèle de parcimonie . . . . .	41
6.2	Modèle de régularité . . . . .	42
6.3	Comparaison avec le modèle $BV - l^1$ . . . . .	43

**7 Conclusion et perspectives****49**

# Chapitre 1

## Présentation

Depuis maintenant plusieurs années, la technologie numérique a pris le monopole dans plusieurs domaines, en particulier celui de l'imagerie. Bien que le passage analogique/numérique (échantillonnage) induise une perte d'information, les avantages sont certains. Un des problèmes majeurs apparu avec la technologie numérique est celui de la reconstruction d'une image à partir des observations discrètes dont on dispose. Ces observations peuvent aussi bien être issues d'un échantillonnage régulier qu'irrégulier. Dans ce rapport, nous nous sommes intéressés au cas particulier de la reconstruction d'images satellitaires à partir d'un échantillonnage irrégulier.

### 1.1 Position du problème

Dans le cadre de la collaboration franco-italienne du programme ORFEO (programme permettant à la fois l'acquisition d'images optiques et radar), il est prévu du côté français de lancer deux satellites optiques haute résolution. Ces satellites, nommés Pléiades, et qui devraient être lancés d'ici 2010 pour le premier et en 2011 pour le second auront les propriétés suivantes :

- Longueurs d'onde d'acquisition : bleu, vert, rouge et proche infrarouge.
- Résolution : 0.7m.
- Fauchée (champ de vue) : 20km.

Par ailleurs, de manière à pouvoir répondre à des besoins de cartographie fine, notamment en zone urbaine et en complément de prises de vues aériennes, les satellites Pléiades devront permettre l'acquisition de paires stéréoscopiques quasi-instantanées ainsi que la réalisation de mosaïques d'images de tailles 120km par 120km. Enfin, leur mobilité permettra d'acquérir des images en tout point du globe en l'espace d'une journée.

Dans ce cadre d'acquisition, la reconstruction d'une image super-résolue à partir d'une paire stéréoscopique peut s'avérer intéressante. Si l'on considère deux prises de vue d'une même scène mais avec un angle différent, alors on peut considérer qu'une des deux images est



échantillonnée irrégulièrement par rapport à la première. Dans ce contexte, la reconstruction d'une image à partir de l'autre ou la fusion de la paire stéréoscopique peuvent être vues comme des problèmes de reconstruction à partir d'un échantillonnage irrégulier.

## 1.2 Images utilisées

Comme il a été décrit précédemment, les images utilisées sont des paires stéréoscopiques. La figure 1.1 montre un exemple d'une paire stéréoscopique ainsi que l'image des disparités d'une image à l'autre.



FIG. 1.1 – Exemple d'une paire stéréoscopique ©CNES (à gauche et au centre) et des disparités d'une image à l'autre (à droite). Plus un élément se sera déplacé entre les deux images et plus il sera sombre sur l'image des disparités. Les images n'étant pas prises au même moment, on remarque que les voitures ont bougé d'une image à l'autre. De même, l'ombre de l'immeuble en bas au centre est légèrement différente.

Dans le cadre de ce travail, le CNES a mis à notre disposition deux couples d'images : le couple d'images de la prison Saint-Michel à Toulouse prises à l'aide du capteur Pélican et celui de Marseille, simulation du satellite Pléiades. Ensuite pour chaque couple, la société CS nous a fourni une image des disparités entre les paires stéréos. Le paragraphe suivant décrit l'algorithme qui a été utilisé par CS.

### 1.2.1 Construction des images

Pour trouver les valeurs des disparités entre les paires stéréos, la société CS a utilisé l'algorithme MARC (*Multirésolution Algorithm for Refined Correlation*) développé par le CNES. Dans cet algorithme, le modèle utilisé est le suivant :

$$\tilde{u}_b(x) = u_b(x + \theta(x)) + g_b(x) \quad (1.1)$$

où  $u_b$  et  $u_b$  sont les paires stéréos et  $\theta(x)$  est un décalage en fonction de  $x$ . Dans ce modèle, les images sont considérées comme faiblement bruitées ( $g_b$  représente un bruit gaussien). La

valeur du déplacement est donné en maximisant le rapport de corrélation suivant :

$$\rho_{x_0}(m) = \frac{\int \varphi(x_0 - x) u_b(x + m) \tilde{u}_b(x) dx}{\sqrt{\int \varphi(x_0 - x) u_b^2(x + m) dx \int \varphi(x_0 - x) \tilde{u}_b^2(x) dx}} \quad (1.2)$$

Dans cette expression,  $\varphi$  est la fenêtre de corrélation, elle définit l'intervalle sur lequel la corrélation est calculée. Dans [7], une étude est faite sur le choix de taille et de forme optimal de cette fenêtre afin de limiter les artefacts dus au processus de stéréovision, comme par exemple l'adhérence (dilatation de certains objets autour de contours très marqués).

Dans cet algorithme, une hypothèse est faite sur les valeurs de  $\theta(x)$  qui ne doivent pas être trop fortes par rapport aux variations de  $u_b$ . Les points trouvés ne respectant pas cette hypothèse ne sont pas pris en compte. Pour résoudre ce problème, l'algorithme utilise une approche multi-échelle. Enfin, pour obtenir une précision subpixelique de la valeur des disparités, il est nécessaire d'obtenir une image continue lors du calcul du coefficient de corrélation. Cette étape est expliquée dans [7].

Une fois la valeur des disparités obtenue, il est possible d'utiliser deux images différentes pour le problème d'échantillonnage irrégulier qui consiste à reconstruire une image à partir de la deuxième et des disparités. La figure 1.2 montre la grille irrégulière obtenue en appliquant les disparités entre les deux images d'une paire stéréo à une grille régulière.

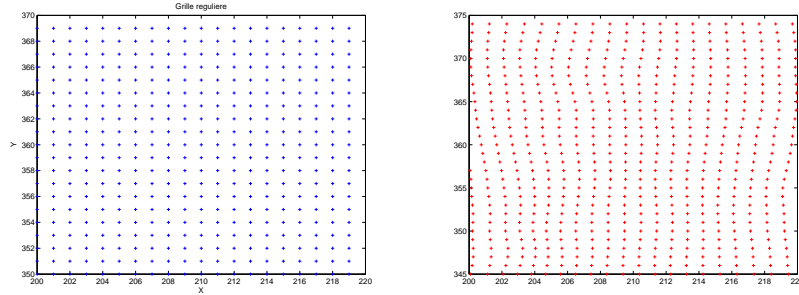


FIG. 1.2 – Comparaison d'une grille régulière de taille 20x20 et d'une grille irrégulière. Cette grille irrégulière est obtenue par application de la disparité entre les deux images d'une paire stéréo sur une petite zone.

Finalement, nous décrivons les images par le modèle suivant :

$$u = \Delta_\Lambda(h * I) + \varepsilon \quad (1.3)$$

avec :

$$\left\{ \begin{array}{l} I \text{ la scène observée,} \\ h \text{ la réponse impulsionnelle du système optique dont l'expression} \\ \quad \text{est fournie par le CNES (3.17),} \\ \Delta_{\Lambda} \text{ la grille irrégulière d'échantillonnage} \\ \quad \text{définie par } \Delta_{\Lambda}(\cdot) = \sum_{\lambda_k \in \Delta} \delta(\cdot - \lambda_k), \\ \quad \text{les } \lambda_k \text{ sont les positions des échantillons et } \delta \text{ représente la fonction Dirac,} \\ \quad \text{définie par } \delta(x) = \begin{cases} 1 & \text{si } x = 0, \\ 0 & \text{sinon} \end{cases} \\ \varepsilon \text{ un bruit dû au capteur optique du système et aux erreurs de disparités. Dans le cadre} \\ \quad \text{de notre application, nous considérons en priorité ces erreurs de disparités, nous} \\ \quad \text{modéliserons ce bruit comme étant impulsionnel (partie 1.3).} \end{array} \right. \quad (1.4)$$

### 1.3 Approches variationnelles

Il existe de nombreux algorithmes de reconstruction à partir d'un échantillonnage irrégulier (cf. [9, 11] par exemple). La plupart de ces algorithmes considèrent la grille irrégulière comme étant une grille régulière perturbée par un léger décalage  $\xi$  [14]. Néanmoins lorsque ce décalage devient plus important, ces algorithmes deviennent très approximatifs. Il est difficile de résoudre ce problème en essayant une inversion directe dans le domaine de Fourier car les fonctions de transfert des optiques sont des filtres passes-bas et s'annulent quand on s'approche de la fréquence de Nyquist. En conséquence, une inversion directe fait exploser le bruit contenu dans l'image même si celui-ci est très faible (voir figure 1.3). Nous formulons donc notre problème comme la recherche de la solution exacte  $u^*$  qui vérifie l'équation suivante :

$$g = \Delta_{\Lambda} \cdot (h * u^*) + \varepsilon \quad (1.5)$$

où  $g$  est l'observation, *i.e.* l'image échantillonnée irrégulièrement. Nous cherchons à minimiser  $\varepsilon$ , soit  $g - \Delta_{\Lambda} \cdot (h * u)$  ce qui est formulé par :

$$u^* = \arg \min_{u \in \mathbb{R}^N} \|\Delta_{\Lambda} \cdot (h * u) - g\|_p^p \quad (1.6)$$

où  $\|\cdot\|_p^p$  désigne une norme  $l^p$  à la puissance  $p$ . Le choix de  $p$  dépend du type de bruit à minimiser. Pour un bruit gaussien, un choix naturel est  $p = 2$ , pour un bruit de type impulsionnel on prendra  $p = 1$  [3]. Cependant, la formulation (1.6) (moindres carrés) fait également exploser le bruit. Il est alors nécessaire de régulariser les solutions présentes *via*, par exemple, une approche variationnelle :

$$u^* = \arg \min_{u \in \mathbb{R}^N} \underbrace{\|\Delta_{\Lambda} \cdot (h * u) - g\|_p^p}_{\text{Attache aux données}} + \underbrace{\lambda J(u)}_{\text{Régularisation}} \quad (1.7)$$

où  $J(u)$  est l'opérateur de régularisation qui est fonction de  $u$  et de ses dérivées,  $\lambda$  est le paramètre de régularisation. Certains opérateurs favorisent mieux la parcimonie (séparation bruit/signal) et permettent de reconstruire plus finement l'image tout en filtrant le bruit (cf. chapitre 4). Un opérateur très répandu en traitement d'image est  $J(u) = \|\nabla u\|_1$  [22]. Cette régularisation se nomme la Variation Totale (VT) et permet de conserver les contours de l'image mais lisse les textures [1] (effet "cartoon").

Les images représentant les disparités entre les couples d'images peuvent présenter des erreurs. Ces erreurs ont de grandes répercussions sur la reconstruction de l'image, en particulier sur le repositionnement des contours des objets. Ceci peut être vu comme du bruit impulsionnel sur l'intensité. Notre modèle doit donc être robuste à ce genre de bruit (une application possible après le rééchantillonnage pourrait être la fusion des images). De plus, nous souhaitons garder une certaine qualité de l'image et donc ne pas lisser les contours. Nous nous plaçons donc dans le cas  $p = 1$  car nous souhaitons minimiser un bruit impulsionnel et nous régularisons le modèle en utilisant la variation totale, on cherche donc :

$$\arg \min_{u \in \mathbb{R}^N} \|\Delta_{\Lambda} \cdot (h * u) - g\|_1 + \lambda \|\nabla u\|_1 \quad (1.8)$$

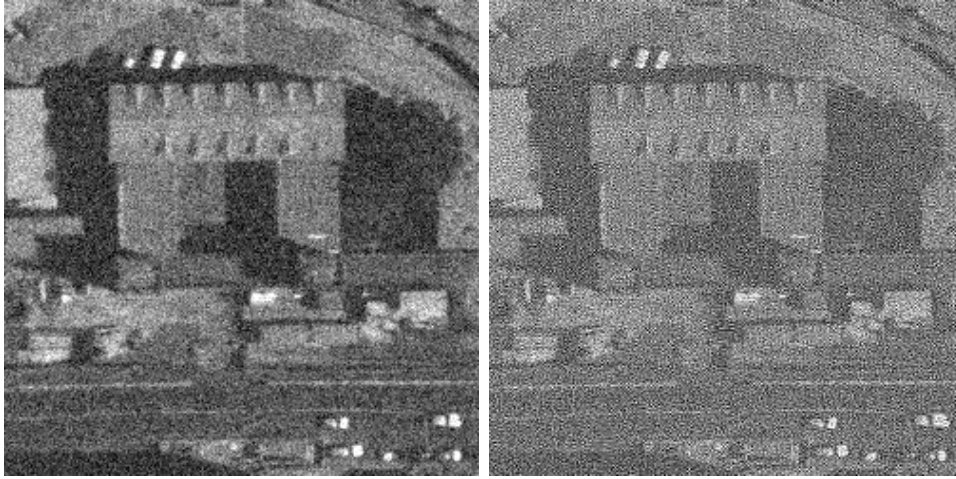


FIG. 1.3 – Inversion directe du système pour une image bruitée avec un bruit blanc gaussien ( $RSB = 3.58dB$ ). A gauche l'image de départ ©CNES, à droite le résultat après inversion directe.

Afin d'expliciter le terme  $\Delta_{\Lambda} \cdot (h * u)$ , nous passons dans le domaine de Fourier (on désigne par  $\mathcal{F}$  l'opérateur de transformée de Fourier discrète et  $\mathcal{F}^{-1}$  l'opérateur inverse) :

$$\arg \min_{u \in \mathbb{R}^N} \|\Delta_{\Lambda} \cdot (h * u) - g\|_1 + \lambda \|\nabla u\|_1 = \arg \min_{u \in \mathbb{R}^N} \|\mathcal{F}^{-1}(\mathcal{F}(\Delta_{\Lambda} \cdot (h * u))) - g\|_1 + \lambda \|\nabla u\|_1$$

$$= \arg \min_{u \in \mathbb{R}^N} \|\mathcal{F}^{-1}(\mathcal{F}(\Delta_\Lambda) * (\mathcal{F}(h)\mathcal{F}(u))) - g\|_1 + \lambda \|\nabla u\|_1 \quad (1.9)$$

En notant  $G$  la transformée de Fourier de  $\Delta_\Lambda$ , et en substituant le produit matriciel au produit de convolution, (1.9) peut se réécrire sous la forme :

$$\arg \min_{u \in \mathbb{R}^N} \|\mathcal{F}^{-1}(G\mathcal{F}(h)\mathcal{F}(u)) - g\|_1 + \lambda \|\nabla u\|_1 \quad (1.10)$$

L'expression de  $\mathcal{F}(h)$  est celle donnée par le CNES (3.17), on la notera  $H$ . Enfin, en remplaçant l'opérateur  $\mathcal{F}$  par son expression matricielle  $F$ , nous obtenons :

$$\arg \min_{u \in \mathbb{R}^N} \|F^{-1}(GHFu) - g\|_1 + \lambda \|\nabla u\|_1 \quad (1.11)$$

En notant  $S = F^{-1}G$ , (1.11) peut se réécrire sous la forme :

$$\arg \min_{u \in \mathbb{R}^N} \|SHFu - g\|_1 + \lambda \|\nabla u\|_1 \quad (1.12)$$

Avec  $F$  l'opérateur de transformée de Fourier,  $H$  la transformée de Fourier de la fonction de transfert de l'optique  $h$  et, enfin,  $S$  un opérateur permettant de passer d'une image complexe échantillonnée régulièrement à une image réelle échantillonnée irrégulièrement (expression donnée à la partie 3.2). Pour plus de simplicité, nous écrirons par la suite :

$$\arg \min_{u \in \mathbb{R}^N} \|Au - g\|_1 + \lambda \|\nabla u\|_1 \quad (1.13)$$

Cette modélisation est celle que E. Bughin avait utilisé dans [3]. Pour résoudre ce problème, il avait également utilisé une descente de gradient qui convergeait en  $O(\frac{1}{\sqrt{k}})$  (nous verrons dans le chapitre 3 comment contourner la non-dérivabilité de la norme  $l^1$ ). Ce taux de convergence n'est pas optimal, il est donc crucial de trouver des algorithmes rapides tels que ceux proposés dans [10] ou [25]. Dans [10], la méthode des points intérieurs est utilisée. Dans [25], une méthode est développée pour résoudre ces fonctionnelles à l'aide d'un algorithme proposé par Y. Nesterov [17]. Dans la suite de ce rapport, nous détaillons un algorithme encore plus récent et plus performant du même auteur [20].

## Chapitre 2

# Etat de l'art : algorithmes rapides

### 2.1 Gradient composite

Dans un premier temps, Y. Nesterov [20] cherche à déterminer le minimum local d'une fonction  $\phi$  définie par :

$$\phi(x) = f(x) + \Psi(x), \quad x \in Q \quad (2.1)$$

avec  $f$  continûment dérivable et convexe,  $\Psi$  une fonction convexe fermée (c'est à dire une fonction convexe dont l'épigraphe est un ensemble fermé) sur un ensemble  $Q$ . Une fonction dérivable  $f$  est fortement convexe sur  $Q$  de paramètre de convexité  $\nu \geq 0$  si  $\forall x, y \in Q$  :

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq \nu \|x - y\|_2^2 \quad (2.2)$$

Le cas  $\nu = 0$  correspond à une fonction dérivable convexe. Une hypothèse est faite sur  $\Psi$  qui doit être une fonction *simple* c'est à dire une fonction où  $\arg \min_{y \in Q} (\Psi(y) + \frac{1}{2} \|y - x\|_2^2)$  est calculable. La fonction  $f$  est continûment différentiable et son gradient est supposé  $L_f$ -lipschitzien :

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L_f \|x - y\|_2, \quad \forall x, y \in Q \quad (2.3)$$

Nous disposons du lemme fondamental suivant (A.48) :

$$f(x) \leq f(y) + \langle \nabla f(y), x - y \rangle + \frac{L}{2} \|x - y\|^2, \quad \forall x, y \in Q \quad (2.4)$$

avec  $0 \leq L \leq L_f$ . Y. Nesterov généralise ce lemme en y ajoutant la fonction  $\Psi$ , il obtient ainsi :

$$f(x) + \Psi(x) \leq f(y) + \langle \nabla f(y), x - y \rangle + \frac{L}{2} \|x - y\|^2 + \Psi(x) \quad (2.5)$$

D'où, grâce à (2.1) :

$$\phi(x) \leq f(y) + \langle \nabla f(y), x - y \rangle + \frac{L}{2} \|x - y\|^2 + \Psi(x) \quad (2.6)$$

La fonction  $\phi$  à minimiser est donc approchée par le terme de droite de l'expression précédente. Y. Nesterov utilise cette approximation pour approcher le minimum de la fonction  $\phi$ . La constante  $L$  nous dispense de connaître la valeur exacte de la constante de Lipschitz  $L_f$  (qui peut être difficile à déterminer), une simple estimation de cette constante est suffisante, les algorithmes la réajustent au fil des itérations (cf. parties 2.2 et 2.3). Y. Nesterov définit :

$$m_L(y; x) = f(y) + \langle \nabla f(y), x - y \rangle + \frac{L}{2} \|x - y\|^2 + \Psi(x) \quad (2.7)$$

$$T_L(y) = \arg \min_{x \in Q} m_L(y; x) \quad (2.8)$$

L'opérateur  $T_L(y)$  réalise une descente de gradient sur cette approximation, permettant ainsi de calculer le minimum local sur l'ensemble  $Q$ . Comme  $f$  est convexe, le minimum local sur  $Q$  devient un minimum global. Le premier algorithme [20] illustre ce gradient composite.

## 2.2 Méthode à base de gradient

Y. Nesterov donne l'algorithme suivant [20] :

---

### Algorithm 1 Méthode de gradient

---

```

Initialiser  $y_0$  et  $L_0 \in [0, L_f]$ 
 $M_0 = L_0$ 
for  $k \geq 0$  do
   $L = M_k$ 
  repeat
     $T = T_L(y_k)$ 
    if  $\phi(T) > m_L(y_k; T)$  then
       $L = L\gamma_u$ 
    end if
  until  $\phi(T) \leq m_L(y_k; T)$ 
   $y_k = T$ 
   $L = \max(L_0, L/\gamma_d)$ 
end for

```

---

avec  $\gamma_u \geq 1$  et  $\gamma_d \geq 1$ , les deux paramètres qui ajustent la valeur de  $L$  afin d'avoir la meilleure estimation. Y. Nesterov démontre [20] que cet algorithme converge en  $O\left(\frac{1}{k}\right)$ , il pose :

$$y_k(\alpha) = \alpha x^* + (1 - \alpha)y_k \in Q, \alpha \in [0, 1] \quad (2.9)$$

Alors :

$$\phi(y_{k+1}) \leq \phi(y_k) \leq m_{M_k}(y_k; T_L(y_k)) \leq \min_{y \in Q} \left[ \phi(y) + \frac{M_k}{2} \|y - y_k\|_2^2 \right] \quad (2.10)$$

$$\leq \min_{0 \leq \alpha \leq 1} \left[ \phi(\alpha x^* + (1 - \alpha)y_k) + \frac{M_k \alpha^2}{2} \|y_k - x^*\|_2^2 \right] \quad (2.11)$$

Il pose également :

$$R = \max_{y_k \in Q} \|y_k - x^*\|_2 \quad (2.12)$$

Alors :

$$\phi(y_{k+1}) \leq \min_{0 \leq \alpha \leq 1} \left[ \phi(y_k) - \alpha(\phi(y_k) - \phi(x^*)) + \frac{\gamma_u L_f \alpha^2}{2} R^2 \right] \quad (2.13)$$

Si  $\phi(y_0) - \phi(x^*) \geq \gamma_u L_f R^2$ , alors la solution optimale est  $\alpha^* = 1$  et donc :

$$\phi(y_1) - \phi(x^*) \leq \frac{\gamma_u L_f R^2}{2} \quad (2.14)$$

Sinon la solution optimale est  $\alpha^* = \frac{\phi(y_k) - \phi(x^*)}{\gamma_u L_f R^2} \leq \frac{\phi(y_0) - \phi(x^*)}{\gamma_u L_f R^2} \leq 1$  et donc :

$$\phi(y_{k+1}) \leq \phi(y_k) - \frac{[\phi(y_k) - \phi(x^*)]^2}{\gamma_u L_f R^2} \quad (2.15)$$

Il définit :

$$\lambda_k = \frac{1}{\phi(y_k) - \phi(x^*)} \quad (2.16)$$

En remplaçant cette équation dans (2.15), il obtient :

$$\lambda_{k+1} \geq \lambda_k + \frac{1}{2\gamma_u L_f R^2} \geq \lambda_0 + \frac{k+1}{2\gamma_u L_f R^2} \quad (2.17)$$

avec :

$$\lambda_0 = \frac{1}{\phi(y_0) - \phi(x^*)} \geq \frac{1}{\gamma_u L_f R^2} \quad (2.18)$$

Finalement, pour tout  $k \geq 0$  :

$$\phi(y_k) - \phi(x^*) \leq \frac{\gamma_u L_f R^2}{k+2} \quad (2.19)$$

De la même façon, si  $\phi$  est fortement convexe sur  $Q$  de paramètre de convexité  $\nu_\phi$  alors si  $\frac{\nu_\phi}{L_f} \geq 2\gamma_u$  :

$$\phi(y_k) - \phi(x^*) \leq \left( \frac{\gamma_u L_f}{\nu_\phi} \right)^k (\phi(y_0) - \phi(y^*)) \leq \frac{1}{2^k} (\phi(y_0) - \phi(y^*)) \quad \forall k \geq 0 \quad (2.20)$$

avec  $y^* = y_k(\alpha^*)$ . Sinon :

$$\phi(y_k) - \phi(x^*) \leq \left( 1 - \frac{\nu_\phi}{4\gamma_u L_f} \right)^k (\phi(y_0) - \phi(y^*)) \quad \forall k \geq 0 \quad (2.21)$$

Cet algorithme est similaire à celui proposé par P. L. Combettes [6] et converge également en  $O\left(\frac{1}{k}\right)$ . Cependant cet algorithme peut être accéléré [18] pour atteindre un taux de convergence en  $O\left(\frac{1}{k^2}\right)$ .



## 2.3 Méthode accélérée à base de gradient

Pour cette version accélérée [20] de l'algorithme, Y. Nesterov considère le problème suivant :

$$\min_{x \in E} (\phi(x) = f(x) + \Psi(x)) \quad (2.22)$$

où  $\Psi$  est une fonction fermée fortement convexe sur  $E$ . Cette formulation admet que le domaine de définition de  $\Psi$  soit différent de  $E$ , ceci permet de résoudre les problèmes sous contraintes. Afin d'améliorer le taux de convergence du précédent algorithme, Y. Nesterov utilise la technique de l'estimation de fonctions [19], il met à jour récursivement les séquences suivantes :

- la séquence de minimisation  $x_k$ .
- la séquence croissante des  $A_k = A_{k-1} + a_k, \forall k \geq 1, A_0 = 0$
- la séquence des fonctions estimées :

$$\psi_k(x) = l_k(x) + A_k \Psi(x) + \frac{1}{2} \|x - x_0\|_2^2, \quad \forall k \geq 0 \quad (2.23)$$

où  $x_0$  est la solution initiale, et  $l_k(x)$  sont les fonctions linéaires de  $E$ . Le point important de cet algorithme est qu'à chaque itération, les relations suivantes soient respectées :

$$\begin{cases} \mathcal{R}_k^1 : A_k \phi(x_k) \leq \psi_k^* = \min_{x \in E} \psi_k(x) \\ \mathcal{R}_k^2 : \psi_k(x) \leq A_k \phi(x) + \frac{1}{2} \|x - x_0\|_2^2, \forall x \in E \end{cases} \quad (2.24)$$

Ces deux relations justifient le taux de convergence de l'algorithme (2.26). Y. Nesterov propose l'algorithme suivant :

---

**Algorithm 2** Méthode de gradient accélérée

---

Initialiser  $\psi_0(x) = \frac{1}{2} \|x - x_0\|_2^2, A_0 = 0, L_0 \in [0, L_f], \nu \in [0, \nu_\Psi]$

**for**  $k \geq 0$  **do**

$L = L_k$

**repeat**

        Résoudre  $a$  de  $\frac{a^2}{A_k + a} = 2 \frac{1 + \nu A_k}{L}$

        Calculer  $y = \frac{A_k x_k + a v_k}{A_k + a}$ , puis  $T_L(y)$

**if**  $\langle \phi'(T_L(y)), y - T_L(y) \rangle < \frac{1}{L} \|\phi'(T_L(y))\|_2^2$  **then**

$L = L \gamma_u$

**end if**

**until**  $\langle \phi'(T_L(y)), y - T_L(y) \rangle \geq \frac{1}{L} \|\phi'(T_L(y))\|_2^2$

$y_k = y, M_k = L, a_{k+1} = a$

$L_{k+1} = M_k / \gamma_d, x_{k+1} = T_{M_k}(y_k), A_{k+1} = A_k + a_{k+1}$

$\psi_{k+1}(x) = \psi_k(x) + a_{k+1} [f(x_{k+1}) + \langle \nabla f(x_{k+1}), x - x_{k+1} \rangle + \Psi(x)]$

**end for**

---

avec  $\nu_\Psi$  le paramètre de convexité de la fonction  $\Psi$  et :

$$v_k = \arg \min_{x \in E} \psi_k(x) \quad (2.25)$$

Une itération de cet algorithme est deux fois plus longue que celle de l'algorithme présenté précédemment car le gradient de la fonction  $f$  est calculé deux fois, cependant il est démontré dans [20] que son taux de convergence est meilleur puisqu'il converge en  $O\left(\frac{1}{k^2}\right)$ . En effet, de part les relations  $\mathcal{R}_k^1$  et  $\mathcal{R}_k^2$  :

$$\begin{aligned} A_k \phi(x_k) &\leq \phi_k^* = \min_{x \in E} \phi_k(x) \leq \phi_k(x) \leq A_k \phi(x) + \frac{1}{2} \|x - x_0\|_2^2 \\ A_k(\phi(x_k) - \phi(x)) &\leq \frac{1}{2} \|x - x_0\|_2^2 \\ \phi(x_k) - \phi(x^*) &\leq \frac{1}{2A_k} \|x^* - x_0\|_2^2, \quad \forall k \geq 1 \end{aligned} \quad (2.26)$$

D'autre part, d'après l'algorithme :

$$L_k \leq M_k \leq \gamma_u L_f, \quad \forall k \geq 0 \quad (2.27)$$

$$A_{k+1} = A_k + a_{k+1} = \frac{M_k a_{k+1}^2}{2(1 + \nu A_k)}, \quad \forall k \geq 0 \quad (2.28)$$

De plus, pour  $\nu \geq 0$  :

$$\begin{aligned} A_{k+1} &\leq A_{k+1} (1 + \nu A_k) = \frac{M_k}{2} (A_{k+1} - A_k)^2 \\ &= \frac{M_k}{2} [A_{k+1}^{1/2} - A_k^{1/2}]^2 [A_{k+1}^{1/2} + A_k^{1/2}]^2 \leq \frac{M_k}{2} [A_{k+1}^{1/2} - A_k^{1/2}]^2 [2A_{k+1}^{1/2}]^2 \\ &\leq 2A_{k+1} M_k [A_{k+1}^{1/2} - A_k^{1/2}]^2 \leq 2A_{k+1} \gamma_u L_f [A_{k+1}^{1/2} - A_k^{1/2}]^2 \end{aligned} \quad (2.29)$$

D'où :

$$A_k \geq \frac{k^2}{2\gamma_u L_f}, \quad \forall k \geq 0 \quad (2.30)$$

De la même façon, si  $\nu > 0$ , alors :

$$\nu A_k A_{k+1} < A_{k+1} (1 + \nu A_k) \leq 2A_{k+1} \gamma_u L_f [A_{k+1}^{1/2} - A_k^{1/2}]^2 \quad (2.31)$$

D'où :

$$A_k \geq A_k^{1/2} \left[ 1 + \sqrt{\frac{\nu}{2\gamma_u L_f}} \right] \quad (2.32)$$

En tenant compte du fait que  $A_0 = \frac{1}{M_0} \geq \frac{1}{\gamma_u L_f}$ , l'équation (2.32) peut se réécrire comme :

$$A_k \geq \frac{1}{\gamma_u L_f} \left[ 1 + \sqrt{\frac{\nu}{2\gamma_u L_f}} \right]^{2(k-1)}, \quad \forall k \geq 1 \quad (2.33)$$

En conséquence, Y. Nesterov obtient le taux de convergence suivant :

$$\phi(x_k) - \phi(x^*) \leq \frac{\gamma_u L_f \|x^* - x_0\|_2^2}{k^2}, \quad \forall k \geq 1 \quad (2.34)$$

De plus, si  $\Psi$  est fortement convexe ( $\nu > 0$ ), alors le taux de convergence satisfait à la fois l'équation précédente et celle-ci :

$$\phi(x_k) - \phi(x^*) \leq \frac{\gamma_u L_f}{2} \|x^* - x_0\|_2^2 \left[ 1 + \sqrt{\frac{\nu}{2\gamma_u L_f}} \right]^{-2(k-1)}, \quad \forall k \geq 1 \quad (2.35)$$

## Chapitre 3

# Solution proposée

Cette partie se consacre à la mise en oeuvre de la version accélérée de l'algorithme de Y. Nesterov pour le problème de restauration.

### 3.1 Résolution

Le problème est le suivant :

$$\arg \min_{u \in \mathbb{R}^N} \|Au - g\|_1 + \lambda \|\nabla u\|_1 \quad (3.1)$$

Pour mettre en oeuvre au mieux cet algorithme, il est nécessaire de correctement choisir quels termes représenteront les fonctions  $f(u)$  et  $\Psi(u)$ . La fonction  $\Psi$  doit être simple et convexe. Cependant dans notre problème, aucun des deux termes n'est une fonction simple et convexe. Nous pourrions rajouter un terme du type  $\frac{\gamma}{2}\|u - g\|^2$ , mais ceci modifierait sensiblement le modèle. Or il apparait que la fonction  $\Psi$  n'est pas indispensable pour obtenir le taux de convergence annoncé par Y. Nesterov pour son algorithme. Nous prenons donc  $\Psi(u) = 0$  (son paramètre de convexité vaut ainsi  $\nu_\Psi = 0$ ). En conséquence, nous prenons  $f(u) = \|Au - g\|_1 + \lambda \|\nabla u\|_1$ . Cette fonction est bien convexe, mais non-dérivable car la norme  $l^1$  n'est pas dérivable en 0. Nous allons donc la régulariser de la façon suivante :

$$\|u\|_{1,\mu} = \sum_{i=1}^N \sqrt{u_i^2 + \mu^2} \quad (3.2)$$

Finalement, nous avons :

$$f(u) = \|Au - g\|_{1,\mu} + \lambda J_\mu(u) \quad (3.3)$$

$$\Psi(u) = 0 \quad (3.4)$$

avec  $J_\mu(u) = \|\nabla u\|_{1,\mu}$ . La suite des calculs est donnée dans l'annexe A, nous obtenons :

$$T_L(y) = y - \frac{1}{L} \nabla f(y) \quad (3.5)$$

$$\nabla f(u) = A^* \left( \frac{(Au - g)}{\sqrt{(Au - g)^2 + \mu^2}} \right) - \lambda \operatorname{div} \left( \frac{\nabla u}{\sqrt{|\nabla u|^2 + \mu^2}} \right) \quad (3.6)$$

$$v_k = x_0 - \sum_{i=1}^k a_i \nabla f(x_i) \quad (3.7)$$

$$a = \frac{1}{L} + \sqrt{\left(\frac{1}{L}\right)^2 + 2A_k \frac{1}{L}} \quad (3.8)$$

$$L_f = \frac{\|A\|_2^2 + \lambda \|\operatorname{div}\|_2^2}{\mu} \quad (3.9)$$

Pour plus de simplicité, nous définissons :  $w_k = \sum_{i=1}^k a_i \nabla f(x_i)$ . Ainsi :

$$w_{k+1} = w_k + a_{k+1} \nabla f(x_{k+1}), \quad \forall k \geq 0, \quad w_0 = 0 \quad (3.10)$$

(3.7) devient :

$$v_k = x_0 - w_k \quad (3.11)$$

Nous connaissons la valeur de la constante de Lipschitz  $L_f$ , il n'est donc pas nécessaire de l'estimer, l'algorithme se résume ainsi à :

---

**Algorithm 3** Méthode de gradient accélérée

---

Initialiser  $A_0 = 0, w_0 = 0$

**for**  $k \geq 0$  **do**

$$a = \frac{1}{L_f} + \sqrt{\left(\frac{1}{L_f}\right)^2 + \frac{2}{L_f} A_k}$$

$$v_k = x_0 - w_k$$

$$y_k = \frac{A_k x_k + a v_k}{A_k + a}$$

$$x_{k+1} = y_k - \frac{1}{L_f} \nabla f(y_k)$$

$$w_{k+1} = w_k + a \nabla f(x_{k+1})$$

$$A_{k+1} = A_k + a$$

**end for**

---

avec  $x_0$  la solution initiale à notre problème. La régularisation de la norme  $l^1$  nous permet seulement d'obtenir un taux de convergence en  $O(\frac{1}{k})$ . Cependant si notre objectif est de minimiser le problème régularisé, alors nous obtenons bien un taux de convergence en  $O(\frac{1}{k^2})$ . Pour certains problèmes, le choix de la fonction  $\Psi$  peut, en plus, apporter une accélération non négligeable à l'algorithme. Pour un problème de type  $l^2 - l^1$ , il est avantageux de suivre l'idée de Y. Nesterov [20]. Il suggère de réallouer le terme  $\Psi$  au sein de la fonction  $f$ . En effet, il propose de choisir un terme fortement convexe pour la fonction  $\Psi$ , typiquement  $\Psi(u) = \frac{\gamma}{2} \|u - g\|^2$  (de paramètre de convexité  $\gamma$ ) et de poser :

$$\hat{f}(u) = f(u) - \Psi(u) = f(u) - \frac{\gamma}{2} \|u - g\|^2 \quad (3.12)$$

Sous la condition que le paramètre de convexité de la fonction  $\Psi$  soit inférieur à celui de  $f$ , la fonction  $\hat{f}$  est convexe. Cette méthode permet, en outre, de minimiser exactement notre problème. En effet :

$$\min_{u \in E} \phi(u) = \min_{u \in E} [\hat{f}(u) + \Psi(u)] = \min_{u \in E} [f(u) - \Psi(u) + \Psi(u)] = \min_{u \in E} f(u) \quad (3.13)$$

L'algorithme est de plus nettement plus rapide que si nous avions simplement choisi  $\Psi(u) = 0$ .

## 3.2 Réalisation

Pour la mise en oeuvre de cet algorithme nous utilisons la librairie ITK qui fournit en outre, un algorithme de FFT (VNL), algorithme non optimal, mais sous licence libre. Les opérateurs  $\nabla$  et  $\text{div}$  seront programmés en utilisant les schémas numériques proposés par A. Chambolle [4] :

$$\nabla_x u(i, j) = \begin{cases} u(i+1, j) - u(i, j) & \text{si } i < N, \\ 0 & \text{si } i = N \end{cases} \quad (3.14)$$

$$\nabla_y u(i, j) = \begin{cases} u(i, j+1) - u(i, j) & \text{si } j < N, \\ 0 & \text{si } j = N \end{cases} \quad (3.15)$$

$$\text{div } v = \begin{cases} v_x(i, j) - v_x(i-1, j) & \text{si } 1 < i < N \\ v_x(i, j) & \text{si } i = 1 \\ -v_x(i-1, j) & \text{si } i = N \end{cases} + \begin{cases} v_y(i, j) - v_y(i, j-1) & \text{si } 1 < j < N \\ v_y(i, j) & \text{si } j = 1 \\ -v_y(i, j-1) & \text{si } j = N \end{cases} \quad (3.16)$$

L'expression de la fonction de transfert de l'optique est celle fournie par le CNES :

$$H(f_x, f_y) = e^{-\alpha \sqrt{f_x^2 + f_y^2}} e^{-\beta f_y} \text{sinc}(\pi a f_x) \text{sinc}(\pi a f_y) \quad (3.17)$$

Pour calculer l'opérateur de transformée de Fourier inverse irrégulière ainsi que son adjoint, nous avons conservé la méthode employée par E. Bughin [3] qui avait choisi d'utiliser l'USFFT (*Unequally Spaced Fast Fourier Transform*) proposée par G. Beylkin [2]. Dans son papier, G. Beylkin démontre que la transformée de Fourier d'une image peut être vue comme une interpolation spline d'ordre  $m$ . En effet, une transformée inverse de Fourier s'écrit de la façon suivante :

$$u(\lambda_j) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} a_{kl} e^{\frac{i2\pi\lambda_{j,1}k}{N}} e^{\frac{i2\pi\lambda_{j,2}l}{N}} \quad (3.18)$$

avec  $a_{kl}$  les coefficients de Fourier. Ceci est une interpolation classique sinus cardinal, qui peut se réécrire sous la forme :

$$u(\lambda_j) = \sum_{k=0}^{2N-1} \sum_{l=0}^{2N-1} w_{kl} e^{\frac{i2\pi\lambda_{j,1}k}{N}} e^{\frac{i2\pi\lambda_{j,2}l}{N}} \quad (3.19)$$

avec :

$$w_{kl} = \begin{cases} a_{kl} & \text{pour } (k, l) \in [0, N/2 - 1]^2, \\ a_{(k-N)l} & \text{pour } k \in [N + N/2, 2N - 1], l \in [0, N/2 - 1], \\ a_{k(l-N)} & \text{pour } k \in [0, N/2 - 1], l \in [N + N/2, 2N - 1], \\ a_{(k-N)(l-N)} & \text{pour } k \in [N + N/2, 2N - 1], l \in [N + N/2, 2N - 1], \\ 0 & \text{sinon} \end{cases} \quad (3.20)$$

Cette étape correspond à un suréchantillonnage par mise à zéro des hautes fréquences (*zero padding*) pour le cas d'une transformée de Fourier non centrée (dans ce cas, les hautes fréquences sont au milieu de l'image). G. Beylkin [2] cherche ensuite les coefficients  $v_{kl}$  de l'interpolation spline tels que :

$$u(\lambda_j) = \sum_{k=0}^{2N-1} \sum_{l=0}^{2N-1} v_{kl} \beta^{(m)}(2 * \lambda_{j,1} - k) \beta^{(m)}(2 * \lambda_{j,2} - l) \quad (3.21)$$

Nous avons la relation suivante :

$$\sum_{k_1=0}^{2N-1} \sum_{l_1=0}^{2N-1} w_{k_1 l_1} e^{\frac{i 2\pi \lambda_{j,1} k_1}{N}} e^{\frac{i 2\pi \lambda_{j,2} l_1}{N}} = \sum_{k_2=0}^{2N-1} \sum_{l_2=0}^{2N-1} v_{k_2 l_2} \beta^{(m)}(2 * \lambda_{j,1} - k_2) \beta^{(m)}(2 * \lambda_{j,2} - l_2) \quad (3.22)$$

La seconde partie de l'équation correspond à une convolution de la fonction  $v$  par une spline. En passant (3.22) dans l'espace de Fourier, nous obtenons la relation suivante :

$$\widehat{V}_{kl} = \frac{w_{kl}}{\widehat{b}_{kl}} \quad (3.23)$$

Où  $\widehat{V}$  est la transformée de Fourier de  $v$ , et  $\widehat{b}$  est la transformée de Fourier de  $\beta^{(m)}$ . L'algorithme est alors le suivant :

- Partir d'une image régulière dans l'espace de Fourier.
- Faire un suréchantillonnage en ajoutant des 0 dans les hautes fréquences. Nous obtenons alors des coefficients de Fourier  $w_{kl}$ .
- Diviser les coefficients  $w_{kl}$  par les coefficients de Fourier d'une fonction spline d'ordre  $m$ .
- Repasser dans l'espace réel pour obtenir une image suréchantillonnée et déconvoluée par une fonction spline. Ceci nous donne une image  $v$ .
- Calculer l'interpolation spline de  $v$  aux points irréguliers  $\lambda_j$ . Nous obtenons alors une approximation de  $u$  aux points  $\lambda_j$ .

De la même façon l'algorithme pour le calcul de l'adjoint est le suivant :

- Calculer l'interpolation spline pour obtenir une image suréchantillonnée dans l'espace réel.
- Prendre la transformée de Fourier.
- Faire une déconvolution des splines dans l'espace de Fourier.
- Couper les hautes fréquences pour obtenir une image de la taille désirée.

### 3.3 Résultats

Cet algorithme a été testé sur deux images mises à disposition par le CNES : l'image de la prison Saint-Michel à Toulouse prise à l'aide du capteur Pélican (de taille 1000 par 1000 pixels) et celle de Marseille (de taille 972 par 972 pixels), simulations du satellite Pléiades (cf. figure 3.1).



FIG. 3.1 – Images de départ (de gauche à droite) : Toulouse et Marseille (échantillonnages irréguliers), ©CNES.

Le résultat sur la figure 3.2 a été obtenu en 110 itérations, calculé en 15 minutes (cf. figure 3.3). La méthode non accélérée, avec les mêmes paramètres, atteint le même résultat après 2h30 de calculs (2300 itérations). La qualité du rééchantillonnage est très bonne puisque les seules différences sont liées aux objets qui se sont déplacés durant les deux prises de vues (cf. figure 3.2). En terme de performances, par rapport à la descente de gradient, nous gagnons plus d'un facteur 20 sur le nombre d'itérations. Cependant, l'itération est plus longue dans notre cas, car le gradient de la fonction  $f$  doit être calculé une fois de plus, et donc le gain de temps de calcul est voisin de 10, ce qui est déjà important. Nous allons maintenant étudier plus en détails l'influence des paramètres sur la méthode proposée.



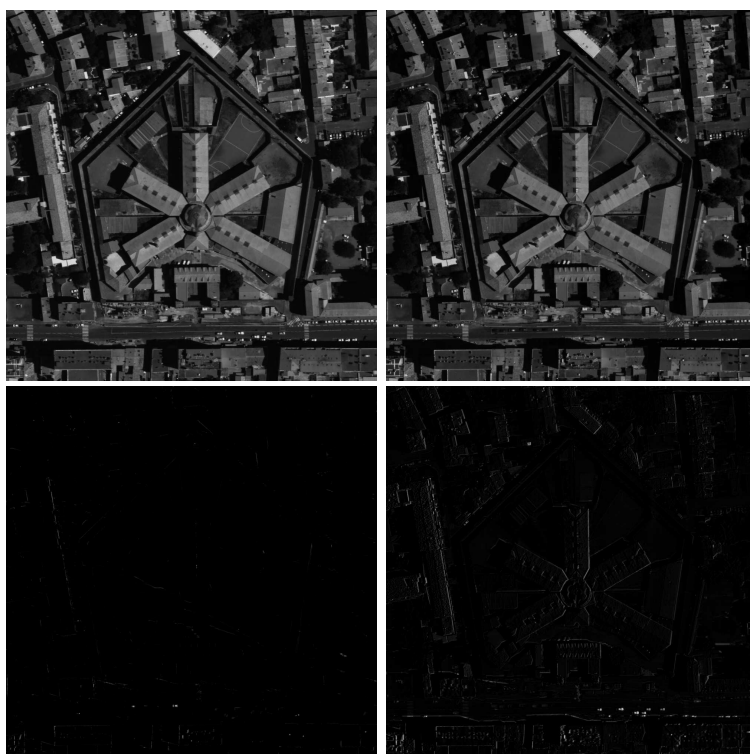


FIG. 3.2 – De gauche à droite et de haut en bas : l'image obtenue, l'image à obtenir, la différence des deux images et enfin la différence de l'image à obtenir avec l'image de départ (i.e avant rééchantillonnage).

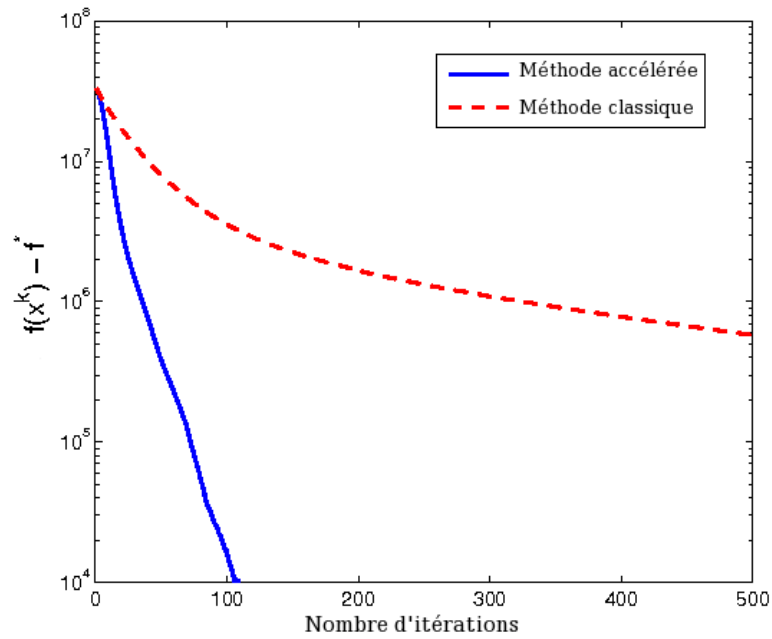


FIG. 3.3 – Comparaison des algorithmes (algorithme de Nesterov version accélérée et méthodes classiques) sur une image de taille 1000x1000 pixels.

### Influence du paramètre $\lambda$ :

Le paramètre  $\lambda$  est le paramètre de régularisation de la variation totale. Ce terme est donc lié au filtrage de l'image considérée. C'est pour cela qu'il faut le choisir en fonction de la puissance de bruit contenu dans l'image. Les différents tests représentés sur la figure 3.4 montrent le résultat de l'algorithme en fonction de  $\lambda$  pour un bruit gaussien, bien que notre modèle ne soit pas très robuste à ce genre de bruit, pour lequel une norme  $l^2$  sur le terme d'attache aux données serait plus appropriée.



FIG. 3.4 – Image obtenue pour une image bruitée avec un bruit blanc gaussien ( $RSB = 4.80$  dB) selon différentes valeurs de  $\lambda$  (de gauche à droite :  $\lambda = 0.1$ ,  $\lambda = 0.7$ ,  $\lambda = 1$ ).

Comme on peut le voir, une forte valeur de  $\lambda$  filtre correctement l'image mais la lisse aussi énormément. Pour éviter cela, et en tenant compte du fait que nos images sont faiblement bruitées, nous prendrons  $\lambda = 0.3$ . De plus, nous avons préféré être efficace sur le bruit impulsionnel, de façon à atténuer les erreurs sur les images des disparités, ce qui justifie l'utilisation de la norme  $l^1$ . La figure 3.5 montre l'efficacité de la norme  $l^1$  sur un bruit impulsionnel de type "poivre & sel" de probabilité 10%. En effet, utiliser une norme  $l^2$  pour ce genre de bruit oblige à fortement lisser l'image (forte valeur de  $\lambda$ ), alors que la norme  $l^1$  a complètement supprimé le bruit.

### Influence du paramètre $\mu$ :

Le paramètre  $\mu$  est le terme qui régularise la norme  $l^1$ . Une valeur trop importante de  $\mu$  faussera cette régularisation ce qui se traduit par un effet de flou sur l'image comme le montre la figure 3.6. Cependant ceci n'est visible que lorsque le paramètre  $\lambda$  est important. La vitesse de convergence de l'algorithme étant en  $\frac{1}{\mu}$  (figure 3.7), il est plutôt avantageux de prendre un  $\mu$  assez grand, les images étant faiblement bruitées, nous prendrons un  $\lambda$  faible et cet effet ne sera pas présent. Néanmoins cette valeur de  $\mu$  dépend aussi de la dynamique des images. L'image de Toulouse possède une dynamique beaucoup plus grande que celle de

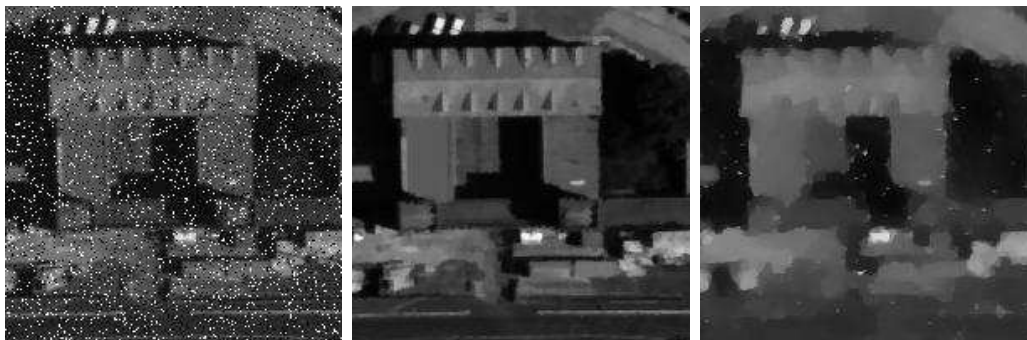


FIG. 3.5 – Rééchantillonnage et filtrage de bruit de type “poivre & sel” (de gauche à droite : image d’origine bruitée, résultat de la norme  $l^1$ , résultat de la norme  $l^2$ ).

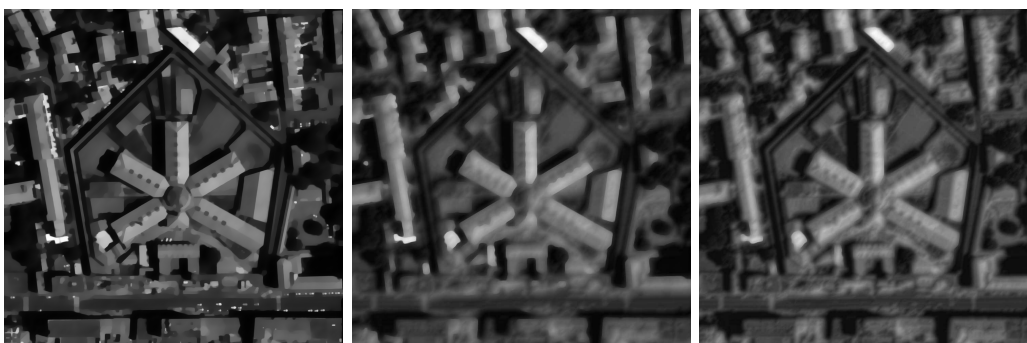


FIG. 3.6 – Effet de flou selon la valeur de  $\mu$  (de gauche à droite :  $\mu = 1$ ,  $\mu = 5$ ,  $\mu = 10$  pour  $\lambda = 10$ ).

Marseille. En conséquence, nous adapterons la valeur de  $\mu$  en fonction de l'image à traiter : on prendra  $\mu = 2$  pour l'image de Marseille et  $\mu = 10$  pour celle de Toulouse.

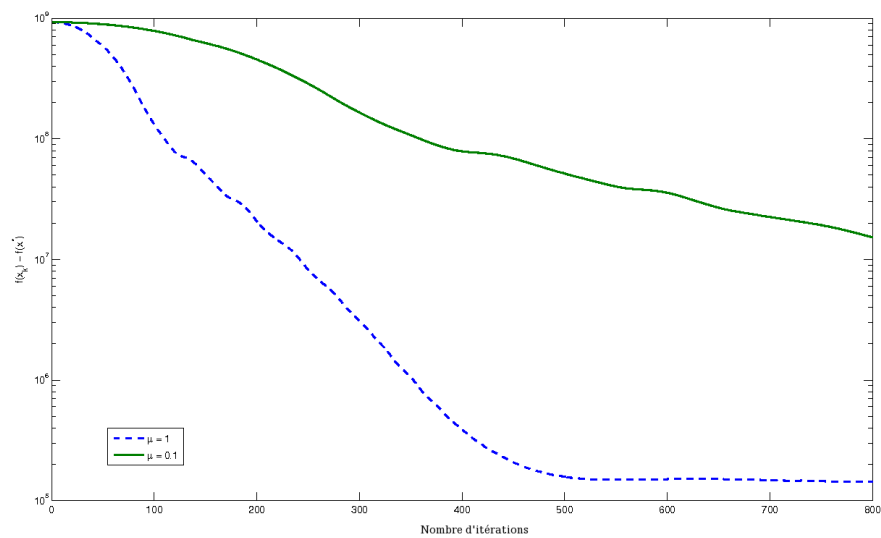


FIG. 3.7 – Convergence de l'algorithme en fonction de  $\mu$  sur une image de taille 1000x1000 pixels.

## Chapitre 4

# Régularisation dans le domaine des ondelettes

Nous avons vu précédemment que la variation totale était efficace sur les zones homogènes de l'image ainsi que sur les contours mais qu'elle se comportait mal sur les textures. Pour résoudre ce problème, nous allons effectuer une régularisation dans le domaine des ondelettes. Les ondelettes permettent de mieux représenter les zones hétérogènes (et conservent ainsi le détail des textures) que la variation totale. D'autre part, elles offrent une meilleure parcimonie (séparation bruit/signal), cette régularisation sera également plus efficace pour le filtrage du bruit gaussien. Nous reformulons le problème de la façon suivante :

$$\arg \min_{u \in \mathbb{R}^N} \|Au - g\|_1 + \lambda \|\Upsilon u\|_1 \quad (4.1)$$

avec  $\Upsilon$  la transformation en ondelettes. Cette modélisation n'est pas courante et relativement récente [23]. En effet, pour ce problème, une modélisation nettement plus répandue dans la littérature est une modélisation de parcimonie [8] :

$$\arg \min_{y \in \mathbb{R}^N} \|A\Upsilon^{-1}y - g\|_1 + \lambda \|y\|_1 \quad (4.2)$$

où  $y$  est le vecteur des coefficients de la transformée en ondelettes et  $\Upsilon^{-1}$  la transformée en ondelettes inverse. Dans le premier modèle (modèle de régularisation), la minimisation est faite directement sur le signal tandis que pour le modèle de parcimonie, la minimisation est faite sur les coefficients de sa transformée en ondelettes. Le chapitre 6 compare ces deux modèles.

Les ondelettes classiques ne présentant pas d'invariance par translation et rotation, nous avons choisi d'utiliser la "Dual-Tree Complex Wavelet Transform" (Transformée en Ondelettes Complexes) de N. G. Kingsbury [12].

## 4.1 Transformée en ondelettes complexes

N. G. Kingsbury a introduit dans [12] une transformée en ondelettes complexes qui permet de résoudre les problèmes d'oscillations, d'aliasing et d'invariance par translation et rotation des transformations en ondelettes réelles classiques. Il est important de noter que la transformée de Fourier ne souffre pas de ces problèmes, car, à l'inverse des transformées en ondelettes réelles, elle est basée sur des sinusoides complexes :

$$e^{j\Omega t} = \cos(\Omega t) + j \sin(\Omega t) \quad (4.3)$$

Ces deux composantes sinusoidales forment une paire de transformation de Hilbert (c'est à dire qu'elles sont déphasées de  $90^\circ$  l'une de l'autre), ensemble elles forment un signal analytique  $e^{j\Omega t}$ , qui présente de bonnes propriétés pour le traitement du signal (dont une représentation sur une seule moitié des fréquences :  $\Omega \geq 0$ ). En s'inspirant de ce modèle, N. G. Kingsbury construit une ondelette complexe à partir de deux ondelettes réelles :

$$\psi_c(t) = \psi_r(t) + j\psi_i(t) \quad (4.4)$$

Par analogie avec l'équation (4.3),  $\psi_r(t)$  est réelle et paire et  $\psi_i(t)$  est réelle et impaire. De plus, si  $\psi_r(t)$  et  $\psi_i(t)$  forment une paire de transformation de Hilbert, alors  $\psi_c(t)$  est un signal analytique.

Une approche pour obtenir cette ondelette analytique est d'utiliser deux transformations en ondelettes réelles, la première donne la partie réelle de la transformation, tandis que la deuxième donne la partie imaginaire, introduisant ainsi une redondance d'un facteur 2 (4 en 2D).

Les bancs de filtres utilisés pour cette transformation sont représentés sur la figure 4.1. La transformation inverse est donnée sur la figure 4.2. Chacune des deux transformations en ondelettes réelles utilise deux séries de filtres différents, qui doivent également être différents des filtres du premier niveau pour que la transformation en ondelettes soit analytique. Soient  $h^e(n)$  et  $g^e(n)$  les filtres passe-bas et passe-haut du banc de filtres supérieur,  $h^o(n)$  et  $g^o(n)$  les filtres passe-bas et passe-haut du banc de filtres inférieur, nous avons :

$$\psi(t) = \psi_{g^e}(t) + j\psi_{g^o}(t) \quad (4.5)$$

Ainsi  $\psi_{g^e}(t)$  est paire tandis que  $\psi_{g^o}(t)$  est impaire. Les coefficients de l'arbre supérieur sont donc sous-échantillonnés de manière à ne garder que les coefficients pairs tandis que l'arbre inférieur conserve les coefficients impairs. Les filtres du premier niveau sont identiques ( $h^e(n) = h^o(n)$  et  $g^e(n) = g^o(n)$ ), les deux arbres sont ainsi décalés d'un échantillon à ce niveau), tous les coefficients sont donc conservés, et on observe une invariance parfaite au premier niveau.

Les filtres assurent que  $\psi(t)$  soit analytique, autrement dit que  $\psi_{g^o}(t)$  soit la transformation de Hilbert de  $\psi_{g^e}(t)$ .

$$\psi_{g^o}(t) \approx \mathcal{H}\{\psi_{g^e}(t)\} \quad (4.6)$$

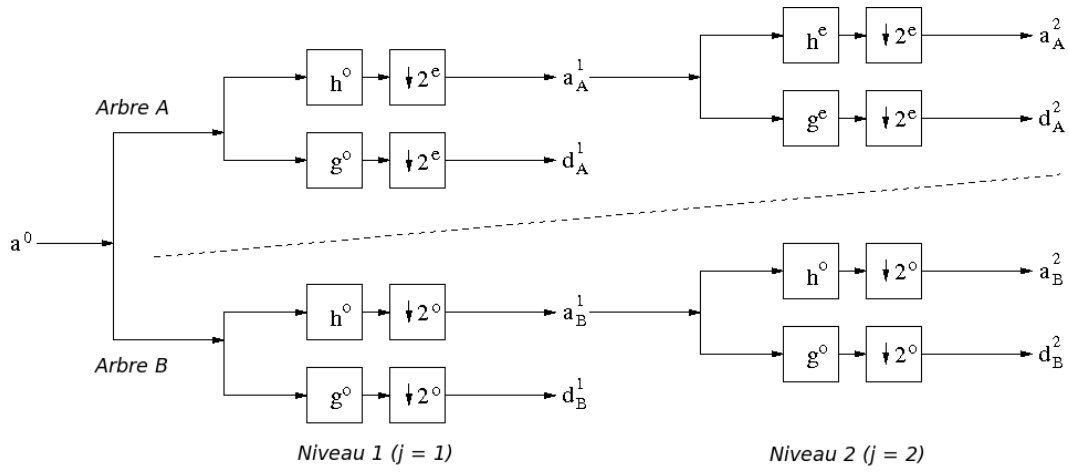


FIG. 4.1 – Transformée en ondelettes complexes 1D.

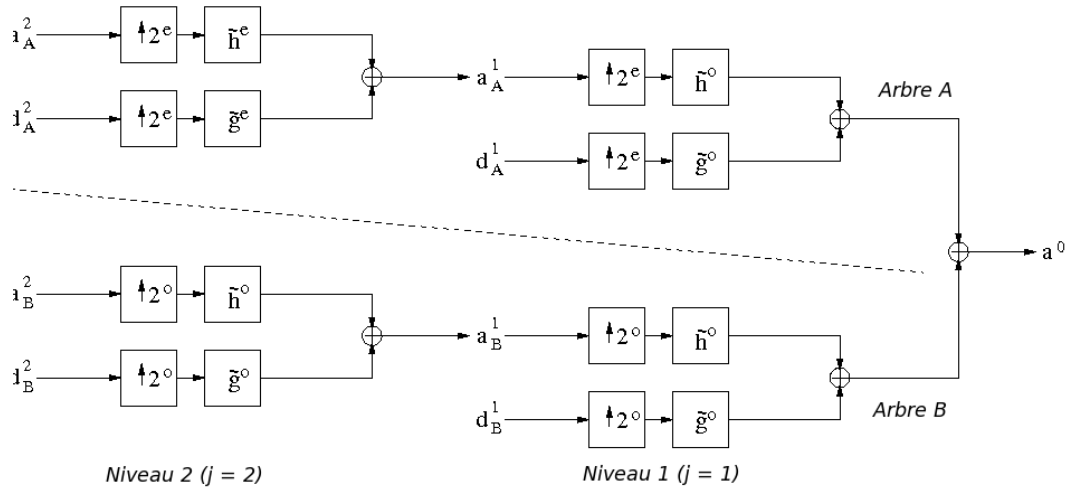


FIG. 4.2 – Transformée inverse en ondelettes complexes 1D.



Ceci implique la relation suivante sur les filtres :

$$g^o(n) \approx g^e(n - 0.5) \quad (4.7)$$

Aux niveaux  $j > 1$  les filtres doivent donc être décalés de  $\frac{1}{2}$  échantillon ( $\frac{1}{4}$  en 2D) afin d'obtenir une invariance par translation. Ceci est réalisé en utilisant des filtres de longueurs différentes, i.e. des filtres de longueur paire  $h^e, g^e$  pour un arbre et des filtres de longueur impaire  $h^o, g^o$  pour l'autre arbre. N. G. Kingsbury obtient ainsi une transformation en ondelettes approximativement invariante par translation.

En utilisant des bancs de filtres séparables, la transformée 1D peut être étendue à des signaux bidimensionnels. N. G. Kingsbury [15] a montré qu'une telle transformée restait efficace du point de vue du temps de calcul, et que l'on obtenait de meilleures propriétés de directionalité qu'avec l'équivalent réel malgré la séparabilité. La propriété de reconstruction exacte est conservée, ainsi que la presque invariance par translation.

On note  $(a_A, d_A)$  et  $(a_B, d_B)$  les signaux d'approximation et les détails pour les deux arbres  $A$  (arbre supérieur) et  $B$  (arbre inférieur). Les détails  $d_A$  et  $d_B$  peuvent donc être interprétés comme la partie réelle et la partie imaginaire d'un signal complexe  $z = d_A + i d_B$ . La propriété importante de cette transformée est que le seuillage des coefficients  $|z|$  donne beaucoup moins d'artéfacts que le seuillage des coefficients d'une transformée réelle [13].

La régularisation  $l^1$  de notre modèle réalise indirectement un seuillage (seuillage doux de seuil  $\lambda$  dans le cas du modèle de parcimonie, cf. chapitre 6), ce qui a pour conséquence de lisser notre image comme le montre la figure 4.3. Il faut donc être prudent, dans le cas d'une déconvolution, à correctement choisir le paramètre  $\lambda$  pour éviter de trop lisser l'image.



FIG. 4.3 – Image bruitée dont les coefficients ont été seuillés en utilisant un seuillage doux. De gauche à droite : image d'origine bruitée par un bruit blanc gaussien ( $RSB = 4.57dB$ ), résultat par seuillage doux ( $RSB = 12.60dB$ ).

## 4.2 Mise en oeuvre

Afin d'utiliser cette régularisation dans notre algorithme, il est nécessaire de déterminer l'expression de  $\Upsilon^*$  l'opérateur adjoint de la transformation en ondelettes complexes  $\Upsilon$ . Pour simplifier ce calcul, nous considérons la transformation non décimée sur un seul niveau. Les coefficients complexes  $z$  sont donc déterminés de la façon suivante :

$$z = g^o \star x \quad (4.8)$$

En utilisant une écriture matricielle, cette équation peut se réécrire comme le produit de la transformation en ondelettes complexes  $\Upsilon$  et du vecteur  $x$  :

$$z = \Upsilon x \quad (4.9)$$

Dans ce cas,  $\Upsilon$  est égale à la matrice de Toeplitz du filtre  $g^o$  :

$$\Upsilon = \begin{pmatrix} g^o(0) & g^o(1) & \cdots & \cdots & g^o(n-1) \\ g^o(-1) & g^o(0) & g^o(1) & \cdots & g^o(n-2) \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & g^o(1) \\ g^o(-n+1) & \cdots & \cdots & g^o(-1) & g^o(0) \end{pmatrix} \quad (4.10)$$

L'opérateur transposé conjugué (ou adjoint)  $\Upsilon^*$  s'écrit alors de la façon suivante :

$$\Upsilon^* = \begin{pmatrix} g^o(0) & g^o(-1) & \cdots & \cdots & g^o(-n+1) \\ g^o(1) & g^o(0) & g^o(-1) & \cdots & g^o(-n+2) \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & g^o(-1) \\ g^o(n-1) & \cdots & \cdots & g^o(1) & g^o(0) \end{pmatrix} \quad (4.11)$$

C'est la matrice de Toeplitz du filtre renversé  $g^o$ . En conséquence pour réaliser l'adjoint, nous utiliserons la structure de la transformée inverse et les filtres duaux seront les filtres retournés de la transformée en ondelettes complexes. Ce calcul a été validé en vérifiant la propriété suivante :

$$\langle \Upsilon x, y \rangle = \langle x, \Upsilon^* y \rangle \quad \forall x, y \in \mathbb{R}^N \quad (4.12)$$

Pour implanter l'algorithme, il est également nécessaire de calculer la constante de Lipschitz du nouveau problème. Le calcul est similaire au précédent modèle (cf. A.33), par analogie nous posons  $\nabla = \Upsilon$ , nous obtenons alors :

$$L_f = \frac{\|A\|_2^2 + \lambda \|\Upsilon\|_2^2}{\mu} \quad (4.13)$$

Les filtres de la transformée en ondelettes complexes sont tous de norme inférieure ou égale à 1, nous pouvons donc écrire que  $\|\Upsilon\|_2^2 \leq 1$ .

### 4.3 Résultats

Ce modèle a été testé sur le problème de rééchantillonnage, les résultats sont donnés sur la figure 4.4. Les modèles étant différents, nous pouvons difficilement comparer leur vitesse de convergence, néanmoins, la régularisation par ondelettes nécessite moins d'itérations pour converger (82 itérations contre 110 itérations pour la variation totale) cependant notre implantation de la transformée en ondelettes complexes n'est pas optimale et une itération est 4 fois plus longue qu'une itération de l'algorithme avec variation totale.

Nous pouvons remarquer que les résultats sont bien meilleurs en terme de qualité de reconstruction, les textures n'étant plus lissées comme avec la régularisation par variation totale. Néanmoins, le seuillage doux intrinsèque à cette régularisation lisse légèrement l'image et diminue l'intensité des petits éléments.

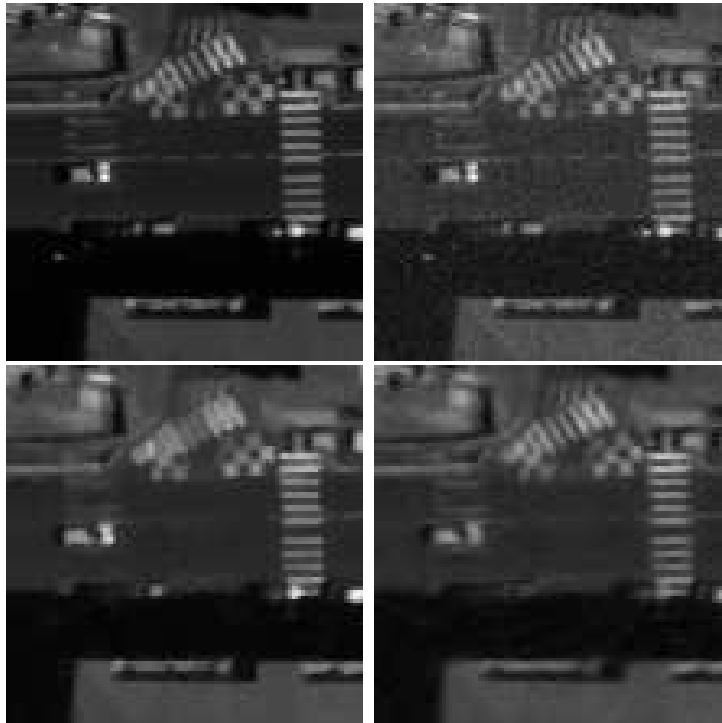


FIG. 4.4 – De gauche à droite et de haut en bas, l'image d'origine, l'image bruitée ( $RSB = 15.62 \text{ dB}$ ), le résultat avec la variation totale ( $RSB = 24.09 \text{ dB}$ ), et le résultat avec la régularisation par ondelettes ( $RSB = 24.42 \text{ dB}$ ).

## Chapitre 5

# Comparaison dual/primal

Le problème présenté dans le chapitre précédent peut être écrit sous sa forme duale [4, 24, 25]. Cette formulation duale semble être plus performante que la formulation primale [25]. Ce chapitre se consacre à la comparaison entre les modélisations duale et primale sur un problème de type  $BV - l^1$ , puis sur le problème de reconstruction. Les méthodes sont comparées grâce à l'algorithme de Y. Nesterov.

### 5.1 Modèle $BV - l^1$

Nous considérons le problème suivant (problème de type  $BV - l^1$  [21]) :

$$\arg \min_{u \in \mathbb{R}^N} \lambda \|u - g\|_1 + \|\nabla u\|_1 + \frac{\varepsilon}{2} \|u - g\|_2^2 \quad (5.1)$$

Le terme  $\frac{\varepsilon}{2} \|u - g\|_2^2$  est requis pour effectuer la comparaison primal/dual. Pour appliquer l'algorithme de Y. Nesterov, ce problème doit être différentiable, on introduit donc le paramètre  $\mu$  pour régulariser la norme  $l^1$ . L'équation (5.1) devient :

$$\arg \min_{u \in \mathbb{R}^N} \lambda \|u - g\|_{1,\mu} + \|\nabla u\|_{1,\mu} + \frac{\varepsilon}{2} \|u - g\|_2^2 \quad (5.2)$$

### 5.2 Algorithme primal

Pour cet algorithme, nous appliquons simplement le schéma de Nesterov directement sur le problème (5.2). Nous prenons :

$$f(u) = \lambda \|u - g\|_{1,\mu} + \|\nabla u\|_{1,\mu} \quad (5.3)$$

$$\Psi(u) = \frac{\varepsilon}{2} \|u - g\|_2^2 \quad (5.4)$$

Pour simplifier les calculs nécessaires à la mise en oeuvre de cet algorithme, nous utilisons l'opérateur proximal [5, 6]. Cet opérateur est défini de la façon suivante :

$$\text{prox}_{\alpha\Psi}(x) = \arg \min_{y \in Y} \left( \alpha\Psi(y) + \frac{1}{2}\|y - x\|_2^2 \right) \quad (5.5)$$

Dans le cas de fonction indicatrice pour  $\Psi$ , l'opérateur proximal est un opérateur de projection sur le même ensemble [6] :

$$\text{prox}_{\chi_K}(x) = \Pi_K(x) = \arg \min_{y \in K} \|y - x\|_2^2 \quad (5.6)$$

avec  $\Pi_K$  l'opérateur de projection sur cet ensemble (son expression est déterminée par la caractérisation de l'ensemble  $K$ , voir section suivante) et  $\chi_K$  la fonction indicatrice sur l'ensemble  $K$  :

$$\chi_K(y) = \begin{cases} 0 & \text{si } y \in K \\ \infty & \text{sinon} \end{cases} \quad (5.7)$$

L'algorithme présenté à la section 3.1 peut se réécrire de la façon suivante [24] :

---

**Algorithm 4** Méthode de gradient accélérée sur le problème primal  $BV - l^1$

---

Initialiser  $A_0 = 0$ ,  $w_0 = 0$

**for**  $k \geq 0$  **do**

$$a = \frac{1+\varepsilon A_k}{L_f} + \sqrt{\left(\frac{1+\varepsilon A_k}{L_f}\right)^2 + 2A_k \frac{1+\varepsilon A_k}{L_f}}$$

$$v_k = \text{prox}_{A_k\Psi}(x_0 - w_k)$$

$$y_k = \frac{A_k x_k + a v_k}{A_k + a}$$

$$x_{k+1} = \text{prox}_{\frac{1}{L_f}\Psi}\left(y_k - \frac{\nabla f(y_k)}{L_f}\right)$$

$$w_{k+1} = w_k + a \nabla f(x_{k+1})$$

$$A_{k+1} = A_k + a$$

**end for**

---

avec  $x_0$  la solution initiale et :

$$\nabla f(u) = \lambda \frac{(u - g)}{\sqrt{|u - g|^2 + \mu^2}} - \text{div} \left( \frac{\nabla u}{\sqrt{|\nabla u|^2 + \mu^2}} \right) \quad (5.8)$$

$$L_f = \frac{\lambda + \|\text{div}\|^2}{\mu} \quad (5.9)$$

### 5.3 Algorithme dual

Pour l'écriture duale de cet algorithme, nous utilisons une autre approximation de la norme  $l^1$  basée sur la régularisation de Moreau-Yosida [16] :

$$\|x\|_{1,\mu} = \max_{y \in \mathbb{R}^N, \|y\|_\infty \leq 1} \left( \langle x, y \rangle - \frac{\mu}{2} \|y\|_2^2 \right) \quad (5.10)$$

Nous réécrivons le problème (5.2) en utilisant cette définition :

$$\arg \min_{u \in \mathbb{R}^N} \left( \lambda \|u - g\|_{1,\mu} + \|\nabla u\|_{1,\mu} + \frac{\varepsilon}{2} \|u - g\|_2^2 \right) \quad (5.11)$$

$$= \arg \min_{u \in \mathbb{R}^N} \left( \max_{y \in \mathbb{R}^N, \|y\|_\infty \leq 1} \lambda \langle u - g, y \rangle + \langle \nabla u, y \rangle + \frac{\varepsilon}{2} \|u - g\|_2^2 - \frac{\mu}{2} \|y\|_2^2 \right) \quad (5.12)$$

$$= \arg \min_{u \in \mathbb{R}^N} \left( \max_{y \in \mathbb{R}^N, \|y\|_\infty \leq 1} \langle Cu - F, y \rangle + \frac{\varepsilon}{2} \|u - g\|_2^2 - \frac{\mu}{2} \|y\|_2^2 \right) \quad (5.13)$$

$$(5.14)$$

avec :

$$C = \begin{pmatrix} \lambda I \\ \nabla \end{pmatrix} \quad F = \begin{pmatrix} \lambda g \\ 0 \end{pmatrix} \quad (5.15)$$

La recherche du max sur  $y$  est, en fait, indépendante de la recherche de l'arg min sur  $u$ , nous pouvons donc les inverser :

$$\arg \min_{u \in \mathbb{R}^N} \left( \lambda \|u - g\|_{1,\mu} + \|\nabla u\|_{1,\mu} + \frac{\varepsilon}{2} \|u - g\|_2^2 \right) \quad (5.16)$$

$$= \max_{y \in \mathbb{R}^N, \|y\|_\infty \leq 1} \left( \arg \min_{u \in \mathbb{R}^N} \langle Cu - F, y \rangle + \frac{\varepsilon}{2} \|u - g\|_2^2 - \frac{\mu}{2} \|y\|_2^2 \right) \quad (5.17)$$

Soit :

$$s(u) = \langle Cu - F, y \rangle + \frac{\varepsilon}{2} \|u - g\|_2^2 - \frac{\mu}{2} \|y\|_2^2 \quad (5.18)$$

Nous cherchons :

$$\arg \min_{u \in \mathbb{R}^N} s(u) \quad (5.19)$$

Nous avons :

$$\nabla s(u) = C^* y + \varepsilon(u - g) = 0$$

D'où :

$$u = \frac{-C^* y}{\varepsilon} + g \quad (5.20)$$

avec  $C^*$  la matrice adjointe de  $C$  :

$$C^* = (\lambda I \quad -\text{div}) \quad (5.21)$$

L'équation (5.2) peut alors se réécrire sous la forme :

$$\max_{y \in \mathbb{R}^N, \|y\|_\infty \leq 1} \left( \langle Cg - F, y \rangle - \frac{1}{2\varepsilon} \|C^*y\|_2^2 - \frac{\mu}{2} \|y\|_2^2 \right) \quad (5.22)$$

$$= - \min_{y \in \mathbb{R}^N, \|y\|_\infty \leq 1} \left( \frac{1}{2\varepsilon} \|C^*y\|_2^2 - \langle Cg - F, y \rangle + \frac{\mu}{2} \|y\|_2^2 \right) \quad (5.23)$$

Pour ce problème nous utilisons également l'algorithme de Y. Nesterov, légèrement modifié du fait de la présence de la condition  $\|y\|_\infty \leq 1$ , qui se traduit par des projections sur l'ensemble  $K$  :

$$K = \{y \in \mathbb{R}^N; \|y\|_\infty \leq 1\} \quad (5.24)$$

Pour cet ensemble, l'opérateur de projection est défini de la façon suivante (projection sur une boule  $l^\infty$  de rayon 1) :

$$\Pi_K(x) = \begin{cases} x & \text{si } |x| < 1, \\ \frac{x}{|x|} & \text{sinon} \end{cases} \quad (5.25)$$

Nous prenons :

$$f(y) = \frac{1}{2\varepsilon} \|C^*y\|_2^2 - \langle Cg - F, y \rangle \quad (5.26)$$

$$\Psi(y) = \frac{\mu}{2} \|y\|_2^2 + \chi_K(y) \quad (5.27)$$

L'algorithme s'écrit alors de la façon suivante :

---

**Algorithm 5** Méthode de gradient accélérée sur le problème dual de  $BV - l^1$

---

Initialiser  $A_0 = 0, w_0 = 0$

**for**  $k \geq 0$  **do**

$$a = \frac{1+\mu A_k}{L_f} + \sqrt{\left(\frac{1+\mu A_k}{L_f}\right)^2 + 2A_k \frac{1+\mu A_k}{L_f}}$$

$$v_k = \Pi_K(x_0 - w_k)$$

$$y_k = \frac{A_k x_k + a v_k}{A_k + a}$$

$$x_{k+1} = \Pi_K\left(y_k - \frac{\nabla f(y_k)}{L_f}\right)$$

$$w_{k+1} = w_k + a \nabla f(x_{k+1})$$

$$A_{k+1} = A_k + a$$

**end for**

---

avec  $x_0$  la solution initiale et  $L_f$  la constante de Lipschitz du gradient de la fonction  $f$  :

$$L_f = \frac{1}{\varepsilon} \|C^*\|_2^2 \quad (5.28)$$

De plus :

$$\|C^*\|_2^2 = \max_{\|q\|_2 \leq 1} \|C^*q\|_2^2 \quad (5.29)$$

$$= \max_{\|q\|_2 \leq 1} \|\lambda q - \operatorname{div} q\|_2^2 \quad (5.30)$$

$$\leq \max_{\|q\|_2 \leq 1} (\lambda^2 \|q\|_2^2 + \|\operatorname{div}\|_2^2 \|q\|_2^2) \quad (5.31)$$

$$\leq \max_{\|q\|_2 \leq 1} ((\lambda^2 + \|\operatorname{div}\|_2^2) \|q\|_2^2) \quad (5.32)$$

D'où :

$$L_f = \frac{1}{\varepsilon} (\lambda^2 + \|\operatorname{div}\|_2^2) \quad (5.33)$$

La figure 5.1 montre les différentes courbes de convergence entre les deux algorithmes. Nous remarquons qu'il est plus intéressant d'utiliser notre algorithme sur le problème dual plutôt que sur le problème primal. Le modèle dual converge plus rapidement vers la solution et de manière plus stable. Une descente de gradient sur ce problème est relativement inefficace ce qui montre bien la supériorité des schémas de Y. Nesterov pour ce type de problèmes.

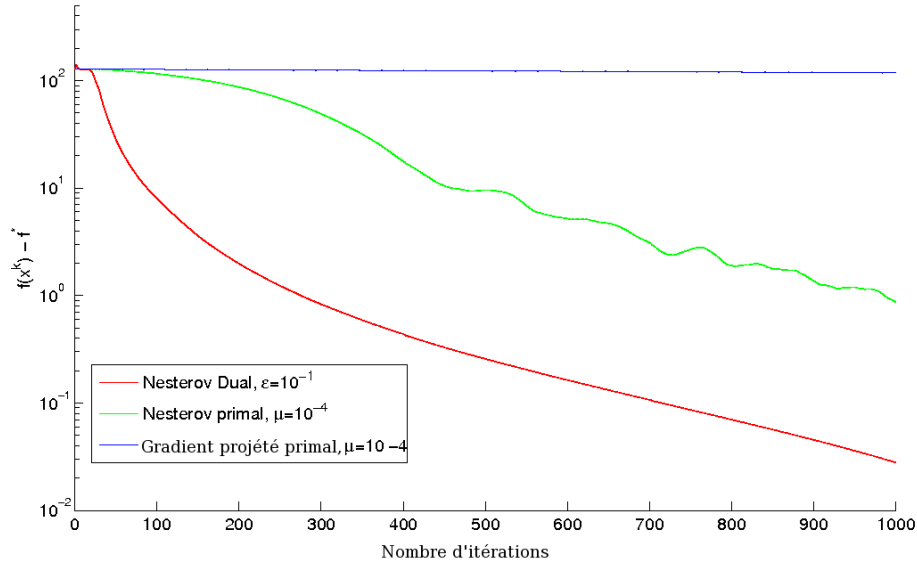


FIG. 5.1 – Comparaison des algorithmes primal, dual et de la descente de gradient sur le modèle  $BV - l^1$ .



## 5.4 Application au problème de reconstruction

Nous modifions légèrement le modèle du problème donné afin de pouvoir implanter son dual. Le nouveau problème de reconstruction s'écrit ainsi :

$$\arg \min_{u \in \mathbb{R}^N} \left( \|Au - g\|_1 + \lambda \|\nabla u\|_1 + \frac{\varepsilon}{2} \|u - g\|_2^2 \right) \quad (5.34)$$

Sous sa forme duale, ce problème devient :

$$\max_{y \in \mathbb{R}^N, \|y\|_\infty \leq 1} \left( \langle Cg - F, y \rangle - \frac{1}{2\varepsilon} \|C^*y\|_2^2 - \frac{\mu}{2} \|y\|_2^2 \right) \quad (5.35)$$

$$= - \min_{y \in \mathbb{R}^N, \|y\|_\infty \leq 1} \left( \frac{1}{2\varepsilon} \|C^*y\|_2^2 - \langle Cg - F, y \rangle + \frac{\mu}{2} \|y\|_2^2 \right) \quad (5.36)$$

avec :

$$C = \begin{pmatrix} A \\ \lambda \nabla \end{pmatrix} \quad F = \begin{pmatrix} g \\ 0 \end{pmatrix} \quad C^* = (A^* \quad -\lambda \operatorname{div}) \quad (5.37)$$

Nous prenons :

$$f(y) = \frac{1}{2\varepsilon} \|C^*y\|_2^2 - \langle Cg - F, y \rangle \quad (5.38)$$

$$\Psi(y) = \frac{\mu}{2} \|y\|_2^2 + \chi_K(y) \quad (5.39)$$

D'où :

$$L_f = \frac{1}{\varepsilon} (\|A^*\|_2^2 + \lambda^2 \|\operatorname{div}\|_2^2) \quad (5.40)$$

L'algorithme pour résoudre ce problème est le même que précédemment :

---

**Algorithm 6** Méthode de gradient accélérée sur le problème dual

---

Initialiser  $A_0 = 0, w_0 = 0$

**for**  $k \geq 0$  **do**

$$a = \frac{1+\mu A_k}{L_f} + \sqrt{\left(\frac{1+\mu A_k}{L_f}\right)^2 + 2A_k \frac{1+\mu A_k}{L_f}}$$

$$v_k = \Pi_K(x_0 - w_k)$$

$$y_k = \frac{A_k x_k + a v_k}{A_k + a}$$

$$x_{k+1} = \Pi_K\left(y_k - \frac{\nabla f(y_k)}{L_f}\right)$$

$$w_{k+1} = w_k + a \nabla f(x_{k+1})$$

$$A_{k+1} = A_k + a$$

**end for**

---

La figure 5.2 montre les résultats des deux modèles sur ce problème de rééchantillonnage. Nous remarquons clairement que la méthode primale est bornée tandis que la méthode

duale ne l'est pas. De plus, on remarque que cette méthode est plus efficace sur les premières itérations même si, à la fin, les deux méthodes se rejoignent. Cette méthode semble également beaucoup plus stable. En terme de temps de calcul, une itération de la méthode duale est deux fois plus longue qu'une itération de la méthode primale, mais est plus efficace pour déterminer le minimum.

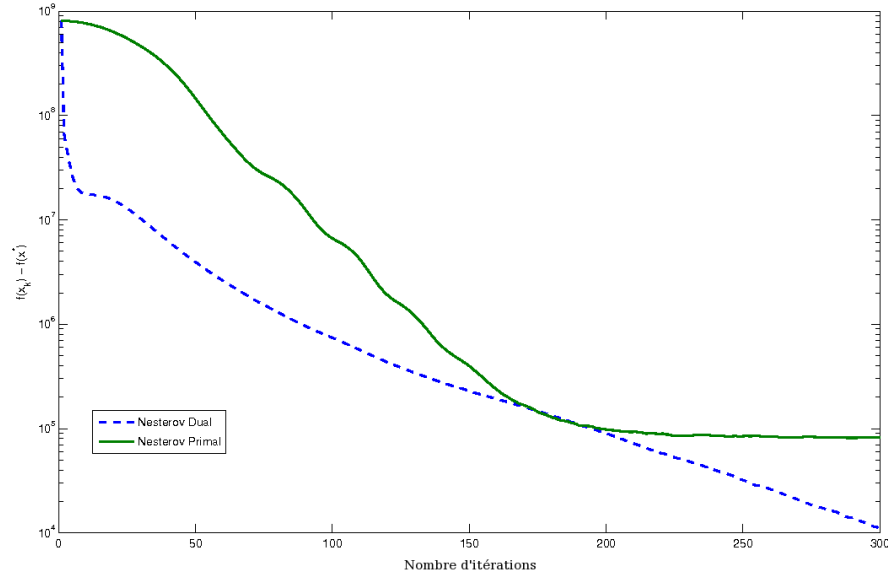


FIG. 5.2 – Comparaison des algorithmes primal et dual sur le problème considéré.



## Chapitre 6

# Comparaison régularité/parcimonie

La régularisation par ondelettes effectuée dans ce rapport n'est pas une modélisation habituelle. En effet, un modèle plus répandu dans la littérature est le modèle de parcimonie. Ce chapitre établit une comparaison entre ces deux modèles.

### 6.1 Modèle de parcimonie

Le modèle de parcimonie [8] est le modèle le plus utilisé à l'heure actuelle pour les ondelettes. Il s'écrit de la manière suivante :

$$y^* = \arg \min_{y \in Y} \frac{\lambda}{2} \|Ay - x_0\|_2^2 + \|y\|_1 \quad (6.1)$$

avec  $x_0$  les observations. L'opérateur  $A$  est l'opérateur de reconstruction. Il s'agit, en fait, d'une minimisation sur les coefficients en ondelettes d'un signal plutôt que sur le signal lui-même.

Pour effectuer la comparaison entre les modèles, nous utilisons pour commencer une transformée en ondelettes de Haar, redondante, non décimée.

Pour cela, l'étape de décomposition (opérateur  $A^{-1}$ ) consiste à créer, à partir de l'image d'origine, 4 images décalées circulairement de 1 pixel les unes par rapport aux autres (décalage vers le bas, la droite, la diagonale bas-droite et nous disposons de l'image d'origine). Nous appliquons, ensuite sur chaque image une transformée 2D en ondelettes de Haar sur 4 niveaux (le décalage nous permet à la fin de conserver tous les coefficients) multipliée par un facteur  $\frac{1}{2}$  (nécessaire pour obtenir l'orthonormalité). Nous obtenons alors une transformation redondante d'un facteur 4 non décimée. Ainsi  $Y = \mathbb{R}^{4N}$ , où  $N$  est le nombre de pixels

de l'image.

Pour la recombinaison (opérateur  $A$ ), nous appliquons, sur chaque image, la transformation inverse de Haar. Chaque image recomposée est ensuite décalée (décalage de 1 pixel vers le haut, la gauche, la diagonale haut-gauche et nous disposons de l'image d'origine). Nous en réalisons la somme, puis nous multiplions par le facteur  $\frac{1}{2}$ .

De part l'orthonormalité des ondelettes de Haar, nous pouvons facilement démontrer que  $A^* = A^{-1}$ .

Afin de comparer ces deux modèles, nous utilisons une descente de gradient réécrite pour utiliser l'opérateur proximal. Ceci peut s'écrire de la manière suivante :

$$y_{k+1} = \text{prox}_{\Psi} \left( y_k - \frac{1}{L} \nabla f(y_k) \right) \quad (6.2)$$

avec  $\Psi(x) = \|x\|_1$ ,  $f(x) = \frac{\lambda}{2} \|Ax - x_0\|_2^2$  et  $L$  la constante de Lipschitz du gradient de la fonction  $f$  :

$$L = \lambda \|A^*\|_2^2 \quad (6.3)$$

$$\|A^*\|_2^2 = 1 \quad (6.4)$$

D'après [6], nous avons :

$$\text{prox}_{\gamma \|\cdot\|_1} x = \text{sign}(x) \max(\|x\|_1 - \gamma, 0) \quad (6.5)$$

Ceci correspond à un seuillage doux de seuil  $\gamma$ . Nous obtenons l'algorithme suivant :

---

**Algorithm 7** Descente de gradient sur le modèle de parcimonie

---

```

Initialiser  $y_0$ 
for  $k \geq 0$  do
   $\eta_k = \nabla f(y_k) = \lambda A^*(Ay_k - x_0)$ 
   $y_{k+1} = \text{sign}(y_k - \frac{\eta_k}{\lambda}) \max(\|y_k - \frac{\eta_k}{\lambda}\|_1, 1)$ 
end for

```

---

## 6.2 Modèle de régularité

Le modèle de régularité se présente sous la forme :

$$x^* = \arg \min_{x \in X} \frac{\lambda}{2} \|x - x_0\|_2^2 + \|A^*x\|_1 \quad (6.6)$$

avec  $X = \mathbb{R}^N$ . Pour ce modèle, nous sommes obligés de passer à la forme duale, car nous ne pouvons pas appliquer directement l'opérateur proximal. En effet, nous ne pouvons pas

choisir  $\Psi(x) = \|A^*x\|_1$  car ceci n'est pas une fonction simple. Sous sa forme duale, l'équation (6.6) se réécrit :

$$q^* = -\arg \min_{\|q\|_\infty \leq 1} \frac{1}{2\lambda} \|Aq\|_2^2 - \langle x_0, Aq \rangle \quad (6.7)$$

La solution  $x^*$  est donnée par  $x^* = -\frac{A^*q^*}{\lambda} + x_0$ . Nous prenons  $\Psi$  comme étant la fonction indicatrice sur l'ensemble  $K$ , défini par :

$$K = \{q \in \mathbb{R}^N; \|q\|_\infty \leq 1\} \quad (6.8)$$

La descente de gradient peut alors se réécrire de la manière suivante :

$$q^{k+1} = \text{prox}_{\frac{1}{L_f}\Psi} \left( q^k - \frac{\nabla f(q^k)}{L_f} \right) = \Pi_K \left( q^k - \frac{\nabla f(q^k)}{L_f} \right) \quad (6.9)$$

avec :

$$\Psi(q) = \chi_K(q) \quad (6.10)$$

$$f(q) = \frac{1}{2\lambda} \|Aq\|_2^2 - \langle x_0, Aq \rangle \quad (6.11)$$

D'où :

$$L_f = \frac{1}{\lambda} \|A^*\|_2^2 = \frac{1}{\lambda} \quad (6.12)$$

L'algorithme est donc le suivant :

---

**Algorithm 8** Descente de gradient sur le modèle de régularité

---

```

Initialiser  $y_0$ 
for  $k \geq 0$  do
   $\eta_k = \nabla f(y_k) = \frac{1}{\lambda} A^*(Ay_k) - A^*x_0$ 
   $y_{k+1} = \Pi_K(y_k - \lambda \eta_k)$ 
end for

```

---

Les figures 6.1 et 6.2 montrent les résultats de reconstruction pour ces deux modèles. La reconstruction avec le modèle de parcimonie donne de gros artéfacts, alors que ceux-ci sont moindres avec le modèle de régularité. C'est pour cela qu'on le préférera.

Cette décomposition en ondelettes n'est, comme on peut le voir, pas très bien adaptée, elle permet cependant de bien mettre en évidence les défauts relatifs aux différents modèles.

### 6.3 Comparaison avec le modèle $BV - l^1$

On recommence l'expérience en y intégrant cette fois-ci une déconvolution par une gaussienne ( $\sigma = 1$ ) et en utilisant les ondelettes de N.G. Kingsbury. Nous rajoutons également une comparaison avec le résultat du modèle  $BV - l^1$ . Nous comparons ainsi :



FIG. 6.1 – De gauche à droite et de haut en bas : l'image d'origine, l'image d'origine bruitée ( $\sigma = 15$ ), le résultat du modèle de parcimonie et enfin le résultat du modèle de régularité.



FIG. 6.2 – De gauche à droite : zoom sur le résultat du modèle de régularité et zoom sur le résultat du modèle de parcimonie.

$$y^* = \arg \min_{y \in Y} \frac{\lambda}{2} \|H\Upsilon^{-1}y - x_0\|_2^2 + \|y\|_1 \quad (6.13)$$

$$x^* = \arg \min_{x \in X} \frac{\lambda}{2} \|Hx - x_0\|_2^2 + \|\Upsilon x\|_1 \quad (6.14)$$

$$x^* = \arg \min_{x \in X} \frac{\lambda}{2} \|Hx - x_0\|_2^2 + \|\nabla x\|_1 \quad (6.15)$$

Comme on peut difficilement utiliser la dualité, nous avons choisi d'utiliser une descente de gradient sur chaque problème en régularisant la norme  $l^1$  avec un paramètre  $\mu$  identique pour chaque modèle :

$$\|u\|_{1,\mu} = \sum_{i=1}^N \sqrt{u_i^2 + \mu^2} \quad (6.16)$$

Les algorithmes sont les suivants :

---

**Algorithm 9** Descente de gradient sur le modèle de régularité

---

```

Initialiser  $y_0, L_f = \lambda \|H\|_2^2 \|\Upsilon^{-1}\|_2^2 + \frac{1}{\mu}$ 
for  $k \geq 0$  do
   $\eta_k = \nabla f(y_k) = \lambda \Upsilon^{-*} H^* (H \Upsilon^{-1} y_k - x_0) + \frac{y_k}{\sqrt{|y_k|^2 + \mu^2}}$ 
   $y_{k+1} = y_k - \frac{1}{L_f} \eta_k$ 
end for

```

---

avec  $\|H\|_2^2 = \|\Upsilon^{-1}\|_2^2 = 1$  et  $H^* = H$ . L'opérateur  $\Upsilon^{-*}$  se construit en utilisant la structure de la transformée directe et les filtres sont les filtres de la transformée en ondelettes complexes inverse retournés (cf. chapitre 4).

---

**Algorithm 10** Descente de gradient sur le modèle de parcimonie

---

```

Initialiser  $y_0, L_f = \lambda \|H\|_2^2 + \frac{1}{\mu} \|\Upsilon\|_2^2$ 
for  $k \geq 0$  do
   $\eta_k = \nabla f(y_k) = \lambda H^* (H y_k - x_0) + \Upsilon^* \left( \frac{\Upsilon y_k}{\sqrt{|\Upsilon y_k|^2 + \mu^2}} \right)$ 
   $y_{k+1} = y_k - \frac{1}{L_f} \eta_k$ 
end for

```

---

avec  $\|\Upsilon\|_2^2 = 1$ .



---

**Algorithm 11** Descente de gradient sur le modèle  $BV - l^1$ 


---

Initialiser  $y_0, L_f = \lambda \|H\|_2^2 + \frac{1}{\mu} \|\text{div}\|_2^2$   
**for**  $k \geq 0$  **do**  
 $\eta_k = \nabla f(y_k) = \lambda H^*(Hy_k - x_0) - \text{div} \left( \frac{\nabla y_k}{\sqrt{|\nabla y_k|^2 + \mu^2}} \right)$   
 $y_{k+1} = y_k - \frac{1}{L_f} \eta_k$   
**end for**

---

avec  $\|\text{div}\|_2^2 = 8$ . La figure 6.3 illustre le résultat obtenu. Comme précédemment, le modèle de régularité donne beaucoup moins d'artéfacts que le modèle de parcimonie, cependant les deux modèles détériorent autant les contours en lissant l'image. La variation totale débruite correctement l'image tout en conservant les contours, malheureusement ce sont les textures qui sont lissées. Il est difficile de dire quel est le modèle le plus performant, néanmoins le modèle de régularité semble bien approprié pour de nombreuses applications dont la restauration d'images satellitaires par exemple, où la faible présence de bruit permet de ne pas trop lisser l'image tout en faisant de la déconvolution.

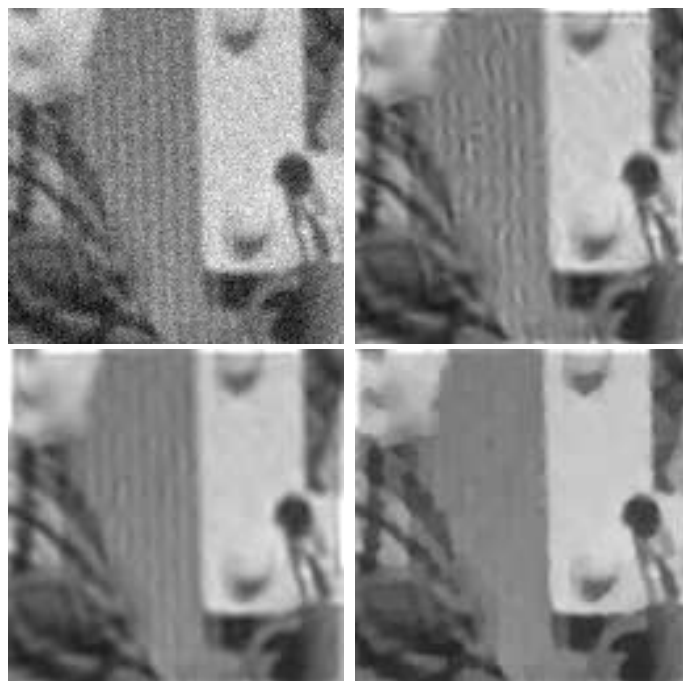


FIG. 6.3 – De gauche à droite et de haut en bas : l'image d'origine floutée et bruitée ( $\sigma = 15$ ), le résultat du modèle de parcimonie, le résultat du modèle de régularité et le résultat de la variation totale.



## Chapitre 7

# Conclusion et perspectives

En conclusion, nous avons adapté un schéma numérique récent pouvant convenir à une vaste catégorie de problèmes de traitement d'images, dont la restauration. Il conduit à un algorithme très rapide et présente un taux de convergence bien meilleur que les principales méthodes de minimisation actuelles. Nous avons également comparé un modèle de type  $l^1 - TO$  (Transformation en Ondelettes) et  $l^1 - VT$  (Variation Totale), chacun des deux apportant son lot d'avantages et de défauts : la variation totale est efficace sur les contours des objets, mais lisse les textures tandis que la transformation en ondelettes restaure correctement les textures, mais lisse l'image et crée des artéfacts.

En perspectives de ce travail, il serait intéressant d'approfondir ce problème de reconstruction en l'élargissant à un problème de super-résolution à partir de paires stéréoscopiques. Enfin, il serait également intéressant d'approfondir la comparaison parcimonie/régularité en utilisant une décomposition sur un dictionnaire pour la transformation en ondelettes.



## Annexe A

### Calcul de $T_L(y)$ :

On cherche :

$$T_L(y) = \arg \min_{x \in \mathcal{R}^N} m_L(y; x) \quad (\text{A.1})$$

avec :

$$m_L(y; x) = f(y) + \langle \nabla f(y), x - y \rangle + \frac{L}{2} \|x - y\|_2^2 \quad (\text{A.2})$$

On a :

$$\nabla_x f(y) + \nabla_x (\langle \nabla f(y), x - y \rangle) + \nabla_x \left( \frac{L}{2} \|x - y\|_2^2 \right) = 0 \quad (\text{A.3})$$

$$\nabla f(y) + L(x - y) = 0 \quad (\text{A.4})$$

On en déduit :

$$T_L(y) = x = y - \frac{1}{L} \nabla f(y) \quad (\text{A.5})$$

### Calcul de $v_k$ :

Nous cherchons maintenant l'expression de  $v_k = \arg \min_x \psi_k(x)$ . Nous pouvons écrire par récurrence que :

$$\psi_k(x) = \psi_0(x) + \sum_{i=1}^k a_i [f(x_i) + \langle \nabla f(x_i), x - x_i \rangle + \Psi(x)] \quad (\text{A.6})$$

Avec  $\psi_0(x) = \frac{1}{2} \|x - x_0\|_2^2$ . De plus :

$$\nabla \psi_k(x) = \nabla \psi_0(x) + \sum_{i=1}^k a_i [\nabla f(x_i) + \nabla \Psi(x)] = 0 \quad (\text{A.7})$$

$$x - x_0 + \sum_{i=1}^k a_i \nabla f(x_i) = 0 \quad (\text{A.8})$$

Soit :

$$v_k = x = x_0 - \sum_{i=1}^k a_i \nabla f(x_i) \quad (\text{A.9})$$

### Calcul de $a$ :

On cherche  $a$  la solution de l'équation :

$$\frac{a^2}{A_k + a} = 2 \frac{1}{L} \quad (\text{A.10})$$

En posant  $\lambda = \frac{1}{L}$  on a :

$$\frac{1}{2}a^2 - \lambda a - A_k \lambda = 0 \quad (\text{A.11})$$

Le discriminant de cette équation donne :

$$\Delta = \lambda^2 + 2A_k \lambda \quad (\text{A.12})$$

$a$  est la solution positive des deux solutions car la suite  $A_k$  doit être croissante, soit :

$$a = \lambda + \sqrt{\lambda^2 + 2A_k \lambda} = \frac{1}{L} + \sqrt{\left(\frac{1}{L}\right)^2 + \frac{2}{L}A_k} \quad (\text{A.13})$$

### Différentielle de $\|A. - g\|_{1,\mu}$ :

Pour calculer la différentielle de  $\|A. - g\|_{1,\mu}$ , nous allons faire un calcul au sens de Gateaux (dérivée selon un vecteur). Par exemple, si l'on cherche la dérivée d'une fonction  $F$  en  $u$ , nous allons calculer :

$$\lim_{t \rightarrow 0} \frac{F(u) + F(u + th)}{\|th\|} \quad (\text{A.14})$$

Par soucis de clarté, on note  $F_\mu$  la norme  $l^1$  régularisée  $\|A. - g\|_{1,\mu}$ . On cherche alors la dérivée de  $F_\mu$  définie par :

$$F_\mu : u \longrightarrow F_\mu(u) = \sum_i \sqrt{((Au)_i - g_i)^2 + \mu^2} \quad (\text{A.15})$$

Soit  $\phi_\mu$  la fonction de  $\mathbb{R}$  dans  $\mathbb{R}$  définie par :

$$\phi_\mu : x \longrightarrow \sqrt{x^2 + \mu^2} \quad (\text{A.16})$$

On a donc la relation suivante :

$$F_\mu(u) = \sum_i \phi_\mu((Au)_i - g_i) \quad (\text{A.17})$$

Calculons maintenant le développement à l'ordre 1 de  $F_\mu(u + th)$ ,  $t \in \mathbb{R}$  et  $h$  appartenant au même espace que  $u$  :

$$\begin{aligned} F_\mu(u + th) &= \sum_{i=1}^N \left( \phi_\mu((Au)_i - g_i) + t(Ah)_i \phi'_\mu((Au)_i - g_i) \right) \\ &= F_\mu(u) + \sum_{i=1}^N t(Ah)_i \phi'_\mu((Au)_i - g_i) \end{aligned} \quad (\text{A.18})$$

En utilisant  $\phi'_\mu(x) = x/\sqrt{x^2 + \mu^2}$  et en notant que  $\sum_i a_i b_i = \langle a, b \rangle$  on peut réécrire (A.18) sous la forme :

$$F_\mu(u + th) = F_\mu(u) + t \langle Ah, \frac{(Au - g)}{\sqrt{(Au - g)^2 + \mu^2}} \rangle \quad (\text{A.19})$$

En utilisant la relation  $\langle Au, v \rangle = \langle u, A^*v \rangle$  dans (A.19), on a alors :

$$F_\mu(u + th) = F_\mu(u) + t \langle h, A^* \left( \frac{(Au - g)}{\sqrt{(Au - g)^2 + \mu^2}} \right) \rangle \quad (\text{A.20})$$

En reportant (A.17) et (A.20) dans (A.14), on obtient la dérivée de  $F_\mu$  selon la direction  $h$  :

$$\langle \frac{h}{\|h\|}, F'_\mu(u) \rangle = \langle \frac{h}{\|h\|}, A^* \left( \frac{(Au - g)}{\sqrt{(Au - g)^2 + \mu^2}} \right) \rangle \quad (\text{A.21})$$

Et donc :

$$F'_\mu(u) = A^* \left( \frac{(Au - g)}{\sqrt{(Au - g)^2 + \mu^2}} \right) \quad (\text{A.22})$$

### Différentielle de $J_\mu(u)$ :

Pour calculer cette dérivée, il suffit de faire le même calcul que précédemment en posant  $A = \nabla$  et  $g = 0$ . De plus, en utilisant le fait que  $\nabla^* = -\text{div}$ , on obtient la formule suivante :

$$F'_\mu(u) = -\text{div} \left( \frac{\nabla u}{\sqrt{|\nabla u|^2 + \mu^2}} \right) \quad (\text{A.23})$$



### Calcul de la constante de Lipschitz $L_f$ :

Soit :

$$F_\mu(u) = \|Au - g\|_{1,\mu} + \lambda J_\mu(u) \quad (\text{A.24})$$

On recherche une constante  $L_f$  telle que  $\forall(u, v) \in \mathbb{R}^N \times \mathbb{R}^N$ , on a :

$$\|\nabla F_\mu(u) - \nabla F_\mu(v)\|_2 \leq L_f \|u - v\|_2 \quad (\text{A.25})$$

On a, alors, les relations suivantes :

$$\begin{aligned} \|\nabla F_\mu(u) - \nabla F_\mu(v)\|_2 &= \left\| A^* \left( \frac{(Au - g)}{\sqrt{(Au - g)^2 + \mu^2}} \right) - A^* \left( \frac{(Av - g)}{\sqrt{(Av - g)^2 + \mu^2}} \right) \right. \\ &\quad \left. - \lambda \left( \operatorname{div} \left( \frac{\nabla u}{\sqrt{|\nabla u|^2 + \mu^2}} \right) - \operatorname{div} \left( \frac{\nabla v}{\sqrt{|\nabla v|^2 + \mu^2}} \right) \right) \right\|_2 \end{aligned} \quad (\text{A.26})$$

Comme l'opérateur  $A^*$  est linéaire,  $\|A^*u - A^*v\|_2^2 = \|A^*(u - v)\|_2^2 \leq \|A^*\|_2^2 \|u - v\|_2^2$ . Il en est de même pour l'opérateur  $\operatorname{div}$ . On a donc en utilisant l'inégalité triangulaire :

$$\begin{aligned} \|\nabla F_\mu(u) - \nabla F_\mu(v)\|_2 &\leq \|A^*\|_2 \left\| \frac{(Au - g)}{\sqrt{(Au - g)^2 + \mu^2}} - \frac{(Av - g)}{\sqrt{(Av - g)^2 + \mu^2}} \right\|_2 \\ &\quad + \lambda \|\operatorname{div}\|_2 \left\| \frac{\nabla u}{\sqrt{|\nabla u|^2 + \mu^2}} - \frac{\nabla v}{\sqrt{|\nabla v|^2 + \mu^2}} \right\|_2 \end{aligned} \quad (\text{A.27})$$

On définit la fonction  $\psi_\mu$  pour tout  $x \in \mathbb{R}^N$  par :

$$\psi_\mu(x) = \begin{pmatrix} \frac{x_1}{\sqrt{x_1^2 + \mu^2}} \\ \vdots \\ \frac{x_N}{\sqrt{x_N^2 + \mu^2}} \end{pmatrix} = \begin{pmatrix} \phi_\mu(x_1) \\ \vdots \\ \phi_\mu(x_N) \end{pmatrix} \quad (\text{A.28})$$

On a alors  $\forall(x, y) \in \mathbb{R}^N \times \mathbb{R}^N$  :

$$\|\psi_\mu(x) - \psi_\mu(y)\|_2 = \sqrt{\sum_{i=1}^N (\phi_\mu(x_i) - \phi_\mu(y_i))^2} \quad (\text{A.29})$$

En remarquant de plus que  $\forall(x, y) \in \mathbb{R} \times \mathbb{R}$  :

$$\phi_\mu(x) - \phi_\mu(y) = \int_y^x \phi'_\mu(t) dt \leq \|\phi'_\mu\|_\infty |x - y| \quad (\text{A.30})$$

Donc :

$$\begin{aligned} \|\psi_\mu(x) - \psi_\mu(y)\|_2 &\leq \sqrt{\sum_{i=1}^N \|\phi'_\mu\|_\infty^2 |x_i - y_i|^2} \\ &\leq \|\phi'_\mu\|_\infty \|x - y\|_2 \end{aligned} \quad (\text{A.31})$$

En introduisant  $\psi_\mu$  dans (A.27), on a alors :

$$\begin{aligned} \|\nabla F_\mu(u) - \nabla F_\mu(v)\|_2 &\leq \|A^*\|_2 \|\psi_\mu(Au - g) - \psi_\mu(Av - g)\|_2 \\ &\quad + \lambda \|\text{div}\|_2 \|\psi_\mu\|_\infty (\|\nabla u\| - \|\nabla v\|)_2 \\ &\leq \|A^*\|_2 \|\psi_\mu\|_\infty \|Au - Av\|_2 + \lambda \|\text{div}\|_2 \|\psi_\mu\|_\infty \|\nabla u - \nabla v\|_2 \\ &\leq \|\psi_\mu\|_\infty (\|A^*\|_2 \|A\|_2 + \lambda \|\text{div}\|_2 \|\nabla\|_2) \|u - v\|_2 \end{aligned} \quad (\text{A.32})$$

En remarquant que  $\|\psi_\mu\|_\infty = \frac{1}{\mu}$ , on trouve finalement la valeur de  $L_f$  suivante :

$$L_f = \frac{\|A\|_2^2 + \lambda \|\text{div}\|_2^2}{\mu} \quad (\text{A.33})$$

Le calcul de  $\|A\|_2$  est donné dans la section suivante et on prendra  $\|A\|_2 \leq 1$ , celui de  $\|\text{div}\|_2$  est connu [25] et on sait que  $\|\text{div}\|_2 \leq 2\sqrt{2}$ .

### Calcul de $\|A\|_2$ par la méthode des puissances itérées :

La méthode des puissances itérées est une méthode permettant de calculer la valeur propre dominante d'une matrice diagonalisable. Soit  $A$  une matrice de  $\mathbb{R}^n \times \mathbb{R}^n$  diagonalisable. On note  $(\lambda_i)_{i=1..n}$  les valeurs propres ordonnées de  $A$ . c'est-à-dire :

$$|\lambda_1| > |\lambda_2| > \dots > |\lambda_n| \quad (\text{A.34})$$

Soit  $v_0$  un vecteur quelconque de  $\mathbb{R}^n$ , et  $(u_1, \dots, u_n)$  une base de  $\mathbb{R}^n$ . On peut alors l'écrire sous la forme :

$$v_0 = \sum_{i=1}^n \alpha_i u_i, \quad \alpha_i \in \mathbb{R} \quad (\text{A.35})$$

Soit la suite de vecteurs :

$$w_k = \begin{cases} Av_0, & \text{si } k = 1 \\ Aw_{k-1}, & \text{si } k \in \mathbb{N}^* \setminus \{1\} \end{cases} \quad (\text{A.36})$$

En utilisant la définition (A.35) de  $v_0$ , on peut alors écrire :

$$w_k = \sum_{i=1}^n \alpha_i \lambda_i^k u_i, \quad \forall k \in \mathbb{N}^* \quad (\text{A.37})$$

Ou encore

$$w_k = \lambda_1^k \left( \alpha_1 u_1 + \sum_{i=2}^n \alpha_i \left( \frac{\lambda_i}{\lambda_1} \right)^k u_i \right), \quad \forall k \in \mathbb{N}^* \quad (\text{A.38})$$

Or d'après (A.34),  $\lim_{k \rightarrow \infty} \left( \frac{\lambda_i}{\lambda_1} \right)^k = 0$ . Donc le vecteur  $w_k$  converge vers un vecteur propre de  $A$  associé à la valeur propre  $\lambda_1$ , soit :

$$Aw_\infty = \lambda_1 w_\infty \quad (\text{A.39})$$

On a donc l'algorithme suivant :

$$\begin{cases} v_0 \in \mathbb{R}^n \\ w_k = v_k \\ v_k = \frac{w_k}{\|w_k\|_\infty} \end{cases} \quad (\text{A.40})$$

avec :

$$\lim_{k \rightarrow \infty} \|w_k\|_\infty = |\lambda_1| \quad (\text{A.41})$$

### Lemme du gradient :

Ici  $f$  est une fonction convexe dont le gradient est  $L_f$ -lipschitzien. Soit  $\phi(\gamma) = f(x + \gamma(y - x))$  :

$$\begin{aligned} \|\phi'(\alpha) - \phi'(\beta)\|_2 &= \|(y - x)^T (\nabla f(x + \alpha(y - x)) - \nabla f(x + \beta(y - x)))\|_2 \\ &\leq \|(y - x)\|_2 \|\nabla f(x + \alpha(y - x)) - \nabla f(x + \beta(y - x))\|_2 \end{aligned} \quad (\text{A.42})$$

Or :

$$\|\nabla f(u) - \nabla f(v)\|_2 \leq L_f \|u - v\|_2 \quad (\text{A.43})$$

D'où :

$$\|\phi'(\alpha) - \phi'(\beta)\|_2 \leq \|(y - x)\|_2^2 L_f \|\alpha - \beta\|_2 \quad (\text{A.44})$$

De plus :

$$\begin{aligned} f(y) - f(x) - (y - x)^T \nabla f(x) &= \phi(1) - \phi(0) - \phi'(0) = \int_0^1 \phi'(\alpha) d\alpha - \phi'(0) \\ &= \int_0^1 [\phi'(\alpha) - \phi'(0)] d\alpha \leq \int_0^1 \|y - x\|_2^2 L_f \alpha d\alpha = \frac{L_f}{2} \|y - x\|_2^2 \end{aligned} \quad (\text{A.45})$$

D'où :

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L_f}{2} \|x - y\|_2^2 \quad (\text{A.46})$$

D'autre part, comme  $f$  est convexe :

$$f(y) - f(x) \geq \langle \nabla f(x), y - x \rangle \quad (\text{A.47})$$

On obtient ainsi :

$$f(x) + \langle \nabla f(x), y - x \rangle \leq f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L_f}{2} \|x - y\|_2^2 \quad (\text{A.48})$$



# Bibliographie

- [1] J. Bect, L. Blanc-Féraud, G. Aubert, and A. Chambolle. A  $l^1$ -unified variational framework for image restoration. In T. Pajdla and J. Matas, editors, *Proc. European Conference on Computer Vision (ECCV)*, volume LNCS 3024, Springer, pages 1–13, Prague, Czech Republic, 2004.
- [2] G. Beylkin. On the fast Fourier transform of function with singularities. *Applied and Computational Harmonic Analysis*, 2(4) :363–381, 1995.
- [3] E. Bughin, L. Blanc-Féraud, and J. Zerubia. Satellite image reconstruction from an irregular sampling. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Las Vegas, USA, 2008.
- [4] A. Chambolle. An algorithm for total variation minimization and applications. *J. Math. Imaging Vis.*, 20(1-2) :89–97, 2004.
- [5] C. Chaux, P. L. Combettes, J.-C. Pesquet, and V. R. Wajs. A forward-backward algorithm for image restoration with sparse representations. In *Signal Processing with Adaptive Sparse Structured Representations (SPARS'05)*, pages 49–52, Rennes, France, Nov 2005.
- [6] P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, 4(4) :1168–1200, 2005.
- [7] J. Delon and B. Rougé. Analytic study of the stereoscopic correlation. Technical Report 2004-18, CMLA, ENS Cachan, France, 2004.
- [8] R.A. DeVore and B.J. Lucier. Fast wavelet techniques for near-optimal image processing. *Military Communications Conference, 1992. MILCOM '92, Conference Record. 'Communications - Fusing Command, Control and Intelligence', IEEE*, pages 1129–1135 vol.3, Oct 1992.
- [9] H. G. Feichtinger, K. Gröchenig, and T. Strohmer. Efficient numerical methods in non-uniform sampling theory. *Numerische Mathematik*, 69 :423–440, 2000.
- [10] H. Haoying Fu, M. Ng, M. Nikolova, and J. Barlow. Efficient minimization methods of mixed  $l1$ - $l1$  and  $l2$ - $l1$  norms for image restoration. *SIAM Journal on Scientific computing*, 27(6), 2006.

- [11] K. Gröchenig and T. Strohmer. *Numerical and theoretical aspects of non-uniform sampling of band-limited images*. Kluwer / Plenum, f. m ed. edition, 2000.
- [12] R. G. Baraniuk I. W. Selesnick and N. G. Kingsbury. The dual-tree complex wavelet transform. *IEEE Signal Processing Magazine*, 22(6) :123–151, 2005.
- [13] A. Jalobeanu, L. Blanc-Féraud, and J. Zerubia. Satellite image debbluring using complex wavelet packets. *IJCV*, 51(3) :205–217, 2003.
- [14] M. I. Kadec. The exact value of the Payley-Wiener constant. *Soviet Math Doklady*, 5 :559–561, 1964.
- [15] N. G. Kingsbury. The dual-tree complex wavelet transform : a new technique for shift invariance and directional filters. In *IEEE Digital Signal Processing Workshop*, Bryce Canyon UT, USA, 1998.
- [16] J.J. Moreau. Proximité et dualité dans un espace hilbertien. *Bull. Soc. Math. Fr.*, 93 :273–299, 1965.
- [17] Y. Nesterov. A method for unconstrained convex minimization problem with the rate of convergence  $O(\varepsilon^{-2})$ . *Doklady AN SSSR*, 269(3) :543–547, 1983.
- [18] Y. Nesterov. *Introductory lectures on convex optimization. A basic course*. Kluwer, 2004.
- [19] Y. Nesterov. Accelerating the cubic regularization of newton’s method on convex problems. *Mathematic Programming*, 2005.
- [20] Y. Nesterov. Gradient methods for minimizing composite objective function. *CORE Discussion Paper*, (76), 2007.
- [21] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4) :259–268, 1992.
- [22] L.I. Rudin and S. Osher. Total variation based image restoration with free local constraints. In *Proc. IEEE International Conference in Image Processing (ICIP)*, pages 31–35, Austin, TX, USA, 1994.
- [23] C. Vonesch and M. Unser. Fast wavelet-regularized image deconvolution. *Biomedical Imaging : From Nano to Macro, 2007. ISBI 2007. 4th IEEE International Symposium on*, pages 608–611, Apr 2007.
- [24] P. Weiss. Algorithmes rapides d’optimisation convexe. applications à la restauration d’images et à la détection de changements. Thèse de doctorat, Université de Nice Sophia Antipolis, Nov 2008.
- [25] P. Weiss, G. Aubert, and L. Blanc-Féraud. Efficient schemes for total variation minimization under some convex constraints in image processing. Rapport de recherche 6260, INRIA, 2007.



---

Unité de recherche INRIA Sophia Antipolis  
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes  
4, rue Jacques Monod - 91893 ORSAY Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique  
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)

Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)

Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

---

Éditeur  
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)  
<http://www.inria.fr>  
ISSN 0249-6399