Alterner les descentes proximales application au mod`ele ROF

Introduction	
6.1 Le modèle Rudin-Osher	Fatemi
6.1.1 Position du problème	
6.1.2 Cas 1D : complexité e	$n O(N) \dots \dots$
6.1.3 Cas 2D : stratégies d'	
6.2 Théorie des descentes a	ternées multiples
6.2.1 Position du problème	
$6.2.2 \text{Algorithme} \dots .$	
$6.2.3 \text{Accélération} \dots \dots$	
6.2.4 Application au modèl	e ROF
6.3 Résultats expérimentau	«
6.3.1 Choix de la référence	
6.3.2 Descentes alternées p	our l'éclatement sur carrés
6.3.3 Comparaison avec d'a	utres méthodes $\ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots 225$
$6.3.4 \text{Discussion} \dots \dots$	
$Conclusion \ldots \ldots \ldots \ldots \ldots$	

Introduction

L'acquisition de tout signal numérique s'accompagne inévitablement de *bruit*, c'està-dire d'une dégradation de signal original. Le débruitage est donc un sujet central du traitement du signal. Une bonne méthode de débruitage doit corriger la dégradation observée tout en préservant tant que possible les caractéristiques fondamentales du signal acquis.

Parmi les méthodes les plus connues de débruitage d'images, la méthode ROF repose sur une formulation variationnelle du problème. Elle consiste à définir une fonctionnelle d'énergie à deux termes, dont l'un est la norme TV du signal. Ce modèle est populaire car la norme TV est une bonne norme pour les images, en ce sens que les images naturelles ont généralement une faible norme TV. Ainsi, rechercher le signal le plus proche du signal observé et de norme TV la plus faible possible est une manière naturelle de débruiter. L'intérêt principal de l'utilisation de la norme TV est qu'elle préserve les discontinuités des images, ce que ne parviennent pas toujours à faire les autres méthodes de débruitage. Malheureusement, le débruitage d'une image est généralement une étape préliminaire à un autre traitement, et ne constitue pas une fin en soi. Or, la minimisation de la fonctionnelle ROF n'est pas toujours très efficace. Il a été montré que ce problème se résout en O(N) pour des signaux 1D, mais de tels algorithmes ne sont pas généralisables pour des dimensions supérieures. Néanmoins, des stratégies dites d'éclatement ont été introduites pour exploiter cette efficacité du cas 1D dans le problème 2D (qui est le cas des images en niveaux de gris). Ces algorithmes sont en particulier parallélisables, car ils décomposent le problème 2D en plusieurs problèmes 1D indépendants.

L'objectif de ce chapitre est de généraliser ces méthodes au cas de la couleur. Expérimentalement, l'application simple de ces algorithmes sur des images couleurs ne produit pas des résultats satisfaisants. C'est pourquoi nous en proposons une modification, dont nous montrons qu'elle converge. Une comparaison expérimentale de l'algorithme proposé avec d'autres algorithmes permet d'en noter la performance.

6.1 Le modèle Rudin-Osher-Fatemi

6.1.1 Position du problème

Débruitage pur Le modèle ROF (RUDIN-OSHER-FATEMI) a été introduit en 1992 [7] comme modèle de débruitage pur d'un signal. Par débruitage *pur*, on sous-entend qu'aucune déconvolution n'est réalisée sur le signal (comme ça peut être le cas avec les problèmes de déconvolution). Le modèle de formation du signal est donc

$$g = u + n$$

où $g: \Omega \subset \mathbb{R}^n \to \mathbb{R}^m$ est le signal observé, $u: \Omega \subset \mathbb{R}^n \to \mathbb{R}^m$ le signal idéal (inconnu) et $n: \Omega \subset \mathbb{R}^n \to \mathbb{R}^m$ un bruit blanc gaussien additif (inconnu également). L'objectif est donc d'estimer le signal idéal à partir de son observation bruitée g. Le débruitage d'une image en niveaux de gris correspond au cas n = 2 et m = 1 tandis que le débruitage d'une image en couleurs (RGB par exemple) correspond au cas n = 2 et m = 3.

Les méthodes les plus simples (et certainement les plus utilisées) procèdent par un filtrage linéaire de l'image. En d'autres termes, pour estimer la valeur u(x) d'un pixel x, on calcule la moyenne de l'image bruitée g sur un certain voisinage du pixel en question. Si ce voisinage est un voisinage spatial (une fenêtre centrée en x par exemple), alors le résultat est généralement une image certes débruitée, mais floue. Une stratégie pour éviter cet écueil a été proposée avec l'utilisation des *NL-means* (moyennage non local), qui consiste à considérer un voisinage dans l'espace des couleurs : on moyenne des pixels qui présentent le même aspect visuel sur une petite fenêtre. Cette approche permet de mieux préserver les discontinuités de l'image, mais implique de trouver pour tout pixel suffisamment de répliques pour pouvoir le débruiter. Ce genre de méthode repose également sur un nombre important de paramètres, qui sont nécessaires pour définir les différentes proximités.

Le défi principal du débruitage pur est donc de parvenir à supprimer (autant que possible) le bruit, tout en préservant les discontinuités de l'image. Le cadre de travail le plus naturel est donc l'espace BV des fonctions à variations bornées [6]. Dans le modèle ROF, l'idée centrale est de trouver une estimation du signal de norme TV minimale, étant entendu que les images dites *naturelles* présentent une faible norme TV.

Modèle variationnel Le modèle ROF est un modèle variationnel, où la fonctionnelle d'énergie comporte deux termes. Le premier terme est un terme d'attache aux données ; il est défini par la norme L^2 pour gérer le bruit blanc gaussien. Le second terme est un terme de régularisation ; ainsi qu'on l'a déjà évoqué, dans le modèle ROF, il s'agit de la norme TV. On cherche donc à résoudre le problème

$$\min_{v \in \mathrm{BV}(\Omega;\mathbb{R}^m)} \left\{ \frac{1}{2\lambda} \int_{\Omega} |v(x) - g(x)|^2 \,\mathrm{d}x + \mathrm{TV}(v) \right\}$$
(6.1)

où $\lambda > 0$ est un paramètre permettant de modifier les importances respectives du terme d'attache aux données et du terme de régularité. Plus λ est grand, plus la régularisation TV est importante. Il est à noter que la présence du terme quadratique rend ce problème fortement convexe, donc la solution est unique.

Dans leur article original [7], les auteurs minimisent cette fonctionnelle d'énergie grâce à des équations aux dérivées partielles non linéaires. Nous allons montrer dans ce qui suit que d'autres approches plus efficaces peuvent être envisagées pour résoudre ce problème.

6.1.2 Cas 1D : complexité en O(N)

Commençons par considérer le cas simple d'un signal discret 1D (n = 1 et m = 1).

Problème discret La version discrète du modèle ROF est donnée par

$$\min_{v^h \in \mathbb{R}^N} \left\{ \frac{1}{2\lambda} \sum_{i=0}^{N-1} (v^h_i - g^h_i)^2 + \sum_{i=0}^{N-1} |(\nabla^h v^h)_i| \right\}$$
(6.2)

où le signal observé est un vecteur réel $g^h = (g_i^h)_{i \in [\![0]; N-1]\!]}$ de taille N et où le vecteur dérivée $\nabla^h v^h = ((\nabla^h v^h)_i)_{i \in [\![0]; N-1]\!]}$ est défini par les différences finies

$$(\nabla^h v^h)_i = \begin{cases} v_{i+1}^h - v_i^h & \text{si } i < N-1\\ 0 & \text{sinon.} \end{cases}$$

Notons que le problème (6.3) peut encore s'écrire sous la forme

$$\min_{v^h \in \mathbb{R}^N} \left\{ \frac{1}{2\lambda} \| v^h - g^h \|_2^2 + \mathrm{TV}_{1\mathrm{D}}^h(v^h) \right\}$$
(6.3)

où on définit la norme TV discrète d'un signal 1D v^h par $\mathrm{TV}_{1\mathrm{D}}^h(v^h) = \|\nabla^h v^h\|_1$. On en déduit que résoudre le problème ROF 1D revient essentiellement à évaluer l'opérateur proximal associé à la fonction $\mathrm{TV}_{1\mathrm{D}}^h$.

Algorithme de Condat Proposé en 2013 par Laurent CONDAT [2], cette méthode repose sur la caractérisation des *sur-solutions* et *sous-solutions* du problème (6.3), c'està-dire des majorants et des minorants de la solution, qui satisfont un certain nombre de conditions. Celles-ci découlent de l'équation d'EULER associé au problème étudié. En faisant des hypothèses de constance locale de la solution, on construit progressivement des sur- et sous-solutions, jusqu'à ce qu'une impossibilité apparaisse (typiquement, la sur-solution passant sous la solution ou la sous-solution passant au-dessus de la solution). On peut alors identifier le point jusqu'auquel les hypothèses sont correctes, puis reprendre l'estimation de la solution à partir de ce point. La complexité de cet algorithme est de O(N) dans la majorité des cas, mais est moins bonne dans le pire des cas. Algorithme de Johnson L'algorithme proposé par Nicholas JOHNSON en 2010 [5] repose quant à lui sur la programmation dynamique. Le principe est de transformer le problème global (6.3) en une série de problèmes locaux (définis sur les *i* premiers coefficients du vecteur). L'idée est de considérer l'estimation d'un vecteur de *i* points comme la minimisation d'une énergie à trois termes : l'énergie du dernier point *i*, l'énergie du vecteur privé de ce point, et l'énergie d'interaction entre ces deux composantes du vecteur initial, qui ne dépendant que du *i*-ème et du *i* – 1-ème point. Si l'énergie optimale du vecteur composé des *i* – 1 premiers points peut être rapidement calculée quelle que soit la configuration testée, alors le problème est simplifié, car il s'agit uniquement de trouver le *i*-ème point, les *i* – 1 premiers points étant alors pris optimaux pour ce choix. JOHNSON montre que ce schéma de résolution repose en pratique entièrement sur la recherche du zéro d'une fonction affine par morceaux qui est progressivement mise à jour. Or, l'encodage d'une telle fonction est peu coûteuse, car il s'agit uniquement d'en stocker les points de rupture. La complexité de l'algorithme est en O(N).

Malheureusement, ces deux approches exploitent des propriétés liées à l'unidimensionnalité du signal, et ne peuvent donc pas être étendues à des cas plus généraux (n > 1 ou m > 1).

6.1.3 Cas 2D : stratégies d'éclatement

Intéressons-nous à présent au cas des images en niveaux de gris, qui correspond au cas n = 2 et m = 1 des signaux 2D. L'idée des stratégies d'éclatement est de transformer le problème considéré en une série de sous-problèmes que l'on sait résoudre de manière efficace. Dans le cas de l'éclatement horizontal/vertical par exemple, on pourra exploiter l'efficacité des algorithmes de CONDAT et de JOHNSON pour résoudre les sous-problèmes qui apparaissent.

Les approches étudiées dans ce paragraphe repose sur la méthode d'éclatement de DYKSTRA, présentée au chapitre 1. On parle également parfois de *descentes par coordonnées*, car, dans le cas de l'éclatement par ligne/colonne par exemple, il s'agit essentiellement de traiter chaque coordonnée successivement. Cette section reprend en grande partie la démarche déjà présentée dans [1].

Problème discret Le problème discret correspondant prend la forme générique suivante :

$$\min_{v^h \in \mathbb{R}^{N_x \times N_y}} \left\{ \frac{1}{2\lambda} \sum_{i=0}^{N_x - 1} \sum_{j=0}^{N_y - 1} (v^h_{i,j} - g^h_{i,j})^2 + \sum_{i=0}^{N_x - 1} \sum_{j=0}^{N_y - 1} \| (\nabla^h v^h)_{i,j} \|_k \right\}.$$
 (6.4)

Le gradient $\nabla^h v^h$ de l'image v^h est cette fois défini comme suit

$$(\nabla^h v^h)_{i,j} = \begin{pmatrix} (\delta^h_x v^h)_{i,j} \\ (\delta^h_y v^h)_{i,j} \end{pmatrix}$$

où les différences finies sont données par

$$(\delta_x^h v^h)_{i,j} = \begin{cases} v_{i+1,j}^h - v_{i,j}^h & \text{si } i < N_x - 1\\ 0 & \text{sinon} \end{cases} \text{ et } (\delta_y^h v^h)_{i,j} = \begin{cases} v_{i,j+1}^h - v_{i,j}^h & \text{si } j < N_y - 1\\ 0 & \text{sinon.} \end{cases}$$

On suppose ici que les images sont des matrices réelles de taille $N_x \times N_y$, d'indices (i,j) appartenant au carré $[0; N_x - 1] \times [0; N_y - 1]$. Pour la version discrète de la norme TV,

on peut choisir k = 2, ce qui revient à considérer la version isotrope de TV :

$$TV_{iso}^{h}(v^{h}) = \sum_{i=0}^{N_{x}-1} \sum_{j=0}^{N_{y}-1} \sqrt{(\delta_{x}^{h}v^{h})_{i,j}^{2} + (\delta_{y}^{h}v^{h})_{i,j}^{2}}$$

ou k = 1, auquel cas on considère la version anisotrope (qui découple les directions horizontales et verticales) :

$$TV_{aniso}^{h}(v^{h}) = \sum_{i=0}^{N_{x}-1} \sum_{j=0}^{N_{y}-1} |(\delta_{x}^{h}v^{h})_{i,j}| + |(\delta_{y}^{h}v^{h})_{i,j}|.$$

Éclatement horizontal/vertical ou ligne/colonne L'idée ici est de transformer le problème discret (6.4) en une série de problèmes ROF 1D, de sorte de pouvoir utiliser un des deux algorithmes présentés plus haut. On va pour cela considérer la version anisotrope de TV, ce qui revient à s'intéresser au problème

$$\min_{v^{h} \in \mathbb{R}^{N_{x} \times N_{y}}} \left\{ \frac{1}{2\lambda} \| v^{h} - g^{h} \|_{2}^{2} + \mathrm{TV}_{\mathrm{h}}^{h}(v^{h}) + \mathrm{TV}_{\mathrm{v}}^{h}(v^{h}) \right\}$$
(6.5)

où on pose $\mathrm{TV}_{h}^{h}(v^{h}) = \|\delta_{x}^{h}v^{h}\|_{1}$ et $\mathrm{TV}_{v}^{h}(v^{h}) = \|\delta_{y}^{h}v^{h}\|_{1}$. Le lagrangien de ce problème est composé de trois termes, un terme différentiable et deux termes dont l'opérateur proximal peut être évalué, puisqu'il ne s'agit ni plus ni moins que de normes TV 1D (horizontale et verticale). On peut donc appliquer la méthode d'éclatement de DYKSTRA. Il s'agit de considérer le problème dual équivalent

$$\min_{x^{h}, y^{h} \in \mathbb{R}^{N_{x} \times N_{y}}} \left\{ (\mathrm{TV}_{\mathrm{h}}^{h})^{*}(y^{h}) + (\mathrm{TV}_{\mathrm{v}}^{h})^{*}(x^{h}) + \frac{1}{2\lambda} \left\| \lambda(x^{h} + y^{h}) - g^{h} \right\|_{2}^{2} \right\}.$$
(6.6)

Les solutions v^* et (x^*, y^*) des problèmes primal et dual sont reliées par la relation

$$v^* = g^h - \lambda \left(x^* + y^* \right).$$

On résout le problème dual par minimisations partielles alternées :

$$\begin{cases} x_{n+1}^{h} = \underset{x \in \mathbb{R}^{N_{x} \times N_{y}}}{\operatorname{argmin}} \left\{ (\mathrm{TV}_{h}^{h})^{*}(x) + \frac{1}{2\lambda} \left\| \lambda(x+y_{n}^{h}) - g^{h} \right\|_{2}^{2} \right\} \\ y_{n+1}^{h} = \underset{y \in \mathbb{R}^{N_{x} \times N_{y}}}{\operatorname{argmin}} \left\{ (\mathrm{TV}_{v}^{h})^{*}(y) + \frac{1}{2\lambda} \left\| \lambda(x_{n+1}^{h} + y) - g^{h} \right\|_{2}^{2} \right\}.$$
(6.7)

L'image v_n^h est donnée à chaque itération par $v_n^h = g^h - \lambda (x_n^h + y_n^h)$. On reconnaît la définition des opérateurs proximaux associés à $(\mathrm{TV}_h^h)^*$ et $(\mathrm{TV}_v^h)^*$, que l'on sait calculer à l'aide de l'identité de MOREAU et de l'algorithme de CONDAT ou l'algorithme de JOHNSON (appliqués en parallèle sur toutes les lignes ou toutes les colonnes du problème). Ces deux mises-à-jours sont en effet respectivement des problèmes de N_x (resp. N_y) minimisations de la forme (6.3) entièrement séparés sur les lignes (resp. les colonnes).

Éclatement carrés pairs/impairs Restons pour l'instant dans le cas de la TV anisotrope k = 1. Remarquons qu'il est possible de décomposer la norme TV en deux termes, l'un portant sur les carrés de coin supérieur gauche de coordonnées paires et l'autre portant sur les carrés de coin supérieur gauche de coordonnées impaires. Plus précisément, commençons par définir le carré de coin supérieur gauche (i,j) en posant pour $(i,j) \in [\![-1\,;\,N_x-1\,]\!] \times [\![-1\,;\,N_y-1\,]\!]$ l'opérateur d'extraction $S_{i,j},$ qui est défini par

$$S_{i,j}v^h = (v^h_{i,j}, v^h_{i,j+1}, v^h_{i+1,j+1}, v^h_{i+1,j})$$

si $i \in [0; N_x - 2]$ et $j \in [0; N_y - 2]$ (autrement, si le carré est entièrement inclus dans le domaine de l'image). Si $i \in [0; N_x - 2]$ mais que $j \in \{-1, N_y - 1\}$, alors

$$S_{i,-1}v^h = (v^h_{i,0}, v^h_{i+1,0})$$
 et $S_{i,N_y-1}v^h = (v^h_{i,N_y-1}, v^h_{i+1,N_y-1}).$

De même, si $j \in [0; N_y - 2]$ mais que $i \in \{-1, N_x - 1\}$, alors

$$S_{-1,j}v^h = (v^h_{0,j+1}, v^h_{0,j})$$
 et $S_{N_x-1,j}v^h = (v^h_{N_x-1,j}, v^h_{N_x-1,j+1})$

Enfin, dans les autres cas, on a

$$S_{-1,-1}v^{h} = (v_{0,0}^{h}), \qquad S_{N_{x}-1,-1}v^{h} = (v_{N_{x}-1,0}^{h}),$$
$$S_{-1,N_{y}-1}v^{h} = (v_{0,N_{y}-1}^{h}) \quad \text{et} \quad S_{N_{x}-1,N_{y}-1}v^{h} = (v_{N_{x}-1,N_{y}-1}^{h})$$

Désormais, on dira que le carré de coin supérieur gauche (i,j) est pair (resp. impairs) si i et j sont pairs (resp. impairs). Si on ne considère que les carrés pairs (respectivement impairs), alors on constate d'une part que tous les pixels $u_{i,j}$ sont présents exactement une fois et, d'autre part, que les carrés (au sens large, c'est-à-dire même réduits à un segment ou à un point) sont tous disjoints.

On définit ensuite l'opérateur D, en posant pour tout carré $S = {}^{t}(s_1, s_2, s_3, s_4)$

$$Ds = {}^{t}(s_2 - s_1, s_3 - s_2, s_4 - s_3, s_1 - s_4)$$

puis $Ds = s_2 - s_1$ pour tout carré $S = {}^{t}(s_1, s_2)$ et enfin Ds = 0 si $s = (s_1) \in \mathbb{R}$. Son adjoint sur \mathbb{R}^4 , noté D^* , est donné pour tout $\xi = (\xi_1, \xi_2, \xi_3, \xi_4)$ par

$$D^*\xi = (\xi_4 - \xi_1, \xi_1 - \xi_2, \xi_2 - \xi_3, \xi_3 - \xi_4)$$

tandis qu'il vaut $D^*\xi = (-\xi,\xi)$ si s n'est pas un véritable carré. Grâce aux remarques faites plus haut, on peut alors vérifier que

$$TV_{aniso}^{h}(v^{h}) = \underbrace{\sum_{\substack{i=0\\i \text{ pair}}}^{N_{x}-1} \sum_{\substack{j=0\\j \text{ pair}}}^{N_{y}-1} \|D(Sv^{h})_{i,j}\|_{1}}_{= TV_{e}^{h}(v^{h})} + \underbrace{\sum_{\substack{i=-1\\i \text{ impair} j \text{ impair}}}^{N_{x}-1} \sum_{\substack{j=-1\\i \text{ impair} j \text{ impair}}}^{N_{y}-1} \|D(Sv^{h})_{i,j}\|_{1}}_{= TV_{o}^{h}(v^{h})}$$

À nouveau, on peut utiliser la méthode d'éclatement de DYKSTRA, ce qui revient cette fois à résoudre le problème dual

$$\min_{x^{h}, y^{h} \in \mathbb{R}^{N_{x} \times N_{y}}} (\mathrm{TV}_{\mathrm{e}}^{h})^{*}(x^{h}) + (\mathrm{TV}_{\mathrm{o}}^{h})^{*}(y^{h}) + \frac{1}{2\lambda} \left\| \lambda(x^{h} + y^{h}) - g^{h} \right\|_{2}^{2}$$
(6.8)

par minimisations partielles alternées :

$$\begin{cases} x_{n+1}^{h} = \underset{x \in \mathbb{R}^{N_{x} \times N_{y}}}{\operatorname{argmin}} \left\{ (\mathrm{TV}_{e}^{h})^{*}(x) + \frac{1}{2\lambda} \left\| \lambda(x+y_{n}^{h}) - g^{h} \right\|_{2}^{2} \right\} \\ y_{n+1}^{h} = \underset{y \in \mathbb{R}^{N_{x} \times N_{y}}}{\operatorname{argmin}} \left\{ (\mathrm{TV}_{o}^{h})^{*}(y) + \frac{1}{2\lambda} \left\| \lambda(x_{n+1}^{h} + y) - g^{h} \right\|_{2}^{2} \right\}.$$
(6.9)

Calculons les deux conjuguées convexes $(\mathrm{TV}_{\mathrm{e}}^{h})^{*}$ et $(\mathrm{TV}_{\mathrm{o}}^{h})^{*}$. Par définition de la conjuguée convexe, on a pour tout $x \in \mathbb{R}^{N_{x} \times N_{y}}$

$$(\mathrm{TV}_{\mathrm{e}}^{h})^{*}(x) = \sup_{v \in \mathbb{R}^{N_{x} \times N_{y}}} \left\{ \langle v, x \rangle - \sum_{\substack{i=0\\i \text{ pair } j \text{ pair}}}^{N_{x}} \sum_{\substack{j=0\\j \text{ pair}}}^{N_{y}} \|D(Sv)_{i,j}\|_{1} \right\}.$$

On peut facilement vérifier que

$$\langle v, x \rangle = \sum_{\substack{i=0\\i \text{ pair } j \text{ pair }}}^{N_x - 1} \sum_{\substack{j=0\\j \text{ pair }}}^{N_y - 1} \langle (Sv)_{i,j}, (Sx)_{i,j} \rangle.$$

Ainsi, pour calculer la conjuguée convexe $(TV_e^h)^*(x)$, il suffit de savoir résoudre

$$\sup_{s \in \mathbb{R}^4} \left\{ \langle s, S_{i,j} x \rangle - \| Ds \|_1 \right\} \quad \text{et} \quad \sup_{s \in \mathbb{R}^2} \left\{ \langle s, S_{i,j} x \rangle - \| Ds \|_1 \right\}$$

suivant la taille du vecteur $(Sx)_{i,j}$ (on ignore le cas trivial où $(Sx)_{i,j}$ est un singleton, car il n'y a alors pas de régularisation). On peut montrer que

$$\sup_{s \in \mathbb{R}^4} \left\{ \langle s, S_{i,j} x \rangle - \| Ds \|_1 \right\} = \begin{cases} 0 & \text{si } x = D^* \xi \text{ et } \|\xi\|_{\infty} \le 1 \\ +\infty & \text{sinon} \end{cases}$$

et
$$\sup_{s \in \mathbb{R}^2} \left\{ \langle s, S_{i,j} x \rangle - \| Ds \|_1 \right\} = \begin{cases} 0 & \text{si } x = D^* \xi \text{ et } \|\xi\|_{\infty} \le 1 \\ +\infty & \text{sinon.} \end{cases}$$

Il s'ensuit que, si on pose $\mathcal{K} = \{D^*\xi \mid ||\xi||_{\infty} \leq 1, \xi \in \mathbb{R}^4 \text{ ou } \xi \in \mathbb{R}^2\}$, alors

$$(\mathrm{TV}_{\mathrm{e}}^{h})^{*}(x) = \sum_{\substack{i=0\\i \text{ pair } j \text{ pair }}}^{N_{x}-1} \sum_{\substack{j=0\\j \text{ pair }}}^{N_{y}-1} \chi_{\mathcal{K}}(S_{i,j}x) \qquad \text{et} \qquad (\mathrm{TV}_{\mathrm{o}}^{h})^{*}(y) = \sum_{\substack{i=-1\\i \text{ impair } j \text{ impair }}}^{N_{x}-1} \sum_{\substack{j=-1\\j \text{ impair }}}^{N_{y}-1} \chi_{\mathcal{K}}(S_{i,j}y).$$

Puisque
$$||g||_2^2 = \sum_{\substack{i=0\\i \text{ pair}}}^{N_x-1} \sum_{\substack{j=0\\j \text{ pair}}}^{N_y-1} ||S_{i,j}g||_2^2 = \sum_{\substack{i=-1\\i \text{ impair}}}^{N_x-1} \sum_{\substack{j=-1\\j \text{ impair}}}^{N_y-1} ||S_{i,j}g||_2^2$$

Р

on en déduit que les minimisations partielles alternées peuvent s'écrire à l'aide de l'opérateur S : pour tout (i, j) pairs,

$$S_{i,j}x_{n+1}^h = \operatorname*{argmin}_{s \in \mathbb{R}^d} \left\{ \chi_{\mathcal{K}}(s) + \frac{1}{2\lambda} \|\lambda s - S_{i,j}(g^h - \lambda y_n^h)\|_2^2 \right\}$$

et pour tout (i,j) impairs,

$$S_{i,j}y_{n+1}^{h} = \operatorname*{argmin}_{s \in \mathbb{R}^{d}} \left\{ \chi_{\mathcal{K}}(s) + \frac{1}{2\lambda} \|\lambda s - S_{i,j}(g^{h} - \lambda x_{n+1}^{h})\|_{2}^{2} \right\}$$

avec dans les deux cas d'égal à la taille de $S_{i,j}g^h$. En particulier, on cherche à résoudre le problème générique sur le carré (en ayant factorisé par λ dans la norme)

$$\min_{s \in \mathbb{R}^4} \left\{ \chi_{\mathcal{K}}(s) + \frac{1}{2} \|s - s_0\|_2^2 \right\} \quad \text{avec } s_0 \in \mathbb{R}^4,$$
$$\min_{\substack{\xi \in \mathbb{R}^4 \\ \|\xi\|_{\infty} \le 1}} \left\{ \frac{1}{2} \|D^*\xi - s_0\|_2^2 \right\}.$$

qui s'écrit aussi

Il est possible de résoudre ce problème de manière exacte en utilisant la méthode de NEWTON, mais [1] assure qu'il est suffisant de procéder à deux minimisations partielles alternées : la première en minimisant l'énergie en (ξ_2,ξ_4) pour (ξ_1,ξ_3) fixées (égales à leur valeur précédente) et la seconde en minimisant l'énergie en (ξ_1,ξ_3) pour (ξ_2,ξ_4) fixées (égales à leurs nouvelles valeurs).

6.2 Théorie des descentes alternées multiples

On propose dans cette section un algorithme qui alterne plusieurs pas de descentes proximales (généralisées) pour résoudre, entre autres, les problèmes duaux posés par la méthode par éclatement de DYKSTRA. On appliquera ensuite cet algorithme au débruitage ROF des images en couleurs RGB.

6.2.1 Position du problème

On s'intéresse dans cette section au problème de la forme

$$\min_{\substack{x \in X \\ y \in Y}} \left\{ f(x) + g(y) + \frac{1}{2} \|Ax + By - c\|^2 \right\}$$
(6.10)

où $f: X \to \mathbb{R}$ et $g: Y \to \mathbb{R}$ sont des fonctions convexes, propres et s.c.i. Les opérateurs linéaires $A: X \to Z$ et $B: Y \to Z$ sont supposés continus, tandis que $c \in Z$.

Pour résoudre ce problème, une première approche consiste à alterner les minimisations partielles, respectivement en x et en y. Néanmoins, la forme des fonctions fet g peuvent rendre ces deux opérations complexes. Si le minimiseur ne possède pas de forme close, alors la minimisation peut se faire de manière approchée, à l'aide par exemple de la méthode de NEWTON. La précision de la résolution est alors cruciale, car, si l'erreur d'approximation est trop importante, le schéma alternatif peut alors perdre se stabilité.

6.2.2 Algorithme

Pour résoudre le problème (6.10), on se propose de considérer une autre approche que celle de la minimisation alternée. Au lieu de minimiser alternativement selon chaque variable x et y, on propose de calculer des pas de descentes proximales. Celles-ci peuvent être basées sur des itérations de BREGMAN pour plus de généralités.

Il s'agit donc de considérer l'algorithme suivant, où K et L sont deux entiers naturels non nuls : on choisit $x_K^O \in X$ et $y_L^0 \in Y$, puis, pour tout $n \ge 0$,

$$\begin{cases} x_0^{n+1} = x_K^n \\ \begin{cases} \forall k \in [\![\,0\,;\,K-1\,]\!] \\ x_{k+1}^{n+1} = \operatorname*{argmin}_{x \in X} \left\{ f(x) + \frac{1}{2} \, \|Ax + By^n - c\|^2 + \frac{1}{2} \, \|x - x_k^{n+1}\|_M^2 \right\} \\ x^{n+1} = \frac{1}{K} \sum_{k=1}^K x_k^{n+1} \quad \text{et} \quad y_0^{n+1} = y_L^n \\ \begin{cases} \forall \ell \in [\![\,0\,;\,L-1\,]\!] \\ y_{\ell+1}^{n+1} = \operatorname*{argmin}_{y \in Y} \left\{ g(y) + \frac{1}{2} \, \|Ax^{n+1} + By - c\|^2 + \frac{1}{2} \, \|y - y_\ell^{n+1}\|_P^2 \right\} \\ y^{n+1} = \frac{1}{L} \sum_{\ell=1}^L y_\ell^{n+1} \end{cases} \end{cases}$$

où M et P sont deux matrices symétriques. Si M et P sont l'identité, alors cet algorithme alterne donc K descentes proximales en x puis L descentes proximales en y, avec des moyennages entre chaque ensemble de descentes. Dans le cas général, les descentes proximales sont remplacées par des itérations de BREGMAN (voir chapitre 1). L'intérêt de cet algorithme se manifeste évidemment si les descentes proximales peuvent être évaluées de manière exacte. Nous allons montrer que cet algorithme converge bien vers la solution du problème (6.10).

DÉMONSTRATION : On commence par considérer une forme plus générale des misesà-jours pour les variables x et y. On initialise d'une part $\hat{x}_0 \in X$, puis on définit les itérations suivantes :

$$\forall k \in [[0; K-1]], \qquad \hat{x}_{k+1} = \operatorname*{argmin}_{x \in X} \left\{ f(x) + \frac{1}{2} \|Ax + B\bar{y} - c\|^2 + \frac{1}{2} \|x - \hat{x}_k\|_M^2 \right\}$$

d'autre part, on initialise avec $\hat{y}_0 \in Y$ et on définit les points

$$\forall \ell \in [\![0]; L-1]\!], \qquad \hat{y}_{\ell+1} = \operatorname*{argmin}_{y \in Y} \left\{ g(y) + \frac{1}{2} \|A\tilde{x} + By - c\|^2 + \frac{1}{2} \|y - \hat{y}_{\ell}\|_P^2 \right\}$$

On pose alors

 et

$$\tilde{x} = \frac{1}{K} \sum_{k=1}^{K} \hat{x}_k$$
 et $\tilde{y} = \frac{1}{L} \sum_{\ell=1}^{L} \hat{y}_\ell$

Conditions d'optimalité Soit $n \ge 0, k \in [[0; K-1]]$ et $\ell \in [[0; L-1]]$. Les conditions nécessaires d'optimalité pour \hat{x}_{k+1} s'écrivent

$$-A^*(A\hat{x}_{k+1} + B\bar{y} - c) - M(\hat{x}_{k+1} - \hat{x}_k) \in \partial f(\hat{x}_{k+1})$$
(6.11)

tandis que celles pour $\hat{y}_{\ell+1}$ sont données par

$$B^*(A\tilde{x} + B\hat{y}_{\ell+1} - c) - N(\hat{y}_{\ell+1} - \hat{y}_{\ell}) \in \partial g(\hat{y}_{\ell+1}).$$
(6.12)

Convexité de f La convexité de la fonction f implique que, pour tout $p \in \partial f(\hat{x}_{k+1})$, on a l'inégalité suivante

$$\forall x \in X, \quad f(x) \ge f(\hat{x}_{k+1}) + \langle p, x - \hat{x}_{k+1} \rangle$$

ce qui implique, grâce à (6.11), que pour tout $x \in X$,

$$f(x) \ge f(\hat{x}_{k+1}) - \langle A^*(A\hat{x}_{k+1} + B\bar{y} - c), x - \hat{x}_{k+1} \rangle - \langle M(\hat{x}_{k+1} - \hat{x}_k), x - \hat{x}_{k+1} \rangle.$$

Utilisons l'identité $-2\langle a,b\rangle = ||a||^2 + ||b||^2 - ||a+b||^2$ pour réécrire l'inégalité précédente en remplaçant le premier produit scalaire :

$$f(x) + \frac{1}{2} \|Ax + B\bar{y} - c\|^{2} \ge f(\hat{x}_{k+1}) + \frac{1}{2} \|A\hat{x}_{k+1} + B\bar{y} - c\|^{2} + \frac{1}{2} \|A(x - \hat{x}_{k+1})\|^{2} - \langle M(\hat{x}_{k+1} - \hat{x}_{k}), x - \hat{x}_{k+1} \rangle.$$
(6.13)

Le produit scalaire restant se décomposant des deux manières suivantes

$$\langle M(\hat{x}_{k+1} - \hat{x}_k), x - \hat{x}_{k+1} \rangle = \langle M(\hat{x}_{k+1} - \hat{x}_k), x - \hat{x}_k \rangle - \|\hat{x}_{k+1} - \hat{x}_k\|_M^2$$

$$\langle M(\hat{x}_{k+1} - \hat{x}_k), x - \hat{x}_{k+1} \rangle = -\|x - \hat{x}_{k+1}\|_M^2 + \langle M(x - \hat{x}_k), x - \hat{x}_{k+1} \rangle,$$

on peut sommer ces deux relations et utiliser la symétrie de M pour établir que

 $2 \langle M(\hat{x}_{k+1} - \hat{x}_k), x - \hat{x}_{k+1} \rangle = -\|\hat{x}_{k+1} - \hat{x}_k\|_M^2 - \|x - \hat{x}_{k+1}\|_M^2 + \|x - \hat{x}_k\|_M^2.$ Ainsi, en revenant à (6.13),

$$f(x) + \frac{1}{2} \|Ax + B\bar{y} - c\|^{2} + \frac{1}{2} \|x - \hat{x}_{k}\|_{M}^{2} \ge f(\hat{x}_{k+1}) + \frac{1}{2} \|A\hat{x}_{k+1} + B\bar{y} - c\|^{2} + \frac{1}{2} \|x - \hat{x}_{k+1}\|_{M}^{2} + \frac{1}{2} \|A(x - \hat{x}_{k+1})\|^{2} + \frac{1}{2} \|\hat{x}_{k+1} - \hat{x}_{k}\|_{M}^{2}.$$

$$(6.14)$$

Moyennage local Si on moyenne les inégalités (6.14) pour k entre 0 et K - 1, on obtient

$$f(x) + \frac{1}{2} \|Ax + B\bar{y} - c\|^2 + \frac{1}{2K} \|x - \hat{x}_0\|_M^2$$

$$\geq \frac{1}{K} \sum_{k=0}^{K-1} f(\hat{x}_{k+1}) + \frac{1}{2K} \sum_{k=0}^{K-1} \|A\hat{x}_{k+1} + B\bar{y} - c\|^2 + \frac{1}{2K} \|x - \hat{x}_K\|_M^2$$

$$+ \frac{1}{2K} \sum_{k=0}^{K-1} \|A(x - \hat{x}_{k+1})\|^2 + \frac{1}{2K} \sum_{k=0}^{K-1} \|\hat{x}_{k+1} - \hat{x}_k\|_M^2$$

où les termes dans la somme des $||x - \hat{x}_k||_M^2$ se télescopent. Ainsi, la convexité des fonctions $f, x \mapsto ||Ax + B\bar{y} - c||^2$ et $x \mapsto ||A(x - \hat{x}_{k+1})||^2$ assure que

$$f(x) + \frac{1}{2} \|Ax + B\bar{y} - c\|^{2} + \frac{1}{2K} \|x - \hat{x}_{0}\|_{M}^{2}$$

$$\geq f(\tilde{x}) + \frac{1}{2} \|A\tilde{x} + B\bar{y} - c\|^{2} + \frac{1}{2K} \|x - \hat{x}_{K}\|_{M}^{2} \qquad (6.15)$$

$$+ \frac{1}{2} \|A(x - \tilde{x})\|^{2} + \frac{1}{2K} \sum_{k=0}^{K-1} \|\hat{x}_{k+1} - \hat{x}_{k}\|_{M}^{2}.$$

Convexité de g et moyennage local En utilisant cette fois la convexité de la fonction g, on montre de même que

$$g(y) + \frac{1}{2} \|A\tilde{x} + By - c\|^{2} + \frac{1}{2L} \|y - \hat{y}_{0}\|_{P}^{2}$$

$$\geq g(\tilde{y}) + \frac{1}{2} \|A\tilde{x} + B\tilde{y} - c\|^{2} + \frac{1}{2L} \|y - \hat{y}_{L}\|_{P}^{2} \qquad (6.16)$$

$$+ \frac{1}{2} \|B(y - \tilde{y})\|^{2} + \frac{1}{2L} \sum_{\ell=0}^{L-1} \|\hat{y}_{\ell+1} - \hat{y}_{\ell}\|_{P}^{2}.$$

Somme de (6.15) et (6.16) Sommons maintenant les deux inégalités (6.15) et (6.16), ce qui donne

$$f(x) + g(y) + \frac{1}{2K} \|x - \hat{x}_0\|_M^2 + \frac{1}{2L} \|y - \hat{y}_0\|_P^2 + \frac{1}{2} \|Ax + B\bar{y} - c\|^2 + \frac{1}{2} \|A\tilde{x} + By - c\|^2 \geq f(\tilde{x}) + g(\tilde{y}) + \frac{1}{2K} \|x - \hat{x}_K\|_M^2 + \frac{1}{2L} \|y - \hat{y}_L\|_P^2 + \frac{1}{2} \|A\tilde{x} + B\bar{y} - c\|^2 + \frac{1}{2} \|A\tilde{x} + B\tilde{y} - c\|^2 + \frac{1}{2} \|A(x - \tilde{x})\|^2 + \frac{1}{2} \|B(y - \tilde{y})\|^2 + \frac{1}{2K} \sum_{k=0}^{K-1} \|\hat{x}_{k+1} - \hat{x}_k\|_M^2 + \frac{1}{2L} \sum_{\ell=0}^{L-1} \|\hat{y}_{\ell+1} - \hat{y}_\ell\|_P^2.$$

$$(6.17)$$

Ajoutons ensuite $||Ax - By - c||^2/2$ et $||B(y - \bar{y})||^2/2$ des deux côtés de l'inégalité, puis réarrangeons les termes :

$$\begin{aligned} f(x) + g(y) &+ \frac{1}{2} \|Ax - By - c\|^2 \\ &+ \frac{1}{2K} \|x - \hat{x}_0\|_M^2 + \frac{1}{2L} \|y - \hat{y}_0\|_P^2 + \frac{1}{2} \|B(y - \bar{y})\|^2 \\ &\geq f(\tilde{x}) + g(\tilde{y}) + \frac{1}{2} \|A\tilde{x} + B\tilde{y} - c\|^2 \\ &+ \frac{1}{2K} \|x - \hat{x}_K\|_M^2 + \frac{1}{2L} \|y - \hat{y}_L\|_P^2 + \frac{1}{2} \|B(y - \tilde{y})\|^2 \\ &+ \frac{1}{2K} \sum_{k=0}^{K-1} \|\hat{x}_{k+1} - \hat{x}_k\|_M^2 + \frac{1}{2L} \sum_{\ell=0}^{L-1} \|\hat{y}_{\ell+1} - \hat{y}_\ell\|_P^2 \\ &+ \frac{1}{2} \|A(x - \tilde{x})\|^2 + \frac{1}{2} \|B(y - \bar{y})\|^2 \\ &- \frac{1}{2} \|Ax + B\bar{y} - c\|^2 - \frac{1}{2} \|A\tilde{x} + By - c\|^2 \\ &+ \frac{1}{2} \|Ax + By - c\|^2 + \frac{1}{2} \|A\tilde{x} + B\bar{y} - c\|^2. \end{aligned}$$

$$(6.18)$$

Simplifions les trois dernières lignes de cette inégalité. Commençons par développer les quatre carrés des deux dernières lignes, ce qui nous donne le produit scalaire

$$-\frac{1}{2} \|Ax + B\bar{y} - c\|^{2} - \frac{1}{2} \|A\tilde{x} + By - c\|^{2} + \frac{1}{2} \|Ax + By - c\|^{2} + \frac{1}{2} \|A\tilde{x} + B\bar{y} - c\|^{2} = \langle A(x - \tilde{x}), B(y - \bar{y}) \rangle.$$
(6.19)

En le combinant avec les deux carrés restants, on obtient $||A(x - \tilde{x}) + B(y - \bar{y})||^2/2$, qui est positif. On peut alors simplifier l'expression de (6.18), en minorant le membre de droite

$$f(x) + g(y) + \frac{1}{2} ||Ax + By - c||^{2} + \frac{1}{2K} ||x - \hat{x}_{0}||_{M}^{2} + \frac{1}{2L} ||y - \hat{y}_{0}||_{P}^{2} + \frac{1}{2} ||B(y - \bar{y})||^{2} \geq f(\tilde{x}) + g(\tilde{y}) + \frac{1}{2} ||A\tilde{x} + B\tilde{y} - c||^{2} + \frac{1}{2K} ||x - \hat{x}_{K}||_{M}^{2} + \frac{1}{2L} ||y - \hat{y}_{L}||_{P}^{2} + \frac{1}{2} ||B(y - \tilde{y})||^{2}.$$

$$(6.20)$$

Moyennage global Notons (x^*, y^*) une solution du problème (6.10) et notons \mathcal{E} son lagrangien. Ensuite, spécifions les différents points des itérations, en choisissant pour tous $0 \le k \le K$ et $0 \le \ell \le L$

$$(\hat{x}_k, \hat{y}_\ell) = (x_k^{n+1}, y_\ell^{n+1}), \quad x^n = \frac{1}{K} \sum_{k=1}^K x_k^n, \quad \bar{y} = y^n = \frac{1}{L} \sum_{\ell=1}^L y_\ell^n, \quad (\tilde{x}, \tilde{y}) = (x^{n+1}, y^{n+1})$$

En moyennant les inégalités (6.20) pour n entre 0 et N-1, les termes $||B(y-y^n)||^2$ se

télescopent, ce qui entraîne

$$\begin{split} \mathcal{E}(x,y) &+ \frac{1}{2KN} \sum_{n=0}^{N-1} \|x - x_0^{n+1}\|_M^2 + \frac{1}{2LN} \sum_{n=0}^{N-1} \|y - y_0^{n+1}\|_P^2 + \frac{1}{2N} \|B(y - y^0)\|^2 \\ &\geq \frac{1}{N} \sum_{n=1}^N \mathcal{E}(x^n, y^n) + \frac{1}{2KN} \sum_{n=0}^{N-1} \|x - x_K^{n+1}\|_M^2 + \frac{1}{2LN} \sum_{n=0}^{N-1} \|y - y_L^{n+1}\|_P^2 \\ &+ \frac{1}{2N} \|B(y - y^N)\|^2. \end{split}$$

Si on initialise chaque étape globale $n \geq 1$ en posant

$$x_0^n = x_K^{n-1} \qquad \text{et} \qquad y_0^n = y_L^{n-1}$$

alors on obtient quatre sommes télescopiques

$$\mathcal{E}(x,y) + \frac{1}{2KN} \|x - x_K^0\|_M^2 + \frac{1}{2LN} \|y - y_L^0\|_P^2 + \frac{1}{2N} \|B(y - y^0)\|^2$$

$$\geq \frac{1}{N} \sum_{n=1}^N \mathcal{E}(x^n, y^n) + \frac{1}{2KN} \|x - x_K^N\|_M^2 + \frac{1}{2LN} \|y - y_L^N\|_P^2 + \frac{1}{2N} \|B(y - y^N)\|^2.$$
(6.21)

On introduit alors les deux moyennes

$$X^{N} = \frac{1}{N} \sum_{n=1}^{N} x^{n}$$
 et $Y^{N} = \frac{1}{N} \sum_{n=1}^{N} y^{n}$

puis, en utilisant l'inégalité de JENSEN dans (6.21) et en appliquant cette dernière à $(x,y) = (x^*,y^*)$, on obtient

$$\begin{aligned} \mathcal{E}(x^*, y^*) &+ \frac{1}{2KN} \|x^* - x_K^0\|_M^2 + \frac{1}{2LN} \|y^* - y_L^0\|_P^2 + \frac{1}{2N} \|B(y^* - y^0)\|^2 \\ &\geq \mathcal{E}(X^N, Y^N) + \frac{1}{2KN} \|x^* - x_K^N\|_M^2 + \frac{1}{2LN} \|y^* - y_L^N\|_P^2 + \frac{1}{2N} \|B(y^* - y^N)\|^2 \end{aligned}$$

qui se lit également

$$\mathcal{E}(X^{N}, Y^{N}) - \mathcal{E}(x^{*}, y^{*}) \leq \frac{1}{N} \left\{ \frac{\|x^{*} - x_{K}^{0}\|_{M}^{2} - \|x^{*} - x_{K}^{N}\|_{M}^{2}}{2K} + \frac{\|y^{*} - y_{L}^{0}\|_{P}^{2} - \|y^{*} - y_{L}^{N}\|_{P}^{2}}{2L} + \frac{\|B(y^{*} - y^{0})\|^{2} - \|B(y^{*} - y^{N})\|^{2}}{2} \right\}$$

$$\left. + \frac{\|B(y^{*} - y^{0})\|^{2} - \|B(y^{*} - y^{N})\|^{2}}{2} \right\}$$

$$(6.22)$$

ce qui prouve que la suite des (X_N, Y_N) définit une suite minimisante pour le lagrangien \mathcal{E} , avec une erreur décroissant en O(1/N).

6.2.3 Accélération

On propose maintenant une variante de l'algorithme, en introduisant une accélération de type FISTA pour obtenir une convergence plus rapide, donné par les mises-àjours suivantes pour $n \ge 0$:

$$\begin{cases} x_0^{n+1} = x^n + \frac{1}{t_{n+1}} \left(x^{n-1} - x^n \right) + \frac{t_n}{t_{n+1}} \left(x_K^n - x^{n-1} \right) \\ \begin{cases} \forall k \in [\![0]; K-1]\!] \\ x_{k+1}^{n+1} = \operatorname*{argmin}_{x \in X} \left\{ f(x) + \frac{1}{2} \, \|Ax + B\bar{y}^n - c\|^2 + \frac{1}{2} \, \|x - x_k^{n+1}\|_M^2 \right\} \\ x^{n+1} = \frac{1}{K} \sum_{k=1}^K x_k^{n+1} \quad \text{et} \quad y_0^{n+1} = y^n + \frac{1}{t_{n+1}} \left(y^{n-1} - y^n \right) + \frac{t_n}{t_{n+1}} \left(y_L^n - y^{n-1} \right) \\ \begin{cases} \forall \ell \in [\![0]; L-1]\!] \\ y_{\ell+1}^{n+1} = \operatorname*{argmin}_{y \in Y} \left\{ g(y) + \frac{1}{2} \, \|Ax^{n+1} + By - c\|^2 + \frac{1}{2} \, \|y - y_\ell^{n+1}\|_P^2 \right\} \end{cases} \\ y^{n+1} = \frac{1}{L} \sum_{\ell=1}^L y_\ell^{n+1} \\ \bar{y}^{n+1} = y^{n+1} + \frac{t_{n+1} - 1}{t_{n+2}} \left(y^{n+1} - y^n \right) \end{cases}$$

où on choisit l'initialisation $x_K^0, x^{-1} = x^0 \in X$ et $y_L^0, y^{-1} = y^0 \in Y$, et où la suite des paramètres de relaxation $(t_n)_{n \in \mathbb{N}}$ doit vérifier

$$\forall n \ge 0, \quad t_n^2 \ge t_{n+1}(t_{n+1} - 1).$$
 (6.23)

Le choix $t_n = (n+1)/2$ convient par exemple.

DÉMONSTRATION : On vérifie que $t_{n+1} > 0$ pour tout $n \in \mathbb{N}$, ce qui permet d'écrire l'inégalité (6.20) au point

$$(x,y) = \frac{t_{n+1} - 1}{t_{n+1}} (x^n, y^n) + \frac{1}{t_{n+1}} (x^*, y^*)$$

Si on utilise par ailleurs la convexité de ${\cal E}$ et qu'on multiplie l'inégalité résultante par $t^2_{n+1},$ on obtient

$$\begin{split} t_{n+1}(t_{n+1}-1)\left(\mathcal{E}(x^n,y^n)-\mathcal{E}(x^*,y^*)\right) &+ \frac{1}{2} \|B((t_{n+1}-1)\,y^n+y^*-t_{n+1}\,\bar{y})\|^2 \\ &+ \frac{1}{2K} \,\|(t_{n+1}-1)\,x^n+x^*-t_{n+1}\,\hat{x}_0\|_M^2 + \frac{1}{2L} \,\|(t_{n+1}-1)\,y^n+y^*-t_{n+1}\,\hat{y}_0\|_P^2 \\ &\geq t_{n+1}^2 \left(\mathcal{E}(x^{n+1},y^{n+1})-\mathcal{E}(x^*,y^*)\right) + \frac{1}{2} \,\|B((t_{n+1}-1)\,y^n+y^*-t_{n+1}\,y^{n+1})\|^2 \\ &+ \frac{1}{2K} \,\|(t_{n+1}-1)\,x^n+x^*-t_{n+1}\,x_K^{n+1}\|_M^2 \\ &+ \frac{1}{2L} \,\|(t_{n+1}-1)\,y^n+y^*-t_{n+1}\,y_L^{n+1}\|_P^2. \end{split}$$

Posons

$$\bar{y} = y^n + \frac{t_n - 1}{t_{n+1}} (y^n - y^{n-1}),$$

on obtient $(t_{n+1}-1) y^n + y^* - t_{n+1} \bar{y} = (t_n-1) y^{n-1} + y^* - t_n y^n$. Si on choisit (avec par convention $x^{-1} = x^0$ et $y^{-1} = y^0$)

$$\hat{x}_0 = x_0^{n+1} = x^n + \frac{1}{t_{n+1}} \left(x^{n-1} - x^n \right) + \frac{t_n}{t_{n+1}} \left(x_K^n - x^{n-1} \right)$$

 et

$$\hat{y}_0 = y_0^{n+1} = y^n + \frac{1}{t_{n+1}} \left(y^{n-1} - y^n \right) + \frac{t_n}{t_{n+1}} \left(y_L^n - y^{n-1} \right)$$

on a alors d'une part

$$(t_{n+1}-1)x^n + x^* - t_{n+1}\hat{x}_0 = (t_n-1)x^{n-1} + x^* - t_n x_K^n$$

 et

 $(t_{n+1}-1)y^n + y^* - t_{n+1}\hat{y}_0 = (t_n-1)y^{n-1} + y^* - t_n y_L^n.$

d'autre part. Ainsi, (6.20) devient

$$\begin{split} t_{n+1}(t_{n+1}-1) \left(\mathcal{E}(x^n, y^n) - \mathcal{E}(x^*, y^*) \right) \\ &+ \frac{1}{2} \left\| B((t_n-1) y^{n-1} + y^* - t_n y^n) \right\|^2 \\ &+ \frac{1}{2K} \left\| (t_n-1) x^{n-1} + x^* - t_n x_K^n \right\|_M^2 + \frac{1}{2L} \left\| (t_n-1) y^{n-1} + y^* - t_n y_L^n \right\|_P^2 \\ &\geq t_{n+2}(t_{n+2}-1) \left(\mathcal{E}(x^{n+1}, y^{n+1}) - \mathcal{E}(x^*, y^*) \right) \\ &+ \frac{1}{2} \left\| B((t_{n+1}-1) y^n + y^* - t_{n+1} y^{n+1}) \right\|^2 \\ &+ \frac{1}{2K} \left\| (t_{n+1}-1) x^n + x^* - t_{n+1} x_K^{n+1} \right\|_M^2 + \frac{1}{2L} \left\| (t_{n+1}-1) y^n + y^* - t_{n+1} y_L^{n+1} \right\|_P^2. \end{split}$$

Sommons pour n entre 0 et N-1; après télescopage des termes, on obtient alors (puisque $t_1 = 0$)

$$\frac{1}{2} \|B((t_0 - 1) y^{-1} + y^* - t_0 y^0)\|^2
+ \frac{1}{2K} \|(t_0 - 1) x^{-1} + x^* - t_0 x_K^0\|_M^2 + \frac{1}{2L} \|(t_0 - 1) y^{-1} + y^* - t_0 y_L^0\|_P^2
\ge t_{N+1}(t_{N+1} - 1) \left(\mathcal{E}(x^N, y^N) - \mathcal{E}(x^*, y^*)\right) + \frac{1}{2} \|B((t_N - 1) y^{N-1} + y^* - t_N y^N)\|^2
+ \frac{1}{2K} \|(t_N - 1) x^{N-1} + x^* - t_N x_K^N\|_M^2 + \frac{1}{2L} \|(t_N - 1) y^{N-1} + y^* - t_N y_L^N\|_P^2$$

qui se réécrit $\mathcal{E}(x^N, y^N) - \mathcal{E}(x^*, y^*) \le \frac{C_N}{t_{N+1}(t_{N+1} - 1)}$

avec

$$C_{N} = \frac{\|B(y^{*} - t_{0} y^{0} + (t_{0} - 1) y^{-1})\|^{2} - \|B(y^{*} - t_{N} y^{N} + (t_{N} - 1) y^{N-1})\|^{2}}{2} + \frac{\|x^{*} - t_{0} x_{K}^{0} + (t_{0} - 1) x^{-1}\|_{M}^{2} - \|x^{*} - t_{N} x_{K}^{N} + (t_{N} - 1) x^{N-1}\|_{M}^{2}}{2K} + \frac{\|y^{*} - t_{0} y_{L}^{0} + (t_{0} - 1) y^{-1}\|_{P}^{2} - \|y^{*} - t_{N} y_{L}^{N} + (t_{N} - 1) y^{N-1}\|_{P}^{2}}{2L}.$$
(6.24)

On vient ainsi à nouveau de prouver la convergence de l'algorithme, avec cette fois une erreur décroissant en $O(1/N^2)$.

6.2.4 Application au modèle ROF

Montrons à présent comment cet algorithme peut être utilisé dans une approche d'éclatement pour résoudre le problème ROF dans le cas des images couleurs. Version discrète de TV 2D couleur Dans le cas des images couleurs (on se restreint ici au cas des images à m = 3 canaux couleurs RGB, mais la démarche est généralisable au cas multi-spectral avec m > 3), l'image u peut se décomposer en trois images mono-valuées

$$u = (u^{\mathrm{R}}, u^{\mathrm{G}}, u^{\mathrm{B}})$$
 avec $u^{\mathrm{R}}, u^{\mathrm{G}}, u^{\mathrm{B}} \in \mathbb{R}^{N_x \times N_y}$.

Une manière simple de définir la norme TV sur de telles images est de choisir une norme TV sur les images en niveaux de gris, puis de sommer cette norme sur les trois canaux. En choisissant par exemple la TV (spatialement) anisotrope, on obtient

$$\sum_{i=0}^{N_x-1} \sum_{j=0}^{N_y-1} \| (\nabla^h (v^h)^{\mathbf{R}})_{i,j} \|_1 + \| (\nabla^h (v^h)^{\mathbf{G}})_{i,j} \|_1 + \| (\nabla^h (v^h)^{\mathbf{B}})_{i,j} \|_1$$
(6.25)

alors que la version (spatialement) isotrope donne

$$\sum_{i=0}^{N_x-1} \sum_{j=0}^{N_y-1} \| (\nabla^h (v^h)^{\mathbf{R}})_{i,j} \|_2 + \| (\nabla^h (v^h)^{\mathbf{G}})_{i,j} \|_2 + \| (\nabla^h (v^h)^{\mathbf{B}})_{i,j} \|_2.$$
(6.26)

L'inconvénient majeur de ces deux versions de TV est qu'elles découplent totalement les différents canaux couleur. Or, les discontinuités de couleurs ont tendance à s'aligner sur ces canaux. On peut imaginer coupler les variations sur chaque canal en remplaçant la somme par la racine carrée de la somme des carrés par exemple :

$$\sum_{i=0}^{N_x-1} \sum_{j=0}^{N_y-1} \sqrt{\|(\nabla^h(v^h)^{\mathbf{R}})_{i,j}\|_2^2 + \|(\nabla^h(v^h)^{\mathbf{G}})_{i,j}\|_2^2 + \|(\nabla^h(v^h)^{\mathbf{B}})_{i,j}\|_2^2}.$$
 (6.27)

Une telle version, contrairement à la version anisotrope présentée en début de ce paragraphe, couple non seulement les variations horizontales et verticales, mais également les variations sur chaque canal. On la désignera par la suite sous le nom de TV isotrope. En suivant le formalisme de l'article IPOL [3], on remarque que pour généraliser la définition de la norme TV aux images couleurs, une manière de procéder est de commencer par définir le gradient de couleur en (i,j) comme une matrice de taille 2 × 3

$$\begin{split} (\nabla^{h}v^{h})_{i,j} &= \left((\nabla^{h}(v^{h})^{\mathrm{R}})_{i,j}, (\nabla^{h}(v^{h})^{\mathrm{G}})_{i,j}, (\nabla^{h}(v^{h})^{\mathrm{B}})_{i,j} \right) \\ &= \begin{pmatrix} (\delta^{h}_{x}(v^{h})^{\mathrm{R}})_{i,j} & (\delta^{h}_{x}(v^{h})^{\mathrm{G}})_{i,j} & (\delta^{h}_{x}(v^{h})^{\mathrm{B}})_{i,j} \\ (\delta^{h}_{y}(v^{h})^{\mathrm{R}})_{i,j} & (\delta^{h}_{y}(v^{h})^{\mathrm{G}})_{i,j} & (\delta^{h}_{y}(v^{h})^{\mathrm{B}})_{i,j} \end{pmatrix} \\ (\nabla^{h}v^{h})_{i,j} &= \begin{pmatrix} (\delta^{h}_{x}v^{h})_{i,j} \\ (\delta^{h}_{y}v^{h})_{i,j} \end{pmatrix} \quad \text{avec} \quad \begin{cases} (\delta^{h}_{x}v^{h})_{i,j} = \left((\delta^{h}_{x}(v^{h})^{\mathrm{R}})_{i,j} & (\delta^{h}_{x}(v^{h})^{\mathrm{G}})_{i,j} & (\delta^{h}_{y}(v^{h})^{\mathrm{B}})_{i,j} \right) \\ (\delta^{h}_{y}v^{h})_{i,j} &= \left((\delta^{h}_{y}(v^{h})^{\mathrm{R}})_{i,j} & (\delta^{h}_{y}(v^{h})^{\mathrm{G}})_{i,j} & (\delta^{h}_{y}(v^{h})^{\mathrm{B}})_{i,j} \right) \end{split}$$

puis de choisir une norme sur cette matrice. On peut par exemple définir pour tous entiers a et b la norme $\|\cdot\|_{a,b}$ en posant pour toute matrice $A = (a_{i,j})_{\substack{1 \le i \le 2\\ 1 \le j \le 3}}$

$$\|A\|_{a,b} = \left\| \left(\|A_{\cdot,1}\|_a, \|A_{\cdot,2}\|_a, \|A_{\cdot,3}\|_a \right) \right\|_b.$$

Les différentes normes TV couleur proposées plus haut correspondent donc respectivement à la définition de $\|(\nabla v^h)_{i,j}\|_{1,1}$, $\|(\nabla v^h)_{i,j}\|_{2,1}$ et $\|(\nabla v^h)_{i,j}\|_{2,2}$. On peut également choisir d'intervertir l'ordre de traitement respectif de la coordonnée spatiale et du canal couleur, en définissant pour tous entiers α et β la norme $\|\cdot\|^{\alpha,\beta}$ en posant pour toute matrice $A = (a_{i,j})_{\substack{1 \le i \le 2\\1 \le i \le 3}}$

$$||A||^{\alpha,\beta} = \left\| \left(||A_{1,\cdot}||_{\alpha}, ||A_{2,\cdot}||_{\alpha} \right) \right\|_{\beta}.$$

Ce choix nous permet par exemple de proposer une version anisotrope (spatialement) de la norme TV qui couple par ailleurs les canaux couleurs, en choisissant $\alpha = 2$ et $\beta = 1$:

$$\|(\nabla^{h}v^{h})_{i,j}\|^{2,1} = \left\| \left(\|(\delta^{h}_{x}v^{h})_{i,j}\|_{2}, \|(\delta^{h}_{y}v^{h})_{i,j}\|_{2} \right) \right\|_{1} = \|(\delta^{h}_{x}v^{h})_{i,j}\|_{2} + \|(\delta^{h}_{y}v^{h})_{i,j}\|_{2}.$$
(6.28)

La version isotrope (6.27) est alors obtenue en choisissant $\alpha = \beta = 2$.

On peut étendre la définition des normes matricielles $||A||^{\alpha,\beta}$ et $||A||_{a,b}$ à toute matrice A. La norme $||A||^{\alpha,\beta}$ (resp. $||A||_{a,b}$) est donc obtenue en appliquant la norme ℓ_{α} aux lignes (resp. ℓ_a aux colonnes) de la matrice A, avant d'appliquer la norme ℓ_{β} (resp. ℓ_b) au vecteur résultant. Dans ce cas, on montre aisément que $||A||^{\alpha,\beta} = ||^{t}A||_{a,b}$.

Éclatement sur les carrés pairs/impairs : version anisotrope On choisit dans ce paragraphe de considérer la version spatialement anisotrope de TV (6.28) (obtenue en choisissant (α,β) = (2,1)), ce qui revient à s'intéresser au problème ROF :

$$\min_{v^h \in \mathbb{R}^{N_x \times N_y}} \left\{ \frac{1}{2\lambda} \sum_{i=0}^{N_x - 1} \sum_{j=0}^{N_y - 1} \|v_{i,j}^h - g_{i,j}^h\|_2^2 + \sum_{i=0}^{N_x - 1} \sum_{j=0}^{N_y - 1} \|(\nabla^h v^h)_{i,j}\|^{2,1} \right\}$$
(6.29)

On applique la méthode par éclatement sur les carrés sur le problème (6.29). En reprenant les mêmes notations que dans la section précédente, on rappelle que la méthode d'éclatement de DYKSTRA nous amène à considérer le problème dual

$$\min_{\substack{x^h \in \mathbb{R}^{3 \times N_x \times N_y} \\ y^h \in \mathbb{R}^{3 \times N_x \times N_y}}} \left\{ (\mathrm{TV}^h_{\mathrm{e}})^* (x^h) + (\mathrm{TV}^h_{\mathrm{o}})^* (y^h) + \frac{1}{2\lambda} \left\| \lambda (x^h + y^h) - g^h \right\|_2^2 \right\}$$

car l'anisotropie spatiale de la norme TV choisie permet à nouveau de séparer la norme TV sur les carrés pairs et les carrés impairs, en écrivant :

$$\sum_{i=0}^{N_x-1} \sum_{j=0}^{N_y-1} \| (\nabla^h v^h)_{i,j} \|^{2,1} = \underbrace{\sum_{i=0}^{N_x-1} \sum_{j=0}^{N_y-1} \| D(S_{i,j}v^h) \|^{2,1}}_{i \text{ pair } j \text{ pair } j \text{ pair } } + \underbrace{\sum_{i=-1}^{N_x-1} \sum_{j=-1}^{N_y-1} \| D(S_{i,j}v^h) \|^{2,1}}_{i \text{ impair } j \text{ impair } } = \operatorname{TV}_{\mathrm{o}}^h(v^h)$$

Des calculs analogues à ceux de la section précédente permettent de montrer que

$$(\mathrm{TV}_{\mathrm{e}}^{h})^{*}(x) = \sum_{\substack{i=0\\i \text{ pair}}}^{N_{x}-1} \sum_{\substack{j=0\\j \text{ pair}}}^{N_{y}-1} \chi_{\mathcal{K}}(S_{i,j}x) \quad \text{et} \quad (\mathrm{TV}_{\mathrm{o}}^{h})^{*}(y) = \sum_{\substack{i=-1\\i \text{ impair}}}^{N_{x}-1} \sum_{\substack{j=-1\\j \text{ impair}}}^{N_{y}-1} \chi_{\mathcal{K}}(S_{i,j}y)$$

avec $\mathcal{K} = \{D^* \xi \mid \xi = (\xi_k)_{k \in [\![1;4]\!] \text{ ou } k \in [\![1;2]\!]}, \text{ avec } \xi_k \in \mathbb{R}^3 \text{ et } ||\xi_k||_2 \leq 1\}.$ On cherche donc à résoudre le problème

$$\min_{\substack{x^h \in \mathbb{R}^{3 \times N_x \times N_y} \\ y^h \in \mathbb{R}^{3 \times N_x \times N_y}}} \left\{ \sum_{\substack{i=0 \ i \text{ pair } j \text{ pair }}}^{N_x - 1} \sum_{\substack{j=0 \\ i \text{ pair } j \text{ pair }}}^{N_x - 1} \chi_{\mathcal{K}}(S_{i,j}x^h) + \sum_{\substack{i=-1 \\ i \text{ impair } j \text{ impair }}}^{N_y - 1} \chi_{\mathcal{K}}(S_{i,j}y^h) + \frac{1}{2\lambda} \|\lambda(x^h + y^h) - g^h\|_2^2 \right\}.$$

Introduisons deux variables auxiliaires ξ_x^h et ξ_y^h , de taille $3 \times N_x \times N_y$. Si on définit ensuite les opérateurs D^x et D^y en posant pour tous *i* et *j* pairs

$$S_{i,j}(D^x \xi^h_x) = D^*(S_{i,j} \xi^h_x)$$

puis, pour *i* et *j* impairs, $S_{i,j}(D^y \xi^h_y) = D^*(S_{i,j} \xi^h_y)$

alors on peut faire le changement de variables $x^h = D^x \xi^h_x$ et $y^h = D^y \xi^h_y$. Dans ce cas, on peut réécrire le problème sous la forme

$$\min_{\substack{\xi_x^h, \xi_y^h \in \mathbb{R}^{3 \times N_x \times N_y} \\ \|(\xi_x^h)_{i,j}\|_2 \le 1 \\ \|(\xi_y^h)_{i,j}\|_2 \le 1 \\ \|(\xi_y^h)_{i,j}\|_2 \le 1 }} \left\{ \frac{1}{2} \|D^x \xi_x^h + D^y \xi_y^h - g^h / \lambda\|_2^2 \right\}$$

qui est bien de la forme (6.10). L'image v^h est alors donnée à chaque itération par $v_n^h = g^h - \lambda (D^x(\xi_x^h)_n - D^y(\xi_y^h)_n) = g^h - \lambda (x_n^h + y_n^h)$. On peut donc utiliser l'algorithme proposé au paragraphe précédent, en choisissant

$$M = \frac{1}{\tau} I - D^{x} (D^{x})^{*} \quad \text{et} \quad P = \frac{1}{\tau} I - D^{y} (D^{y})^{*}.$$

On en déduit en particulier que

$$\|\xi\|_M^2 = \frac{1}{\tau} \,\|\xi\|^2 - \|D^x\xi\|^2 \qquad \text{et} \qquad \|\xi\|_P^2 = \frac{1}{\tau} \,\|\xi\|^2 - \|D^y\xi\|^2$$

qui définissent des normes si $\tau \|D^x\|^2 \leq 1$ et $\tau \|D^y\|^2 \leq 1$. Un calcul rapide assure que $\|D^x\| = \|D^y\| = 2$. On obtient alors les itérations suivantes : pour tout $n \geq 0$,

$$\begin{cases} (\xi_x^h)_0^{n+1} = (\xi_x^h)_K^n \\ \begin{cases} \forall k \in [\![0]; K-1]\!] \\ (\xi_x^h)_{k+1}^{n+1} = \operatorname*{argmin}_{\substack{\xi_x^h \in \mathbb{R}^{3 \times N_x \times N_y} \\ \parallel (\xi_x^h)_{i,j} \parallel_2 \leq 1 \end{cases}} \left\{ \frac{1}{2} \| D^x \xi_x^h + (D^y \xi_y^h)^n - g^h / \lambda \|^2 + \frac{1}{2\tau} \| \xi_x^h - (\xi_x^h)_k^{n+1} \|^2 \\ - \frac{1}{2} \| D^x \xi_x^h - D^x (\xi_x^h)_k^{n+1} \|^2 \right\} \\ \{ (\xi_x^h)^{n+1} = \frac{1}{K} \sum_{k=1}^K (\xi_x^h)_k^{n+1} \quad \text{et} \quad (\xi_y^h)_0^{n+1} = (\xi_y^h)_L^n \\ \begin{cases} \forall \ell \in [\![0]; L-1]\!] \\ (\xi_y^h)_{\ell+1}^{n+1} = \operatorname*{argmin}_{\xi_y^h \in \mathbb{R}^{3 \times N_x \times N_y}} \left\{ \frac{1}{2} \| D^x (\xi_x^h)^{n+1} + D^y \xi_y^h - g^h / \lambda \|^2 + \frac{1}{2\tau} \| \xi_y^h - (\xi_y^h)_\ell^{n+1} \|^2 \\ - \frac{1}{2} \| D^y \xi_y^h - D^y (\xi_y^h)_\ell^{n+1} \|^2 \right\} \\ (\xi_y^h)^{n+1} = \frac{1}{L} \sum_{\ell=1}^L (\xi_y^h)_\ell^{n+1} \end{cases}$$

qui, après simplification, s'écrit

$$\begin{cases} (\xi_x^h)_0^{n+1} = (\xi_x^h)_K^n \\ \begin{cases} \forall k \in \llbracket 0 ; K-1 \rrbracket, \forall (i,j) \text{ pair} \\ S_{i,j}(\xi_x^h)_{k+1}^{n+1} = \operatorname{proj}_{B(0,1)} \left(S_{i,j}(\xi_x^h)_k^{n+1} + \tau D \left(D^* (S_{i,j}\xi_x^h)_k^{n+1} + S_{i,j} (D^y (\xi_y^h)^n - g^h / \lambda) \right) \right) \\ (\xi_x^h)^{n+1} = \frac{1}{K} \sum_{k=1}^K (\xi_x^h)_k^{n+1} \quad \text{et} \quad (\xi_y^h)_0^{n+1} = (\xi_y^h)_L^n \\ \begin{cases} \forall \ell \in \llbracket 0 ; L-1 \rrbracket, \forall (i,j) \text{ impair} \\ S_{i,j}(\xi_y^h)_{\ell+1}^{n+1} = \operatorname{proj}_{B(0,1)} \left(S_{i,j}(\xi_y^h)_{\ell}^{n+1} + \tau D \left(D^* (S_{i,j}\xi_y^h)_{\ell}^{n+1} + S_{i,j} (D^x (\xi_x^h)^{n+1} - g^h / \lambda) \right) \right) \\ (\xi_y^h)^{n+1} = \frac{1}{L} \sum_{\ell=1}^L (\xi_y^h)_\ell^{n+1} \end{cases}$$

où B(0,1) est la boule unité pour la norme $\|\cdot\|^{2,\infty}$. On remarque en particulier que chaque pas de descentes proximales s'écrit comme la projection sur la boule unité d'un certain vecteur, qui est donc calculable de manière exacte et efficace.

Éclatement sur les carrés pairs/impairs : version pseudo-isotrope La version anistrope étudiée dans le paragraphe précédent découple totalement les variations horizontales et verticales, mais la version isotrope ($\alpha = \beta = 2$) n'est pas éclatable sur les carrés pairs et impairs. Néanmoins, il est possible de conserver un certain couplage des directions horizontales et verticales dans la norme TV, tout en en préservant l'éclatement, en introduisant

$$TV_{pseudo-iso}^{h}(v^{h}) = \underbrace{\sum_{\substack{i=0\\ i \text{ pair } j \text{ pair }}}^{N_{x}-1} \sum_{\substack{j=0\\ i \text{ pair } j \text{ pair }}}^{N_{y}-1} \|D(Sv^{h})_{i,j}\|^{2,2}}_{= TV_{e}^{h}(v^{h})} + \underbrace{\sum_{\substack{i=-1\\ i \text{ impair } j \text{ impair }}}^{N_{x}-1} \sum_{\substack{j=-1\\ i \text{ impair } j \text{ impair }}}^{N_{y}-1} \|D(Sv^{h})_{i,j}\|^{2,2}}_{= TV_{o}^{h}(v^{h})}$$

qui peut être vue comme une approximation de la TV isotrope (6.27). On la désignera par la suite sous le terme de TV pseudo-isotrope, pour la distinguer de la version isotrope définie L'algorithme reste alors le même, à l'exception des projections qui se font sur la boule unité pour la norme $\|\cdot\|^{2,2}$.

Mesure de la convergence Pour mesurer la convergence, on utilise le *primal dual gap* associé au problème primal-dual étudié ici

$$\min_{v^h \in \mathbb{R}^{3 \times N_x \times N_y}} \sup_{\substack{x^h \in \mathbb{R}^{3 \times N_x \times N_y} \\ y^h \in \mathbb{R}^{3 \times N_x \times N_y}}} \left\{ \frac{1}{2\lambda} \|v^h - g^h\|_2^2 + \langle v^h, x^h + y^h \rangle - (\mathrm{TV}^h_{\mathrm{e}})^*(x^h) - (\mathrm{TV}^h_{\mathrm{o}})^*(y^h) \right\}$$

qui correspond à la différence (positive) entre l'énergie primale, donnée par

$$E_{\mathcal{P}}(v^{h}) = \sup_{\substack{x^{h} \in \mathbb{R}^{3 \times N_{x} \times N_{y}} \\ y^{h} \in \mathbb{R}^{3 \times N_{x} \times N_{y}}}} \left\{ \frac{1}{2\lambda} \|v^{h} - g^{h}\|_{2}^{2} + \langle v^{h}, x^{h} + y^{h} \rangle - (\mathrm{TV}_{e}^{h})^{*}(x^{h}) - (\mathrm{TV}_{o}^{h})^{*}(y^{h}) \right\}$$
$$= \frac{1}{2\lambda} \|v^{h} - g^{h}\|_{2}^{2} + \mathrm{TV}_{e}^{h}(v^{h}) + \mathrm{TV}_{o}^{h}(v^{h})$$



(a) Image originale u (b) Image bruitée g

(c) Image débruitée V^*

FIGURE 6.1 – Image originale u (à gauche), image bruitée g (au milieu) et image débruitée V^* (à droite). Le bruit ajouté est un bruit blanc gaussien additif, de variance 100 (les images sont à valeurs entre 0 et 255). L'image débruité V^* est obtenue ici en utilisant l'algorithme PDHG, en poussant les itérations jusqu'à convergence. Source : détail de l'image *Hepatica nobilis flowers*, par Archenzo.

et l'énergie duale :

$$E_{\mathcal{D}}(x^{h}, y^{h}) = \min_{v^{h} \in \mathbb{R}^{3 \times N_{x} \times N_{y}}} \left\{ \frac{1}{2\lambda} \|v^{h} - g^{h}\|_{2}^{2} + \langle v^{h}, x^{h} + y^{h} \rangle - (\mathrm{TV}_{\mathrm{e}}^{h})^{*}(x^{h}) - (\mathrm{TV}_{\mathrm{o}}^{h})^{*}(y^{h}) \right\}$$

Cette différence présente un unique zéro, atteint en (v^*, x^*, y^*) . Une manière de mesurer la convergence est donc de calculer à chaque itération n cette différence. Cette quantité vaut

$$\mathcal{G}(v_n^h, x_n^h, y_n^h) = E_{\mathcal{P}}(v^h) - E_{\mathcal{D}}(x^h, y^h) = \mathrm{TV}_{\mathrm{e}}^h(v^h) + \mathrm{TV}_{\mathrm{o}}^h(v^h) - \langle v^h, x^h + y^h \rangle.$$

Un critère d'arrêt peut alors être choisi, en stoppant les itérations dès que la valeur du gap tombe en-dessous d'un certain seuil.

6.3 Résultats expérimentaux

On présente dans cette section quelques résultats et comparaisons expérimentaux pour le problème ROF 2D couleur. Le code utilisé est écrit en C++ et utilisé en Matlab grâce à la fonction Mex. La parallélisation est réalisée grâce à OpenMP et les expériences ont été lancées sur une machine à deux cœurs.

6.3.1 Choix de la référence

Expérience Pour tester l'algorithme proposé dans la section précédente, on ajoute artificiellement du bruit gaussien sur une image initiale, considérée sans bruit (cf figure 6.1). L'image choisie est une image RGB, de taille 201×201 , à valeurs entre 0 et 1. Le bruit ajouté est un bruit blanc gaussien, de variance $(10/255)^2$. Le paramètre λ est alors choisi égal à 0,1. La reconstruction v^* est alors la sortie de l'algorithme considérée.

Reconstruction de référence La pertinence du modèle ROF (ainsi que le choix du paramètre λ) n'est pas étudiée ici, mais la qualité de la résolution du problème ROF lui-même. L'objectif est donc de résolute de manière exacte le problème ROF (dans sa version isotrope, donnée par (6.27)), et on évaluera les résultats selon cet objectif.

Dans ce cadre, les TV anistrope et pseudo-isotrope sont considérées ici comme des approximations de la TV isotrope.

Afin d'évaluer la minimisation du problème ROF par la méthode des descentes alternées, on choisit comme méthode de référence l'algorithme PDHG (cf. chapitre 5), étudié par exemple dans [4]. Comme cette méthode permet de résoudre le problème ROF avec TV isotrope, on utilisera la sortie V^* de cet algorithme comme reconstruction de référence (en poussant les itérations jusqu'à convergence). En d'autres termes, on supposera que V^* est le minimiseur du problème ROF considéré.

Outils de comparaison Pour comparer les résultats obtenus avec la reconstruction de référence, notée V^* , on utilisera deux outils. Le premier est l'énergie de la reconstruction v^* , donnée par

$$E(v^*) = \frac{1}{2\lambda} \|v^* - g^h\|^2 + \mathrm{TV}^h_{\mathrm{iso}}(v^*).$$

On supposera que le minimum de cette fonctionnelle est atteint en V^* , et vaut 293,7285. Aussi on mesurera la distance au minimum en calculant la différence relative

$$\delta E(v^*) = \frac{E(v^*) - E(V^*)}{E(V^*)}$$

Plus cette quantité est faible, meilleure est la minimisation. On utilisera également l'erreur quadratique moyenne, définie comme suit

$$\operatorname{err}(v^*) = \frac{\|V^* - v^*\|_2^2}{N_x N_y}$$

où le domaine de l'image g^h est de taille $N_x \times N_y$. Plus cette erreur est faible, plus v^* est proche du minimiseur V^* .

6.3.2 Descentes alternées pour l'éclatement sur carrés

Qualité de la minimisation On a choisi comme critère d'arrêt d'utiliser le primal dual gap introduit à la section précédente, et de stopper les itérations lorsque le gap moyen valait moins de 0,01 par pixel. Les images débruitées sont présentées à la figure 6.2. Quatre méthodes sont testées : la version pseudo-isotrope et la version anisotrope (avec et sans accélération). Pour cette expérience, on choisit K = L = 3 le nombre d'itérations locales. On calcule alors pour chaque méthode (TV anisotrope/pseudoisotrope, version non accélérée/accélérée) l'énergie associée à l'image v^* obtenue. Pour la version pseudo-isotrope, celle-ci vaut 301,6866 (soit $\delta E = 2,71\%$) pour la version classique et 304,9170 (soit $\delta E = 3,81\%$) pour la version accélérée; pour la version anisotrope, elle vaut respectivement 304,2137 (soit $\delta E = 3,57\%$) et 301,7970 (soit $\delta E = 2,75\%$).

Les reconstructions obtenues présentent par ailleurs les erreurs quadratiques moyennes de 0,0002 pour la version pseudo-isotrope et la version anisotrope non accélérée; elle est de 0,0001 pour la version anistrope accélérée.

Nombre d'itérations locales et globales Comparons pour chaque expérience le nombre d'itérations globales, noté N, nécessaires avant la sortie de la boucle principale sur n, en fonction du nombre d'itérations locales K = L. Le résultat de cette expérience est donné par le tableau 6.3. On rappelle que le nombre de descentes total est donné par (K + L)N, ce qui nous permet également de donner la durée moyenne d'une des-



(a) N = 25

(b) N = 12



FIGURE 6.2 – Résultats obtenus par la méthode présentée. Ligne du haut : version pseudoisotrope, algorithme non accéléré à gauche et accéléré à droite. Ligne du bas : version pseudoisotrope, algorithme non accéléré à gauche et accéléré à droite. En légende, le nombre d'itérations globales N. Le nombre d'itérations locales vaut ici K = L = 3 pour les quatre expériences.

Itérations locales $K = L$	1	2	3	4	5	6	7	8	9	10
Itérations globales N	37	29	25	23	22	21	20	20	20	19
Temps total (en secondes)	0,090	0,086	0,089	0,095	0,102	0,110	0,115	0,127	0,138	0,141
Temps par descente (en millisecondes)	1,216	0,741	0,590	0,516	0,463	0,437	0,411	0,397	0,383	0,371
			(a)							
Itérations locales $K = L$	1	2	3	4	5	6	7	8	9	10
Itérations globales N	20	14	12	12	12	12	12	12	12	12
Temps total (en secondes)	0,081	0,065	0,061	0,068	0,074	0,081	0,088	0,095	0,101	0,110
Temps par descente (en millisecondes)	2,025	1,161	0,847	0,708	0,617	0,563	0,495	0,495	0,468	$0,\!458$
			(b)							
Itérations locales $K = L$	$\parallel 1$	2	3	4	5	6	7	8	9	10
Itérations globales N	189	138	114	100	92	86	82	80	77	76
Temps total (en secondes)	0,575	0,566	0,588	0,622	0,670	0,718	0,770	0,838	0,886	0,953
Temps par descente (en millisecondes)	1,521	1,026	0,860	0,778	0,728	0.696	0.671	0.655	0,639	0.627
				,	,	,	,	, ,	,	0,027
		I	(c)		,	,	,	,		0,027
Itérations locales $K = L$	1	2	(c) 3	4	5	6	7	8	9	10
Itérations locales $K = L$ Itérations globales N	1 90	2	(c) 3 48	4	5	6 40	7	8	9 37	10
Itérations locales $K = L$ Itérations globales N Temps total (en secondes)	1 90 0,451	2 58 0,348	(c) 3 48 0,341	4 44 0,358	5 41 0,377	6 40 0,411	7 38 0,432	8 38 0,472	9 37 0,497	
Itérations locales $K = L$ Itérations globales N Temps total (en secondes) Temps par descente (en millisecondes)	$ 1 \\ 90 \\ 0,451 \\ 2,506 $	2 58 0,348 1,500	(c) 3 48 0,341 1,184	4 44 0,358 1,017	5 41 0,377 0,920	6 40 0,411 0,856	7 38 0,432 0,812	8 38 0,472 0,776	9 37 0,497 0,746	$ \begin{array}{r} 10 \\ \hline 36 \\ 0,521 \\ 0,724 \end{array} $
Itérations locales $K = L$ Itérations globales N Temps total (en secondes) Temps par descente (en millisecondes)	$ \begin{array}{c c} 1 \\ 90 \\ 0,451 \\ 2,506 \\ \end{array} $	2 58 0,348 1,500	(c) 3 48 0,341 1,184 (d)		5 41 0,377 0,920	6 40 0,411 0,856	7 38 0,432 0,812	8 38 0,472 0,776	9 37 0,497 0,746	$ \begin{array}{c} 10 \\ 36 \\ 0,521 \\ 0,724 \end{array} $

FIGURE 6.3 – Nombre d'itérations nécessaires avant convergence. De haut en bas : (a) TV pseudo-isotrope classique, (b) TV pseudo-isotrope accéléré,
(c) TV anisotrope classique et (d) TV anisotrope accéléré. Pour chaque expérience, on donne le nombre d'itérations globales N nécessaires avant arrêt
de l'algorithme, en fonction du nombre d'itérations locales $K = L$, le temps total d'exécution et la durée moyenne d'une descente (qui est le temps
total divisé par $2N(K+L)$).



FIGURE 6.4 – Influence du nombre d'itérations locales K = L sur le temps d'exécution. Pour le même critère d'arrêt, on affiche le temps de calcul nécessaire avant l'arrêt des itérations en fonction du nombre d'itérations locales. Les durées sont également consultables dans le tableau 6.3.

cente. La figure 6.4 permet quant à elle de visualiser le temps d'exécution en fonction du nombre d'itérations locales K = L. Pour des raisons de stabilité, tous les temps donnés sont en réalité des moyennes réalisées sur dix expériences identiques.

Gap Pour comparer la convergence des versions accélérée et non accélérée dans le cas de la TV pseudo-isotrope, on affiche l'allure des deux *gaps* (ramenés au nombre de pixels) dans la figure 6.5.

On teste également l'influence du critère d'arrêt. Dans la figure 6.6, on affiche le temps d'exécution en fonction du seuil choisi pour le gap ainsi que l'énergie atteinte.

6.3.3 Comparaison avec d'autres méthodes

Algorithme PDHG En définissant un critère d'arrêt basé sur le *primal-dual gap* (on stoppe les itérations lorsque le gap – calculé toutes les cinquantes itérations – descent en-dessous de la valeur 0,001), l'algorithme s'arrête pour l'image testée au bout de 1 750 itérations, pour 4,1957 secondes de temps de calculs. L'énergie du résultat vaut 304,3616 et l'erreur quadratique moyenne est de 0,0002. L'erreur relative sur l'énergie est de 3,62%. Le résultat obtenu est présenté à la figure 6.7(a).

Éclatement carrés pairs/impairs sur chaque canal On teste également l'éclatement sur les carrées proposé dans [1]. Celle-ci, décrite dans la section 6.1.3, est conçue pour les images en niveaux de gris. On choisit donc de l'appliquer sur chaque canal couleur, ce qui supprime toute corrélation entre les canaux. Le résultat est présentée à la figure 6.7(b) Le temps d'exécution vaut dans ce cas 0,0781 secondes, l'énergie vaut 314,9710 (d'erreur relative 7,23%) et l'erreur moyenne vaut 0,0005.

6.3.4 Discussion

Performance Malgré l'approximation due à la définition de la TV pseudo-isotrope, les résultats obtenus pour cette version de TV présentent une énergie proche de la



FIGURE 6.5 – Allure des gaps moyens (échelle logarithmique en ordonnée) au cours des itérations. En bleu, la méthode classique et en rouge, la version accélérée (TV pseudo-isotrope, K = L = 3). Remarquer les oscillations du gap dans la version accélérée.



FIGURE 6.6 – Influence du choix du critère d'arrêt sur l'énergie et le temps d'exécution.



FIGURE 6.7 – Comparaison avec deux autres méthodes. De gauche (a) : PDHG avec critère d'arrêt et à droite (b) éclatement sur carrés pour chaque canal.

reconstruction de référence donnée par l'algorithme PDHG, avec une différence relative $\delta E = 2,71\%$ pour le meilleur résultat, qui est celui obtenu avec la TV pseudoisotrope, sans accélération. En outre, les quatre résultats restent meilleurs que celui obtenu en résolvant le problème indépendamment sur chaque canal. La version pseudoisotrope accélérée est néamnoins moins bon que celui obtenu en utilisant un critère d'arrêt pour l'algorithme PDHG.

En terme d'erreur quadratique, le meilleur résultat est étonnamment obtenu pour la version accélérée avec la TV anisotrope. Néanmoins, compte tenu du temps de calcul de cette expérience (près de six fois plus que pour la version accélérée avec la TV pseudoisotrope), et du nombre d'itérations (qui quadruple), la convergence des deux versions n'est pas comparable. Il semblerait donc que le *gap* ne mesure pas la convergence de la même manière dans les deux cas. Elle reste dans tous les cas comparable à celle obtenue avec l'algorithme PDHG.

Accélération La version accélérée converge effectivement plus vite que la version classique. Pour un nombre égal d'itérations locales, le nombre d'itérations globales est divisé en moyenne par 1,8 dans le cas de la TV pseudo-isotrope et par 2,2 dans le cas anisotrope. En termes de temps de calculs, ce facteur est moins important, car la sur-relaxation ajoute des calculs supplémentaires. Il est en moyenne de 1,5 (versions pseudo-isotrope et anisotrope confondues). Néanmoins, dans la figure 6.5, on remarque que, dans le cas de la version pseudo-isotrope, le *gap* oscille dans la version accélérée, ce qui peut expliquer que la reconstruction est légèrement moins bonne (en tant d'énergie et en terme d'erreur) que pour la version non accélérée.

Temps d'exécution Théoriquement, la convergence dépend à la fois du nombre de descentes total, donné respectivement pour chaque variable duale par la quantité KN et LN et du nombre d'itérations globales, donnée par N. Les encadrements (6.22) et (6.24) montrent ainsi que l'erreur décroît avec N, mais que le facteur C_N décroît quant à lui avec K et L. Ainsi, pour un même nombre de descentes total, plus K et L sont petits, et plus le facteur C_N est grand, si bien que l'erreur décroît lentement; aussi, un nombre d'itérations globales N plus grand est nécessaire pour s'approcher de la convergence. Néanmoins, si on augmente K et L, on ne peut espérer faire baisser N au-delà d'une certaine valeur, car C_N possède un terme constant qui ne dépend ni de K et ni de L. Dans le cas non accéléré (6.22), si K et L sont divisés par deux par exemple, alors il faut au moins doubler la valeur de N pour obtenir une borne comparable pour l'erreur. Ainsi, en théorie, ajouter des itérations locales n'est pas forcément une bonne stratégie.

En pratique, le critère le plus utile reste le temps d'exécution. Or celui-ci dépend non seulement du nombre de descentes total, mais également du temps de calcul de chacune de ces descentes. L'éclatement sur les carrés implique en pratique de paralléliser les calculs sur chaque carré. L'initialisation de la parallélisation prend elle-même un certain temps, aussi plus on fait de calculs successifs sur un même carré, moins on perd de temps. Cela explique que l'évolution des temps de calculs observés dans le tableau 6.3. Pour la version accélérée, le temps d'exécution décroît dans un premier temps avec la valeur de K = L, car on perd moins de temps à initialiser la parallélisation. Ensuite, à partir de K = L = 4, la durée des calculs croît, car le nombre d'itérations globales Nvarie très peu, tandis qu'on ajoute des pas de descentes. Le calcul de la durée moyenne d'une descente tend à confirmer cette interprétation, puisqu'elle décroît avec K = L. Notons par ailleurs que cette durée moyenne est plus élevée pour la version accélérée, car il comprend le temps dédié à la sur-relaxation.

La valeur optimale pour K = L semble, pour un même algorithme, peu varier d'une expérience à l'autre. Ainsi, pour la version accélérée avec la TV pseudo-isotrope, il semble que, généralement, le choix optimal soit k = L = 3. Pour le cas non accéléré, il semble davantage se situer autour de la valeur 2.

Précision de la reconstruction Enfin, en analysant les courbes de la figure 6.6, on voit que, de manière attendue, l'énergie décroît lorsque le seuil choisi pour le gap décroît, car on est alors plus proche de la convergence. De même, le temps de calcul suit une tendance inverse. Ainsi, plus le seuil choisi est petit, meilleure est la reconstruction, mais plus longue est l'exécution de l'algorithme. Un bon compromis pourrait être de choisir un seuil compris entre 0,1 et 0,2, puisque l'énergie vaut alors autour de 316, ce qui conduit à une différence relative $\delta E \approx 7,5\%$, pour un temps de calcul situé autour du centième de seconde. La reconstruction est alors dans ce cas visuellement proche de celle obtenue pour un seuil de 0,01.

Conclusion

Nous avons proposé dans ce chapitre un algorithme qui permet d'utiliser la méthode d'éclatement sur les carrés pairs/impairs proposé dans [1] pour le problème ROF couleur. La version de la variation totale choisie dans le cadre de cette méthode n'est pas la version isotrope classique, mais une approximation qui couple néanmoins les directions horizontales et verticales. L'intérêt principal de l'algorithme proposé réside dans son efficacité computationnelle : il bénéficie non seulement de la structure parallèle de la méthode d'éclatement utilisée, mais il permet de plus de gagner du temps sur les calculs locaux en enchaînant sur chaque carré des pas de descentes. Des tests expérimentaux ont permis de vérifier les performances de cette méthode, tant en termes de temps de calculs qu'en termes de qualité de la minimisation.

Néanmoins, le choix optimal du nombre d'itérations locales reste obscur. S'il est clair qu'il ne doit être choisit trop petit pour bénéficier du gain de temps apporté par les descentes locales, ni trop grand d'après l'analyse théorique de la convergence, aucune heuristique n'a été établie pour le sélectionner de manière optimale. On a observé qu'elle variait entre la version accélérée et la version classique. Dépend-elle d'autres facteurs?

Un travail futur consisterait à utiliser l'algorithme de descentes alternées proposé pour généraliser l'approche adoptée ici pour résoudre d'autres problèmes de type ROF (avec des TV basées sur les valeurs singulières du gradient, par exemple), mais également à d'autres types de problèmes de forme proche. On pense en particulier à la déconvolution d'images, où la formulation variationnelle ne diffère que d'un opérateur linéaire dans le terme d'attache aux données.

Références

- Antonin CHAMBOLLE and Thomas POCK. A remark on accelerated block coordinate descent for computing the proximity operators of a sum of convex functions. SMAI Journal of Computational Mathematics, 1:29–54, 2015.
- [2] Laurent CONDAT. A direct algorithm for 1D total variation denoising. *IEEE Signal Processing Letters*, 20(11) :1054–1057, 2013.