

# Un nouvel algorithme proximal pour l'optimisation convexe non lisse

Laurent CONDAT

GIPSA-lab – CNRS, Université de Grenoble  
11 rue des mathématiques, campus BP 46, 38402 St Martin d'Hères Cedex, France  
Laurent.Condat@gipsa-lab.grenoble-inp.fr

**Résumé** – Nous proposons un nouvel algorithme d'optimisation par éclatement pour minimiser une somme de fonctions convexes lisses ou non lisses, composées ou non avec des opérateurs linéaires. Cette formulation générique inclut, entre autres, de nombreux problèmes inverses régularisés en traitement d'images. L'algorithme proposé, facile à implémenter, et dont la convergence est prouvée, permet de résoudre exactement le problème. Toutes les opérations sont explicites : il n'y a pas de boucle interne ou de système linéaire à résoudre à chaque itération. Les méthodes classiques par éclatement de Douglas–Rachford et explicite-implicite, ainsi que la méthode récente de Chambolle et Pock, toutes d'une efficacité avérée, sont retrouvées comme des cas particuliers. Un exemple d'application en traitement d'image est montré.

**Abstract** – We propose a new optimization algorithm to minimize a sum of convex functions, which may be smooth or not and composed or not with linear operators. This generic formulation encompasses, for instance, numerous regularized inverse problems in imaging. The proposed splitting algorithm, easy to implement and whose convergence is proved, allows to solve the problem exactly. All the operations involved are explicit: there is no inner loop or linear system to solve at each iteration. The classical Douglas–Rachford and forward–backward splitting methods, as well as the recent algorithm of Chambolle–Pock, all with well-known efficiency, are recovered as particular cases. An example of application in imaging is shown.

## 1 Introduction et formulation du problème

De nombreux problèmes en traitement du signal ou d'image [1–4], en apprentissage [5, 6], ou en vision par ordinateur [7, 8], peuvent se formuler comme la minimisation d'une somme de fonctions convexes, non nécessairement différentiables, éventuellement composées avec des opérateurs linéaires. De plus, les variables vivent dans des espaces de grande dimension, égale par exemple au nombre de pixels si l'inconnue est une image. L'utilisation de fonctions non lisses comme la norme  $\ell_1$  permet de contraindre la solution à être parcimonieuse en un certain sens, une approche dont l'intérêt n'est plus à démontrer [5, 9]. La variation totale est l'exemple type d'une fonction non lisse composée avec un opérateur linéaire ; elle permet de régulariser de manière efficace de nombreux problèmes inverses mal posés en traitement d'image [10].

Ici, nous nous intéressons au problème générique suivant :

$$\text{Trouver } \hat{x} \in \arg \min_{x \in \mathcal{X}} F(x) + G(x) + \sum_{m=1}^M H_m(L_m x), \quad (1)$$

pour un certain entier  $M \in \mathbb{N}$ , où

- $\mathcal{X}$  est un espace de Hilbert réel.
- les fonctions  $F$ ,  $G$ ,  $H_m$ , à valeurs dans  $\mathbb{R} \cup \{+\infty\}$ , sont convexes, propres (ne valent pas  $+\infty$  partout) et semi-continues inférieurement.  $F$  et  $G$  sont définies sur  $\mathcal{X}$ , alors que chaque fonction  $H_m$  est définie sur un certain espace de Hilbert  $\mathcal{Y}_m$ .

- les opérateurs  $L_m : \mathcal{X} \rightarrow \mathcal{Y}_m$  sont linéaires et bornés ;
- la fonction  $F$  est différentiable sur  $\mathcal{X}$  et son gradient  $\nabla F$  est  $\beta$ -Lipschitz continu, pour un certain réel  $\beta > 0$ , c'est-à-dire que

$$\|\nabla F(x) - \nabla F(x')\| \leq \beta \|x - x'\|, \quad \forall x, x' \in \mathcal{X}. \quad (2)$$

- l'ensemble des minimiseurs est supposé non vide.
- la contrainte de qualification suivante est satisfaite :  $(0, \dots, 0) \in \text{sri}\{(L_m x - y_m)_{1 \leq m \leq M} \mid x \in \text{dom}(G) \text{ et } y_m \in \text{dom}(H_m), \forall m = 1, \dots, M\}$ , où  $\text{sri}$  indique l'intérieur relatif fort et  $\text{dom}$  le domaine.

Nous renvoyons le lecteur à un ouvrage d'analyse convexe, tel que [11], pour les définitions des notions classiques employées ci-dessus.

Définissons l'opérateur proximal de Moreau [11, 12] de la fonction  $G$  comme

$$\text{prox}_G : \mathcal{X} \rightarrow \mathcal{X}, x \mapsto \arg \min_{x' \in \mathcal{X}} \frac{1}{2} \|x - x'\|^2 + G(x'), \quad (3)$$

ainsi que, de manière similaire, les opérateurs  $\text{prox}_{H_m}$ . Il existe des formes explicites simples pour les opérateurs proximaux d'une large classe de fonctions d'intérêt pratique [12, 13].

Les algorithmes proximaux du premier ordre (qui ne font appel qu'au gradient ou à l'opérateur proximal des fonctions et ne requièrent ni n'utilisent aucune information sur les dérivées secondes) sont appropriés pour les problèmes à grande échelle : si par exemple  $\mathcal{X} = \mathbb{R}^N$  avec  $N \approx 10^6$ , il n'est pas envisageable de manipuler des matrices de taille  $N \times N$ , comme la Hessienne d'une fonction. Nous nous intéressons

aux algorithmes qui procèdent par éclatement (*splitting*), c'est-à-dire que les opérations effectuées à chaque itération n'impliquent les fonctions qu'une par une, au moyen d'appels aux opérateurs  $L_m$ , leurs adjoints  $L_m^*$ ,  $\nabla F$ , et les opérateurs proximaux de  $G$  et des  $H_m$ . Nous nous plaçons ainsi dans le cas où il est facile (typiquement de complexité linéaire) d'appliquer, à un élément quelconque, chacun de ces opérateurs, mais difficile et non souhaitable d'appliquer un opérateur du type  $(I + L_m^* L_m)^{-1}$ ,  $\text{prox}_{H_m \circ L_m}$ , ou  $\text{prox}_{F+G}$ .

Les algorithmes proximaux classiques, permettant de minimiser la somme de deux fonctions, sont les algorithmes explicite-implicite (*forward-backward*) et de Douglas-Rachford. Nous renvoyons à [12] pour une description des ces méthodes et de plusieurs autres, et une liste étendue de références associées. Plusieurs nouvelles méthodes proximales on vu le jour récemment, étendant la classe des problèmes pouvant être résolus [14–19]. Cependant, seule la méthode de [18] permet de résoudre le problème (1) en toute généralité, les autres méthodes n'en traitant que des cas particuliers. Il est possible d'imbriquer les algorithmes, en effectuant des itérations d'un algorithme à l'intérieur d'un autre algorithme, mais cela pose des questions de convergence d'ordres théorique et pratique [20].

L'algorithme que nous proposons, et dont nous prouvons la convergence, résout exactement le problème (1). Contrairement à la méthode de [18], il englobe comme cas particuliers les algorithmes explicite-implicite ( $M = 0$ ), de Douglas-Rachford ( $F = 0$ ,  $M = 1$ ,  $L_1 = \text{Id}$ ) et de Chambolle-Pock [14] ( $F = 0$ ). Il peut ainsi être vu comme une généralisation de ce dernier.

## 2 Algorithme proposé

L'algorithme proposé pour résoudre (1) est le suivant :

**Algorithme proposé.** Choisir les paramètres  $\tau > 0$ ,  $\sigma > 0$ ,  $\rho > 0$  et les estimées initiales  $x^{(0)} \in \mathcal{X}$ ,  $y_1^{(0)} \in \mathcal{Y}_1, \dots, y_M^{(0)} \in \mathcal{Y}_M$ . Puis itérer, pour  $i = 0, 1, \dots$

$$\left[ \begin{array}{l} \tilde{x}^{(i+1)} := \text{prox}_{\tau G}(x^{(i)} - \tau \nabla F(x^{(i)}) - \tau \sum_{m=1}^M L_m^* y_m^{(i)}), \\ x^{(i+1)} := \rho \tilde{x}^{(i+1)} + (1 - \rho)x^{(i)}, \\ \text{Pour } m = 1, \dots, M, \\ \left[ \begin{array}{l} \tilde{y}_m^{(i+1)} := \text{prox}_{\sigma H_m^*}(y_m^{(i)} + \sigma L_m(2\tilde{x}^{(i+1)} - x^{(i)})), \\ y_m^{(i+1)} := \rho \tilde{y}_m^{(i+1)} + (1 - \rho)y_m^{(i)}, \end{array} \right. \end{array} \right.$$

où  $H_m^*$  est la conjuguée de Fenchel-Rockafellar de la fonction  $H_m$  [11]. Il suffit dans notre cadre de connaître l'identité de Moreau : quel que soit l'élément  $y$  et le réel  $\sigma > 0$ ,  $\text{prox}_{\sigma H_m^*}(y) = y - \sigma \text{prox}_{H_m/\sigma}(y/\sigma)$ .

Notons que chaque itérée  $x^{(i)}$  appartient au domaine  $\text{dom}(G) = \{x \in \mathcal{X} \mid G(x) \neq +\infty\}$ .

Le principal résultat de convergence de l'algorithme, dont la preuve est détaillée dans notre article [21], est le suivant :

**Théorème 1.** *Supposons que les paramètres de l'algorithme vérifient :*

- (i)  $\frac{1}{\tau} - \sigma \left\| \sum_{m=1}^M L_m^* L_m \right\| \geq \frac{\beta}{2}$ , où la constante de Lipschitz  $\beta$  est définie en (2).
- (ii)  $\rho \in ]0, \delta[$ , où  $\delta = 2 - \frac{\beta}{2} \left( \frac{1}{\tau} - \sigma \left\| \sum_{m=1}^M L_m^* L_m \right\| \right)^{-1} \in [1, 2[$ .

Alors la séquence  $(x^{(i)})_{i \in \mathbb{N}}$  générée par l'algorithme converge (faiblement en dimension infinie) vers un élément  $\hat{x} \in \mathcal{X}$  solution du problème (1).

Nous renvoyons à notre article [21] pour des variantes de l'algorithme et une étude plus complète de la convergence, incluant notamment des termes d'erreurs et des paramètres de relaxation variables. De plus, un aspect important de l'approche, occulté ici par volonté de simplicité, est qu'elle est primale-duale : l'élément  $(y_1^{(i)}, \dots, y_M^{(i)})$  calculé par l'algorithme converge vers un élément  $(\hat{y}_1, \dots, \hat{y}_M) \in \mathcal{Y}_1 \times \dots \times \mathcal{Y}_M$  solution du problème dual de (1). Notons aussi que des contraintes de qualification autres que celle donnée en section 1 peuvent être trouvées [18]. Enfin, comme précisé dans [21], il est aisé de généraliser le problème (1) afin d'inclure des convolutions infimales, ou d'exprimer le problème et l'algorithme en termes d'inclusions monotones d'opérateurs, plutôt que dans le cadre de l'optimisation ; c'est ce qui a été fait dans [22].

Dans le cas où  $F = 0$  en (1), on a le résultat suivant :

**Théorème 2.** *Supposons que  $F = 0$ , que les espaces  $\mathcal{X}$  et  $\mathcal{Y}_m$  sont de dimension finie et que les paramètres de l'algorithme vérifient :*

- (i)  $\tau \sigma \left\| \sum_{m=1}^M L_m^* L_m \right\| \leq 1$ .
- (ii)  $\rho \in ]0, 2[$ .

Alors la séquence  $(x^{(i)})_{i \in \mathbb{N}}$  générée par l'algorithme converge vers un élément  $\hat{x} \in \mathcal{X}$  solution du problème (1).

Dans le cas  $F = 0$ , notre algorithme prend la forme de l'algorithme de Chambolle-Pock [14], avec en plus une relaxation. Or, prendre  $\rho$  proche de 2, plutôt que  $\rho = 1$  comme dans [14], peut significativement accélérer la convergence. De plus, le choix de paramètres tels que  $\sigma \tau \left\| \sum_{m=1}^M L_m^* L_m \right\| = 1$  (ce que nous recommandons en pratique), garantit bien la convergence (la condition donnée dans [14] est  $\sigma \tau \left\| \sum_{i=1}^m L_i^* L_i \right\| < 1$ ). Cette propriété est importante pour retrouver l'algorithme de Douglas-Rachford dans le cas particulier  $M = 1$ ,  $L_1 = \text{Id}$ , en posant  $\sigma = 1/\tau$  pour ne plus faire apparaître qu'un seul paramètre proximal  $\tau$ .

### 3 Application aux problèmes inverses 2D régularisés par la variation totale

On s'intéresse dans cette section aux problèmes inverses en imagerie. On se place dans l'espace  $\mathcal{X} = \mathbb{R}^{N_h \times N_v}$  des images en niveaux de gris de taille  $N_h$  colonnes  $\times N_v$  lignes (muni du produit scalaire canonique) et on souhaite restaurer ou reconstruire une image  $\hat{x}$  en résolvant le problème

$$\text{Trouver } \hat{x} \in \arg \min_{x \in \Omega} \frac{1}{2} \|Ax - v\|^2 + \lambda \cdot \text{TV}(x), \quad (4)$$

où :

- L'objet  $v \in \mathcal{V}$  représente les données disponibles, appartenant à un espace de Hilbert réel  $\mathcal{V}$ . Un exemple non restrictif est celui où  $v$  est une version dégradée de l'image à restaurer, auquel cas  $\mathcal{V} = \mathcal{X}$ .
- $A : \mathcal{X} \rightarrow \mathcal{V}$  est un opérateur linéaire (d'acquisition/dégradation).
- $\Omega$  est un sous-ensemble de  $\mathcal{X}$ . Par exemple,  $\Omega$  peut être l'ensemble  $(\mathbb{R}^+)^{N_h \times N_v}$  des images à valeurs positives ; un tel choix plutôt que  $\Omega = \mathcal{X}$  permet de limiter l'apparition d'oscillations indésirables dans des problèmes comme la déconvolution.
- $\lambda > 0$  est un paramètre à régler.

La variation totale discrète, dite isotrope, d'une image, notée TV en (4), est définie comme suit. Pour une paire d'image  $(y_h, y_v)$  de  $\mathcal{X}^2$ , on définit la norme  $\ell_{1,2}$  comme

$$\|(y_h, y_v)\|_{1,2} = \sum_{k_h=1}^{N_h} \sum_{k_v=1}^{N_v} \sqrt{y_h[k_h, k_v]^2 + y_v[k_h, k_v]^2}. \quad (5)$$

Notons  $H = \lambda \|\cdot\|_{1,2}$ . Pour tout  $\sigma > 0$ , on a

$$\text{prox}_{\sigma H^*} : (y_h, y_v) \mapsto \frac{(y_h, y_v)}{\max(\sqrt{y_h^2 + y_v^2}/\lambda, 1)}, \quad (6)$$

où les opérations sont à comprendre pixel à pixel (il est normal que  $\sigma$  n'intervienne pas dans cette formule).

On définit le gradient discret d'une image comme l'opérateur linéaire  $D : \mathcal{X} \rightarrow \mathcal{X}^2$ , qui à une image  $x$  associe une paire d'image  $(y_h, y_v)$  de différences finies horizontales et verticales sur  $x$ , respectivement :  $\forall k_h = 1, \dots, N_h, k_v = 1, \dots, N_v$ ,

$$\begin{aligned} y_h[k_h, k_v] &= x[k_h, k_v] - x[k_h - 1, k_v] \text{ si } k_h \geq 2, 0 \text{ sinon,} \\ y_v[k_h, k_v] &= x[k_h, k_v] - x[k_h, k_v - 1] \text{ si } k_v \geq 2, 0 \text{ sinon.} \end{aligned} \quad (7)$$

On a alors simplement  $\text{TV}(x) = \|Dx\|_{1,2}$ . Notons que  $\|D^*D\| \leq 8$  [14] et que l'opérateur adjoint de  $D$  associe à une paire d'image  $(y_h, y_v)$  l'image  $x = D^*(y_h, y_v)$  avec <sup>1</sup>

$$\begin{aligned} x[k_h, k_v] &= \{y_h[k_h, k_v] - y_h[k_h + 1, k_v] \text{ si } k_h \leq N_h - 1, \\ & y_h[k_h, k_v] \text{ sinon}\} + \{y_v[k_h, k_v] - y_v[k_h, k_v + 1] \text{ si } \\ & k_v \leq N_v - 1, y_v[k_h, k_v] \text{ sinon}\}. \end{aligned} \quad (8)$$

On voit ainsi que le problème (4) peut se mettre sous la forme (1), avec :

1. L'opérateur  $D^*$  ainsi défini n'est pas exactement l'adjoint de  $D$ , mais il s'écrit plus simplement et donne exactement les mêmes résultats, pour peu que la variable duale  $y$  ait ses valeurs initialisées à zéro dans l'algorithme.

- $F(x) = \frac{1}{2} \|Ax - v\|^2$ , dont le gradient est  $\nabla F(x) = A^*(Ax - v)$ , de constante de Lipschitz  $\beta = \|A\|^2$ .
- $G(x) = \{0 \text{ si } x \in \Omega, +\infty \text{ sinon}\}$ .  $\text{prox}_G$  est simplement la projection vers l'élément le plus proche (au sens de la norme de  $\mathcal{X}$ ) dans  $\Omega$ .
- $M = 1$  (on omet donc les indices  $m$  par simplicité),  $\mathcal{Y} = \mathcal{X}^2$ ,  $\lambda \cdot \text{TV} = H \circ L$  avec  $H(y) = \lambda \|y\|_{1,2}$ ,  $L = D$ .

On peut remarquer que le problème (4) peut se mettre sous la forme (1) d'une manière différente, en posant  $F = 0$ ,  $M = 2$ ,  $H_1 = \frac{1}{2} \|\cdot - v\|^2$ ,  $L_1 = A$ , et  $H_2 \circ L_2 = \lambda \cdot \text{TV}$ . Ces deux formulations donnent des algorithmes différents. Savoir lequel est le plus efficace dépend naturellement du problème considéré et de la manière dont les algorithmes sont implémentés. Ceci étant, nous pensons que la version donnée plus haut, avec  $F = \frac{1}{2} \|A \cdot -v\|^2$ , est la plus efficace en règle générale. L'algorithme est en effet sériel dans ce cas, c'est-à-dire que l'étape de descente de gradient par rapport à  $F$  met à jour la variable  $x$  avant application de l'opérateur proximal de  $G$ , ce qui améliore la solution avant la mise à jour de la variable duale  $y$ . Au contraire, les mises à jour des variables  $y_m$  se font de manière indépendante et antagoniste, avant d'être essentiellement moyennées pour la mise à jour de  $x$ . Pour la même raison, la présence de la fonction  $G$  est utile dans la formulation (1), plutôt que de l'assimiler à une fonction  $H_m$  avec  $L_m = \text{Id}$ .

Notons qu'il existe dans la littérature un algorithme différent du nôtre et permettant aussi de résoudre (4) avec des opérations explicites, dans le cas  $\Omega = \mathcal{X}$  [23].

Pour illustrer l'approche, considérons le problème classique de la déconvolution, illustré en figure 1 : l'image  $v$  disponible est supposée être une version floue et corrompue par du bruit blanc additif gaussien d'une image inconnue, que l'on cherche à estimer. L'image restaurée est obtenue en résolvant le problème (4) à l'aide de l'algorithme proposé. L'opérateur de dégradation  $A$  correspond à une convolution avec un filtre passe-bas, avec des conditions de bord symétriques (on a  $\|A\| = 1$ ). L'ensemble  $\Omega$  est celui des images à valeurs dans  $[0, 255]$ . Un code Matlab implémentant la méthode et générant les images de la figure 1 est disponible sur la page web de l'auteur.

### 4 Conclusion

Nous avons proposé un nouvel algorithme, avec les résultats de convergence associés, pour résoudre une classe générale de problèmes d'optimisation convexe. L'algorithme, de type primal-dual, procède par éclatement : les opérateurs impliqués sont appelés individuellement et explicitement.

L'avantage principal de l'algorithme réside dans sa généralité et sa simplicité d'implémentation. Cela le rend particulièrement approprié pour prototyper de nouvelles méthodes, par exemple pour tester différents type de régularisation pour des problèmes inverses. Il est certain que pour de nombreux problèmes, il est possible d'exploiter la structure pour obtenir un algorithme plus rapide que l'algorithme « universel » proposé.

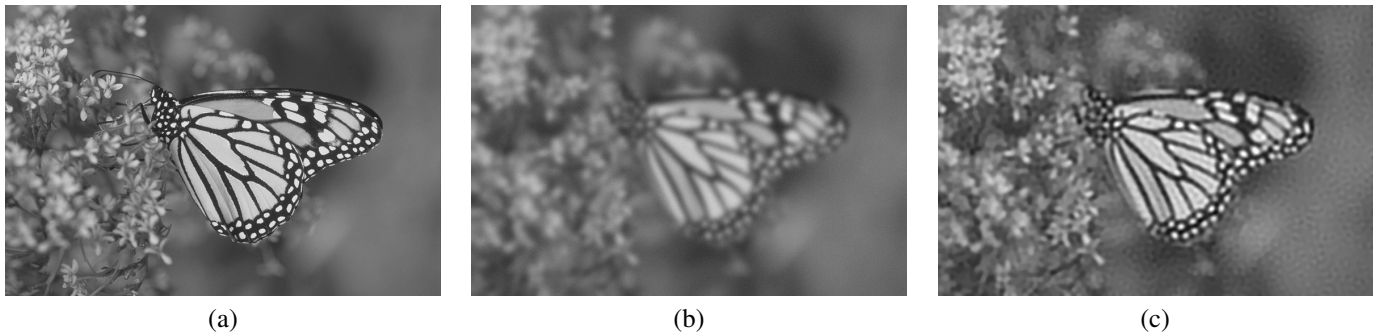


FIGURE 1 – Exemple de déconvolution : à partir de l’image (b) qui est une version dégradée (floue et bruitée) de l’image inconnue (a), on obtient l’image déconvoluée (c), solution du problème (4) avec  $\lambda = 0.02$ , par l’algorithme proposé, stoppé après 300 itérations. Le filtre de flou est un filtre gaussien d’écart-type 5 et le bruit est blanc gaussien d’écart-type 3. Les paramètres utilisés sont  $\sigma = \lambda$ ,  $\tau = 0.99/(1/2 + 8\sigma)$ ,  $\rho = 1$ .

C’est pourquoi nous réservons la comparaison des vitesses de convergence (et de l’espace mémoire requis) de différents algorithmes applicables à différents problèmes-type pour une publication tutorielle plus étendue ultérieure.

L’algorithme peut être généralisé dans le cadre d’une métrique variant à chaque itération [24] ou accéléré dans certains cas grâce à des préconditionneurs [25]. En outre, l’étude théorique de la vitesse de convergence et la possibilité éventuelle d’accélération pour les algorithmes primaux-duaux reste un thème de recherche largement inexploré. Nous aimerions étudier ces questions dans le futur.

## Références

- [1] L. Chaari, E. Chouzenoux, N. Pustelnik, C. Chaux et S. Moussaoui, “OPTIMED : Optimisation itérative pour la résolution de problèmes inverses de grande taille”, *Traitement du signal*, vol. 28, n° 3–4, p. 329–374, 2011.
- [2] S. Becker, E. Candès et M. Grant, “Templates for convex cone problems with applications to sparse signal recovery”, *Mathematical Programming Computation*, vol. 3, n° 3, p. 165–218, sept. 2011.
- [3] S. Anthoine, J. F. Aujol, C. Mélot et Y. Boursier, “Some proximal methods for Poisson intensity CBCT and PET”, *Inverse Problems and Imaging*, vol. 6, n° 4, nov. 2012.
- [4] E. Y. Sidky, J. H. Jørgensen et X. Pan, “Convex optimization problem prototyping for image reconstruction in computed tomography with the Chambolle–Pock algorithm”, *Physics in Medicine and Biology*, vol. 57, n° 10, p. 3065–3091, 2012.
- [5] F. Bach, R. Jenatton, J. Mairal et G. Obozinski, “Optimization with sparsity-inducing penalties”, *Foundations and Trends in Machine Learning*, vol. 4, n° 1, p. 1–106, 2012.
- [6] E. J. Candès, X. Li, Y. Ma et J. Wright, “Robust principal component analysis?”, *Journal of ACM*, vol. 58, n° 1, p. 1–37, 2009.
- [7] D. Cremers, T. Pock, K. Kolev et A. Chambolle, “Convex relaxation techniques for segmentation, stereo and multiview reconstruction”, dans *Markov Random Fields for Vision and Image Processing*. MIT Press, 2011.
- [8] J. Lellmann, D. Breitenreicher et C. Schnörr, “Fast and exact primal-dual iterations for variational problems in computer vision”, dans *Proc. of ECCV: Part II*, Heraklion, Crete, Greece, 2010, p. 494–505.
- [9] J.-L. Starck, F. Murtagh et J. Fadili, *Sparse Image and Signal Processing: Wavelets, Curvelets, Morphological Diversity*, Cambridge University Press, 2010.
- [10] A. Chambolle, V. Caselles, D. Cremers, M. Novaga et T. Pock, “An introduction to total variation for image analysis”, dans *Theoretical Foundations and Numerical Methods for Sparse Recovery*. 2010, vol. 9, p. 263–340, De Gruyter, Radon Series Comp. Appl. Math.
- [11] H. H. Bauschke et P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, Springer, New York, 2011.
- [12] P. L. Combettes et J.-C. Pesquet, “Proximal splitting methods in signal processing”, dans *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, H. H. Bauschke, R. Burachik, P. L. Combettes, V. Elser, D. R. Luke et H. Wolkowicz, Eds. Springer-Verlag, New York, 2010.
- [13] C. Chaux, P. L. Combettes, J.-C. Pesquet et V. R. Wajs, “A variational formulation for frame based inverse problems”, *Inverse Problems*, vol. 23, p. 1495–1518, juin 2007.
- [14] A. Chambolle et T. Pock, “A first-order primal–dual algorithm for convex problems with applications to imaging”, *Journal of Mathematical Imaging and Vision*, vol. 40, n° 1, p. 120–145, 2011.
- [15] L. M. Briceño-Arias et P. L. Combettes, “A monotone+skew splitting model for composite monotone inclusions in duality”, *SIAM J. Optim.*, vol. 21, n° 4, p. 1230–1250, 2011.
- [16] H. Raguet, J. Fadili et G. Peyré, “Generalized forward–backward splitting”, *SIAM Journal on Imaging Sciences*, 2013, à paraître.
- [17] J.-C. Pesquet et N. Pustelnik, “A parallel inertial proximal optimization method”, *Pacific Journal of Optimization*, vol. 8, n° 2, p. 273–305, avr. 2012.
- [18] P. L. Combettes et J.-C. Pesquet, “Primal–dual splitting algorithm for solving inclusions with mixtures of composite, Lipschitzian, and parallel-sum type monotone operators”, *Set-Valued and Variational Analysis*, vol. 20, n° 2, p. 307–330, 2012.
- [19] L. M. Briceño-Arias, “Forward–Douglas–Rachford splitting and forward–partial inverse method for solving monotone inclusions”, preprint arXiv:1212.5942, déc. 2012.
- [20] P. Machart, S. Anthoine et L. Baldassarre, “Optimal computational trade-off of inexact proximal methods”, preprint arXiv:1210.5034, 2012.
- [21] L. Condat, “A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms”, *J. Optimization Theory and Applications*, 2013, à paraître.
- [22] B. C. Vũ, “A splitting algorithm for dual monotone inclusions involving cocoercive operators”, *Advances in Computational Mathematics*, vol. 38, n° 3, p. 667–681, avr. 2013.
- [23] I. Loris et C. Verhoeven, “On a generalization of the iterative soft-thresholding algorithm for the case of non-separable penalty”, *Inverse Problems*, vol. 27, n° 12, 2011.
- [24] P. L. Combettes et B. C. Vũ, “Variable metric forward–backward splitting with applications to monotone inclusions in duality”, preprint arXiv:1206.6791, 2012.
- [25] T. Pock et A. Chambolle, “Diagonal preconditioning for first order primal–dual algorithms in convex optimization”, dans *Proc. of ICCV*, nov. 2011.