Algorithmes rapides d'optimisation convexe. Applications à la reconstruction d'images et à la détection de changements.

Pierre Weiss

▶ **To cite this version:**

Pierre Weiss. Algorithmes rapides d'optimisation convexe. Applications à la reconstruction d'images et à la détection de changements.. Mathématiques [math]. Université Nice Sophia Antipolis, 2008. Français. NNT : . tel-00349452

HAL Id: tel-00349452

https://theses.hal.science/tel-00349452

Submitted on 31 Dec 2008

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

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

UNIVERSITÉ DE NICE - SOPHIA ANTIPOLIS - UFR Sciences

École Doctorale : Sciences et Technologies de l'Information et de la Communication

THÈSE

pour obtenir le titre de

Docteur en Sciences
de l'Université de Nice - Sophia Antipolis

Mention : Automatique, Traitement du signal et des images

par

Pierre WEISS

Algorithmes rapides d'optimisation convexe. Applications à la restauration d'images et à la détection de changements.

Thèse dirigée par Laure Blanc-Féraud et Gilles Aubert
et préparée à l'INRIA Sophia Antipolis dans le projet Ariana

Soutenance prévue le 21 Novembre 2008 devant le jury suivant :

M. Patrick Louis Combettes	Professeur Université Paris 6	Rapporteur
M. Jérôme Idier	Directeur de recherche, CNRS/IRCCyN	Rapporteur
M. Henri Maître	Professeur TELECOM ParisTech	Président
M. Antonin Chambolle	Directeur de recherche CNRS/CMAP	Examinateur
M. Lionel Moisan	Professeur Université Paris 5	Examinateur
Mme. Laure Blanc-Féraud	Directrice de recherche, CNRS/I3S	Directrice
M. Gilles Aubert	Professeur UNSA	Co-directeur

Table des matières

Table des matières **1**

Remerciements **5**

Introduction générale **9**

I Méthodes de premier ordre rapides pour l'optimisation convexe. Applications au traitement d'images **13**

1 Introduction **15**
1.1 Pourquoi l'optimisation convexe ? . . . 15
1.2 Le cadre mathématique . . . 18
1.3 Plan de la partie . . . 19
1.4 Contributions . . . 20

2 Notations et rappels d'optimisation convexe **21**
2.1 Notations . . . 21
2.2 Rappels d'optimisation convexe . . . 22
2.3 Quelques classes de fonctions convexes . . . 27
2.3.1 Fonctions Lipschitz différentiables . . . 27
2.3.2 Fonctions fortement convexes . . . 28
2.3.3 Fonctions simples . . . 29
2.3.4 Fonctions max . . . 31

3 Méthodes de premier ordre à un pas **33**
3.1 Méthodes à un pas : fonctions différentiables . . . 33
3.1.1 La descente de gradient dans le cas non contraint ($K = X$) . . . 34
3.1.2 La descente de gradient projeté dans le cas contraint ($K \subset X$) . . . 35
3.1.3 L'itération proximale pour minimiser la somme d'une fonction différentiable et d'une fonction simple . . . 36
3.2 Méthodes à un pas : fonctions non-différentiables . . . 37

3.2.1 La descente de sous-gradient projeté . . . 37
3.2.2 La descente de sous-gradient avec contraintes fonctionnelles . . . 39
3.2.3 Autres méthodes . . . 40
3.3 Rappels sur la théorie de la complexité . . . 40
3.3.1 Cas Lipschitz différentiable . . . 40
3.3.2 Cas non différentiable . . . 41
3.4 Conclusion . . . 41

4 Algorithmes de minimisation structurels multi-pas 43
4.1 Somme d'une fonction simple et d'une fonction Lipschitz . . . 44
4.1.1 Le principe des schémas multi-pas . . . 44
4.1.2 Comparaison avec les méthodes à un pas . . . 46
4.1.3 Réécriture de l'algorithme avec des fonctions prox . . . 46
4.1.4 Réécriture de l'algorithme avec des projecteurs . . . 47
4.2 Somme d'une fonction max et d'une fonction fortement convexe . . . 48
4.2.1 Idée de la méthode sur un exemple . . . 48
4.2.2 L'algorithme et son taux de convergence . . . 49
4.3 Somme de fonctions max et de fonctions simples . . . 53
4.3.1 Régularisation du problème primal . . . 54
4.3.2 Régularisation du problème dual . . . 56
4.3.3 Régularisation du problème primal-dual . . . 58
4.4 Conclusion . . . 59

5 Application à la restauration d'images 61
5.1 Deux types d'a priori convexes sur les images . . . 62
5.1.1 Les a priori de régularité . . . 63
5.1.2 Les représentations parcimonieuses . . . 63
5.2 Différents types de perturbations . . . 64
5.2.1 Les sources de perturbations . . . 64
5.2.2 La modélisation mathématique des perturbations . . . 66
5.2.3 Résumé . . . 73
5.3 Minimisation sous une contrainte générale . . . 73
5.3.1 Restauration avec critère de régularité . . . 74
5.3.2 Restauration avec critère de parcimonie . . . 80
5.4 Minimisation avec attache aux données l^2 . . . 82
5.4.1 Restauration avec critère de régularité . . . 82
5.4.2 Restauration avec critère de parcimonie . . . 84
5.5 Comparaison algorithmiques des deux a priori . . . 84
5.5.1 Comparaison pour une attache aux données générale . . . 84
5.5.2 Comparaison pour une attache aux données l^2 . . . 85
5.5.3 Une comparaison préliminaire de la qualité de restauration . . . 85
5.6 Mini-guide de l'utilisateur . . . 87
5.6.1 Lipschitz différentiable + Simple . . . 87
5.6.2 Max + Fortement convexe . . . 88
5.6.3 Max + Simple . . . 88

6 Décomposition d'images **89**
6.1 L'origine des modèles de décomposition . . . 89
6.2 Reformulation du problème . . . 90
6.2.1 Comparaison pratique des différents modèles . . . 91

7 Comparaisons expérimentales **95**
7.1 Le problème de Rudin-Osher-Fatemi . . . 95
7.2 Autres problèmes contraints . . . 100
7.3 Discussion . . . 102
7.3.1 Stabilité du schéma de Nesterov . . . 102
7.3.2 Comparaisons avec d'autres méthodes . . . 103

8 Conclusions et perspectives **105**

II Sur l'invariance des lignes de niveau aux changements d'illuminations. Application à la détection de changements. **107**

1 Introduction **109**
1.1 Un bref aperçu de la problèmatique . . . 109
1.1.1 Qu'est-ce que la détection de changements ? . . . 109
1.1.2 Le principe de la méthode variationnelle . . . 110
1.1.3 Les attributs d'images invariants aux conditions d'illumination . . . 110
1.1.4 L'algorithme et ses concurrents . . . 111
1.2 Plan de la partie et contributions . . . 113

2 La formation des images : perspective et modèle de Phong **115**
2.1 Notations . . . 115
2.2 Perspective, scène et lumière . . . 115
2.3 Hypothèses sur la surface et le modèle d'illumination . . . 116

3 Invariance des lignes de niveau aux changements d'illumination **119**
3.1 Définition et propriétés des lignes de niveau . . . 119
3.1.1 Définition . . . 119
3.1.2 Quelques résultats mathématiques . . . 120
3.2 Rappels de géométrie différentielle . . . 122
3.3 Invariance des lignes de niveau dans le cas régulier . . . 123
3.4 Invariance des lignes de niveau dans le cas non régulier . . . 127
3.5 Conclusion . . . 130

4 Application à la détection de changements **131**
4.1 Justification des hypothèses . . . 132
4.2 Notre algorithme de détection de changements . . . 134
4.3 Résultats et comparaisons . . . 136
4.3.1 Description d'approches alternatives . . . 136
4.3.2 Résultats sur des images de synthèse . . . 137
4.3.3 Résultats sur des images de télédétection . . . 137

4.4 Conclusion et perspectives 141

A Annexes **149**
A.1 Discrétisation du gradient et de la divergence (couleur) 149
A.2 Projections sur des boules l^p pondérées ($p \in \{1, 2, \infty\}$) 150
A.2.1 Projection sur une boule l^∞ pondérée 150
A.2.2 Projection sur une boule l^1 pondérée 150
A.2.3 Projection sur une boule l^2 pondérée 151
A.3 Une méthode rapide détaillée de A à Z : Décomposition d'images 152
A.3.1 Approximations différentiables 153
A.3.2 Schéma de minimisation 154
A.4 Et la couleur ? 155

Publications et activités de recherche et d'enseignement **157**

Bibliographie **161**

Résumés **171**

Remerciements

Quoiqu'en disent certains, une thèse contient son lot de moments difficiles [1] et ne pourrait pas être réalisée sans de nombreux soutiens humains. Nous commençons donc naturellement notre manuscrit en témoignant notre reconnaissance aux nombreuses personnes qui ont rendu plus facile son élaboration. Nous tenons aussi dès à présent à nous excuser auprès de tous ceux qui mériteraient d'être cités ici et n'apparaissent pas par mégarde.

Mes premières pensées vont naturellement à mes directeurs de thèse Laure Blanc-Féraud et Gilles Aubert qui m'ont guidé et épaulé avec sympathie tout au long de ces trois années de thèse. Je me permet de vous les recommander si vous souhaitez commencer un doctorat !

Je tiens ensuite à remercier chaleureusement Patrick-Louis Combettes et Jérôme Idier qui ont accepté de rapporter ce manuscrit spontanément. Leurs commentaires vont me permettre d'orienter mes futures recherches et je leur en suis très reconnaissant.

Je suis heureux qu'Antonin Chambolle, Henri Maître et Lionel Moisan aient accepté de participer à mon jury de thèse. Les travaux de A.Chambolle sont pour une bonne part à l'origine de la première partie de cette thèse. Quant à H. Maître et L. Moisan, j'ai eu la chance de suivre leurs cours lors de mon Master et ceux-ci ont fortement contibué à mon attrait pour la recherche ainsi qu'à la deuxième partie de cette thèse.

Outre mes directeurs, de nombreuses personnes m'ont aidé lors de mes travaux. Je tiens à remercier en premier lieu Alexis Baudour - mon collègue direct de travail et ami - puisqu'il a comblé certaines de mes lacunes mathématiques et m'a aidé à résoudre de nombreux problèmes obscurs. J'ai eu la chance de collaborer pendant 6 mois avec Alexandre Fournier sur le problème de la détection de changements. Cette collaboration a été riche et stimulante et j'espère renouveler de telles expériences à l'avenir. Omid Amini mérite aussi une mention très spéciale. Il nous a impressionné maintes fois par ses qualités de tireur à la pétanque, mais aussi et surtout par ses compétences mathématiques hors du commun. Il nous a considérablement aidé à résoudre les problèmes de topologie de la deuxième partie de ce manuscrit. Finalement merci à Mickael Carlavan, aux cotés duquel beaucoup de bugs ont disparus et un article pourrait apparaître.

De nombreuses discussions et collaborations ont influencé ce travail. Je pense notamment à celles que j'ai pu avoir avec D. Goldsztejn, T. André, J.F. Aujol, V. Prinet, J. Fadili, M. Antonini. Merci a vous tous.

Enfin, B. Bourgeois, T. Pitarque et A-M. Litovsky m'ont permis de réaliser mon monitorat dans de très bonnes conditions et je tiens à les remercier chaleureusement.

Certains des cours qui ont ponctué mon parcours d'étudiant (à l'oral ou lors de mes lectures) ont éveillé mon intérêt pour la science et m'ont ouvert de nouveaux horizons. Je pense notamment aux

[1] et bien d'autres encore ّ

cours de P. Guillaume, A. Huard, A. Bendali, S. Mallat, B. Polyak, Y. Nesterov, J-M. Morel.

Le long périple du doctorant a été supporté grâce à plusieurs relations professionnelles et extra-professionnelles. Soyons brefs : je remercie particulièrement et par ordre d'apparition François R., Marie R., Guillaume P. (et son vélo), Corinne M. (et son calme), Monsieur Xavier D., Pierre G., Aymen E., Pooran M., Alain L., Florian H., Tamara R. (y los cafecitos), Yann U. (et ses ballons), Giuseppe, Ting P., Olivier Z., Peter H. (Palinka ! !), Avick B., Sylvain V. (et son saxophone), Praveen P., Erwan L., Gabriel P., Dan Y., Sapna N., Maria K. and Colin G. (et leur mariage), Florent L., Marie A., Eric B. (et son smatch), Carlito P., Florent C. (et le Tali), Pedro M. (et son piano), Gabichou, Giovanni G., Michael Ng, Papito Daniele, Neismon F., Farsad K. (et les grands crus), Raymond C.

La fin de cette thèse s'est déroulée dans de bonnes conditions grâce au soutien important de 3 personnes : mes parents Pascale et Michel ainsi que ma mie Alice. A vous trois, merci beaucoup ! Qu'aurais-je fait sans vous ?

Mes derniers remerciements (et non les moindres) vont à Emilie, qui m'a rendu joyeux et supporté pendant une grande partie de cette thèse.

Introduction générale

La recherche sur le traitement numérique des images s'intensifie depuis quelques années. On peut trouver plusieurs causes à cette évolution. Premièrement, les ordinateurs, les téléphones portables et les caméras deviennent des outils de consommation courants et permettent la création de nouvelles applications, dans des domaines tels que la surveillance ou le multimédia. Deuxièmement, des avancées récentes dans les techniques d'acquisition de données (échographie, tomographie, échantillonnage compressif...) recquièrent des traitements numériques de plus en plus complexes. Si le développement à grande échelle des téléphones portables est un changement dont l'intérêt social, économique et écologique est discutable, il nous semble que les progrès plus fondamentaux dans l'observation de l'infiniment petit (microscopie), de l'infiniment grand (téléscopes), des planètes (satellites) ou de l'intérieur de la matière (tomographie ou échographie), peuvent encore mener à des avancées substantielles dans la compréhension du corps humain ou de l'univers. Ces problèmes méritent à notre avis que l'on s'y attarde, et nous apporterons une contribution à l'étude de certains d'entre eux dans ce manuscrit.

Ce manuscrit est divisé en deux parties indépendantes. Dans la première, nous proposons des algorithmes rapides pour la reconstruction d'images ou des applications récentes telles que la décomposition d'images en texture et en géométrie. Dans la deuxième partie, nous nous intéressons au problème de la détection de changements entre des paires d'images de télédétection. Dans les deux paragraphes suivants, nous motivons brièvement ces travaux.

Résolution rapide de problèmes d'optimisation

Les avancées technologiques et les applications envisagées aujourd'hui mènent à des problèmes numériques de plus en plus complexes. L'un des principaux fléaux à surmonter est le volume croissant des données. Donnons quelques exemples. L'image d'un appareil photo grand public est aujourd'hui de 4000×2600 pixels soit 10^7 pixels. Un pixel du satellite d'observation Quickbird représente une surface au sol de 60 par 60 centimètres. Une image de l'Alsace - ma région natale - est donc constituée de plus de 10^{10} pixels. Une séquence vidéo d'une heure trente sur un DVD contient environ 10^{11} pixels. Une image $3D$ d'une planète peut dépasser 10^{12} voxels ! Bien que la puissance des calculateurs augmente régulièrement, traiter de tels volumes de données en un temps raisonnable reste un défi majeur. De plus, les chercheurs rêvent aujourd'hui à de nouvelles techniques d'imagerie en "temps réel". On peut penser, par exemple, à l'acquisition d'images médicales lors d'une opération, en vue d'assister un chirurgien. Ces considérations font qu'il est crucial, lorsqu'on pose un problème, de savoir s'il modélise de façon pertinente le phénomène physique sous-jacent, mais aussi d'évaluer s'il peut ou s'il pourra être résolu efficacement.

Dans la première partie de ce manuscrit, nous allons nous intéresser à des méthodes de traitement d'images reposant sur des problèmes d'optimisation dits convexes. Une caractéristique appréciable de ces problèmes est justement que l'on sait les résoudre de façon approchée en des temps courts. Ceci leur vaut d'être utilisés et étudiés très largement. Par exemple, on trouve plus de 50 articles de revues internationales sur la minimisation de la variation totale. Une telle abondance est souhaitable, car elle permet au domaine de progresser rapidement, mais elle présente aussi un inconvénient : il devient difficile de savoir quelle méthode adopter. A quelques exceptions récentes près, on ne trouve pas dans la littérature d'évaluations théoriques de la rapidité de convergence des algorithmes. Celles-ci permettraient une évaluation rapide des méthodes.

Dans la première partie de ce manuscrit, nous essayons de remédier à ce problème. Nous présentons d'abord plusieurs techniques classiques d'optimisation. Nous évaluons systématiquement leur effica-

cité théorique. Nous proposons ensuite des méthodes originales en cherchant à maximiser leurs taux de convergence. Nous comparons ces algorithmes expérimentalement sur de nombreux problèmes. Il ressort de cette comparaison que le classement théorique des méthodes est effectivement respecté. De plus, les méthodes que nous proposons sont effectivement les plus rapides. Elles représentent à notre connaissance l'état de l'art des méthodes de premier ordre. Notons que ces résultats reposent essentiellement sur des travaux théoriques récents de Yurii Nesterov.

Ces algorithmes peuvent servir à la reconstruction de nombreuses modalités d'images (images optiques, tomographiques, radar,...). Dans ce manuscrit, nous considèrerons plusieurs types de perturbations telles que les bruits gaussiens, uniformes, impulsionnels, les bruits de compressions, les effets de flou ou encore l'échantillonnage irrégulier. Nous nous intéressons aussi à des problèmes tels que la décomposition d'images en texture et en géométrie ou la décomposition d'images sur des dictionnaires.

Détection de changements en milieu urbain

Dans la deuxième partie de cette thèse, nous nous concentrons sur une application précise du traitement d'images : la détection de changements entre deux images satellitaires de zones urbaines. Ce problème intéresse particulièrement les instituts de cartographie tels que l'IGN, qui désirent mettre à jour régulièrement des cartes des villes. La périphérie de certaines villes telles que Mexico ou Pékin croît aujourd'hui de manière incontrôlable et une telle approche est - étonnament - indispensable. La détection de changements peut aussi servir à l'évaluation de dégats après des catastrophes naturelles (tremblements de terres, tempêtes) ou moins naturelles (conflits armés). Finalement, plusieurs organismes aimeraient l'utiliser à des fins de surveillance (piscines illicites, détection d'installations nucléaires,...).

Ce problème est particulièrement difficile, et à notre connaissance, il n'existe pas encore de méthode vraiment fonctionnelle pour le résoudre, bien que plusieurs laboratoires se concentrent essentiellement sur ce problème. Une des difficultés principales à surmonter consiste à s'affranchir des conditions d'illuminations variables entre les deux prises de vue. Cet obstacle va nous mener, dans ce travail, à analyser l'invariance des lignes de niveau aux conditions d'éclairement. On obtient plusieurs résultats :

- Nous redéfinissons les lignes de niveau pour qu'elles jouissent de propriétés d'invariance plus satisfaisantes qu'avec la définition classique.
- Nous ciblons précisément quelles applications peuvent effectivement être construites à partir des lignes de niveau.
- En particulier, nous montrons qu'elles sont un bon outil pour la détection de changements sur des zones urbaines.
- Nous proposons alors un algorithme simple de détection reposant sur un problème d'optimisation convexe.
- Finalement, nous validons notre modèle expérimentalement sur des images de synthèse et des images naturelles. Cette étude montre une supériorité théorique et pratique de notre méthode par rapport à plusieurs approches courantes. Outre ses performances satisfaisantes, une autre propriété intéressante de cet algorithme est qu'il est très rapide (de l'ordre de la seconde sur une image 1000×1000).

Pour conclure cette introduction, nous proposons un petit jeu (figure 1). Il faut trouver sept différences

entre les deux images malgré un changement de contraste monotone global. Notre algorithme de détection de changements est capable de résoudre ce problème instantanément. Et vous ? Pour vérifier votre solution, il faudra bien lire ce manuscrit. La solution est cachée dedans...

FIG. 1 – Les sept différences. En haut : image au temps t_1. En bas : image au temps t_2. Problème : trouver les sept différences malgré le changement de contraste.

Première partie

Méthodes de premier ordre rapides pour l’optimisation convexe. Applications au traitement d’images

Chapitre 1

Introduction

1.1 Pourquoi l'optimisation convexe ?

Une grande partie de cette thèse a été consacrée à l'étude et à la résolution numérique de problèmes d'optimisation convexe. Je tiens à retracer ici le cheminement qui m'a amené à ces problèmes, car il est à l'origine de ma démarche scientifique et je pense qu'il permettra de mieux évaluer l'impact des résultats proposés dans cette thèse.

Le premier problème que mes directeurs de thèse m'ont proposé est la décomposition en texture et en géométrie d'images couleur ou multispectrales. Ce problème fait suite aux travaux de thèse de J.F. Aujol [Aujol *et al.*, 2005]. Une démarche naturelle pour effectuer cette décomposition consiste simplement à remplacer la variation totale classique par la variation totale couleur [Chan *et al.*, 2001; Blomgren et Chan, 1998] puis à utiliser l'algorithme convergent proposé par J.F. Aujol *et al.* [Aujol *et al.*, 2005]. Après quelques manipulations, le problème de décomposition d'une image couleur $f \in \mathbb{R}^n$ peut s'écrire sous la forme discrétisée suivante :

$$\min_{x+y=f} (F(x) + G(y)) \tag{1.1}$$

où F et G sont deux fonctions convexes non différentiables de $\mathbb{R}^n$. Les auteurs de [Aujol *et al.*, 2005] se sont aperçus qu'il peut être résolu plus facilement en pénalisant la contrainte. Il proposent de résoudre :

$$\min_{x,y} \left(F(x) + G(y) + \frac{\gamma}{2} |f - x - y|^2 \right) \tag{1.2}$$

où $\gamma \in \mathbb{R}^+$ doit être choisi grand. Leur algorithme s'écrit alors sous la forme suivante :

1. A y fixé, résoudre de façon itérative : $\min_x \left(F(x) + \frac{\gamma}{2} |f - x - y|^2 \right)$
2. A x fixé, résoudre de façon itérative : $\min_y \left(G(y) + \frac{\gamma}{2} |f - x - y|^2 \right)$

Ils montrent alors que cet algorithme converge vers la solution (x^*_γ, y^*_γ) de (1.2). Puis que :

$$\lim_{\gamma \to \infty} (x^*_\gamma, y^*_\gamma) = (x^*, y^*)$$

où (x^*, y^*) est la solution de (1.1). Cet algorithme a permis de résoudre le problème de décomposition de façon bien plus précise que les essais pionniers [Osher et Vese, 2003; Osher *et al.*, 2003]. Cependant de nombreuses questions restent en suspens :

– Quelle est l'influence de γ sur la précision de la solution ?
– Les problèmes 1. et 2. sont résolus de façon itérative et donc inexacte. Or la preuve de convergence repose sur le fait que ces deux sous-problèmes sont résolus de manière exacte. Quelle est l'influence des erreurs ?
– Comment choisir le nombre d'itérations aux étapes 1. et 2. ?

On présente dans cette thèse une méthode de résolution reposant elle aussi sur une perturbation du problème. On évalue précisément l'impact de cette perturbation et on montre qu'elle peut être diminuée de façon itérative lors du processus de minimisation. On donne un taux de convergence de notre méthode vers une solution du problème (1.1). L'algorithme proposé a pour seul paramètre la précision souhaitée ou le nombre d'itérations. Il a une efficacité nettement supérieure à l'algorithme proposé dans [Aujol *et al.*, 2005]. D'un point de vue pratique, ceci se traduit par une évaluation plus facile, précise et rapide des qualités et défauts du modèle de décomposition. Nous pensons donc avoir répondu convenablement à ces questions. La figure 1.1 montre un exemple de décomposition en utilisant l'algorithme proposé dans [Aujol *et al.*, 2005] et celui proposé dans ce manuscrit. Nous avons utilisé exactement les mêmes paramètres dans les deux méthodes. On peut voir que les résultats sont différents malgré les preuves de convergence. La solution de notre algorithme est plus proche du résultat exact, la fonctions coût est en effet plus basse.

Un autre problème auquel je me suis interessé pendant cette thèse est celui des "extensions Lipschitz minimisantes" ou problème d'Aronsson [Aronsson, 1967; Aronsson *et al.*, 2007]. Ce problème intervient assez naturellement dans des problèmes d'inpainting [Alvarez *et al.*, 1993; Caselles *et al.*, 1998] [1]. A notre connaissance sa meilleure application pratique en traitement du signal est l'interpolation de modèles numériques d'élevation [Almansa *et al.*, 2002]. Le problème est le suivant : soit Ω un ouvert borné et g une fonction Lipschitz définie sur la frontière $\partial\Omega$ de Ω. On cherche une fonction $u \in W^{1,\infty}(\Omega)$ telle que $u = g$ sur $\partial\Omega$ et qui minimise la constante de Lipschitz :

$$\sup_{x,y\in\Omega^2} \left(\frac{|u(x) - u(y)|}{|x - y|} \right)$$

La solution de ce problème n'est pas unique et parmi les solutions, on souhaite généralement trouver la solution de viscosité [Crandall *et al.*, 1992] (dans ce cas, celle qui minimise localement la constante de Lispchitz). On peut montrer que cette solution est la limite quand p tend vers l'infini du problème convexe suivant :

$$\inf_{u\in W^{1,p}(\Omega),\ u|_{\partial\Omega}=g} \left(\int_\Omega |\nabla u|^p d\Omega \right) \tag{1.3}$$

Pour trouver une solution au problème d'Aronsson nous avons essayé d'utiliser cette remarque. En posant $v = u - \phi$ où ϕ est un relèvement de g, le problème approché d'Aronsson devient :

$$\inf_{v\in W^{1,p}(\Omega),\ v|_{\partial\Omega}=0} \left(\int_\Omega |\nabla(v + \phi)|^p d\Omega \right) \tag{1.4}$$

On peut montrer que le problème dual associé s'écrit :

$$\sup_{q\in L^p(\Omega)^2,\ div(q)=0} \left(-\alpha(p) \int_\Omega |q|^{p'} d\Omega + \int_\Omega q\nabla\phi d\Omega \right) \tag{1.5}$$

[1] en pratique ce modèle est cependant mauvais pour interpoler des images naturelles car la donnée doit être Lipschitz sur le bord du domaine d'interpolation . Il interpole donc mal les contours !

FIG. 1.1 – Décomposition d'une image couleur. En haut : résultat obtenu avec l'algorithme des minimisations alternatives (30 minutes de calcul). En bas : résultat obtenu avec les méthodes proposées dans cette thèse (10 minutes pour décroître le saut de dualité d'un facteur 10^6).

où $\frac{1}{p'}+\frac{1}{p} = 1$ et $\alpha(p)$ est un paramètre réel. Nous avons tenté de résoudre ces problèmes numériquement en utilisant des éléments finis et des méthodes de premier ordre. Ces méthodes ne fonctionnent pas : la convergence est très mauvaise. Ces difficultés ont soulevé les questions suivantes :

- Quels sont les problèmes qui peuvent être minimisés efficacement par des méthodes de premier ordre ?
- Quand la dualité simplifie-t'elle les problèmes de minimisation ?

Nous avons partiellement répondu à ces questions. Une classe de fonctions qui peuvent être minimisées efficacement par des méthodes de premier ordre sont les fonctions Lipschitz différentiables (les normes l^p avec $p \neq 2$ ne le sont pas !). On montrera que les problèmes de minimisation d'une somme de fonctions convexes et de fonctions fortement convexes peuvent être résolus très efficacement en considérant les problèmes duaux associés. Ces derniers ont en effet un gradient Lipschitz.

En ce qui concerne le problème d'Aronsson, une solution élégante et efficace a été proposée pendant que nous travaillions sur ce problème. Nous référons le lecteur intéressé au travail de A. Oberman[Oberman, 2005]. Ce dernier propose une solution convergente reposant sur des résultats de [Barles et Souganidis., 1991].

1.2 Le cadre mathématique

On peut exprimer un problème d'optimisation dans différents cadres mathématiques. Différentes "communautés" scientifiques proposent des algorithmes de minimisation adaptés à chaque niveau d'abstraction. De façon très schématique on peut séparer trois communautés :

1. **Les informaticiens...** Pour eux les fonctions à minimiser vont d'un ensemble de cardinal fini tel que $\{0, 1, ..., 255\}^n$ dans un autre ensemble de cardinal fini. Cette formulation est la plus adaptée à la structure des ordinateurs. On peut trouver pour ces problèmes une solution globale en un nombre fini d'opérations. Ces problèmes peuvent cependant être extrêmement compliqués. La difficulté est de réduire l'espace de recherche le plus efficacement possible.

   Cette approche est très fructueuse puisqu'elle a permis à Jérôme Darbon (communication personnelle) de résoudre le problème de Rudin-Osher-Fatemi et le problème $BV-L^1$ en $O(n\log(n))$ opérations. C'est à ma connaissance l'approche la plus rapide pour résoudre ces problèmes (J. Darbon a montré qu'elle est optimale). Ces algorithmes restent cependant adaptés à des classes relativement restreintes de problèmes.
2. **Les numériciens...** Pour eux les fonctions à minimiser vont de $\mathbb{R}^n$ dans $\mathbb{R}$. Ce cadre nous semble assez naturel puisqu'en langage *C*, un nombre exprimé au format *double* a environ 15 chiffres significatifs. De plus les images numériques sont elles mêmes discrètes et formées d'un nombre n fini de pixels. Les algorithmes proposés par cette communauté peuvent souvent être étendus aux espaces fonctionnels. Dans cette première partie nous nous plaçons clairement dans ce cadre de travail.
3. **Les analystes...** Ils posent les problèmes d'optimisation dans des espaces fonctionnels tels que les espaces de Sobolev, les espaces de Besov, l'espace BV,... Cette démarche présente plusieurs avantages. Elle s'applique dans un cadre plus général que le précédent et se transpose de façon triviale au cas discret. Elle peut servir pour des traitements analogiques, bien que ceux-ci deviennent assez rares en traitement des images. Les preuves de convergence requièrent souvent plus d'hypothèses et peuvent apporter des informations plus fines sur le comportement des méthodes.

Bien entendu, les trois “communautés” interagissent largement. De plus, on peut souvent montrer que la solution d’un problème discrétisé converge vers la solution du problème continu sous-jacent lorsque les pas d’échantillonnage tendent vers 0. Par exemple, les développement limités permettent de montrer la convergence d’une solution discrète dans des espaces de fonctions régulières tels que $W^{1,2}$. Dans certains cas la théorie de la Gamma-convergence [Braides, 2002; Aubert et Kornprobst., 2006] permet de montrer une convergence des solutions discrètes vers les solutions continues dans des espaces non hilbertiens. Poser les problèmes dans $\mathbb{R}^n$ nous semble donc être une solution raisonnable et on le verra, fructueuse.

1.3 Plan de la partie

Cette partie est découpée en 7 chapitres. Nous les décrivons brièvement ci-dessous :

– *Chapitre 2 :* il contient essentiellement des notations et des rappels d’optimisation convexe. Un lecteur averti peut se contenter de lire les parties (2.3.3) et (2.3.4) qui contiennent des définitions originales (fonctions simples et fonctions max). Tout au long du manuscrit, des renvois sont faits vers cette partie.
– *Chapitre 3 :* c’est un chapitre bibliographique. Il contient des rappels sur les taux de convergence des schémas de minimisation de premier ordre à un pas (peu connus et récents). On montre que la plupart des méthodes d’optimisation à un pas (descentes de gradient et itérations proximales notamment) convergent en $O\left(\frac{1}{k}\right)$ où k est le nombre d’itérations. On rappelle ensuite que la théorie de la complexité indique que les schémas de minimisation de premier ordre ont une efficacité majorée.
– *Chapitre 4 :* il contient les contributions méthodologiques principales introduites dans cette thèse. Nous présentons d’abord un schéma d’optimisation multi-pas de premier ordre adapté à la minimisation de la somme d’une fonction simple et d’une fonction Lipschitz différentiable. Ce schéma a été proposé par Y. Nesterov [Nesterov, 2007a] et converge en $O\left(\frac{1}{k^2}\right)$. Nous montrons ensuite comment étendre cet algorithme au problème de la minimisation de la somme d’une fonction max et d’une fonction fortement convexe. On présente ensuite une méthode d’optimisation en $O\left(\frac{1}{k}\right)$ adaptée à la minimisation de la somme d’une fonction max et d’une fonction simple. Ces taux de convergence théoriques représentent à notre connaissance l’état de l’art.
– *Chapitre 5 :* nous montrons d’abord que de nombreux problèmes de restauration d’images (débruitage, décompression, déconvolution, agrandissement...) peuvent s’exprimer sous un formalisme unifié : on minimise une fonction convexe non-différentiable (ou mal conditionnée) sous des contraintes s’exprimant comme des normes l^p de transformées affines. Nous montrons ensuite comment appliquer les algorithmes du chapitre 4 pour résoudre ces problèmes.
– *Chapitre 6 :* nous montrons comment appliquer les algorithmes du chapitre 4 pour décomposer des images en texture et en géométrie. Nous proposons (encore) un nouvel algorithme de décomposition. Et montrons sa supériorité pratique.
– *Chapitre 7 :* nous comparons les méthodes proposées dans cette thèse avec plusieurs schémas d’optimisation utilisés récemment. Ces comparaisons donnent un avantage très net aux méthodes proposées et confirment les résultats théoriques.
– *Annexes :* les annexes contiennent certains détails nécessaires pour réimplémenter les algorithmes. On présente notamment la discrétisation des opérateurs gradient et divergence pour les images multispectrales, des algorithmes rapides de projection sur des boules l^p pondérées, et on

détaille entièrement une méthode de résolution du problème de décomposition en texture et en géométrie de Y. Meyer.

1.4 Contributions

Nous pensons que cette première partie contient des contributions d'ordre méthodologiques, applicatif et de synthèse :

- *Méthodologique :* nous présentons plusieurs schémas d'optimisation originaux. Ces schémas sont très largement inspirés des travaux de Y. Nesterov [Nesterov, 2005b; Nesterov, 2007a; Nesterov, 2004; Nesterov, 2005a]. Cependant, nous étendons ses travaux à des classes de problèmes plus larges. De plus, nous proposons des variantes qui s'avèrent plus efficaces en pratique.
- *Applicatif :* les algorithmes proposés sont très performants. Par exemple, le problème de Rudin-Osher-Fatemi peut être résolu à raison de 3 images 300×300 par seconde sur un ordinateur portable. Nous résolvons certains problèmes de façon précise, ce qui n'était pas le cas jusqu'à présent. Ceci permet par exemple d'évaluer avec précision les limites des modèles de décompression ou de décompositions d'images. De plus, la rapidité de ces algorithmes peut encourager les industriels à utiliser les modèles reposant sur des problèmes d'optimisation.
- *Synthèse :* nous présentons de façon synthétique les résultats de convergence des méthodes d'optimisation de premier ordre. Nous détaillons notamment de vastes classes de problèmes qui peuvent être résolus efficacement. A notre connaissance les résultats de convergence présentés représentent l'état de l'art des méthodes de premier ordre. Nous présentons aussi de façon unifiée le problème de la restauration d'images. Ceci nous permet d'utiliser un algorithme unique pour résoudre de très nombreux problèmes à première vue éloignés. Ceux-ci étaient jusqu'à présent résolus avec des algorithmes ayant une efficacité très variable.

Chapitre 2

Notations et rappels d'optimisation convexe

2.1 Notations

Dans cette partie, nous décrivons quelques notations que nous utiliserons tout au long de manuscrit.

Nous noterons $X = \mathbb{R}^n$ et $Y = \mathbb{R}^m$. Soit :

$$\min_{x \in K \subset X} (f(x)) \tag{2.1}$$

un problème d'optimisation. X^* représente l'ensemble de ses minimiseurs. x^* représente un de ses minimiseurs. $f^* = f(x^*)$ est la valeur du minimum.

Soit $x \in X$, $x_i \in \mathbb{R}$ représente la i-ème composante de x.

$\langle \cdot, \cdot \rangle_X$ représente un produit scalaire sur X. Sauf si on le précise, ce produit scalaire sera le produit scalaire canonique. Pour $x^1, x^2 \in X$ on aura :

$$\langle x^1, x^2 \rangle_X = \sum_{i=1}^{n} x_i^1 x_i^2 \tag{2.2}$$

$\langle \cdot, \cdot \rangle_Y$ représente le produit scalaire sur Y. Sauf si on le précise, ce produit scalaire sera le produit scalaire canonique.

$|\cdot|_X$ et $|\cdot|_Y$ représentent les normes associées aux produits scalaires. Pour $x \in X$, on a :

$$|x|_X = \sqrt{\langle x, x \rangle} \tag{2.3}$$

$|\cdot|_{l^p(X)}$, $p \in [1, \infty[$ est la norme l^p sur X :

$$|x|_{l^p(X)} = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p} \tag{2.4}$$

$|\cdot|_{l^\infty(X)}$ est la norme l^∞ sur X :

$$|x|_{l^\infty(X)} = \max_{i \in \{1,2,\ldots,n\}} (|x_i|) \tag{2.5}$$

Soit $A : X \to Y$ une transformée linéaire. A^* représente son conjugué complexe. A^{-*} représente le conjugué complexe de A^{-1}.

Définition 2.1 (Norme euclidienne d'un opérateur)**.** *Soit $B : X \to Y$ un opérateur linéaire. La norme de B est définie par :*

$$\|B\| = \max_{x \in X, |x|_X \leq 1} (|Bx|_Y)$$

2.2 Rappels d'optimisation convexe

Dans cette partie, on rappelle très rapidement les notions d'analyse convexe indispensables à la compréhension de ce manuscrit. Le lecteur intéressé par la preuve des différentes propositions énoncées peut se référer aux manuscrits classiques [Rockafellar, 1970; Ekeland et Temam, 1976; Hiriart-Urruty et Lemarechal, 1996; Polyak, 1987; Nesterov, 2004].

Définition 2.2 (Domaine d'une fonction)**.** *Soit $f : X \to \overline{\mathbb{R}}$ une fonction. Le domaine de f noté $dom(f)$, est défini par $dom(f) = \{x \in X, f(x) < \infty\}$.*

Par la suite on supposera toujours que $dom(f) \neq \emptyset$.

Définition 2.3 (Fonction convexe)**.** *Soit $f : X \to \overline{\mathbb{R}}$ une fonction. Cette fonction est dite convexe si $dom(f)$ est convexe et si $\forall \lambda \in [0,1]$, $\forall (x,y) \in dom(f)$:*

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y) \tag{2.6}$$

La proposition suivante peut servir à prouver la convergence de certains algorithmes :

Proposition 2.1. *Soit $f : X \to \overline{\mathbb{R}}$ une fonction convexe et $x \in int(dom(f))$. Alors f est localement Lipschitz continue en x. Ceci implique notamment que f est continue sur $int(dom(f))$.*

Pour étudier les fonctions convexes, on a très souvent besoin d'une notion similaire : les ensembles convexes.

Définition 2.4 (Ensemble convexe)**.** *Soit $K \subset X$ un ensemble. Cet ensemble est dit convexe si $\forall (x,y) \in K$, $\forall \lambda \in [0,1]$:*

$$\lambda x + (1-\lambda)y \in K \tag{2.7}$$

La proposition suivante relie les deux notions :

Proposition 2.2. *Soit $f : X \to \overline{\mathbb{R}}$ une fonction convexe. Alors $\forall \alpha \in \overline{\mathbb{R}}$ les ensembles $\{x, f(x) \leq \alpha\}$ sont soit convexes soit vides.*

Définition 2.5 (Projecteur)**.** *Soit $K \subset X$ un ensemble convexe. Le projecteur sur K est défini par :*

$$\Pi_K(x) = arg \min_{u \in K} (|u - x|_X)$$

Définition 2.6 (Fonction propre)**.** *Une fonction convexe F sur X est propre si et seulement si F n'est pas identiquement égale à l'infini $+\infty$ et qu'elle n'atteint pas la valeur $-\infty$ sur X.*

Définition 2.7 (Fonction indicatrice)**.** *Soit $K \in X$ un sous-ensemble convexe non vide de X. La fonction indicatrice de K, notée $\chi_K : X \to \overline{\mathbb{R}}$ est définie par :*

$$\chi_K(x) = \begin{cases} 0 & si\ x \in K \\ \infty & sinon \end{cases} \tag{2.8}$$

Définition 2.8 (Sous-différentiel et sous-gradient)**.** *Soit $f : X \to \overline{\mathbb{R}}$ une fonction convexe. Le sous-différentiel de f au point $x \in int(dom(f))$ est défini par :*

$$\partial f(x) = \{\eta \in X, f(x) + \langle \eta, (y - x)\rangle_X \leq f(x), \forall y \in X\} \tag{2.9}$$

$\eta \in \partial f(x)$ est appelé sous-gradient.

Le sous-différentiel au point x a une interprétation géométrique simple. C'est l'ensemble des hyperplans qui passent par le point $(x, f(x))$ et dont le graphe passe sous le graphe de f. $\partial f(x)$ est réduit à un singleton si et seulement si f est dérivable en x. Dans ce cas, $\partial f(x) = \nabla f(x)$.

On considère maintenant le problème d'optimisation suivant :

$$\min_{x \in X}(f(x)) \tag{2.10}$$

où $f : X \to \overline{\mathbb{R}}$. On rappelle quelques résultats d'optimisation convexe qu'on utilisera tout au long du manuscrit :

Théorème 2.1. *Si f est convexe alors l'ensemble des minimiseurs X^* est convexe.*

Théorème 2.2. *Si f est convexe et différentiable alors les deux propositions suivantes sont équivalentes :*
- *x appartient à l'ensemble X^* des solutions de (2.10).*
- *$\nabla f(x) = 0$.*

Théorème 2.3. *Si f est convexe alors les deux propositions suivantes sont équivalentes :*
- *x appartient à l'ensemble X^* des solutions de (2.10).*
- *$0 \in \partial f(x)$.*

Les théorèmes suivants seront souvent appliqués (implicitement) dans le manuscrit [Polyak, 1987], page 135 :

Théorème 2.4. *Soit $f : X \to \overline{\mathbb{R}}$ une fonction convexe qui admet un minimum unique x^*. Toute suite de points satisfaisant $\lim_{k\to\infty} f(x^k) = f(x^*)$ converge vers x^*.*

Théorème 2.5. *Soit $f : X \to \overline{\mathbb{R}}$ une fonction convexe qui admet un ensemble de minima X^*. Soit $\{x^k\}$ une suite de points satisfaisant $\lim_{k\to\infty} f(x^k) = f(x^*)$. Toute sous-suite convergente de x^k converge vers un point de X^*.*

Pour éviter les confusions rappelons une appellation classique :

Définition 2.9 (Taux de convergence linéaire)**.** *Une méthode d'optimisation a un taux de convergence linéaire si elle génère une suite d'estimées $\{x^k\}$ qui satisfont :*

$$|x^k - x^*|_X \leq t^k |x^0 - x^*|_X \tag{2.11}$$

où $t \in]0, 1[$ définit le taux de convergence. Parfois on appelle aussi ce taux, taux de convergence géométrique.

Il est important de remarquer que taux de convergence linéaire **ne signifie pas** $f(x^k) - f(x^*) \leq \frac{C}{k}$. Un tel taux de convergence est dit sous-linéaire.

Nous présentons maintenant un théorème de dualité qui reviendra lui aussi très souvent par la suite. Notons qu'il existe différents théorèmes de dualité : la dualité de Kuhn-Tucker, la dualité de Fenchel, la dualité min-max... Nous ne présentons ici que la dualité min-max qui s'avère simple et suffisante pour les applications présentées dans cette thèse. Notre référence pour la dualité est [Ekeland et Temam, 1976], p. 176 :

Théorème 2.6. *Soient $A \subset X$ et $B \subset Y$ deux sous-ensembles fermés, convexes, non vides. Soit $L : A \times B \to \overline{\mathbb{R}}$ une fonction. On suppose :*

$$x \mapsto L(x,y) \ \text{ convexe propre } \ \forall y \in B \tag{2.12}$$

$$y \mapsto L(x,y) \ \text{ concave propre } \ \forall x \in A \tag{2.13}$$

On suppose que A est borné ou qu'il existe $y_0 \in B$ tel que $\lim_{|x|_X \to \infty} L(x, y_0) = +\infty$.
On suppose que B est borné ou qu'il existe $x_0 \in A$ tel que $\lim_{|y|_Y \to \infty} L(x_0, y) = -\infty$.
Alors on a :

$$\min_{x \in A} \left(\max_{y \in B} (L(x,y)) \right) = \max_{y \in B} \left(\min_{x \in A} (L(x,y)) \right) \tag{2.14}$$

Dans les cas pratiques que l'on considèrera dans ce manuscrit, les conditions du théorème seront toujours vérifiées. Nous laisserons souvent le soin au lecteur de s'en convaincre. Nous présentons maintenant un théorème d'optimisation convexe que nous avons démontré pendant notre stage de master. Ce théorème est sans doute connu, mais nous n'avons trouvé aucun livre qui le présente. De nombreux problèmes reviennent à minimiser une somme pondérée de fonctionnelles convexes. On s'intéresse ici à ce problème assez général :

Théorème 2.7. *Soient F et G deux fonctions convexes de $\mathbb{R}^n$ dans $\mathbb{R}^+$, et λ un réel de $\mathbb{R}^+$. Soit U_λ l'ensemble des solutions du problème suivant :*

$$\inf_{u \in \mathbb{R}^n} (F(u) + \lambda G(u)) \tag{2.15}$$

Soit U_α l'ensemble des solutions du problème suivant :

$$\inf_{u,\ G(u) \leq \alpha} (F(u)) \tag{2.16}$$

On a alors :

$$\forall \lambda, \forall u_\lambda \in U_\lambda, \exists \alpha \in \mathbb{R}^+, u_\lambda \in U_\alpha \tag{2.17}$$

$$\forall \alpha, \forall u_\alpha \in U_\alpha, \exists \lambda \in \mathbb{R}^+, u_\alpha \in U_\lambda \tag{2.18}$$

Résoudre (2.15) ou (2.16) permet donc d'obtenir des solutions identiques[1].

[1] On ne peut pas obtenir de résultat plus fort. En effet, on peut trouver F et G telles que $U_\lambda \not\subseteq U_\alpha$ ou $U_\alpha \not\subseteq U_\lambda$.

Démonstration. La preuve de (2.17) est simple. Il suffit en effet de choisir $\alpha = G(u_\lambda)$.

La preuve de (2.18) est plus technique. On commence par remarquer que l'équation (2.15) peut être réécrite sous la forme :

$$\inf_{\alpha\in\mathbb{R}^+}\left(\inf_{u,G(u)=\alpha}(F(u)+\lambda G(u))\right) \tag{2.19}$$

Soit encore :

$$\inf_{\alpha\in\mathbb{R}^+}\left(\inf_{u,G(u)=\alpha}(F(u))+\lambda\alpha\right) \tag{2.20}$$

(2.20) est équivalent à :

$$\inf_{\alpha\in\mathbb{R}^+}\left(\inf_{u,G(u)\leq\alpha}(F(u))+\lambda\alpha\right) \tag{2.21}$$

En effet, soient m_1 et m_2 les valeurs des infima dans (2.20) et (2.21) respectivement. On a évidemment $m_2 \leq m_1$. On a aussi $m_1 \leq m_2$. Si ce n'était pas le cas, il existerait u_2 avec $G(u_2) \leq \alpha$ tel que :

$$F(u_2)+\lambda G(u_2) < \inf_{\alpha\in\mathbb{R}^+}\left(\inf_{u,G(u)=\alpha}(F(u)+\lambda G(u))\right)$$

ce qui est absurde. Résoudre (2.15) revient donc à minimiser :

$$\phi_\lambda(\alpha) = \inf_{u,G(u)\leq\alpha}(F(u))+\lambda\alpha \tag{2.22}$$

Pour montrer (2.18), il suffit de vérifier que pour tout α, il existe un λ tel que ϕ_λ atteigne son minimum en α. Pour ce faire, nous utilisons le lemme suivant (démontré par la suite). La fonction Ψ définie par $\Psi(\alpha) = \inf_{u,G(u)\leq\alpha}(F(u))$ est convexe.

Ψ est donc dérivable à gauche et à droite. On note Ψ'_+ et Ψ'_- ses dérivées à droite et à gauche. De même, on note $\phi'_{\lambda+}$ et $\phi'_{\lambda-}$ les dérivées à gauche et à droite de ϕ_λ. Cette fonction est convexe et atteint son minimum pour les points x tels que $\phi'_+(x) \geq 0$ et $\phi'_-(x) \leq 0$. En choisissant $\lambda \in [-\Psi'_+(\alpha), -\Psi'_-(\alpha)]$ on s'assure que ϕ_λ atteint son minimum en α ce qui termine la démonstration.

□

Lemme 2.1. *Si F et G sont deux fonctionnelles convexes de $\mathbb{R}^n$, alors la fonction :*

$$\Psi(\alpha) = \inf_{u,G(u)\leq\alpha}(F(u)) \tag{2.23}$$

est convexe.

Démonstration. On va démontrer que :

$$\Psi\left(\frac{\alpha+\beta}{2}\right) \leq \frac{\Psi(\alpha)+\Psi(\beta)}{2}$$

Les fonctions F et G sont convexes, elles vérifient donc :

$$F\left(\frac{g_1+g_2}{2}\right) \leq \frac{F(g_1)+F(g_2)}{2} \tag{2.24}$$

$$G\left(\frac{g_1+g_2}{2}\right) \leq \frac{G(g_1)+G(g_2)}{2} \tag{2.25}$$

La deuxième inéquation nous permet d'obtenir les inclusions suivantes :

$$\begin{aligned} & \{(u_1,u_2), G(u_1) \leq \alpha \text{ et } G(u_2) \leq \beta\} && (2.26)\\ \subset\ & \{(u_1,u_2), G(u_1)+G(u_2) \leq \alpha+\beta\} && (2.27)\\ \subset\ & \left\{(u_1,u_2), G\left(\frac{u_1+u_2}{2}\right) \leq \frac{\alpha+\beta}{2}\right\} && (2.28)\end{aligned}$$

On peut donc écrire :

$$\begin{aligned} \Psi\left(\frac{\alpha+\beta}{2}\right) &= \inf_{u, G(u)\leq\frac{\alpha+\beta}{2}} (F(u)) \\ &= \inf_{(u_1,u_2), G\left(\frac{u_1+u_2}{2}\right)\leq\frac{\alpha+\beta}{2}} \left(F\left(\frac{u_1+u_2}{2}\right)\right) && (2.29)\\ &\leq \inf_{(u_1,u_2), G(u_1)\leq\alpha \text{ et } G(u_2)\leq\alpha} \left(\frac{F(u_1)+F(u_2)}{2}\right) && (2.30)\\ &= \frac{1}{2}\inf_{u_1, G(u_1)\leq\alpha}(F(u_1)) + \frac{1}{2}\inf_{u_2, G(u_2)\leq\beta}(F(u_2)) && (2.31)\\ &\leq \frac{\Psi(\alpha)+\Psi(\beta)}{2} && (2.32)\end{aligned}$$

□

L'intérêt de ce théorème est que dans certaines situations, il est plus facile de trouver les solutions du problème (2.16) que celles du problème (2.15).

Nous présentons maintenant un lemme dont on se servira plusieurs fois au cours de la thèse. Bien qu'il semble être classique, nous ne l'avons pas trouvé dans les livres standards d'optimisation. On propose une démonstration simple ci-après.

Lemme 2.2. *Soit $\varphi : Y \to \overline{\mathbb{R}}$ une fonction convexe fermée. Soit :*

$$\hat{\phi}(x) = \min_{y\in X} (\langle y, x\rangle_X + \varphi(y)) \tag{2.33}$$

La fonction $\hat{\phi}$ est concave. De plus elle satisfait $\forall x \in int(dom(\hat{\phi}))$, $\partial\hat{\phi}(x) = Y(x)$, où $Y(x)$ est l'ensemble des minimiseurs de (2.33). En particulier, si φ est strictement convexe alors $\hat{\phi}$ est définie sur tout $\mathbb{R}^n$, elle est différentiable et si on note $y(x)$ l'unique minimiseur de (2.33) on a :

$$\nabla\hat{\phi}(x) = y(x) \tag{2.34}$$

Démonstration. $\hat{\phi}$ est concave car c'est le minimum de fonctions linéaires en x. Soit $x \in int(dom(\hat{\phi}))$ et $\eta \in Y(x)$. On a par définition :

$$\hat{\phi}(x) = \langle\eta, x\rangle_Y + \varphi(\eta) \tag{2.35}$$

De plus on a pour tout $x_0 \in X$:

$$\hat{\phi}(x_0) \leq \langle\eta, x_0\rangle_X + \varphi(\eta) \tag{2.36}$$

En soustrayant ces deux inégalités, on obtient pour tout $x_0 \in X$:

$$\hat{\phi}(x_0) \leq \hat{\phi}(x) + \langle \eta, x_0 - x \rangle_X \tag{2.37}$$

D'où $\eta \in \partial\hat{\phi}(x)$. Et donc :

$$Y(x) \subset \partial\hat{\phi}(x) \tag{2.38}$$

Montrons maintenant l'inclusion inverse. On raisonne par l'absurde. Soit $x \in int(dom(\hat{\phi}))$. Supposons qu'il existe $\eta \notin Y(x)$ tel que $\eta \in \partial\hat{\phi}(x)$. On a donc :

$$\hat{\phi}(x_0) \leq \hat{\phi}(x) + \langle \eta, x_0 - x \rangle_Y \ \forall x_0 \in X \tag{2.39}$$

$$\hat{\phi}(x) < \langle \eta, x \rangle_Y + \varphi(\eta) \tag{2.40}$$

En additionnant les deux inégalités on obtient pour tout $x_0 \in Y$, $\hat{\phi}(x_0) < \varphi(\eta) + \langle \eta, x_0 \rangle_X$. Donc $\eta \in dom(\varphi)$ et $0 \notin \partial\varphi(\eta) + x_0$ pour tout $x_0 \in X$. On a donc $\partial\varphi(\eta) = \emptyset$. Or on a supposé φ convexe, fermée. Son sous-différentiel n'est donc pas réduit à l'ensemble vide sur son domaine.

□

2.3 Quelques classes de fonctions convexes

Dans cette partie, nous décrivons certaines classes de fonctions convexes. Nous verrons par la suite que les problèmes de minimisation composés de combinaisons linéaires de telles fonctions peuvent généralement être minimisés de façon efficace. Nous donnons ci-après quelques unes de leurs propriétés remarquables.

2.3.1 Fonctions Lipschitz différentiables

Définition 2.10 (Fonction Lipschitz différentiable)**.** *Soit $f : dom(f) \to \mathbb{R}$ une fonction différentiable. f est dite Lipschitz différentiable si :*

$$\forall (x, y) \in dom(f),\ |\nabla f(x) - \nabla f(y)| \leq L|x - y| \tag{2.41}$$

La classe des fonctions Lipschitz différentiables est stable par addition :

Lemme 2.3. *Soient f_1 et f_2 deux fonctions respectivement L_1 et L_2 Lipschitz différentiables. Soient α_1 et α_2 deux réels positifs. Dans ce cas $\alpha_1 f_1 + \alpha_2 f_2$ est $\alpha_1 L_1 + \alpha_2 L_2$ Lipschitz différentiable.*

La proposition suivante [Polyak, 1987; Nesterov, 2005b] est très souvent utilisée pour démontrer la convergence de schémas de minimisation ainsi que pour évaluer leur taux de convergence.

Lemme 2.4. *Une fonction L-Lipschitz différentiable satisfait $\forall (x, y) \in dom(f)$:*

$$|f(y) - f(x) - \langle \nabla f(x), y - x \rangle| \leq \frac{L}{2}|y - x|^2 \tag{2.42}$$

Ceci signifie qu'une fonction Lipschitz différentiable est comprise entre deux fonctions quadratiques. Pour une fonction convexe, on obtient un résultat plus fort :

Lemme 2.5. *Si f est convexe et Lipschitz-différentiable alors on obtient les inégalités suivantes :*

$$0 \leq f(y) - f(x) - \langle \nabla f(x), y - x \rangle_X \leq \frac{L}{2}|x - y|_X^2 \tag{2.43}$$

Ces lemmes permettent de démontrer assez facilement la convergence d'une descente de gradient. Considérons en effet l'itération suivante :

$$x^{k+1} = x^k - \tau \nabla f(x^k) \tag{2.44}$$

Dans ce cas, on a :

$$\begin{aligned} f(x^{k+1}) &= f(x^k - \tau \nabla f(x^k)) && (2.45)\\ &\leq f(x^k) + \langle \nabla f(x^k), x^{k+1} - x^k \rangle_X + \frac{L}{2}|x^{k+1} - x^k|_X^2 && (2.46)\\ &\leq f(x^k) + (\tau^2 \frac{L}{2} - \tau)|\nabla f(x^k)|_X^2 && (2.47)\\ & && (2.48) \end{aligned}$$

Le terme $\tau^2 \frac{L}{2} - \tau$ est strictement négatif pour $\tau < \frac{2}{L}$ et il admet un minimum en $\tau = \frac{1}{L}$. En faisant ce choix, on s'assure que :

$$f(x^{k+1}) \leq f(x^k) - \frac{1}{2L}|\nabla f(x^k)|_X^2 \tag{2.49}$$

A notre connaissance, tous les calculs de taux de convergence des méthodes de gradients s'appuient sur (2.49). Les exemples de fonctions Lipschitz différentiables que nous utiliserons dans ce manuscrit sont les suivantes :

- $f(x) = \frac{x^2}{2}$ est 1-Lipschitz differentiable.
- $f(x) = \sqrt{x^2 + \mu^2}$ est $\frac{1}{\mu}$-Lipschitz differentiable.
- $f(x) = \begin{cases} |x| & \text{si} |x| \geq \mu \\ \frac{|x|^2}{2\mu} + \frac{\mu}{2} & \text{sinon} \end{cases}$ est $\frac{1}{\mu}$-Lipschitz différentiable.

2.3.2 Fonctions fortement convexes

Les fonctions fortement convexes sont utilisées abondamment dans ce manuscrit. Nous verrons qu'elles peuvent souvent être minimisées plus rapidement que les fonctions Lipschitz différentiables.

Définition 2.11 (Fonction fortement convexe)**.** *Une fonction $f : X \to \overline{\mathbb{R}}$ est dite fortement convexe si $\forall \lambda \in [0,1],\ \forall (x,y) \in dom(f)$:*

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y) - \frac{l}{2}\lambda(1-\lambda)|x - y|_X^2 \tag{2.50}$$

Le paramètre $l \geq 0$ est appelé paramètre de convexité. La classe des fonctions fortement convexe est stable par addition :

Lemme 2.6. *Soient f_1 et f_2 deux fonctions fortement convexes de paramètres de convexité respectif l_1 et l_2 (éventuellement nuls). Soient α_1 et α_2 deux réels positifs. Alors $\alpha_1 f_1 + \alpha_2 f_2$ est une fonction fortement convexe de paramètre de convexité $\alpha_1 l_1 + \alpha_2 l_2$.*

Pour rendre un problème convexe fortement convexe, on peut donc le perturber légèrement en lui ajoutant une fonction fortement convexe de faible amplitude sur le domaine de minimisation. Ce résultat sera utilisé plusieurs fois par la suite. La définition de la forte convexité permet de montrer le résultat central suivant :

Lemme 2.7. *Soit $f : X \to \overline{\mathbb{R}}$ une fonction fortement convexe de paramètre de convexité l. Alors $\forall (x,y) \in dom(f),\ \forall \eta \in \partial f(x)$:*

$$f(y) \geq f(x) + \langle \eta, y - x \rangle + \frac{l}{2}|y - x|_X^2 \tag{2.51}$$

De plus f admet un unique minimum x^* sur $dom(f)$ et pour tout $x \in dom(f)$:

$$f(x) \geq f(x^*) + \frac{l}{2}|x - x^*|_X^2 \tag{2.52}$$

La dernière inégalité permet de passer d'un taux de convergence sur l'objectif à un taux de convergence sur la distance au minimiseur.

2.3.3 Fonctions simples

Les fonctions qu'on appelle "simples" dans ce manuscrit n'ont pas encore de nom consacré dans la littérature. Nous les définissons à partir de l'opérateur prox [Moreau, 1962; Moreau, 1963; Combettes et Wajs, 2005].

Définition 2.12 (Opérateur prox)**.** *Soit $\Psi : X \to \overline{\mathbb{R}}$ une fonction convexe. L'opérateur :*

$$\begin{aligned} prox_{\Psi} :\ & X \to X \\ & x \mapsto \arg\min_{y \in X} \left(\Psi(y) + \tfrac{1}{2}|x - y|_X^2 \right) \end{aligned} \tag{2.53}$$

est appelé opérateur proximal associé à Ψ.

Définition 2.13 (Fonction simple)**.** *Une fonction $\Psi : X \to \overline{\mathbb{R}}$ est dite "simple" si elle est convexe et que $prox_{\lambda\Psi}(x)$ peut être calculé exactement pour tout $\lambda \in \mathbb{R}^+$ et tout $x \in X$.*

Notons que le problème (2.53) peut être résolu de manière analytique ou avec une stratégie de minimisation exacte. Pour mieux appréhender cette définition, nous donnons ci-dessous quelques exemples de fonctions simples :

- Si Ψ est la fonction caractéristique d'un ensemble convexe Q (cette fonction est notée χ_Q), alors le problème (2.53) est un problème de projection. On a donc :

  $$prox_{\chi_Q}(x) = \Pi_Q(x)$$

  Les fonctions caractéristiques sont donc simples si on sait effectuer une projection sur leur domaine de manière exacte.
- Les fonctions $\Psi(x) = |x|_{l^p(X)}$ avec $p \in \{1, 2, \infty\}$ sont simples. [2]

[2] Le cas de la norme l^∞ est le plus difficile à résoudre. Le calcul peut se faire avec un algorithme de tri en $O(n \log(n))$.

– Les fonctions qui s'écrivent sous la forme $\Psi(x) = \sum_{i=1}^{n} \phi(x_i)$ sont simples si on sait résoudre le problème univarié suivant :

$$\min_{\alpha \in \mathbb{R}} \left((\gamma - \alpha)^2 + \lambda \phi(\alpha) \right)$$

où $\gamma \in \mathbb{R}$ et $\lambda \in \mathbb{R}^+$.

– Une fonction du type $\Psi(x) = |Ax|_{l^p(Y)}$ où $A : X \to Y$ est une transformée linéaire et $p \in [1, \infty]$ n'est pas simple en général. Par exemple, si A est la discrétisation de l'opérateur gradient et que $p = 1$, le problème (2.53) devient le problème de Rudin-Osher-Fatemi qui doit pour le moment être résolu de manière itérative et inexacte.

Les fonctions simples commencent à être utilisées abondamment en traitement du signal (voir [Chaux *et al.*, 2007; Chan *et al.*, Accepted for publication in 2008; Dupé *et al.*, 2008] et leurs références), notammment grâce à des résultats théoriques récents dûs à [Combettes et Wajs, 2005; Nesterov, 2007a].

Expliquons maintenant pourquoi ces fonctions présentent un intérêt particulier. Soit f une fonction L-Lipschitz différentiable et Ψ une fonction simple. On a vu précédemment (2.43) que f satisfait :

$$f(x) \leq f(y) + \langle \nabla f(y), x - y \rangle_X + \frac{L}{2} |x - y|_X^2 \tag{2.54}$$

D'où :

$$f(x) + \Psi(x) \leq f(y) + \langle \nabla f(y), x - y \rangle_X + \frac{L}{2} |x - y|_X^2 + \Psi(x) \tag{2.55}$$

Si l'on souhaite résoudre le problème suivant :

$$\min_{x \in X} (f(x) + \Psi(x)) \tag{2.56}$$

une idée naturelle à la vue de (2.55) consiste à appliquer un schéma du type :

$$x^{k+1} = \arg\min_{x \in X} \left(f(x^k) + \langle \nabla f(x^k), x - x^k \rangle_X + \frac{L}{2} |x - x^k|_X^2 + \Psi(x) \right) \tag{2.57}$$

ainsi on s'assure que :

$$f(x^{k+1}) + \Psi(x^{k+1}) \leq f(x^k) + \Psi(x^k) \tag{2.58}$$

La suite $f(x^k) + \Psi(x^k)$ est donc décroissante monotone. L'itération (2.57) peut être réécrite sous la forme :

$$\arg\min_{x \in X} \left(\frac{L}{2} \left(|x - x^k + \frac{\nabla f(x^k)}{L}|_X^2 - |\frac{\nabla f(x^k)}{L}|_X^2 \right) + \Psi(x) \right) \tag{2.59}$$

$$= \arg\min_{x \in X} \left(\frac{1}{2} |x - x^k + \frac{\nabla f(x^k)}{L}|_X^2 + \frac{1}{L} \Psi(x) \right) \tag{2.60}$$

$$= \text{prox}_{\frac{1}{L}\Psi} \left(x^k - \frac{\nabla f(x^k)}{L} \right) \tag{2.61}$$

Si on sait la calculer (i.e. que Ψ est simple), l'itération suivante :

$$x^{k+1} = \text{prox}_{\frac{1}{L}\Psi}\left(x^k - \frac{\nabla f(^k)}{L}\right) \tag{2.62}$$

assure une décroissance monotone de l'énergie. La preuve de convergence de cet algorithme est proposée dans [Combettes et Wajs, 2005]. On donnera aussi son taux de convergence (voir théorème (3.13)).

2.3.4 Fonctions max

Nous présentons ici une classe de fonctions introduites dans [Nesterov, 2005b]. Elles ne semblent pas posséder de nom commun dans la littérature. Dans ce manuscrit, nous les appellerons fonctions "max". Ces fonctions ont la particularité de pouvoir être régularisées. Nous verrons aussi que la minimisation de la somme d'une de ces fonctions avec une fonction fortement convexe, peut être faite de manière très rapide en résolvant un problème adjoint.

Définition 2.14 (Fonction max)**.** *On appelle fonction de type max une fonction* $f : X \to \overline{\mathbb{R}}$ *qui s'écrit sous la forme :*

$$f(x) = \max_{y \in Y}(\langle Ax, y\rangle_Y - \phi(y)) \tag{2.63}$$

où $A : X \to Y$ *est une application linéaire,* ϕ *est une fonction simple.*

Comme ϕ est simple, ce maximum peut être calculé de manière exacte. Dans cette thèse nous utiliserons souvent les normes l^p, $p \in \{1, 2, \infty\}$ de transformées linéaires de l'image. Ces fonctions peuvent s'écrire sous la forme (2.63) en posant $\phi = 0$ et en utilisant la dualité :

$$|Ax|_{l^p(Y)} = \max_{y \in Y, |y|_{l^q(Y)} \leq 1} (\langle Ax, y\rangle_Y) \tag{2.64}$$

Dans cette équation $A : X \to Y$ est une transformée linéaire. q est le conjugué de p $\left(\frac{1}{q} + \frac{1}{p} = 1\right)$. On prêtera une attention particulière au cas où A est l'opérateur discret gradient et où $q \in \{1, 2, \infty\}$. Notons que la représentation sous forme max d'une fonction n'est pas forcément unique. Nous utiliserons essentiellement deux caractéristiques de ces fonctions :

- La première est la possibilité de les régulariser en assurant une erreur d'approximation faible.
- La seconde est la possibilité de définir facilement un problème adjoint. Dans certains cas, on peut trouver des méthodes de minimisation du problème adjoint plus efficaces que celles adaptée à la résolution du problème direct.

Régularisation des fonctions max

Nous allons voir dans ce paragraphe que les fonctions de type (2.64) peuvent être régularisées efficacement. Cette idée est due à Moreau-Yosida [Moreau, 1965; Lemaréchal et Sagastizabal, 1997]. Elle a été adaptée à un cas un peu plus général par Y. Nesterov [Nesterov, 2005b]. L'idée est la suivante :

Lemme 2.8. *Soit* f *une fonction max s'écrivant sous la forme :* $f(x) = \max_{y \in Y}(\langle Ax, y\rangle_Y - \phi(y))$*). Soit* d *:* $dom(\phi) \subset Y \to \mathbb{R}$ *une fonction fortement convexe de paramètre de convexité* $l > 0$ *(une telle fonction*

est souvent appelée fonction prox de l'ensemble dom(ϕ)). Soit $\mu > 0$ un paramétre de régularisation. La fonction :

$$f_\mu(x) = \max_{y \in Y}(\langle Ax, y\rangle_Y - \phi(y) - \mu d(y)) \tag{2.65}$$

est bien définie, elle est convexe et Lipschitz différentiable. Soit y_μ le minimiseur de (2.65). On a alors [3] *:*

$$\nabla f_\mu(x) = A^* y_\mu(x) \tag{2.66}$$

Ce gradient est Lipschitz de paramètre :

$$L_\mu = \frac{\|A\|^2}{\mu l} \tag{2.67}$$

Finalement, si on suppose dom(ϕ) borné, l'approximation f_μ satisfait :

$$f_\mu(x) \leq f(x) \leq f_\mu(x) + \mu D \tag{2.68}$$

avec $D = \max_{y \in dom(\phi)}(d(y))$. f_μ est donc une bonne approximation différentiable de f.

Formulation d'un problème adjoint

Imaginons que l'on souhaite résoudre un problème du type :

$$\min_{x \in X}(f(x) + \Psi(x)) \tag{2.69}$$

Où f est une fonction de type max (2.63) et Ψ est une fonction convexe coercive. Si la fonction $L : (x, y) \mapsto \langle Ax, y\rangle_Y - \phi(y) + \Psi(x)$ satisfait les hypothèses du théorème 2.6, on peut écrire :

$$\begin{aligned}
& \min_{x \in X}(f(x) + \Psi(x)) && (2.70)\\
= \ & \min_{x \in X}(\max_{y \in Y}(\langle Ax, y\rangle_Y - \phi(y)) + \Psi(x)) && (2.71)\\
= \ & \max_{y \in Y}(\min_{x \in X}(\langle Ax, y\rangle_Y + \Psi(x)) - \phi(y)) && (2.72)\\
= \ & \max_{y \in Y}(\min_{x \in X}(\langle x, A^* y\rangle_X + \Psi(x)) - \phi(y)) && (2.73)
\end{aligned}$$

Dans les cas où l'on sait calculer $\min_{x \in K}(\langle x, A^* y\rangle_X + \Psi(x))$ explicitement, on verra que le problème (2.72) peut parfois être résolu plus efficament que le problème (2.70).

[3] c'est une conséquence directe du lemme 2.2

Chapitre 3

Méthodes de premier ordre à un pas

Dans ce chapitre, nous commencons par présenter plusieurs méthodes “classiques” d’optimisation convexe de premier ordre. Ces méthodes ont pour objectif de résoudre le problème suivant :

$$\min_{x \in K} (f(x)) \tag{3.1}$$

où $K \subset X$ est un ensemble convexe non vide et $f : X \to \overline{\mathbb{R}}$ est une fonction convexe propre. Toutes les méthodes que nous allons présenter reposent sur une itération du type :

$$x^{k+1} = F(x^k, f(x^k), \eta^k, K) \tag{3.2}$$

où $\eta^k \in \partial f(x^k)$. Ce sont des méthodes de premier ordre à un pas au sens où elles n’utilisent que l’information apportée par les propriétés liées à x^k pour calculer l’estimée x^{k+1}. Ces méthodes sont très couramment utilisées, mais leurs propriétés de convergence globale et leur taux de convergence semblent assez mal connus. Nous les rappelons donc dans cette partie. Notons que les taux de convergence donnés sont ajustés : si on annonce qu’une méthode a un taux de convergence en $O\left(\frac{1}{k}\right)$, il existe des problèmes pour lesquels la méthode ne peut pas converger plus rapidement que $O\left(\frac{1}{k}\right)$.

Nous avons implémenté et comparé presque toutes ces méthodes durant cette thèse. Nous montrerons des comparaisons pratiques dans le chapitre (7). Nous traitons les cas différentiables, non différentiables, contraints et non contraints séparément. Nous rappelons ensuite quelques résultats de la théorie de la complexité : la rapidité de convergence de tous les schémas de premier ordre est majorée.

3.1 Méthodes à un pas : fonctions différentiables

Dans cette partie, nous considérons le problème suivant :

$$\min_{x \in K} (f(x)) \tag{3.3}$$

où f est convexe, L-Lipschitz différentiable et $K \subset X$ est un ensemble convexe. Notons que ce problème admet un ensemble de solutions convexe qu’on note X^*.

3.1.1 La descente de gradient dans le cas non contraint ($K = X$)

Nous commençons par présenter le cas $K = X$. L'algorithme peut-être "le plus simple" pour résoudre le problème (3.3) est alors le suivant :

$$\begin{cases} x^0 \in X \\ x^{k+1} = x^k - \tau^k \nabla f(x^k) \end{cases} \tag{3.4}$$

Le théorème suivant [Polyak, 1987, p. 21], assure sous quelles conditions il converge :

Théorème 3.8. *Si* $0 < \tau^k < \frac{2}{L}$*, la descente de gradient (3.4) assure* $\lim_{k\to\infty} d(x^k, X^*) = 0$*, où d est la distance d'un point à un ensemble.*

La preuve est simplement basée sur le fait que la suite $f(x^k)$ est une suite monotone décroissante (voir l'équation (2.49)). Un résultat bien moins connu [1] et pourtant intéressant est le suivant [Nesterov, 2004, p. 70] :

Théorème 3.9. *Si* $\tau^k = \frac{1}{L}$*, la descente de gradient à pas constant (3.4) assure :*

$$f(x^k) - f^* \leq \frac{2L|x_0 - x^*|_X^2}{k+4} \tag{3.5}$$

Ce taux de convergence permet de comprendre quels sont les facteurs qui influencent la convergence de l'algorithme :

- **La distance au minimiseur**. La vitesse de convergence en dépend quadratiquement. C'est donc un facteur important.
- **La constante de Lipschitz**. La convergence de l'algorithme en dépend linéairement.

La dépendance de l'objectif au nombre d'itérations est en $O\left(\frac{1}{k}\right)$. Un tel taux de convergence est satisfaisant lors des premières itérations mais il est mauvais asymptotiquement. Sous l'unique hypothèse que f est Lipschitz différentiable, les descentes de gradients permettent donc d'obtenir des solutions approchées assez rapidement, mais elles sont très lentes en général pour obtenir des solutions ayant une bonne précision. Ce mauvais comportement asymptotique indique qu'il faut faire très attention aux choix du point de départ, et si possible transformer le problème pour diminuer sa constante de Lipschitz.

Souvent la constante de Lipschitz L est inconnue et elle peut être difficile à évaluer précisément. Pour obtenir un algorithme convergent, il suffit de choisir τ très petit, mais dans ce cas, on peut montrer que la convergence devient mauvaise, même lors les premières itérations. D'autres stratégies de choix de pas ont donc été élaborées. Aucune de ces stratégies ne permet d'améliorer la convergence en $O(\frac{1}{k})$ mais elles peuvent permettre d'accélerer la convergence dans les premières itérations. Nous les rappelons brièvement ci-dessous :

- **Descente de plus forte pente.** A chaque itération on choisit :

$$\tau^k = \arg\min_{\tau>0} \left(f(x^k - \tau \nabla f(x^k)) \right) \tag{3.6}$$

  Ce problème peut rarement être résolu de façon formelle. Pour le résoudre il faut effectuer une recherche linéaire. Elle est souvent faite par des méthodes dichotomiques.

[1] A notre connaissance, très peu de livres présentent ce résultat si ce n'est [Nesterov, 2004; Iouditski, 2007]

– **Descente de Gradient avec recherche linéaire d'Armijo.** Le pas τ^k est choisi de manière à respecter les conditions suivantes :

$$f(x^k - \tau^k \nabla f(x^k)) \leq f(x^k) - \epsilon \tau^k |\nabla f(x^k)|_X \tag{3.7}$$

$$f(x^k - \eta\tau^k \nabla f(x^k)) \geq f(x^k) - \epsilon \eta\tau^k |\nabla f(x^k)|_X \tag{3.8}$$

où $\epsilon \in]0,1[$ et $\eta > 1$ sont des paramètres de la méthode. Bien sûr si $\nabla f(x^k) = 0$, l'algorithme s'arrête car x^k est un minimiseur de (3.3). Ces conditions peuvent être obtenues de façon itératives en utilisant des méthodes dichotomiques.

– **La stratégie de Barzilai-Borwein.** Une stratégie utilisée récemment pour des problèmes de minimisation de la variation totale [Zhu, 2008] consiste à choisir τ^k de manière à ce que $\frac{1}{\tau^k} I$ ait un comportement similaire à l'inverse de la hessienne $(\nabla^2 f(x^k))^{-1}$. Ainsi la méthode de premier ordre se rapproche d'une méthode de deuxième ordre. Diverses versions sont proposées dans [Barzilai et Borwein, 1988; Zhu, 2008]. Elles mènent à des implémentations du type :

$$\tau^k = \frac{|x^k - x^{k-1}|_X^2}{\langle x^k - x^{k-1}, \nabla f(x^k) - \nabla f(x^{k-1})\rangle_X} \tag{3.9}$$

En pratique ces implémentations sont souvent plus efficaces que des implémentations à pas constant bien que le calcul (3.9) implique un surcoût à chaque itération. A notre connaissance, il n'existe pas d'étude des taux de convergence de telles méthodes.

La descente de gradient peut avoir un taux de convergence bien meilleur dans le cas où la fonction f est fortement convexe de paramètre de convexité l (voir définition 2.11). On obtient dans ce cas un taux de convergence linéaire [Nesterov, 2004], p70 :

Théorème 3.10. *Si f est L-Lipschitz différentiable et l-fortement convexe, alors la descente de gradient (3.4) avec $\tau^k = \frac{2}{l+L}$ assure :*

$$|x^k - x^*|_X \leq \left(\frac{L/l - 1}{L/l + 1}\right)^k |x^0 - x^*|_X \tag{3.10}$$

et :

$$f(x^k) - f^* \leq \frac{L}{2}\left(\frac{L/l - 1}{L/l + 1}\right)^{2k} |x^0 - x^*|_X^2 \tag{3.11}$$

Ce taux de convergence est bien meilleur asymptotiquement que les taux en $O\left(\frac{1}{k}\right)$ précédents. Le terme $L/l \geq 1$ (aussi appelé conditionnement de f) doit être le plus proche de 1 possible pour obtenir une bonne convergence. La méthode de gradient est donc particulièrement efficace pour minimiser des fonctions quadratiques bien conditionnées.

3.1.2 La descente de gradient projeté dans le cas contraint ($K \subset X$)

La descente de gradient projeté est une des premières méthodes inventées pour résoudre le problème d'optimisation contraint (3.3) (elle semble avoir été introduite dans [Polyak, 1987]). Cette méthode est étonnament assez peu appliquée dans le domaine du traitement des images. Le problème de Rudin-Osher-Fatemi régularisé peut être traité de manière très naturelle (et convergente) avec cet algorithme. Il y a eu un regain d'intérêt pour cette méthode récemment (voir [Chambolle, 2004; Zhu, 2008] par

exemple pour des applications à la minimisation de la variation totale). Nous rappelons dans cette partie son principe, ses conditions d'applicabilité ainsi que son taux de convergence.

L'algorithme du gradient projeté s'écrit comme suit :

$$\begin{cases} x^0 \in X \\ x^{k+1} = \Pi_K(x^k - \tau^k \nabla f(x^k)) \end{cases} \tag{3.12}$$

Les conditions de convergence sont exactement les mêmes que celles de la descente de gradient sans projection [Polyak, 1987] page 207 :

Théorème 3.11. *Si $0 < \tau^k < \frac{2}{L}$, la descente de gradient projeté (3.12) assure que :*

- $\lim_{k\to\infty} d(x^k, X^*) = 0$.
- *Si f est fortement convexe alors la convergence de x^k vers x^* est géométrique.*
- *Si f est l-fortement convexe ($l > 0$), L-Lipschitz et que $\tau^k = \tau$, alors on a :*

$$|x^k - x^*|_X \leq cq^k \tag{3.13}$$

avec $q = \max(|1 - \tau l|, |1 - \tau L|)$.

De la même façon que dans le cas non contraint, différentes stratégies de recherche linéaire peuvent être élaborées pour accélérer la convergence. Ces stratégies sont très proches de celles citées en 3.1.1. En pratique il suffit d'arrêter la recherche linéaire si on sort du domaine K. Le lecteur intéressé peut notamment se référer à [Zhu, 2008] et aux références de ce manuscrit pour plus de détails. Un travail récent de Y. Nesterov [Nesterov, 2007a] permet de calculer le taux de convergence de la descente de gradient projeté avec un peu de travail. Nous présentons le résultat ci-dessous :

Théorème 3.12. *Si $\tau^k = \frac{1}{L}$ et si les ensembles de niveau de f sont bornés :*

$$\exists R < \infty, \quad \textbf{\textit{diamètre}}(\{x \in K, f(x) \leq f(x^0)\}) \leq R$$

alors la descente de gradient projeté (3.12) assure que :

$$f(x^k) - f(x^*) \leq \frac{2LR^2}{k+2} \tag{3.14}$$

3.1.3 L'itération proximale pour minimiser la somme d'une fonction différentiable et d'une fonction simple

Récemment, plusieurs auteurs se sont aperçus que la descente de gradient pouvait être adaptée à un cadre plus général que celui présenté dans les deux sections précédentes. Nous nous intéressons ici au problème suivant :

$$\min_{x\in X} (\phi(x) = f(x) + \Psi(x)) \tag{3.15}$$

où $f : X \to \mathbb{R}$ est une fonction L-Lipschitz différentiable et $\Psi : X \to \overline{\mathbb{R}}$ est une fonction simple. Ce cadre est plus général que le précédent puisque la fonction indicatrice d'un ensemble sur lequel on sait faire une projection est simple. Comme on l'a vu dans le paragraphe (2.3.3), lorsque l'on souhaite minimiser la somme d'une fonction simple et d'une fonction Lipschitz, l'algorithme suivant :

$$\begin{cases} x^0 \in X \\ x^{k+1} = \text{prox}_{\frac{1}{L}\Psi}\left(x^k - \frac{\nabla f(x^k)}{L}\right) \end{cases} \tag{3.16}$$

assure que $\phi(x^k)$ est décroissante monotone. On peut en fait montrer que cet algorithme est convergent [Combettes et Wajs, 2005] et évaluer son taux de convergence [Nesterov, 2007a]. Le résultat de convergence est le suivant :

Théorème 3.13. *Supposons que les ensembles de niveau de f sont bornés :*

$$\textbf{\textit{diamètre}}(\{x \in K, f(x) \leq f(x^0)\}) \leq R$$

Dans ce cas, l'algorithme (3.16) assure que :

$$\phi(x^k) - \phi(x^*) \leq \frac{2LR^2}{k+2} \tag{3.17}$$

Si de plus, ϕ est l-fortement convexe, on obtient :

$$\phi(x^k) - \phi(x^*) \leq \left(1 - \frac{l}{4L}\right)^k (\phi(x^0) - \phi(x^*)) \tag{3.18}$$

Cette méthode plus générale que les précédentes permet de minimiser efficacement de nombreuses fonctionnelles de traitement d'images. Nous verrons plus tard qu'elle peut être accélérée en utilisant l'information apportée par les gradients de toutes les itérations.

3.2 Méthodes à un pas : fonctions non-différentiables

On verra dans le chapitre 5 que les problèmes d'optimisation liés au traitement d'images sont souvent non différentiables. Les méthodes présentées précédemment sont donc rarement applicables tel qu'el. De nombreuses méthodes d'optimisation de premier ordre ont été proposées pour résoudre le problème suivant :

$$\min_{x \in K} (f(x)) \tag{3.19}$$

où f est convexe et K est convexe. Nous présentons les plus représentatives ci-dessous.

3.2.1 La descente de sous-gradient projeté

L'une des premières méthodes proposées pour résoudre (3.19) est la descente de sous-gradient projeté. Elle s'écrit :

$$\begin{cases} x^0 \in X \\ x^{k+1} = \Pi_K\left(x^k - \tau^k \frac{\eta^k}{|\eta^k|_X}\right) \end{cases} \tag{3.20}$$

où η^k est n'importe quel élément de $\partial f(x^k)$. Contrairement au cas Lipschitz différentiable, le sous-gradient η^k ne donne pas une approximation polynomiale valable de f en x^k. C'est un hyperplan qui permet uniquement de couper $\mathbb{R}^n$ en deux. Dans une partie $f(x)$ est plus grande que $f(x^k)$. Il faut donc rechercher la solution du problème de minimisation dans l'autre. Les conditions de convergence de l'algorithme (3.20) sont différentes du cas Lipschitz [Polyak, 1987] :

Théorème 3.14. *Si τ^k satisfait $\lim_{k\to\infty} \tau^k = 0$ et $\sum_{k=0}^{\infty} \tau^k = \infty$, alors la méthode du sous-gradient projeté (3.20) assure que :*

$$\lim_{k\to\infty} \min_{0\leq i\leq k}(f(x^i)) = f^*$$

On rappelle que la convergence de l'objectif implique la convergence des sous-suites vers X^* (voir théorème 2.5). Cette méthode est donc convergente. Le choix de τ^k peut cependant être très difficile. De nombreuses suites satisfont les conditions du théorème et un mauvais choix peut entraîner un taux de convergence très lent. Il existe en fait deux choix simples qui - on le verra plus tard - sont optimaux [Nesterov, 2004] :

Théorème 3.15. *Soit f une fonction M-Lipschitz sur une boule $B(x^*, R)$. La méthode (3.20) assure :*

$$\min_{0\leq i\leq k}(f(x^i)) - f^* \leq M\frac{R^2 + \sum_{i=0}^{k}(\tau^i)^2}{2\sum_{i=0}^{k}\tau^i} \tag{3.21}$$

En particulier, une suite optimale est $\tau^i = \frac{r}{\sqrt{i+1}}$ où $r \in \mathbb{R}^+$. Elle assure :

$$M\frac{R^2 + \sum_{i=0}^{k}(\tau^i)^2}{2\sum_{i=0}^{k}\tau^i} \leq \frac{R^2 + r\ln(k+1)}{2r\sqrt{k+1}} \tag{3.22}$$

Bien que cette méthode soit optimale, sa convergence est très lente. En pratique d'autres méthodes simples sont bien meilleures. Son problème principal est qu'elle ne peut pas converger plus rapidement que $\frac{1}{\sqrt{k}}$. On l'imagine facilement en minimisant une fonction de type $|x|_X$ sur $\mathbb{R}^n$. La méthode oscille autour de 0 à une distance qui décroît en $O\left(\frac{1}{\sqrt{k}}\right)$.

Un autre choix de pas optimal qui s'avère souvent bien plus efficace en pratique est le suivant [Polyak, 1987, p. 142] :

Théorème 3.16. *La méthode (3.20) avec :*

$$\tau^k = \frac{f(x^k) - f^*}{|\eta^k|_X} \tag{3.23}$$

assure que :

$$\lim_{k\to\infty} \sqrt{k}(f(x^k) - f^*) = 0 \tag{3.24}$$

Ce choix de pas permet de minimiser la fonction $|x|_X$ en une itération. Il assure une convergence géométrique si f admet un minimum aigu ($f(x) - f^* \geq \alpha|x - x^*|_X$). Peu de fonctions en traitement d'images satisfont cette condition, mais elle indique que cette stratégie de descente peut s'avérer très efficace en pratique.

Généralement, la valeur f^* est inconnue. Plusieurs auteurs proposent de l'évaluer itérativement. Dans [Combettes et Luo, 2002], les auteurs proposent une méthode convergente et montrent qu'elle est efficace en pratique. Dans [Kiwiel, 1996] l'auteur propose plusieurs méthodes optimales en $O\left(\frac{1}{\sqrt{k}}\right)$.

D'autres variantes des descentes de sous-gradient existent. Citons à titre indicatif une méthode multi-pas "Primale-duale" proposée par Y. Nesterov [Nesterov, 2007b]. Cette méthode est elle aussi optimale. L'article est particulièrement intéressant car il présente l'historique des méthodes de premier ordre.

3.2.2 La descente de sous-gradient avec contraintes fonctionnelles

La méthode précédente n'est applicable que si la contrainte K est simple (i.e. que l'on peut effectuer une projection sur K). De nombreux problèmes ne satisfont pas cette hypothèse. Nous présentons ci-dessous un dernier schéma qui permet de résoudre à peu près tous les problèmes convexes de traitement d'images que l'on peut imaginer. Le fait qu'il fonctionne aussi généralement indique cependant qu'il ne peut pas être très efficace en pratique.

Dans le majorité des cas, les contraintes sont exprimées de façon fonctionnelle. On cherche à minimiser f sur l'ensemble $K \cap Q$ où K est simple et Q s'écrit sous la forme :

$$Q = \{x \in \mathbb{R}^n, \bar{f}(x) \leq 0\} \tag{3.25}$$

avec :

$$\bar{f}(x) = \max_{i=1..m} f_i(x) \tag{3.26}$$

Dans (3.26), chaque fonction f_i est convexe. On suppose de plus que K est borné de diamètre R. On considère le schéma suivant :

1. Choisir $x^0 \in K$ et définir $h^k = \frac{R}{\sqrt{k+0.5}}$.
2. A l'itération k :
   - Calculer $\underline{f}(x^k)$ et $\eta^k \in \partial \underline{f}(x^k)$.
   - Calculer $\bar{f}(x^k)$ et $\bar{\eta}^k \in \partial \bar{f}(x^k)$.
   - Poser :
   $$p^k = \begin{cases} \eta^k, & \text{si } \underline{f}(x^k) < h^k |\bar{\eta}^k|_X \\ \bar{\eta}^k, & \text{si } \bar{f}(x^k) \geq h^k |\bar{\eta}^k|_X \end{cases}$$
   - Poser $x^{k+1} = \Pi_K(x^k - h^k \frac{p^k}{|p^k|_X})$

L'idée sous-jacente à ce schéma est de se déplacer tantôt dans une direction qui permet de minimiser f tantôt dans une direction qui permet de se rapprocher de l'ensemble des contraintes suivant la distance à cet ensemble. On peut montrer [Nesterov, 2004, p. 144] :

Théorème 3.17. *Supposons que f est M-Lipschitz continue sur une boule $B(x^*, R)$, et que la norme des sous-gradients est bornée sur cette boule :*

$$M' = \max_{\{1 \leq i \leq m,\ x \in B(x^*,R)\}} (|\bar{\eta}|_X, \bar{\eta} \in \partial f_i(x)) \leq \infty$$

Dans ce cas, le schéma précédent assure qu'il existe $i \leq k$ tel que :

$$f(x^i) - f^* \leq \frac{\sqrt{3}MR}{\sqrt{k - 1.5}} \quad \text{et} \quad \bar{f}(x^i) \leq \frac{\sqrt{3}M'R}{\sqrt{k - 1.5}} \tag{3.27}$$

Ce schéma simple qui permet de résoudre la plupart des problèmes d'optimisation convexe a donc un taux de convergence en $O\left(\frac{1}{\sqrt{k}}\right)$. On verra plus tard qu'il est optimal. D'autres approches ont été proposées pour résoudre ce genre de problèmes (voir par exemple [Combettes et Pesquet, 2004] et les références de cet article). Ce dernier théorème montre que sous la seule hypothèse qu'on soit capable de calculer les valeurs des fonctions et des sous-gradients en temps polynomial, trouver une solution de précision ϵ à un problème d'optimisation convexe peut être réalisé en temps polynomial. C'est un résultat central de la théorie de la complexité en optimisation convexe.

3.2.3 Autres méthodes

L'optimisation convexe non-différentiable a une importance cruciale et il existe de nombreuses méthodes de premier ordre concurrentes au sous-gradient projeté. Citons à titre indicatif les méthodes des plans coupant et les méthodes d'approximation linéaires de f.

Pour les plans coupant, à chaque itération on calcule un sous-gradient. Ce sous-gradient définit un hyperplan qui coupe X en deux parties. On sait que le minimum se trouve dans la partie opposée à la direction du sous-gradient. A chaque itérations, on peut affiner un polytope qui définit l'espace de recherche.

Dans les méthodes d'approximation linéaires, on construit une approximation linéaire par morceaux de f. A chaque itération, on calcule un sous-gradient. Celui-ci permet de définir un hyperplan minorant f. A chaque itérations, on peut donc affiner une minoration convexe, linéaire par morceaux de la fonction f. On peut minimiser cette enveloppe plutôt que f. C'est la méthode de Kelley [Kelley, 1960]. Une autre solution plus stable est la méthode des niveaux [Nesterov, 2004]. Ces méthodes sont cependant difficiles à implémenter et chaque itération peut être très coûteuse en temps de calcul. Ces méthodes ne semblent donc pas adpatées pour le moment à l'optimisation en très grande dimension.

3.3 Rappels sur la théorie de la complexité

On a vu dans la section précédente quelques méthodes de minimisation couramment utilisées en optimisation convexe. On peut se demander s'il n'en existe pas de meilleures (i.e. qui convergent plus rapidement). La théorie de la complexité permet de répondre à cette question. Nous rappelons ici brièvement les principaux résultats de cette théorie.

3.3.1 Cas Lipschitz différentiable

Nous considérons ici les problèmes s'écrivant de la façon suivante :

$$\min_{x \in X} (f(x)) \tag{3.28}$$

où f est convexe, L-Lipschitz différentiable. On note cette classe de problèmes $\mathcal{P}$. On rappelle que $X = \mathbb{R}^n$.

On se concentre sur une classe de méthodes de minimisation $\mathcal{M}$. Une méthode de minimisation m appartient à $\mathcal{M}$ si elle génère une suite $\{x^k\}$ satisfaisant la condition suivante :

$$x^{k+1} \in x^0 + Lin\left(\{\nabla f(x^0), \nabla f(x^1), ..., \nabla f(x^k)\}\right)$$

On peut montrer le résultat suivant [Nesterov, 2004, p. 61] :

Théorème 3.18. *Pour tout $k \leq \frac{n-1}{2}$ et tout $x^0 \in X$, il existe une fonction $f \in C^\infty(X)$, L-Lipschitz différentiable telle que pour toute méthode $m \in \mathcal{M}$, on ait :*

$$f(x^k) - f^* \geq \frac{3L|x^0 - x^*|_X^2}{32(k+1)^2} \tag{3.29}$$

et :

$$|x^k - x^*|_X^2 \geq \frac{1}{8}|x^0 - x^*|^2 \tag{3.30}$$

Ce résultat est lourd de conséquences. Il indique qu'on ne peut pas obtenir de taux de convergence satisfaisant sur la distance au minimiseur en faisant uniquement une hypothèse de différentiabilité sur la fonction à minimiser. De plus il indique qu'aucune méthode de premier ordre ne peut dépasser un taux de convergence en $O\left(\frac{1}{k^2}\right)$ uniformément sur tous les problèmes convexes différentiables.

Rappelons aussi que la descente de gradient - avec toutes ses stratégies de recherche linéaire - n'avait un taux de convergence qu'en $O\left(\frac{1}{k}\right)$. Ce dernier théorème semble donc indiquer qu'on peut peut-être trouver des méthodes plus efficaces. On présentera dans la prochaine section une méthode "optimale" trouvée récemment par Y. Nesterov [Nesterov, 2007a].

3.3.2 Cas non différentiable

Dans cette partie nous considérons le problème suivant :

$$\min_{x \in X} (f(x)) \tag{3.31}$$

où f est convexe, M-Lipschitz continue sur une boule $B(x^*, R)$. On note cette classe de problèmes $\mathcal{P}'$.

On se concentre sur une classe de méthode de minimisation $\mathcal{M}'$. Une méthode de minimisation m' appartient à $\mathcal{M}'$ si elle peut s'écrire de la façon suivante :

$$x^{k+1} \in x^0 + Lin\left(\{\eta^0, \eta^1, ..., \eta^k)\}\right)$$

où $\forall i,\ \eta^i \in \partial f(x^i)$. On peut montrer le résultat suivant [Nesterov, 2004], p138 :

Théorème 3.19. *Pour tout $k \leq n-1$, il existe une fonction f convexe M-Lipschitz sur $B(x^*, R)$ telle que pour toute méthode $m' \in \mathcal{M}'$, on ait :*

$$f(x^k) - f^* \geq \frac{MR}{2(1+\sqrt{k+1})} \tag{3.32}$$

Aucune méthode de premier ordre reposant sur les sous-gradients ne peut donc dépasser un taux de convergence en $O\left(\frac{1}{\sqrt{k}}\right)$. Les méthodes de descente de sous-gradient présentées dans la partie précédentes sont donc optimales. Ces méthodes - bien qu'optimales - peuvent toutefois être très lentes. Imaginons par exemple que $M = R = 1$ et que l'on souhaite trouver une solution approchée x^k qui satisfasse $f(x^k) - f^* \leq 10^{-3}$. Avec une méthode de sous-gradient, on aura besoin - au pire des cas - de l'ordre de 10^6 itérations ! La borne de complexité obtenue pour des problèmes différentiables était bien plus encourageante. Avec un taux de convergence en $\frac{1}{k^2}$, on n'a besoin que de 32 itérations pour obtenir cette solution !

3.4 Conclusion

Les méthodes d'optimisation que nous avons présentées permettent de résoudre presque tous les problèmes d'optimisation convexe qui existent en traitement d'image. Ceci pourrait laisser penser que les problèmes d'optimisation convexe sont simples à résoudre. C'est faux lorsqu'on parle de problèmes d'optimisation en très grande dimension. Il est absurde de réaliser plus de 10000 itérations d'une descente de sous-gradient avec pas décroissant sur une image de taille 1000×1000. Or notre expérience nous montre que c'est approximativement ce qu'il faut pour obtenir une solution qui approche correctement (au sens visuel) les propriétés de la solution exacte. Les descentes de gradient

se comportent très bien sur des problèmes différentiables bien conditionnés, cependant plusieurs centaines d'itérations peuvent être nécessaires pour obtenir une approximation satisfaisante. Ce nombre est encore trop élevé pour penser à des applications en temps réel.

Il nous semble donc nécessaire de chercher de nouvelles techniques plus efficaces d'optimisation. Dans le chapitre suivant, nous présentons plusieurs algorithmes de minimisation de premier ordre et montrons que leurs taux de convergence sont meilleurs d'un ordre de grandeur que ceux présentés ici.

Chapitre 4

Algorithmes de minimisation structurels multi-pas

Dans le chapitre précédent, on a vu que les taux de convergence des schémas de premier ordre étaient majorés. Ce résultat est à interpréter avec précaution. Imaginons par exemple que l'on souhaite résoudre le problème suivant :

$$\min_{x\in X}\left(f(x)=\frac{1}{2}|Ax-b|_Y^2+|x|_{l^1(X)}\right) \tag{4.1}$$

où $A : X \to Y$ est une transformée linéaire et $b \in Y$ une donnée à traiter. Ce problème a récemment pris beaucoup d'importance en traitement du signal [Chen *et al.*, 1998; Tropp, 2006] et en statistiques [Miller, 2002]. Dans le cas où A est une matrice aléatoire il devient un problème d'échantillonnage compressif (compressed sensing) [Candès, 2006].

Une analyse rapide de ce problème indique qu'il est convexe non-différentiable. Si on utilise uniquement cette information et les sous-gradients de l'énergie, on ne peut pas espérer pouvoir montrer un taux de convergence supérieur à $O\left(\frac{1}{\sqrt{k}}\right)$. Une analyse plus fine de l'energie indique qu'elle est composée de la somme d'une fonction Lipschitz différentiable $\left(\frac{1}{2}|Ax-b|_Y^2\right)$ et d'une fonction simple ($|x|_{l^1(X)}$). On peut donc obtenir un taux de convergence bien meilleur en $O\left(\frac{1}{k}\right)$ en utilisant l'algorithme (3.16). On va voir que de récents développements permettent même d'obtenir des taux de convergence en $O\left(\frac{1}{k^2}\right)$.

Les bornes de complexité que l'on a indiquées dans le chapitre précédent ne sont donc vraies que si on dispose d'une information limitée sur les fonctions à minimiser telles que leurs valeurs et leurs gradients ou sous-gradients. Il est possible de dépasser ces bornes de complexité sur des classes de problèmes plus restreintes pour lesquelles on peut extraire plus de propriétés des fonctions.

Dans la suite de ce chapitre, nous allons présenter plusieurs classes de problèmes non-différentiables pour lesquelles on peut trouver des algorithmes qui dépassent les bornes de complexité. On appelle ces méthodes "algorithmes de minimisation structurels" car elles exploitent la structure des fonctions en jeu. Les algorithmes que nous allons présenter sont multi-pas dans le sens où à l'itération k la direction de descente n'est pas calculée uniquement à partir de $\nabla f(x^k)$ mais à partir de $Lin(\{\nabla f(x^0), ..., \nabla f(x^k)\})$.

Toutes les classes que l'on présente sont composées de sommes pondérées des fonctions listées dans le paragraphe 2.3.

4.1 Somme d'une fonction simple et d'une fonction Lipschitz

On a vu dans le chapitre précédent que les méthodes de premier ordre ne pouvaient pas avoir un taux de convergence plus rapide que $O\left(\frac{1}{k^2}\right)$ uniformément sur tous les problèmes convexes différentiables. Les méthodes que l'on a présenté dans le paragraphe 3.1 avaient toutes un taux de convergence de l'ordre de $O\left(\frac{1}{k}\right)$. Yurii Nesterov a essayé de combler cette différence et a trouvé plusieurs schémas de minimisation en $O\left(\frac{1}{k^2}\right)$ présentés dans [Nesterov, 1983; Nesterov, 2005b; Nesterov, 2005a; Nesterov, 2007a]. Ces schémas sont tous multi-pas au sens où la direction de descente n'est pas calculée seulement à partir du gradient à l'itération courante, mais à partir du sous-espace linéaire engendré par les gradients calculés aux itérations précédentes. Cette idée est similaire aux techniques de gradient conjugué mais dans un cadre non quadratique. Notons [Polyak, 1987], p.171 que le gradient conjugué (méthode optimale pour minimiser des fonctions convexes quadratiques) a un taux de convergence en $O\left(\frac{1}{k^2}\right)$. Ce taux est identique à celui obtenu par Y. Nesterov, mais dans un cadre bien plus limité. Il nous semble donc étonnant que ces méthodes ne soient pas utilisées bien plus largement en calcul numérique. En effet, très peu de travaux utilisent ces résultats pour le moment, et à notre connaissance peu d'auteurs essaient d'améliorer les résultats de Y. Nesterov.

4.1.1 Le principe des schémas multi-pas

Dans ce paragraphe, nous présentons brièvement les schémas optimaux de Y. Nesterov. Nous utilisons la présentation la plus récente de ces travaux [Nesterov, 2007a]. On s'intéresse au problème suivant :

$$\min_{x\in X}\left(\phi(x) = f(x) + \Psi(x)\right) \tag{4.2}$$

où f est L-Lipschitz différentiable, Ψ est une fonction simple, l-fortement convexe ($l \geq 0$). Dans le schéma [Nesterov, 2007a], trois objets sont actualisés à chaque itération :

- Une suite minimisante $\{x^k\}$.
- Une suite croissante de coefficients $\{A^k\}$ construite de la façon suivante :

$$A^0 = 0,\ A^{k+1} = A^k + a^{k+1},\ \text{ avec } a^{k+1} > 0 \tag{4.3}$$

- Une suite de fonctions $\{\psi^k\}$ approchant $A^k\phi$:

$$\psi^k(x) = g^k(x) + A^k\Psi(x) + \frac{1}{2}|x - x^0|_X^2 \tag{4.4}$$

  où $x^0 \in dom(\Psi)$ est le point de départ de la méthode et les fonctions g^k sont des fonctions linéaires construites à partir des gradients $\nabla f(x^k)$.

L'astuce de la méthode est de maintenir récursivement les deux relations suivantes :

$$\left\{\begin{array}{lcl} A^k\phi(x^k) & \leq & \min_{x\in X}\left(\psi^k(x)\right) \\ \psi^k(x) & \leq & A^k\phi(x) + \frac{1}{2}|x - x^0|_X^2,\ \forall x \in X \end{array}\right. \tag{4.5}$$

Ceci assure clairement :

$$\phi(x^k) - \phi(x^*) \leq \frac{|x^* - x^0|}{2A^k} \tag{4.6}$$

Il faut donc trouver le moyen de faire augmenter la suite A^k le plus rapidement possible en maintenant les relations (4.5). Y. Nesterov [Nesterov, 2007a] présente une stratégie qui peut s'écrire sous la forme algorithmique suivante [1] :

Algorithme 4.1: Descente de gradient accéléré

Entrée : Le nombre d'itérations N, un point de départ $x^0 \in dom(\Psi)$.

Sortie : x^N une estimée de x^*.

début

Poser $\psi^0(x) = |x - x^0|_X^2$

Poser A^0=0

pour *k allant de* 0 *à N* **faire**

Calculer a^{k+1}, solution de l'équation du second ordre $\frac{(a^{k+1})^2}{A^k + a^{k+1}} = 2\frac{1 + lA^k}{L}$

Poser $\psi^k(x) = \underbrace{\sum_{i=1}^{k} a^i \left(f(x^i) + \langle \nabla f(x^i), x - x^i \rangle_X \right)}_{g^k(x)} + A^k \Psi(x) + \frac{1}{2}|x - x^0|_X^2$

Calculer $v^k = \arg\min_{x \in X}(\psi^k(x))$

Poser $y^k = \frac{A^k x^k + a^{k+1} v^k}{A^k + a^{k+1}}$

Poser $x^{k+1} = \arg\min_{x \in X} \left(f(y^k) + \langle \nabla f(y^k), x - y^k \rangle_X + \frac{L}{2}|x - y^k|_X^2 + \Psi(x) \right)$

Poser $A^{k+1} = A^k + a^{k+1}$

fin

fin

Le résultat central de Y. Nesterov est le suivant :

Théorème 4.20. *L'algorithme (4.1) assure que* $A^k \geq \frac{k^2}{2L}$ *et donc :*

$$\phi(x^k) - \phi(x^*) \leq \frac{L|x^* - x^0|_X^2}{k^2} \tag{4.7}$$

Si $l > 0$ *(i.e.* Ψ *est fortement convexe) alors il assure :*

$$A^k \geq \max\left(\frac{k^2}{2L}, \frac{1}{L}\left(1 + \sqrt{\frac{l}{2L}}\right)^{2(k-1)} \right) \tag{4.8}$$

D'où :

$$\phi(x^k) - \phi(x^*) \leq \frac{L}{2}|x^* - x^0|_X^2 \left(1 + \sqrt{\frac{l}{2L}}\right)^{-2(k-1)} \tag{4.9}$$

Pour peu que les calculs des arg min soient calculables, ce schéma présente donc un taux de convergence optimal en $O\left(\frac{1}{k^2}\right)$. Ce taux devient linéaire dans le cas fortement convexe.

[1] Nous avons simplifié volontairement le schéma de Y. Nesterov en supposant la constante de Lispchitz de f connue. Son schéma permet de l'estimer itérativement.

4.1.2 Comparaison avec les méthodes à un pas

Comparons maintenant les taux de convergence de cette méthode avec l'algorithme à un pas (3.16) :

- Dans le cas non fortement convexe ($l = 0$), l'algorithme à un pas (3.16) assurait $\phi(x^k) - \phi(x^*) \leq \frac{2LR^2}{k+2}$. Le taux de convergence (4.7) de l'algorithme multi-pas est donc supérieur d'un ordre de grandeur. Si l'on souhaite obtenir une solution de précision 10^{-3}, pour un problème où $R = |x^0 - x^*|_X = L = 1$, il faudra au pire des cas 32 itérations au schéma accéléré de Nesterov et 2000 itérations au schéma classique.
- Dans le cas fortement convexe, l'algorithme à un pas (3.16) assurait :

$$\phi(x^k) - \phi(x^*) \leq \left(1 - \frac{l}{4L}\right)^k (\phi(x^0) - \phi(x^*)) \tag{4.10}$$

Notons $a = \frac{l}{L}$. Dans ce cas, la borne supérieure de l'erreur (4.10) est proportionnelle à :

$$\left(1 + \frac{a}{4-a}\right)^{-k} \tag{4.11}$$

De la même façon la borne supérieure du taux de convergence (4.9) de la méthode multi-pas est proportionnelle à :

$$\left(1 + (\sqrt{2} + .5)a\right)^{-k} \tag{4.12}$$

En considérant que $a \in [0, 1]$ [2], on a $\frac{1}{4-a} \in \left[\frac{1}{4}, \frac{1}{3}\right]$. Dans tous les cas, cette constante est bien inférieure à $\sqrt{2} + .5 \sim 1.9$.

On voit donc que le schéma de minimisation accéléré a une efficacité de loin supérieure au schéma à un pas plus classique. En traitement d'images, ceci se traduit par des temps de calculs divisés par des facteurs 100 pour obtenir des solutions visuellement satisfaisantes à certains problèmes d'optimisation mal conditionnés (voir par exemple la figure 5.8 ou le chapitre 7).

4.1.3 Réécriture de l'algorithme avec des fonctions prox

L'algorithme (4.1) est difficile à implémenter tel quel. Pour l'utiliser en pratique, on peut le réécrire (avec un peu de travail) sous la forme suivante :

[2] L'inverse du "conditionnement" a peut être supérieur à 1 dans les rares cas où le paramètre de forte convexité de Ψ est plus grand que la constante de Lispchitz de f. Nous n'avons pas rencontré de tels problèmes pendant cette thèse.

Algorithme 4.2: Descente de gradient accélérée version implémentable

Entrée : Le nombre d'itérations N, un point de départ $x^0 \in dom(\Psi)$.

Sortie : x^N une estimée de x^*.

début

Poser $A = 0$

Poser $g = 0$

Poser $x = x^0$

pour *k allant de* 0 *à* N **faire**

$t = 2\dfrac{1 + lA}{L}$.

$a = \dfrac{t + \sqrt{t^2 + 4tA}}{2}$.

$v = \text{prox}_{A\Psi}\left(x^0 - g\right)$

$y = \dfrac{Ax + av}{A + a}$

$x = \text{prox}_{\frac{1}{L}\Psi}\left(y - \dfrac{\nabla f(y)}{L}\right)$

$g \leftarrow g + a\nabla f(x)$

$A \leftarrow A + a$

fin

fin

Pour utiliser cet algorithme, il suffit de savoir calculer l'opérateur $\text{prox}_{\alpha\Psi}$ pour tout $\alpha \in \mathbb{R}^+$. C'est le cas - par définition - si Ψ est simple. En terme de complexité algorithmique, chaque itération de ce schéma coûte deux appels à la fonction prox et deux calculs de ∇f. Chaque itération de ce schéma coûte donc deux fois celle de l'algorithme à un pas (3.16). En pratique, ce surcoût est généralement largement compensé par le gain en efficacité de la méthode.

4.1.4 Réécriture de l'algorithme avec des projecteurs

Lorsque la fonction Ψ est la fonction indicatrice d'un ensemble convexe K, l'algorithme peut encore être réécrit sous une forme qu'on utilisera souvent dans les expériences numériques [Weiss *et al.*, 2007a].

Algorithme 4.3: Descente de gradient accélérée dans le cas contraint

Entrée : Le nombre d'itérations N, un point de départ $x^0 \in dom(\Psi)$.

Sortie : x^N une estimée de x^*.

début

Poser $k = -1$

Poser $g^{(-1)} = 0$

Poser $x^0 \in K$ (une estimée précise de x^* si possible)

Poser $L =$ constante de Lipschitz de ∇f

pour *k allant de* 0 *à* $N-1$ **faire**

Calculer $\eta^k = \nabla f(x^k)$

Calculer $y^k = \Pi_K\left(x^k - \frac{\eta^k}{L}\right)$

Poser $g^k = g^{k-1} + \frac{k+1}{2}\eta^k$

Calculer $v^k = \Pi_K\left(x^0 - \frac{g^k}{L}\right)$

Poser $x^{k+1} = \frac{2}{k+3}v^k + \frac{k+1}{k+2}y^k$

fin

fin

Notons que dans ce schéma, on ne calcule ∇f qu'une seule fois par itération. Ce schéma peut donc diviser par deux les temps de calcul de l'algorithme 4.2.

Dans la suite de ce chapitre, nous allons voir que ces schémas d'optimisation permettent de résoudre efficacement d'autres problèmes que ceux composés de la somme d'une fonction simple et d'une fonction Lipschitz.

4.2 Somme d'une fonction max et d'une fonction fortement convexe

La méthode que nous présentons dans ce chapitre a été trouvée pendant la thèse. C'est en analysant l'algorithme de A. Chambolle [Chambolle, 2004] que nous nous sommes aperçus qu'il pouvait être généralisé à une classe de problèmes bien plus large. L'algorithme que nous proposons a une efficacité très nettement supérieure à l'algorithme initial [Chambolle, 2004]. Nous donnons son taux de convergence en norme l^2 et en termes d'objectif, ce qui était jusqu'à présent une question ouverte.

4.2.1 Idée de la méthode sur un exemple

Expliquons brièvement le principe de la technique sur un exemple simple. Dans [Chambolle, 2004], l'auteur propose un schéma efficace et convergent pour résoudre le problème de Rudin-Osher-Fatemi :

$$\min_{y \in Y}\left(|\nabla y|_{l^1(X)} + \lambda |y - y_0|_Y^2\right) \tag{4.13}$$

où $\nabla : Y \to X$ est la discrétisation de l'opérateur gradient (voir annexe A.1), y_0 est une donnée à débruiter et $\lambda > 0$ est un paramètre du modèle. Exprimons le problème dual à (4.13). On note $K = \{x \in X, |x|_{l^\infty(X)} \leq 1\}$. On a :

$$\min_{y\in Y}\left(|\nabla y|_{l^1(X)} + \lambda|y-y_0|_Y^2\right) \tag{4.14}$$

$$= \min_{y\in Y}\left(\max_{x\in K}(\langle \nabla y, x\rangle_X) + \lambda|y-y_0|_Y^2\right) \tag{4.15}$$

$$= \min_{y\in Y}\left(\max_{x\in K}\left(\langle y, -\operatorname{div} x\rangle_Y + \lambda|y-y_0|_Y^2\right)\right) \tag{4.16}$$

$$= \max_{x\in K}\left(\min_{y\in Y}\left(\langle y, -\operatorname{div} x\rangle_Y + \lambda|y-y_0|_Y^2\right)\right) \tag{4.17}$$

$$= \max_{x\in K}\left(-\frac{|\operatorname{div} x|_Y^2}{4\lambda} + \langle y_0, -\operatorname{div} x\rangle_Y\right) \tag{4.18}$$

A, Chambolle montre que la solution du problème dual (4.18) permet de déterminer la solution du problème primal (4.13). L'intérêt du problème dual est qu'il est Lipschitz différentiable alors que le problème primal est non différentiable. On peut donc le résoudre efficacement avec des méthodes de type descente de gradient projeté [3]. Dans la suite nous présentons une généralisation de cette méthode.

4.2.2 L'algorithme et son taux de convergence

On s'intéresse ici aux problèmes du type suivant :

$$\min_{y\in Y}(f(y)+\varphi(y)) \tag{4.19}$$

où f est une fonction max et φ est une fonction l-fortement convexe. Par définition, f peut s'écrire sous la forme :

$$f(y) = \max_{x\in X}(\langle By, x\rangle_X - \Psi(x)) \tag{4.20}$$

où $B : Y \to X$ est une fonction linéaire et $\Psi : X \to \overline{\mathbb{R}}$ est une fonction simple μ-fortement convexe ($\mu \geq 0$). On suppose de plus que le domaine de Ψ est borné [4]. Le problème (4.19) peut être réécrit :

$$\min_{y\in Y}(f(y)+\varphi(y)) \tag{4.21}$$

$$= \min_{y\in Y}\left(\max_{x\in X}(\langle By, x\rangle_X - \Psi(x)) + \varphi(y)\right) \tag{4.22}$$

$$= \max_{x\in X}\left(\underbrace{\min_{y\in Y}(\langle By, x\rangle_X + \varphi(y))}_{\hat{f}(x)} - \Psi(x)\right) \tag{4.23}$$

$$= \max_{x\in X}\left(\hat{f}(x) - \Psi(x)\right) \tag{4.24}$$

$$= -\min_{x\in X}\left(-\hat{f}(x) + \Psi(x)\right) \tag{4.25}$$

[3] dans [Chambolle, 2004] l'auteur propose une méthode originale, mais lente par rapport aux développements récents.

[4] Cette hypothèse peut être remplacée par des hypothèses de croissance plus que linéaire à l'infini de Ψ. En pratique, nous n'aurons pas besoin de l'affiner.

D'après le lemme 2.8, la fonction $\hat{f}$ est concave, Lipschitz différentiable de constante de Lipschitz $L = \frac{\|B\|^2}{l}$.

Le problème (4.25) est appelé problème dual ou adjoint à (4.19). Ce problème est la somme d'une fonction simple et d'une fonction Lipschitz différentiable. Il peut donc être minimisé efficacement par l'algorithme multi-pas 4.2. Analysons maintenant son taux de convergence. Notons $\phi(y) = f(y) + \varphi(y)$ et $\hat{\phi}(x) = -\hat{f}(x) + \Psi(x)$. Notre résultat principal est le suivant :

Théorème 4.21. *Soit l'algorithme suivant :*

Algorithme 4.4: Descente de gradient dual accélérée

Entrée : Le nombre d'itérations N, un point de départ $x^0 \in dom(\Psi)$.

Sortie : $\bar{y}^N$ une estimée de y^*.

début

Poser $\psi^0(x) = |x - x^0|_X^2$

Poser A^0=0

Poser $L = \dfrac{\|B\|^2}{l}$

pour k *allant de* 0 *à* $N-1$ **faire**

Calculer a^{k+1}, solution de l'équation du second ordre $\dfrac{(a^{k+1})^2}{A^k + a^{k+1}} = 2\dfrac{1+\mu A^k}{L}$

$v^k = \arg\min_{x \in X}(\psi^k(x))$

$z^k = \dfrac{A^k x^k + a^{k+1} v^k}{A^k + a^{k+1}}$

$x^{k+1} = \arg\min_{x \in X}\left(-\hat{f}(z^k) - \langle \nabla \hat{f}(z^k), x - z^k\rangle_X + \dfrac{L}{2}|x - z^k|_X^2 + \Psi(x)\right)$

$y^{k+1} = \arg\min_{y \in Y}\left(\langle y, B^* x^k\rangle_Y + \varphi(y)\right)$

$\psi^{k+1}(x) = \psi^k(x) + a^{k+1}\left(-\hat{f}(x^{k+1}) - \langle \nabla \hat{f}(x^{k+1}), x - x^{k+1}\rangle_X) + \Psi(x)\right)$

$A^{k+1} = A^k + a^{k+1}$

fin

Poser $\bar{y}^N = \dfrac{1}{A^N}\sum_{i=1}^{N} a^i y^i$

fin

Il assure que :

$$\phi(\bar{y}^N) - \phi(y^*) \leq \frac{D}{A^N} \tag{4.26}$$

avec $D = \max_{x \in dom(\Psi)}\left(|x - x^0|_X^2\right)$ *et* $A^N \geq \max\left(\frac{lN^2}{\|B\|^2}, \frac{\mu}{\|B\|^2}\left(1 + \sqrt{\frac{\mu l}{2\|B\|^2}}\right)^{2(k-1)}\right)$. *De plus on a :*

$$|\bar{y}^N - y^*|_Y^2 \leq \frac{2}{l}\frac{D}{A^N} \tag{4.27}$$

On obtient donc un taux de convergence en $O\left(\frac{1}{k^2}\right)$ pour une classe assez large de fonctions non-différentiables. Le gain par rapport aux méthodes de sous-gradient en $O\left(\frac{1}{\sqrt{k}}\right)$ est très grand. Notons de

plus qu'on obtient un taux de convergence en norme ce qui est inhabituel (voir théorème 3.18 équation (3.30)).

Démonstration. On a vu précédemment que :

$$\min_{y\in Y}(\phi(y)) = \max_{x\in X}(-\hat{\phi}(x)) \tag{4.28}$$

Soit y^* et x^* la solution des problèmes primal (gauche) et dual (droite) respectivement. On a $\phi(y^*) = -\hat{\phi}(x^*)$ et :

$$\phi(y) \geq -\hat{\phi}(x) \quad \forall (x,y) \in X \times Y \tag{4.29}$$

Le schéma (4.2) appliqué au problème suivant :

$$\min_{x\in X}\left(-\hat{f}(x) + \Psi(x)\right) \tag{4.30}$$

génère une suite de points $\{x^k\}$ qui satisfait :

$$\begin{aligned} A^k\hat{\phi}(x^k) &\leq \min_{x\in X}\left(\psi^k(x)\right) && (4.31)\\ &\leq \min_{x\in X}\left(\frac{1}{2}|x-x^0|_X^2 + A^k\Psi(x) - \sum_{i=1}^{k} a^i[\hat{f}(x^i) + \langle\nabla\hat{f}(x^i), x-x^i\rangle_X]\right) && (4.32)\end{aligned}$$

Notons $y^k = \arg\min_{y\in Y}\left(\langle y, B^*x^k\rangle_Y + \varphi(y)\right)$. Ce minimium est unique car φ est fortement convexe. On a $\hat{f}(x^k) = \langle y^k, B^*x^k\rangle_Y + \varphi(y^k)$. D'après le lemme (2.2), $\hat{f}$ est dérivable et sa dérivée est $\nabla\hat{f}(x^k) = By^k$. On a donc :

$$\hat{f}(x^i) + \langle\nabla\hat{f}(x^i), x-x^i\rangle_X = \varphi(y^i) + \langle y^i, B^*x\rangle_Y \tag{4.33}$$

Notons :

$$\bar{y}^k = \frac{1}{A^k}\sum_{i=1}^{k} a^i y^i \tag{4.34}$$

On obtient alors :

$$\min_{x\in X}\left(\psi^k(x)\right) \tag{4.35}$$

$$\overset{(4.32)}{\leq} \quad \min_{x\in X}\left(\frac{1}{2}|x-x^0|_X^2 + A^k\Psi(x) - \sum_{i=1}^{k} a^i[\varphi(y^i) + \langle y^i, B^*x\rangle_Y]\right) \tag{4.36}$$

$$\overset{(4.34)+\varphi \text{ convexe}}{\leq} \quad \min_{x\in X}\left(\frac{1}{2}|x-x^0|_X^2 + A^k\Psi(x) - A^k\varphi(\vec{y}^k) - A^k\langle \vec{y}^k, B^*x\rangle_Y\right) \tag{4.37}$$

$$\leq \quad -A^k\varphi(\vec{y}^k) - A^k\max_{x\in X}\left(\langle B\vec{y}^k, x\rangle_Y - \Psi(x) - \frac{1}{2A^k}|x-x^0|_X^2\right) \tag{4.38}$$

$$\overset{(4.20)}{\leq} \quad -A^k\varphi(\vec{y}^k) - A^k f(\vec{y}^k) + \frac{1}{2}|\vec{x}^k - x^0|_X^2 \tag{4.39}$$

$$\leq \quad -A^k\phi(\vec{y}^k) + \frac{1}{2}|\vec{x}^k - x^0|_X^2 \tag{4.40}$$

A la ligne (4.39), $\vec{x}^k$ peut être n'importe quel élément de $\arg\max_{x\in X}\left(\langle B\vec{y}^k, x\rangle_Y - \Psi(x)\right)$. Finalement, en utilisant (4.31) on obtient :

$$\phi(\vec{y}^k) \leq -\hat{\phi}(x^k) + \frac{1}{2A^k}|\vec{x}^k - x^0|_X^2 \tag{4.41}$$

En soustrayant $\phi(y^*)$ à gauche et $-\hat{\phi}(x^*)$ à droite (il sont égaux), on obtient :

$$\phi(\vec{y}^k) - \phi(y^*) \leq \hat{\phi}(x^*) - \hat{\phi}(x^k) + \frac{1}{2A^k}|\vec{x}^k - x^0|_X^2 \tag{4.42}$$

En utilisant le théorème 4.7, on a finalement :

$$\phi(\vec{y}^k) - \phi(y^*) \quad \leq \quad \frac{|x^* - x^0|^2}{2A^k} + \frac{1}{2A^k}|\vec{x}^k - x^0|_X^2 \tag{4.43}$$

Reste à majorer $|x^* - x^0|_X^2$ et $|\vec{x}^k - x^0|_X^2$ par D pour obtenir le résultat annoncé :

$$\phi(\vec{y}^k) - \phi(y^*) \quad \leq \quad \frac{D}{A^k} \tag{4.44}$$

En utilisant le théorème 4.20 on conclut sur la croissance des A^k en choisissant $L = \frac{\|B\|^2}{l}$. Finalement, on utilise la relation (2.52) pour obtenir le taux de convergence en norme.

□

Pour conclure, nous donnons une version implémentable de l'algorithme (4.4) pour faciliter son utilisation.

Algorithme 4.5: Schéma dual accéléré implémentable

Entrée : Le nombre d'itérations N, un point de départ $x^0 \in dom(\Psi)$.

Sortie : $\overline{y}^N$ une estimée de y^*.

début

Poser A=0

Poser $L = \frac{\|B\|^2}{l}$

Poser $g = 0$

Poser $x = x^0$

Poser $\overline{y} = 0$

pour k *allant de* 0 *à* N **faire**

$t = 2\frac{1+\mu A}{L}$.

$a = \frac{t + \sqrt{t^2 + 4tA}}{2}$.

$v = \text{prox}_{A\Psi}\left(x^0 - g\right)$

$z = \frac{Ax + av}{A + a}$

$x = \text{prox}_{\frac{1}{L}\Psi}\left(z + \frac{\nabla \hat{f}(z)}{L}\right)$

$y = \arg\min_{y \in Y} \left(\langle y, B^* x\rangle_Y + \varphi(y)\right)$

$\overline{y} \leftarrow \overline{y} + ay$

$g \leftarrow g - a\nabla \hat{f}(x)$

$A \leftarrow A + a$

fin

$\overline{y}^N = \frac{\overline{y}}{A}$

fin

4.3 Somme de fonctions max et de fonctions simples

Dans les deux parties précédentes on a présenté deux techniques qui permettent de résoudre des problèmes tels que celui de Rudin-Osher-Fatemi ou celui de l'échantillonnage compressif. Ces algorithmes ne sont cependant utilisables que s'il est possible de rendre différentiable une partie du problème. Pour l'échantillonage compressif ou la décomposition sur un dictionnaire (4.1), on avait besoin de la différentiabilité du terme $|Ax - b|_Y^2$. Pour le problème de Rudin-Osher-Fatemi (4.13), on avait besoin de la différentiabilité de la fonction coût duale (4.18) $|y_0 + \frac{1}{2\lambda}\text{div}\, x|_X^2$.

De nombreux problèmes récents de traitement d'images ne satisfont pas cette hypothèse. C'est le cas notamment de certains modèles de décomposition d'images en texture et en géométrie. Prenons l'exemple simple du modèle $BV - L^1$ analysé notamment dans [Alliney, 1997; Nikolova, 2004; Chan et Esedoglu, 2005; Darbon et Sigelle, 2006]. Celui-ci s'écrit sous la forme suivante :

$$\min_{x \in X}\left(|\nabla x|_{l^1(Y)} + \lambda |x - x_0|_{l^1(X)}\right) \tag{4.45}$$

La fonction à minimiser peut être vue comme la somme de deux fonctions de type max ou comme la somme d'une fonction simple et d'une fonction de type max. Le terme $|\nabla x|_{l^1(Y)}$ est non différentiable,

la méthode de la section 4.1 n'est donc pas applicable. La fonction $|x - x_0|_{l^1(X)}$ est non fortement convexe, la méthode de la section 4.2 n'est donc pas - elle non plus - applicable. Pour le moment, la seule méthode applicable que l'on ait présentée est la méthode de sous-gradient (3.20) qui converge en $O\left(\frac{1}{\sqrt{k}}\right)$.

Dans cette section, nous proposons trois méthodes pour résoudre ce genre de problèmes. Nous nous intéressons aux problèmes du type :

$$\min_{x \in X} (f(x) + \Psi(x)) \tag{4.46}$$

où f est une fonction max et Ψ est une fonction simple. On note $\phi = f + \Psi$. Les schémas que l'on va présenter permettent aussi de minimiser la somme d'une fonction Lipschitz différentiable, d'une fonction max et d'une fonction simple. Nous avons enlevé la fonction Lipschitz différentiable pour simplifier la présentation. Toutes les méthodes suggérées ont un taux de convergence en $O\left(\frac{1}{k}\right)$ et partagent le même principe :

- Régulariser la fonction max pour la rendre Lipschitz différentiable.
- Minimiser la fonction régularisée à l'aide d'un schéma efficace de Y. Nesterov. L'efficacité du schéma permet de compenser l'erreur d'approximation.

Ce principe a été initialement proposé par Y. Nesterov dans [Nesterov, 2005b]. Nous présentons son principe puis nous proposons une deuxième version originale ou seule la fonction duale est régularisée. Nous avons récemment découvert une variante où les fonctions primale et duale sont régularisées dans [Nesterov, 2005a]. Ces différents schémas ont tous le même taux de convergence théorique en $O\left(\frac{C}{k}\right)$, mais la constante C dépend du schéma. Suivant les problèmes, il faut donc préférer l'une ou l'autre de ces versions.

4.3.1 Régularisation du problème primal

Dans ce paragraphe, nous présentons une stratégie de régularisation du problème primal (4.46). Montrons le principe de cette stratégie sur le problème $BV - L^1$. Plutôt que de résoudre (4.45), on résout le problème perturbé suivant :

$$\min_{x \in X} \left(\sum_{i=1}^{n} \sqrt{|(\nabla x)|_i^2 + \mu^2} + \lambda |x - x_0|_{l^1(X)} \right) \tag{4.47}$$

On peut montrer que $|\sum_{i=1}^{n} \sqrt{|(\nabla x)_i|^2 + \mu^2} - |\nabla x|_{l^1(Y)}| \leq n\mu$. C'est donc une bonne approximation de la norme l^1 [5]. De plus, cette nouvelle fonction est $\frac{\|\text{div}\|^2}{\mu}$-Lipschitz différentiable. On peut donc résoudre (4.47) avec le schéma de Nesterov accéléré 4.2. Dans la suite, nous généralisons cette approche et analysons son taux de convergence.

On suppose ici que dans l'équation (4.46), f peut s'écrire sous la forme :

$$f(x) = \max_{y \in Y} (\langle Ax, y \rangle_Y - \varphi(y)) \tag{4.48}$$

[5] Le fait que l'erreur d'approximation augmente linéairement en n n'est pas gênant. Pour un problème en dimension n, il est naturel de souhaiter obtenir une solution de précision $n\epsilon$.

où $\varphi : Y \to \overline{\mathbb{R}}$ est une fonction convexe de domaine borné. Le lemme 2.8 permet alors de définir une approximation lisse de f. On choisit une fonction fortement convexe $d : dom(\varphi) \to \mathbb{R}$ de paramètre de forte convexité $l > 0$. On peut alors définir l'approximation de f suivante :

$$f_\mu(x) = \max_{y \in Y}(\langle Ax, y\rangle_Y - \varphi(y) - \mu d(y)) \tag{4.49}$$

Cette fonction a un gradient Lispchitz dont la constante est :

$$L_\mu = \frac{\|A\|^2}{\mu l} \tag{4.50}$$

De plus l'approximation f_μ satisfait :

$$f_\mu(x) \leq f(x) \leq f_\mu(x) + \mu D \tag{4.51}$$

avec $D = \max_{y \in dom(\varphi)}(d(y))$. On peut maintenant définir le problème approché de (4.46) :

$$\min_{x \in X}\Big(f_\mu(x) + \Psi(x)\Big) \tag{4.52}$$

Ce problème peut être résolu efficacement avec le schéma multi-pas de Y. Nesterov. On note $\{x_\mu^k\}$ la suite de points générée par l'algorithme (4.1), $\phi_\mu = f_\mu + \Psi$, x^* la solution du problème (4.46) et x_μ^* la solution du problème (4.52). On peut alors montrer le résultat suivant :

Théorème 4.22. *La stratégie suivante :*

– *Fixer un nombre d'itérations N*

– *Fixer un point de départ* x_μ^0

– *Choisir le paramètre de régularisation* μ *de la façon suivante :* $\mu = \mu(N) = \dfrac{\|A\| \cdot |x_\mu^0 - x_\mu^*|_X}{N\sqrt{lD}}$

– *Appliquer l'algorithme 4.1 au problème (4.52) avec les paramètres ci-dessus*

assure que :

$$0 \leq \phi(x_\mu^N) - \phi(x^*) \leq 2\frac{\|A\| \cdot |x_\mu^0 - x_\mu^*|_X \sqrt{D}}{N\sqrt{l}} \tag{4.53}$$

Démonstration. D'après (4.51), on a $\forall x \in X$:

$$\phi(x) \leq \phi_\mu(x) \leq \phi(x) + \mu D \tag{4.54}$$

De plus, d'après le théorème 4.20, on a :

$$\phi_\mu(x_\mu^k) \leq \phi_\mu(x_\mu^*) + \frac{\|A\|^2 |x_\mu^0 - x_\mu^*|_X^2}{\mu l k^2} \tag{4.55}$$

D'où :

$$\begin{aligned}
\phi(x_\mu^k) &\leq \phi_\mu(x_\mu^k) && (4.56)\\
&\leq \phi_\mu(x_\mu^*) + \frac{\|A\|^2 |x_\mu^0 - x_\mu^*|_X^2}{\mu l k^2} && (4.57)\\
&\leq \phi_\mu(x^*) + \frac{\|A\|^2 |x_\mu^0 - x_\mu^*|_X^2}{\mu l k^2} && (4.58)\\
&\leq \phi(x^*) + \mu D + \frac{\|A\|^2 |x_\mu^0 - x_\mu^*|_X^2}{\mu l k^2} && (4.59)
\end{aligned}$$

Finalement au bout de N itérations, on sait que :

$$0 \leq \phi(x_\mu^N) - \phi(x^*) \leq \mu D + \frac{\|A\|^2 |x_\mu^0 - x_\mu^*|_X^2}{\mu l N^2} \tag{4.60}$$

On peut maintenant minimiser la fonction $\mu \mapsto \mu D + \frac{\|A\|^2 |x_\mu^0 - x_\mu^*|_X^2}{\mu l N^2}$. Le minimum est atteint pour $\mu = \mu(N) = \frac{\|A\| \cdot |x_\mu^0 - x_\mu^*|_X}{N\sqrt{lD}}$ et il vaut $2\frac{\|A\| \cdot |x_\mu^0 - x_\mu^*|_X \sqrt{D}}{N\sqrt{l}}$.

□

Notons que dans les hypothèses du théorème 4.22, on a besoin de connaître le terme $|x_\mu^0 - x_\mu^*|_X$. C'est rarement le cas. En choisissant une majoration de ce terme on maintient le taux de convergence en $O\left(\frac{1}{k}\right)$. Une telle majoration peut notamment être facilement obtenue si Ψ a un domaine borné. Notons aussi qu'on a dû faire l'hypothèse que φ a un domaine borné pour s'assurer que l'approximation f_μ était proche de f. La pratique montre que cette hypothèse n'est pas forcément nécessaire. Expliquer ce phénomène est un thème d'étude intéressant qu'on reprendra peut-être dans nos travaux futurs.

Nous avons donc présenté une stratégie de minimisation qui converge en $O\left(\frac{1}{k}\right)$. Ce taux de convergence est bien plus satisfaisant que le taux des méthodes de sous-gradient en $O\left(\frac{1}{\sqrt{k}}\right)$ (pourtant optimales !). La régularisation des normes l^1 ou l^∞ est une approche souvent critiquée dans les articles de recherche. L'argument généralement invoqué est qu'en minimisant une approximation, on ne converge pas vers la bonne solution. Le résultat de convergence qu'on a présenté montre au contraire que *régulariser des fonctions non-différentiables est une bonne solution pour exploiter la connaissance de leur structure.*

4.3.2 Régularisation du problème dual

Dans cette partie, nous présentons une stratégie de régularisation alternative. Cette stratégie a été utilisée récemment dans plusieurs travaux [Aujol et Chambolle, 2005; Aujol *et al.*, 2006; Huang *et al.*, 2008]. Dans la suite nous analysons son efficacité théorique (ce qui à notre connaissance est un résultat original). Nous proposons aussi une stratégie de minimisation plus efficace que celles proposées dans [Aujol et Chambolle, 2005; Aujol *et al.*, 2006; Huang *et al.*, 2008]. Dans ces articles, les auteurs utilisent des algorithmes de minimisations alternées. Nous n'avons pas trouvé d'intérêt (en termes de taux de convergence) à ces approches, car une minimisation directe peut être effectuée et évite d'obtenir des erreurs difficilement contrôlables à chaque itération.

On considère le problème suivant :

$$\min_{y\in Y}\left(\underbrace{f(y)+\varphi(y)}_{\phi(y)}\right) \tag{4.61}$$

où φ est une fonction simple et f est une fonction max s'écrivant sous la forme :

$$f(y)=\max_{x\in X}\left(\langle Ay,x\rangle_X-\Psi(x)\right) \tag{4.62}$$

Dans cette équation, Ψ est une fonction convexe de domaine borné[6]. L'idée de la méthode que l'on présente consiste à régulariser non pas la fonction primale, mais la fonction duale. Plutôt que de résoudre (4.61) on résout le problème perturbé suivant :

$$\min_{y\in Y}\left(\underbrace{f(y)+\varphi(y)+\frac{l}{2}|y-y^0|_Y^2}_{\phi_l(y)}\right) \tag{4.63}$$

où $y^0\in dom(\varphi)$ est une estimée de la solution y^* de (4.61). La fonction $y\rightarrow f(y)+\frac{l}{2}|y-y^0|_Y^2$ est fortement convexe de paramètre de convexité l. On peut donc utiliser l'algorithme présenté dans la partie 4.2 pour résoudre (4.63). La dualité min-max donne :

$$\min_{y\in Y}\left(\phi(y)+\frac{l}{2}|y-y^0|_Y^2\right) \tag{4.64}$$

$$=\min_{y\in Y}\left(\max_{x\in X}\left(\langle Ay,x\rangle_X-\Psi(x)\right)+\varphi(y)+\frac{l}{2}|y-y^0|_Y^2\right) \tag{4.65}$$

$$=\max_{x\in X}\left(\underbrace{\min_{y\in Y}\left(\langle y,A^*x\rangle_X+\varphi(y)+\frac{l}{2}|y-y^0|_Y^2\right)}_{\hat{f}_l(x)}-\Psi(x)\right) \tag{4.66}$$

$$=-\min_{x\in X}\left(\underbrace{-\hat{f}_l(x)+\Psi(x)}_{\hat{\phi}_l(x)}\right) \tag{4.67}$$

L'algorithme 4.4 génère une suite de points $\{\vec{y}_l^k\}$ qui converge vers la solution y_l^* du problème (4.63). Notons x_l^* la solution du problème (4.67), y_l^* la solution du problème (4.63) et y^* la solution de (4.61). On peut alors montrer le résultat suivant :

Théorème 4.23. *Soit la stratégie suivante :*

- *Choisir un point y^0 (proche de y^*) et x^0 (proche de x_l^*).*
- *Choisir un nombre d'itérations N.*

[6]les conditions d'applicabilité de la méthode de régularisation du problème dual sont donc les mêmes que celles de la méthode de régularisation du problème primal

– *Poser* $l = \dfrac{|x^0 - x_l^*|_X \|A\| \sqrt{2}}{|y^0 - y^*|_Y \cdot N}$

– *Appliquer l'algorithme 4.4 au problème (4.63).*

Elle assure que :

$$\phi(\overline{y}_l^N) - \phi(y^*) \leq \frac{\sqrt{2}|x^0 - x_l^*|_X |y^0 - y^*|_Y \|A\|}{N} \tag{4.68}$$

Démonstration. On a :

$$\begin{aligned}
\phi(\overline{y}_l^N) &\overset{\text{def}}{\leq} \phi_l(\overline{y}_l^N) && (4.69)\\
&\overset{4.26}{\leq} \phi_l(y_l^*) + \frac{|x^0 - x_l^*|_X^2}{A^N} && (4.70)\\
&\overset{\min}{\leq} \phi_l(y^*) + \frac{|x^0 - x_l^*|_X^2}{A^N} && (4.71)\\
&\overset{\text{def}}{\leq} \phi(y^*) + \frac{l}{2}|y^* - y^0|_Y^2 + \frac{|x^0 - x_l^*|_X^2}{A^N} && (4.72)\\
&\overset{4.26}{\leq} \phi(y^*) + \frac{l}{2}|y^* - y^0|_Y^2 + \frac{|x^0 - x_l^*|_X^2 \cdot \|A\|^2}{k^2 l} && (4.73)
\end{aligned}$$

Au bout de N itérations on s'assure donc que :

$$0 \leq \phi(\overline{y}_l^N) - \phi(y^*) \leq \frac{l}{2}|y^* - y^0|_Y^2 + \frac{|x^0 - x_l^*|_X^2 \cdot \|A\|^2}{k^2 l} \tag{4.74}$$

En minimisant la partie de droite de (4.74) par rapport à l, on obtient le résultat annoncé.

□

Comme dans la partie précédente, il faut être capable d'évaluer $|x^0 - x_l^*|_X$ et $|y^0 - y^*|_Y$ pour appliquer la stratégie proposée dans le théorème 4.23. Ceci est rarement possible mais il suffit en fait de choisir l de l'ordre de $\frac{1}{N}$ pour assurer que l'erreur décroît elle aussi en $\frac{1}{N}$. La stratégie présentée est donc implémentable et converge en $O\left(\frac{1}{k}\right)$.

4.3.3 Régularisation du problème primal-dual

Dans le cas particulier où $dom(\varphi)$ et $dom(\Psi)$ sont bornés, Y. Nesterov dans [Nesterov, 2005a] propose une méthode de minimisation qui repose sur la régularisation des fonctions primales et duales. L'intérêt de cette méthode est que les paramètres de régularisation sont diminués itérativement lors du processus de minimisation. Ainsi, ce n'est pas la peine de fixer le nombre d'itérations pour savoir comment choisir les paramètres de régularisation l et μ. Nous détaillons le principe de cette méthode et l'appliquons sur le problème de décomposition en texture et en géométrie dans l'annexe A.3. Ce schéma de minimisation semble n'avoir jamais été implémenté en pratique (le papier [Nesterov, 2005a] ne contient aucun exemple). Notre implémentation mène à des résultats décevants par rapport aux méthodes présentées précédemment. Bien que son principe soit très intéressant, nous ne la recommandons donc pas en pratique.

Finalement, on peut se demander s'il est intéressant de régulariser à la fois les fonctions primales et duales. Ainsi la fonction à minimiser devient à la fois fortement convexe et Lipschitz différentiable. Cette stratégie permet de minimiser la fonction approchée avec un taux de convergence linéaire (4.8). Cependant, le conditionnement du problème est trop mauvais, et cette méthode ne semble pas apporter de gain en pratique.

4.4 Conclusion

Dans ce chapitre, nous avons présenté plusieurs algorithmes de minimisation dont les performances théoriques dépassent nettement celles des méthodes plus classiques présentées dans le chapitre 3. Deux d'entre eux sont à notre connaissance originaux (voir parties 4.2 et 4.3.2).

Une comparaison des taux de convergence théoriques ne suffit pas forcément à montrer qu'une méthode est plus efficace qu'une autre. En effet, les taux que l'on a présentés jusqu'à présent sont les plus précis *uniformément* sur de larges classes de problèmes, mais ils peuvent être bien meilleurs sur des instances particulières de ces classes. De plus, nous n'avons pas étudié l'impact des bruits de quantification qui apparaissent nécessairement pour des calculs en arithmétique finie et qui peuvent dégrader notablement l'efficacité d'une méthode. Nous vérifierons donc expérimentalement, dans le chapitre 7, que le classement des méthodes est effectivement respecté. Les méthodes que nous venons de présenter sont bien plus efficaces que les méthodes du chapitre 3.

Par la suite, nous allons montrer que ces algorithmes peuvent être appliqués à de nombreux problèmes de restauration rencontrés couramment en traitement des images. Nous traiterons notamment les problèmes de restauration avec des a priori de régularité ou de parcimonie. Nous traiterons différents types de perturbations telles que le flou, des bruits additifs impulsionnels, gaussiens, uniformes ou de compression. Nous montrerons que les algorithmes présentés permettent aussi de résoudre efficacement des problèmes plus complexes tels que la reconstruction d'images floues, échantillonnées irrégulièrement avec l'un des bruits cités précédemment. Nous les appliquerons aussi pour des tâches de décomposition d'images en texture et en géométrie.

Chapitre 5

Application à la restauration d'images

La restauration d'images numériques est un domaine très vaste. Il existe des méthodes de restauration très variées adaptées à de nombreux systèmes optiques différents. Dans cette thèse nous nous focaliserons sur les méthodes dites variationnelles. Un lecteur intéressé par une vision plus large des différentes méthodes de restauration peut se référer à la liste non exhaustive suivante :

- Le livre [Mallat, 1999] présente la théorie et les applications des ondelettes.
- Le livre [Aubert et Kornprobst., 2006] présente les méthodes variationnelles en continu et les approches par équations aux dérivées partielles.
- Les livres [Tarantola, 2005; Idier, 2008] présentent de façon détaillée les problèmes inverses et les approches bayésiennes sur différents problèmes de restauration.
- Les articles [Buades *et al.*, 2005b; Buades *et al.*, 2005a] présentent un nouveau formalisme de restauration : les approches non locales.

De façon assez générale on peut considérer qu'une image numérique est formée par un système physique qui restitue une partie de l'information présente dans l'image analogique originale. En mathématiques, on peut représenter le système d'acquisition de l'image numérique par une fonction :

$$\begin{array}{rccl} \tilde{F}: & \mathcal{H} & \rightarrow & Y \\ & \mathbf{y_0} & \rightarrow & y_0 = \tilde{F}(\mathbf{y_0}) \end{array} \tag{5.1}$$

Dans cette équation, l'espace $\mathcal{H}$ est un espace fonctionnel (L^2, Sobolev, BV, Besov,...) modélisant tout ce qui peut représenter une image analogique. $\tilde{F}$ modélise le système d'acquisition (échantillonnage, flou, bruits, compression, ...). $\mathbf{y_0}$ est l'image continue "parfaite" que l'on souhaite retrouver à partir de l'observation y_0. Notons tout de suite qu'il est impossible de restituer toute l'information contenue dans $\mathbf{y_0}$ à partir d'un nombre n fini de pixels, à moins que $\mathcal{H}$ soit lui-même un espace de dimension finie ou dénombrable (fonctions à spectre borné, fonctions creuses dans une base). Retrouver $\mathbf{y_0}$ semble donc sans espoir, par contre on peut essayer de l'approcher au mieux possible. Le problème de la restauration d'images peut donc être formulé comme suit : trouver l'image la plus "proche" de $\mathbf{y_0}$ à partir de y_0.

Etant donné un système optique $\tilde{F}$ et une image numérique y_0, on sait que $\mathbf{y^0} \in \tilde{F}^{-1}(y^0)$. Cet ensemble est rarement réduit à un singleton. Pour approcher $\mathbf{y_0}$ il faut alors définir des a priori sur l'image. Par exemple, on peut rechercher l'image la plus "régulière" parmi les éléments de $\tilde{F}^{-1}(y_0)$. Ceci peut se faire à l'aide d'une énergie $\tilde{\phi} : \mathcal{H} \rightarrow \mathbb{R}$ sur les images. On définit l'image restaurée y^* comme suit :

$$y^* = \underset{y \in \tilde{F}^{-1}(y_0)}{\arg\inf} \left(\tilde{\phi}(y)\right) \tag{5.2}$$

Un système informatique est incapable de travailler dans des espaces de fonctions de dimension infinie. Pour résoudre le problème (5.2) on a donc souvent recours à l'astuce suivante. On définit une famille d'espaces $\{\mathcal{H}^m\}_m$ de dimension finie ($m \geq n$) qui satisfait $\forall y \in \mathcal{H}$:

$$\lim_{m \to \infty} |\Pi_{\mathcal{H}^m}(y) - y|_{\mathcal{H}} = 0 \tag{5.3}$$

où $|\cdot|_{\mathcal{H}}$ est une norme sur $\mathcal{H}$. Pour certains espaces $\mathcal{H}$, on peut obtenir des approximations plus fines telles que :

$$|\Pi_{\mathcal{H}^m}(y) - y|_{\mathcal{H}} = \frac{cte}{m^\alpha} \tag{5.4}$$

avec $\alpha > 0$. De telles techniques sont couramment utilisées lorsqu'on utilise des éléments finis ou des différences finies dans des espaces de fonctions régulières. La théorie de la gamma convergence [Braides, 2002] permet souvent de montrer la convergence de solutions discrétisées vers les solutions continues. Pour des travaux en traitement d'images, nous renvoyons le lecteur aux articles [Chambolle et Lions, 1997; Aubert *et al.*, 2004]. Dans la suite de ce chapitre, on identifiera $\mathcal{H}^m$ à $\mathbb{R}^m$ pour simplifier les notations. Dans beaucoup de cas, on ne cherchera pas à augmenter la résolution de l'image et donc m sera égal à n. Finalement, les problèmes de restauration d'images que nous considérons dans cette thèse s'écrivent sous la forme suivante :

$$y^* = \underset{y \in F^{-1}(y_0)}{\arg\inf} \left(\phi(y)\right) \tag{5.5}$$

où $\phi : \mathbb{R}^m \to \mathbb{R}$ est l'a priori sur l'image et $F^{-1}(y_0) = \Pi_{\mathcal{H}^m}(\tilde{F}^{-1}(y_0))$. Le reste de ce chapitre est organisé comme suit :

- On présente d'abord deux types d'a priori convexes ϕ sur les images qui ont emergé relativement récemment. Ces a priori mènent à des résultats de restauration satisfaisants. Ils donnent lieu à des problèmes pouvant être résolus rapidement (en utilisant les algorithmes présentés dans cette thèse notamment), et permettent de faire une analyse mathématique de certaines propriétés des solutions (existence, unicité, principe du maximum, ...).
- On montre finalement que de nombreux types de dégradations d'images permettent d'exprimer $F^{-1}(y_0)$ comme un ensemble convexe assez simple.
- On montre ensuite à travers de nombreux exemples que les méthodes d'optimisation structurelles présentées précédemment permettent de résoudre efficacement tous les problèmes de restauration présentés.

5.1 Deux types d'a priori convexes sur les images

Le formalisme de restauration du type (5.5) existe depuis bientôt 30 ans, et de très nombreux a priori sur les images ont été proposés. Nous n'allons donc pas faire une revue exhaustive des a priori proposés dans la littérature, mais restreignons notre présentation à deux types d'a priori convexes qui ont récemment montré leur efficacité dans de nombreux travaux.

5.1.1 Les a priori de régularité

Des travaux tels que [Geman et Geman, 1984; Rudin *et al.*, 1992a; Charbonnier *et al.*, 1994; Aubert et Vese, 1997; Chambolle *et al.*, 1998] ont permis de mettre en avant des a priori de régularité du type :

$$\phi(y) = |By|_{l^1(X)} \tag{5.6}$$

où $B : Y = \mathbb{R}^m \to X$ est une transformée linéaire de l'image. Quelques fois la norme l^1 est remplacée par une approximation du type :

$$\phi(y) = \sum_{i=1}^{m} \varphi(|(By)_i|) \tag{5.7}$$

où $\varphi : \mathbb{R}^+ \to \mathbb{R}$ est une fonction convexe se comportant comme une valeur absolue à l'infini. Ces a priori peuvent être vus comme des semi-normes liées à des espaces fonctionnels :

- Si $B = \nabla$, $|By|_{l^1(X)}$ correspond à la variation totale. Elle a été introduite dans [Rudin *et al.*, 1992a], puis a été très largement utilisée. Elle mesure d'une certaine façon la quantité d'oscillations que contient une fonction. Par exemple, un bruit blanc a une variation totale très forte tandis qu'une fonction constante par morceaux a une variation totale modérée si l'ensemble de ses discontinuités à une longueur finie. Minimiser la variation totale revient donc à supposer qu'une fonction a d'autant plus de chances de modéliser une image que celle-ci est "régulière". Son intérêt réside essentiellement dans le fait qu'elle préserve les discontinuités. Notons cependant que c'est un modèle d'images assez grossier qui ne permet pas de préserver les textures (parties oscillantes des images) [Gousseau et Morel, 2001]. Il reste aujourd'hui très apprécié pour reconstruire les parties lisses par morceaux des images. Un autre intérêt de la variation totale est que le calcul du gradient est simple et rapide à calculer. Ainsi, on peut obtenir des algorithmes de minimisation plus rapides que pour d'autres transformées telles que des transformées en ondelettes redondantes.
- Si B est une transformée en ondelettes, $|By|_{l^1(Y)}$ est une semi-norme équivalente à la semi-norme de l'espace de Besov $B^1_{1,1}$ [Chambolle *et al.*, 1998; Meyer, 2001; Aujol et Chambolle, 2005]. Cet espace est inclus dans BV. Notons qu'il ne contient pas les fonctions indicatrices d'ensembles à périmètre fini. On peut donc s'attendre à ce qu'il préserve moins bien les contours que la semi-norme BV. Il a été utilisé notamment pour accélérer certains algorithmes de décomposition d'images [Aujol *et al.*, 2006].
- Enfin, notons que récemment, des transformées redondantes telles que les ondelettes complexes ont été utilisées et fournissent de bons résultats de restauration [Carlavan *et al.*, 2009]. Nous présentons un exemple sur la figure 5.10.

5.1.2 Les représentations parcimonieuses

Une autre façon de modéliser les images consiste à considérer qu'il existe une base dans laquelle leur représentation est creuse. Ceci signifie qu'une image y_0 peut s'exprimer comme la somme pondérée d'un nombre fini (faible) d'atomes élémentaires :

$$\mathbf{y_0} = \sum_{i \in I} \mathbf{x}_i \varphi_i \tag{5.8}$$

où $m = Card(I) < \infty$ et $x_i \in \mathbb{R}$ sont des coefficients de pondération. Les fonctions φ_i sont parfois appelées atomes. Cette approche a été formalisée dans [Mallat et Zhang, 1993; Chen *et al.*, 1998]. La recherche de bases adaptées aux images a donné lieu à de nombreuses transformées. Les premières transformées proposées ont sans doute été les transformées de Fourier ou les transformées en cosinus. Lorsque des algorithmes rapides ont existé, les analyse temps-fréquences en ondelettes se sont rapidement montrées bien plus efficaces [Mallat, 1999]. Récemment de nombreuses autres transformées ont vu le jour. Elles sont adaptées à différent types d'images. Citons notamment les curvelets [Candès et Donoho, 1999], les bandlets [Le Pennec et Mallat, 2005], les beamlets [Donoho et Huo, 2002], les transformées en ondelettes redondantes [Kingsbury, 1998; Chaux *et al.*, 2006].

Dans un cadre discret, ces transformées peuvent s'écrire sous la forme d'une matrice $B \in \mathbb{R}^{n \times m}$ dont les colonnes sont constituées des fonctions φ_i discrétisées. On appelle souvent cette matrice dictionnaire. On peut écrire :

$$y_0 = Bx \tag{5.9}$$

où $y_0 \in \mathbb{R}^n$ et $x \in \mathbb{R}^m$. Lorsque $m > n$, le dictionnaire est dit redondant.

Si l'on souhaite utiliser cet a priori, le problème de restauration (5.5) peut être exprimé comme suit :

$$x^* = \underset{Bx \in F^{-1}(y_0)}{\arg\inf} \left(|x|_{l^0(X)} \right) \tag{5.10}$$

où $|\cdot|_{l^0(\mathbb{R}^m)}$ est la norme l^0 définie comme le nombre de composantes non nulles de x. La solution y^* de (5.5) est alors définie comme $y^* = Bx^*$. Minimiser une norme l^0 est un problème combinatoire connu pour être NP-complet. Il ne peut donc pas être résolu pour des problèmes de très grande dimension. Pour approcher la solution de (5.10) on remplace souvent la norme l^0 par son enveloppe convexe qui est la norme l^1. Le problème (5.10) devient alors :

$$x^* = \underset{Bx \in F^{-1}(y_0)}{\arg\inf} \left(|x|_{l^1(X)} \right) \tag{5.11}$$

Cette stratégie a été initialement proposée dans un cadre moins général dans [Chen *et al.*, 1998]. Dans certains cas, on sait montrer que la solution de (5.11) est la même que celle du problème (5.10) [Tropp, 2006].

5.2 Différents types de perturbations

5.2.1 Les sources de perturbations

Un imageur numérique est constitué de nombreux composants optiques et électroniques qui introduisent presque tous des perturbations sur l'image. Le schéma de la figure 5.1 est issu de la thèse [Jalobeanu, 2001]. Il résume le principe d'acquisition d'une image satellite ou d'une image d'appareil photo numérique.

Chaque étape de l'acquisition introduit des perturbations sur l'image. Certaines sont déterministes, d'autres sont probabilistes. Les perturbations les plus importantes sont généralement le flou et le bruit.

Le flou a trois sources principales :

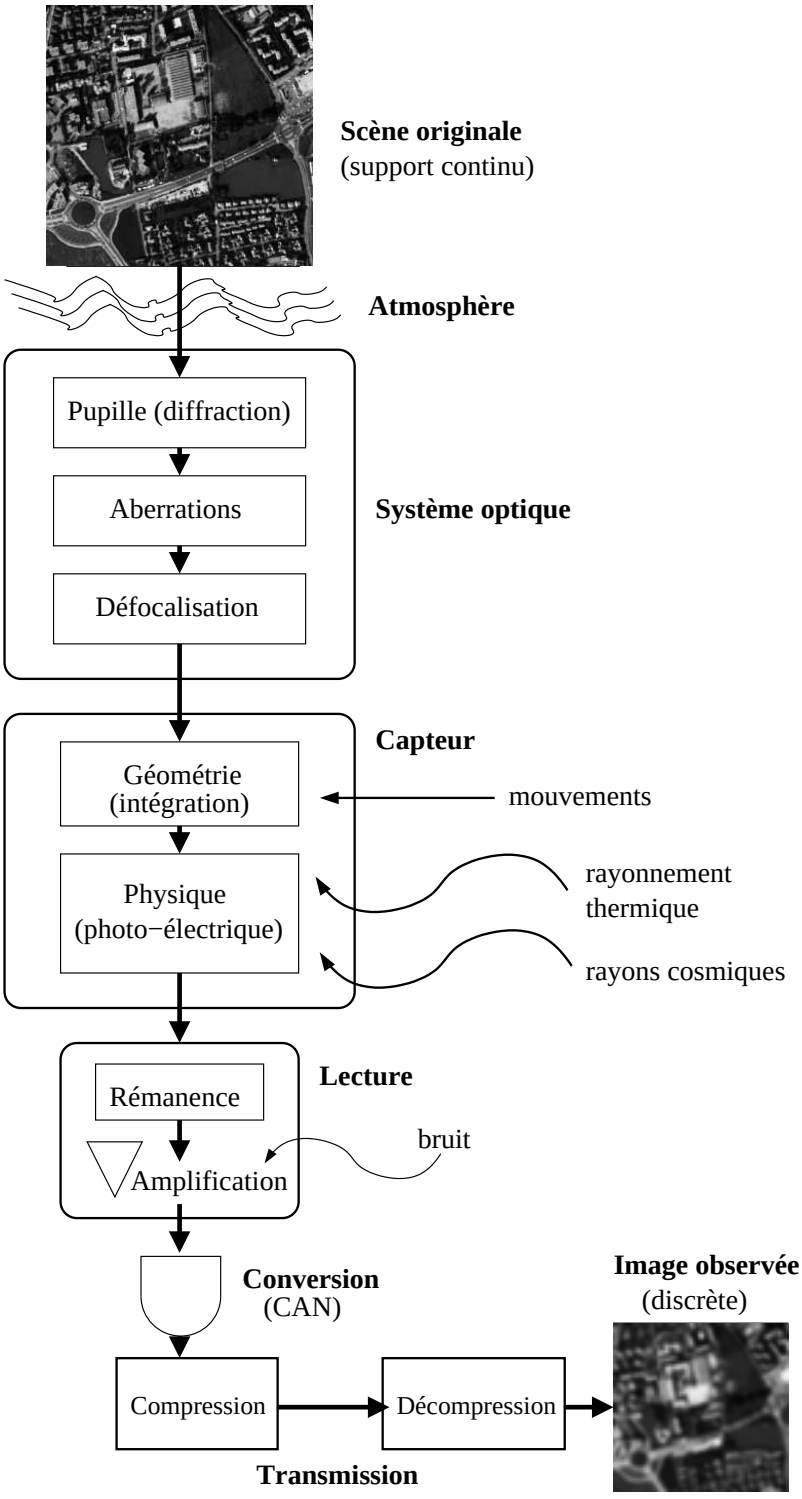

FIG. 5.1 – Diagramme résumant le processus de formation d'une image numérique. Nous tenons à remercier A. Jalobeanu de nous avoir fourni cette image.

– Le flou de diffraction. Ce flou est inévitable, il est dû au fait que l'ouverture du diaphragme est limitée.
– Le flou de capteur. Les capteurs CCD intègrent la lumière sur leur domaine.
– Le flou de bouger. Il est présent dès que l'imageur ou la scène est en mouvement.

Les bruits ont aussi des origines variées liées notamment aux imperfections des appareils de mesure et au système de transmission de l'information. On distingue :

– Le bruit photonique. Il est dû à l'arrivée aléatoire des photons.
– Le bruit d'amplification. Il est lié au système de transmission électronique de l'image. Il dépend notamment de la température des composants eléctroniques.
– Les bruits de compression. Aujourd'hui l'image est rarement transmise sans compression préalable. Les algorithmes de compression introduisent aussi des perturbations sur l'image.

Il existe d'autres perturbations importantes de l'image dont nous ne parlerons pas pendant cette thèse. Nous les citons à titre indicatif :

– Les aberrations chromatiques. La distance focale optimale dépend de la longueur d'onde de la lumière. Il est donc impossible d'avoir une mise au point adaptée à tous les canaux de couleurs.
– Le vignettage. L'intensité lumineuse n'est pas répartie de façon uniforme sur l'image.
– L'astigmatisme et le coma. Tous les rayons lumineux provenant d'un point identique ne convergent pas forcément vers le même point du plan image. Ceci introduit des perturbations non linéaires.

5.2.2 La modélisation mathématique des perturbations

En mathématiques, un modèle d'acquisition d'images assez général peut s'écrire sous la forme suivante :

$$y_0 = F(\mathbf{y_0}) = C(\Delta_\Lambda(\mathbf{y_0} \star h) + b) \tag{5.12}$$

Dans cette équation, l'opération $\cdot \star h$ est un produit de convolution avec le filtre h. Ce filtre est appelé fonction d'étalement et permet de représenter tous les types de flous linéaires cités précédemment. b est un bruit additif dont les propriétés dépendent de l'application. Nous en présenterons plusieurs dans la suite de cette thèse. Notons qu'il existe aussi des bruits multiplicatifs que nous ne traiterons pas ici [Aubert et Aujol, 2008]. Δ_Λ est un opérateur d'échantillonnage. Il peut être vu comme une somme de Diracs positionnés de façon régulière ou non. C est un opérateur de compression de l'image.

Ce modèle très général rend l'ensemble $F^{-1}(y_0)$ difficile à exprimer et à utiliser en pratique. A notre connaissance, aucun modèle de restauration n'a encore été proposé dans la littérature pour résoudre le problème (5.5) avec un opérateur F prenant en compte autant de dégradations. Par contre plusieurs modèles prennent en compte une ou deux de ces perturbations. On montre dans la suite de cette partie que pour beaucoup de ces perturbations, l'ensemble $F^{-1}(y_0)$ a la particularité de s'écrire sous la forme suivante :

$$F^{-1}(y_0) = \{y \in Y, |\lambda(Ay - y_0)|_{l^P(Y)} \leq \alpha\} \tag{5.13}$$

Dans ce problème, A est une transformée linéaire (identité, transformée en ondelettes, transformée de Fourier,...), λ est une matrice diagonale dont les éléments appartiennent à $[0, \infty]$, p appartient à $\{1, 2, \infty\}$, finalement y_0 est l'image à restaurer. Nous détaillons ci-dessous plusieurs applications suivant le choix de A et de p.

$A = Id$**,** $p \in \{1, 2, \infty\}$ **- débruitage ou décomposition**

Une image est très souvent perturbée par un bruit additif. Cette dégradation se modélise simplement de la façon suivante : $y_0 = y + b$. y est l'image originale (discrète), b est un bruit blanc additif et y_0 est l'image bruitée. Imaginons maintenant qu'on dispose d'une mesure de probabilité $P(y)$ sur l'espace des images proportionnelle à $exp(-\phi(y))$ [1]. En utilisant le principe du maximum a posteriori, on peut alors montrer que le "meilleur" moyen de retrouver y à partir de y_0 consiste à résoudre le problème suivant :

$$\inf_{y \in Y, |y-y_0|_p \leq \alpha} (\phi(y)) \tag{5.14}$$

- Il faut choisir $p = 1$ si b est un bruit impulsionnel ou un bruit laplacien [Alliney, 1997; Nikolova, 2004; Chan et Esedoglu, 2005; Darbon et Sigelle, 2006]. La figure 5.3 montre un exemple de restauration par variation totale.
- Il faut choisir $p = 2$ si b est un bruit gaussien [Rudin *et al.*, 1992a]. La figure 5.2 montre un exemple de restauration par variation totale.
- Il faut choisir $p = \infty$ si b est un bruit uniforme [Weiss *et al.*, 2006]. La figure 5.4 montre un exemple de restauration par variation totale.

Enfin, α est un paramètre qui dépend de la variance du bruit. Notons dès à présent que toutes les images ont été obtenues à partir de l'algorithme présenté dans le paragraphe 4.3.1. Le bruit peut être non homogène et avoir une variance variable en fonction de la position sur l'image. Dans ce cas, on peut résoudre le problème suivant :

$$\inf_{y \in Y, |\lambda(y-y_0)|_p \leq \alpha} (\phi(y)) \tag{5.15}$$

où $\lambda = diag(\lambda_i)$ avec $\lambda_i \in [0, \infty]$ est une matrice diagonale qui permet de traiter différemment différentes parties de l'image. Au niveau des pixels qui satisfont $\lambda_i = \infty$ le modèle impose $y_i^* = (y_0)_i$. Au niveau des pixels où $\lambda_i = 0$, la valeur de y_i^* dépend seulement de l'a priori ϕ. Cette idée a notamment été proposée dans [Rudin et Osher, 1994; Bertalmio *et al.*, 2003]. Notons aussi que ce formalisme permet de faire de la complétion d'images (inpainting) [Chan et Shen, 2005].

$A =$ **transformée en ondelettes,** $p \in \{1, \infty\}$

Dans ce paragraphe, nous présentons trois modèles de restauration d'images dégradées par des traitements informatiques :

1. La restauration d'images compressées.
2. La restauration d'images qui ont été seuillées dans une base d'ondelettes.
3. Le débruitage de bruit blanc.

Détaillons ces trois problèmes :

- L'état de l'art des modèles de compression d'images partagent le même principe :
    1. Transformer l'image dans un espace où elle a une représentation creuse. Ceci est souvent fait en utilisant une transformation linéaire inversible.

[1] La constante de normalisation n'est pas infinie si on suppose que les images sont d'amplitude bornée.

FIG. 5.2 – De gauche à droite : image originale (valeur dans $[0,1]$) - image bruitée (bruit gaussien) - restauration par variation totale avec $p = 2$ et $\alpha = |b|_{l^2(Y)}$.

FIG. 5.3 – De gauche à droite : image originale (valeur dans $[0,1]$) - image bruitée (bruit poivre et sel 40%) - image restaurée par variation totale avec $p = 1$ et $\alpha = |b|_{l^1(Y)}$.

FIG. 5.4 – De gauche à droite : image originale dans $[0,1]$ - image originale + bruit $U([-0.2, 0.2])$ - image débruitée avec $p = \infty$ et $\alpha = |b|_{l^\infty(Y)}$.

2. Quantifier les coefficients de l'image transformée pour réduire l'entropie de la nouvelle représentation.
3. Utiliser un algorithme de compression sans perte pour coder les coefficients quantifiés.

La première transformée utilisée a été la transformée en cosinus locale (standard *jpeg*). Le nouveau standard *jpeg*2000 utilise une transformée en ondelettes Daubechies 9-7. Ce genre de compression introduit des effets de type "Gibbs" aux bords des contours car les ondelettes sont des fonctions oscillantes.
Dans la suite y représente l'image originale et y_0 l'image compressée. L'opérateur de dégradation F peut s'écrire :

$$F(y) = A^{-1}(Q(Ay)) \tag{5.16}$$

où Q est un quantifieur (uniforme ou non) et A est une transformée linéaire généralement inversible (transformée en cosinus locale, transformée en ondelettes, bandelets [Mallat, 1999; Pennec et Mallat, 2005]). Une façon naturelle de retrouver y à partir de y_0 est de rechercher l'image minimisant ϕ dans l'ensemble $F^{-1}(y_0)$ [Alter *et al.*, 2005; Durand et Froment, 2003]. Ce problème revient à résoudre :

$$\inf_{y\in Y, \forall i\in[1..n],\ \ |(A(y-y_0))_i|\leq\frac{\alpha_i}{2}} (\phi(y))$$

où les coefficients α_i représentent des pas de quantification. Il est facile de réécrire ce problème sous la forme [Weiss *et al.*, 2008a] :

$$\inf_{y\in Y, |\lambda A(y-y_0)|_\infty\leq 1} (\phi(y)) \tag{5.17}$$

Les coefficients de la matrice diagonale λ appartiennent à $[0,\infty]$. La figure 5.5 montre des résultats de compression et décompression en utilisant un codeur similaire à l'algorithme jpeg2000 [Antonini *et al.*, 1992]. Dans ce travail de thèse, nous avons évalué l'efficacité d'un tel algorithme sur une large base de données d'images. La conclusion de cette étude - qui n'avait pas encore pu être menée faute d'algorithmes performants - est que les modèles de décompression fournissent des résultats assez peu convaincants. Voir [Weiss *et al.*, 2008a] et [Weiss *et al.*, 2008b] pour plus de détails. Un autre enseignement de cette étude est qu'utiliser la variation totale régularisée fournit de meilleurs résultats de décompression en des temps plus courts.

– Un seuillage dans le domaine ondelettes est très souvent réalisé pour débruiter des image [Donoho, 1995; Donoho et Johnstone, 1998]. Une telle opération donne des résultats satisfaisants en des temps très courts, mais introduit des artefacts de type "Gibbs" si les ondelettes ne sont pas redondantes. On peut montrer que résoudre un problème du type (5.17), (où A est une transformée en ondelettes) permet de réduire ces artefacts (voir [Durand et Nikolova, 2007] pour plus de détails).
– Récemment un modèle similaire à (5.17), utilisant une norme l^1 à la place de la norme l^∞ a été proposée pour débruiter des images [Durand et Nikolova, 2007]. Nous renvoyons le lecteur interessé à la référence [Durand et Nikolova, 2007] pour plus de détails sur la construction des coefficients λ_i.

FIG. 5.5 – Exemple de décompression d'image - HG : image originale (valeurs dans $[0,1]$) - HD : Image compressée en utilisant des ondelettes Daubechies 9-7 (c'est une implémentation similaire à *jpeg*2000) - BG : solution du problème (5.17) en utilisant la variation totale régularisée - BD : solution de (5.17) en utilisant la variation totale.

A = **Transformée de Fourier,** $p = 2$ **- Déconvolution d'images, agrandissement**

– Un des problèmes les plus fondamentaux du traitement des images est la déconvolution (pour les non spécialistes : enlever le flou). Une telle dégradation est généralement modélisée de la façon suivante : $y_0 = h \star y + b$. y est l'image originale recherchée, b est un bruit blanc gaussien et h est un noyau de convolution représentant les dégradations dues au système optique et aux capteurs. Pour retrouver l'image originale, on peut résoudre le problème suivant :

$$\inf_{y \in Y, |h \star y - y_0|_2 \leq \alpha} (\phi(y)) \tag{5.18}$$

L'opérateur $h\star$ est linéaire, il peut donc être représenté par une matrice H de taille $n \times n$. On peut montrer (voir [Ng *et al.*, 2000] par exemple) que la FFT bidimensionnelle diagonalise H si $\star$ représente la convolution avec des conditions de bord périodiques, et que la DCT bidimensionnelle diagonalise H si h est symétrique et que $\star$ représente la convolution avec des conditions de bord de Neumann. Dans les deux cas, on voit que H s'écrit sous la forme $H = A^{-1}\lambda A$, où λ est une matrice diagonale et A est soit la FFT, soit la DCT. Comme ces deux transformations sont des isométries de Y dans Y, on a :

$$|h \star y - y_0|_2 = |Hy - y_0|_2 = |\lambda Ay - Ay_0|_2 \tag{5.19}$$

Finalement le problème de la déconvolution en présence de bruit gaussien peut s'écrire sous la forme :

$$\inf_{y \in Y, |\lambda Ay - Ay_0|_2 \leq \alpha} (\phi(y)) \tag{5.20}$$

La figure 5.6 montre un exemple de résultat de déconvolution utilisant la variation totale. Pour résoudre ce problème, nous avons utilisé l'algorithme du paragraphe 4.3.1. Nous détaillerons son principe dans la partie suivante.

– D'après le théorème de Shannon, on peut penser que la meilleure manière d'agrandir une image consiste à utiliser une technique de zero-padding. Malheureusement, ce procédé introduit des oscillations près des bords. Un moyen simple de les éviter consiste à résoudre le problème suivant :

$$\inf_{y \in Y, |\lambda(Ay - f)|_2 \leq \alpha} (\phi(y)) \tag{5.21}$$

Dans ce cas :

– f_0 est la transformée de Fourier de l'image réduite qu'on a agrandie en ajoutant des coefficients nuls dans les hautes fréquences.
– A est une transformée de Fourier et λ est une matrice diagonale telle que $\lambda_i = \infty$ aux endroits où f_i est connu et $\lambda_i = 0$ sinon.

Ce problème est un cas particulier d'une classe de techniques d'agrandissement proposée par François Malgouyres [Malgouyres et Guichard, 2002].

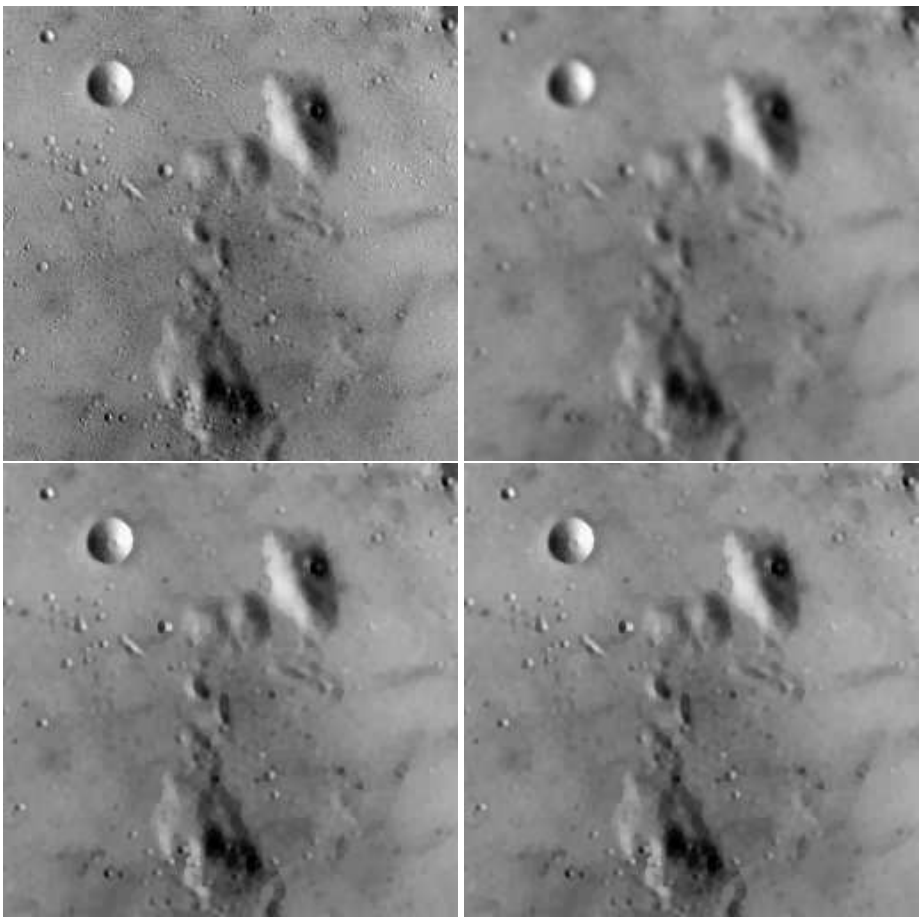

FIG. 5.6 – Déconvolution d'images - HG : image originale - HD : image convoluée bruitée - BG : Solution du problème (5.20) en utilisant la variation totale régularisée - BD : solution du problème (5.20) en utilisant la variation totale.

5.2.3 Résumé

Nous résumons les applications citées précédemment dans le tableau ci-dessous. Nous résolvons le problème suivant :

$$\min_{y \in F^{-1}(y_0)} (\phi(y)) \quad \text{avec} \quad F^{-1}(y_0) = \{y \in Y, |\lambda(Ay - y_0)|_{l^p(Y)}\} \tag{5.22}$$

	$p = 1$	$p = 2$	p=∞
A =Identité	débruitage bruit impulsionnel, décomposition d'images [Alliney, 1997; Nikolova, 2004; Chan et Esedoglu, 2005; Darbon et Sigelle, 2006]	débruitage de bruits gaussiens [Rudin *et al.*, 1992a]	débruitage de bruits bornés [Weiss *et al.*, 2006]
A = Transformée de Fourier	déconvolution robuste [Fu *et al.*, 2005]	Déconvolution d'images [Rudin et Osher, 1994; Chan *et al.*, 1999; Bertalmio *et al.*, 2003], Aggrandissement d'images [Malgouyres et Guichard, 2002]	Pas de reférence connue
A = Transformée en ondelettes ou en cosinus locale	Décomposition d'images ou débruitage [Durand et Nikolova, 2007]	débruitage d'images (pas de référence connue)	Débruitage de bruits de compression [Alter *et al.*, 2005; Tramini *et al.*, 1998; Coifman et Sowa, 2000]

5.3 Minimisation sous une contrainte générale

Dans cette section, nous allons montrer comment appliquer les algorithmes du chapitre 4 pour résoudre les différents problèmes cités précédemment. L'application de ces algorithmes n'est pas toujours triviale, et nous pensons que les exemples donnés dans cette partie peuvent aider à l'utilisation des algorithmes multi-pas. Nous donnons aussi plusieurs commentaires sur le choix pratique des paramètres des algorithmes tels que le paramètre de régularisation et le nombre d'itérations. Nous traitons le cas de la restauration avec a priori de régularité et a priori de parcimonie de façon séparée. Nous résolvons plusieurs problèmes originaux tels que le problème de la reconstruction robuste avec échantillonage irrégulier et convolution. Nous finissons par comparer les problèmes de restauration avec critères de parcimonie et de régularité en terme de complexité numérique.

5.3.1 Restauration avec critère de régularité

Version contrainte

On a vu que lorsqu'on utilise un a priori de régularité, les problèmes de restauration peuvent souvent s'écrire sous la forme :

$$\min_{x\in X}\left(|Bx|_{l^1(Y)}+\chi_Q(x)\right) \tag{5.23}$$

où $B : X \to Y$ est une transformée linéaire, $Q = \{x \in X, |\lambda(Ax - x_0)|_{l^p(X)} \leq \alpha\}$, A est une transformée linéaire inversible et $p \in \{1, 2, \infty\}$. Ce problème est convexe et non différentiable. Pour le résoudre, il faut analyser les propriétés des différentes fonctions :

– La fonction $x \mapsto |Bx|_{l^1(Y)}$ peut être vue comme une fonction de type max. On a en effet :

$$|Bx|_{l^1(Y)} = \max_{y\in Y}\left(\langle Bx, y\rangle_Y - \chi_K(y)\right) \tag{5.24}$$

avec $K = \{y \in Y, |y|_{l^\infty(Y)} \leq 1\}$. Notons que χ_K est une fonction de domaine borné, on peut donc régulariser le terme $|Bx|_1$ (voir lemme 2.8).

– La contrainte dans l'équation (5.23) peut elle-aussi s'exprimer sous la forme d'une fonction max. En effet, soit $\|\cdot\|$ une norme sur Y et $\|\cdot\|_*$ sa norme duale associée ($\|y\|_* = \max_{x,\|x\|\leq 1}(\langle y, x\rangle_Y)$). On peut écrire :

$$\begin{aligned}
&\max_{y\in Y}\left(\langle\lambda(Ax - x_0), y\rangle_Y - \alpha\|y\|\right) && (5.25)\\
=\ &\max_{r\geq 0}\left(\max_{y\in Y,\|y\|\leq r}\left(\langle\lambda(Ax - x_0), y\rangle_Y\right) - \alpha r\right) && (5.26)\\
=\ &\max_{r\geq 0}\left(r\|\lambda(Ax - x_0)\|_* - \alpha r\right) && (5.27)\\
=\ &\chi_Q(x) && (5.28)
\end{aligned}$$

avec $Q = \{x \in X, \|\lambda(Ax - x_0)\|_* \leq \alpha\}$. Cependant le domaine de la fonction $y \mapsto \|y\|$ n'est pas borné et on ne peut donc pas régulariser cette fonction avec la technique de Moreau-Yosida (on ne peut pas assurer une erreur d'approximation finie). On ne peut pas non plus considérer que χ_Q est une fonction simple car on ne sait pas en général faire de projection sur Q.

Aucun des algorithmes présentés dans le chapitre 4 ne semble donc applicable. Pour remédier à ce problème, on propose de faire le changement de variable : $z = Ax - x_0$. Le problème (5.23) devient alors :

$$\min_{z\in X,|\lambda z|_{l^p(X)}\leq\alpha}\left(|BA^{-1}(z + x_0)|_{l^1(Y)}\right) \tag{5.29}$$

Ainsi la fonction $z \mapsto |BA^{-1}(z + x_0)|_{l^1 Y}$ est encore fonction max, et cette fois, la contrainte peut être exprimée sous la forme de la fonction indicatrice d'un ensemble sur lequel on sait faire des projections. On propose en annexe A.2 des algorithmes rapides et exacts de projections sur des boules l^p pondérées. Le problème (5.29) peut donc être vu comme la somme d'une fonction max et d'une fonction simple. On peut résoudre ce problème avec un taux de convergence en $O\left(\frac{1}{k}\right)$ en utilisant la méthode proposée en 4.3. Détaillons l'algorithme.

Régularisation de la norme l^1 Notons $f(x) = |BA^{-1}(x+x_0)|_{l^1(Y)}$. La première étape de l'algorithme consiste à calculer une approximation différentiable de f. La stratégie la plus largement répandue consiste à utiliser :

$$\tilde{f}_\mu(x) = \sum_{i=1}^{n} \sqrt{(BA^{-1}(x+x_0))_i + \mu^2} \tag{5.30}$$

On va voir que l'approximation de Moreau-Yosida (voir lemme (2.8)) possède des propriétés plus satisfaisantes. On définit :

$$f_\mu(x) = \max_{y \in Y, |y|_{l^\infty(Y)} \leq 1} \left(\langle BA^{-1}(x+x_0), y \rangle_Y - \frac{\mu}{2}|y|_Y^2 \right) + \frac{\mu}{2} \tag{5.31}$$

On montre facilement que $f_\mu(u) = \sum_{i=1}^{n} \psi_\mu(|(BA^{-1}(x+x_0))_i|)$ avec $\psi_\mu : \mathbb{R} \to \mathbb{R}$ définie comme suit :

$$\psi_\mu(x) = \begin{cases} |x| & \text{si } |x| \geq \mu \\ \frac{x^2}{2\mu} + \frac{\mu}{2} & \text{sinon} \end{cases} \tag{5.32}$$

ψ_μ est connue sous le nom de fonction de Huber. L'approximation f_μ (5.31) semble plus appropriée que l'approximation $\tilde{f}_\mu$ (5.30), car toutes les deux sont $\frac{\|BA^{-1}\|^2}{\mu}$-Lipschitz différentiables [2]. Mais on a pour tout $x \in X$:

$$0 \leq \tilde{f}_\mu(x) - f(x) \leq n\mu \tag{5.33}$$

tandis que :

$$0 \leq f_\mu(x) - f(x) \leq \frac{n\mu}{2} \tag{5.34}$$

L'approximation f_μ semble donc être deux fois meilleure vis-à-vis des schémas de premier ordre. Notons que l'approximation (5.30) a été initialement proposée pour être minimisée par des schémas de second ordre [Vogel et Oman, 1996] qui nécessitent des propriétés de régularité allant jusqu'au second ordre. Dans toutes les expériences numériques que nous avons menées, la minimisation de (5.32) mène à des solutions qui ont une norme l^1 plus faible que (5.30). Leur aspect visuel est cependant très similaire. La complexité numérique du calcul de ∇f_μ et de $\nabla \tilde{f}_\mu$ est identique. Nous pensons donc que l'approximation f_μ doit être choisie préférentiellement en vue de l'utilisation de schémas de premier ordre. Notons finalement que :

$$\nabla f_\mu(x) = A^{-*}B^*\Psi \quad \text{avec } \Psi_i = \begin{cases} \frac{(BA^{-1}(x+x_0))_i}{|(BA^{-1}(x+x_0))_i|} & \text{si } |(BA^{-1}(x+x_0))_i| \geq \mu \\ \frac{(BA^{-1}(x+x_0))_i}{\mu} & \text{sinon} \end{cases} \tag{5.35}$$

Comment choisir le paramètre de régularisation ? Nous pouvons maintenant utiliser l'algorithme 4.3 pour résoudre le problème suivant :

$$z^* = \arg\min_{z \in K} f_\mu(z) \tag{5.36}$$

[2] Rappelons que la constante de Lipschitz détermine le taux de convergence des schémas de minimisation du premier ordre.

Il renvoie une estimée z^N de (5.36). Le théorème 4.22 indique comment choisir le paramètre de régularisation μ en fonction du nombre d'itérations (ou vice-versa). Il ne reste plus qu'à poser $x^N = A^{-1}(z^n + x_0)$ pour obtenir une estimée de (5.23).

En pratique, on souhaite généralement utiliser le paramètre μ qui assure la meilleure qualité d'image. On peut observer que c'est rarement le cas $\mu = 0$ qui fournit les meilleurs résultats. La minimisation de la norme l^1 mène souvent à des solutions x^* telles que Bx^* contient beaucoup de 0. Par exemple, lorsque $B = \nabla$, on minimise la variation totale. Un 0 de Bx^* signifie que l'image est localement constante. M. Nikolova montre dans [Nikolova, 2001] que cet effet est inévitable : la singularité en 0 de la norme l^1 favorise des solutions telles que Bx^* soit creux.

Un moyen d'éviter ce problème est d'utiliser un opérateur régularisé tel que (5.31). En pratique, pour la restauration d'images, ce choix mène à de meilleurs RSB et des résultats plus satisfaisants visuellement. La figure 5.7 illustre ce fait. L'image Lena est dégradée par un bruit blanc gaussien additif de variance variable. Elle est ensuite restaurée en utilisant divers paramètres de régularisation μ. On voit que la valeur optimale de μ est autour de 0.002, et ce indépendamment du SNR initial.

Pour des applications de restauration d'images naturelles (d'amplitude comprise entre 0 et 1), nos expériences nous ont mené à la conclusion que la valeur de μ optimale devrait être choisie dans l'intervalle $[0.001, 0.005]$. Peu de différences visuelles sont observées dans cet intervalle.

Dans certaines situations on souhaite tout de même minimiser exactement la norme l^1 ($\mu = 0$). Expérimentalement, pour des images normalisées dans l'intervalle $[0, 1]$, on remarque que la solution de (5.36) en choisissant $\mu \leq 0.005$ est identique perceptuellement à la solution non régularisée. D'après cette remarque on peut inférer qu'une précision suffisante à atteindre est :

$$f(z^N) - f(z^*) \leq 0.005 \times \frac{n}{2} \tag{5.37}$$

En utilisant le théorème (4.22), pour obtenir une précision ϵ, il ne faut pas plus de $\frac{\sqrt{D}\|A^{-1}B\| \cdot |x_\mu^0 - x_\mu^*|_X}{\sqrt{l}}$ itérations. Dans notre cas, $l = 1$, D est le carré du rayon de l'ensemble des contraintes, et $|x_\mu^0 - x_\mu^*|_X \leq \sqrt{D}$. Dans tous les problèmes que l'on a cités précédemment (excepté la déconvolution), on a $D \sim \theta n$ avec $\theta \in [0, 1[$. Pour obtenir une solution de précision ϵ, il ne faut donc pas plus de $\frac{n\|A^{-1}B\|}{\epsilon}$ itérations. Pour obtenir la précision (5.37), il suffit donc de $\frac{\|A\|}{0.0025}$. Lorsque A est une isométrie, $\|A^{-1}B\| = \|B\|$. Le nombre d'itérations théoriques pour obtenir une précision suffisante est donc inférieure à $400\|B\|$. Pour la variation totale, ce nombre est de l'ordre de 1000 itérations.
Le taux de convergence théorique indique donc que les méthodes proposées convergent vers une solution visuellement satisfaisante en un nombre d'itérations raisonnable. En pratique, le nombre d'itérations nécessaires est de l'ordre de 100.

Version lagrangienne

Quelques fois on préfère au problème (5.23) sa version lagrangienne :

$$\min_{x \in X} (|Bx|_{l^1 Y} + |\lambda(Ax - x_0)|_{l^p(X)}) \tag{5.38}$$

le théorème 2.7 indique que les deux problèmes ont des solutions identiques pour un certain choix du paramètre λ. La fonction à minimiser est la somme de deux fonctions max. Pour le résoudre, on peut donc régulariser ces deux fonctions puis minimiser les fonctions régularisées avec l'algorithme rapide 4.1. Une autre solution consiste à ajouter une fonction fortement convexe du type $\frac{\epsilon}{2}|x - x_0|^2$

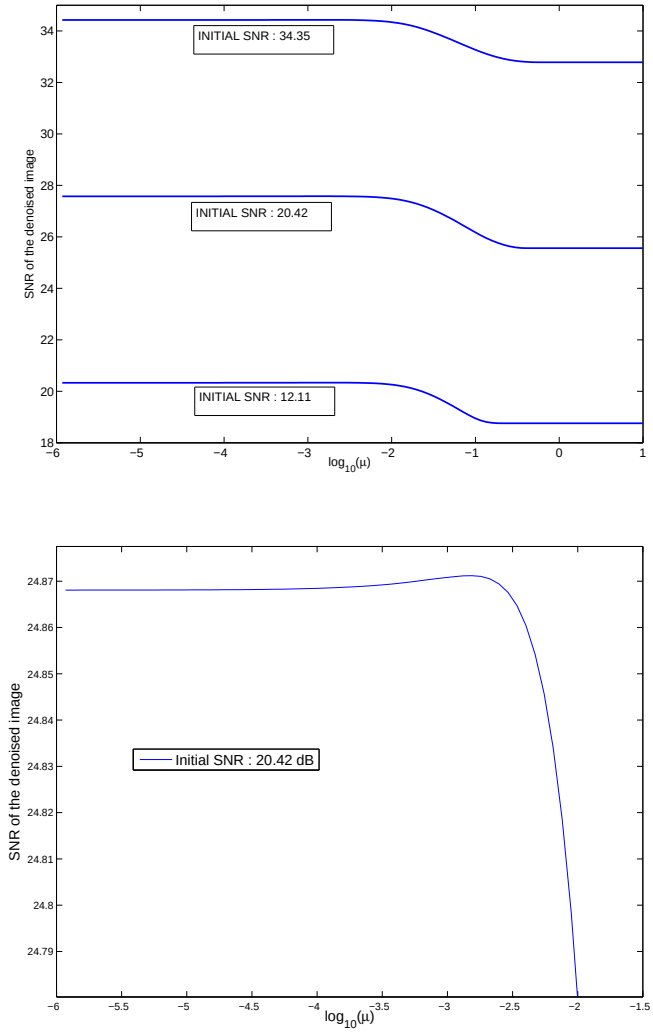

FIG. 5.7 – SNR de l'image débruitée en fonction du paramètre de régularisation μ. Haut : résultats pour 3 SNR initiaux différents. Bas : détail d'une courbe.

puis à résoudre un problème dual. Dans les deux cas, on obtient un taux de convergence en $O\left(\frac{1}{k}\right)$. Nous montrons ci-dessous comment utiliser ces techniques sur un exemple simple : la résolution du problème $BV - l^1$. La généralisation au problème général (5.38) peut se faire sans grandes difficultés. Le problème $BV - l^1$ s'écrit comme suit :

$$\min_{x\in X}\left(|\nabla x|_{l^1(Y)} + \lambda|x - x_0|_{l^1(X)}\right) \tag{5.39}$$

Dans ce cas, $\nabla : X \to Y$ est une dicrétisation du gradient. Le terme $\phi_{0,0}(x) = |\nabla x|_{l^1(Y)} + \lambda|x - x_0|_{l^1(X)}$ peut être réécrit sous la forme suivante :

$$\phi_{0,0}(x) = \max_{z\in K}\left(\langle Cx - F, z\rangle_Z\right) \tag{5.40}$$

où $Z = Y \times X$, $C = \begin{bmatrix} \nabla \\ \lambda I \end{bmatrix}$, $F = \begin{bmatrix} 0 \\ \lambda x_0 \end{bmatrix}$ et $K = \left\{z \in Z, z = \begin{bmatrix} y \\ x \end{bmatrix},\ |x|_{l^\infty(X)} \leq 1,\ |y|_{l^\infty(Y)} \leq 1\right\}$.

Pour résoudre (5.39), on se propose de le régulariser de la façon suivante :

$$\min_{x\in X}\left(\underbrace{\max_{z\in K}\left(\langle Cx - F, z\rangle_Z - \frac{\mu}{2}|z|_Z^2\right) + \frac{\epsilon}{2}|x - x_0|_X^2}_{\phi_{\mu,\epsilon}}\right) \tag{5.41}$$

le paramètre μ permet de régulariser la fonction primale. Le paramètre ϵ permet de régulariser la fonction duale.

Résolution du problème primal Le problème primal ainsi régularisé devient :

$$\min_{x\in X}\left(|\nabla x|_{1,\mu} + \lambda|x - x_0|_{1,\mu} + \frac{\epsilon}{2}|x - x_0|_X^2\right) \tag{5.42}$$

où $|\cdot|_{1,\mu}$ est la fonction de Huber de paramètre μ définie à partir la fonction (5.32). Pour le résoudre, on peut utiliser l'algorithme 4.2 en posant $f(x) = |\nabla x|_{1,\mu} + \lambda|x - x_0|_{1,\mu}$ et $\Psi(x) = \frac{\epsilon}{2}|x - x_0|_X^2$.

La fonction f a une constante de Lipschitz $L \leq \frac{\|C\|^2}{\mu} \leq \frac{\|\nabla\|^2+\lambda^2}{\mu}$. La fonction Ψ est fortement convexe de paramètre de forte convexité $l = \epsilon$. Si on souhaite utiliser l'algorithme 4.2, on peut choisir $\epsilon = 0$ et $\mu > 0$ dépendant de la précision souhaitée.

Résolution du problème dual On a :

$$\min_{x\in X}\left(\max_{z\in K}\left(\langle Cx - F, z\rangle_Z - \frac{\mu}{2}|z|_Z^2\right) + \frac{\epsilon}{2}|x - x_0|_X^2\right) \tag{5.43}$$

$$= \max_{z\in K}\left(\min_{x\in X}\left(\langle Cx - F, z\rangle_Z + \frac{\epsilon}{2}|x - x_0|_X^2\right) - \frac{\mu}{2}|z|_Z^2\right) \tag{5.44}$$

$$= -\min_{z\in K}\left(\underbrace{\frac{1}{2\epsilon}|C^*z|_X^2 - \langle f, C^*z\rangle_Z + \langle F, z\rangle_Z}_{-\hat{f}(z)} + \frac{\mu}{2}|z^2|_Z\right) \tag{5.45}$$

Pour résoudre (5.45) on peut utiliser l'algorithme 4.4 qui renvoie une estimée $\overline{x}^N$ de la solution de (5.42). La fonction $\hat{f}(z)$ a un gradient Lipschitz de constante $L \leq \frac{\|C\|^2}{\epsilon} \leq \frac{\|\nabla\|^2+\lambda^2}{\epsilon}$. La fonction $\Psi(z) = \frac{\mu}{2}|z^2|_Z + \chi_K(z)$ est fortement convexe de paramètre de convexité $l = \mu$.

Comparaison numérique La figure 5.9 montre le résultat d'une restauration d'image entâchée de bruit poivre et sel avec cet algorithme. La figure 5.8 montre la façon dont converge le schéma primal et le schéma dual en choisissant des paramètres d'approximation μ et ϵ très petits. Ce graphique montre plusieurs propriétés intéressantes.

- Au bout de 1000 itérations, le schéma accéléré de Nesterov appliqué au problème dual régularisé permet d'obtenir $\phi_{0,0}(x^k) - \phi_{0,0}(x^*) \leq 10^{-4}(\phi_{0,0}(x^0) - \phi_{0,0}(x^*))$ malgré l'erreur d'approximation. Une telle précision est comparable à ce que l'on peut obtenir avec des méthodes d'ordre plus élevé.
- Sur cet exemple, les méthodes accélérées ont un comportement radicalement plus rapide qu'une simple descente de gradient. Plusieurs expériences confirmeront ce fait dans le chapitre 7. Il est particulièrement clair ici. Comme on a choisi $\mu = 10^{-4}$, le pas dans la descente de gradient est lui aussi très petit (de l'ordre de 10^{-4}), ce qui explique le très mauvais comportement de cette méthode. Lors de nos expériences numériques, on a montré qu'il faut 10^6 itérations à la descente de gradient pour atteindre la précision du schéma de Nesterov au bout de 1000 itérations.
- Pour rendre la comparaison des méthodes primales et duales plus sensée, nous avons choisi μ et ϵ de telle façon que la précision asymptotique soit identique (i.e. que $|\phi_{\mu,0}(x^*_\mu) - \phi_{0,0}(x*)| = |\phi_{0,\lambda}(x^*_\lambda) - \phi_{0,0}(x*)|$). La méthode de régularisation du problème primal semble ici plus lente que la méthode de régularisation du problème dual. Nous avons observé ceci sur plusieurs problèmes.

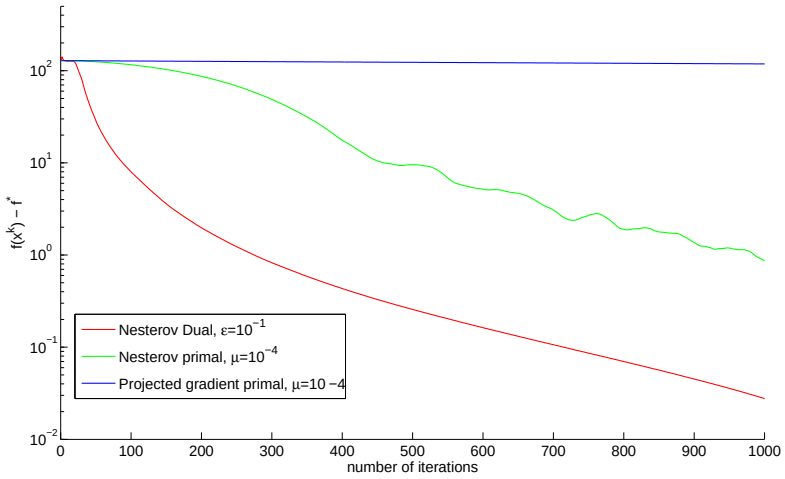

FIG. 5.8 – Comparaison de la vitesse de convergence des approche primales et duales avec descente de gradient ou schémas accélérés. (Note : dans la figure, on a posé $f(x^k) = \phi_{0,0}(x^k)$ et $f^* = \phi_{0,0}(x^*)$)

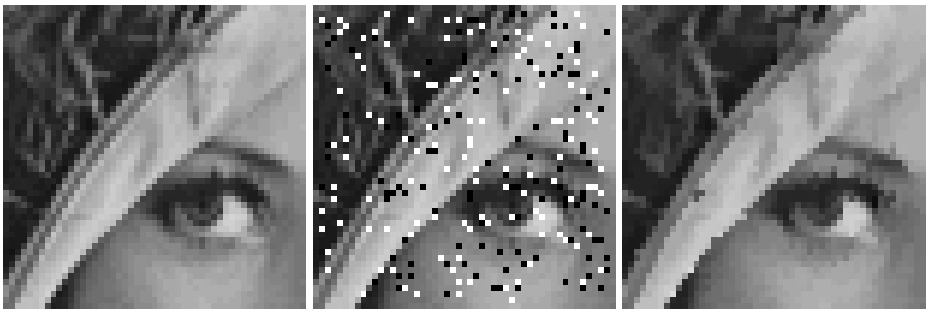

FIG. 5.9 – De gauche à droite : image originale, image avec bruit poivre et sel, image restaurée par modèle $BV - l^1$.

Echantillonnage irrégulier et déconvolution robuste Pour conclure cette partie, nous donnons l'exemple de la déconvolution robuste d'une image avec échantillonnage irrégulier. Ces expériences ont été menées par Mikael Carlavan, stagiaire du projet ARIANA. Nous renvoyons le lecteur intéressé par les détails du modèle aux articles [Almansa *et al.*, 2006; Bughin *et al.*, 2008; Carlavan *et al.*, 2009]. Il consiste à résoudre un problème du type :

$$\min_{x \in X} \left(|Bx|_{l^1 Y} + |\lambda(Ax - x_0)|_{l^1(X)} \right) \tag{5.46}$$

avec un algorithme similaire à ceux que l'on vient de présenter. Les auteurs de [Almansa *et al.*, 2006] utilisent une norme l^2 comme attache aux données, nous l'avons remplaçée par une norme l^1 pour être plus robuste à des erreurs de recalage intervenant dans ce problème. Mikael Carlavan a comparé deux opérateurs B : le gradient et la transformée en ondelettes complexes de N. Kingsbury [Kingsbury, 1998]. Les résultats obtenus sont présentés sur la figure 5.10.

On voit que les résultats obtenus avec une régularisation par ondelettes complexes sont meilleurs : ils préservent mieux les textures. La variation totale préserve un petit mieux les contours. En termes de SNR, la méthode de régularisation par ondelettes complexes est meilleure.

5.3.2 Restauration avec critère de parcimonie

On a vu en (5.11) que lorsqu'on souhaite utiliser des a priori de parcimonie pour résoudre les problèmes de restauration, on est ramené à la résolution d'un problème du type :

$$y^* = \underset{y \in Y, |\lambda(ABy - x_0)|_{l^p(X)} \leq \alpha}{\arg\inf} \left(|y|_{l^1(Y)} \right) \tag{5.47}$$

La fonction $|y|_{l^1(Y)}$ peut être vue comme une fonction simple. Malheureusement, la projection sur l'ensemble $\{y \in Y, |\lambda(ABy - x_0)|_{l^p(X)} \leq \alpha\}$ est rarement calculable exactement si B est une transformée redondante, et les algorithmes présentés ne sont donc pas applicables. On peut par contre résoudre le problème lagrangien :

$$y^* = \underset{y \in Y}{\arg\inf} \left(|y|_{l^1(Y)} + |\lambda(ABy - x_0)|_{l^p(X)} \right) \tag{5.48}$$

Pour ce problème, on peut régulariser la fonction $|\lambda(ABy - x_0)|_{l^p(X)}$, considérer la fonction $|y|_{l^1(Y)}$ comme une fonction simple puis appliquer l'algorithme 4.2 au problème régularisé. On obtient alors

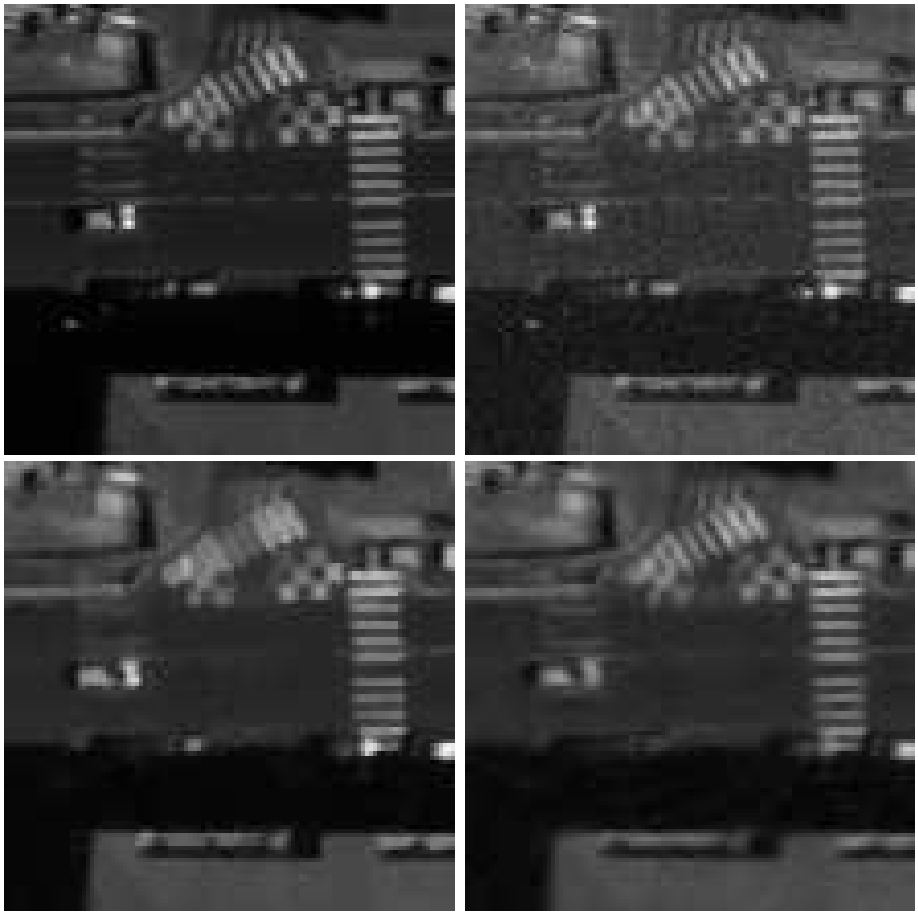

FIG. 5.10 – HG : image originale - HD : image à restaurer (SNR : 15.6 dB) - BG : image restaurée avec la variation totale (SNR : 24.1dB) - BD : image restaurée avec ondelettes complexes (SNR : 24.4dB)

un taux de convergence en $O\left(\frac{1}{k}\right)$.

5.4 Minimisation avec attache aux données l^2

Dans cette partie, on s'intéresse à la minimisation d'une norme l^1 de transformée affine avec une attache aux données de type l^2. De tels problèmes arrivent typiquement lorsque l'on souhaite restaurer des images entachées de bruits blancs gaussiens. Le principe du maximum a posteriori mène alors à des modèles de restauration du type suivant (a priori de régularité) :

$$\min_{y\in Y}\left(|By|_{l^1(X)} + |\lambda(Ay - y_0)|_X^2\right) \tag{5.49}$$

ou du type suivant (a priori de parcimonie) :

$$\min_{x\in X}\left(|x|_{l^1(X)} + |\lambda(AB^*x - y_0|_X^2\right) \tag{5.50}$$

On détaille ci-dessous les méthodes pour résoudre ces problèmes.

5.4.1 Restauration avec critère de régularité

Notons $\varphi(y) = |\lambda(Ay - y_0)|_X^2$. Lorsque A est inversible (i.e. sa plus petite valeur propre est strictement positive), φ est fortement convexe comme le montre la proposition suivante :

Proposition 5.3. *Soit $\lambda = diag(\lambda_i)$ une matrice diagonale avec $\lambda_i \in]0,\infty]$. On pose $\lambda_- = \min_i(\lambda_i)$ et $\lambda_-(A)$ la plus petite valeur propre de A. Alors φ est l-fortement convexe avec $l \geq 4\lambda_-^2\lambda_-(A)^2$.*

Démonstration. φ est différentiable avec $\nabla\varphi(y) = 2A^*\lambda^2(Ay - y_0)$. On a donc pour tout $(y_1, y_2) \in Y$:

$$|\nabla\varphi(y_1) - \nabla\varphi(y_2)|_2 = 2|A^*\lambda^2A(y_1 - y_2)|_2 \geq 2\lambda_-^2\lambda_-^2(A)|y_1 - y_2|_2 \tag{5.51}$$

□

On peut maintenant formuler le problème adjoint à (5.49) en utilisant la dualité min-max :

$$\begin{aligned}
& \min_{y\in Y}\left(|By|_{l^1X} + |\lambda(Ay - y_0)|_Y^2\right) && (5.52)\\
= \ & \min_{y\in Y}\left(\max_{x\in K}\left(\langle By, x\rangle_X\right) + |\lambda(Ay - y_0)|_Y^2\right) && (5.53)\\
= \ & \max_{x\in K}\left(\min_{y\in Y}\left(\langle y, B^*x\rangle_Y + |\lambda(Ay - y_0)|_Y^2\right)\right) && (5.54)\\
= \ & \max_{x\in K}\left(\langle A^{-1}y_0, B^*x\rangle_Y - \frac{1}{4}|\lambda^{-*}A^{-*}B^*x|_Y^2\right) && (5.55)
\end{aligned}$$

Dans ces équations $K = \{x \in X, |x|_{l^\infty(X)} \leq \alpha\}$. On peut résoudre le problème différentiable (5.55) avec la version 4.2 de l'algorithme de Y. Nesterov si B est une trame (i.e. $\forall y \in Y$, $|By|_X \geq \epsilon|y|_Y$ avec $\epsilon > 0$). Dans ce cas on obtient un taux de convergence linéaire. Si B a un noyau non réduit à 0, on peut utiliser la version 4.3 de l'algorithme accéléré et on obtient un taux de convergence en $O\left(\frac{1}{k^2}\right)$ d'après le théorème 4.21.

Pour finir, donnons l'exemple de l'implémentation du problème de Rudin-Osher-Fatemi avec cette technique. On a déjà vu en (4.18) que :

$$\min_{y\in Y}\left(|\nabla y|_{l^1(X)}+\lambda|y-y_0|_Y^2\right)=\max_{x\in K}\left(-\frac{|\text{div } x|_Y^2}{4\lambda}+\langle y_0,-\text{div } x\rangle_Y\right) \tag{5.56}$$

La partie gauche de (5.56) est le problème de Rudin-Osher-Fatemi avec pour donnée initiale y_0 [3]. Nous notons sa solution y^*. Le problème de droite est son problème adjoint. Nous notons sa solution x^*. En reprenant les notations du paragraphe 4.2 on a :

$$\phi(y)=|\nabla y|_{l^1(X)}+\lambda|y-y_0|_Y^2 \tag{5.57}$$

$$\hat{f}(x)=-\frac{|\text{div } x|_Y^2}{4\lambda}+\langle y_0,-\text{div } x\rangle_Y \tag{5.58}$$

$$\nabla\hat{f}(x)=\frac{1}{2\lambda}\nabla\text{div } x+\nabla y_0 \tag{5.59}$$

La constante de Lipschitz de $\nabla\hat{f}$ est $L\leq\frac{\|\text{div }\|^2}{2\lambda}$. L'algorithme 4.3 appliqué au problème adjoint devient :

Algorithme 5.1: Descente de gradient accélérée pour le problème de Rudin-Osher-Fatemi

Entrée : Le nombre d'itérations N.
Sortie : x^N une estimée de x^*.
début
  Poser $k=-1$
  Poser $g^{-1}=0$, $u^{-1}=0$
  Poser $x^0\in K$ (une estimée précise de x^* si possible)
  Poser $L=\frac{\|\text{div }\|^2}{2\lambda}$
  pour *k allant de* 0 *à* N **faire**
    Calculer $\eta^k=\nabla\left(y_0+\frac{1}{2\lambda}\text{div } x^k\right)$
    Calculer $y^k=\Pi_K\left(x^k+\frac{\eta^k}{L}\right)$
    Poser $g^k=g^{k-1}+\frac{k+1}{2}\eta^k$
    Calculer $v^k=\Pi_K\left(x^0+\frac{g^k}{L}\right)$
    Poser $x^{k+1}=\frac{2}{k+3}v^k+\frac{k+1}{k+2}y^k$
    Poser $u^k=u^{k-1}+(i+1)\left(\frac{\text{div } y^k}{2\lambda}+y_0\right)$
  fin
  Poser $u^N=\frac{2u^k}{(N+1)(N+2)}$.
fin

En utilisant le théorème 4.21, on obtient comme résultat de convergence :

[3] Nous avons interverti X et Y pour ne pas avoir à adapter les notations des algorithmes.

$$\phi(u^N) - \phi(y^*) \leq \frac{|x^0 - x^*|_X^2 \|\text{div}\|^2}{\lambda k^2} \leq \frac{n\|\text{div}\|^2}{\lambda k^2} \tag{5.60}$$

De plus on a :

$$|u^N - y^*|_Y^2 \leq \frac{n\|\text{div}\|^2}{\lambda^2 k^2} \tag{5.61}$$

Notons que trouver le taux de convergence d'un algorithme de résolution du problème de Rudin-Osher-Fatemi avec des méthodes de premier ordre était jusqu'à présent un problème ouvert. Notons aussi que les méthodes de premier ordre semblent pour le moment plus efficaces que les méthodes de second ordre pour résoudre ce problème. Dans de nombreux schémas récents, la résolution de (5.56) (ou d'un problème de structure équivalente) n'est qu'un problème auxiliaire qui doit être effectué plusieurs fois [Bect *et al.*, 2004; Figueiredo et Nowak, 2003; Daubechies *et al.*, 2004; Aujol et Chambolle, 2005; Aujol *et al.*, 2006; Huang *et al.*, 2008]. Jusqu'à présent, les approches de premier ordre sont toujours favorisées.

5.4.2 Restauration avec critère de parcimonie

La résolution du problème (5.50) est directe. Il suffit en effet d'utiliser le schéma présenté dans la partie 4.1.1. Avec les notations de cette partie, on pose $\Psi(x) = |x|_{l^1(Y)}$ (la fonction simple) et $f(x) = |\lambda(AB^*x - y_0)|_X^2$ (la fonction Lipschitz-différentiable). On a :

$$\nabla f(x) = 2BA^*\lambda^2(AB^*x - y_0) \tag{5.62}$$

Ce gradient est Lipschitz. Notons $\lambda_+ = \max_i \lambda_i$ et $\lambda_- = \min_i \lambda_i$. La constante de Lipschitz de ∇f satisfait alors :

$$L \leq \lambda_+^2 \|B\|^2 \|A\|^2 \tag{5.63}$$

De plus, si B est une trame ($\forall x \in X,\ \|Bx\| \geq \epsilon|x|_X$), alors f est fortement convexe de paramètre de forte convexité $l \geq \lambda_-^2 \epsilon^2 \lambda_+(A)^2$.

On obtient donc un taux de convergence linéaire lorsque B est une trame et un taux de convergence en $O\left(\frac{1}{k^2}\right)$ dans le cas général.

5.5 Comparaison algorithmiques des deux a priori

On a présenté dans cette thèse deux types d'a priori qu'on a appelé a priori de régularisation et a priori de parcimonie. Il est naturel de se demander lequel des deux est le meilleur. Pour comparer les deux approches, plusieurs critères entrent en jeu : les temps de calcul, la qualité de la restauration, les propriétés analytiques des solutions. Nous ne traitons ici que le premier point. Les deux autres peuvent former l'objet d'une étude future.

5.5.1 Comparaison pour une attache aux données générale

On a vu que les problèmes de restauration avec a priori de régularité peuvent s'écrire sous la forme suivante :

$$\min_{\{x\in X\}} \left(|Bx|_{l^1(Y)} + |\lambda(Ax - x_0)|_{l^p(X)} \right) \tag{5.64}$$

avec $B : X \to Y$. Les problèmes de restauration avec a priori de parcimonie peuvent s'écrire sous la forme suivante :

$$\min_{y\in Y} \left(|y|_{l^1(Y)} + |\lambda(AB^*y - x_0)|_{l^p(X)} \right) \tag{5.65}$$

Les deux problèmes peuvent être résolus avec le même taux de convergence théorique en $O\left(\frac{1}{k}\right)$. On peut donc considérer qu'ils ont la même complexité numérique.

5.5.2 Comparaison pour une attache aux données l^2

Lorsque l'on souhaite résoudre un problème de restauration d'image entâchée de bruit gaussien, le principe du maximum a posteriori mène à un problème du type :

$$\min_{x\in X} \left(|Bx|_{l^1(Y)} + \lambda |x - x_0|_X^2 \right) \tag{5.66}$$

en utilisant un a priori de régularité. Lorsqu'on utilise un a priori de parcimonie, on obtient un problème du type :

$$\min_{y\in Y} \left(|x|_{l^1(Y)} + \lambda |B^*y - x_0|_X^2 \right) \tag{5.67}$$

où B est un dictionnaire ou une trame. On a vu que les problèmes (5.66) et (5.67) peuvent tous deux être résolus en $O\left(\frac{1}{k^2}\right)$ lorsque B a un noyau non réduit à 0. Lorsque B est une trame, les deux problèmes peuvent être résolus avec un taux de convergence linéaire identique dépendant seulement de la plus petite et de la plus grande valeur singulière de B^*.

Les a priori de régularité et de parcimonie mènent donc à des problèmes de complexité algorithmique équivalente.

5.5.3 Une comparaison préliminaire de la qualité de restauration

Pour finir ce chapitre, nous proposons une comparaison préliminaire de la qualité de restauration avec des critères de régularité ou de parcimonie. Nous bruitons une image avec un bruit gaussien additif et résolvons les problèmes (5.66) et (5.67). Cette comparaison n'est qu'indicative, car nous avons choisi une transformée très simple et pas assez redondante. Nous avons posé $B = \begin{bmatrix} \tau_{0,0}A \\ \tau_{1,0}A \\ \tau_{0,1}A \\ \tau_{1,1}A \end{bmatrix}$ où A est une transformée décimée de Haar à 5 niveaux, et où l'opérateur $\tau_{i,j}$ est un opérateur de translation d'une distance i suivant l'axe x et d'une distance j suivant l'axe y. Ainsi, l'opérateur $B : \mathbb{R}^n \to \mathbb{R}^{4n}$ est invariant par translation jusqu'au second niveau de décomposition. C'est une trame ajustée. Les atomes élémentaires de cette trame sont des fonctions indicatrices de carrés. De telles fonctions modélisent mal les structures des images.

La figure 5.11 présente les résultats de restauration. Nous avons choisi le paramètre λ de telle manière que la norme l^2 des résidus soient identiques. Avec cette transformée, les résultats de restauration avec critères de régularité sont bien plus satisfaisants (amélioration de $1.3dB$).

FIG. 5.11 – HG : image originale - HD : image à restaurer (SNR : 24.61 dB) - BG : image restaurée avec des critères de régularité (SNR : 31.10 dB) - BD : image restaurée avec des critères de parcimonie (SNR : 29.79 dB)

Nous pensons qu'une comparaison théorique est nécessaire pour pouvoir porter des conclusions plus valables sur la pertinence des deux a priori. Des comparaisons pratiques en utilisant des transformées plus redondantes de type curvelet permettraient aussi de conclure avec plus de précision. Nous nous attaquerons sans doute à ce problème dans la suite de nos travaux.

5.6 Mini-guide de l'utilisateur

Nous venons de montrer de façon détaillée comment utiliser les schémas de minimisation multi-pas proposés dans le chapitre 4. Pour conclure ce chapitre, nous résumons brièvement comment et dans quels cas utiliser ces algorithmes. Ce paragraphe est destiné à ceux qui souhaitent rapidement savoir quelle méthode utiliser et comment l'utiliser pour un problème particulier. Tous les algorithmes présentés s'appliquent à un problème du type :

$$\min_{x\in X}\left(f(x)+\Psi(x)\right) \tag{5.68}$$

où f et Ψ sont convexes. La première étape pour appliquer ces algorithmes consiste invariablement à caractériser la classe de fonctions à laquelle appartiennent f et Ψ. Ces fonctions peuvent appartenir simultanément à plusieurs classes. Dans ce cas, il faut privilégier les méthodes suivant l'ordre suivant :

1. Si f est Lipschitz différentiable et si Ψ est simple : passer au paragraphe 5.6.1. On obtient un taux de convergence supérieur à $O\left(\frac{1}{k^2}\right)$.
2. Si f est une somme de fonctions max et de fonctions Lipschitz et si Ψ est fortement convexe : passer au paragraphe 5.6.2. On obtient un taux de convergence supérieur à $O\left(\frac{1}{k^2}\right)$.
3. Si f est une fonction max et Ψ est simple : passer au paragraphe 5.6.3. On obtient un taux de convergence supérieur à $O\left(\frac{1}{k}\right)$.

Finalement, notons qu'assez souvent, un problème ne s'exprime pas directement sous la forme (5.68). Cependant, en manipulant le problème initial astucieusement, on peut souvent se ramener à cette formulation. Nous espérons que les quelques exemples donnés précédemment et dans le chapitre à venir peuvent donner des idées pour transformer un problème sous la forme (5.68).

5.6.1 Lipschitz différentiable + Simple

Le prototype des problèmes de ce type est le suivant :

$$\min_{x\in X}\left(\frac{1}{2}|Ax-b|_Y^2+\lambda|x|_{l^1(X)}\right) \tag{5.69}$$

où $A : X \to Y$ est une transformée linéaire. La fonction $f : x \mapsto \frac{1}{2}|Ax-b|_Y^2$ est $\|A\|^2$-Lipschitz différentiable. La fonction $\Psi(x) : x \mapsto \lambda|x|_{l^1(X)}$ est simple. Ce genre de problème arrive typiquement lorsque l'on souhaite décomposer une image sur un dictionnaire, lorsque l'on souhaite déconvoluer des images entâchées de bruits gaussiens ou encore pour des problèmes de type échantillonage compressif.

La stratégie que l'on suggère consiste à appliquer l'algorithme de Y. Nesterov présenté dans la section 4.1. La version implémentable est l'algorithme 4.2. Celui-ci converge linéairement s'il existe $\epsilon > 0$ tel que $\|Ax\| \geq \epsilon|x|_X$ (A est une trame). Il converge en $O\left(\frac{1}{k^2}\right)$ si le noyau de A n'est pas réduit à $\{0\}$.

5.6.2 Max + Fortement convexe

Le prototype des problèmes de ce type est le suivant :

$$\min_{y \in Y} \left(|By|_{l^1(X)} + \frac{\lambda}{2} |Ay - y_0|_Y^2 \right) \tag{5.70}$$

où $B : Y \to X$ est une transformée linéaire et $A : Y \to Y$ est une transformée linéaire inversible. La fonction $f : y \mapsto |By|_{l^1(X)}$ est une fonction max. La fonction $\Psi : y \mapsto \frac{\lambda}{2} |Ay - y_0|_Y^2$ est une fonction fortement convexe. Par exemple si $B = \nabla$ et $A = Id$, ce problème est celui de Rudin-Osher-Fatemi. A peut représenter un produit de convolution inversible, et dans ce cas, l'équation (5.70) devient un problème de déconvolution.

La stratégie que nous suggérons est d'appliquer l'algorithme présenté dans la section 4.2. Il consiste à formuler un problème dual et à le résoudre avec l'algorithme 4.5. Cet algorithme converge linéairement si B est une trame. Il converge en $O\left(\frac{1}{k^2}\right)$ si le noyau de B n'est pas réduit à $\{0\}$.

5.6.3 Max + Simple

Ce formalisme incorpore un très grand nombre de problèmes, tels que la déconvolution avec des bruits gaussiens, des bruits impulsionnels, des bruits uniformes, des bruits de compression, avec des a priori de régularité ou de parcimonie, sous forme contrainte ou lagrangienne. Nous conseillons deux stratégie de minimisation dans cette thèse :

1. La stratégie à préférer si les calculs ne sont pas trop compliqués (il faut le vérifier à chaque fois), consiste à remplaçer le problème (5.68) par le suivant :

   $$\min_{x \in X} \left(f(x) + \Psi(x) + \epsilon |x - x^0|_X^2 \right) \tag{5.71}$$

   On exprime alors le problème dual à (5.71) et on le résoud avec l'algorithme 4.4. Un exemple est proposé dans le paragraphe 5.3.1. Cet algorithme converge en $O\left(\frac{1}{k}\right)$.

2. S'il est trop difficile d'exprimer un problème dual, on remplace le problème (5.68) par le suivant :

   $$\min_{x \in X} \left(f_\mu(x) + \Psi(x) \right) \tag{5.72}$$

   où f_μ est une approximation différentiable de f. On résoud ce problème avec l'algorithme 4.2. On obtient un taux de convergence en $O\left(\frac{1}{k}\right)$.

Chapitre 6

Décomposition d'images

Dans ce chapitre on va voir que les algorithmes proposés peuvent aussi être appliqués aux problèmes de décomposition d'images en texture et en géométrie. Nous proposons une comparaison rapide des différents modèles de décomposition et en proposons un nouveau qui fournit des résultats très satisfaisants.

6.1 L'origine des modèles de décomposition

La première application de la variation totale en traitement d'images fut proposée par Rudin-Osher-Fatemi dans [Rudin *et al.*, 1992a]. Elle consistait à résoudre :

$$\inf_{u\in BV}\left(\int_\Omega |Du| + \int_\Omega |u-f|^2 d\Omega\right) \tag{6.1}$$

Dans son livre [Meyer, 2001], Y. Meyer a étudié ce modèle théoriquement et a mis en évidence son incapacité à séparer la partie géométrique d'une image et ses parties oscillantes. Il propose alors un modèle dit de décomposition en texture et en géométrie. Pour simplifier la présentation, nous présentons ce modèle dans le cadre discret. Le lecteur intéressé par le formalisme continu peut se référer à [Meyer, 2001; Aujol *et al.*, 2005] pour plus de détails. Y. Meyer définit une norme :

$$|v|_G = \inf_{g\in Y, div(g)=v}(|g|_{l^\infty(Y)}) \tag{6.2}$$

et propose de décomposer une image f en une partie géométrique u et une partie oscillante v en résolvant le problème suivant :

$$\inf_{(u,v)\in X^2,\, f=u+v}\left(|\nabla u|_{l^1(X)} + \lambda |v|_G\right) \tag{6.3}$$

La norme G d'une fonction est faible pour des fonctions oscillantes à moyenne nulle et grande pour des fonctions "géométriques" (par exemple des fonctions constantes par morceaux). C'est pourquoi ce modèle devrait permettre d'extraire plus efficacement les parties oscillantes d'une image que le modèle classique de Rudin-Osher-Fatemi. Notons que ce problème est convexe.

6.2 Reformulation du problème

Dans son travail, Y. Meyer n'a pas proposé de méthode numérique de résolution. Les premiers auteurs à s'être intéressés à cette tâche sont L. Vese et S. Osher dans [Osher et Vese, 2003]. Plus tard d'autres chercheurs se sont attaqués à ce problème. Citons notamment les travaux de J.F. Aujol et. al. dans [Aujol, 2004] et de D. Goldfarb et. al. dans [Goldfarb et Yin, 2005]. Le premier travail résout le problème en le régularisant puis en utilisant un schéma de premier ordre. Les auteurs montrent la convergence de leur algorithme. Le second résout le problème en utilisant l'algorithme de second order cone programming. Il converge vers une solution exacte. Ces deux techniques restent cependant améliorables. La première converge trop lentement et on a pu vérifier qu'elle ne donne qu'une assez mauvaise approximation de la solution. La seconde méthode donne une solution exacte (à la précision machine près), mais elle ne permet pas de traiter de grosses images car elle requiert une quantité de mémoire et des temps de calcul très importants. Dans la suite de cette partie, nous présentons une méthode de résolution originale.

Notons $J(u) = |\nabla u|_{l^1(Y)}$. Le problème de Y. Meyer discrétisé s'écrit :

$$\min_{u\in X}\left(J(u) + \lambda \min_{g\in Y, div(g)=f-u}(|g|_{l^\infty(Y)})\right) \tag{6.4}$$

Proposition 6.4. *Le problème (6.4) peut être reformulé comme suit :*

$$\inf_{g\in Y, \|g\|_\infty\leq\alpha}(J(f - div(g))) \tag{6.5}$$

Démonstration. Il suffit d'utiliser le changement de variable $u = f - div(g)$. Ceci permet d'obtenir un problème d'optimisation qui ne dépend que de la variable g. L'opérateur div est surjectif de Y dans $\tilde{X} = X - \{(\gamma, \gamma, ..., \gamma), \gamma \in \mathbb{R}\}$. On peut donc écrire :

$$\inf_{u\in X}\left(J(u) + \lambda \inf_{g\in Y, div(g)=f-u}(\|g\|_\infty)\right) = \inf_{g\in Y}(J(f - div(g)) + \lambda\|g\|_\infty) \tag{6.6}$$

En remplaçant le paramètre lagrangien λ en contrainte, on obtient le résultat annoncé. □

Au lieu de résoudre le problème (6.5), on peut résoudre :

$$\inf_{g\in Y, |g|_{l^\infty(Y)}\leq\alpha}\left(J_\mu(f - div(g))\right) \tag{6.7}$$

où J_μ est une régularisation de J et on obtient un algorithme avec un taux de convergence en $O(\frac{1}{k})$. Notons que si on remplace la norme l^∞ par une norme l^2 dans l'équation (6.5), on retrouve le modèle de Osher-Solé-Vese [Osher *et al.*, 2003]. Celui-ci peut être résolu en $O\left(\frac{1}{k^2}\right)$ en utilisant une approche duale. On peut aussi remplaçer la norme l^∞ par une norme l^1, ce qui donne lieu à un modèle de décomposition inédit que nous utiliserons dans la suite. Cette nouvelle formulation du problème permet de contraindre facilement les propriétés du champ g, ce qui peut être utile pour faire des décompositions d'images adaptatives.

6.2.1 Comparaison pratique des différents modèles

Nous comparons ici les résultats de décomposition en utilisant :

– le modèle de Meyer :

$$\inf_{g\in Y,|g|_{l^\infty(Y)}\leq\alpha}\Big(J_\mu(f-div(g))\Big) \tag{6.8}$$

– le modèle de Vese-Osher :

$$\inf_{g\in Y,|g|_{l^2(Y)}\leq\alpha}\Big(J_\mu(f-div(g))\Big) \tag{6.9}$$

– le modèle $BV-l^1$:

$$\inf_{x\in X,|x-f|_{l^1(X)\leq\alpha}\leq\alpha}\Big(J_\mu(x)\Big) \tag{6.10}$$

– le modèle inédit :

$$\inf_{g\in Y,|g|_{l^1(Y)}\leq\alpha}\Big(J_\mu(f-div(g))\Big) \tag{6.11}$$

Tous ces problèmes peuvent être résolus efficacement avec l'algorithme 4.2.

FIG. 6.1 – Image à décomposer

Dans toutes les expériences, on choisit pour paramètre de régularisation $\mu = 0.001$. Après 150 itérations de faible coût, on n'observe plus d'évolution de la décomposition. Pour obtenir un résultat similaire, les descentes de gradient projeté nécessitent plus de 5000 itérations. Pour pouvoir comparer les différents modèles, on a choisi le paramètre α de telle manière à ce que toutes les parties texturées aient la même norme l^2.

La figure 6.2 montre que le modèle de Y. Meyer ne permet pas de correctement retirer les parties oscillantes des habits de Barbara. On peut montrer théoriquement que l'amplitude de la texture (composante v) est bornée par un paramètre dépendant linéairement de α dans (6.5). Ce résultat est sans doute à l'origine du résultat décevant. Sur l'exemple donné, le modèle de Vese-Osher donne des résultats plus satisfaisants. Ceci a déjà été remarqué dans [Yin *et al.*, 2007].

Le modèle $BV-l^1$ sépare correctement les parties oscillantes et géométriques. La même observation peut être faite lorsque l'on minimise la norme l^1 du champ g dans (6.5). On remarque que

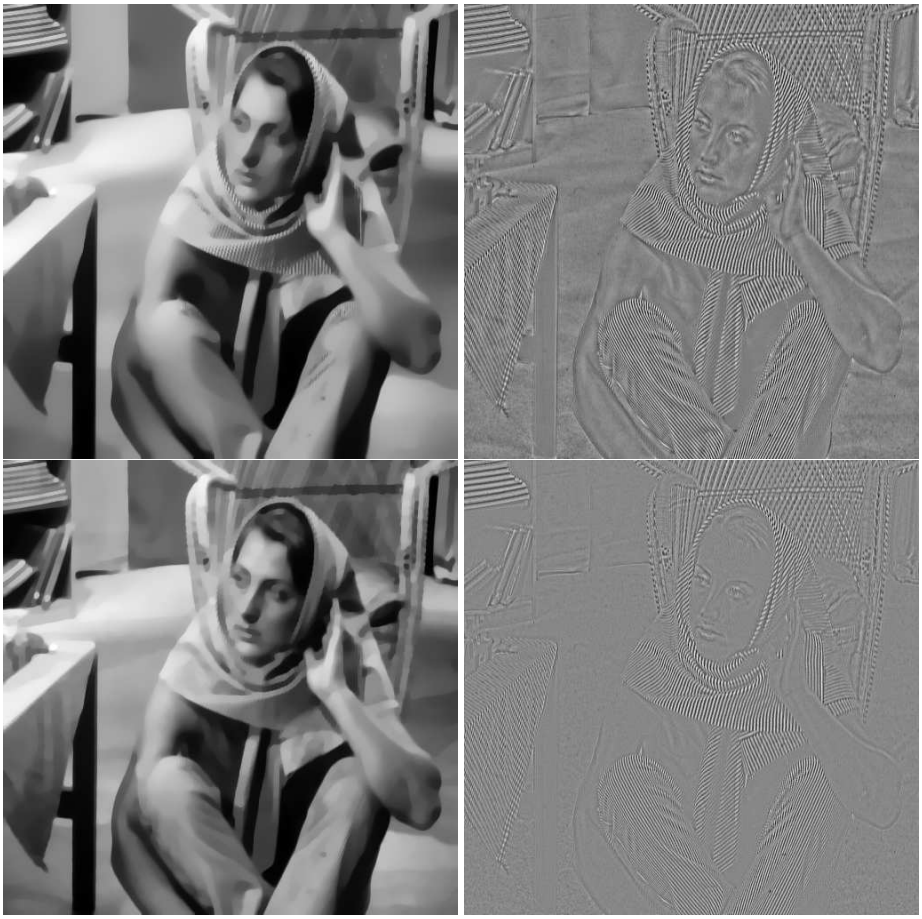

FIG. 6.2 – Décomposition en partie géométrique et en partie oscillante. Haut : modèle de Y. Meyer. Bas : modèle de Osher-Solé-Vese

les deux décompositions sont très similaires. Le modèle inédit (6.11) semble cependant extraire plus efficacement les parties oscillantes (chaise en arrière plan et habits par exemple). Ce modèle semble aussi donner lieu à une partie géométrique légèrement plus floue. Nous pensons que ceci doit être dû au fait que la discrétisation est plus diffusive. Elle repose en effet sur des différences finies d'ordre 4. Ces résultats expérimentaux indiquent qu'il serait sans doute intéressant d'étudier les propriétés analytiques de ce nouveau modèle.

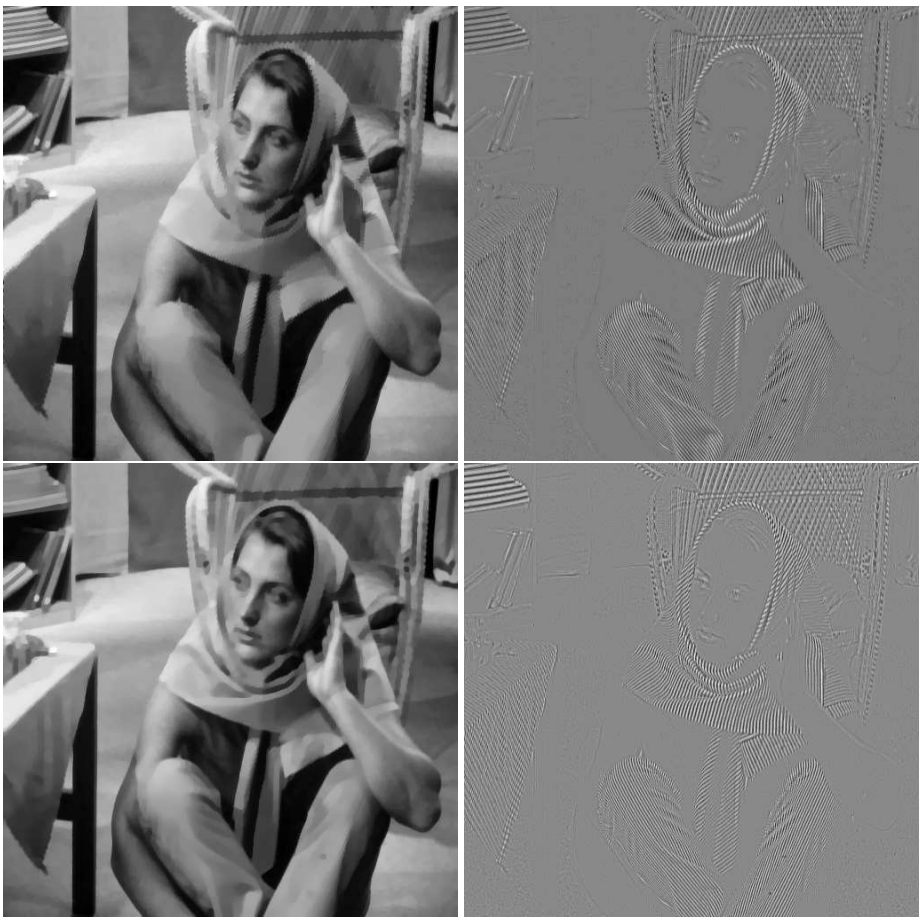

FIG. 6.3 – Décompositions en partie géométrique et en partie oscillante. Haut : modèle $BV - l^1$ - Bas : résultat de la minimisation de la norme l^1 du champ g dans (6.5)

Pour finir, notons qu'un autre modèle de décomposition d'images a été proposé dans [Starck *et al.*, 2005]. Ce modèle peut lui aussi être résolu par les méthodes présentées dans cette thèse.

Chapitre 7

Comparaisons expérimentales

Dans cette partie, nous comparons les algorithmes de minimisation proposés avec plusieurs autres approches. Nous nous concentrons uniquement sur les problèmes liés à la variation totale. Nous avons soumis un papier a SIAM Journal of Scientific Computing sur ce sujet [Weiss *et al.*, 2007b].

De très nombreuses méthodes ont été proposées pour résoudre ces problèmes. Certaines reposent sur des équations aux dérivées partielles [Rudin et Osher, 1994; Rudin *et al.*, 1992a], avec des schémas semi-implicites [Krishnan *et al.*, 2006] ou des méthodes de point fixe [Vogel et Oman, 1996]. D'autres reposent sur la minimisation d'une énergie discrétisée. On peut notamment citer les méthodes de sous-gradient [Combettes et Luo, 2002], les méthodes à base de projections sur des hyperplans [Combettes et Pesquet, 2004], les méthodes de quasi Newton (aussi appelées semi-quadratiques) [Charbonnier *et al.*, 1994], les méthodes de type Newton [Li et Santosa, 1996], les méthodes de second order cone programming [Goldfarb et Yin, 2005], les méthodes de points interieurs [Fu *et al.*, 2005], ou encore des approches par graph-cuts [Darbon et Sigelle, 2006; Chambolle, 2005]. Récemment plusieurs auteurs ont proposé des méthodes duales ou primales-duales [Chan *et al.*, 1999; Hintermüller et Stadler, 2006; Chambolle, 2004; Zhu, 2008]. Ce petit état des lieux est assez loin d'être exhaustif. La bibliographie de ce domaine contient plus de 50 articles de journaux et sa taille augmente régulièrement. Il est donc impossible de réimplémenter toutes ces méthodes. De plus, leur efficacité dépend fortement de choix comme les préconditionneurs. Nous restreignons donc nos comparaisons numériques aux méthodes classiques de premier ordre. Pour les nommer : les méthodes de sous-gradients et les descentes de gradient.

7.1 Le problème de Rudin-Osher-Fatemi

Le premier problème que nous considérons est celui de Rudin-Osher-Fatemi. De nombreux articles récents donnent des résultats de convergence pour ce problème et il permet de comparer les approches primales et duales. Le problème de Rudin-Osher-Fatemi consiste à résoudre :

$$\inf_{x\in X} \underbrace{\left(|\nabla x|_{l^1(X)} + \frac{\lambda}{2}|x - x_0|_2^2\right)}_{\phi(x)} \tag{7.1}$$

Il peut être régularisé de la façon suivante :

$$\inf_{x\in X} \left(|\nabla x|_{1,\mu} + \frac{\lambda}{2}|x - x_0|_2^2\right) \tag{7.2}$$

où $|\cdot|_{1,\mu}$ est la fonction de Huber définie en (5.32). Le problème dual à (7.1) peut être exprimé de la façon suivante :

$$\max_{y \in K} \left(-\frac{|\mathrm{div}\, y|_X^2}{4\lambda} + \langle x_0, -\mathrm{div}\, y \rangle_Y \right) \tag{7.3}$$

où $K = \{y \in Y, |y|_{l^\infty(Y)} \leq 1\}$. Le problème (7.1) peut-être résolu avec au moins 5 méthodes présentées dans cette thèse :

*Algo*1 La première méthode est la plus rapide théoriquement. Elle consiste à résoudre le problème dual (7.3) avec un schéma de Nesterov (voir algorithme 5.1). On a montré un taux de convergence en $O\left(\frac{1}{k^2}\right)$.

*Algo*2 La deuxième méthode consiste à résoudre le problème dual (7.3) avec une méthode de gradient projeté. On a montré un taux de convergence en $O\left(\frac{1}{k}\right)$.

*Algo*3 La troisième méthode consiste à résoudre le problème régularisé (7.2) avec un schéma de Nesterov. Là aussi on a montré un taux de convergence en $O\left(\frac{1}{k}\right)$.

*Algo*4 La quatrième méthode est la descente de gradient appliquée au problème régularisé (7.2). On peut montrer que cette méthode converge en $O\left(\frac{1}{\sqrt{k}}\right)$ [Weiss *et al.*, 2007b].

*Algo*5 La cinquième méthode est la descente de sous-gradient projeté avec pas optimal (3.20). η^k doit appartenir au sous-différentiel de la variation totale en x^k pour obtenir un algorithme convergent. On choisit :

$$\eta^k = -div(\Psi) \text{ avec } \Psi_i = \begin{cases} \frac{(\nabla x^k)_i}{|(\nabla x^k)_i|_2} & \text{si } |(\nabla x^k)_i|_2 > 0 \\ 0 & \text{sinon} \end{cases} \tag{7.4}$$

Le pas de la descente τ^k est choisi en fonction du théorème 3.23. Pour appliquer cette stratégie, il faut connaître f^*. Comme on ne connait pas cette valeur, on laisse un programme tourner jusqu'à convergence (*Algo*1), il renvoit une valeur très proche de f^*. Cet algorithme est inutilisable en pratique. Il sert uniquement de référence.

On utilise l'image Lena de taille 256×256 à valeurs dans $[0, 1]$. On lui ajoute un bruit gaussien ($\sigma = .15$). Ceci correspond aux images 7.1. La figure bas-gauche (BG) est donnée pour montrer que la stratégie de régularisation de la variation totale n'affecte pas la solution de façon perceptuelle.

Les figures 7.2, 7.3 et 7.4 montrent respectivement les courbes de convergence en termes d'objectif ($\phi(x^k) - \phi(x^*)$), en terme de distance l^2 à la solution ($|\phi(x^k) - \phi(x^*)|_X$) et en terme de distance l^∞ à la solution ($|\phi(x^k) - \phi(x^*)|_{l^\infty(X)}$).

Ces courbes contiennent de nombreuses informations que nous dépouillons ci-dessous :

– On remarque d'abord que la précision des algorithmes se basant sur une régularisation du problème (*Algo*3 et *Algo*4) est bornée inférieurement par une constante positive due à l'erreur d'approximation. Il est donc inutile, après avoir fixé le paramètre de régularisation μ, de faire un trop grand nombre d'itérations. Les stratégies présentées dans les théorèmes 4.22 et 4.23 sont de bonnes approximations en pratique.
– La descente de sous-gradient projeté (*Algo*5) converge assez rapidement dans les premières itérations, puis elle semble osciller autour du minimum avec une décroissance lente.
– Le classement pratique des algorithmes correspond exactement au classement théorique. En termes imagés, on a *Algo*1 >> *Algo*2 ~ *Algo*3 >> *Algo*4 ~ *Algo*5. *Algo*1 est bien le meilleur algorithme vis-à-vis de tous les critères. L'algorithme *Algo*2 se comporte assez bien dans les

FIG. 7.1 – HG : image originale - HD : image bruitée - BG : solution de (7.2) en utilisant l'algorithme *Algo*3 et en posant $\mu = 10^{-3}$ - BD : solution exacte de (7.1) obtenue en appliquant *Algo*1 jusqu'à convergence

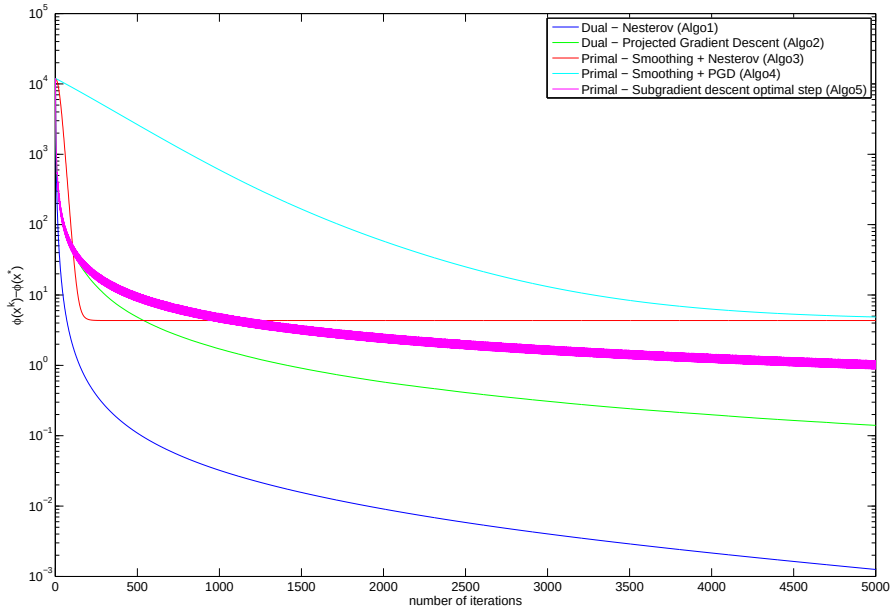

FIG. 7.2 – Convergence des différents algorithmes en terme d'objectif.

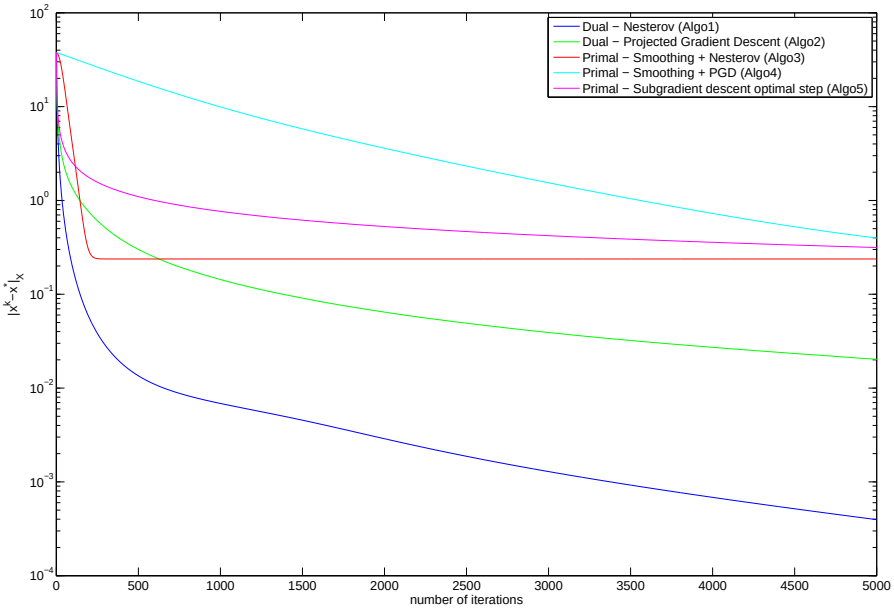

FIG. 7.3 – Convergence des différents algorithmes en terme de distance l^2 au minimiseur.

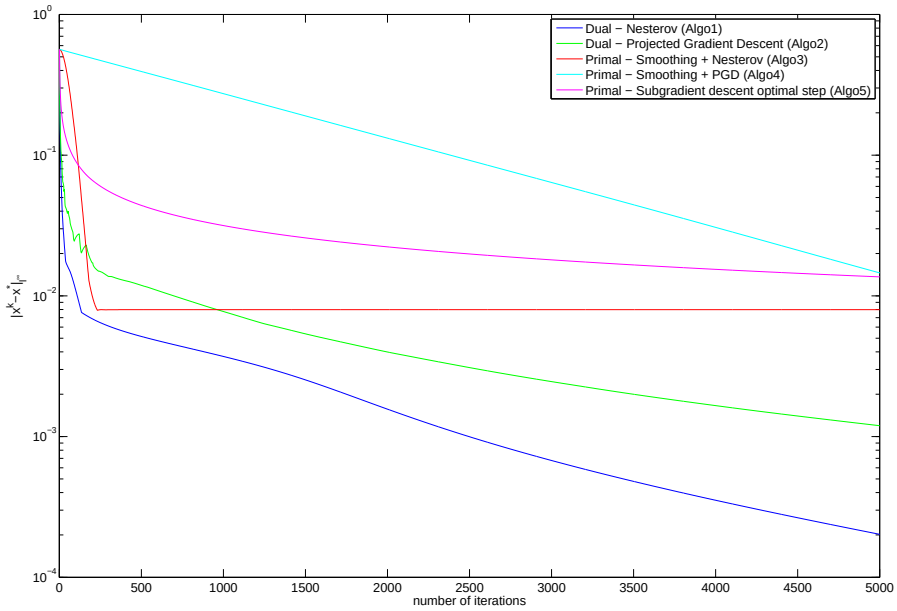

FIG. 7.4 – Convergence des différents algorithmes en terme de distance l^∞ au minimiseur.

premières itérations. Son comportement asymptotique est bien moins bon que celui de *Algo*1. L'algorithme *Algo*3 semble être moins efficace que *Algo*2. Il a cependant la même efficacité théorique en $O\left(\frac{1}{k}\right)$. Ce fait est confirmé par la petite plage d'itérations sur lequel il donne de meilleurs résultats que *Algo*2. Finalement les algorithmes *Algo*4 et *Algo*5 qui convergent en $O\left(\frac{1}{\sqrt{k}}\right)$ doivent être évités.

– Dans les formulations primales et duales, les algorithmes accélérés de Nesterov se comportent bien mieux que les méthodes de descente à un pas.
– La convergence en norme est aussi bonne que la convergence de l'objectif. Ce n'est pas très étonnant pour la norme l^2 à la vue de l'inégalité (2.52). C'est plus inattendu pour la norme l^∞.
– Pour obtenir une précision en norme l^∞ de $1/255 \sim 4 \times 10^{-3}$, l'algorithme *Algo*1 a besoin d'environ 900 itérations. Ce résultat devrait réjouir J. Darbon qui a récemment proposé un algorithme en $O(n\log(n))$ pour résoudre le problème de Rudin-Osher-Fatemi avec la même précision ([Darbon, 2008]). Ce résultat semble indiquer que les techniques à base de graph-cut devraient être préférées aux techniques présentées dans cette thèse. Pour le moment, elle ne permettent de résoudre efficacement que les problèmes à base de variation totale anisotrope avec des attaches aux données sans interactions entre les pixels. Espérons qu'elle puisse être étendues à des classes de fonctions plus larges à l'avenir.

Pour conclure, nous avons vu que les schémas accélérés se comportent bien mieux que les schémas à un pas. Cette expérience confirme aussi que le passage en dualité permet de largement accélérer la minimisation des problèmes fortement convexes.

On a vu que *Algo*2 et *Algo*3 convergent tous deux en $O\left(\frac{1}{k}\right)$. On a voulu vérifier qu'en pratique,

leur comportement était similaire, malgré le paramètre de régularisation introduit dans *Algo*3. Dans la figure 7.5, on régularise l'image Lena en posant $\lambda = 1$. On peut voir que pour tout nombre d'itérations k, il existe une paire (k, μ) telle que la précision obtenue avec *Algo*3 est la même que celle obtenue avec *Algo*2. Ce résultat indique bien que la technique "régularisation + schéma accéléré" a à peu près la même efficacité que l'algorithme de A. Chambolle[1]. Il est un peu plus dur d'utilisation, car il faut fixer le nombre d'itérations en fonction du paramètre de régularisation μ, mais il peut être utilisé pour une classe de problèmes bien plus large.

Une dernière remarque intéressante est que la précision maximale qui peut être obtenue avec la technique de régularisation dépend de μ linéairement. Ceci confirme une fois de plus la théorie.

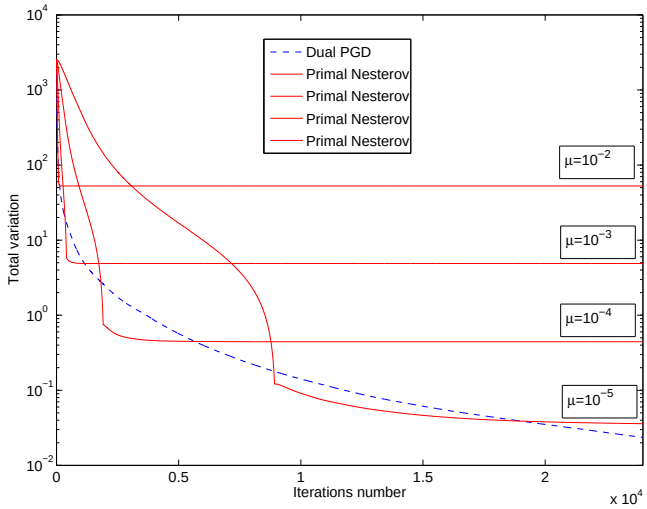

FIG. 7.5 – Comparaison de *Algo*2 et *Algo*3 avec différents paramètres de régularisation μ

7.2 Autres problèmes contraints

Nous nous intéressons maintenant à la comparaison des méthodes primales. Nous comparons les algorithmes *Algo*3, *Algo*4 et *Algo*5 pour résoudre le problème de minimisation de la variation totale sous contraintes (5.5) avec différentes fonctions F. Dans tous les cas, on pose $\mu = 0.001$ [2]. Nous testons l'efficacité du schéma sous différentes contraintes. Les modèles testés sont :

– Le problème de Rudin-Osher-Fatemi (Fig. 7.6).
– Le problème $BV - l^1$ (Fig. 7.7).
– Le problème de déconvolution (Fig. 7.8).

On remarque que dans tous les cas, le schéma de Nesterov a une efficacité très nettement supérieure à la descente de gradient projeté. La descente de sous-gradient projeté avec pas optimal permet de faire décroître l'objectif de façon très rapide dès les premières itérations. Asymptotiquement, son taux de convergence semble plus mauvais que l'approche "Nesterov + régularisation". Notre conclusion est

[1] l'algorithme *Algo*2 a été proposé dans ([Chambolle, 2005])
[2] ce choix mène à des solutions qui sont identiques perceptuellement aux solutions en posant $\mu = 0$

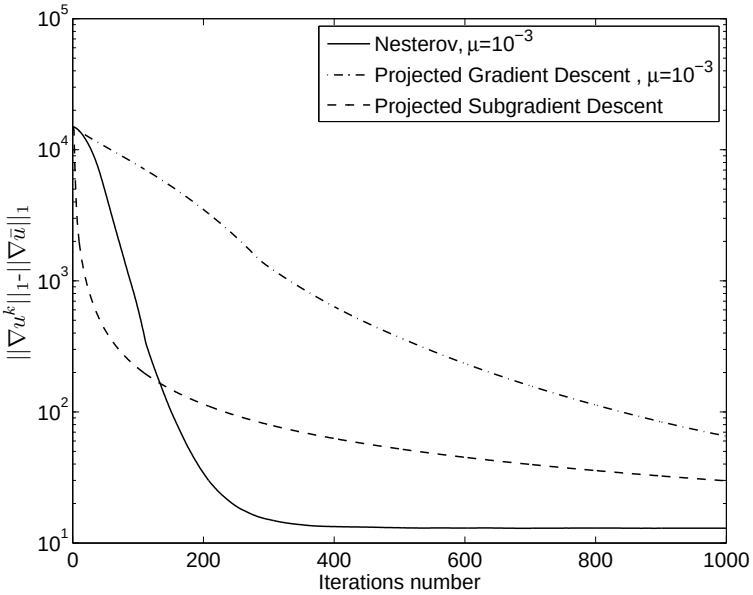

FIG. 7.6 – Evolution de $\|\nabla u^k\|_1 - \|\nabla \bar{u}\|_1$ avec échelle logarithmique pour le problème de Rudin-Osher-Fatemi. Résultats sur la figure 7.1.

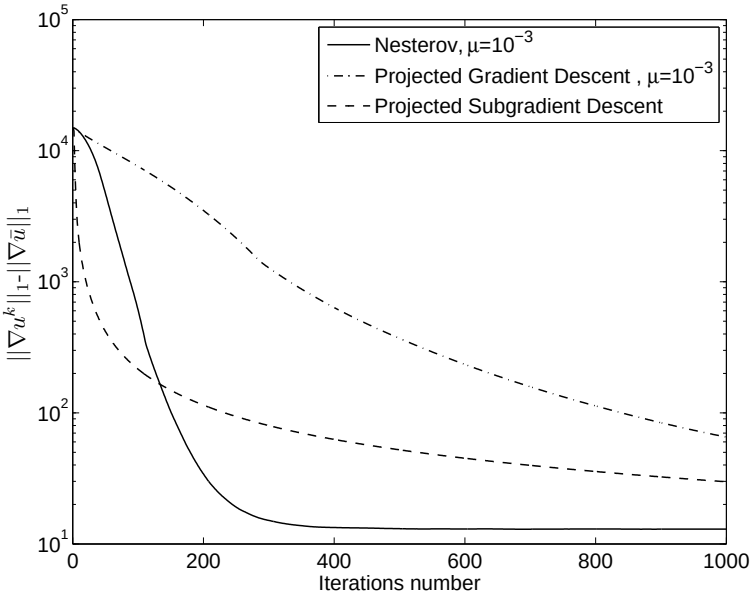

FIG. 7.7 – Evolution de $\|\nabla u^k\|_1 - \|\nabla \bar{u}\|_1$ en échelle logarithmique pour le problème $BV - l^1$. Résultats sur la figure 6.3.

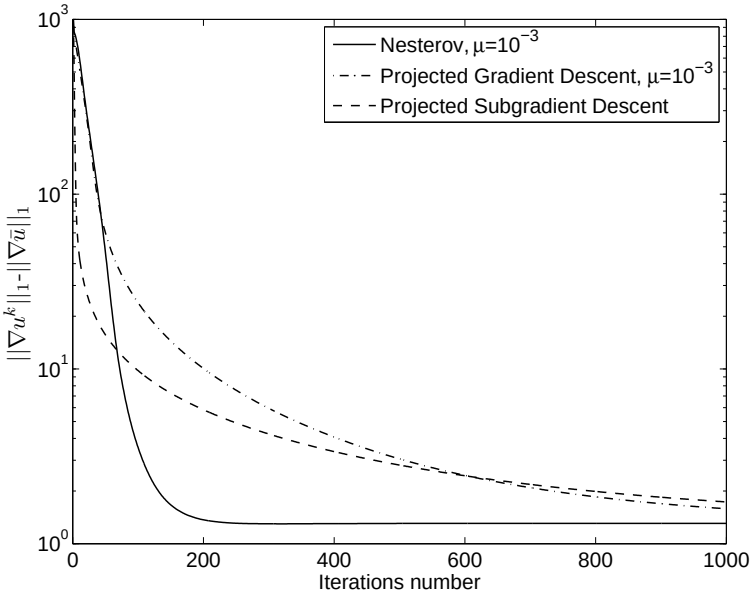

FIG. 7.8 – Evolution de $\|\nabla u^k\|_1 - \|\nabla \bar{u}\|_1$ en échelle logarithmique pour le problème de déconvolution. Résultats sur la figure 5.6.

que la méthode de sous-gradient projeté avec une suite de pas $\{\tau^k\}$ précalculée peut être intéressante pour obtenir une solution approchée en très peu d'itérations. Pour obtenir des solutions précises en des temps de calcul légèrement supérieurs, l'approche proposée est bien plus appropriée.

7.3 Discussion

7.3.1 Stabilité du schéma de Nesterov

Plusieurs personnes ayant testé les méthodes de Y. Nesterov nous ont fait part de leur étonnement à la vue de ces résultats. En effet, leurs tests avaient montré de nettes instabilités du schéma qui le rendait moins bon que les simples descentes de gradient par exemple. Notre conclusion est radicalement opposée. Dans tous les cas testés, la méthode accélérée multi-pas a une efficacité très nettement supérieure aux méthodes à un pas.

Nous avons tout de même remarqué qu'aymptotiquement, la méthode de Y. Nesterov peut se comporter moins bien qu'une simple descente de gradient. Elle permet de réduire le saut de dualité d'un facteur variant entre 10^5 et 10^{10} (ce qui est largement suffisant pour les besoins du traitement d'images), tandis que les méthodes de gradient permettent de réduire ce saut d'un facteur environ 10 fois supérieur (avec des temps de convergence jusqu'à 1000 fois supérieurs). Nous pensons que ce défaut de la méthode est dû au fait que la fonction ψ^k (équation (4.4)) est composée d'une grande somme de gradients pondérés par les coefficients croissants A^k. Un tel calcul est instable en arithmétique finie. Il peut être intéressant en pratique de réinitialiser cette variable périodiquement, si l'on souhaite obtenir de très bonnes précisions.

7.3.2 Comparaisons avec d'autres méthodes

Les méthodes de second ordre sont couramment utilisées pour résoudre les problèmes de restauration évoqués. Il nous semble que ces méthodes représentent les concurrents les plus redoutables à notre approche. De nombreux articles suggèrent la minimisation semi-quadratique [Geman et Yang, 1995; Charbonnier *et al.*, 1994]. M. Nikolova et R. Chan ont montré récemment que ces méthodes étaient strictement équivalentes aux méthodes de quasi-Newton [Nikolova et Chan, 2007]. M. Allain *et al.* ont montré que leur taux de convergence est linéaire [Allain *et al.*, 2006]. Un tel taux de convergence est meilleur asymptotiquement que notre décroissance en $\left(O\frac{1}{k^\alpha}\right)$. En pratique, on observe une convergence bien plus rapide pour les méthodes de quasi-Newton (une dizaine d'itérations suffisents pour obtenir des solutions stables). La contrepartie réside clairement dans le fait qu'il faut résoudre un très grand système linéaire à chaque itération. L'efficacité de cette méthode dépend donc fortement du conditionnement des systèmes linéaires et du choix de préconditionneurs. Des expériences préliminaires sans préconditionnement donnent un avantage net aux méthodes proposées dans cette thèse. Bien entendu, il faudrait mener plus d'expériences pour confirmer ce résultat.

Les méthodes de second order cone programming ont été proposées récemment [Yin *et al.*, 2007] et mènent à des solutions très précises. Cependant, les temps de calcul sont très grands. Cette approche ne nous semblent pas raisonnable pour résoudre des problèmes en très grande dimension. Cela reste une très bonne stratégie pour évaluer les qualités et défauts d'un modèle.

Des approches très prometteuses par graph-cuts ont récemment été proposée pour résoudre les problèmes de variation totale [Darbon et Sigelle, 2006; Chambolle, 2005]. Les auteurs résolvent les problèmes $BV - l^1$, $BV - l^2$ et $BV - l^\infty$. Ils montrent qu'ils obtiennent des solutions exactes (à un facteur de quantification près) en un nombre fini d'itérations. J. Darbon semble avoir trouvé une méthode de résolution optimale en $O(n\log(n))$ ([Darbon, 2008] + communication privée). Un tel taux est bien meilleur que tout ce qu'on a proposé dans cette thèse. Cependant la méthode n'est pour le moment applicable qu'à une classe de problème très restreinte (variation totale anisotrope + contraintes simples). L'approche présentée dans cette thèse permet de résoudre une classe de problèmes bien plus vaste avec une efficacité restant très bonne.

Nous pensons que des solutions très précises n'ont pas grand intérêt en traitement d'images. Le système visuel humain est incapable de percevoir de petites modifications de l'image. Par exemple un bruit uniforme d'amplitude 5 que l'on ajoute à une image codée sur 256 niveaux est imperceptible. Nous pensons donc que les méthodes de premier ordre sont particulièrement adaptées au traitement d'images. Les méthodes que nous avons proposées convergent vers des solutions approchées en des temps très courts et les algorithmes proposés sont faciles à implémenter.

Chapitre 8

Conclusions et perspectives

Dans cette première partie de thèse, nous avons proposé de nombreux schémas de minimisation de premier ordre adaptés à de larges classes de fonctions couramment utilisées en traitement d'images. Nous avons expliqué théoriquement l'intérêt des techniques duales d'optimisation ainsi que l'intérêt des méthodes de régularisation. L'application des schémas de Y. Nesterov à ces divers problèmes montre qu'elles ont un avantage très net sur d'autres techniques récemment proposées dans le domaine du traitement d'images. Nous espérons donc que cette thèse permettra de dynamiser la recherche théorique et pratique sur les schémas multi-pas qui semblent pour le moment être étudiés par très peu d'auteurs (A notre connaissance uniquement Y. Nesterov).

Pendant cette thèse, nous nous sommes posé de nombreuses questions qui n'ont pas encore trouvé de réponse et nous motivent particulièrement à continuer dans la recherche. Nous les détaillons ci-dessous :

- *Comment minimiser des fonctions à dérivées non bornées ?* Dans cette thèse, on a toujours eu implicitement recours à la propriété (2.43) des fonctions Lipschitz différentiables. Si on ôte cette propriété, plus aucun des algorithmes structurels décrits ne fonctionnent. On peut donc se demander comment minimiser des fonctions de type $x \mapsto \log(Ax)$ où A est une transformée linéaire [1] ou des fontions du type $x \mapsto |Ax|_{l^p(Y)}$ avec $p \notin \{1, 2, \infty\}$[2]. Etrangement, ces fonctions différentiables semblent plus compliquées à minimiser que certaines fonctions non différentiables telles que les normes l^1 de transformées linéaires. Une partie de réponse peut venir de la théorie des fonctions auto-concordantes [Nesterov et Nemirovskii, 1994; Nesterov et Todd, 1997] dont le log fait partie. Pour ces fonctions, des taux de convergence satisfaisants sont obtenus avec des méthodes de second ordre du type points intérieurs. Aucun taux de convergence n'a été trouvé à notre connaissance pour des méthodes de premier ordre.
- *Quelles sont les fonctions pour lesquelles les méthodes de premier ordre permettent de définir des taux de convergence en norme ?* Très souvent, nous avons présenté des taux de convergence en termes d'objectif. Le théorème 3.18 indique que trouver des taux de convergence en norme est en général impossible lorsque l'on souhaite minimiser des fonction à gradient Lipschitz. Il serait intéressant de trouver des classes de problèmes (non fortement convexes) pour lesquelles on peut donner des taux de convergence en norme.

[1] Une telle fonction apparaît notamment dans les problèmes de déconvolution avec bruit de Poisson.

[2] Précisons que la minimisation de normes l^p ($p \neq 2$) nous semble être le pire problème rencontré pendant cette thèse. Nous ne voyons pas comment le résoudre autrement qu'en utilisant des techniques de descente avec pas décroissant qui convergent en $O\left(\frac{1}{\sqrt{k}}\right)$. Les techniques de régularisation notamment n'ont aucun intérêt pratique.

– *Qui des a priori de régularité ou des a priori de parcimonie fournissent les meilleurs résultats de restauration ? Quels sont les liens entre les solutions des deux problèmes ?* Ces deux a priori sur les images fournissent de bons résultats de restauration et sont aujourd'hui très largement utilisés. Dans cette thèse, nous avons montré que la complexité algorithmique de la minimisation des deux critères est équivalente. Il serait intéressant de comparer ces critères en pratique et en théorie.
– *Comment comparer les méthodes proposées aux méthodes d'ordre supérieur ?* Les algorithmes de second ordre tels que les méthodes de points intérieurs sont de sérieux concurrents aux méthodes de premier ordre proposées dans cette thèse. La comparaison théorique de ces méthodes n'est pas triviale, car il faut prendre en compte la complexité de la résolution de systèmes linéaires. Pour montrer aux adeptes des méthodes de type Newton que la question est loin d'être tranchée, rappelons que la technique du gradient conjugué (largement utilisée pour inverser des systèmes linéaires) converge en $O\left(\frac{1}{k^2}\right)$ (au pire des cas), tout comme la méthode optimale de Y. Nesterov.

Deuxième partie

Sur l’invariance des lignes de niveau aux changements d’illuminations. Application à la détection de changements.

Chapitre 1

Introduction

La deuxième partie de cette thèse est quasi indépendante de la précédente. Nous y abordons le problème de la détection de changements entre deux images satellitaires de haute résolution par des méthodes variationnelles. Ce problème va nous mener à étudier et à utiliser l'invariance des lignes de niveau d'une image à des variations des conditions d'illumination. Ces résultats ont vu le jour grâce à une collaboration avec Alexandre Fournier (doctorant du projet Ariana). Dans cette introduction, nous commençons par définir le problème de la détection de changements et montrons qu'il doit être effectué à partir d'attributs d'images invariants aux conditions de prises de vue. Nous faisons ensuite un bref rappel bibliographique des attributs d'images invariants aux variations d'illumination. Nous détaillons finalement le plan de la partie et énumérons nos contributions.

1.1 Un bref aperçu de la problèmatique

1.1.1 Qu'est-ce que la détection de changements ?

Que doit faire un algorithme de détection de changements entre deux images ? Pour commencer cette partie, nous proposons une définition originale de ce problème. Nous renvoyons le lecteur à l'article [Radke *et al.*, 2005] pour voir le problème sous des angles différents.

La difficulté principale pour définir un algorithme de détection de changements est de trouver les changements de scène autres que ceux qui sont dus à des conditions d'acquisition différentes. Formulons ce problème mathématiquement. Soient $u_1 : \Omega \to \mathbb{R}$ et $u_2 : \Omega \to \mathbb{R}$ deux photographies d'une même région, prises à deux instants différents. On considère que l'image u_1 est issue d'une "scène" S_1. S_1 peut regrouper plusieurs propriétés du monde réel telles que sa géométrie 3D et ses propriétés de réflectance. L'image u_1 est prise sous un ensemble de conditions de prise de vue L_1 (angle de vue, conditions d'illuminations,...). On suppose aussi que l'image u_2 est issue d'une scène S_2 sous un ensemble de conditions L_2. Sous ces hypothèses très générales (qu'on détaillera plus précisément par la suite), on peut écrire :

$$\begin{cases} u_1 = u_{S_1,L_1} \\ u_2 = u_{S_2,L_2} = u_{S_1,L_2} + c_{1,2} \end{cases} \tag{1.1}$$

où u_{S_1,L_1} est l'image de la scène S_1 sous les conditions L_1, u_{S_1,L_2} est l'image de la scène S_1 sous les conditions L_2 et $c_{1,2}$ est une image des changements entre l'image u_1 et l'image u_2. Dans ce travail le problème de la détection de changements est défini comme suit :

Trouver l'image $c_{1,2}$ à partir de u_1 et u_2.

1.1.2 Le principe de la méthode variationnelle

Pour retrouver $c_{1,2}$, on peut introduire des *a priori* sur l'image de changements ainsi que sur les liens entre u_{S_1,L_1} et u_{S_1,L_2}. Ceci nous mène à la résolution d'un problème du type suivant :

$$c_{1,2} = \arg\inf_{c \in X} \left(\phi(c) + \Psi(u_2 - c, u_1)\right) \tag{1.2}$$

où $\phi : X \to \mathbb{R}$ est un a priori sur l'image de changements et $\Psi : X \times X \to \mathbb{R}$ est une fonction qui favorise les couples d'images pouvant être issues d'une même scène. On cherche donc l'image la plus probable parmi les changements qui ne sont pas dus aux conditions d'acquisition. Nous venons de poser les bases du problème. Toute la difficulté consiste maintenant à trouver et à justifier des a priori.

Dans ce travail, nous nous intéressons surtout à l'a priori Ψ. Celui-ci doit favoriser les images pouvant provenir d'une même scène sous des conditions de vue différentes. Mathématiquement, on va donc chercher à construire une fonction qui satisfait les conditions suivantes :

$$\forall (L_1, L_2) \quad \begin{cases} \Psi(u_{S,L_1}, u_{S,L_2}) & = & 0 \ \ \forall S \\ \Psi(u_{S_1,L_1}, u_{S_2,L_2}) & > & 0 \ \ \text{si } S_1 \neq S_2 \end{cases} \tag{1.3}$$

Pour assurer une telle invariance, la fonction Ψ doit être construite à partir d'attributs d'images invariants aux conditions de prises de vues. Il existe une littérature vaste concernant ce domaine. Nous référons le lecteur intéressé par un aperçu au livre [Mundy et Zisserman, 1992]. Suivant la définition que l'on donne à "prise de vue", ces attributs vont être différents. Par exemple, si on suppose que "prise de vue" signifie position de l'imageur, il faudra trouver des attributs invariants aux transformées affines ou homographiques. Dans ce travail, nous supposerons que les deux images u_1 et u_2 sont prises du même endroit avec le même angle de vue [1]. Par contre, nous supposerons que la position et l'intensité de la source de lumière varient. Nous allons donc devoir construire Ψ à partir d'attributs invariants aux conditions d'illumination.

1.1.3 Les attributs d'images invariants aux conditions d'illumination

La recherche de caractéristiques d'images invariantes à l'illumination est un problème récurrent en traitement d'images. La référence [Chen *et al.*, 2008] donne un bon état de l'art de ce domaine. Nombre d'applications, comme la reconnaissance, l'égalisation/réhaussement de contraste, le recalage d'images et la détection de changements doivent être construites à partir de tels attributs.

L'ensemble des contours ou des discontinuités d'une image sont probablement la caractéristique la plus utilisée [Marr, 1982]. Ceux-ci sont généralement dus à des discontinuités de l'élévation ou de l'albédo de la scène et sont invariants aux changements d'illumination dans le sens où ils apparaissent sur une image pour la plupart des conditions d'illumination. Récemment, Caselles *et al.* [Caselles *et al.*, 1999a] ont proposé un outil alternatif. Les auteurs montrent que les lignes de niveau sont invariantes aux *changements de contraste locaux*. L'ensemble des lignes de niveau (ou carte topographique) présente deux avantages importants par rapport aux contours. Premièrement, dans un cadre discret, la définition des contours repose en général sur un seuil, contrairement à la carte topographique. Deuxièmement, la carte topographique permet de reconstruire une image, ce qui n'est pas

[1] c'est une hypothèse raisonnable étant donné que les satellites d'observation sont en orbite héliosynchrone et passent au même endroit périodiquement

le cas de caractéristiques singulières telles que les discontinuités ou les jonctions multiples. Certaines applications des cartes topographiques – comme la détection de changements [Monasse et Guichard, 2000; Ballester *et al.*, 2000], le recalage [Monasse, 1999] ou le réhaussement de contraste [Caselles *et al.*, 1999b] ont été récemment proposées et donnent des résultats convaincants. Cependant, si les lignes de niveaux sont invariantes aux changements de contrastes locaux, elles ne sont pas invariantes aux changements d'illumination [2]. On peut en effet trouver des scènes dont les lignes de niveaux varient en fonction des conditions d'éclairement. La figure 1.1 présente un exemple de surface dont les lignes de niveau ne sont pas invariantes vis-à-vis des changements d'illumination. Un tel exemple montre que les méthodes citées plus haut ne sont pas complètement justifiées.

Dans cette partie, nous trouvons des conditions nécessaires et suffisantes sur la géométrie de la scène pour que les lignes de niveau soient invariantes aux variations de la direction d'incidence de la lumière. Dans un premier temps, nous montrons qu'elles sont invariantes si et seulement si la scène est développable et que son albédo varie selon certaines directions. Ces surfaces présentent un intérêt limité, car elles modélisent peu de scènes réelles. Ceci nous mène à l'étude de l'invariance pour des surfaces développables par morceaux. Nous montrons que les images de ces objets ont des lignes de niveau "presque" invariantes.

1.1.4 L'algorithme et ses concurrents

La plupart des constructions humaines peuvent être considérées comme développables par morceaux. Les résultats théoriques démontrés dans cette partie indiquent qu'on peut tirer parti des lignes de niveau pour réaliser un algorithme adapté à la détection de changements sur des couples d'images recalées. Cet algorithme est particulièrement adapté pour la détection de changements sur des images de chantiers [3] ou des images urbaines en télédétection. Nous définissons donc une fonction Ψ qui tire parti des lignes de niveaux des images. Cette approche donne lieu à un algorithme simple et invariant aux conditions d'illumination qui fournit de bons résultats.

D'autres auteurs ont tenté d'élaborer des algorithmes de détection de changements robustes aux variations d'illumination. Les revues sur la détection de changements [Lu *et al.*, 2004; Radke *et al.*, 2005] en donnent quelques exemples. Les premières tentatives furent des égalisations linéaires de contraste [Wiemker, 1997]. Quelquefois, ces égalisations peuvent être locales (normalisation de la variance et de la moyenne sur des fenêtres) [Lillestrand, 1972]. L'utilisation du ratio entre les deux images [Toth *et al.*, 2000; Watanabe *et al.*, 1998] est parfois préconisé. Plus récemment des auteurs ont suggéré un réhaussement global de contraste [Theiler et Perkins, 2007]. On verra que ces approches ne permettent pas de corriger correctement toutes les variations d'illumination. Des tests utilisant ces techniques mènent à un grand nombre de fausses alarmes. Certains travaux sont plus proches de notre approche. Dans [Watanabe *et al.*, 1998], les auteurs supposent que les toits sont plats et que leur réflectance est lambertienne. Sous ces hypothèses, ils montrent que le ratio entre les deux images peut être utilisé afin de détecter les ombres et les changements quelle que soit la direction d'illumination. Leurs hypothèses sur la scène sont cependant très restrictives. Dans [Monasse et Guichard, 2000; Ballester *et al.*, 2000], les auteurs proposent un algorithme de détection de changements qui s'appuie largement sur les lignes de niveau. On verra que la définition des lignes de niveau qu'ils proposent ne les rend pas invariantes aux changements d'illumination, même pour des scènes très simples. A

[2] Il nous semble que la définition d'un changement de contraste local [Caselles *et al.*, 1999a] est un peu artificielle. On peut voir les changements de contraste locaux comme la classe de transformations la plus large qui assure l'invariance des cartes topographiques.

[3] C'est un problème réel pour l'évaluation de l'avancement des travaux.

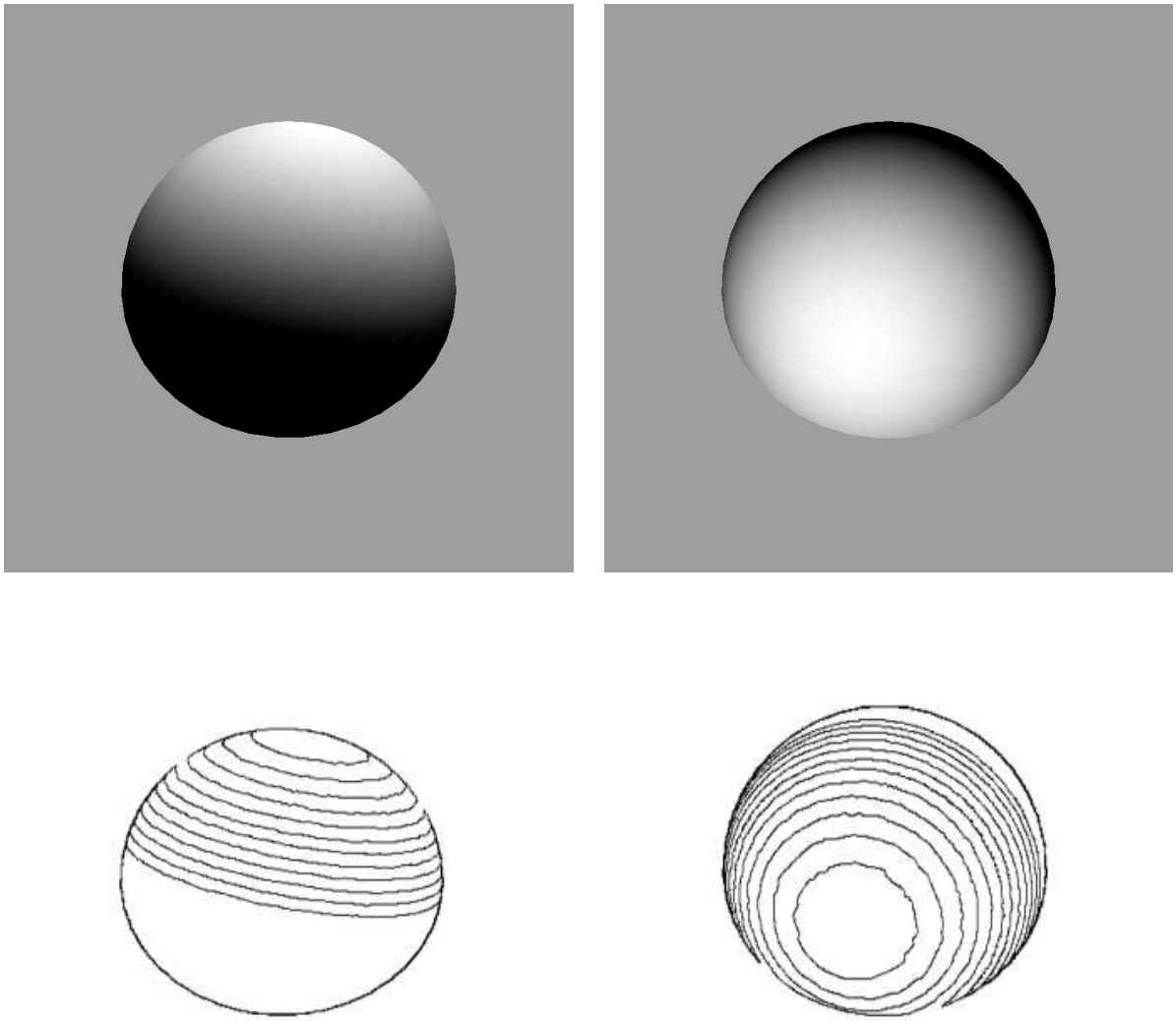

FIG. 1.1 – Haut : images d'un dôme en 3 dimensions illuminé d'après le modèle de Phong en utilisant deux directions d'incidence différentes. Bas : quelques lignes de niveau associées aux illuminations respectives.

la fin de cette partie, on compare ces différentes approches sur des images de synthèse et des images Quickbird haute résolution. Ces expériences donnent un avantage net à notre méthode.

1.2 Plan de la partie et contributions

Cette partie est divisée en 5 chapitres courts :

- Dans le deuxième chapitre, on introduit les notations et on présente brièvement le modèle de formation d'images (le modèle de Phong). On définit précisément la notion de "scène" et de "conditions de prise de vue".
- Le troisième chapitre contient une étude théorique de quelques propriétés des lignes de niveau. Nous commençons par les définir. Cette définition est différente de celle proposée dans [Caselles *et al.*, 1999a]. Nous montrons ensuite plusieurs résultats annexes qui serviront essentiellement à montrer les résultats du chapitre 4.
- Le quatrième chapitre contient les résultats théoriques les plus importants de cette partie. On caractérise les scènes qui fournissent des lignes de niveau invariantes aux changements d'illumination.
- Dans le quatrième chapitre, on présente un algorithme simple de détection de changements reposant sur les lignes de niveau. On compare ensuite ses performances sur des images de synthèse et sur des images satellitaires Quickbird.

Nous pensons que cette étude contient plusieurs contributions originales :

- La modélisation du problème de détection de changements qu'on a proposé dans l'introduction est originale. Nous espérons que ce formalisme pourra inspirer des travaux futurs.
- Nous évaluons l'invariance des lignes de niveau sous un angle nouveau. Nous modélisons complètement le principe de formation de l'image depuis les propriétés de réflectance des surfaces, les conditions d'illumination et les transformées en perspective. Cette vision des choses nous mène à redéfinir les lignes de niveau. Nous montrons que cette nouvelle définition fournit de meilleurs résultats d'invariance en théorie et en pratique.
- Nous caractérisons précisément l'espace des scènes qui donne lieu à une invariance des lignes de niveau. Nous pensons que ce résultat théorique aide à savoir rapidement si l'utilisation des lignes de niveau est sensée ou non pour le développement d'une application.
- D'un point de vue pratique, on propose un algorithme de détection de changements. On montre qu'il a une efficacité plus grande que les approches généralement utilisées. De plus, il fonctionne en temps réel. Nous pensons que cet algorithme peut servir en l'état comme une brique élémentaire d'un système de détection de changement plus evolué. Il peut aussi servir à assister un opérateur pour faire une détection de changements manuelle (nous avons pu l'expérimenter pendant cette thèse).

Chapitre 2

La formation des images : perspective et modèle de Phong

Dans ce chapitre, nous présentons brièvement le modèle de formation d'images que nous considérons dans cette partie. Nous définissons précisément ce qu'est une scène et ce que signifie conditions de prises de vue.

2.1 Notations

Nous commençons par présenter plusieurs notations propres à cette partie. Soit Ω un sous-ensemble connexe de $\mathbb{R}^2$. Soit $u : \Omega \to \mathbb{R}$ une fonction C^2. $\nabla u = [u_1, u_2]$ représente le gradient de u, $\nabla^2 u = \begin{bmatrix} u_{11} & u_{12} \\ u_{12} & u_{22} \end{bmatrix}$ représente sa Hessienne. Soit $p : \mathbb{R}^2 \to \mathbb{R}^2$, $J(p)$ représente la jacobienne de p. Soient x et y des éléments de $\mathbb{R}^n$. $x \mathbin{/\!/} y$ signifie que x et y sont colinéaires (0 est colinéaire à tout élément de $\mathbb{R}^n$). Soit $\omega \subset \Omega$. On note $\overline{\omega}$ la fermeture de ω (vis-à-vis de la topologie induite par la métrique euclidienne). L'intérieur de ω (défini comme le plus grand ouvert inclus dans ω) sera désigné indifféremment par $\mathring{\omega}$ ou int(ω). $\mu_{\mathbb{R}^n}$ est la mesure de Lebesgue dans $\mathbb{R}^n$. $\mathcal{M}_{m,n}$ correspond à l'espace des matrices à m lignes et n colonnes.

2.2 Perspective, scène et lumière

Les notations qui suivent sont illustrées sur la figure 2.1 :

- Ω représente le plan image. C'est un ouvert connexe de $\mathbb{R}^2$.
- $\mathbf{\Omega}$ représente le plan objet.
- $s : \mathbf{\Omega} \to \mathbb{R}$ désigne l'élévation de la scène.
- $N(\mathbf{x})$ représente la normale à la surface de la scène au point $(\mathbf{x}, s(\mathbf{x}))$.
- $P : (\mathbf{x}, z) \mapsto x$ est une projection 3D sur Ω. Cette projection peut être par exemple une projection en perspective ou une projection homographique. On verra que les résultats énoncés ne dépendent pas de cette projection.
- p est l'application définie par :

$$\begin{array}{rccc} p : & \mathbf{\Omega} & \to & \Omega \\ & \mathbf{x} & \mapsto & P(\mathbf{x}, s(\mathbf{x})) \end{array} \tag{2.1}$$

On suppose que p est un difféomorphisme C^1. Un difféomorphisme étant bijectif, cela implique que la caméra peut voir tous les points de la surface.

Tout au long de cette partie, on prend comme convention de noter en gras les symboles relatifs au plan objet et en police régulière les symboles relatifs au plan image. Par exemple, $\mathbf{\Omega} = p^{-1}(\Omega)$ représente le plan objet. De même si $x \in \Omega$, on définit implicitement $\mathbf{x} = p^{-1}(x)$ qui est un point de $\mathbf{\Omega}$.

Finalement, l est un vecteur de $\mathbb{R}^3_*$. $\frac{l}{|l|}$ représente la direction d'incidence de la lumière et $|l|$ représente son intensité.

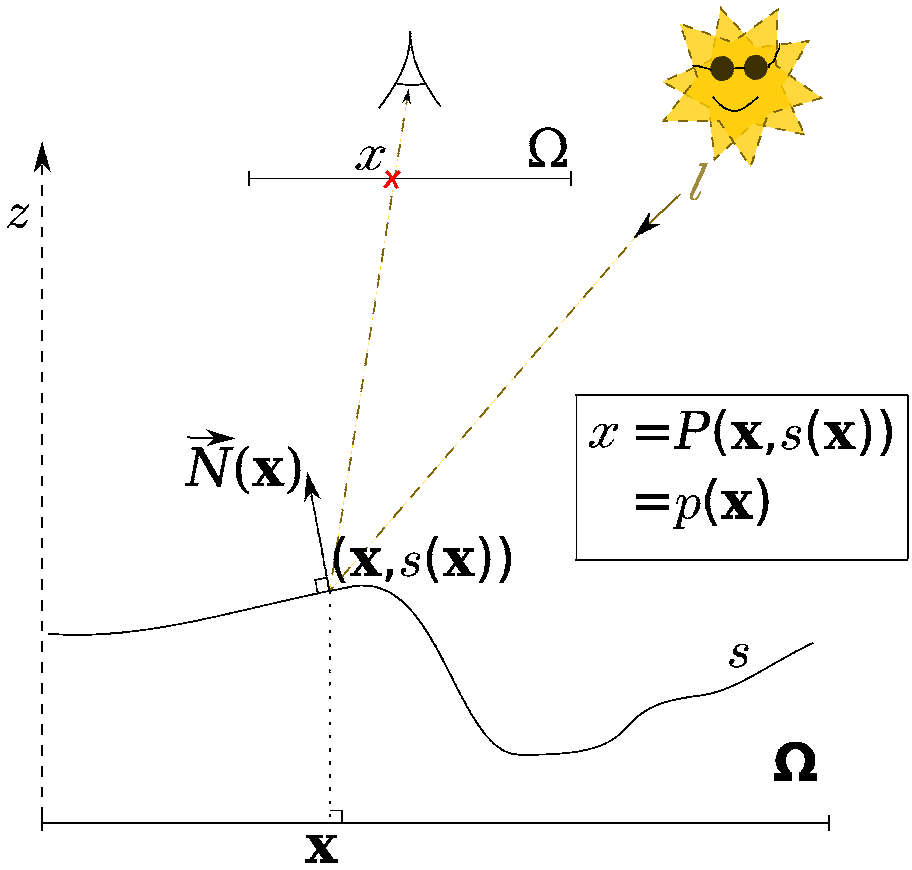

FIG. 2.1 – Notations

2.3 Hypothèses sur la surface et le modèle d'illumination

Pour modéliser les interactions entre les surfaces et la lumière, on utilise le modèle empirique de Phong [Phong, 1975]. Dans ce modèle, l'image d'une surface est restituée à partir de 3 composantes illustrées sur la figure 2.2 :

– La composante ambiante. Elle modélise la lumière qu'une surface va réémettre lorsqu'elle est illuminée par une lumière ambiante (éclairage uniforme dans toutes les directions en tout point de la scène).

– La composante diffuse. Elle correspond à une lumière qui serait réémise uniformément dans toutes les directions de l'espace. Elle ne dépend donc pas de la position de l'observateur.
– La composante spéculaire. C'est une composante qui modélise l'effet "miroir" d'une surface. Elle dépend de la position de l'observateur et de la direction d'incidence de la lumière.

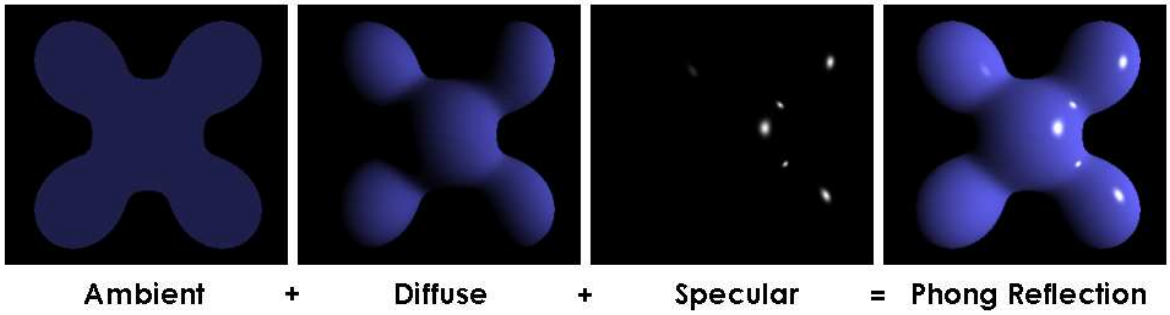

FIG. 2.2 – Les 3 composantes du modèle de Phong. Source : Wikipedia.en

Dans notre travail, on fait les hypothèses suivantes sur la surface et la lumière :

Hypothèse 2.1. *On considère que la lumière est composée de lumière ambiante d'amplitude γ (composante présente uniformément dans toute la scène) et de lumière directionnelle (tous les rayons de lumière incidente sont parallèles et d'intensité égale).*

Hypothèse 2.2. *On suppose que la surface des objets est lambertienne avec un albédo variable α : $\mathbf{\Omega} \to \mathbb{R}_*^+$ [Horn et Brooks, 1989].*

L'albédo peut être vu comme la couleur d'un objet. Un objet noir a un albédo nul (aucune lumière n'est réfléchie), tandis qu'un objet blanc a un albédo grand. Une surface lambertienne est une surface qui n'a pas de composante spéculaire. Les images des objets que l'on considère sont donc la somme de la composante ambiante et de la composante diffuse uniquement.

Hypothèse 2.3. *Afin d'éviter la présence d'ombres, on suppose que l'angle entre l et N est strictement inférieur à $\pi/2$ sur un ensemble de directions $\mathcal{L} \subset \mathbb{R}^3$ de mesure de Lebesgue non nulle. L'ensemble des illuminations possibles $[l, \alpha]$ est noté $\mathscr{L} = \mathcal{L} \times \mathbb{R}^+$.*

Avec ces hypothèses, une scène S est décrite complètement par le couple (élévation,albédo). On notera donc $S = (s, \alpha)$. Les conditions d'illumination sont complètement décrites par le vecteur $L = [l, \gamma] \in \mathscr{L}$ (lumière directionnelle l dans $\mathbb{R}^3$ et lumière ambiante γ dans $\mathbb{R}$). Sous les hypothèses précédentes, l'image u de la scène S sous les conditions d'illumination L peut s'écrire comme suit [Phong, 1975] :

$$u_{S,L}(x) = (\langle l, N(\mathbf{x})\rangle + \gamma) \cdot \alpha(\mathbf{x}) \tag{2.2}$$

où $\langle \cdot, \cdot \rangle$ représente le produit scalaire canonique de $\mathbb{R}^4$. Notons que les résultats que l'on va donner sont également valables pour le modèle de formation d'image suivant :

$$u_{S,L}(\mathbf{x}) = \varphi\left((\langle l, N(\mathbf{x})\rangle + \gamma) \cdot \alpha(\mathbf{x})\right) \tag{2.3}$$

où $\varphi : \mathbb{R} \to \mathbb{R}$ est une fonction strictement monotone qui modélise un changement global de contraste. Pour simplifier les notations, on utilise uniquement le modèle décrit dans l'équation 2.2.

Chapitre 3

Invariance des lignes de niveau aux changements d'illumination

Dans ce chapitre, nous présentons les résultats théoriques principaux de cette partie : on caractérise géométriquement les scènes qui donnent lieu à des lignes de niveau invariantes aux changements d'illumination. Ces résultats théoriques seront appliqués dans le chapitre suivant au problème pratique de la détection de changements. Ce chapitre est divisé en quatre sections :

- Premièrement, nous définissons les lignes de niveau et montrons quelques-unes de leurs propriétés.
- Nous rappelons ensuite quelques éléments de géométrie différentielle indispensables à la compréhension des résultats d'invariance.
- Nous analysons l'invariance des lignes de niveau, lorsque la scène et l'albédo sont réguliers.
- Nous analysons ensuite cette invariance lorsque la scène et l'albédo présentent des discontinuités.

3.1 Définition et propriétés des lignes de niveau

Dans cette section, nous définissons les lignes de niveaux et donnons plusieurs de leurs propriétés mathématiques. Un lecteur interessé uniquement par les résultats applicatifs de cette thèse peut se contenter de lire le paragraphe 3.1.1. Le paragraphe 3.1.2 contient plusieurs résultats qui servent essentiellement à montrer le théorème 3.26 sur l'invariance des lignes de niveau.

3.1.1 Définition

Soit $u : \Omega \to \mathbb{R}$ une fonction. On suppose que u est définie dans tout Ω. Les lignes de niveau sont définies de la façon suivante :

Définition 3.15 (Lignes de niveau)**.** *Les lignes de niveau de u sont les composantes connexes [Janisch, 1984] des ensembles* $\{x \in \Omega, u(x) = \alpha\}$

Cette définition est généralement utilisée pour des fonctions C^1 dont le gradient ne s'annule pas. Dans ce cas, on peut montrer que les lignes de niveau sont des courbes de Jordan fermées, ou des courbes dont les extrémités appartiennent à la frontière de Ω. Pour la classe générale de fonctions que l'on considère, les lignes de niveau peuvent être n'importe quel objet connexe du plan tel que des

courbes, des morceaux de plans, des points, des fractals... Notons que même des fonctions infiniment régulières ne donnent pas lieu à des lignes de niveau "simples" comme le montre le théorème suivant [Agr, 2002], p.97 :

Théorème 3.24. *Tout fermé de $\mathbb{R}^n$ s'exprime comme le zéro d'une fonction C^∞.*

Le terme "ligne" est donc un abus de langage. Les auteurs de [Caselles *et al.*, 1999a; Guichard et Morel, 2001] donnent une autre définition des lignes de niveau :

Définition 3.16 (lignes de niveau [Caselles *et al.*, 1999a; Guichard et Morel, 2001])**.** *Soit u une fonction semi-continue supérieurement. Les lignes de niveau de u sont définies comme les frontières des composantes connexes des ensembles de niveaux $\{x \in \Omega,\ u(x) \leq \lambda\}$.*

Les lignes de niveau ainsi définies sont des courbes ; Cependant, nous verrons plus loin que cette définition ne garantit pas l'invariance aux changements d'illumination, même pour des scènes très simples.

3.1.2 Quelques résultats mathématiques

Dans ce paragraphe, nous caractérisons les couples d'images qui ont les mêmes lignes de niveau. Les lemmes présentés dans ce paragraphe serviront plus tard pour caractériser les scènes donnant lieu à des lignes de niveau invariantes par changements d'illumination. Un lecteur intéressé essentiellement par les résultats applicatifs peut passer outre ces quelques résultats.

Les lemmes 3.9 et 3.10 servent à montrer le lemme 3.12 qui sera important par la suite.

Lemme 3.9. *Soit $C : \Omega \to \mathbb{R}^4_*$ une application (nous ne faisons aucune hypothèse de régularité).*
Soit $L \in \mathscr{L}$. On note $L^\perp$ l'hyperplan orthogonal à L.
Soit $\omega_L = \text{int}(\{x \in \Omega, C(x) \in L^\perp\})$.
Pour presque tout $L \in \mathscr{L} \subset \mathbb{R}^4$, $\omega_L = \emptyset$.

Démonstration. On note $Y = \{L \in \mathscr{L}, \omega_L \neq \emptyset\}$. Soit $\Omega_\mathbb{Q} = \{\mathbb{Q}^2 \cap \Omega\}$. $\Omega_\mathbb{Q} \subset \mathbb{Q}^2$, $\Omega_\mathbb{Q}$ est alors un ensemble dénombrable.

Soit $a_i \in \Omega_\mathbb{Q}$ et $Y_i = \{L \in \mathscr{L}, a_i \in \omega_L\}$. Y_i est un sous-ensemble inclus dans un sous-espace vectoriel strict de $\mathbb{R}^4$. Dans le cas contraire, il existerait 4 éléments de Y_i, L_1, L_2, L_3, L_4 formant une base de $\mathbb{R}^4$. Étant donné que $C(a_i) \perp L_j \ \ \forall j \in \{1, 2, 3, 4\}$, cela impliquerait $C(a_i) = 0$, ce qui contredirait $C(a_i) \in \mathbb{R}^4_*$.

La remarque précédente implique que $\mu_{\mathbb{R}^4}(Y_i) = 0$ et donc $\mu_{\mathbb{R}^4}(\bigcup_{a_i \in \Omega_\mathbb{Q}} Y_i) = 0$. De plus $\bigcup_{a_i \in \Omega_\mathbb{Q}} Y_i = Y$ (car chaque ouvert non-vide ω_L contient un élément de $\mathbb{Q}^2$). Par conséquent, $\mu_{\mathbb{R}^4}(Y) = 0$. □

Lemme 3.10. *Soit $\omega \subset \Omega$ un ouvert. Soient u_1 et u_2 deux fonctions $C^1(\Omega)$ telles que : $\forall x \in \omega$, $\nabla u_1(x) \,/\!/\, \nabla u_2(x)$, $\nabla u_1(x) \neq 0$ et $\nabla u_2(x) \neq 0$. Alors u_1 et u_2 ont les mêmes lignes de niveau sur ω.*

Démonstration. ∇u_1 et ∇u_2 étant non-nulles sur ω, le théorème des fonctions implicites implique que les lignes de niveau de u_1 et u_2 sont des courbes C^1. De plus, ces courbes peuvent être définies uniquement grâce à leurs normales respectives : $\frac{\nabla u_1}{|\nabla u_1|}$ et $\frac{\nabla u_2}{|\nabla u_2|}$ qui sont égales. □

Lemme 3.11. *Soient u_1 et u_2 deux fonctions $C^1(\Omega)$. S'il existe un point $x \in \Omega$ tel que $\nabla u_1(x)$ n'est pas colinéaire à $\nabla u_2(x)$, alors u_1 et u_2 n'ont pas les mêmes lignes de niveau sur Ω.*

Démonstration. Comme $\nabla u_1(x)$ et $\nabla u_2(x)$ ne sont pas colinéaires, ils sont non nuls tous les deux. Comme u_1 et u_2 sont C^1, il existe un ouvert ω' sur lequel $\nabla u_1 \neq 0$ et $\nabla u_2 \neq 0$. Comme la fonction $\frac{x}{|x|}$ est continue sur $\mathbb{R}^2 - \{(0,0)\}$, il existe un ouvert $\omega \subset \omega'$ sur lequel $\frac{\nabla u_1}{|\nabla u_1|}$ n'est pas colinéaire à $\frac{\nabla u_2}{|\nabla u_2|}$. On peut appliquer le théorème des fonction implicites sur cet ouvert et on se rend compte que l'équation de la ligne de niveau de u_1 passant par x et celle de u_2 sont différentes sur ω, ce qui conclut la preuve. □

Nous voilà prêt à montrer le résultat le plus important (et difficile) du chapitre :

Lemme 3.12. *Soient u_1 et u_2 deux fonctions $C^1(\Omega)$ telles que $\forall x \in \Omega,\ \nabla u_1(x) \,/\!/\, \nabla u_2(x)$. Les propositions suivantes sont équivalentes :*

- *Prop. 1 : u_1 et u_2 n'ont pas les mêmes lignes de niveau ;*
- *Prop. 2 : Il existe un ouvert non-vide $\omega \subset \Omega$ tel que :*

$$\forall x \in \omega, \quad \begin{cases} \nabla u_1(x) = 0 \\ \nabla u_2(x) \neq 0 \end{cases} \tag{3.1}$$

$$\text{ou } \forall x \in \omega, \quad \begin{cases} \nabla u_1(x) \neq 0 \\ \nabla u_2(x) = 0 \end{cases} . \tag{3.2}$$

Démonstration. La démonstration *Prop. 2* $\Rightarrow$ *Prop. 1* est simple :

Etant donné que $\nabla u_1 = 0$ sur ω, u_1 est constante sur ω et est donc incluse dans une ligne de niveau de u_1, alors que $\nabla u_2 \neq 0$ implique que $\exists (x,y) \in \omega^2$ tels que $u_2(x) \neq u_2(y)$. En conséquence, ω n'est pas inclus dans une ligne de niveau de u_2.

On montre maintenant *Prop. 1* $\Rightarrow$ *Prop. 2* par l'absurde.

On suppose qu'il existe deux fonctions $C^1(\Omega)$, u_1 et u_2 telles que :

(H_1) $\forall x \in \Omega,\ \nabla u_1(x) \,/\!/\, \nabla u_2(x)$

(H_2) il n'existe pas d'ensemble ouvert ω tel que :

$$\forall x \in \omega, \begin{cases} \nabla u_1(x) = 0 \\ \nabla u_2(x) \neq 0 \end{cases} \tag{3.3}$$

$$\text{ou } \forall x \in \omega, \begin{cases} \nabla u_1(x) \neq 0 \\ \nabla u_2(x) = 0 \end{cases} \tag{3.4}$$

(H_3) Les lignes de niveau de u_1 sont différentes des lignes de niveau de u_2

Soient :

$$\Omega_{1,0} = \{x \in \Omega, \nabla u_1(x) = 0\}$$

$$\Omega_{2,0} = \{x \in \Omega, \nabla u_2(x) = 0\}$$

1. *Dans un premier temps, on remarque que $\mathring{\Omega}_{1,0} = \mathring{\Omega}_{2,0}$.*
   En effet, supposons :
   $$\begin{cases} \mathring{\Omega}_{1,0} \subset \Omega_{2,0} \\ \mathring{\Omega}_{2,0} \subset \Omega_{1,0} \end{cases} \tag{3.5}$$
   $\mathring{\Omega}_{1,0}$ étant un ouvert, on a $\mathring{\Omega}_{1,0} \subset \mathring{\Omega}_{2,0}$. En appliquant le même raisonnement, on obtient $\mathring{\Omega}_{2,0} \subset \mathring{\Omega}_{1,0}$, et finalement $\mathring{\Omega}_{1,0} = \mathring{\Omega}_{2,0}$.

Supposons maintenant que le système d'équations 3.5 est faux :

$$\mathring{\Omega}_{1,0} \not\subset \Omega_{2,0} \text{ or } \mathring{\Omega}_{2,0} \not\subset \Omega_{1,0} \tag{3.6}$$

En conséquence, quitte à inverser les indices, il existe $x \in \mathring{\Omega}_{1,0} \setminus \Omega_{2,0}$ tel que $\nabla u_1(x) = 0$ et $\nabla u_2(x) \neq 0$. Par conséquent, d'après la continuité de ∇u_2 il existe un ouvert $\omega_x \subseteq \mathring{\Omega}_{1,0}$ contenant x tel que $\nabla u_2 \neq 0$ sur ω_x. Par ailleurs $\omega_x \subseteq \mathring{\Omega}_1^0$ implique que $\nabla u_1(x) = 0$ sur ω_x. Ce qui contredit (H_2).

2. On note $\tilde{\Omega}^+ = \Omega \setminus \overline{\mathring{\Omega}_{1,0}}$. *Sur* $\tilde{\Omega}^+$, *les lignes de niveau* u_1 *sont différentes de celles de* u_2.
   Encore une fois, on utilise une démonstration par l'absurde. On suppose que les lignes de niveau de u_1 sont les mêmes que les lignes de niveau de u_2 sur $\tilde{\Omega}^+$. D'après le paragraphe précédent, ∇u_1 et ∇u_2 sont nuls sur $\mathring{\Omega}_{1,0} = \mathring{\Omega}_{2,0}$. Par continuité, ceci est également valable sur $\overline{\mathring{\Omega}_{1,0}}$. Par conséquent u_1 et u_2 sont toutes deux constantes sur chaque composante connexe U_i de $\overline{\mathring{\Omega}_{1,0}}$. Chaque U_i est une ligne de niveau u_1 et u_2 sur $\overline{\mathring{\Omega}_{1,0}}$. Considérons maintenant une ligne de niveau A de u_1 dans $\tilde{\Omega}^+$. On peut considérer deux cas :
   - Soit $\forall i,\ U_i \cap A = \emptyset$. Dans ce cas, A est clairement une ligne de niveau de u_1 dans Ω. Le même raisonnement pouvant être appliqué à u_2, A est également une ligne de niveau de u_2 dans Ω.
   - Soit $\exists i,\ U_i \cap A \neq \emptyset$. A et U_i étant tous deux des ensembles connexes, $A \cup U_i$ est un ensemble connexe. u_1 étant continue, il vient que u_1 est constante sur $A \cup U_i$. De même, u_2 est constante sur $A \cup U_i$.

   Les lignes de niveau de u_1 et u_2 sont donc les mêmes sur Ω ce qui est contradictoire avec nos hypothèses.
3. Soit $\omega_1 \subset \Omega$ un ouvert tel que $\nabla u_1 = 0$ sur ω_1. Nécessairement, $\omega_1 \subset \mathring{\Omega}_1$ et donc $\omega_1 \cap \tilde{\Omega}^+ = \emptyset$. Il n'existe donc pas d'ouvert de $\tilde{\Omega}^+$ sur lequel $\nabla u_1 = 0$. Ce résultat est également valable avec u_2. Soit $\Omega_1^+ = \{x, \nabla u_1(x) \neq 0\}$ et $\Omega_2^+ = \{x, \nabla u_2(x) \neq 0\}$. Ω_1^+ et Ω_2^+ sont des ouverts denses dans $\tilde{\Omega}_+$. D'après le théorème de Baire, il vient que $\Omega_1^+ \cap \Omega_2^+$ est un ouvert dense dans $\tilde{\Omega}^+$.
4. D'après le Lemme 3.10, on sait que les lignes de niveau de u_1 et u_2 sont identiques sur $\Omega_1^+ \cap \Omega_2^+$. La continuité de u_1 et u_2 assure qu'elles sont également identiques sur $\overline{\Omega_1^+ \cap \Omega_2^+} = \tilde{\Omega}^+$. Ce résultat est en contradiction avec la conclusion du 2).

□

3.2 Rappels de géométrie différentielle

Nous commençons par rappeler brièvement deux définitions importantes en géométrie différentielle :

Définition 3.17 (Courbure gaussienne)**.** *La courbure gaussienne d'une surface de* $\mathbb{R}^3$ *est définie comme le produit de ses deux courbures principales. Avec nos notations, la courbure gaussienne de s est définie comme* $\det(\nabla^2 s)$.

Définition 3.18 (Surface développable)**.** *On appelle surface développable une surface* C^2 *dont la courbure gaussienne est nulle partout [Spivak, 1999].*

Les exemples classiques de surfaces développables sont les plans, les cylindres et les cônes. Une surface développable possède les propriétés suivantes [Spivak, 1999] :

– Chaque point de la surface appartient à une ligne (la génératrice) entièrement contenue dans la surface.
– Le plan tangent à la surface est le même en chaque point de la génératrice.

3.3 Invariance des lignes de niveau dans le cas régulier

Dans ce qui suit, nous caractérisons les scènes qui présentent des lignes de niveau invariantes à l'illumination lorsque s est C^2 et α est C^1. On peut montrer que sous ces hypothèses de régularité, $u_{S,L}$ est C^1 pour tout $L \in \mathscr{L}$. Ceci va nous permettre de calculer explicitement le gradient spatial de $u_{S,L}$. L'espace de scènes dans ce paragraphe est défini comme suit :

Définition 3.19. *Θ est l'ensemble des scènes $S = (s, \alpha)$ telles que s est C^2 développable, α est C^1 et varie uniquement dans la direction orthogonale aux génératrices de s. Aux points où s est plane ($det(\nabla^2 s) = 0$), α peut varier dans n'importe quelle direction.*

Nous allons montrer l'invariance des lignes de niveau sur cet ensemble étape par étape. Pour commencer, on se concentre sur une propriété locale des images : la direction du gradient. Plusieurs algorithmes invariants aux changements de contraste reposent sur cette caractéristique. Par exemple, les auteurs de [Lisani et Morel, 2003] l'utilisent en détection de changements, les auteurs de [Cao et Bouthemy, 2006; Chen *et al.*, 2000; Chen *et al.*, 2008] l'utilisent pour calculer des mesures de dissimilarité entre deux images et les auteurs de [Droske et Rumpf, 2004] l'utilisent pour effectuer des recalages non rigides d'images. Le théorème suivant caractérise les scènes pour lesquelles ces approches sont complètement justifiées mathématiquement :

Théorème 3.25. *Soient $s \in C^2(\Omega)$ et $\alpha \in C^1(\Omega)$. Les propositions suivantes sont équivalentes :*
– *Prop. 1 :* $\forall (L_1, L_2) \in \mathscr{L} \times \mathscr{L},\ \forall x \in \Omega,\ \nabla u_{S,L_1}(x) \mathbin{/\!/} \nabla u_{S,L_2}(x)$
– *Prop. 2 :* $S \in \Theta$

Ce théorème montre que la direction du gradient, dans le cas régulier, est une caractéristique invariante aux changements d'illumination si et seulement si la scène appartient à Θ. Lorsque nous rédigions ces résultats, nous avons découvert que des résultats analogues ont été obtenus dans [Chen *et al.*, 2000; Chen *et al.*, 2008]. Les auteurs montrent que la direction du gradient est presque invariante aux changements d'illumination dans le sens où sa distribution par rapport à l'orientation de la lumière est concentrée le long d'un vecteur pour la plupart de scènes.

Démonstration. On exprime d'abord le gradient de u en fonction de s, α et (l, γ). Des considérations simples en géométrie donnent $N(\mathbf{x}) = \Psi(\nabla s(\mathbf{x})) = \frac{(-s_1(\mathbf{x}), -s_2(\mathbf{x}), 1)}{\sqrt{s_1^2(\mathbf{x}) + s_2^2(\mathbf{x}) + 1}}$, où s_1 et s_2 sont les dérivées partielles de s suivant les deux axes du plan objet. Soit $\mathbf{x} = p^{-1}(x)$. En utilisant la règle de dérivation des fonctions composées et l'équation 2.2, on obtient :

$$\nabla u_{S,L}(x) = \left(\underbrace{l}_{\in \mathcal{M}_{1,3}} \cdot \left[\underbrace{\Psi'(\nabla s(\mathbf{x}))}_{\in \mathcal{M}_{3,2}} \cdot \underbrace{\nabla^2 s(\mathbf{x})}_{\in \mathcal{M}_{2,2}} \underbrace{\alpha(\mathbf{x})}_{\in \mathbb{R}} + \underbrace{\Psi(\nabla s(\mathbf{x}))}_{\in \mathcal{M}_{3,1}} \cdot \underbrace{\nabla \alpha(\mathbf{x})}_{\in \mathcal{M}_{1,2}} \right] + \underbrace{\gamma \nabla \alpha(\mathbf{x})}_{\in \mathcal{M}_{1,2}} \right) \cdot \underbrace{J(p^{-1})(x)}_{\in \mathcal{M}_{2,2}}$$

Cette équation peut être réécrite comme suit :

$$\nabla u_{S,L}(x) = (l \cdot A(\mathbf{x}) + \gamma \nabla \alpha(\mathbf{x})) \cdot J(p^{-1})(x) \tag{3.7}$$

avec :

$$A(\mathbf{x}) = \left([M_1, M_2] \cdot \begin{bmatrix} s_{11} & s_{12} \\ s_{12} & s_{22} \end{bmatrix} + N \cdot [\alpha_1, \alpha_2] \right)(\mathbf{x}) \tag{3.8}$$

$M_1(\mathbf{x})$ et $M_2(\mathbf{x}) \in \mathcal{M}_{3,1}$ sont les deux colonnes de $\Psi'(\nabla s(\mathbf{x}))\alpha(\mathbf{x})$ et $N(\mathbf{x}) = \Psi(\nabla s(\mathbf{x}))$.

Ces résultats préliminaires étant donnés nous pouvons maintenant démontrer *Prop. 1* $\Rightarrow$ *Prop. 2*. Soit S une scène telle que :

$$\forall (L_1, L_2) \in \mathscr{L} \times \mathscr{L}, \ \ \forall x \in \Omega, \ \ \nabla u_{S,L_1}(x) \ /\!/ \ \nabla u_{S,L_2}(x) \tag{3.9}$$

Soit $D : \Omega \to \mathbb{S}^2$ la direction d'invariance de $\nabla u_{S,L}$. Aux points x où $\forall L \in \mathscr{L}, \ \ \nabla u_{S,L}(x) = 0$, D peut correspondre à n'importe quel vecteur de $\mathbb{R}^2$.

Comme l'équation (3.9) est vraie pour tout $([l_1, \gamma_1], [l_2, \gamma_2]) \in \mathscr{L} \times \mathscr{L}$, elle est vraie en particulier pour $\gamma_1 = \gamma_2 = 0$. En conséquence, on néglige le terme $\gamma \nabla \alpha$ dans l'équation (3.7). La relation (3.9) peut alors s'écrire :

$$\forall l \in \mathcal{L}, \ \forall x \in \Omega, \quad l \cdot A(\mathbf{x}) \cdot J(p^{-1})(x) \quad /\!/ \quad D(x) \tag{3.10}$$

p étant un difféomorphisme, $J(p^{-1})(x)$ est une matrice 2×2 inversible. La propriété (3.10) ne dépend donc pas de $J(p^{-1})$ et est vraie seulement si la matrice $A(\mathbf{x})$ possède deux lignes colinéaires. Ceci peut être réécrit de la façon suivante :

$$\begin{cases} (s_{11}M_1 + s_{12}M_2 + \alpha_1 N)(\mathbf{x}) &= a(\mathbf{x})C(\mathbf{x}) \\ (s_{12}M_1 + s_{22}M_2 + \alpha_2 N)(\mathbf{x}) &= b(\mathbf{x})C(\mathbf{x}) \end{cases} \tag{3.11}$$

où $(a, b) : \Omega \to \mathbb{R}^2$ et $C : \Omega \to \mathcal{M}_{3,1}$ sont des fonctions C^0. Un calcul élémentaire (mais fastidieux) mène à $\det([M_1, M_2, N](\mathbf{x})) > 0$, de telle façon que les vecteurs $M_1(\mathbf{x}), M_2(\mathbf{x}), N(\mathbf{x})$ forment une base de $\mathbb{R}^3$. Par conséquent, Le système d'équations (3.11) implique :

$$\begin{pmatrix} s_{11} \\ s_{12} \\ \alpha_1 \end{pmatrix}(\mathbf{x}) \ /\!/ \ \begin{pmatrix} s_{12} \\ s_{22} \\ \alpha_2 \end{pmatrix}(\mathbf{x}) \tag{3.12}$$

de sorte que :

$$\begin{pmatrix} s_{11} \\ s_{12} \end{pmatrix}(\mathbf{x}) \ /\!/ \ \begin{pmatrix} s_{12} \\ s_{22} \end{pmatrix}(\mathbf{x}) \ /\!/ \ \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix}(\mathbf{x}) \tag{3.13}$$

Finalement, l'équation (3.13) conduit à $\forall \mathbf{x} \in \mathbf{\Omega}$, $\det(\nabla^2 s(\mathbf{x})) = 0$ ce qui implique que s est développable. On peut vérifier facilement que le vecteur propre de $\nabla^2 s(\mathbf{x})$ est $\begin{pmatrix} s_{12} \\ s_{22} \end{pmatrix}(\mathbf{x})$. Ce vecteur propre pointe dans la direction orthogonale à la génératrice de s passant par le point $\mathbf{x}$. L'équation 3.13 indique donc que $\alpha(\mathbf{x})$ varie dans la direction orthogonale à cette génératrice. Pour les points tels que $\nabla^2 s(\mathbf{x}) = 0$, α peut varier dans n'importe quelle direction.

Montrons maintenant *Prop. 2* $\Rightarrow$ *Prop. 1*. Soit $S = (s, \alpha) \in \Theta$. En conséquence, $\forall \mathbf{x} \in \mathbf{\Omega}$, $\det(\nabla^2 s(\mathbf{x})) = 0$. Il existe donc $(c_1, c_2, a, b) : \mathbf{\Omega} \to \mathbb{R}^4$ tels que :

$$\begin{bmatrix} s_{11} & s_{12} \\ s_{12} & s_{22} \end{bmatrix} = \begin{bmatrix} c_1 a & c_2 a \\ c_1 b & c_2 b \end{bmatrix} \tag{3.14}$$

avec $c_1 b = c_2 a$. De plus, comme le gradient varie orthogonalement aux génératrices, il existe c_3 tel que :

$$[\alpha_1, \alpha_2] = c_3[a, b] \tag{3.15}$$

En utilisant les équations (3.14) et (3.15), on peut écrire la matrice A (voir équation (3.8)) sous la forme :

$$A = [aC, bC] \tag{3.16}$$

avec $C = M_1 c_1 + M_2 c_2 + N c_3$. On peut ensuite simplifier l'équation (3.7) sous la forme :

$$\nabla u_{S,L}(x) = ((\langle l, C\rangle + \gamma c_3)[a, b])(\mathbf{x}) \cdot J(p^{-1})(x) \tag{3.17}$$

ce qui nous amène à $\nabla u_{S,L}(x) \;/\!/\; [a, b](\mathbf{x}) \cdot J(p^{-1})(x)$ pour tout $L \in \mathscr{L}$.

□

On se concentre maintenant sur les résultats concernant l'invariance des lignes de niveau. Le théorème 3.25 permet de montrer aisément la proposition suivante :

Corollaire 3.1. *Les lignes de niveau sont invariantes aux changements d'illumination seulement si la scène appartient à* Θ.

Démonstration. Il s'agit d'une conséquence directe du théorème 3.25 et du lemme 3.11. □

Naturellement, nous aimerions maintenant montrer la réciproque du corollaire 3.1. Hélas, elle est fausse. Toutes les scènes appartenant à Θ n'ont pas des lignes de niveau invariantes aux changements d'illumination. Par exemple, la figure 3.1 montre un cône avec un albédo constant. Si la lumière incidente est dirigée exactement dans la direction de l'axe du cône (gauche), la radiométrie du cône est uniforme et son image est en conséquence composée d'une seule ligne de niveau. Dans tous les autres cas, (image de droite), les lignes de niveau correspondent aux génératrices du cône.

Nous pouvons énoncer un résultat plus faible, qui complète le corollaire 3.1. La preuve de ce théorème fait intervenir les lemmes du chapitre précédent :

Théorème 3.26. *Soit* $S \in \Theta$. *Pour presque toutes les paires de conditions d'illumination* $(L_1, L_2) \in \mathscr{L} \times \mathscr{L}$ *(par rapport à la mesure de Lebesgue de* $\mathbb{R}^4 \times \mathbb{R}^4$*) les lignes de niveau de* u_{S,L_1} *sont les mêmes que celles de* u_{S,L_2}

Démonstration. On suppose $S \in \Theta$. Ceci implique (cf. démonstration du théorème 3.25) que $\nabla u_{S,L} = \langle L, C\rangle \cdot [a, b]$ avec $C : \Omega \to \mathbb{R}^4_*$ et $[a, b] : \Omega \to \mathbb{R}^2$, des fonctions $C^0(\Omega)$. On a donc $\forall (L_1, L_2) \in \mathscr{L} \times \mathscr{L}$, $\forall x \in \Omega$, $\nabla u_{S,L_1}(x) \;/\!/\; \nabla u_{S,L_2}(x)$.

Soit $\Omega^+ = \{x \in \Omega, [a, b](x) \neq 0\}$. Cet ensemble est un ouvert car a et b sont $C^0(\Omega)$. Soit

$$\omega_L = \text{int}(\{x \in \Omega^+, \nabla u_{S,L}(x) = 0\}) \tag{3.18}$$

Cet ensemble est également caractérisé par :

$$\omega_L = \text{int}(\{x \in \Omega^+, C(x) \in L^\perp\}) \tag{3.19}$$

D'après le lemme 3.9, pour presque tout $L_1 \in \mathscr{L}$, $\omega_{L_1} = \emptyset$. En conséquence, pour presque tout $L_1 \in \mathscr{L}$, $\forall \omega$ ouvert de Ω^+, $\exists x \in \omega$ tel que $\nabla u_{S,L_1}(x) \neq 0$. La continuité de $\nabla u_{S,L_1}$ implique l'existence

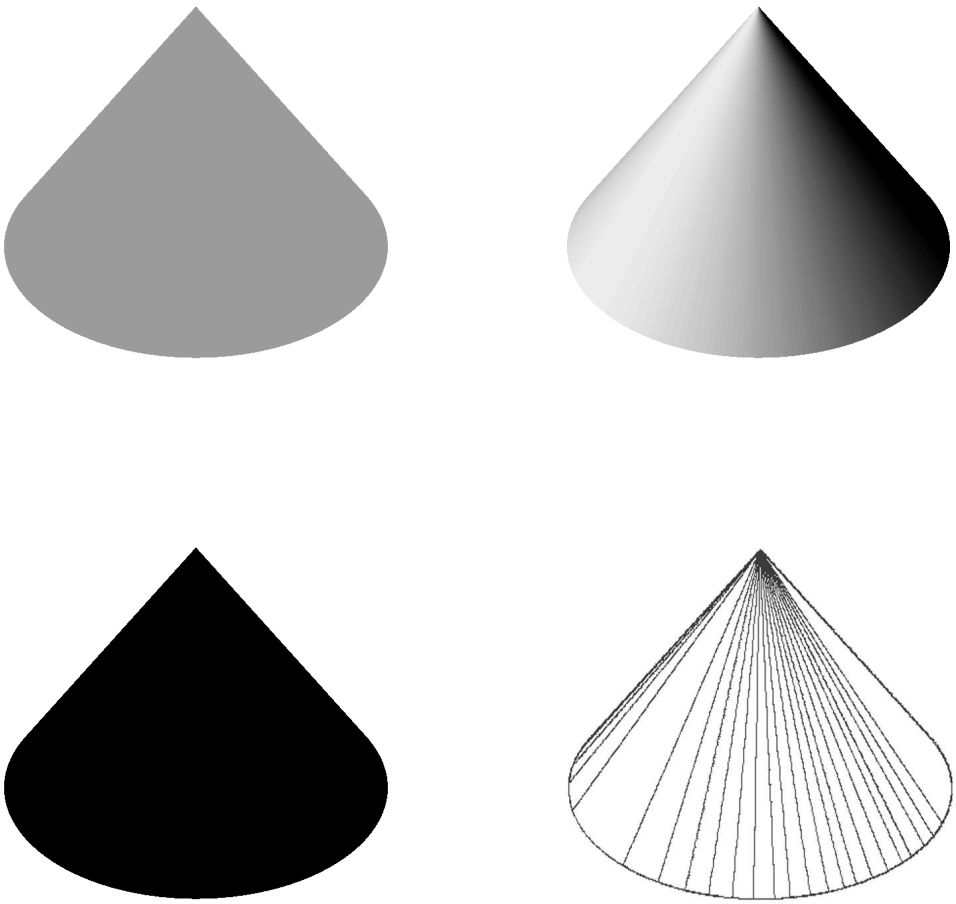

FIG. 3.1 – Haut : images d'un cône sous deux lumières d'incidences différentes. Bas : quelques unes de leurs lignes de niveau.

d'un ouvert $\omega_x \subset \Omega^+$ contenant x et sur lequel $\nabla u_{S,L_1} \neq 0$. De plus, pour presque tout $L_2 \in \mathscr{L}$, $\omega_{L_2} = \emptyset$, de sorte qu'il existe $x' \in \omega_x$ tel que $\nabla u_2(x') \neq 0$.

Finalement, pour presque tout $(L_1, L_2) \in \mathscr{L} \times \mathscr{L}$, $\forall \omega \subseteq \Omega^+,\ \exists x' \in \omega$ tel que :

$$\begin{cases} \nabla u_1(x') \neq 0 \\ \nabla u_2(x') \neq 0 \end{cases}$$

De plus, $\forall x \in \Omega \setminus \Omega^+$, $\nabla u_1(x) = \nabla u_2(x) = 0$.

On remarque par ailleurs que la contraposée du lemme 3.12 est :

$$u_1 \text{ et } u_2 \text{ ont les mêmes lignes de niveau} \Leftrightarrow \tag{3.20}$$

$$\forall \omega \text{ ouvert de } \Omega^+,\ \exists x \in \omega, \tag{3.21}$$

$$\begin{cases} \nabla u_1(x) \neq 0 \\ \nabla u_2(x) \neq 0 \end{cases} \text{ or } \begin{cases} \nabla u_1(x) = 0 \\ \nabla u_2(x) = 0 \end{cases} \tag{3.22}$$

On en déduit que pour presque tout $(L_1, L_2) \in \mathscr{L} \times \mathscr{L}$, u_{S,L_1} et u_{S,L_2} ont les mêmes lignes de niveau. □

Ce théorème et le corollaire 3.1 montrent qu'il existe "presque" une équivalence entre ces deux propositions :

– La scène a des lignes de niveau invariantes.
– La scène appartient à Θ.

3.4 Invariance des lignes de niveau dans le cas non régulier

Dans la section précédente, nous avons caractérisé géométriquement toutes les scènes qui produisent des lignes de niveau invariantes. Ces scènes sont celles qui appartiennent à Θ. Malheureusement, l'espace Θ contient trop peu de surfaces pour modéliser des scènes réalistes [1]. On pourrait donc conclure que les lignes de niveau ne sont pas un bon outil pour assurer l'invariance aux changements d'illumination. En réalité nous allons voir que les lignes de niveau sont quasi invariantes pour un espace de scène plus large que le précédent. Nous nous intéressons dans cette section au cas où S est une fonction C^2 développable par morceaux et que son albédo varie orthogonalement aux génératrices sur chaque morceau. On commence par donner une définition précise de cet espace :

Définition 3.20. *Ξ est l'espace des scènes $\overline{S} = (\overline{s}, \overline{\alpha})$ tel qu'il existe un ensemble fini $\{\omega_i\}_{i \in I}$ et une scène $S = (s, \alpha)$ qui satisfont :*

– *$\forall i \in I$, $\boldsymbol{\omega_i} \subset \boldsymbol{\Omega}$ est un ouvert connexe de mesure non-nulle.*
– *$\forall (i, j)$, $\boldsymbol{\omega_i} \cap \boldsymbol{\omega_j} = \emptyset$.*
– *$\cup_{i \in I} \overline{\boldsymbol{\omega}}_i = \boldsymbol{\Omega}$.*
– *$\forall i$, $S|_{\omega_i} \in \Theta$ (la restriction de S à $\boldsymbol{\omega_i}$ appartient à Θ).*

[1] Une exception notable est celle des documents courbés. Une feuille de papier courbée est une surface développable. On peut donc imaginer utiliser le théorème 3.26 pour restaurer les images de documents courbés, et ce indépendamment des conditions d'éclairement. Cette idée a déjà été exploitée par quelques auteurs dans le domaine du shape from shading [Courteille *et al.*, 2007; Tan *et al.*, 2006; Wada *et al.*, 1997]

– *Finalement, on suppose que $S|_{\omega_i}$ (de même que $N|_{\omega_i}$) admet une limite à la frontière de ω_i. Ceci permet de définir $\overline{S} = (\overline{s}, \overline{\alpha})$ partout. Par exemple, on peut la définir comme suit :*

$$\begin{cases} (\overline{s}, \overline{\alpha})(\mathbf{x}) = (s, \alpha)(\mathbf{x}) & \text{si } \mathbf{x} \in \cup_{i \in I} \omega_i \\ \begin{cases} \overline{s}(\mathbf{x}) = \lim\limits_{r \to 0} \sup\limits_{\mathbf{y} \in B(\mathbf{x}, r)} s(\mathbf{y}) \\ \overline{\alpha}(\mathbf{x}) = \lim\limits_{r \to 0} \sup\limits_{\mathbf{y} \in B(\mathbf{x}, r)} \alpha(\mathbf{y}) \end{cases} & \text{sinon} \end{cases}$$

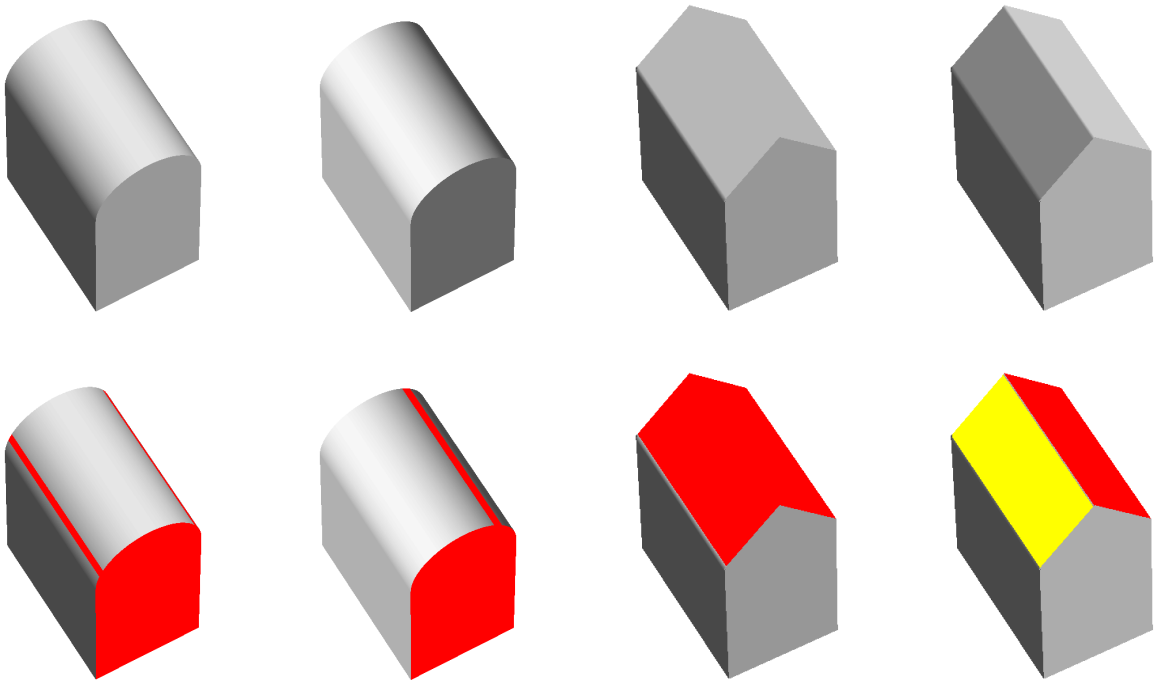

FIG. 3.2 – Exemples de non-invariance des lignes de niveau dans le cas C^2 par morceaux. Les éléments colorés représentent des lignes de niveau non singulières.

Les propriétés d'invariance que l'on va donner sont plus faibles que précédemment. La figure 3.2 illustre avec deux exemples pourquoi on n'obtient pas une invariance complète. Sur la partie gauche, la plupart des lignes de niveau du toit en forme de cylindre sont des segments du toit. Selon l'orientation de la lumière, un ou plusieurs de ces segments fusionne avec le mur du bâtiment. En conséquence, les lignes de niveau ne sont pas invariantes. La partie droite de la figure 3.2 montre l'image d'un bâtiment avec un toit à deux versants plats. Si la direction de la lumière appartient au plan bisecteur de ces deux versants, ces derniers auront la même radiométrie. Pour la plupart des illuminations, le toit sera constitué de deux lignes de niveau (jaune et rouge), alors que pour un ensemble de mesure nulle d'illuminations, il sera constitué d'une seule ligne de niveau. Dans ce qui suit, on exprime ces remarques de manière formelle.

Soit ω un sous ensemble de $\mathbf{\Omega}$. On pose les notations suivantes :

– $\kappa(\mathbf{x}, L)$ est la ligne de niveau de $u_{S,L}$ telle que $x = p(\mathbf{x}) \in \kappa(\mathbf{x}, L)$;

– $\kappa_\omega(\mathbf{x}, L)$ est la ligne de niveau de $u_{S,L|\omega}$ telle que $x \in \kappa_\omega(\mathbf{x}, L)$.

Proposition 3.5. *Soit S un élément de Ξ. Soient ω_i et ω_j deux ouverts connexes adjacents. Deux lignes de niveau adjacentes de $u_{S,L|\omega_i}$ et $u_{S,L|\omega_j}$ ne fusionnent pour presque aucun L.*

Démonstration. Soit $z \in \overline{\kappa}_{\omega_i}(\mathbf{x}_i, L) \cap \overline{\kappa}_{\omega_j}(\mathbf{x}_j, L)$ et $\mathbf{z} = p^{-1}(z)$.

On a :

$$\forall x \in \kappa_{\omega_i}(\mathbf{x}_i, L),\ u(x_i) = u(x)$$

Et en particulier :

$$\begin{aligned} u(x_i) &= \lim_{\substack{x \to z \\ x \in \omega_i}} u(x) \\ u(x_i) &= \lim_{\substack{x \to z \\ x \in \omega_i}} \left(\left(\langle l, N \rangle + \gamma \right) \alpha \right)(\mathbf{x}) \end{aligned}$$

$z \in \overline{\kappa}_{\omega_i}(\mathbf{x}_i, L) \cap \overline{\kappa}_{\omega_i}(x_i, L)$ donc, $z \in \overline{\omega}_i \cap \overline{\omega}_j$. Comme p est un difféomorphisme, $\mathbf{z} \in \overline{\boldsymbol{\omega}}_i \cap \overline{\boldsymbol{\omega}}_j$ et $\lim_{\substack{x \to z \\ x \in \omega_i}} \mathbf{x} = \mathbf{z}$. En conséquence, d'après les hypothèses précédentes, on a :

$$\begin{aligned} u(x_i) &= \langle l, \lim_{\substack{x \to z \\ x \in \omega_i}} (\alpha N)(\mathbf{x}) \rangle + \gamma \lim_{\substack{x \to z \\ x \in \omega_i}} \alpha(\mathbf{x}) && (3.23) \\ &= \langle l, \lim_{\substack{\mathbf{x} \to \mathbf{z} \\ x \in \boldsymbol{\omega}_i}} (\alpha N)(\mathbf{x}) \rangle + \gamma \lim_{\substack{\mathbf{x} \to \mathbf{z} \\ \mathbf{x} \in \boldsymbol{\omega}_i}} \alpha(\mathbf{x}) && (3.24) \\ & && (3.25) \end{aligned}$$

De même, on a :

$$u(x_j) = \langle l, \lim_{\substack{\mathbf{x} \to \mathbf{z} \\ \mathbf{x} \in \boldsymbol{\omega}_j}} (\alpha N)(\mathbf{x}) \rangle + \gamma \lim_{\substack{\mathbf{x} \to \mathbf{z} \\ \mathbf{x} \in \boldsymbol{\omega}_j}} \alpha(\mathbf{x}) \tag{3.26}$$

D'où $u(x_i) = u(x_j)$ si et seulement si :

$$\underbrace{\langle l, \lim_{\substack{\mathbf{x} \to \mathbf{z} \\ \mathbf{x} \in \boldsymbol{\omega}_i}} (\alpha N)(\mathbf{x}) - \lim_{\substack{\mathbf{x} \to \mathbf{z} \\ \mathbf{x} \in \boldsymbol{\omega}_j}} (\alpha N)(\mathbf{x}) \rangle}_{T_1} + \gamma \left(\lim_{\substack{\mathbf{x} \to \mathbf{z} \\ \mathbf{x} \in \boldsymbol{\omega}_i}} \alpha(\mathbf{x}) - \lim_{\substack{\mathbf{x} \to \mathbf{z} \\ \mathbf{x} \in \boldsymbol{\omega}_j}} \alpha(\mathbf{x}) \right) = 0 \tag{3.27}$$

On considère alors deux cas :

– si $\lim_{\substack{\mathbf{x} \to \mathbf{y} \\ \mathbf{x} \in \boldsymbol{\omega}_i}} (N, \alpha)(\mathbf{x}) \neq \lim_{\substack{\mathbf{x} \to \mathbf{y} \\ \mathbf{x} \in \boldsymbol{\omega}_j}} (N, \alpha)(\mathbf{x})$, alors l'équation (3.27) est vérifiée si et seulement si L réside dans un hyperplan particulier de $\mathbb{R}^4$. Un tel espace est de mesure de Lebesgue nulle.

– si $\lim_{\substack{\mathbf{x} \to \mathbf{y} \\ \mathbf{x} \in \boldsymbol{\omega}_i}} (N, \alpha)(\mathbf{x}) = \lim_{\substack{\mathbf{x} \to \mathbf{y} \\ \mathbf{x} \in \boldsymbol{\omega}_j}} (N, \alpha)(\mathbf{x})$, alors on est ramenés au cas C^2, en conséquence : $u_{S,L}(x_i)$ et $u_{S,L}(x_i)$ forment une unique ligne de niveau qui est invariante pour presque toutes les conditions d'illumination.

□

Ce résultat permet de montrer la quasi invariance des lignes de niveau sur une classe de scène particulièrement intéressante :

Corollaire 3.2. *Soit S une scène plane par morceaux ayant un albédo constant sur chaque morceau. Les lignes de niveau de $u_{S,L}$ sont alors les mêmes pour presque tout L (ce sont les images des morceaux).*

Démonstration. C'est une conséquence de la proposition 3.5. □

3.5 Conclusion

Dans ce chapitre, nous avons analysé l'invariance des lignes de niveau aux changements d'illumination. Premièrement, nous avons redéfini les lignes de niveau pour qu'elles jouissent de propriétés d'invariance plus importantes. Nous avons montré que même avec cette nouvelle définition, elles ne sont pas invariantes pour toutes les scènes. Nous avons alors caractérisé géométriquement les scènes qui produisent des lignes de niveau invariantes. Cet espace de scènes ne modélise de façon satisfaisante quasi aucun élément du monde réel. Nous avons donc montré la quasi invariance des lignes de niveau sur des surfaces développables par morceaux.

Ces surface modélisent assez bien la structure des villes ou plus généralement, des constructions humaines. Dans le chapitre suivant, nous allons nous servir de cette remarque pour construire un algorithme de détection de changements invariant aux conditions d'illuminations.

Chapitre 4

Application à la détection de changements

On s'intéresse dans ce chapitre au problème de détection de changements en mettant à profit les résultats précédents. Le problème de la détection de changements est très difficile. A notre connaissance, aucune méthode ne permet aujourd'hui d'effectuer une détection fiable. D'après des résultats récents, ce problème est mal posé [Chen *et al.*, 2000]. Sans une interprétation sémantique, il est en effet impossible de décider si deux images représentent la même scène sous deux conditions d'illumination différentes ou bien deux scènes différentes [Chen *et al.*, 2000]. Les principales difficultés à surmonter pour définir un algorithme de détection efficace sont les changements d'illumination et les changements de prise de vue. A ces difficultés s'ajoutent la quantité importante de détails haute-résolution qui rendent instables les algorithmes. Ce dernier point a poussé certains auteurs à se concentrer uniquement sur les changements importants du paysage urbain [Lisani et Morel, 2003].

Cependant, les opérateurs humains sont capables d'effectuer une détection de changement manuelle (bien que de nombreuses situations soient ambigües) au prix de longues heures d'effort. Pour effectuer la détection, les humains ont souvent recours à une interprétation de la scène. Cette remarque indique qu'il faut introduire des hypothèses *a priori* sur la géométrie des scènes étudiées. Certains auteurs [Li *et al.*, 2006; Lillestrand, 1972; Fournier *et al.*, 2008] tentent de détecter des objets dont les bords sont des lignes ou des polygones. Ceux-ci ont de grandes probabilités de correspondre à des bâtiments. D'autres auteurs supposent que l'élévation de la scène est constante par morceaux [Watanabe *et al.*, 1998]. Dans ce chapitre, nous supposerons que les scènes appartiennent à Ξ (voir définition 3.20). Nous proposons un algorithme simple qui égalise le constraste de deux images. Après ce pré-traitement, une simple différence pixel à pixel fournit des résultats encourageants, que ce soit sur des images issues de scènes synthétiques OpenGL ou des images réelles (Quickbird).

4.1 Justification des hypothèses

Pour que notre algorithme fonctionne bien, nous faisons de nombreuses hypothèses sur le paysage urbain. Nous les détaillons et justifions ci-dessous :

Hypothèse 1 les surfaces ont une réflectance lambertienne ;

Hypothèse 2 la lumière a deux composantes : une composante ambiante et une composante directionnelle ;

Hypothèse 3 La scène appartient à Ξ ;

Hypothèse 4 Les deux images sont parfaitement recalées ;

Hypothèse 5 On néglige le problème des ombres projetées.

- L'Hypothèse 1 est assez naturelle dans le sens où la plupart des surfaces urbaines sont mates (asphalte, béton). Cette hypothèse n'est pas valable dans le cas de surfaces mouillées ou de vitres par exemple.
- L'Hypothèse 2 est une approximation valable. On peut considérer qu'il n'y a qu'une seule source ponctuelle de lumière à l'infini (le soleil). La lumière ambiante est due à la diffusion de la lumière solaire dans l'atmosphère et à ses reflexions sur les divers objets de la scène.
- L'Hypothèse 3 s'appuie sur la structure géométrique des scènes urbaines. Elle est plus dure à justifier puisqu'elle dépend des normes locales d'architecture de chaque partie du monde. Toutefois, de manière générale en architecture on remarque que les scènes sont développables par morceaux. Les toits en forme de dôme par exemple sont relativement rares. La figure 4.1 est issue de [Lafarge *et al.*, 2008]. Dans cet article, les auteurs cherchent à reconstruire des

paysages urbains en 3 dimensions. Le dictionnaire utilisé contient uniquement des bâtiments qui satisfont cette hypothèse.

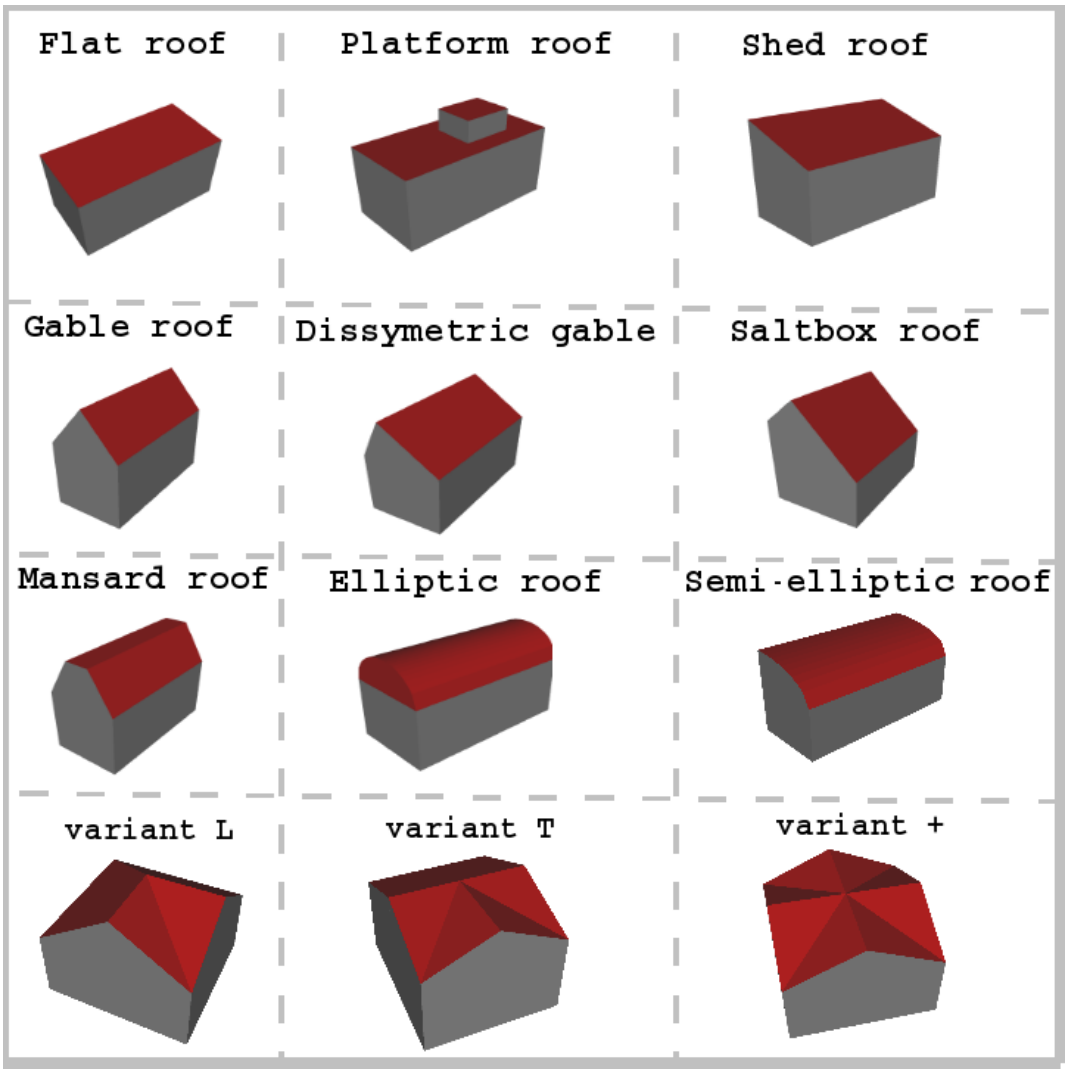

FIG. 4.1 – Dictionnaire de toits utilisés dans [Lafarge *et al.*, 2008] pour modéliser des villes d'architecture européenne. Elles sont toutes constituées de morceaux de surfaces développables. Images fournies gracieusement par F. Lafarge.

– L'Hypothèse 4 est une hypothèse forte. En théorie, il est possible de commander au satellite de prendre une photographie à deux instants différents du même endroit avec le même angle de vue. Les images dont on dispose ne satisfont pas cette hypothèse en général. Pour résoudre ce problème, des techniques de recalage non rigide sont nécessaires. Des perspectives intéressantes voient le jour dans des articles tels que [Droske et Rumpf, 2004; Kybic et Unser, 2003; Liu *et al.*, 2006]. Cependant, même après recalage, des erreurs de parallaxe peuvent persister (voir figure 4.2). On verra plus tard que plusieurs fausses alarmes détectées par notre méthode sont expliquées par les erreurs de parallaxe.
– L'hypothèse sur les ombres est bien entendu déraisonnable. Nous aurons besoin d'un post trai-

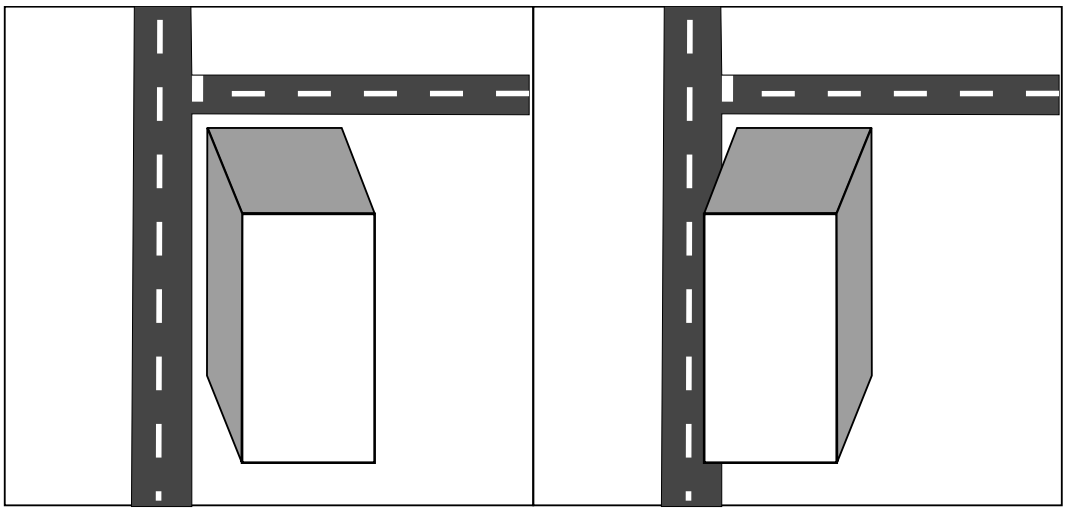

FIG. 4.2 – Illustration des erreurs de parallaxe : deux scènes identiques sont représentées avec des géométries différentes.

tement pour retirer les changements dûs aux ombres. Il existe plusieurs techniques pour trouver les ombres sur des images (voir par exemple [Watanabe *et al.*, 1998]).

4.2 Notre algorithme de détection de changements

Sous les hypothèses évoquées précédemment, on a vu dans le chapitre précédent que les lignes de niveau devraient être “presque” invariantes aux changements d’illumination. Nous proposons une méthode d’égalisation de contraste et une procédure de détection de changements qui tirent parti de ce résultat. Reprenons le formalisme présenté dans l’introduction. Soient u_1 et u_2 deux images parfaitement recalées, prises sous des conditions d’illuminations différentes L_1 et L_2. Soit S_1 la scène 3D ayant donné lieu à u_1. Avec ces notations on peut écrire :

$$\begin{cases} u_1 = u_{S_1,L_1} \\ u_2 = u_{S_1,L_2} + c_{1,2} \end{cases} \tag{4.1}$$

où $c_{1,2}$ est l’image des changements de l’image u_1 à l’image u_2. Dans cette équation, u_{S_1,L_2} et $c_{1,2}$ sont inconnus. Afin de les déterminer, on introduit des *a priori*. On résout alors un problème du type suivant :

$$c_{1,2} = \arg\inf_{c \in X} \left(\phi(c) + \Psi(u_2 - c, u_1)\right) \tag{4.2}$$

où $\phi : X \to \mathbb{R}$ est un a priori sur l’image de changements et $\Psi : X \times X \to \mathbb{R}$ est une fonction qui favorise les couples d’images pouvant être issues d’une même scène. Dans la plupart des applications, les changements n’apparaissent que sur une petite surface de l’image. Dans ce travail préliminaire, nous posons simplement $\phi(c) = |c|_{l^1(X)}$, car la norme l^1 favorise les structures “parcimonieuses”.

Nous discutons maintenant le choix de la fonction Ψ. D’après le théorème 3.25, on sait que l’on doit chercher c parmi les fonctions telles que $\nabla(u_2 - c) \mathbin{/\!/} \nabla u_1$. Ceci mène naturellement à considérer des énergies Ψ du type suivant :

$$\Psi(u,v) = \int_{\Omega} \varphi(|\nabla u \cdot \nabla^{\perp} v|) d\Omega \tag{4.3}$$

où φ est une fonction croissante telle que $\varphi(0) = 0$, u est l'inconnue et v est fixée. Si φ est convexe, alors Ψ est convexe en u et le problème (4.2) peut être résolu efficacement avec les méthodes présentées dans la première partie. Nous avons testé plusieurs fonctions φ, mais les résultats ne sont pas convaincants. Les énergies basées sur le gradient introduisent des interactions trop locales et ne donnent pas de bons résultats. Nous nous sommes donc tournés vers l'utilisation des lignes de niveau. D'après le théorème 3.26 et la proposition 3.5, il est naturel de considérer que u_{S_1,L_2} appartient à l'espace des images qui ont les mêmes lignes de niveau que u_1. On appelle cet espace χ_{u_1}. On pose alors simplement :

$$\Psi(u,u_1) = \begin{cases} 0 & \text{si } u \in \chi_{u_1} \\ \infty & \text{sinon} \end{cases} \tag{4.4}$$

Finalement, pour déterminer $c_{1,2}$, le problème à résoudre est le suivant :

$$\inf_{u \in \chi_{u_1}} \left(|u_2 - u|_{l^1(X)} \right) \tag{4.5}$$

et on pose $c_{1,2} = u_2 - \overline{u}$ où $\overline{u}$ est la solution de l'équation (4.5). Ce problème peut être formulé comme suit :

« trouver l'image u la plus proche de u_2 qui possède les mêmes lignes de niveau que u_1. »

Pour résoudre le problème (4.5), il faut discrétiser χ_{u_1}. On propose la stratégie suivante :

1. On pose $u_Q = \lfloor \frac{u_1}{\Delta} \rfloor \Delta$ (quantification uniforme).
2. Pour chaque niveau $k\Delta$ ($k \in \mathbb{Z}$), on sépare les composantes connexes $\Omega_{k,j}$ de l'ensemble $\Omega_k = \{x \in \mathbb{R}^n, u_Q(x) = k\Delta\}$. Dans les simulations, on utilise la 8-connexité.

On définit χ_{u_1} comme l'ensemble des images constantes sur chaque composante $\Omega_{k,j}$. Avec cette définition, la solution du problème (4.5) peut être calculée analytiquement :

$$\overline{u}|_{\omega_{k,j}} = \text{mediane}(u_2|_{\omega_{k,j}}) \tag{4.6}$$

Ce type d'algorithme a déjà été utilisé et analysé avec des motivations complètement différentes (traitement des images couleurs) dans [Caselles *et al.*, 2002]. C'est un algorithme très rapide (moins de 0.4 secondes sur une image 1000×1000 sur un Processeur Intel Xeon à 1.86 GHz).

Un détail important est que cet algorithme est asymétrique. On peut donc résoudre le problème suivant :

$$\inf_{u \in \chi_{u_2}} \left(|u_1 - u|_{l^1(X)} \right) \tag{4.7}$$

afin de déterminer une seconde image de changements $c_{2,1} = u_1 - \overline{u}$. Cette asymétrie permet notamment de déterminer à quelle scène appartient un objet détecté (cf. figure 4.4)

4.3 Résultats et comparaisons

Dans cette section, nous comparons l'approche proposée avec deux algorithmes classiques : la projection monotone [Moisan, 2005] et la comparaison des arbres d'ensembles de niveau à partir de la FLST (Fast Level Set Transform) [Monasse et Guichard, 2000; Ballester *et al.*, 2000]. Le premier est un outil fondamental en traitement d'images. Le second est fondé sur des principes assez similaires aux nôtres. Nous commençons par les décrire brièvement, puis nous les comparons sur des images de synthèse ainsi que des images réelles.

4.3.1 Description d'approches alternatives

Projection monotone

La projection monotone (décrite dans [Moisan, 2005]) est similaire à une égalisation globale de contraste des deux images. Nous l'avons choisie car elle est plus générale que les égalisations de contraste linéaires couramment utilisées [Radke *et al.*, 2005; Wiemker, 1997]. On suppose à nouveau que les deux images peuvent être écrites comme suit :

$$\begin{cases} u_1 = u_{S_1,L_1} \\ u_2 = u_{S_1,L_2} + c_{1,2} \end{cases} \tag{4.8}$$

La projection monotone s'appuie sur l'hypothèse suivante : deux images de la même scène prises sous des conditions d'illuminations différentes différent uniquement par un changement de contraste global et monotone. Ceci implique que $u_{S_1,L_2} = g \circ u_{S_1,L_1}$ où $g : \mathbb{R} \to \mathbb{R}$ est une fonction croissante. Pour déterminer les changements, on cherche alors la fonction $\overline{g}$ qui minimise l'énergie suivante :

$$\overline{g} = \underset{g\text{croissante}}{\arg\min} \left(|g \circ u_1 - u_2|_X^2 \right) \tag{4.9}$$

Finalement, on pose $c_{1,2} = \overline{g} \circ u_1 - u_2$. Le problème (4.9) peut être résolu en $O(n)$ opérations où n est le nombre de pixels [Moisan, 2005] avec un algorithme simple.

Comparaison des arbres d'ensembles de niveau

Les auteurs de [Ballester *et al.*, 2000; Monasse et Guichard, 2000] proposent un algorithme invariant aux changements de contraste pour la détection de changements. Cet algorithme est proche du nôtre étant donné qu'il fait appel aux lignes de niveau.

Le principe de l'algorithme est le suivant : premièrement, chaque image est décomposée en un arbre de "formes" (les composantes connexes des ensembles de niveau). La figure 4.5 représente la FLST de deux fonctions constantes par morceaux. Les deux arbres sont ensuite comparés. On considère qu'une forme dans l'arbre de u_1 appartient aussi à l'arbre de u_2 s'il existe une forme dans cet arbre qui a approximativement les mêmes moments (barycentre, aire,...). Sur la figure 4.5, les formes sans correspondance sont celles possédant l'étiquette *2*. Les images de changements $c_{1,2}$ et $c_{2,1}$ sont alors recomposées à partir des formes sans correspondance. Nous renvoyons le lecteur à [Ballester *et al.*, 2000; Monasse et Guichard, 2000] pour de plus amples détails.

4.3.2 Résultats sur des images de synthèse

Afin d'illustrer les résultats présentés dans ce chapitre, nous avons développé une application OpenGL, permettant de générer des scènes correspondant à notre modèle. Elle permet de visualiser facilement ces scènes sous des conditions d'illumination différentes. Les images en haut de la figure 4.3 montrent des prises de vues de deux scènes "urbaines". Dans cet exemple, quelque bâtiments ont disparu ou apparu, certains ont été déplacés et enfin, quelques-uns ont été modifiés.

Dans cet exemple jouet, la scène n'appartient pas à Ξ car un des bâtiments est un dôme. Les autres hypothèses de notre modèles sont satisfaites. Les résultats de détection de la figure 4.4 montrent clairement les avantages de notre méthode :

- Le résultat de l'algorithme de projection monotone est peu satisfaisant. Un changement global de contraste ne peut pas reproduire les inversions locales de radiométrie dues aux changements de direction d'illumination. Ceci explique la grande quantité de fausses alarmes sur les toits en forme de dents de scie par exemple.
- Les mauvais résultats de la FLST sont plus subtils. Premièrement, dans nos simulations, nous comparons les formes uniquement à partir de leurs barycentres et leurs surfaces. Ces caractéristiques sont trop simples pour donner des résultats satisfaisants. Dans [Monasse et Guichard, 2000], les auteurs suggèrent l'utilisation de moments d'ordre supérieur. Ceci présente l'inconvénient de devoir règler plus de paramètres et d'augmenter les temps de calculs. De plus, même munis d'une bonne mesure de comparaison, la méthode devrait échouer parce que la FLST n'est pas invariante aux changements d'illumination. Par exemple, la figure 4.5 montre une image d'un toit simple à deux versants. Les FLST de l'image sous deux illuminations différentes sont loin d'être identiques. Quelque soit la mesure de comparaison, cet algorithme détectera la forme labélisée "2" comme changée.
- Le résultat de notre algorithme est beaucoup plus convaincant. On remarque qu'il échoue au niveau du dôme (fausses alarmes dues à la courbure gaussienne non nulle). Ceci était prévisible étant donné que le dôme n'est pas une surface développable. Plusieurs lignes de niveau sont détectées comme changées. Ceci est dû au fait que les lignes de niveau ne sont que quasi-invariantes, comme on l'a montré dans le chapitre précédent. Entre les deux prises de vue, le bâtiment au toit en forme de cylindre en haut au milieu a bougé le long de son axe. Seules les parties qui ne se recouvrent pas on été détectées. Sur cet exemple, il faut clairement interpréter la scène pour détecter tout le bâtiment.

4.3.3 Résultats sur des images de télédétection

On s'intéresse maintenant à des images réelles. Les hypothèses que nous avons émises sur la surface des scènes ne sont valables qu'à des échelles importantes. Les tuiles des toits, par exemple, peuvent rarement être considérées comme développables tandis qu'à grande échelle le toit satisfait cette hypothèse. Pour appliquer l'algorithme que nous avons décrit précédemment, on commence donc par décomposer u_1 et u_2 en une partie texturée et une partie géométrique. On travaille uniquement sur la partie géométrique. Dans nos simulations, nous avons utilisé le modèle de décomposition de Rudin-Osher-Fatemi [Rudin *et al.*, 1992b]. Jusqu'à présent, nous n'avons pas considérée les ombres dans notre modèle. Les régions ombragées sont uniquement éclairées par la lumière ambiante. Leur intensité radiométrique peut être considérée comme de l'ordre d'un dixième de celle des régions éclairées par la lumière du soleil. On peut facilement retirer les changements dûs aux ombres en retirant les changements de faible intensité. Des techniques plus sophistiquées pour détecter les ombres ont été

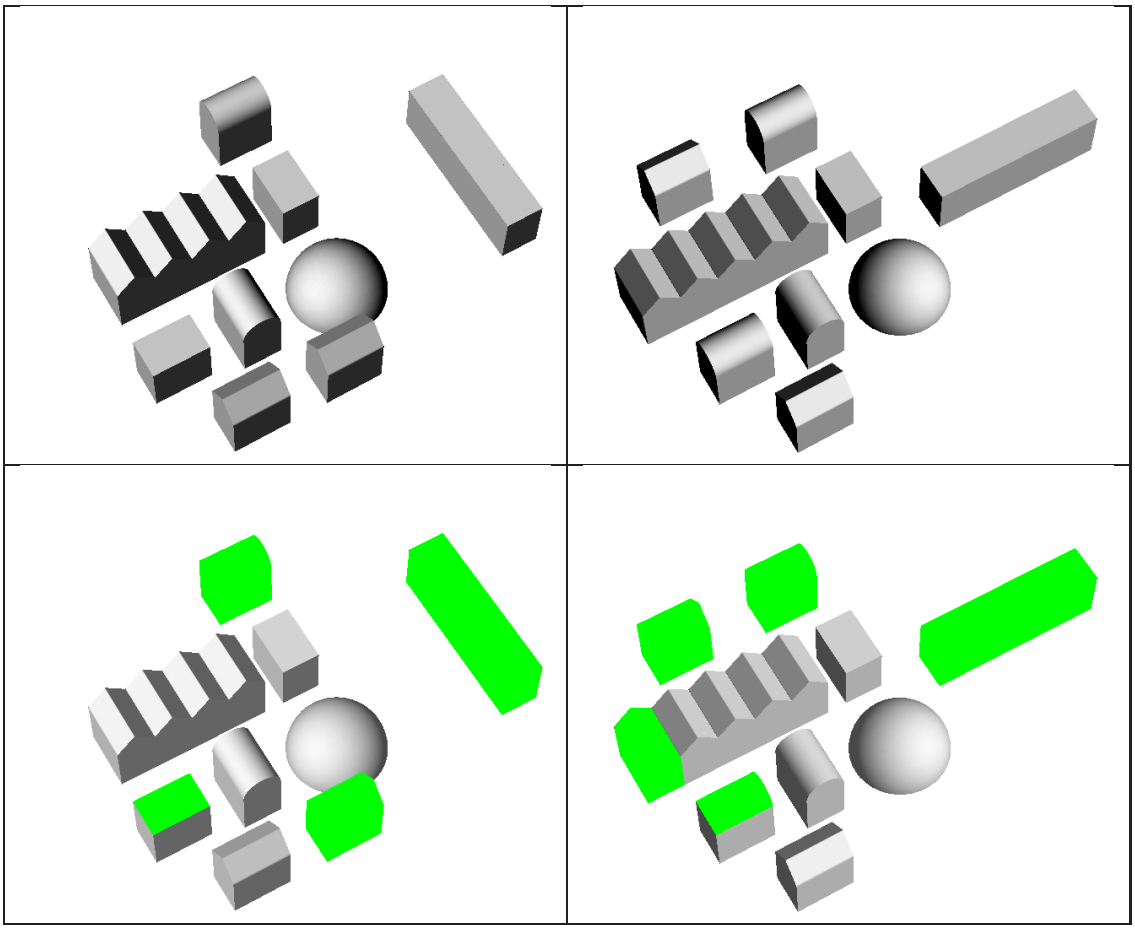

FIG. 4.3 – Exemple jouet. Haut : Deux images sous des conditions différentes d'illumination, avec quelques changements sur les scènes. Bas : vérités de terrain (les changements sont indiqués en vert).

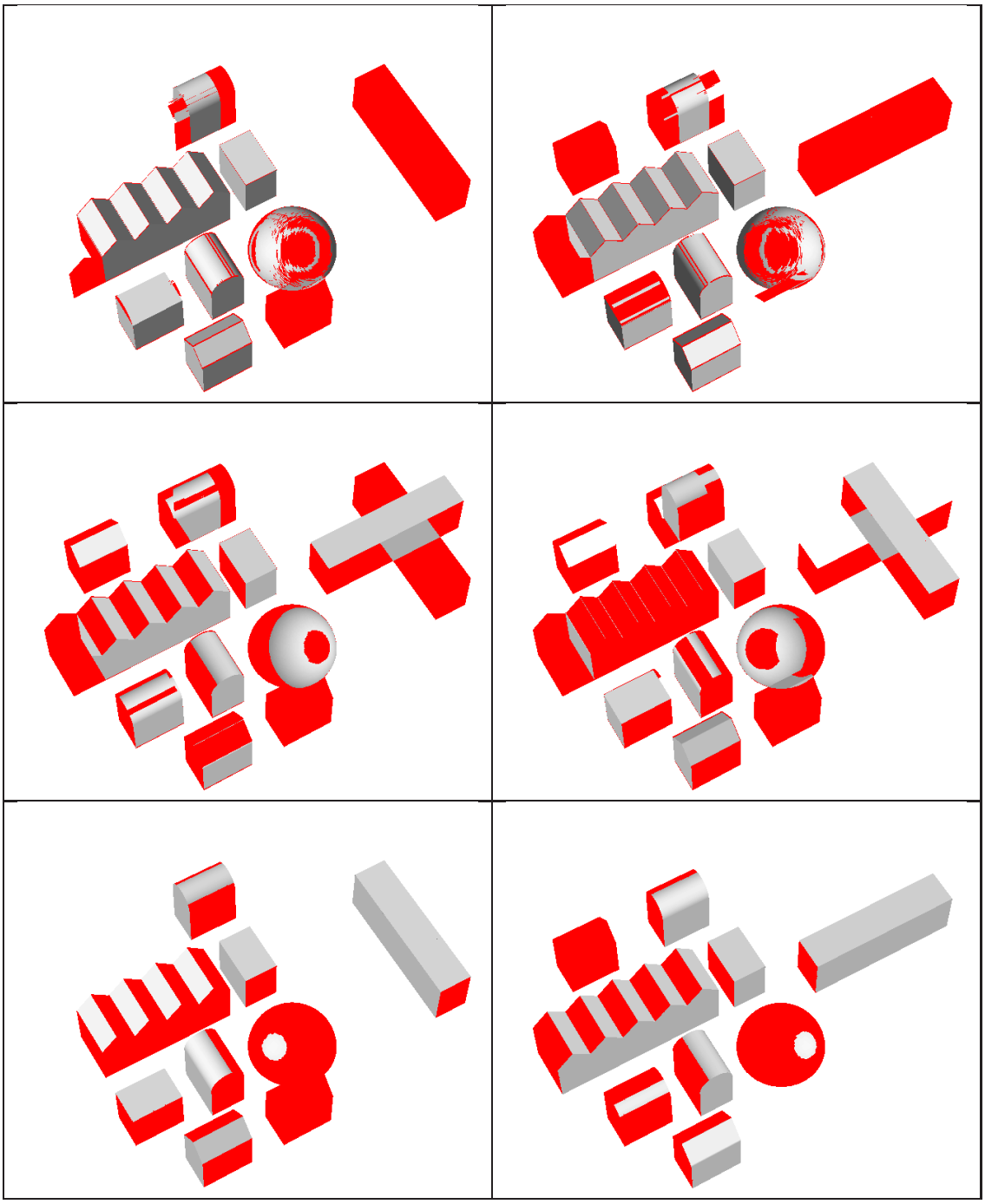

FIG. 4.4 – Exemple jouet. Première ligne, détection de changements en utilisant notre algorithme. Deuxième ligne : détection de changements en utilisant l'algorithme de projection monotone. Troisième ligne : détection de changements en utilisant la FLST.

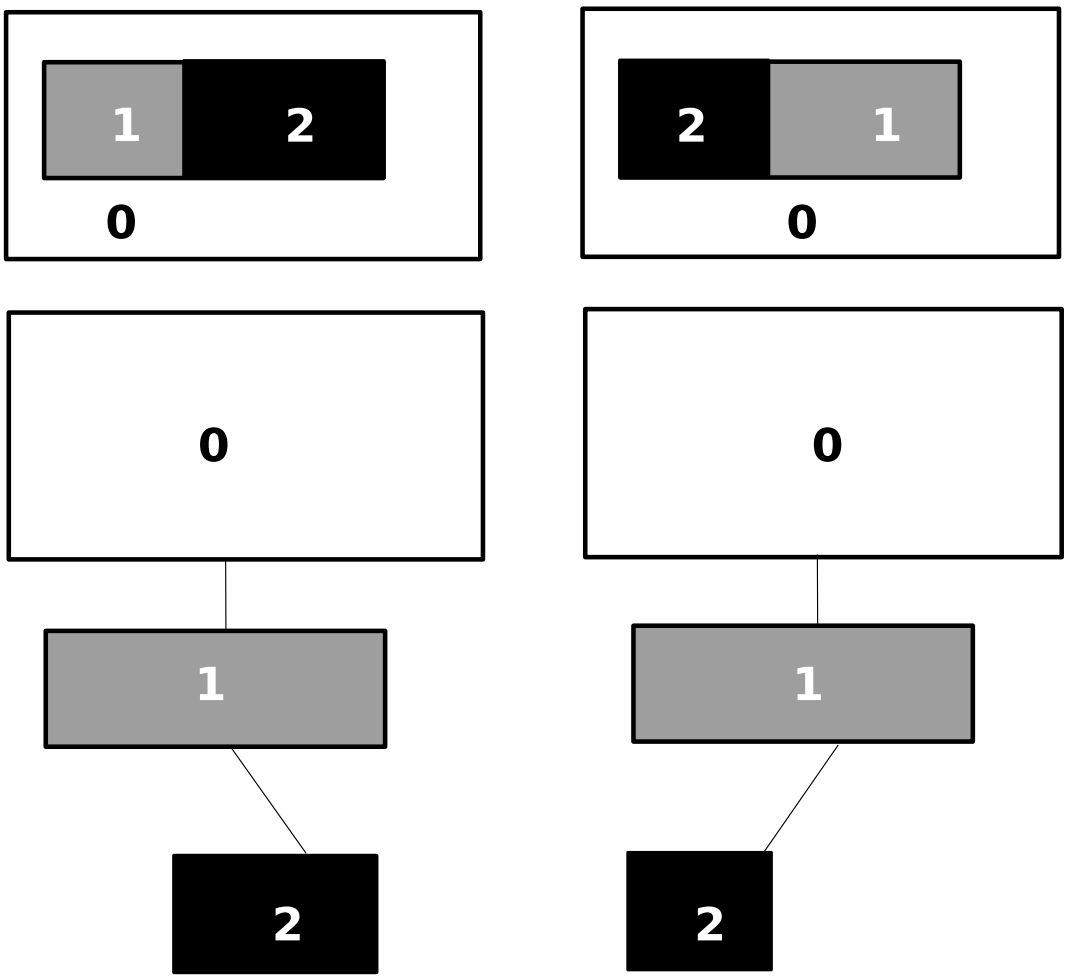

FIG. 4.5 – Haut : images d'un toit simple à deux versants sous deux illuminations différentes. Bas : FLST de chaque image.

proposées [Watanabe *et al.*, 1998].

Nous proposons des comparaisons sur deux paires d'images Quickbird. Les trois méthodes dépendent d'un seuil. On fournit donc des courbes ROC (courbes de caractéristiques d'efficacité) pour comparer les performances des algorithmes quand les seuils varient. Les vérités de terrain ont été faites à la main.

Aéroport d'Abidjan

Sur cette image, la détection de changement est assez aisée et les trois algorithmes donnent de bons résultats. Comparé aux deux approches utilisées pour les comparaisons, notre algorithme donne des résultats satisfaisants, avec peu de faux négatifs et encore moins de faux positifs, et ce en très peu de temps (3 secondes au total pour cette image). Notre implantation de la FLST ne donne pas de résultats satisfaisants comparés aux autres approches. Ceci est probablement dû au fait que nous ne comparons que les aires et barycentres des zones. Les courbes ROC (cf. figure 4.8) montrent que notre algorithme est sensiblement plus performant que les autres approches. Par exemple, pour obtenir 85 % de vrais positifs, on obtient 5 % de fausses alarmes, alors qu'à taux égal de vrais positifs les autres algorithmes génèrent tous deux plus de 20 % de fausses alarmes.

Les principales causes d'échec de notre algorithme sont les erreurs de recalage : quelques lignes peintes sur la piste ne sont pas recalées parfaitement et quelques parois d'immeubles ne coïncident pas exactement. Enfin quelques fausses alarmes sont dues aux changement saisonniers de la végétation autour du parking.

Ville de Pékin

Ce couple d'images est beaucoup plus difficile à traiter que le précédent. En effet, des modifications importantes ont eu lieu sur la scène entre les deux prises de vue. Aussi, nous n'avons pu proposer qu'une vérité de terrain à une échelle grossière. Notre algorithme donne néanmoins des résultats globalement satisfaisants. Avec un seuil correctement choisi, on obtient 75 % de vrais positifs avec 25 % de fausses alarmes. Les autres méthodes génèrent un taux de fausses alarmes bien supérieur. Les mesures fournies sur les courbes ROC (figure 4.10) sont discutables étant donnée la précision de la vérité terrain. Une inspection visuelle mène cependant à la même conclusion. Les seuls résultats exploitables semblent être ceux fournis par notre algorithme.

4.4 Conclusion et perspectives

Dans cette partie, nous avons caractérisé les scènes qui produisent des images dont les lignes de niveau sont invariantes aux changements d'illumination. Ce résultat théorique nous a permis de définir un algorithme simple de détection de changements. Cet algorithme donne des résultats satisfaisants par rapport aux techniques existantes. Ses taux de bonne détection sont à peu près de 75 % sur des scènes difficiles avec seulement 25 % de fausses détections. Nous pensons que cet algorithme simple et rapide pourrait être inclu dans un système de détection plus évolué utilisant plusieurs tests pour prendre une décision. En l'état il peut déjà assister efficacement une détection manuelle des changements. Nous espérons que le formalisme introduit - appuyé par les résultats numériques présentés - encouragera d'autres chercheurs à proposer des algorithmes de détection de changements reposant sur des méthodes variationnelles.

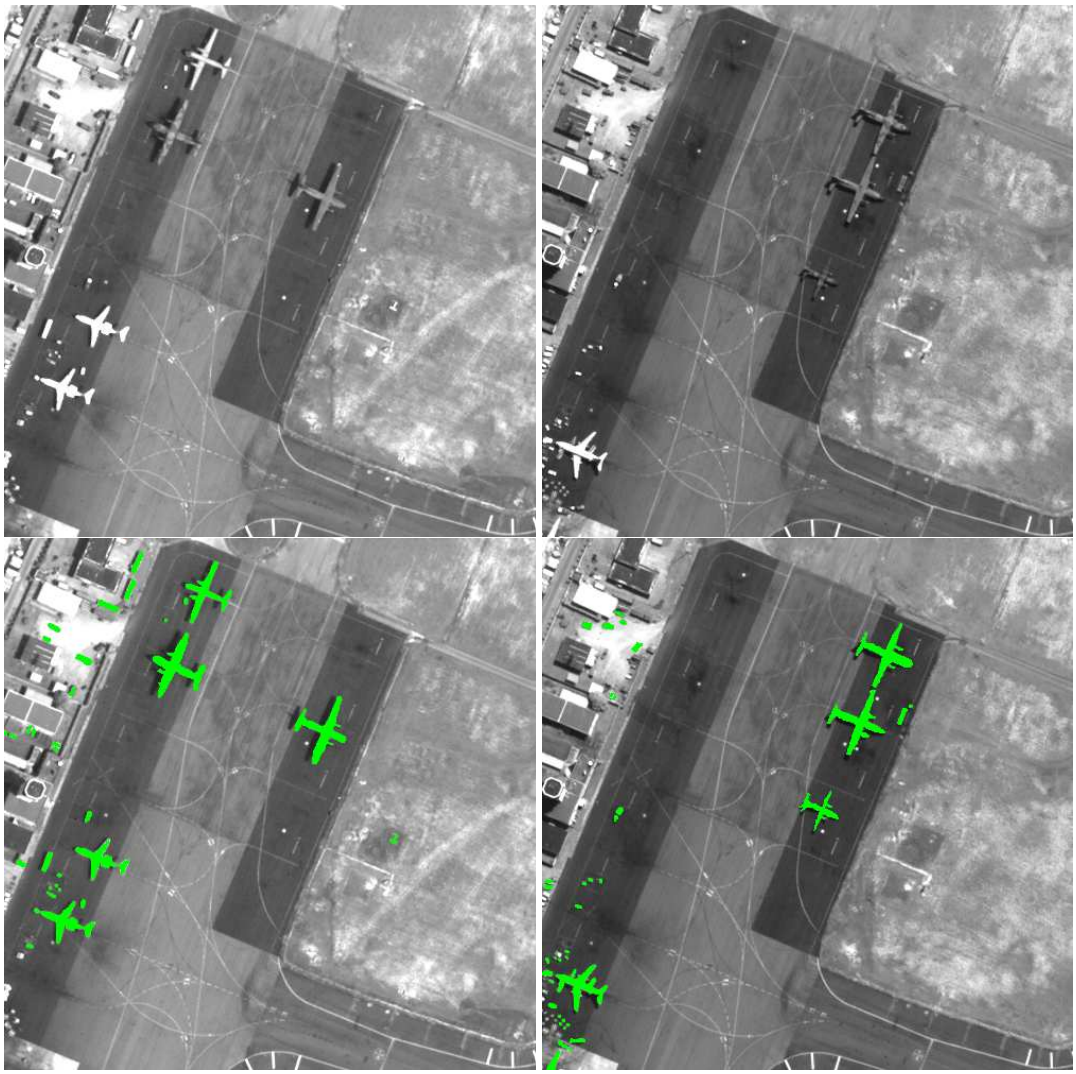

FIG. 4.6 – Images Quickbird (61 cm de résolution) de l'aéroport d'Abidjan. Haut : l'aéroport le 04/02/2003 et le 07/07/2003. Bas : vérités de terrain (les changements sont indiqués en vert).

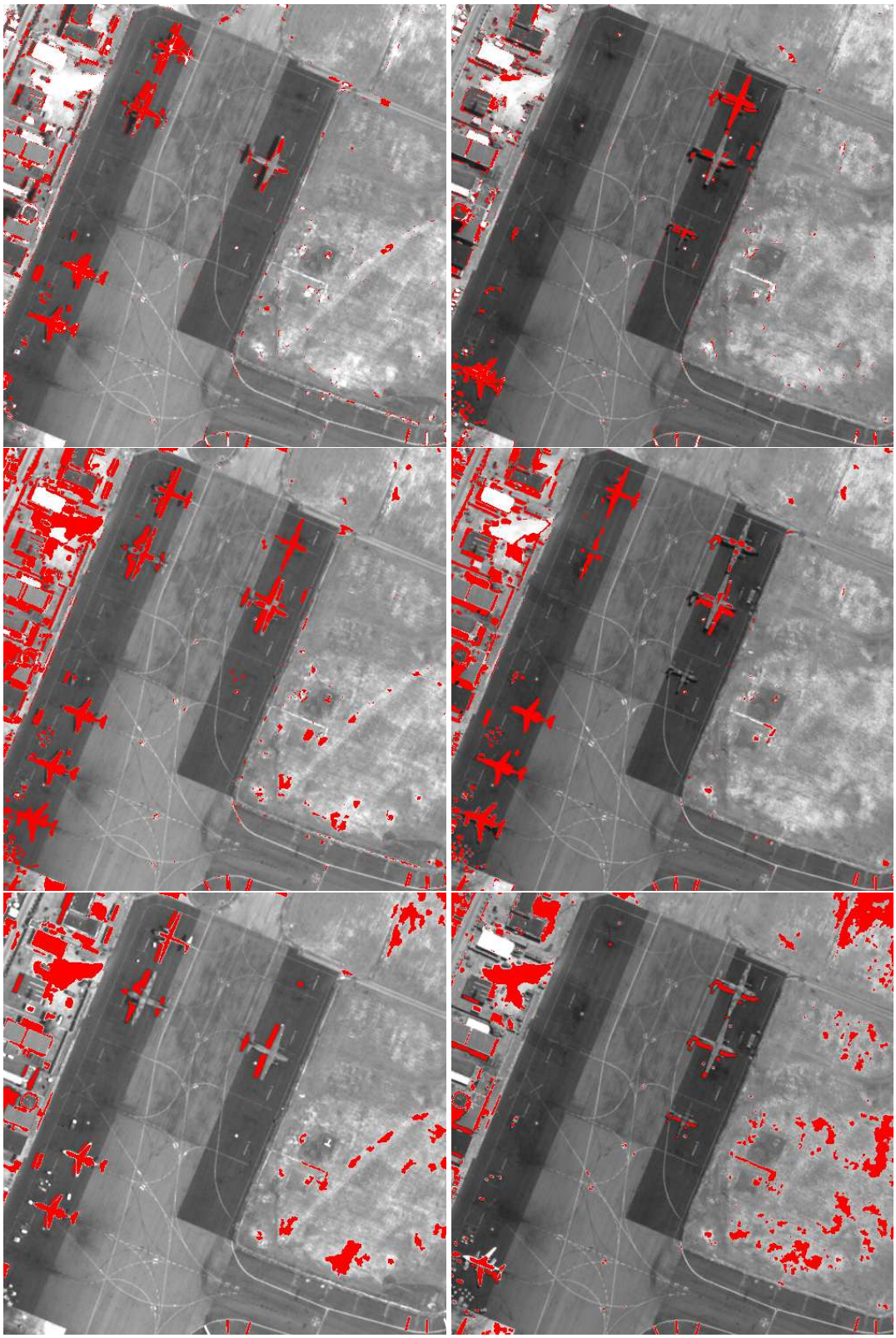

FIG. 4.7 – Images Quickbird (61 cm de résolution) de l'aéroport d'Abidjan. Première ligne : détection de changements en utilisant notre algorithme. Deuxième ligne : détection de changements en utilisant une projection monotone. Troisième ligne : détection de changements en utilisant la FLST.

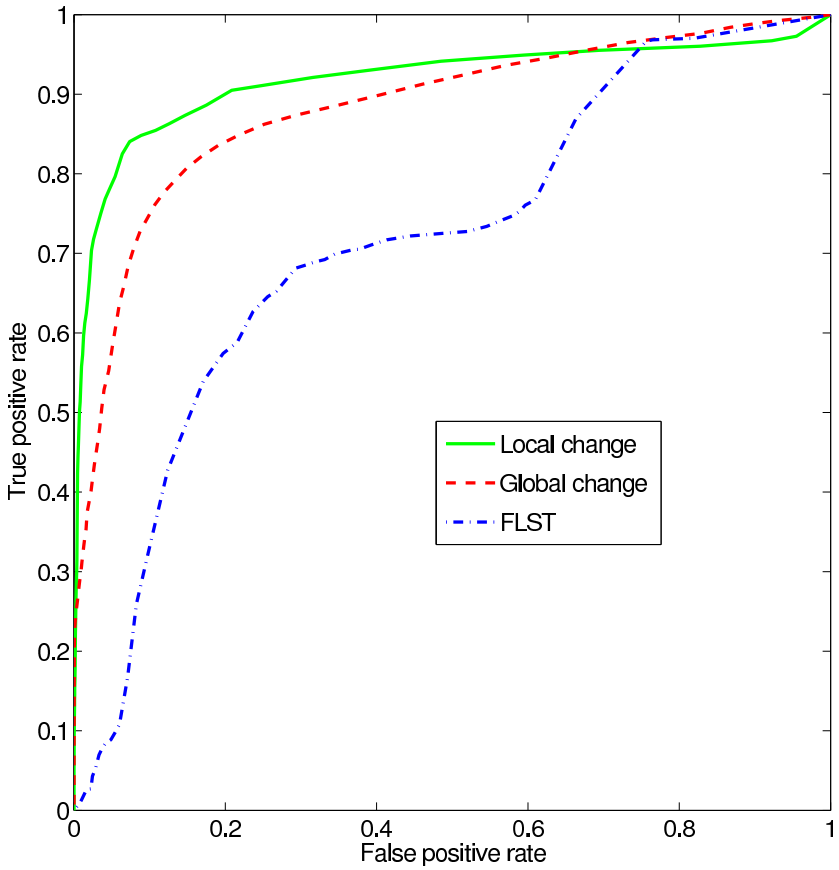

FIG. 4.8 – Courbes ROC pour l'aéroport d'Abidjan.

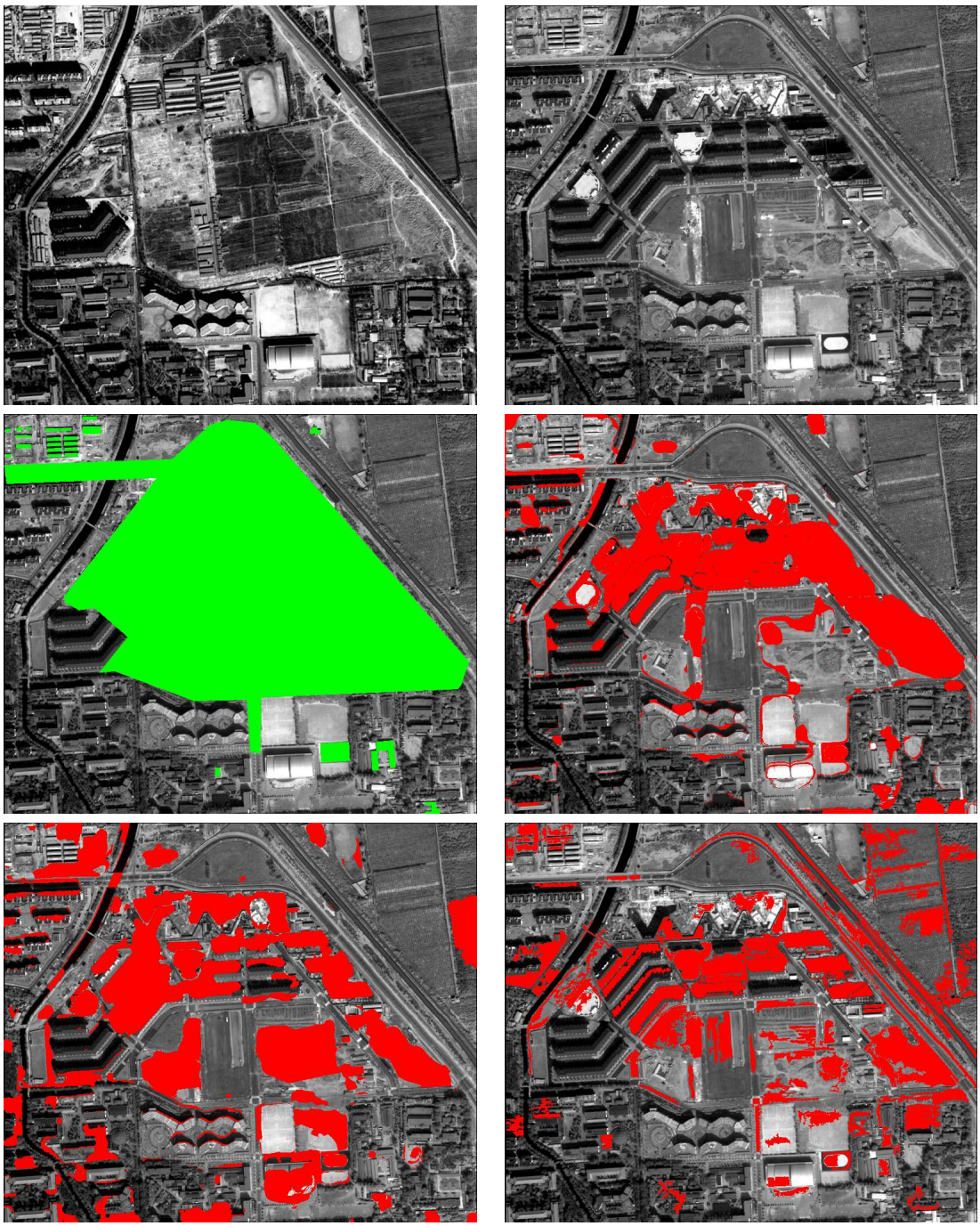

FIG. 4.9 – Images Quickbird (61*cm* de résolution) de Pékin. Première ligne : Pékin en 2001 (gauche) et 2003 (droite). Deuxième ligne - gauche : vérité de terrain (les changements sont indiqués en vert) - droite : Détection de changements en utilisant notre algorithme. Troisième ligne - gauche : détection de changements en utilisant un projection monotone - droite : détection de changements en utilisant la FLST.

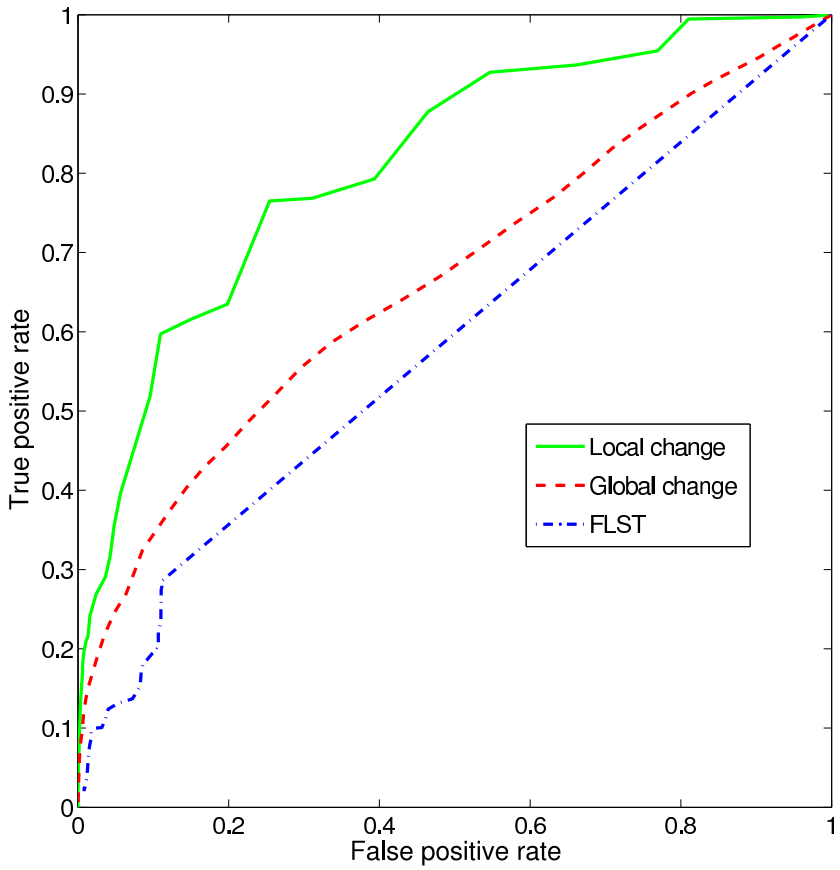

FIG. 4.10 – Courbes ROC pour le couple d'images de la banlieue de Pékin.

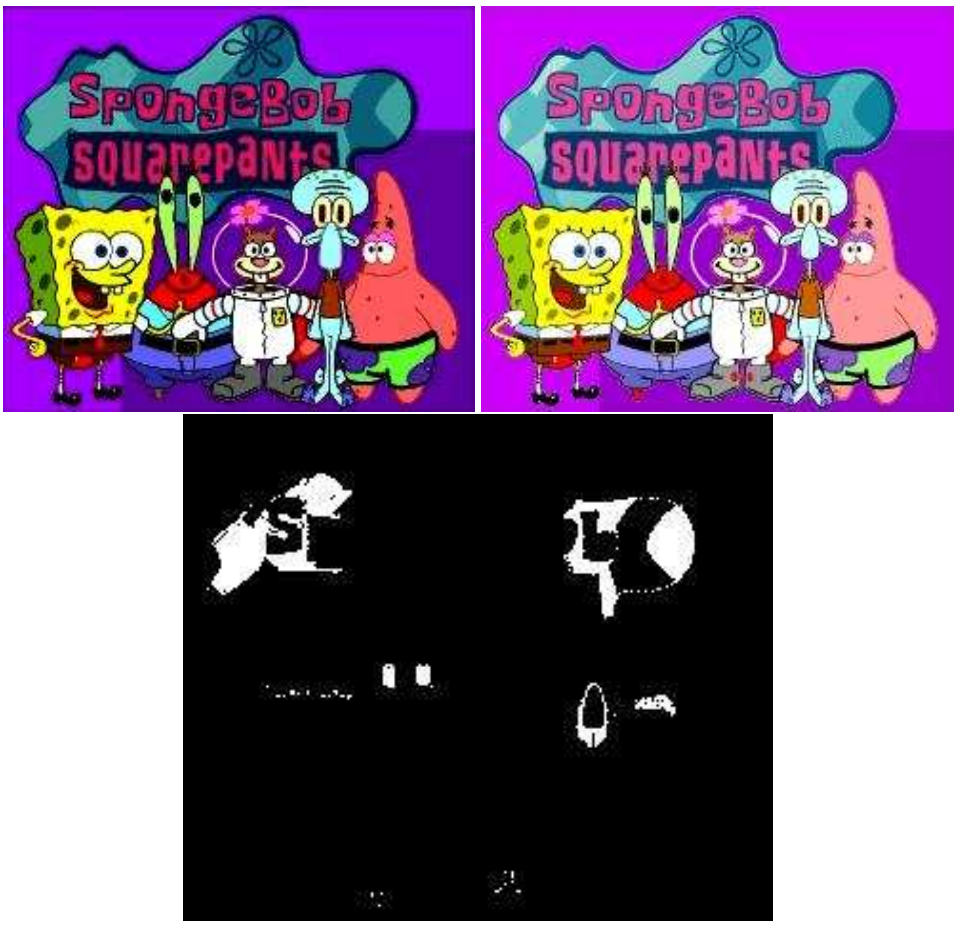

FIG. 4.11 – Et voici la solution du problème proposé en introduction... En haut : trouvez les sept différences. En bas : la solution obtenue par notre algorithme de détection.

Annexe A

Annexes

A.1 Discrétisation du gradient et de la divergence (couleur)

Dans cette partie, pour simplifier les notations, on note $u(i, j, k)$ la valeur de u au pixel (i, j) du canal k. n_x et n_y représentent les nombres de pixels dans les directions horizontales et verticales respectivement.

Pour discrétiser le gradient, on utilise le schéma aux différences finies de premier ordre suivant (il a été introduit dans [Chambolle, 2004]). Pour $u \in X$:

$$(\nabla u)(i, j, k) = ((\partial_1 u)(i, j, k), (\partial_2 u)(i, j, k)) \tag{A.1}$$

∇u est un élément de Y.

$$(\partial_1 u)(i, j, k) = \begin{cases} u(i+1, j, k) - u(i, j, k) & \text{si1 } i < n_x \\ 0 & \text{si } i = n_x \end{cases} \tag{A.2}$$

$$(\partial_2 u)(i, j, k) = \begin{cases} u(i, j+1, k) - u(i, j, k) & \text{si } j < n_y \\ 0 & \text{si } j = n_y \end{cases} \tag{A.3}$$

Cette définition permet de définir la divergence correctement par dualité en imposant :

$$< \nabla u, p >_Y = - < u, div(p) >_X \tag{A.4}$$

Un calcul relativement simple donne :

$$\begin{aligned} (div(p))(i, j, k) \quad = \quad & \begin{cases} p^1(i, j, k) - p^1(i-1, j, k) & \text{si } 1 < i < n_x \\ p^1(i, j, k) & \text{si } i = 1 \\ -p^1(i-1, j, k) & \text{si } i = n_x \end{cases} \\ + \quad & \begin{cases} p^2(i, j, k) - p^2(i, j-1, k) & \text{si } 1 < j < n_y \\ p^2(i, j, k) & \text{si } j = 1 \\ -p^2(i, j-1, k) & \text{si } j = n_y \end{cases} \end{aligned} \tag{A.5}$$

Il est à noter que l'opérateur div est surjectif de Y dans $X - \{(\gamma, \gamma, ..., \gamma), \gamma \in \mathbb{R}\}$.

De plus on peux montrer (en adaptant très peu la preuve de [Chambolle, 2004]) que $\|div\|_2 \leq 2\sqrt{2}$, et ce indépendamment du nombre de canaux. Cette estimée est ajustée.

A.2 Projections sur des boules l^p pondérées ($p \in \{1, 2, \infty\}$)

Dans tout le manuscrit, on suppose que la projection sur des boules l^p pondérées est faisable. Certains opérateurs de projection ne sont pas triviaux à implémenter. Nous proposons donc des solutions à ce problème. Nous notons $K = \{y \in X, |\lambda(y - f)|_p \leq \alpha\}$, où λ est une matrice diagonale dont les éléments λ_i appartiennent à $[0, \infty]$. Le problème de la projection sur K peut s'écrire :

$$\Pi_K(x) = \underset{\{y \in K\}}{\arg\min} \left(|y - x|_2^2 \right) \tag{A.6}$$

Soit $\bar{y}$ la solution de (A.6). Une première remarque importante, valide pour tout p est que si $\lambda_i = 0$, alors $\bar{y_i} = x_i$. Si $\lambda_i = \infty$ alors $\bar{y_i} = f_i$. Dans tous les algorithmes, la première étape consiste donc à fixer ces valeurs connues. Ceci nous permet de restreindre notre attention au cas où $\lambda_i \in]0, \infty[$.

A.2.1 Projection sur une boule l^∞ pondérée

Le projecteur est celui sur des boules l^∞. Il s'écrit sous forme analytique :

$$\bar{y_i} = \begin{cases} x_i & \text{si } |\lambda_i(f_i - x_i)| \leq \alpha \\ f_i + \frac{x_i - f_i}{|x_i - f_i|} \frac{\alpha}{\lambda_i} & \text{sinon} \end{cases} \tag{A.7}$$

A.2.2 Projection sur une boule l^1 pondérée

Après un changement de variables, la projection sur une boule l^1 s'écrit :

$$\Pi_K(x) = \underset{\{u, |\lambda u|_1 \leq \alpha\}}{\arg\min} |u - x|_2^2 \tag{A.8}$$

avec $\lambda_i \in]0, \infty[$ et $\alpha > 0$.

- Si $|\lambda x|_1 \leq \alpha$, alors $\bar{u} = x$.
- Dans les autres cas, l'existence et l'unicité du minimiseur résulte de la forte convexité de $|u - x|_2^2$ et de la convexité de K. Il existe $\sigma \in [0, \infty[$ tel que la solution de (A.8) est donnée par la solution du problème lagrangien suivant :

$$\Pi_K(x) = \underset{u \in \mathbb{R}^n}{\arg\min} |u - x|_2^2 + \sigma |\lambda u|_1 \tag{A.9}$$

La solution de ce problème s'écrit de façon analytique :

$$u(\sigma)_i = \begin{cases} x_i - sgn(x_i) \frac{\sigma \lambda_i}{2} & \text{si } |x_i| \geq \frac{\sigma \lambda_i}{2} \\ 0 & \text{sinon} \end{cases} \tag{A.10}$$

Soit $\Psi(\sigma) = |\lambda u(\sigma)|_1$. Notre problème consiste à trouver $\bar{\sigma}$ tel que $\Psi(\bar{\sigma}) = \alpha$. Ψ est une fonction convexe (donc continue) et décroissante. De plus $\Psi(0) = |\lambda x|_1$, et $\lim_{\sigma \to \infty} \Psi(\sigma) = 0$. D'après le théorème des valeurs intermédiaires, pour tout $\alpha \in [0, |\lambda x|_1]$, il existe $\bar{\sigma}$ tel que $\Psi(\bar{\sigma}) = \alpha$.

$$\Psi(\sigma) = \sum_{i=1}^{n} |\lambda_i \bar{u}_i| \tag{A.11}$$

$$= \sum_{i,|x_i| \geq \sigma\lambda_i/2} \lambda_i(|x_i| - \sigma\lambda_i/2) \tag{A.12}$$

$$= \sum_{i,y_i \geq \sigma} \lambda_i |x_i| - \sigma\lambda_i^2/2 \tag{A.13}$$

avec $y_i = \frac{2|x_i|}{\lambda_i}$. Maintenant il est important de remarquer que Ψ est une fonction décroissante par morceaux linéaire. Les changements de coefficient directeurs ne peuvent intervenir qu'aux valeurs $\sigma = y_j$. Un algorithme pour calculer $\bar{\sigma}$ est donc le suivant :

Algorithme A.1: Projection sur une boule l^1 pondérée

Entrée : x un vecteur à projeter, $\lambda = diag(\lambda_i)$ avec $\lambda_i \in \mathbb{R}_*^+$, $\alpha \in \mathbb{R}^+$

Sortie : $\bar{u}$ la projection de x sur $\{u \in X, |\lambda x|_{l^1 X} \leq \alpha\}$

début

Pour $i \in [1..n]$, calculer $y_i = \frac{2|x_i|}{\lambda_i}$. [O(n) opérations]

Utiliser un *tri rapide*, pour trouver la permutation j telle que $k \rightarrow y_{j(k)}$ soit croissante. [O(n)log(n) opérations]

Calculer les sommes partielles : $\Psi(y_{j(k)}) = E(k) = \sum_{i=k}^{n} \lambda_{j(i)}|x_{j(i)}| - \frac{y_{j(k)}\lambda_{j(i)}^2}{2}$.
E est décroissante. [O(n) opérations]

- Si $E(1) < \alpha$, poser $a_1 = 0$, $b1 = |\lambda x|_1$, $a_2 = y_{j(1)}$, $b2 = E(1)$. [O(1) opérations]
- Sinon, trouver $\bar{k}$ tel que $E(\bar{k}) \geq \alpha$ et $E(\bar{k}+1) < \alpha$.
  Poser $a_1 = y_{j(\bar{k})}$, $b1 = |E(\bar{k})|_1$, $a_2 = y_{j(\bar{k}+1)}$, $b2 = E(\bar{k}+1)$. [O(n) opérations]

Poser $\bar{\sigma} = \frac{(a_2-a_1)\alpha + b_2 a_1 - b_1 a_2}{b_2 - b_1}$. [O(1) opérations]
Poser $\bar{u} = u(\bar{\sigma})$ en utilisant (A.10). [O(n) opérations]

fin

A.2.3 Projection sur une boule l^2 pondérée

La projection sur une boule l^2 pondérée (une ellipsoïde) s'écrit :

$$\Pi_K(x) = \underset{\{y,|\lambda y|_2^2 \leq \alpha\}}{\arg\min} |y - x|_2^2 \tag{A.14}$$

Contrairement aux cas l^∞ et l^1, nous ne proposons pas de solution exacte à ce problème. Nous proposons un algorithme qui mène à des solutions ayant une précision machine en des temps linéaires par rapport au nombre de variables.

- Premièrement, on remarque que $\bar{y} = x$ si $|\lambda x|_2^2 \leq \alpha$.
- Sinon, on peut montrer en utilisant des multiplicateurs de Lagrange qu'il existe un paramètre $\bar{\sigma} > 0$ tel que la solution de (A.14) s'écrive :

$$\bar{y}_i = \frac{x}{\bar{\sigma}|\lambda_i|^2 + 1} \tag{A.15}$$

De plus on sait que $|\lambda\overline{y}|_2^2 = \alpha$. Soit $\Psi(\sigma) = \sum_{i=1}^{n} |\frac{\lambda_i x_i}{\sigma|\lambda_i|^2+1}|_2^2$. On recherche un paramètre $\overline{\sigma}$ tel que $\Psi(\overline{\sigma}) = \alpha$. On peut montrer que Ψ est convexe décroissante. Pour trouver $\overline{\sigma}$ on peut utiliser la méthode de Newton. Elle s'écrit :

1. Poser $k = 0$, $\sigma^k = 0$.
2. Calculer $\alpha^k = \Psi(\sigma^k)$.
3. Calculer $\beta^k = \Psi'(\sigma^k) = -2\sum_{i=1}^{n} \frac{\lambda_i^4 x_i^2}{(\sigma^k \lambda_i^2+1)^3}$.
4. Poser $\sigma^{k+1} = \sigma^k + \frac{(\alpha-\alpha^k)}{\beta^k}$.
5. Poser $k = k + 1$, revenir à 2 jusqu'à ce que $|\alpha^k - \alpha| \leq \epsilon$.
6. Poser $\overline{y} = \frac{x}{\sigma^k|\lambda|^2+1}$.

En théorie ce schéma converge de façon quadratique. Dans toutes nos expériences sur la déconvolution, nous n'avons jamais eu besoin de plus de 15 itérations pour obtenir une solution ayant un précision à 10^{-15}. Les ellipsoides sont dégénérées dans ce cas. Le nombre moyen d'itérations est 6. La projection sur une boule l^2 pondérée est donc très rapide.

A.3 Une méthode rapide détaillée de A à Z : Décomposition d'images

Dans cette partie, on détaille entièrement la démarche qui permet de résoudre le problème de Y. Meyer de façon efficace, en utilisant la technique de régularisation variable et le schéma optimal de minimisation de Y. Nesterov [Nesterov, 2005a; Nesterov, 2007a]. La méthode décrite dans [Nesterov, 2005a] est particulièrement longue à expliquer et à décrire. On se contente donc de montrer ici son principe général ainsi que sa particularisation au problème de Y. Meyer. On souhaite résoudre le problème suivant :

$$\min_{x \in Q_1} (f(x)) \tag{A.16}$$

où Q_1 est un ensemble convexe fermé borné de X et f est convexe sur Q_1. On suppose de plus que f est une fonction qui s'écrit sous la forme :

$$f(x) = \hat{f}(x) + \max_{u \in Q_2} \left(\langle Ax, u \rangle_Y - \hat{\phi}(x) \right) \tag{A.17}$$

où $\hat{f}$ est convexe sur Q_1, Q_2 est un ensemble convexe fermé borné, $\hat{\phi}$ est une fonction convexe sur Q_2 et $A : X \to Y$ est une transformée linéaire. Le problème adjoint à (A.16) s'écrit sous la forme :

$$\max_{u \in Q_2} (\phi(u)) \tag{A.18}$$

$$\phi(u) = -\hat{\phi}(u) + \min_{x \in Q_1} \left(\langle Ax, u \rangle_Y + \hat{f}(x) \right) \tag{A.19}$$

On a de plus $\phi(u) \leq f(x)\ \forall (u, x) \in Y \times X$ (la différence $f(x) - \phi(u)$ est le saut de dualité). L'idée de la méthode proposée dans [Nesterov, 2005a] est la suivante :

– Remplacer f, par une fonction régularisée $f_{\mu_2} \leq f$.
– Remplacer ϕ, par une fonction régularisée $\phi_{\mu_1} \geq \phi$.

– Générer une suite de paramètres (μ_1^k, μ_2^k) et de points (x^k, u^k) tels que :

$$f_{\mu_2}(x^k) \leq \phi_{\mu_1}(u^k) \tag{A.20}$$

$$\lim_{k\to\infty} \mu_1^k = \lim_{k\to\infty} \mu_2^k = 0 \tag{A.21}$$

L'auteur appelle la condition (A.20) *saut de dualité excessif.*

Nous détaillons maintenant le principe de cette méthode sur le problème de décomposition en texture et en géométrie de Y. Meyer. On a vu précédemment que ce problème peut être réécrit sous la forme :

$$\min_{|x|_{l^\infty(Y)}\leq\alpha} \left(|\nabla(I - \operatorname{div} x)|_{l^1(Y)}\right) \tag{A.22}$$

Dans ce problème $I \in X$ est l'image multispectrale à décomposer. $X = \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_c}$ où n_c est le nombre de canaux de l'image. On a aussi $Y = \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_c} \times 2$. $\nabla : X \mapsto Y$ est la discrétisation du gradient. En utilisant les notations introduites précédemment, on a :

$$Q_1 = \{x \in Y, |x|_\infty \leq \alpha\} \tag{A.23}$$

$$f(x) = |\nabla(I - \operatorname{div} x)|_1 \tag{A.24}$$

$$A = -\nabla\operatorname{div} \tag{A.25}$$

$$\hat{\phi}(u) = \langle \nabla I, u\rangle_Y \tag{A.26}$$

$$Q_2 = \{x \in Y, |x|_\infty \leq 1\} \tag{A.27}$$

$$\hat{f}(x) = 0 \tag{A.28}$$

A.3.1 Approximations différentiables

Pour trouver une approximation différentiable des fonctions primales et duales, on utilise la remarque du paragraphe 2.3.4. Il faut choisir deux fonctions fortement convexes. On choisit des fonctions du type $\frac{|x|_Y^2}{2}$ [1]. Notons :

$$f_{\mu_2}(x) = \max_{u\in Q_2} \left(\langle Ax, u\rangle_Y + \langle \nabla I, u\rangle_Y \frac{\mu_2}{2}|u|_Y^2\right) \tag{A.29}$$

D'après le théorème 2.8, f_{μ_2} est une approximation différentiable de la fonction coût primale f. On peut régulariser la fonction duale de la même façon :

$$\phi_{\mu_1}(u) = \langle \nabla I, u\rangle_X + \min_{x\in Q_1} \left(\langle Ax, u\rangle_Y + \frac{\mu_1}{2}|x|_Y^2\right) \tag{A.30}$$

Ces deux problèmes peuvent être résolus de manière analytique :

[1] après avoir listé les fonctions fortement convexes les plus classiques, on se rend compte qu'il n'est pas intéressant numériquement d'en choisir une autre.

Proposition 1.6. *Le maximum (A.29) est atteint pour :*

$$u_{\mu_2}(x) = \Pi_{Q_2}\left(\frac{Ax + \nabla f}{\mu_2}\right) \tag{A.31}$$

De plus, f_{μ_2} est Lipschitz différentiable de constante $L_1 = \frac{\|A\|^2}{\mu_2}$ et on a :

$$\nabla f_{\mu_2}(x) = A^* \Pi_{Q_2}\left(\frac{Ax + \nabla f}{\mu_2}\right) \tag{A.32}$$

Enfin, si on note $D_2 = \max_{u \in Q_2}\left(\frac{1}{2}|u|_Y^2\right) = \frac{n}{2}$, l'approximation f_{μ_2} satisfait $\forall x \in Q_1$, $f(x) - \mu_2 D_2 \leq f_{\mu_2}(x)$.

Proposition 1.7. *Le minimum (A.30) est atteint pour :*

$$x_{\mu_1}(u) = \Pi_{Q_1}\left(-\frac{A^* u}{\mu_1}\right) \tag{A.33}$$

De plus, ϕ_{μ_1} est Lipschitz différentiable, de constante de Lispchitz $L_2 = \frac{\|A\|^2}{\mu_1}$ et on a :

$$\nabla \phi_{\mu_1}(u) = A x_{\mu_1}(u) + \nabla I \tag{A.34}$$

Enfin, si on note $D_1 = \max_{x \in Q_1}\left(\frac{1}{2}|x|_Y^2\right) = \frac{n\alpha^2}{2}$, l'approximation ϕ_{μ_1} satisfait $\forall u \in Q_2$, $\phi_{\mu_1}(u) \leq \phi(u) + \mu_1 D_1$.

A.3.2 Schéma de minimisation

Dans son article [Nesterov, 2005a], l'auteur propose une stratégie pour maintenir le saut de dualité excessif (A.20) tout en réduisant les paramètres de régularisation. Notons [2] :

$$T_{\mu_2}(x) = \arg\min_{y \in Q_1}\left(\langle \nabla f_{\mu_2}(x), y - x\rangle_Y + \frac{L_1}{2}|y - x|_Y^2\right) \tag{A.35}$$

$$= \Pi_{Q_1}\left(x - \frac{\nabla f_{\mu_2}(x)}{L_1}\right) \tag{A.36}$$

Notons aussi :

$$S_{\mu_1}(u) = \arg\max_{v \in Q_2}\left(\langle \nabla \phi_{\mu_1}(u), v - u\rangle_Y - \frac{L_2}{2}|v - u|_Y^2\right) \tag{A.37}$$

$$= \Pi_{Q_2}\left(u + \frac{\nabla \phi_{\mu_1}(u)}{L_2}\right) \tag{A.38}$$

On obtient alors le résultat de convergence suivant :

[2] c'est une descente de gradient sur le problème primal régularisé

Proposition 1.8. *L'algorithme (A.2) assure que :*

$$f(\overline{x}) - \phi(\overline{u}) \leq \frac{4\|A\|}{N}\sqrt{D_1 D_2} \tag{A.39}$$

C'est donc là encore un algorithme en $O\left(\frac{1}{k}\right)$. Son intérêt par rapport aux techniques présentées dans la section 4.3 est que le paramètre de régularisation ne doit pas être choisi en fonction du nombre d'itérations. Ceci simplifie son utilisation.

Algorithme A.2: Technique de saut de dualité excessif appliquée au problème de Y. Meyer

Entrée : Le nombre d'itérations N, un point de départ $x^0 \in Q_1$.
Sortie : $\overline{x}$ une estimée de x^*.
début
  Poser $\mu_1 = \dfrac{2\|A\|}{\alpha}$, $\mu_2 = \|A\|\alpha$
  Calculer $\overline{x} = T_{\mu_2}(x^0)$ et $\overline{u} = u_{\mu_2}(x^0)$.

  pour *k allant de* 0 *à* N **faire**
    $\tau = \frac{2}{k+3}$
    si *k est pair* **alors**
      $\lambda_1 = \frac{2}{k+1}$, $\mu_1 = \lambda_1\|A\|/\alpha$.
      $\lambda_2 = \frac{2}{k+2}$, $\mu_2 = \lambda_2\|A\|\alpha$.
      $L_1 = \|A\|^2/\mu_2$, $L_2 = \|A\|^2/\mu_1$.

      $\hat{x} = (1-\tau)\overline{x} + \tau x_{\mu_1}(\overline{u})$.
      $\overline{u} \leftarrow (1-\tau)\overline{u} + \tau u_{\mu_2}(\hat{x})$.
      $\overline{x} \leftarrow T_{\mu_2}(\hat{x})$.
    fin
    si *k est impair* **alors**
      $\lambda_1 = \frac{2}{k+2}$, $\mu_1 = \lambda_1\|A\|/\alpha$.
      $\lambda_2 = \frac{2}{k+1}$, $\mu_2 = \lambda_2\|A\|\alpha$.
      $L_1 = \|A\|^2/\mu_2$, $L_2 = \|A\|^2/\mu_1$.

      $\hat{u} = (1-\tau)\overline{u} + \tau u_{\mu_2}(\overline{x})$.
      $\overline{x} \leftarrow (1-\tau)\overline{x} + \tau x_{\mu_1}(\hat{u})$.
      $\overline{u} \leftarrow S_{\mu_1}(\hat{u})$.
    fin
  fin
fin

A.4 Et la couleur ?

Nous avons travaillé régulièrement sur le problème de la restauration d'images multi-spectrales au cours de la thèse. Les premiers résultats obtenus ne sont pas suffisamment satisfaisants pour publication. Nous continuerons donc de travailler sur ce sujet pendant l'année de post-doc à venir. Nous

tenons simplement à préciser ici que tous les schémas proposés permettent de résoudre de façon triviale les problèmes à base de variation totale couleur [Chan *et al.*, 2001; Blomgren et Chan, 1998]. La variation totale couleur s'écrit :

$$TV(u) = \sum_{i,j} \sqrt{\sum_{k} |(\nabla u)_{i,j,k}|^2} \tag{A.40}$$

où $(\nabla u)_{i,j,k}$ est la discrétisation du gradient du k-ième canal de u au pixel (i, j). Elle peut se réécrire par dualité sous la forme :

$$TV(u) = \max_{q \in K} (\langle q, \nabla u \rangle_Y) \tag{A.41}$$

où $\langle \cdot, \cdot \rangle_Y$ est le produit scalaire canonique et $\nabla u \in \mathbb{R}^{n_x \times n_y \times n_c \times 2}$ (n_c est le nombre de canaux). Finalement pour obtenir la variation totale couleur, il suffit de choisir :

$$K = \{q \in Y, |q_{i,j}| \leq 1 \ \forall (i, j)\} \tag{A.42}$$

avec $q_{i,j} \in \mathbb{R}^{n_c \times 2}$ et $|\cdot|$ est la norme euclidienne de $\mathbb{R}^{n_c \times 2}$. Tous nos codes fonctionnent avec cette définition de la variation totale couleur. Notons que plusieurs articles ont récemment été publiés pour résoudre les problèmes à base de variation totale couleur [Duval *et al.*, 2008; Bresson et Chan, 2007]. Les algorithmes proposés dans cette thèse sont bien plus performants (ّ). Notons tout de même que les articles [Duval *et al.*, 2008; Bresson et Chan, 2007] contiennent des analyses mathématiques plus fines des propriétés de la variation totale couleur.

Publications et activités de recherche et d'enseignement

Journaux internationaux

1. P. Weiss, A. Fournier, L. Blanc-Féraud et G. Aubert. On the illumination invariance of the level lines under directed light. Application to change detection. Soumis à IEEE Transactions on Pattern Analysis and Machine Intelligence en juillet 2008.
2. P. Weiss, L. Blanc-Féraud et G. Aubert. Efficient schemes for total variation minimization under constraints in image processing. Soumis à SIAM Journal on Scientific Computing en juin 2007. (Minor revision)

Conférences avec actes

1. M. Carlavan et P. Weiss et L. Blanc-Féraud et J. Zerubia. Very efficient first order schemes for solving inverse problems in image restoration. Soumis à IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) avril 2009.
2. A. Fournier et P. Weiss et L. Blanc-Féraud et G. Aubert. A contrast equalization procedure for change detection algorithms : applications to remotely sensed images of urban areas. Dans International Conference on Pattern Recognition (ICPR), décembre 2008.
3. P. Weiss et L. Blanc-Féraud et T. Andre et M. Antonini. Compression artifacts reduction using variational methods : algorithms and experimental study. Dans Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), mars 2008.
4. P. Weiss et L. Blanc-Féraud et G. Aubert. Sur la complexité et la rapidité d'algorithmes pour la minimisation de la variation totale sous contraintes. Dans Proc. GRETSI Symposium on Signal and Image Processing, septembre 2007.

Rapports de recherche

1. P. Weiss et A. Fournier et L. Blanc-Féraud et G. Aubert. On the illumination invariance of the level lines under directed light. Application to change detection. Rapport de Recherche 6612, INRIA, 2008.
2. P. Weiss et L. Blanc-Féraud et G. Aubert. Efficient schemes for total variation minimization under constraints in image processing. Rapport de Recherche 6260, INRIA, juillet 2007.

3. P. Weiss et G. Aubert et L. Blanc-Féraud. Some applications of L infinite norms in image processing. Rapport de Recherche 6115, INRIA, septembre 2006.

Autres conférences sans actes

1. P. Weiss, G. Aubert et L. Blanc-Féraud ont présenté un travail intitulé *Algorithms for fast l^1-norm minimization under different constraints in image processing* à la conférence CANUM, SMAI 2007, 4-8 juin 2007, Praz-sur-Arly, France.
2. P. Weiss, G. Aubert et L. Blanc-Féraud ont présenté un poster intitulé *Fast algorithms for the minimization of constrained total variation* à la Conférence CODE 2007, 18-20 avril 2007, Paris, France.
3. L. Blanc-Féraud, P. Weiss et Gilles Aubert ont présenté un travail intitulé *Some applications of l^∞-norm in image processing* au Workshop *An Interdisciplinary approach to textures and natural image processing*, 8-9 janvier 2007, Paris, France.
4. P. Weiss, L. Blanc-Féraud et Gilles Aubert ont présenté un travail intitulé *Use of the l^∞-norm for some tasks in image processing* à SIAM Conference on Imaging Science. 15-17 mai 2006, Minneapolis, USA.
5. P. Weiss, L. Blanc-Féraud et Gilles Aubert ont présenté un travail intitulé *Some applications of l^∞ constraints in image processing* à la Conférence MIA'2006, 18-21 Septembre 2006, Paris, France.

Enseignements

Dans le cadre de ma thèse, j'ai effectué un monitorat à l'EPU (Ecole Polytechnique Universitaire de Nice Sophia Antipolis, ancienne ESINSA). J'y ai dispensé environ 270 heures de travaux dirigés et de travaux pratiques. Ces 270 heures de cours sont réparties comme suit :

- Première année :
  - 60 heures de TP de logique en 1ère année à l'ESINSA (3 groupes pendant 6 séances).
  - 36 heures de TP de traitement du signal en 3ème année à l'ESINSA (2 groupes pendant 6 séances).
- Deuxième année :
  - 36 heures de TP de traitement du signal en 3ème année à l'ESINSA (2 groupes pendant 6 séances).
  - 40 heures de TD de mathématiques en 1ère année à l'ESINSA (1 groupe pendant 27 séances).
- Troisième année :
  - 54 heures de TP de traitement du signal en 3ème année à l'EPU (3 groupes pendant 6 séances).
  - 30 heures de TD de mathématiques en 1ère année à l'EPU (1 groupe pendant 20 séances).

Activités de relecture

- Conférence SITIS07.
- Conférence SITIS08.

– Journal of Zhejiang University in Applied Maths.
– Journal IEEE Transaction on Signal Processing.
– Journal SIAM journal on Scientific Computing.
– Journal Numerical Algorithms.

Bibliographie

[Agr, 2002]*Les mathématiques pour l'agrégation*. Cours en ligne, 2002.

[Allain *et al.*, 2006]M. Allain, J. Idier et Y. Goussard. On global and local convergence of half-quadratic algorithms. *IEEE Trans. on Image Processing*, 15(5) :1130–1142, 2006.

[Alliney, 1997]S. Alliney. A Property of the Minimum Vectors of a Regularizing Functional Defined by Means of the Absolute Norm. *IEEE Transactions on Signal Proces.*, 45 :913–917, 1997.

[Almansa *et al.*, 2002]A. Almansa, F. Cao, Y. Gousseau et B. Rougé. Interpolation on Digital Elevation Models using AMLE and related Methods. *Transactions in Geoscience and Remote Sensing*, 40(2) :314–325, 2002.

[Almansa *et al.*, 2006]A. Almansa, V. Caselles et G. Haro. Total variation regularized image restoration : the case of perturbed sampling. *SIAM Multiscale Modeling and Simulation*, 5, 2006.

[Alter *et al.*, 2005]F. Alter, S. Durand et J. Froment. Adapted Total Variation for Artifact Free Decompression of Jpeg Images. *Journal of Mathematical Imaging and Vision*, 23 :199–211, 2005.

[Alvarez *et al.*, 1993]L. Alvarez, F. Guichard, P.-L. Lions et J.-M. Morel. Axioms and fundamental equations of image processing. *Arch. Rational Mech. Anal.*, 123(3), 1993.

[Antonini *et al.*, 1992]M. Antonini, M. Barlaud, P. Mathieu et I. Daubechies. Image Coding Using Wavelet Transform. *IEEE Transactions on Image Processing*, 1(2) :205–220, avril 1992.

[Aronsson *et al.*, 2007]G. Aronsson, M.G. Crandall et P. Juutinen. A tour of the theory of absolutely minimizing functions. *Bull. Amer. Math. Soc.*, 41(4) :439–505, 2007.

[Aronsson, 1967]G. Aronsson. Extension of functions satisfying Lipschitz conditions. *Ark. Mat.*, 6(4) :551–561, 1967.

[Aubert *et al.*, 2004]G. Aubert, L. Blanc-Féraud et R. March. Gamma-convergence of discrete functionals with nonconvex perturbation for image classification. *SIAM Journal on Numerical Analysis*, 12(3) :1128–1145, 2004.

[Aubert et Aujol, 2008]G. Aubert et J.-F. Aujol. A variational approach to removing multiplicative noise. *SIAM J. Appl. Math.*, 68(4) :925–946, 2008.

[Aubert et Kornprobst., 2006]G. Aubert et P. Kornprobst. *Mathematical problems in image processing : partial differential equations and the calculus of variations (Second edition)*, volume 147. Springer-Verlag, 2006.

[Aubert et Vese, 1997]G. Aubert et L. Vese. A variational method in image recovery. *SIAM journal on Numerical Analysis*, 34, 1997.

[Aujol *et al.*, 2005]J.-F. Aujol, G. Aubert, L. Blanc-Féraud et A. Chambolle. Image Decomposition into a Bounded Variation Component and an Oscillating Component. *Journal of Mathematical Imaging and Vision*, 22 :71–88, 2005.

[Aujol *et al.*, 2006]J.-F. Aujol, Guy Gilboa, T.F. Chan et S.J. Osher. Structure-Texture Image Decomposition - Modeling, Algorithms, and Parameter Selection. *International Journal of Computer Vision*, 67, 2006.

[Aujol et Chambolle, 2005]J.-F. Aujol et A. Chambolle. Dual norms and image decomposition models. *International Jornal of Computer Vision*, 63, June 2005.

[Aujol, 2004]J.-F. Aujol. Contribution à l'analyse de textures en traitement d'images par méthodes variationnelles et équations aux dérivées partielles. *Thèse de l'université de Nice-Sophia-Antipolis*, 2004.

[Ballester *et al.*, 2000]C. Ballester, E. Cubero-Castan, M. Gonzalez et J. Morel. Contrast invariant image intersection. *Advanced Mathematical Methods in Measurement and Instrumentation*, pages 41–55, 2000.

[Barles et Souganidis., 1991]G. Barles et P.E. Souganidis. Convergence of approximation schemes for fully nonlinear second order equations. *Asymptotic Anal.*, 4(3), 1991.

[Barzilai et Borwein, 1988]J. Barzilai et J. Borwein. Two points step size gradient methods. *Journal of Numerical Analysis*, 8 :141–148, 1988.

[Bect *et al.*, 2004]J. Bect, L. Blanc-Féraud, G. Aubert et A. Chambolle. A L^1-unified variational framework for image restoration. *Proc. European Conference on Computer Vision*, pages 1–13, 2004.

[Bertalmio *et al.*, 2003]M. Bertalmio, V. Caselles, B. Rougé et A. Solé. Total variation image restoration with local constraints. *Journal of scientific computing*, 19, 2003.

[Blomgren et Chan, 1998]P. Blomgren et T.F. Chan. Color TV : Total Variation Methods for Restoration of Vector Valued Images. *IEEE Transactions on Image Processing*, 7(3), 1998.

[Braides, 2002]A. Braides. Gamma-Convergence for beginners. *Oxford lecture series in Mathematics and its applications*, 2002.

[Bresson et Chan, 2007]X. Bresson et T.F. Chan. Fast Minimization of the Vectorial Total Variation Norm and Applications to Color Image Processing. *SIAM Journal on Imaging Sciences*, Submitted 2007.

[Buades *et al.*, 2005a]A. Buades, B. Coll et J. Morel. A Non-Local Algorithm for Image Denoising. *IEEE Conference on Computer Vision and Pattern Recognition*, 2 :60, 2005.

[Buades *et al.*, 2005b]A. Buades, B. Coll et J. M. Morel. A Review of Image Denoising Algorithms, with a New One. *SIAM MMS (Multiscale Model. Simul.)*, 4 :490–530, 2005.

[Bughin *et al.*, 2008]E. Bughin, L. Blanc-Féraud et J. Zerubia. Satellite image reconstruction from an irregular sampling. Dans *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2008.

[Candès et Donoho, 1999]E. Candès et D.L. Donoho. Curvelets - A surprinsingly effective non adaptative representation for object with edges. *Curves and surface fitting*, 1999.

[Candès, 2006]E. Candès. Compressive sampling. *Int. Congress of Mathematics*, 2006.

[Cao et Bouthemy, 2006]F. Cao et P Bouthemy. A general principled method for image similarity validation. Dans *Proc. Int. Workshop on Adaptive Multimedia Retrieval*, volume 4398, pages 57–70, July 2006.

[Carlavan *et al.*, 2009]M. Carlavan, P. Weiss, L. Blanc-Féraud et J. Zerubia. Very efficient first order schemes for solving inverse problems in image restoration. Dans *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, submitted 2009.

[Caselles *et al.*, 1998]V. Caselles, J.-M. Morel et C. Sbert. An axiomatic approach to image interpolation. *IEEE Trans. Image Process.*, 7(3) :376–386, 1998.

[Caselles *et al.*, 1999a]V. Caselles, B. Coll et Jean-Michel Morel. Topographic maps and local contrast changes in natural images. *International Journal of Computer Vision*, 33, 1999.

[Caselles *et al.*, 1999b]V. Caselles, J.L. Lisani, J.M. Morel et G. Sapiro. Shape preserving local histogram modification. *IEEE Transactions on Image Processing*, 8 :220–230, 1999.

[Caselles *et al.*, 2002]V. Caselles, B. Coll et J.M. Morel. Geometry and color in natural images. *JMIV*, 16 :89–107, 2002.

[Chambolle *et al.*, 1998]A. Chambolle, R.A. DeVore, N.-Y. Lee, et B.-J. Lucier. Nonlinear wavelet image processing : variational problems, compression, and noise removal through wavelet shrinkage. *IEEE Trans. Image Process*, 7 :319–335, 1998.

[Chambolle et Lions, 1997]A. Chambolle et P.-L. Lions. Image recovery via total variation minimization and related problems. *Numerische Mathematik*, 76(2) :167–188, 1997.

[Chambolle, 2004]A. Chambolle. An Algorithm for Total Variation Minimization and Applications. *Journal of Mathematical Imaging and Vision*, 20 :89–97, 2004.

[Chambolle, 2005]A. Chambolle. Total variation minimization and a class of binary MRF models. *EMMCVPR*, 578 :136–152, 2005.

[Chan *et al.*, 1999]T.F. Chan, G.H. Golub et P. Mulet. A nonlinear primal-dual method for total variation-based image restoration. *SIAM Journal on Scientific Computing*, 20(6), 1999.

[Chan *et al.*, 2001]T.F. Chan, S.J. Osher et J. Shen. The Digital TV Filter and Nonlinear Denoising. *IEEE Transactions on Image Processing*, 10(2) :231–250, 2001.

[Chan *et al.*, Accepted for publication in 2008]R.H. Chan, S. Setzer et G. Steidl. Inpainting by Flexible Haar-Wavelet Shrinkage. *SIAM Journal on Imaging Sciences*, Accepted for publication in 2008.

[Chan et Esedoglu, 2005]T.F. Chan et S. Esedoglu. Aspects of total variation regularized L^1 function approximation. *SIAM J. Appl. Math.*, pages 1817–1837, 2005.

[Chan et Shen, 2005]T.F. Chan et J. Shen. Image processing and analysis - variational, pde, wavelet, and stochastic methods. *SIAM Publisher*, 2005.

[Charbonnier *et al.*, 1994]P. Charbonnier, L. Blanc-Féraud, G. Aubert et M. Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. *Proceedings of IEEE ICIP (International Conference on Image Processing)*, 2 :168–172, 1994.

[Chaux *et al.*, 2006]C. Chaux, L. Duval et J.C. Pesquet. Image analysis using a dual-tree M-band wavelet transform. *IEEE Transactions on Image Processing*, 15(8) :2397–2412, 2006.

[Chaux *et al.*, 2007]C. Chaux, P. L. Combettes, J.-C. Pesquet et V. R. Wajs. A variational formulation for frame-based inverse problems. *Inverse Problems*, 23 :1495–1518, June 2007.

[Chen *et al.*, 1998]S. Chen, D.L. Donoho et M. Saunders. Atomic decompositions by basis pursuit. *SIAM Journal of scientific computing*, 20 :33–61, 1998.

[Chen *et al.*, 2000]H.F. Chen, P.N. Belhumeur et D.W. Jacobs. In search of illumination invariants. *CVPR*, 2 :254–261, 2000.

[Chen *et al.*, 2008]H.F. Chen, P.N. Belhumeur et D.W. Jacobs. In search of illumination invariants. *submitted to IJCV*, 2008.

[Coifman et Sowa, 2000]R. R. Coifman et A. Sowa. Combining the calculus of variations and wavelets for image enhancement. *Applied and computational harmonic analysis*, 9(1) :1–18, 2000.

[Combettes et Luo, 2002]P. L. Combettes et J. Luo. An adaptive level set method for nondifferentiable constrained image recovery. *IEEE Transactions on Image Processing*, 11 :1295–1304, 2002.

[Combettes et Pesquet, 2004]P. L. Combettes et J. C. Pesquet. Image restoration subject to a total variation constraint. *IEEE Transactions on Image Processing*, 13(9) :1295–1304, 2004.

[Combettes et Wajs, 2005]P. L. Combettes et V. R. Wajs. Signal recovery by proximal forward-backward splitting. *SIAM Journal on Multiscale Modeling and Simulation*, 4(4) :1168–1200, 2005.

[Courteille *et al.*, 2007]F. Courteille, A. Crouzil, J.-D. Durou et P. Gurdjos. Shape from shading for the digitization of curved documents. *Machine Vision and Applications*, 18(5) :301–316, october 2007.

[Crandall *et al.*, 1992]M.G. Crandall, H. Ishii et P.L. Lions. User's guide to viscosity solutions of second order partial differential equations. *Bulletin of the American Mathematical Society*, 1992.

[Darbon et Sigelle, 2006]J. Darbon et M. Sigelle. Image Restoration with Discrete Constrained Total Variation Part I : Fast and Exact Optimization . *Journal of Mathematical Imaging and Vision*, 26(3), 2006.

[Darbon, 2008]J. Darbon. A Simple Efficient Combinatorial Algorithm for Discrete Total Variation Minimization : ROF and the s,t-minimum-cut problem. *UCLA cam report*, July 2008.

[Daubechies *et al.*, 2004]I. Daubechies, M. Defrise et C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Comm. Pure Appl. Math.*, pages 1413–1457, 2004.

[Donoho et Huo, 2002]D.L. Donoho et X. Huo. Beamlets and Multiscale Image Analysis. *Multiscale and Multiresolution Methods : Theory and Applications*, 2002.

[Donoho et Johnstone, 1998]D.L. Donoho et Iain M. Johnstone. Minimax estimation via wavelet shrinkage. *Ann. Statist*, 26 :879–921, 1998.

[Donoho, 1995]D.L. Donoho. De-noising by soft-thresholding. *IEEE Trans. Inform. Theory*, 41 :613–627, 1995.

[Droske et Rumpf, 2004]M. Droske et M. Rumpf. A variational approach to non-rigid morphological registration. *SIAM Appl. Math.*, 64(2) :668–687, 2004.

[Dupé *et al.*, 2008]F.-X. Dupé, M.J. Fadili et J.-L. Starck. Image deconvolution under Poisson noise using sparse representations and proximal thresholding iteration. *International Conference Acoustic Speech and Signal Processing*, 2008.

[Durand et Froment, 2003]S. Durand et J. Froment. Reconstruction of wavelet coefficients using total variation minimization. *SIAM Journal of Scientific Computing*, 24(5) :1754–1767, 2003.

[Durand et Nikolova, 2007]S. Durand et M. Nikolova. Denoising of frame coefficients using L1 data-fidelity term and edge-preserving regularization. *SIAM Journal MMS*, 6 :547–576, 2007.

[Duval *et al.*, 2008]Vincent Duval, J.-F. Aujol et L.A. Vese and. A Projected Gradient Algorithm for Color Image Decomposition. *UCLA CAM report*, July 2008.

[Ekeland et Temam, 1976]I. Ekeland et R. Temam. Convex analysis and variational problems. *Studies in Mathematics and its Applications, American Elsevier Publishing Company*, 1976.

[Figueiredo et Nowak, 2003]M. Figueiredo et R. Nowak. An EM algorithm for wavelet-based image restoration. *IEEE Trans. Image Process.*, pages 906–916, 2003.

[Fournier *et al.*, 2008]A. Fournier, X. Descombes et J. Zerubia. Mixing geometric and radiometric features for change classification. Dans *Proc. SPIE Symposium on Electronic Imaging*, 2008.

[Fu *et al.*, 2005]H. Fu, M. K. NG, M. Nikolova et J.L. Barlow. Efficient Minimization of Mixed $L^2 - L^1$ and $L^1 - L^1$ Norms for Image Restoration. *SIAM J. of Scientific Computing*, 3, 2005.

[Geman et Geman, 1984]S. Geman et D. Geman. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6) :721–741, Novembre 1984.

[Geman et Yang, 1995]D. Geman et C. Yang. Nonlinear image recovery with half-quadratic regularization. *IEEE Transaction on Image Processing*, 7 :932–946, 1995.

[Goldfarb et Yin, 2005]D. Goldfarb et W. Yin. Second-Order Cone Programming Methods for Total Variation Based Image Restoration. *SIAM J. Scientific Computing*, 27 :622–645, 2005.

[Gousseau et Morel, 2001]Y. Gousseau et J.M. Morel. Are Natural Images of Bounded Variation ? *SIAM Journal Mathematical Analysis*, 33(3) :634–648, 2001.

[Guichard et Morel, 2001]F. Guichard et J.M. Morel. *Image Analysis and P.D.E.s*. preprint, 2001.

[Hintermüller et Stadler, 2006]M. Hintermüller et G. Stadler. An infeasible primal-dual algorithm for TV-based inf-convolution-type image restoration. *SIAM Journal of Scientific Computing*, 28, 2006.

[Hiriart-Urruty et Lemarechal, 1996]J.B. Hiriart-Urruty et C. Lemarechal. Convex Analysis and Minimization Algorithms. *Springer-Verlag*, II, 1996.

[Horn et Brooks, 1989]B.K.P. Horn et M.J. Brooks. *Shape from shading*. MIT Press, 1989.

[Huang *et al.*, 2008]Y. Huang, M. Ng et Y. Wen. A Fast Total Variation Minimization Method for Image Restoration. *SIAM Journal on Multiscale Modeling and Simulation*, to appear 2008.

[Idier, 2008]J. Idier. *Bayesian Approach to Inverse Problems*. Wiley, 2008.

[Iouditski, 2007]A. Iouditski. *Efficient methods in optimization*. Cours en ligne de l'INPG, 2007.

[Jalobeanu, 2001]A. Jalobeanu. *Modèles, estimation bayésienne et algorithmes pour la déconvolution d'images satellitaires et aériennes*. Thèse de Doctorat, Universite de Nice Sophia Antipolis, December 2001.

[Janisch, 1984]K. Janisch. *Topology*. Springer, 1984.

[Kelley, 1960]J. E. Kelley. The cutting plane method for solving convex programs. *SIAM*, 8 :452–456, 1960.

[Kingsbury, 1998]NG Kingsbury. The dual-tree complex wavelet transform : a new technique for shift invariance and directional filters. *IEEE Digital Signal Processing Workshop, DSP*, 98, 1998.

[Kiwiel, 1996]K.C. Kiwiel. The efficiency of subgradient projection methods for convex optimization. Part I : General level methods and Part II : Implementations and extensions. *SIAM J. Control Optim.*, 34 :660–697, 1996.

[Krishnan *et al.*, 2006]D. Krishnan, P. Lin et X.C. Tai. An efficient operator splitting method for noise removal in images. *Commun. Comp. Phys.*, 1(5) :847–858, 2006.

[Kybic et Unser, 2003]J. Kybic et M. Unser. Fast parametric elastic image registration. *IEEE Trans. on Image Processing*, 12(11) :1427–1442, November 2003.

[Lafarge *et al.*, 2008]F. Lafarge, X. Descombes, J. Zerubia et M. Pierrot-Deseilligny. Building reconstruction from a single dem. Dans *CVPR*, june 2008.

[Le Pennec et Mallat, 2005]E. Le Pennec et S. Mallat. Bandelet Image Approximation and Compression. *SIAM Multiscale Modeling and Simulation*, 4(3) :992, 2005.

[Lemaréchal et Sagastizabal, 1997]C. Lemaréchal et C. Sagastizabal. Practical aspects of the Moreau-Yosida Regularization I : Theoretical Properties. *SIAM J. on Optimization*, 7 :367–385, 1997.

[Li *et al.*, 2006]W. Li, X. Li, Y. Wu et Z. Hu. A novel framework for urban change detection using vhr satellite images. Dans *ICPR '06 : Proceedings of the 18th International Conference on Pattern Recognition*, pages 312–315, Washington, DC, USA, 2006. IEEE Computer Society.

[Li et Santosa, 1996]Y. Li et F. Santosa. A computational algorithm for minimizing total variation in image restoration. *IEEE Transactions on Image Processing*, 5 :987–995, 1996.

[Lillestrand, 1972]R. Lillestrand. Techniques for change detection. *IEEE Trans. on Computers*, 21(7) :654–659, 1972.

[Lisani et Morel, 2003]J.L. Lisani et J.M. Morel. Detection of major changes in satellite images. Dans *ICIP*, pages 941–944, 2003.

[Liu *et al.*, 2006]D.S. Liu, P. Gong, M. Kelly et Q. Guo. Automatic registration of airborne images with complex local distortion. *Photogrammetric Engineering & Remote Sensing*, 72(9) :1049–1060, September 2006.

[Lu *et al.*, 2004]D. Lu, P. Mausel, E. Brondiacutezio et E. Moran. Change detection techniques. *International Journal of Remote Sensing,*, 25(12) :2365–2401, Juin 2004.

[Malgouyres et Guichard, 2002]F. Malgouyres et F. Guichard. Edge direction preserving image zooming : A mathematical and numerical analysis. *SIAM Journal on Numerical Analysis*, 39(1) :1–37, 2002.

[Mallat et Zhang, 1993]S.G. Mallat et Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12) :3397–3415, 1993.

[Mallat, 1999]S. Mallat. *A Wavelet Tour of Signal Processing, Second Edition (Wavelet Analysis & Its Applications)*. Academic Press, September 1999.

[Marr, 1982]D. Marr. *Vision : A Computational Investigation into the Human Representation and Processing of Visual Information*. W. H. Freeman and Company, NY, 1982.

[Meyer, 2001]Y. Meyer. Oscillating patterns in image processing and in some nonlinear evolution equations. *The Fifteenth Dean Jaqueline B. Lewis Memorial Lectures*, 2001.

[Miller, 2002]A. Miller. Subset selection in regression. *Chapman and Hall*, 2002.

[Moisan, 2005]L. Moisan. *Modeling and Image Processing*. Lecture notes of ens cachan edition, 2005.

[Monasse et Guichard, 2000]P. Monasse et F. Guichard. Fast computation of a contrast-invariant image representation. *IEEE Trans. on Image Processing*, 9 :860–872, 2000.

[Monasse, 1999]P. Monasse. Contrast invariant registration of images. Dans *ICASSP*, volume 6, pages 3221–3224, 1999.

[Moreau, 1962]J.-J. Moreau. Fonctions convexes duales et points proximaux dans un espace hilbertien. *C.R. Acad. Sci. Paris Ser. A Math.*, 255 :2897–2899, 1962.

[Moreau, 1963]J.-J. Moreau. Propriétés des applications 'prox'. *C.R. Acad. Sci. Paris Ser. A Math.*, 256 :1069–1071, 1963.

[Moreau, 1965]J.J. Moreau. Proximité et dualité dans un espace hilbertien. *Bull. Soc. Math. France*, 93 :273–299, 1965.

[Mundy et Zisserman, 1992]J. Mundy et A. Zisserman. *Geometric Invariance in Computer Vision*. MIT Press, Cambridge, 1992.

[Nesterov et Nemirovskii, 1994]Y. Nesterov et A.S. Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. 1994.

[Nesterov et Todd, 1997]Y. Nesterov et MJ Todd. Self-scaled barriers and interior-point methods for convex programming. *Mathematics of Operations Research*, 22(1) :1–42, 1997.

[Nesterov, 1983]Y. Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $O(\epsilon^{-2})$. *Doklady AN SSSR*, 269(3) :543–547, 1983.

[Nesterov, 2004]Y. Nesterov. *Introductory lectures on convex optimization*. Kluwer, 2004.

[Nesterov, 2005a]Y. Nesterov. Excessive gap technique in nonsmooth convex minimization. *SIOPT*, 16 :235–249, 2005.

[Nesterov, 2005b]Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematic Programming, Ser. A*, 103 :127–152, 2005.

[Nesterov, 2007a]Y. Nesterov. Gradient methods for minimizing composite objective function. *CORE discussion paper*, 2007.

[Nesterov, 2007b]Y. Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical programming*, 16 :235–249, June 2007.

[Ng *et al.*, 2000]M.K. Ng, R.H. Chan et W.-C. Tang. A Fast Algorithm for Deblurring Models With Neumann Boundary Conditions. *SIAM J. Sci. Comput.*, 21(3) :851–866, 2000.

[Nikolova et Chan, 2007]M. Nikolova et R. Chan. The equivalence of half-quadratic minimization and the gradient linearization iteration. *IEEE Trans. on Image Processing*, 16 :1623–1627, June 2007.

[Nikolova, 2001]M. Nikolova. Local strong homogeneity of a regularized estimator. *SIAM Journal on Applied Mathematics*, 61(2) :633–658, 2001.

[Nikolova, 2004]M. Nikolova. A variational approach to remove outliers and impulse noise. *Journal of Mathematical Imaging and Vision*, 20 :99–120, 2004.

[Oberman, 2005]A. Oberman. A convergent difference scheme for the infinity Laplacian : construction of absolutely minimizing Lipschitz extensions. *Mathematics of Compuptation*, 74 :1217–1230, 2005.

[Osher *et al.*, 2003]S.J. Osher, A. Sole et L.A. Vese. Image Decomposition and Restoration using Total Variation Minimization and the H^{-1} Norm. *SIAM Multiscale Modeling and Simulation*, 1 :349–370, 2003.

[Osher et Vese, 2003]S.J. Osher et L.A. Vese. Modeling Textures with Total Variation Minimization and Oscillating Patterns. *J. Sci. Comput.*, pages 553–572, 2003.

[Pennec et Mallat, 2005]E. Le Pennec et S. Mallat. Bandelet Image Approximation and Compression. *SIAM Multiscale Modeling*, 2005.

[Phong, 1975]Bui Tuong Phong. Illumination for computer generated pictures. *Comm. ACM*, 18(6) :311–317, June 1975.

[Polyak, 1987]B.T. Polyak. Introduction to Optimization. *Translation Series in Mathematics and Engineering, Optimization Software*, 1987.

[Radke *et al.*, 2005]Richard J. Radke, Srinivas Andra, Omar Al-Kofahi et Badrinath Roysam. Image change detection algorithms : a systematic survey. *IEEE Transactions on Image Processing*, 14(3) :294–307, 2005.

[Rockafellar, 1970]R.T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[Rudin *et al.*, 1992a]L. Rudin, S.J. Osher et E. Fatemi. Nonlinear Total Variation Based Noise Removal. *Physica D*, 60 :259–268, 1992.

[Rudin *et al.*, 1992b]L. Rudin, S.J. Osher et E. Fatemi. Nonlinear Total Variation Based Noise Removal. *Physica D*, 60 :259–268, 1992.

[Rudin et Osher, 1994]L. Rudin et S.J. Osher. Total variation based image restoration with free local constraints. *Proc. IEEE International Conference on Image Processing*, 1 :31–35, 1994.

[Spivak, 1999]M. Spivak. *A comprehensive Introduction to Differential Geometry*, volume 3. Publish or Perish, Inc., 1999.

[Starck *et al.*, 2005]J.-L. Starck, M. Elad et D.L. Donoho. Image Decomposition Via the Combination of Sparse Representation and a Variational Approach. *IEEE Transaction on Image Processing*, 14 :1570–1582, 2005.

[Tan *et al.*, 2006]C.L. Tan, L. Zhang, Z. Zhang et T. Xia. Restoring warped document images through 3d shape modeling. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(2) :195–208, february 2006.

[Tarantola, 2005]A. Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. SIAM, 2005.

[Theiler et Perkins, 2007]J. Theiler et S. Perkins. Resampling approach for anomalous change detection. Dans *Proceedings of SPIE*, volume 6565, page 65651U. SPIE, 2007.

[Toth *et al.*, 2000]D. Toth, T. Aach et V. Metzler. Illumination-invariant change detection. Dans *Image Analysis and Interpretation, 2000. Proceedings. 4th IEEE Southwest Symposium*, pages 3–7, 2000.

[Tramini *et al.*, 1998]S. Tramini, M. Antonini, M. Barlaud et G. Aubert. Quantization Noise Removal for Optimal Transform Decoding. *IEEE International Conference on Image Processing*, pages 381–385, 1998.

[Tropp, 2006]J. Tropp. Just relax : convex programming methods for identifying sparse signals. *IEEE Transactions on Information Theory*, 51 :1030–1051, 2006.

[Vogel et Oman, 1996]C. R. Vogel et M. E. Oman. Iterative methods for total variation denoising. *SIAM Journal on Scientific Computing*, 17(1) :227–238, 1996.

[Wada *et al.*, 1997]T. Wada, H. Ukida et T. Matsuyama. Shape from shading with interreflections under a proximal light source : distortion-free copying of an unfolded book. *IJCV*, 24(2) :125–135, september 1997.

[Watanabe *et al.*, 1998]S. Watanabe, K. Miyajima et N. Mukawa. Detecting changes of buildings from aerial images using shadow and shading model. Dans *Proc. of International Conference on Pattern Recognition*, pages 1408–1412, 1998.

[Weiss *et al.*, 2006]P. Weiss, G. Aubert et L. Blanc-Féraud. Some Applications of l^∞- Constraints in Image Processing. *INRIA Research Report*, (6115), 2006.

[Weiss *et al.*, 2007a]P. Weiss, L. Blanc-Féraud et G. Aubert. Efficient schemes for total variation minimization under constraints in image processing. RR 6260, INRIA, July 2007.

[Weiss *et al.*, 2007b]P. Weiss, L. Blanc-Féraud et G. Aubert. Efficient schemes for total variation minimization under constraints in image processing. *SIAM Journal of Scientific Computing*, Minor revision 2007.

[Weiss *et al.*, 2008a]P. Weiss, L. Blanc-Féraud, T. Andre et M. Antonini. Compression artifacts reduction using variational methods : algorithms and experimental study. Dans *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, mars 2008.

[Weiss *et al.*, 2008b]P. Weiss, L. Blanc-Féraud, T. André et M. Antonini. Some decompression results with variational methods. *http ://www-sop/ariana/RESULTS /Pierre_Weiss_Decompression.pdf*, 2008.

[Wiemker, 1997]R. Wiemker. An iterative spectral-spatial Bayesian labeling approach for unsupervised robust change detection on remotely sensed multispectral imagery. Dans *Proc. Computer Analysis of Images and Patterns (CAIP)*, Kiel, Germany, September 1997.

[Yin *et al.*, 2007]W. Yin, D. Goldfarb et S.J. Osher. A Comparison of Total Variation Based Texture Extraction Models. *Journal of Visual Communication and Image Representation*, 18 :240–252, 2007.

[Zhu, 2008]M. Zhu. *Fast Numerical Algorithms for Total Variation Based Image Restoration*. Thèse de Doctorat, UCLA, July 2008.

RÉSUMÉ

Cette thèse contient des contributions en analyse numérique et en vision par ordinateur.

Dans une première partie, nous nous intéressons à la résolution rapide, par des méthodes de premier ordre, de problèmes d'optimisation convexe. Ces problèmes apparaissent naturellement dans de nombreuses tâches telles que la reconstruction d'images, l'échantillonnage compressif ou la décomposition d'images en texture et en géométrie. Ils ont la particularité d'être non différentiables ou très mal conditionnés. On montre qu'en utilisant des propriétés fines des fonctions à minimiser on peut obtenir des algorithmes de minimisation extrêmement efficaces. On analyse systématiquement leurs taux de convergence en utilisant des résultats récents dûs à Y. Nesterov. Les méthodes proposées correspondent - à notre connaissance - à l'état de l'art des méthodes de premier ordre.

Dans une deuxième partie, nous nous intéressons au problème de la détection de changements entre deux images satellitaires prises au même endroit à des instants différents. Une des difficultés principales à surmonter pour résoudre ce problème est de s'affranchir des conditions d'illuminations différentes entre les deux prises de vue. Ceci nous mène à l'étude de l'invariance aux changements d'illuminations des lignes de niveau d'une image. On caractérise complètement les scènes qui fournissent des lignes de niveau invariantes. Celles-ci correspondent assez bien à des milieux urbains. On propose alors un algorithme simple de détection de changements qui fournit des résultats très satisfaisants sur des images synthétiques et des images Quickbird réelles.

ABSTRACT

This PhD contains contributions in numerical analysis and in computer vision.

In the first part, we focus on the fast resolution, using first order methods, of convex optimization problems. Those problems appear naturally in many image processing tasks like image reconstruction, compressed sensing or texture+cartoon decompositions. They are generally non differentiable or ill-conditioned. We show that they can be solved very efficiently using fine properties of the functions to be minimized. We analyze in a systematic way their convergence rate using recent results due to Y. Nesterov. To our knowledge, the proposed methods correspond to the state of the art of the first order methods.

In the second part, we focus on the problem of change detection between two remotely sensed images taken from the same location at two different times. One of the main difficulty to solve this problem is the differences in the illumination conditions between the two shots. This leads us to study the level line illumination invariance. We completely characterize the 3D scenes which produce invariant level lines. We show that they correspond quite well to urban scenes. Then we propose a variational framework and a simple change detection algorithm which gives satisfying results both on synthetic OpenGL scenes and real Quickbird images.