Précision de modèle et efficacité algorithmique : exemples du traitement de l'occultation en stéréovision binoculaire et de l'accélération de deux algorithmes en optimisation convexe

Pauline Tan

NNT : 2016SACLX092

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PARIS-SACLAY

préparée à

L'ÉCOLE POLYTECHNIQUE

ÉCOLE DOCTORALE N°573

Interfaces : approches interdisciplinaires, fondements, applications et innovation

Spécialité de doctorat : Mathématiques appliquées

par

Mme Pauline Tan

Précision de modèle et efficacité algorithmique : exemples du traitement de l'occultation en stéréovision binoculaire et de l'accélération de deux algorithmes en optimisation convexe

Thèse présentée et soutenue à Palaiseau, le 28 novembre 2016

Composition du jury :

M. Jean-François Aujol	Professeur	Rapporteur
M. Antonin Chambolle	Directeur de recherche	Directeur de thèse
M. Laurent Condat	Chargé de recherche	Examinateur
M. Jalal Fadili	Professeur	Examinateur
M. Pascal Monasse	Chargé de recherche	Co-directeur de thèse
M. Jean-Michel Morel	Professeur	Président du jury
M. Jean-Christophe Pesquet	Professeur	Rapporteur

Remerciements

Ce manuscrit est l'occasion pour moi de remercier les personnes qui ont contribué à sa rédaction.

Je tiens à remercier en tout premier lieu Antonin Chambolle et Pascal Monasse de m'avoir encadrée pendant ces trois années. J'ai particulièrement apprécié leur complémentarité, qui a assuré l'équilibre de cette thèse. La grande liberté et l'autonomie dont j'ai pu bénéficier m'ont énormément plu.

Il me faut ensuite remercier Jean-Michel Morel, non seulement pour avoir accepté de faire partie de mon jury de thèse, mais également pour m'avoir guidée, lors de ma première année à l'ENS Cachan, vers la recherche en traitement d'images. Il n'a cessé de m'encourager depuis, et je lui suis reconnaissante de son soutien et de ses excellents conseils depuis toutes ces années.

Je souhaite également remercier chaleureusement les membres du jury. Tout d'abord, mes rapporteurs, Jean-François Aujol et Jean-Christophe Pesquet, pour leurs commentaires constructifs sur mon manuscrit, en particulier Jean-François Aujol d'avoir été aussi enthousiaste quant à mon travail. Ensuite, Laurent Condat, Jalal Fadili et, à nouveau, Jean-Michel Morel pour avoir accepté de participer en tant qu'examinateurs.

Je remercie Abdul Barakat et Johanne Mensah du Ladhyx de m'avoir donné l'occasion de travailler avec eux sur un projet aussi intéressant qu'a été la détection automatisée de stents. Cette collaboration a été, du point de vue scientifique autant qu'humain, une expérience des plus enrichissantes. Les réunions avec Antoine Lafont, cardiologue à l'Hôpital Européen Georges Pompidou, et avec Machiel van der Leest, directeur général d'Arterial Remodeling Technologies, m'ont fait découvrir un autre aspect de la recherche appliquée.

Enfin, j'ai une pensée pour les nombreuses personnes qui ont contribué à leur façon au travail de ces trois dernières années. Tout d'abord, je souhaite exprimer toute ma sincère reconnaissance à Nasséra Naar et Alexandra Noiret d'avoir rendu mes démarches administratives au CMAP aussi simples que possible, et d'avoir toujours patiemment répondu à mes nombreuses sollicitations. Je souhaite également remercier l'équipe d'Imagine pour leur accueil durant ma dernière année de thèse. Je remercie ensuite Rafael Grompone pour les conversations toujours riches d'enseignement que nous avons eues au CMLA, ainsi que pour son soutien inconditionnel. Merci à Lara Raad d'avoir organisé mes quelques exposés au GTTI. Merci à Aymeric Maury et Massil Achab pour avoir été des voisins d'open-space très civilisés. Je remercie également Neus Sabater pour m'avoir encadrée en stage de L3, puis Sébastien Leprince pour m'avoir accueillie à Caltech lors de mon stage de M1 ; sans conteste, ces deux premières expériences ont affirmé mon goût pour la recherche.

Évidemment, ces remerciements ne seraient pas complets sans une petite pensée pour ma famille et mes proches.

Table des matières

Introduction

Contexte de la thèse

Avec l'intérêt croissant pour les technologies basées sur l'analyse et le traitement des images, et le perfectionnement continu des algorithmes, deux mouvements contraires co-existent. D'une part, la nécessité de calculer toujours plus vite (avec parfois l'idée d'effectuer les opérations en *temps réel*). De l'autre, l'objectif légitime d'améliorer la précision des résultats obtenus. L'avénement des nouvelles technologies et des nouveaux processeurs entretiennent l'espoir qu'un jour, ces deux contraintes pourront être conjointement satisfaites. Néanmoins, il est encore aujourd'hui nécessaire de faire un choix entre ces deux critères.

Ce choix peut reposer sur des critères très variés, mais il est généralement raisonnable de choisir de consacrer du temps de calcul sur des tâches complexes qui constituent la finalité de la méthode, tandis que des tâches annexes ou préliminaires peuvent se contenter d'être traitées de manière rapide, quitte à en sacrifier la qualité.

Le premier mouvement, qui cherche à augmenter la précision des résultats, consiste à améliorer le modèle utilisé, ce qui sous-entend souvent de le complexifier. Le second, qui vise au contraire la rapidité, nécessite généralement à résoudre un problème approché, soit simplifié pour le rend compatible avec des outils de résolution efficaces, soit modifié de sorte de le rendre parallélisable.

Contribution de la thèse

Le travail présenté ici cherche à explorer les deux aspects présentés plus haut à travers différents problèmes classiquement rencontrés en traitement d'images, à savoir la mise en correspondance stéréoscopique et le débruitage.

Gérer les occultations par méthode variationnelle Dans les chapitres 3 et 4, nous nous intéressons à un problème central du traitement d'images qui est la stéréovision binoculaire. Ce domaine, très actif depuis les dernières décennies, cherche à reconstruire une carte du relief à partir d'une paire d'images. C'est un problème difficile, qui présente par ailleurs une complexité algorithmique élevée. Une des difficultés rencontrées est la présence d'*occultation*. Ce phénomène inévitable est, de par sa complexité, largement ignoré par la plupart des méthodes proposées. Dans le chapitre 3, nous avons cherché à le gérer dans un cadre variationnel, en améliorant les fonctionnelles d'énergie classiquement rencontrées, grâce à l'ajout d'un terme inédit dans sa version 2D. Le modèle considéré repose sur une analyse fine du phénomène d'occultation. L'introduction de ce terme rend le problème sous-jacent plus délicat à résoudre, la résolution a été rendue possible grâce à l'adaptation d'un algorithme primal-dual existant.

Spécification de l'intervalle de disparité par pixel dans la méthode des *graph cuts* Dans le chapitre 4, nous restons dans le cadre de la stéréovision binoculaire. Nous sommes à nouveau partis d'un algorithme existant et dont l'efficacité algorithmique est réputée car elle repose sur l'utilisation de graphes. Nous avons exploité cette efficacité pour pouvoir l'appliquer à des problèmes périphériques qui sont le raffinement de cartes de disparité et leur densification. Le premier problème peut se présenter dans le cas où une méthode produit des cartes de précision pixellique de manière fiable mais coûteuse, ne permettant pas d'atteindre une précision sous-pixellique en temps de calcul raisonnables. Le second problème peut se présenter de manière avantageuse lorsqu'une méthode fiable (et coûteuse) n'arrive pas à produire une carte dense. Dans les deux cas, l'idée est de profiter d'une méthode très efficace permettant au post-traitement (raffinement ou densification) d'être très rapide, de sorte qu'on puisse consacrer davantage de temps pour la première estimation, qui pourra elle être basée sur un modèle plus complexe (non étudié dans cette thèse).

Convergence d'algorithmes primaux-duaux : application à l'ADMM Dans le cadre que nous étudions, une autre approche est également possible. Il s'agit d'accélérer des algorithmes existants. Dans le chapitre 5, on s'intéresse au cas de la méthode des directions alternées, qui est fréquemment utilisée dans les algorithmes de traitement d'images. On montre qu'un léger relâchement de contraintes dans les paramètres de cette méthode permet d'obtenir un taux de convergence théorique plus intéressant. Cette amélioration a été en particulier testée sur un problème classique qui est le lissage des images. Dans ce chapitre, nous nous sommes donc attachés à accélérer un algorithme résolvant de manière *exacte* le problème considéré.

Alterner les descentes proximales : application au modèle ROF Enfin, dans le chapitre 6, on s'intéresse à un dernier aspect de cette dualité précision/rapidité, en considérant le problème central du débruitage par le modèle Rudin-Osher-Fatemi. Ce modèle est connu pour donner des résultats satisfaisants, surtout lorsqu'il s'agit d'effectuer un pré-traitement des données. Afin de libérer du temps de calcul pour d'autres tâches plus importantes, il est nécessaire de réduire cette étape au maximum. dans ce chapitre, nous nous proposons de gagner du temps de calcul global grâce à l'utilisation du calcul parallèle. Cette approche conduit dans le cas du modèle ROF à résoudre un problème approché, pour lequel nous avons proposé une variante accélérée d'un algorithme de minimisations alternées. Cette méthode a été appliquée dans le cas du débruitage d'images en couleur.

Le présent manuscrit est donc composé de deux parties relativement indépendantes. Chacune d'entre elles aborde la dualité précision/rapidité développée plus haut d'une manière différente. Ainsi, la première partie (chapitres 3 et 4) porte essentiellement sur le problème de la stéréovision binoculaire. L'objectif est d'améliorer deux approches existantes : dans un cas (chapitre 3), nous améliorons un modèle en y incluant la gestion de l'occultation, d'une manière qui rende le nouveau problème résoluble par *relaxation convexe* ; dans le second (chapitre 4), nous exploitons l'efficacité de la méthode des *graph cuts* pour l'adapter à des problèmes différents. La seconde partie (chapitres 5 et 6) se concentre au contraire sur l'amélioration des algorithmes d'optimisation convexe, en modifiant dans un cas (chapitre 5) des paramètres et dans l'autre cas (chapitre 6) en remplaçant des pas de minimisation par des pas de descente proximale. Dans ces deux cas, les algorithmes modifiés sont d'abord étudiés de manière formelle, afin d'en exhiber

les performances théoriques. Une application sur deux problèmes de débruitage permet alors d'en vérifier les performances pratiques. Enfin, une partie introductive (chapitres 1 et 2) permet au lecteur de se remémorer les notions importantes utilisées tout au long de ce manuscrit, qu'il s'agisse d'optimisation convexe ou de stéréovision binoculaire.

Première partie

Introduction

Chapitre 1

Optimisation convexe et méthodes de descente

Introduction

De nombreux problèmes rencontrés en traitement d'images peuvent être abordés en introduisant une fonctionnelle d'énergie qui traduit le modèle considéré, en mesurant l'écart de toute fonction donnée à ce modèle. Si le modèle est suffisamment réaliste, la solution recherchée est logiquement celle qui en est le plus proche, c'est-à-dire celle qui minimise la fonctionnelle associée. Les fonctionnelles sont généralement composées de plusieurs termes séparés, chacun correspondant à une composante du modèle considéré. Ces termes peuvent être différentiables ou non, mais sont généralement convexes [1]. On se concentre dans ce chapitre sur le cas des fonctionnelles convexes, car il offre un cadre de travail naturel pour la minimisation. La convexité assure en effet l'existence d'une solution (donnée par le minimum global), mais surtout la non-existence de minima locaux. Elle permet donc d'envisager des stratégies basiques de *descente* pour rechercher

1. Ce n'est pas toujours le cas, cf. chapitre 2

un minimum : elles consistent à trouver la direction de descente de l'énergie, qui est donnée par le gradient lorsque la fonctionnelle est différentiable. On verra qu'il est possible d'étendre ce genre de méthodes dans le cas non différentiable.

La régularité (au sens large) des fonctionnelles joue un rôle prépondérant dans le choix et la conception des algorithmes de minimisation. La différentiabilité permet par exemple des calculs explicites dans les schémas de descente de gradient. Néanmoins, la différentiabilité n'est pas toujours acquise. Si la fonctionnelle n'est que partiellement différentiable, on montre qu'il est avantageux d'exploiter la régularité partielle de la fonctionnelle, ce qui conduit à des méthodes dites par *éclatement*. Dans le cas plus général, on introduit un opérateur dit *proximal*, qui est à l'origine d'une classe d'algorithmes dit *proximaux*. Ces derniers comprennent en particulier les méthodes classiques de gradient implicite, gradient explicite ou encore gradient projeté. Une autre sorte de régularité fournit également des résultats intéressants : il s'agit de la *forte convexité*. Cette propriété permet en outre de proposer des algorithmes accélérés, comme on le verra dans le chapitre 5. Enfin, un dernier aspect important en optimisation convexe reste la complexité des calculs. Pour qu'un algorithme soit utilisable sur des données réelles, il est nécessaire d'en assurer la convergence dans un temps raisonnable. On verra dans le chapitre 6 qu'une stratégie d'éclatement peut en réduire la complexité, en permettant notamment les calculs parallèles. Néanmoins, ce genre d'approches implique des résolutions approchées.

L'objectif de ce chapitre est de donc de rappeler et d'établir certains résultats classiques en optimisation convexe continue, en vue de les appliquer dans les deux prochains chapitres. Nous commencerons par des rappels sur le cadre de la convexité (section 1.1). On verra ensuite le cas plus connu des fonctions différentiables, puis on introduira la notion de sous-différentiabilité. Cette notion est au coeur de la théorie des opérateurs proximaux (section 1.3) qui offre une classe très large d'algorithmes, appelés algorithmes proximaux, qui généralisent les méthodes de gradient (section 1.4). Enfin, on présentera des stratégies classiques qui permettent d'exploiter les propriétés de régularité d'une seule partie de la fonctionnelle (méthodes d'éclatement).

1.1 Convexité

Dans cette section, on fait quelques rappels sur les fonctions convexes, en considérant le cas général des fonctions à valeurs dans $\mathbb{R} \cup \{\pm\infty\}$. Ce choix nous permettra en particulier de considérer des problèmes de minimisation sous contraintes, mais sous une forme non contrainte. On rappellera ensuite les résultats d'existence de minimum, puis les conditions d'optimalité du premier ordre, d'abord dans le cas familier des fonctions différentiables, puis dans le cas plus général des fonctions sous-différentiables. Enfin, on introduira les fonctions fortement convexes, pour lesquelles on verra plus tard qu'il est possible de proposer des algorithmes accélérés.

1.1.1 Définitions

Commençons par quelques définitions classiques.

Domaine, fonctions propres Soit X un espace hilbertien, de dual noté X^*. On va considérer dans ce chapitre des fonctions à valeurs dans la ligne réelle étendue $X \to \mathbb{R} \cup \{\pm\infty\}$. On appellera alors *domaine* de la fonction F, noté $\mathrm{dom}(F)$ l'ensemble des points $x \in X$ tels que $f(x) < +\infty$.

Une fonction $F : E \to \mathbb{R} \cup \{\pm\infty\}$ est dite *propre* si elle ne prend pas la valeur $-\infty$ et si elle n'est pas identiquement égale à $+\infty$. Autrement dit, le domaine d'une fonction propre n'est pas vide.

Inégalité de Jensen Un ensemble $C \subset X$ est dit *convexe* si, pour tous élements x_1 et x_2 de C, le segment $[\,x_1\,;x_2\,]$ défini par $\{\lambda\, x_1 + (1-\lambda)\, x_2 \mid \lambda \in [\,0\,;1\,]\}$ est contenu dans C. Une fonction $F : E \to \mathbb{R} \cup \{+\infty\}$ est dite *convexe* si son épigraphe (c'est-à-dire l'ensemble des points situés au-dessus de son graphe) est un ensemble convexe. Elle est dite *concave* si $-F$ est convexe.

On montre que F est convexe si et seulement si elle vérifie l'inégalité de Jensen, qui pour tout couple $(x_1,x_2) \in (\mathrm{dom}(F))^2$ s'écrit

$$\forall\, \lambda \in]\,0\,;1\,[\,, \qquad F\Big(\lambda\, x_1 + (1-\lambda)\, x_2\Big) \leq \lambda\, F(x_1) + (1-\lambda)\, F(x_2).$$

Si cette inégalité est stricte pour tout $x_1 \neq x_2$, alors F est dite *strictement convexe*.

1.1.2 Existence d'un minimiseur

Un des intérêts de l'optimisation convexe repose sur le fait qu'il n'existe pas de minimum local dans lequel les algorithmes de recherche de minimum par descente pourraient être piégés, comme l'atteste le résultat suivant :

Proposition 1 *Soit F une fonction convexe sur X. Si F admet un minimum local en x^*, alors il admet un minimum global en x^*.*

Un autre intérêt des fonctions convexes est l'existence de minimiseurs dans des cas simples à caractériser. Voici dans ce qui suit quelques résultats connus d'existence (et/ou d'unicité) de minimiseurs. Pour plus de détails, le lecteur pourra se reporter par exemple à [10].

Cas général On s'intéresse tout d'abord aux fonctions convexes (non strictement convexes). Commençons par le cas des fonctions continues sur un compact, pour lesquelles l'existence d'un minimiseur est assuré :

Proposition 2 *Soit F une fonction convexe sur un compact $K \subset X$ non vide. On suppose que F est continue sur K. Alors F admet au moins un minimum dans K.*

On enchaîne ensuite avec le cas non borné ; ce cas nécessite de considérer des fonctions dites *coercives*, c'est-à-dire telles que

$$F(x) \to +\infty \qquad \text{si} \qquad \|x\| \to +\infty.$$

On peut alors montrer que, dans ce cas, l'existence d'un minimiseur est également assurée :

Proposition 3 *Soit F une fonction convexe sur X. On suppose que F est continue et coercive sur X. Alors F admet au moins un minimum dans X.*

Ce résultat se généralise à la minimisation sur un fermé quelconque non vide.

Cas de la stricte convexité Lorsqu'on ajoute une hypothèse de stricte convexité, les résultats qui précèdent incluent un résultat d'unicité. En effet, on peut montrer que

Proposition 4 *Soit F une fonction strictement convexe sur X. Alors F admet au plus un minimum sur X.*

Cette proposition nous permet donc d'énoncer un résultat important sur la minimisation d'une fonction strictement convexe, lorsque celle-ci est coercive :

Théorème 1 *Soit F une fonction strictement convexe et coercive sur X. Soit A un fermé non vide. On suppose que F est continue sur A. Alors F admet exactement un minimum dans A.*

Dans tout ce qui suit, on supposera toujours (sans le préciser) l'existence d'au moins un minimiseur.

1.1.3 Conditions d'optimalité

On présente dans ce paragraphe des résultats utiles permettant de caractériser, lorsqu'ils existent, les minima des fonctions considérées. On se focalise plus particulièrement sur les conditions dites *du premier ordre*, qui concernent les fonctions différentiables ou sous-différentiables, et qui donnent un critère sur le gradient ou le sous-gradient.

Cas différentiable Commençons par traiter le cas plus familier des fonctions différentiables. Les conditions nécessaires d'optimalité sont connus sous le nom d'équation ou d'inégalité d'Euler.

Théorème 2 **(Équation d'Euler)** *Soit F une fonction convexe sur X. On suppose que F est différentiable sur X. Alors F admet un minimum en x^* si et seulement si x^* vérifie l'équation d'*Euler

$$\nabla F(x^*) = 0.$$

Ce résultat est également valable sur tout ouvert convexe $\Omega \subset X$. Il **n'est pas valable** sur des ensembles fermés (où le minimum, s'il existe, peut être atteint sur le bord). C'est l'objet du résultat suivant, généralisable à tout convexe Ω :

Proposition 5 **(Inégalité d'Euler)** *Soit F une fonction convexe sur X. On suppose que F est différentiable sur X. Alors F admet un minimum en x^* si et seulement si x^* vérifie l'inégalité d'*Euler

$$\forall\, x \in X, \qquad \langle x - x^*, \nabla F(x^*) \rangle \geq 0.$$

Sous-différentiabilité On quitte maintenant le cadre des fonctions différentiables. Commençons par introduire la notion de sous-différentiabilité, qui généralise celle de la différentiabilité dans le cas des fonctions convexes. Soit $x_0 \in X$ tel que $x_0 \in \mathrm{dom}(F)$ avec F une fonction convexe. On définit le *sous-différentiel* de F en x_0, noté $\partial F(x_0)$, comme étant l'ensemble des points $p \in X^*$ vérifiant

$$\forall\, x \in X, \qquad \langle x - x_0, p\rangle + F(x_0) \leq F(x)$$

appelés, quand ils existent, *sous-gradients de F en x_0*. On dit alors que F est *sous-différentiable en x_0* si son sous-différentiel en x_0 est non vide. Par convention, on définit le sous-différentiel de F en x_0 comme étant l'ensemble vide si $F(x_0) = +\infty$. On peut montrer par ailleurs que le sous-différentiel en x_0 d'une fonction convexe F différentiable en x_0 est donné par le singleton $\{\nabla F(x_0)\}$. Réciproquement, on établit que, si le sous-différentiel de F en x_0 est réduit à un vecteur p, alors F est différentiable en x_0, de gradient p.

Calcul de sous-différentiel Établissons ici quelques règles de calcul de sous-différentiel qui nous seront utiles par la suite. Commençons par remarquer que, pour tout $\alpha > 0$, on a

$$\forall\, x_0 \in \mathrm{dom}(F), \qquad \partial(\alpha F)(x_0) = \alpha\, \partial F(x_0).$$

En effet, on a par définition du sous-différentiel

$$\begin{aligned} x \in \partial(\alpha F)(x_0) \quad &\Longleftrightarrow \quad \forall\, x \in X, \quad \langle x - x_0, p\rangle + \alpha\, F(x_0) \leq \alpha\, F(x) \\ &\Longleftrightarrow \quad \forall\, p \in X, \quad \left\langle x - x_0, \frac{x}{\alpha}\right\rangle + F(x_0) \leq F(x) \\ x \in \partial(\alpha F)(x_0) \quad &\Longleftrightarrow \quad \frac{x}{\alpha} \in \partial F(x_0). \end{aligned}$$

Supposons à présent que f est une fonction convexe différentiable et F convexe. Posons $G = F + f$. Calculons $\partial G(x_0)$ pour tout $x_0 \in \mathrm{dom}(F) \cap \mathrm{dom}(f)$. Si $p \in \partial F(x_0)$, alors

$$\forall\, x \in X, \qquad \langle x - x_0, p\rangle + F(x_0) \leq F(x).$$

On a par ailleurs, puisque $\partial f(x_0) = \{\nabla f(x_0)\}$,

$$\forall\, x \in X, \qquad \langle x - x_0, \nabla f(x_0)\rangle + f(x_0) \leq f(x).$$

En additionnant les deux, on montre que $p + \nabla f(x_0) \in \partial G(x_0)$. Ainsi, on prouve que $\partial F(x_0) + \nabla f(x_0) \subset \partial G(x_0)$ [2]. Supposons maintenant que $x \in \partial G(x_0)$ et démontrons l'inclusion inverse. On a par définition

$$\forall\, x \in X, \forall\, \lambda \in \left]\, 0\,;1\,\right[, \qquad \langle [\lambda x + (1-\lambda)x_0] - x_0, p\rangle + G(x_0) \leq G\Big(\lambda x + (1-\lambda)x_0\Big)$$

soit $\lambda\langle x - x_0, p\rangle + F(x_0) + f(x_0) \leq \lambda\, F(x) + (1-\lambda)\, F(x_0) + f\Big(x_0 + \lambda(x - x_0)\Big)$.

En simplifiant et en utilisant la formule de Taylor-Young au premier ordre pour f, on obtient que

$$\begin{aligned} \lambda\langle x - x_0, p\rangle + f(x_0) \leq \lambda\, F(x) - \lambda\, F(x_0) + f(x_0) + \lambda\langle x - x_0, \nabla f(x_0)\rangle \\ + \lambda\, \|x - x_0\|\, \varepsilon\Big(\lambda(x - x_0)\Big). \end{aligned}$$

2. En réalité, on démontre aussi que, de manière générale, $\partial F_1(x_0) + \partial F_2(x_0) \subset \partial(F_1 + F_2)(x_0)$ pour tout $x_0 \in X$.

En divisant à nouveau par λ puis en le faisant tendre vers 0, il s'ensuit que

$$\forall\, x \in X, \qquad \langle x - x_0, p - \nabla f(x_0)\rangle + F(x_0) \leq F(x).$$

On en déduit que $p - \nabla f(x_0) \in \partial F(x_0)$, donc $\partial G(x_0) \subset \partial F(x_0) + \nabla f(x_0)$. Finalement, si f est différentiable, alors

$$\forall\, x_0 \in X, \qquad \partial G(x_0) = \partial F(x_0) + \nabla f(x_0).$$

Cas sous-différentiable Généralisons à présent les conditions d'optimalité de premier ordre au cas des fonctions sous-différentiables :

Théorème 3 (Équation d'Euler (2)) *Soit F une fonction convexe sur X. On suppose que F est sous-différentiable sur X. Alors F admet un minimum en x^* si et seulement si x^* vérifie l'équation d'*Euler

$$0 \in \partial F(x^*).$$

Démonstration : Puisque $\langle y - x_0, 0\rangle$ est nul pour tout $y \in X$, on en déduit que x_0 minimise F si et seulement si $F(x_0) < +\infty$ et que

$$\forall\, y \in X, \qquad \langle y - x_0, 0\rangle + F(x_0) \leq F(y)$$

c'est-à-dire si et seulement si $0 \in \partial F(x_0)$. ■

1.1.4 Forte convexité

Introduisons enfin la notion de forte convexité, qui apparaîtra en particulier dans le chapitre suivant.

Définition Une fonction $F : E \to \mathbb{R} \cup \{+\infty\}$ est dite *fortement convexe*, de module $\alpha > 0$, si pour tout couple $(x_1, x_2) \in (\mathrm{dom}(F))^2$

$$\forall \lambda \in]\,0\,;1\,[\,, \qquad F\Big(\lambda x_1 + (1-\lambda)\, x_2\Big) \leq \lambda\, F(x_1) + (1-\lambda)\, F(x_2) - \frac{\alpha}{2}\, \lambda(1-\lambda)\, \|x_1 - x_2\|^2.$$

Une fonction fortement convexe est en particulier strictement convexe.

Exemples La fonction $x \mapsto \alpha\, \|x + a\|^2/2$ est fortement convexe de module α pour tout $a \in X$ et tout $\alpha > 0$. On peut également montrer que la somme d'une fonction convexe et d'une fonction fortement convexe est fortement convexe.

On peut enfin établir que, si la fonction F est fortement convexe de module α, alors, pour tout $x \in X$, la fonction

$$y \mapsto F(y) - \frac{\alpha}{2}\, \|x - y\|^2$$

est convexe.

Caractérisation Établissons à présent une caractérisation bien utile de la forte convexité :

Proposition 6 *Soit F une fonction définie sur l'espace euclidien X. Supposons que F est sous-différentiable. Alors F est fortement convexe de module α si et seulement si*

$$\forall\,(x_1,x_2)\in(\mathrm{dom}(F))^2,\qquad F(x_2)\geq F(x_1)+\langle p,x_2-x_1\rangle+\frac{\alpha}{2}\,\|x_2-x_1\|^2$$

avec $p\in\partial F(x_2)$.

Démonstration : • Commençons par prouver le sens direct. Supposons que F est fortement convexe de module α. Par définition, on a pour tout $(x_1,x_2)\in(\mathrm{dom}(F))^2$ et $\lambda\in\,]\,0\,;1\,[$

$$F\Big(\lambda x_2+(1-\lambda)x_1\Big)\leq\lambda\,F(x_2)+(1-\lambda)\,F(x_1)-\frac{\alpha}{2}\,\lambda(1-\lambda)\,\|x_2-x_1\|^2.$$

que l'on peut réécrire

$$F\Big(x_1+\lambda(x_2-x_1)\Big)-F(x_1)\leq\lambda\,[F(x_2)-F(x_1)]-\frac{\alpha}{2}\,\lambda(1-\lambda)\,\|x_2-x_1\|^2.$$

Soit maintenant $p\in\nabla F(x_1)$. Par définition, p vérifie

$$\forall z\in X,\qquad\langle z-x_1,p\rangle+F(x_1)\leq F(z).$$

En particulier, pour $z=x_1+\lambda(x_2-x_1)$,

$$\langle x_1+\lambda(x_2-x_1)-x_1,p\rangle+F(x)\leq F\Big(x_1+\lambda(x_2-x_1)\Big)$$

soit

$$\lambda\,\langle x_2-x_1,p\rangle\leq F(x+\lambda(x_2-x_1))-F(x_1).$$

Ainsi, on obtient que

$$\lambda\,\langle x_2-x_1,p\rangle\leq\lambda\,[F(x_2)-F(x_1)]-\frac{\alpha}{2}\,\lambda(1-\lambda)\,\|x_2-x_1\|^2$$

et puisque λ est strictement positif, on peut simplifier par λ, puis faire tendre λ vers 0 et obtenir l'inégalité recherchée.

• Montrons maintenant l'autre sens. Supposons que F vérifie pour tout $(x,x')\in(\mathrm{dom}(F))^2$

$$F(x')\geq F(x)+\langle p,x'-x\rangle+\frac{\alpha}{2}\,\|x'-x\|^2$$

avec $p\in\nabla F(x)$. Soit $(x_1,x_2)\in(\mathrm{dom}(F))^2$ et $\lambda\in\,]\,0\,;1\,[$. Appliquons cette relation à $x=\lambda\,x_1+(1-\lambda)\,x_2$ et $x'=x_1$:

$$F\Big(\lambda\,x_1+(1-\lambda)\,x_2\Big)\geq F(x_1)+\langle p,\lambda\,x_1+(1-\lambda)\,x_2-x_1\rangle+\frac{\alpha}{2}\,\|\lambda\,x_1+(1-\lambda)\,x_2-x_1\|^2$$

puis à $x'=x_2$:

$$F\Big(\lambda\,x_1+(1-\lambda)\,x_2\Big)\geq F(x_2)+\langle p,\lambda\,x_1+(1-\lambda)\,x_2-x_2\rangle+\frac{\alpha}{2}\,\|\lambda\,x_1+(1-\lambda)\,x_2-x_2\|^2$$

ce qui donne respectivement, en simplifiant,

$$F\Big(\lambda\,x_1+(1-\lambda)\,x_2\Big)\geq F(x_1)+(1-\lambda)\,\langle p,x_2-x_1\rangle+\frac{\alpha}{2}\,(1-\lambda)^2\,\|x_1-x_2\|^2$$

et

$$F\Big(\lambda\,x_1+(1-\lambda)\,x_2\Big)\geq F(x_2)+\lambda\,\langle p,x_1-x_2\rangle+\frac{\alpha}{2}\,\lambda^{\,2}\,\|x_1-x_2\|^2.$$

En multipliant la première inégalité par λ, la seconde par $(1-\lambda)$, puis en les ajoutant, on obtient finalement la relation de forte convexité souhaitée, ce qui achève la preuve. ■

Minimisation Commençons par remarquer que la caractérisation de la forte convexité assure que, si x^* est le minimiseur (unique) de F, alors on a

$$\forall\, x \in \mathrm{dom}(F), \qquad F(x) \geq F(x^*) + \frac{\alpha}{2}\,\|x - x^*\|^2$$

puisque x^* vérifie l'équation d'Euler. On peut par ailleurs signaler le résultat suivant [1] :

Proposition 7 *Soit F une fonction fortement convexe de module α définie sur l'espace euclidien X. Alors F admet un unique minimiseur x^*, et toute suite minimisante (c'est-à-dire toute suite $(x_n)_n$ telle que $(f(x_n))_n$ converge vers $f(x^*)$) converge vers x^*. Par ailleurs, on a pour tout $x \in X$*

$$\|x - x^*\|^2 \leq \frac{4}{\alpha}\Big(F(x) - F(x^*)\Big).$$

1.2 Dualité

1.2.1 Conjuguée convexe ou conjuguée de Legendre-Fenchel

On va introduire dans cette section deux notions importantes, qui sont la semi-continuité inférieure et la conjuguée convexe (ou conjuguée de Legendre-Fenchel, déjà rencontrée dans le chapitre 3).

Semi-continuité inférieure Une fonction $F : X \to \mathbb{R} \cup \{+\infty\}$ est dite *semi-continue inférieurement* (abrégé en s.c.i. [3]) si

$$\forall\, x^* \in X, \qquad \liminf_{x \to x^*} F(x) \geq F(x^*).$$

On peut montrer [10] que F est s.c.i. si et seulement si son épigraphe est fermé.

Conjuguée convexe Soit F une fonction convexe définie sur X. On définit sa conjuguée convexe, ou encore conjuguée de Legendre-Fenchel, en posant

$$\forall\, y \in X^*, \qquad F^*(y) = \sup_{x \in X} \Big\{ \langle x,y \rangle - F(x) \Big\}.$$

Commençons par remarquer que F^* est à valeurs dans $\mathbb{R} \cup \{+\infty\}$ car F est propre. On peut par ailleurs montrer que F^* est convexe et s.c.i. Signalons également le résultat suivant :

Théorème 4 *Si F est s.c.i., alors sa* biconjuguée $F^{**} = (F^*)^*$, *qui est la conjuguée convexe de sa conjuguée convexe, est F elle-même. Autrement dit, on a la relation suivante*

$$\forall\, x \in X, \qquad F(x) = \sup_{y \in X^*} \Big\{ \langle x,y \rangle - F^*(y) \Big\}.$$

Une manière d'interpréter ce résultat est de montrer que la biconjuguée F^{**} de F est en réalité la plus grande fonction convexe, propre et s.c.i. située en-dessous de F. On a donc l'égalité lorsque F possède déjà ces propriétés.

3. En anglais, l.s.c. pour *lower semicontinuous*.

Cas de la forte convexité Lorsque la fonction F est fortement convexe, cela induit une régularité remarquable sur sa conjuguée convexe, comme en témoigne le résultat suivant [2, 8] :

Théorème 5 *Soit F une fonction convexe.*

- *Si F est différentiable et ∇F est lipschitzienne, de constante L, alors F^* est fortement convexe de module $(1/L)$.*
- *Si F est fortement convexe, de module α, alors F^* est différentiable, de gradient lipschitzien, de constante de* Lipschitz $1/\alpha$.

En particulier, si F est convexe, propre et s.c.i., alors $F^{**} = F$, ce qui implique que F est fortement convexe de module α si et seulement si F^* est différentiable, de gradient $(1/\alpha)$-lipschitzien. C'est pourquoi la forte convexité peut être interprétée comme une forme de régularité.

1.2.2 Point-selle

Enfin, rappelons quelques notions utiles à propos des points-selles.

Définition et propriétés Un couple $(\bar{x},\bar{y}) \in X \times Y$ est un point-selle de la fonction $\mathcal{L}$ sur $X \times Y$ si

$$\forall\, x \in X, \quad \forall\, y \in Y, \qquad \mathcal{L}(\bar{x},y) \leq \mathcal{L}(\bar{x},\bar{y}) \leq \mathcal{L}(x,\bar{y}).$$

On peut montrer qu'une fonction $\mathcal{L}$ à valeurs réelles définie sur $X \times Y$ possède un point-selle $(\bar{x},\bar{y})$ sur $X \times Y$ si et seulement si

$$\max_{y\in Y} \inf_{x\in X} \mathcal{L}(x,y) = \min_{x\in X} \sup_{y\in Y} \mathcal{L}(x,y)$$

Ce nombre est alors égal à $\mathcal{L}(\bar{x},\bar{y})$ (appelée *valeur-selle*).

Cas convexe-concave On suppose que pour tout $y \in Y$, la fonction $f_y : x \mapsto \mathcal{L}(x,y)$ est convexe, et que, pour tout $x \in Y$, la fonction $g_x : y \mapsto \mathcal{L}(x,y)$ est concave. On dit alors que la fonction $\mathcal{L}$ est *convexe-concave*. Dans ce cas, l'existence d'un point-selle est assurée par le théorème suivant :

Théorème 6 *Soit $\mathcal{L} : X \times Y \to \mathbb{R}$ une fonction convexe-concave. Posons pour tout $y \in Y$ et pour tout $x \in X$:*

$$f_y : x \mapsto \mathcal{L}(x,y) \qquad \text{et} \qquad g_x : y \mapsto \mathcal{L}(x,y).$$

On suppose que tout $y \in Y$, la fonction g_y est s.c.i. et que pour tout $x \in X$, la fonction f_x est s.c.i. Alors $\mathcal{L}$ possède un point-selle sur $X \times Y$.

Conditions d'optimalité Voyons comment on peut caractériser les points selle d'une fonction $\mathcal{L}$ dans le cas convexe-concave :

Théorème 7 (Équation d'Euler) *Soit $\mathcal{L} : X \times Y \to \mathbb{R}$ une fonction convexe-concave et sous-différentiable. Posons pour tout $y \in Y$ et pour tout $x \in X$:*

$$f_y : x \mapsto \mathcal{L}(x,y) \qquad et \qquad g_x : y \mapsto \mathcal{L}(x,y).$$

Alors $(\bar{x},\bar{y})$ est un point-selle de $\mathcal{L}$ si et seulement si

$$0 \in \partial f_{\bar{y}}(\bar{x}) \qquad et \qquad 0 \in \partial g_{\bar{x}}(\bar{y}).$$

1.3 Opérateur proximal

On introduit dans cette section un opérateur, appelé *opérateur proximal*, introduit par Jean-Jacques Moreau [12]. Il permet de concevoir une classe de méthodes d'optimisation convexe applicables à des fonctions non différentiables (mais sous-différentiables), qui généralisent les méthodes de descente de gradient.

1.3.1 Définition et caractérisation

Point et opérateur proximal Soit F une fonction convexe, s.c.i. et propre, définie sur l'espace euclidien X (avec $d \in \mathbb{N}^*$). On définit pour tout $x \in X$:

$$\operatorname{prox}_F(x) = \operatorname*{argmin}_{y \in X} \left\{ \frac{1}{2}\|x-y\|^2 + F(y) \right\}.$$

appelé *point proximal* de x relativement à la fonction F. L'opérateur qui à tout x associe son point proximal relativement à F est appelé *opérateur proximal* (ou opérateur de proximité) associé à F.

Commençons par vérifier que, pour tout $x \in X$, le point proximal $\operatorname{prox}_F(x)$ est bien défini. Soit $x \in X$. On remarque tout d'abord que la fonction

$$G : y \mapsto \frac{1}{2}\|x-y\|^2 + F(y)$$

est strictement convexe et s.c.i. car c'est la somme d'une fonction fortement convexe et d'une fonction convexe, toutes deux s.c.i. On en déduit que la fonction G admet un unique minimum, ce qui assure la bonne définition de $\operatorname{prox}_F(x)$.

Caractérisation du point proximal On cherche à présent à donner une caractérisation plus manipulable de $\operatorname{prox}_f(x)$. Comme il s'agit du minimiseur d'une fonction convexe, on peut le caractériser à l'aide de l'équation d'Euler. C'est l'objet du résultat suivant :

Proposition 8 *Soit F une fonction convexe, s.c.i. et propre. Alors pour tout $x \in X$,*

$$p = \operatorname{prox}_F(x) \qquad \Longleftrightarrow \qquad x - p \in \partial F(p)$$

Démonstration : • Soit $x \in X$. Supposons que $p \in X$ vérifie $F(p) < +\infty$. Par définition du sous-gradient, $x - p \in \partial F(p)$ est défini par

$$\forall\, y \in X, \qquad \langle y - p, x - p\rangle + F(p) \leq F(y).$$

Or, puisque $\langle y-p,x-p\rangle = \|y-p\|^2/2+\|x-p\|^2/2-\|y-x\|^2/2$, cette définition s'écrit

$$\frac{1}{2}\|y-p\|^2 + \frac{1}{2}\|x-p\|^2 - \frac{1}{2}\|x-y\|^2 + F(p) \leq F(y)$$

soit $$\forall\, y \in X, \qquad \frac{1}{2}\|y-p\|^2 + \frac{1}{2}\|x-p\|^2 + F(p) \leq \frac{1}{2}\|x-y\|^2 + F(y).$$

On en déduit en particulier que, si $x - p \in \partial F(p)$, alors

$$\forall\, y \in X, \qquad \frac{1}{2}\|x-p\|^2 + F(p) \leq \frac{1}{2}\|x-y\|^2 + F(y)$$

ce qui, par définition de l'optimalité, assure que $p = \mathrm{prox}_F(x)$.

• Supposons à présent que $p = \mathrm{prox}_F(x)$. On remarque déjà que, dans ce cas, on a nécessairement $F(p) < +\infty$, car sinon, on aurait $\|p - y\|^2/2 + F(p) = +\infty$ ce qui est absurde pour le minimum d'une fonction propre. On peut donc définir le sous-différentiel de F en p. Par définition de l'optimalité, on a pour tout $y \in X$ et pour tout $\lambda \in]\,0\,;1\,[$

$$\frac{1}{2}\|x-p\|^2 + F(p) \leq \frac{1}{2}\|x-[\lambda p + (1-\lambda)y]\|^2 + F\Big(\lambda p + (1-\lambda)y\Big).$$

On a alors d'une part

$$\begin{aligned}\|x-[\lambda p + (1-\lambda)y]\|^2 &= \|(x-p) + (1-\lambda)(p-y)\|^2 \\ &= \|x-p\|^2 + (1-\lambda)^2\|p-y\|^2 - 2(1-\lambda)\,\langle y-p,x-p\rangle\end{aligned}$$

et d'autre part, par convexité,

$$F\Big(\lambda p + (1-\lambda)y\Big) \leq \lambda F(p) + (1-\lambda)F(y).$$

Il s'ensuit que, après simplification, on a pour tout $y \in X$ et pour tout $\lambda \in]\,0\,;1\,[$,

$$(1-\lambda)\,\langle y-p,x-p\rangle + (1-\lambda)F(p) \leq \frac{1}{2}(1-\lambda)^2\|p-y\|^2 + (1-\lambda)F(y).$$

En divisant par $1-\lambda$ puis en faisant tendre λ vers 1, on montre alors que $x-p$ est un sous-gradient de F en p. ■

Puisque $p = \mathrm{prox}_F(x)$ peut être caractérisé par la relation $x - p \in \partial F(p)$, soit, en d'autres termes, $x \in p + \partial F(p)$ (l'addition s'entendant de manière ensembliste), on peut formellement définir l'opérateur

$$\mathrm{I} + \partial F : \left\{ \begin{array}{ccc} E & \to & \mathcal{P}(E) \\ p & \mapsto & p + \partial F(p). \end{array} \right.$$

Dans ce cas, $p = \mathrm{prox}_F(x)$ équivaut alors à $x \in (\mathrm{I} + \partial F)(p)$, et comme p est unique, cela nous permet de noter l'opérateur proximal

$$\mathrm{prox}_F = (\mathrm{I} + \partial F)^{-1}.$$

Projection sur un convexe fermé L'opérateur proximal peut être vu comme la généralisation de la projection sur un convexe fermé. Commençons par rappeler le résultat suivant :

Théorème 8 *Soit $C \subset E$ un convexe fermé. Il existe une unique application* proj_C *de E dans C, appelée* projection sur le convexe C, *qui à tout $x \in E$ associe le point* $\mathrm{proj}_C(x)$ *de C, telle que la distance de x à C soit égale à celle de x à* $\mathrm{proj}_C(x)$. *Le vecteur* $\mathrm{proj}_C(x)$ *est l'unique point de C vérifiant les deux propositions équivalentes suivantes :*

$$\forall\, y \in C, \quad \big\|x - \mathrm{proj}_C(x)\big\| \leq \big\|x - y\big\| \tag{1.1}$$

$$\forall\, y \in C, \quad \Big\langle x - \mathrm{proj}_C(x), y - \mathrm{proj}_C(x)\Big\rangle \leq 0.$$

La relation (1.1) permet en particulier de caractériser le point $\mathrm{proj}_C(x)$:

$$\mathrm{proj}_C(x) = \operatorname*{argmin}_{y \in C} \left\{ \frac{1}{2}\|x - y\|^2 \right\}.$$

Or, si C est un convexe fermé de X, alors la fonction caractéristique χ_C définie par

$$\forall\, x \in X, \qquad \chi_C(x) = \begin{cases} 0 & \text{si } x \in C \\ +\infty & \text{sinon} \end{cases}$$

est convexe. Montrons à présent que $\mathrm{prox}_{\chi_C} = \mathrm{proj}_C$. En effet, pour tout $x \in X$, on a par définition

$$\mathrm{prox}_{\chi_C}(x) = \operatorname*{argmin}_{y \in X} \left\{ \frac{1}{2}\|x - y\|^2 + \chi_C(y) \right\} = \operatorname*{argmin}_{y \in C} \left\{ \frac{1}{2}\|x - y\|^2 \right\} = \mathrm{proj}_C(x).$$

1.3.2 Identité de Moreau

Décomposition de Moreau Si $X \subset X$ est un sous-espace vectoriel de X, alors on a classiquement la décomposition orthogonale suivante [4] :

$$E = E \oplus E^{\perp}$$

où $X^{\perp}$ est l'ensemble des points qui sont orthogonaux aux points de X, c'est-à-dire

$$E^{\perp} = \Big\{ y \in X \mid \forall\, x \in X, \langle x, y\rangle = 0 \Big\}.$$

On a alors la relation suivante :

$$\forall\, x \in X, \qquad x = \mathrm{proj}_E(x) + \mathrm{proj}_{E^{\perp}}(x).$$

D'après la remarque du paragraphe précédent, puisque X et $X^{\perp}$ sont convexes, on peut réécrire la formule précédente en utilisant des opérateurs proximaux :

$$\forall\, x \in X, \qquad x = \mathrm{prox}_{\chi_E}(x) + \mathrm{prox}_{\chi_{E^{\perp}}}(x).$$

Il est à présent utile de remarquer que $\chi_{E^{\perp}}$ est la conjuguée convexe de χ_E. En effet,

$$\forall\, y \in X, \qquad (\chi_E)^*(y) = \sup_{x \in X} \Big\{ \langle x, y\rangle - \chi_E(x) \Big\} = \sup_{x \in X} \Big\{ \langle x, y\rangle \Big\} = \begin{cases} 0 & \text{si } y \in X^{\perp} \\ +\infty & \text{si } y \notin E^{\perp} \end{cases}$$

ce qui implique que $\forall\, x \in X, \qquad x = \mathrm{prox}_{\chi_E}(x) + \mathrm{prox}_{(\chi_E)^*}(x)$.

On peut généraliser ce résultat à n'importe quel opérateur proximal : c'est l'identité de Moreau.

4. que l'on peut généraliser à un sous-espace fermé d'un espace de Hilbert.

Proposition 9 (Identité de Moreau) *Soit F une fonction convexe, s.c.i. et propre. Alors on a*

$$\forall\, x \in X, \qquad x = \mathrm{prox}_F(x) + \mathrm{prox}_{F^*}(x).$$

Démonstration : Soit $x \in X$. Posons $u = \mathrm{prox}_F(x)$ et $v = x - u$, et montrons que $v = \mathrm{prox}_{F^*}(x)$. La caractérisation de u assure que $x - u \in \partial F(u)$, soit $v \in \partial F(u)$. Par définition du sous-différentiel, on en déduit que

$$\forall\, y \in X, \qquad \langle y - u, v\rangle + F(u) \leq F(y)$$

soit

$$\forall\, y \in X, \qquad \langle u, v\rangle - F(u) \geq \langle y, v\rangle - F(y)$$

Cette dernière relation étant valable pour tout $y \in X$, on peut passer à la borne supérieure, ce qui entraîne que

$$\langle u, v\rangle - F(u) \geq \sup_{y \in X} \Big\{ \langle y, v\rangle - F(y) \Big\} = F^*(v).$$

Par ailleurs, pour tout $y \in X$, on a par optimalité que $F^*(y) \geq \langle u, y\rangle - F(u)$, ce qui implique que

$$\forall\, y \in X, \qquad \langle u, v - y\rangle - F^*(y) \geq F^*(v) \qquad \text{soit} \qquad \langle y - v, u\rangle + F^*(v) \leq F^*(y)$$

ce qui assure que $x - v \in \partial(F^*)(v)$, soit $v = \mathrm{prox}_{F^*}(x)$. ■

On peut encore généraliser ce résultat, et montrer que pour tout $\gamma > 0$,

$$\forall\, x \in X, \qquad x = \mathrm{prox}_{\gamma F}(x) + \gamma\ \mathrm{prox}_{F^*/\gamma}(x/\gamma).$$

Ce résultat est à nouveau appelé *identité de* Moreau.

1.4 Optimisation convexe et méthodes proximales

Soit F une fonction convexe s.c.i. propre. On cherche à résoudre le problème d'optimisation suivant :

$$\min_{x \in X} F(x) \tag{1.2}$$

où F prend généralement la forme d'une somme de fonctions convexes, et est coercive. Elle admet alors un minimum. On s'attachera ici à étudier principalement les cas où F est une fonction convexe ou la somme de deux fonctions convexes (étant entendu qu'on peut très souvent se ramener à l'un de ces deux cas).

Étudions dans cette section différentes méthodes d'optimisation convexe connues à la lumière des opérateurs proximaux. On supposera que le problème étudié admet au moins une solution. Pour une revue plus complète sur ce sujet, le lecteur pourra se reporter à [6].

1.4.1 Méthodes de gradient

Ces méthodes nécessitent que les fonctions en jeu soient régulières (au moins sous-différentiables).

Gradient explicite Commençons par considérer le cas où F est une fonction convexe différentiable, de gradient continu sur X et lipschitzienne, de constante de Lipschitz L. Tout minimum x^* satisfait la condition d'optimalité de premier ordre ($\nabla F(x^*) = 0$). On peut en particulier écrire, pour tout $\tau > 0$, la relation de point fixe

$$x^* - \tau \nabla F(x^*) = x^*.$$

On peut alors considérer l'algorithme itératif de recherche de point fixe :

$$x_0 \in X \qquad \text{et} \qquad \forall\, k \in \mathbb{N}, \qquad x_{k+1} = x_k - \tau \nabla F(x_k) \tag{1.3}$$

connu sous le nom de *méthode (de descente) de gradient explicite* (explicite car on évalue le gradient de F au point précédent déjà connu). Il faut bien sûr s'assurer que ce schéma itératif converge. Par exemple, si F est supposée convexe et ∇F lipschitzienne de constante L, alors un développement de Taylor assure la convergence de ce schéma dès que $\tau < 2/L$.

La méthode du gradient explicite est une méthode bien connue pour résoudre le problème (1.2). Elle s'interprète de la manière suivante : la convexité de la fonction F assurant la non-existence de minima locaux, il suffit pour trouver x^* à partir d'un point x_0 de réaliser des pas de descentes, c'est-à-dire de s'approcher de x^* en suivant une direction dans laquelle F décroît. Pour cela, si F est différentiable, la meilleure direction (localement) est celle donnée par l'opposé du gradient. L'itération (1.3) revient donc à effectuer un pas (fixe) dans cette direction. On voit alors que le choix du pas τ est crucial, en particulier lorsqu'on s'approche de x^* : s'il est choisit trop grand, on dépasse x^* lorsqu'on s'en approche, tandis que, s'il est choisi trop petit, la convergence est trop longue. Notons enfin que le choix de τ dans les deux cas particuliers abordés repose sur la connaissance de certaines constantes (de Lipschitz et éventuellement de la forte convexité de F) qui ne sont pas toujours accessibles.

Gradient implicite Relâchons l'hypothèse de régularité sur ∇F. On peut alors écrire, pour tout $\tau > 0$, la relation

$$x^* + \tau \nabla F(x^*) = x^* \qquad \text{soit} \qquad x^* + \partial(\tau F)(x^*) = \{x^*\}.$$

Ainsi, $(\mathrm{I} + \partial(\tau F))(x^*) = x^*$, qu'on peut encore écrire $x^* = (\mathrm{I} + \partial(\tau F))^{-1}(x^*)$. On en déduit que le minimum est caractérisé par la nouvelle relation de point fixe

$$x^* = \mathrm{prox}_{\tau F}(x^*).$$

Si on sait calculer l'opérateur proximal de F, cette recherche de point fixe incite donc à proposer cette fois le schéma itératif suivant :

$$x_0 \in X \qquad \text{et} \qquad \forall\, k \in \mathbb{N}, \qquad x_{k+1} = \mathrm{prox}_{\tau F}(x_k) \tag{1.4}$$

Cet algorithme est appelé *algorithme du point proximal*. Il a été proposé pour la première fois en 1970 par Bernard Martinet [11]. Réécrivons l'algorithme du point proximal autrement : par définition de l'opérateur proximal,

$$\forall\, n \in \mathbb{N}, \qquad x_{k+1} = (\mathrm{I} + \partial(\tau F))^{-1}(x_k)$$

qu'on peut écrire $\qquad (\mathrm{I} + \partial(\tau F))(x_{k+1}) = x_{k+1} + \tau \nabla F(x_{k+1}) = x_k$

car ici, F est différentiable. L'algorithme proposé devient alors :

$$x_0 \in X \qquad \text{et} \qquad \forall\, n \in \mathbb{N}, \qquad x_{k+1} = x_k - \tau \nabla F(x_{k+1})$$

plus connu sous le nom de *méthode (de descente) de gradient implicite*, car, écrite sous cette forme, pour trouver le point x_{k+1}, on descend dans la direction du gradient de F au point x_{k+1} que l'on est en train de calculer.

La méthode du gradient implicite est *a priori* plus difficile à mettre en place que celle du gradient implicite, car il s'agit d'évaluer le gradient en un point non connu. Si cette opération est réalisable, elle peut alors présenter un intérêt pour le cas où ∇F n'est pas lipschitzien (et pour lequel la convergence de la méthode de gradient explicite n'est pas assurée). On peut en effet considérer la fonction auxiliaire ${}^{\gamma}F$, définie pour tout $\gamma > 0$ par

$$\forall\, x \in X, \qquad {}^{\gamma}F(x) = \min_{y \in X} \left\{ \frac{1}{2\,\gamma} \|x - y\|^2 + F(y) \right\}$$

appelée *enveloppe de* Moreau d'indice γ de la fonction F. Cette fonction est convexe, s.c.i. et propre, et est différentiable, de gradient

$$\forall\, x \in X, \qquad \nabla {}^{\gamma}F(x) = \frac{1}{\gamma}\,(x - \mathrm{prox}_{\gamma\, F}(x)) = \mathrm{prox}_{F^*/\gamma}(x/\gamma).$$

On montre alors en particulier que l'itération (1.4) s'écrit

$$x_{k+1} = x_k - \tau\, \nabla {}^{\tau}F(x_k)$$

qui s'interprète comme une itération de gradient explicite pour la fonction auxiliaire ${}^{\gamma}F$. On peut alors choisir un pas de descente τ assurant la convergence de cet algorithme si $\nabla {}^{\tau}F$ est lipschitzienne par exemple. C'est bien le cas ici, car on peut montrer que la constante de Lipschitz de $\nabla {}^{\tau}F$ vaut $1/\tau$. La méthode de gradient implicite est donc plus stable que celle de gradient explicite. Un autre intérêt de cette approche réside dans le fait qu'elle est facilement généralisable au cas sous-différentiable, comme on le verra au paragraphe suivant.

Sous-gradient implicite Lorsque F n'est pas différentiable, mais sous-différentiable, la condition d'optimalité de premier ordre devient $0 \in \partial F(x^*)$, ce qui permet à nouveau d'écrire la caractérisation du minimum par la relation de point fixe suivante, valable pour tout $\tau > 0$,

$$x^* = \mathrm{prox}_{\tau F}(x^*).$$

Cela conduit donc au même algorithme itératif :

$$x_0 \in X \qquad \text{et} \qquad \forall\, k \in \mathbb{N}, \qquad x_{k+1} = \mathrm{prox}_{\tau F}(x_k)$$

Cette méthode peut à nouveau être interprétée comme une méthode de *sous-gradient implicite*, car l'algorithme considéré peut s'écrire de manière équivalente :

$$x_0 \in X \qquad \text{et} \qquad \forall\, n \in \mathbb{N}, \qquad x_{k+1} = x_k - \tau\, g_{k+1} \qquad \text{où} \qquad g_{k+1} \in \partial F(x_{k+1}).$$

Gradient explicite-implicite On se place maintenant dans le cas où F est de la forme $F = f + g$, avec f différentiable et g sous-différentiable ; on supposera de plus que ∇f est continue et L-lipschitzienne. On s'intéresse donc au problème suivant

$$\min_{x \in X} \Big\{ f(x) + g(x) \Big\}.$$

Il est possible d'appliquer la méthode de sous-gradient implicite à la fonction F, mais on choisit ici de présenter une méthode qui permet de tirer parti de la différentiabilité d'une partie de la fonction F.

La condition d'optimalité de premier ordre assure que, si on note x^* l'optimum, alors on a

$$0 \in \tau\,\partial F(x^*) \qquad \text{soit} \qquad 0 \in \tau\,\nabla f(x^*) + \tau\,\partial g(x^*)$$

i.e. $$x^* - \tau\,\nabla f(x^*) - x^* \in \partial(\tau g)(x^*)$$

où τ est un réel strictement positif quelconque. On en déduit que le problème d'optimisation est équivalent au problème de recherche de point fixe

$$x^* = \operatorname{prox}_{\tau g}\Big(x^* - \tau\,\nabla f(x^*)\Big).$$

On peut alors proposer l'algorithme itératif de recherche de point fixe

$$x_0 \in X \qquad \text{et} \qquad \forall\, k \in \mathbb{N}, \qquad x_{k+1} = \operatorname{prox}_{\tau g}\Big(x_k - \tau\,\nabla f(x_k)\Big)$$

en choisissant correctement le paramètre τ afin d'en assurer la convergence. On décompose généralement cet algorithme en deux étapes :

1. on commence par évaluer un point intermédiaire $x_{k+1/2} = x_k - \tau\,\nabla f(x_k)$; cette étape ne met en jeu que la fonction f, et s'interprète comme une descente de gradient *explicite* ;
2. on calcule ensuite $x_{k+1} = \operatorname{prox}_{\tau g}(x_{k+1/2})$: ce calcul ne dépend que de la fonction g, et s'interprète d'après ce qui précède comme une descente de gradient *implicite*.

Cet algorithme est également connu sous le nom de *forward-backward splitting*. Tout comme pour la méthode de gradient explicite, le pas τ doit être choisi en fonction de la constante de Lipschitz de la partie différentiable de F, afin d'assurer la convergence du schéma proposé.

Dans toutes ces méthodes de gradient, des stratégies de pas τ_n variables peuvent alors être envisagées pour accélérer la convergence. Il est également possible d'ajouter des pas de relaxation (qui consiste à utiliser des points intermédiaires, situés sur la droite reliant le point précédant et le point courant).

Application : méthode du gradient projeté On présente maintenant une application très classique des algorithmes présentés ci-dessus. Considérons une minimisation sur un convexe $C \subset X$, c'est-à-dire le problème

$$\min_{x \in C} f(x)$$

où f est différentiable, de gradient lipschitzien. En remarquant que la contrainte d'appartenant à un convexe peut être intégrée dans une fonction caractéristique χ_C, on peut se ramener au cas précédent en écrivant le problème comme le problème sans contrainte suivant

$$\min_{x \in X} F(x) \qquad \text{avec} \qquad F(x) = f(x) + g(x) \qquad \text{et} \qquad g = \chi_C.$$

La fonction g est ici sous-différentiable. Ainsi, on cherche à trouver le point fixe x^*

$$x^* = \operatorname{prox}_{\tau g}\Big(x^* - \tau\,\nabla f(x^*)\Big) \qquad \text{soit} \qquad x^* = \operatorname{prox}_{g}\Big(x^* - \tau\,\nabla f(x^*)\Big)$$

Or, on rappelle que l'opérateur proximal associé à une fonction caractéristique d'un convexe est la projection sur ce même convexe ; on cherche donc à résoudre

$$x^* = \operatorname{proj}_C\Big(x^* - \tau\,\nabla f(x^*)\Big).$$

La méthode du gradient explicite-implicite s'écrit alors

$$x_0 \in X \qquad \text{et} \qquad \forall\, k \in \mathbb{N}, \qquad x_{k+1} = \operatorname{proj}_C\Big(x_k - \tau\,\nabla f(x_k)\Big)$$

ce qui revient à effectuer dans un premier temps une descente de gradient (explicite), puis à projeter le point ainsi trouvé sur le convexe C, d'où l'appellation de *méthode du gradient projeté*.

1.4.2 Méthodes d'éclatement

On a vu avec la méthode de gradient implicite-explicite qu'il est parfois possible d'exploiter séparément les propriétés des termes composant la fonction F. Dans le cas cité, il s'agit de profiter de la différentiabilité d'un des deux termes, qui est une propriété de régularité plus forte que la sous-différentiabilité. Ce genre d'approches est connu sous le nom de méthode d'*éclatement* (*splitting* en anglais). On propose dans cette section deux autres méthodes d'éclatement classiques.

Méthode d'éclatement de Dykstra La méthode d'éclatement de Dykstra s'applique aux problèmes de la forme

$$\min_{x\in X}\left\{F(x)+\frac{1}{2}\|x-u\|^2\right\} \qquad \text{avec} \qquad F(x)=f(x)+g(x) \tag{1.5}$$

où u est un vecteur de X donné, f et g deux fonctions convexes sous-différentiables. Dans le cas où les fonctions f et g sont les fonctions caractéristiques d'ensembles convexes, on voit que ce problème s'interprète comme la projection sur l'intersection des deux convexes du vecteur u. C'est dans ce cadre que cette méthode a été initialement proposée (c'est pourquoi elle est également connue sous le nom de *méthode de projection de* Dykstra). Cette méthode peut être utilisée lorsque l'opérateur proximal associé à F n'est pas calculable (ou difficilement), mais que ceux associés à f et g respectivement le sont. L'idée est donc d'exploiter la calculabilité de ces deux opérateurs.

On peut réécrire le problème (1.5) en utilisant les conjuguées convexes : on commence par écrire que, pour tout $x \in X$,

$$F(x)+\frac{1}{2}\|x-u\|^2=\sup_{x_1,x_2\in X}\left\{-f^*(x_1)-g^*(x_2)+\frac{1}{2}\|x-u\|^2+\langle x,x_1+x_2\rangle\right\}.$$

Puisque

$$\frac{1}{2}\|x-u\|^2+\langle x,x_1+x_2\rangle=\frac{1}{2}\|x+x_1+x_2-u\|^2-\frac{1}{2}\|x_1+x_2-u\|^2+\frac{1}{2}\|u\|^2,$$

on en déduit que le problème (1.5) est équivalent à

$$\min_{x\in X}\sup_{x_1,x_2\in X}\left\{-f^*(x_1)-g^*(x_2)+\frac{1}{2}\|x+x_1+x_2-u\|^2-\frac{1}{2}\|x_1+x_2-u\|^2+\frac{1}{2}\|u\|^2\right\}$$

et donc également à

$$\max_{x_1,x_2\in X}\inf_{x\in X}\left\{-f^*(x_1)-g^*(x_2)+\frac{1}{2}\|x+x_1+x_2-u\|^2-\frac{1}{2}\|x_1+x_2-u\|^2+\frac{1}{2}\|u\|^2\right\}.$$

Or, pour tout $x_1,x_2 \in X$, on remarque que

$$\begin{aligned}\inf_{x\in X}\left\{-f^*(x_1)-g^*(x_2)+\frac{1}{2}\|x+x_1+x_2-u\|^2-\frac{1}{2}\|x_1+x_2-u\|^2+\frac{1}{2}\|u\|^2\right\}\\ =-f^*(x_1)-g^*(x_2)-\frac{1}{2}\|x_1+x_2-u\|^2+\frac{1}{2}\|u\|^2\end{aligned}$$

où le minimum est atteint pour $x^* = u - x_1 - x_2$. On peut donc montrer que pour résoudre le problème (1.5), il suffit de résoudre le problème dual

$$\min_{x_1,x_2\in X}\left\{f^*(x_1)+g^*(x_2)+\frac{1}{2}\|x_1+x_2-u\|^2\right\}. \tag{1.6}$$

Si on note (x_1^*,x_2^*) la solution de ce problème fortement convexe, la solution x^* du problème initial (primal) est alors donnée par

$$x^* = u - x_1^* - x_2^*.$$

Le problème dual (1.6) peut être résolu par minimisation alternée : on minimise le lagrangien dual par rapport (par exemple) à y_1 pour y_2 fixé, puis l'inverse (avec ou non une mise-à-jour de la première variable duale entre les deux minimisations). Chacune de ces deux minimisations partielles peut être interprétée comme l'évaluation des opérateurs proximaux associés à f et g respectivement.

Méthode de Douglas-Rachford On relâche dans ce paragraphe l'hypothèse de régularité sur les fonctions à minimiser, mais on suppose que les fonctions sont convexes, s.c.i. et propres, ce qui nous permet de manipuler leur conjuguée convexe. L'idée est à nouveau d'exploiter la possibilité d'évaluer les opérateurs proximaux de chacun des termes composant le lagrangien, alors que celui de la somme n'est pas calculable.

On cherche à minimiser le problème de la forme [5]

$$\min_{x\in X} \Big\{ f(x) + g(x) \Big\} \tag{1.7}$$

où les fonctions f et g sont toutes les deux convexes, s.c.i. et propres. On suppose par ailleurs que leur somme définit une fonction coercive. Ce problème admet donc au moins une solution. Pour la caractériser, on commence par noter que, puisque g est s.c.i. et propre, elle est égale à sa biconjuguée convexe, et le problème étudié devient donc

$$\min_{x\in X} \Big\{ f(x) + g^{**}(x) \Big\} = \min_{x\in X} \Big\{ f(x) + \sup_{y\in X} \Big\{ \langle x,y\rangle - g^*(y) \Big\} \Big\}$$

ce qui nous amène à considérer le problème de recherche de point-selle

$$\min_{x\in X} \sup_{y\in X} \Big\{ f(x) + \langle x,y\rangle - g^*(y) \Big\}.$$

Les solutions (x^*,y^*) de ce problème satisfont les équations d'Euler

$$-y^* \in \partial f(x^*) \qquad \text{et} \qquad x^* \in \partial g^*(y^*)$$

ce qui implique que, pour tout $\tau > 0$,

$$x^* - \tau\, y^* \in x^* + \tau\, \partial f(x^*) \qquad \text{et} \qquad y^* + \tau^{-1}\, x^* \in y^* + \tau^{-1}\, \partial g^*(y^*)$$

soit
$$x^* = (\mathrm{I} + \tau\, \partial f)^{-1}(x^* - \tau\, y^*) = \mathrm{prox}_{\tau f}(x^* - \tau\, y^*)$$

et
$$y^* = \Big(\mathrm{I} + \tau^{-1}\, \partial g^*\Big)^{-1}(y^* + \tau^{-1}\, x^*) = \mathrm{prox}_{\tau^{-1}\, g^*}(y^* + \tau^{-1}\, x^*).$$

L'identité de Moreau assure alors que

$$\mathrm{prox}_{\tau^{-1}\, g^*}(y^* + \tau^{-1}\, x^*) = y^* + \tau^{-1}\, x^* - \tau^{-1}\, \mathrm{prox}_{\tau g}\Big(\tau\, (y^* + \tau^{-1}\, x^*)\Big)$$

ce qui entraîne donc

$$y^* = y^* + \tau^{-1}\, x^* - \tau^{-1}\, \mathrm{prox}_{\tau g}\Big(\tau\, (y^* + \tau^{-1}\, x^*)\Big)$$

5. La méthode proposée par Jim Douglas et Henri Rachford dans [9] en 1956 visait originellement à résoudre des problèmes linéaires de la forme $u = A\, x + B\, x$, avec A et B des matrices définies positives.

soit $\mathrm{prox}_{\tau g}(x^* + \tau\, y^*) - x^* = 0$. Ainsi, les solutions du problème (1.7) sont caractérisées par (pour $\lambda > 0$)

$$\begin{cases} x^* = \mathrm{prox}_{\tau f}(x^* - \tau\, y^*) \\ 0 = \lambda \Big[\mathrm{prox}_{\tau g}(2\, x^* - x^* + \tau\, y^*) - x^* \Big] \end{cases}$$

soit, en posant $y = x^* - \tau\, y^*$ et en ajoutant y dans la dernière équation,

$$\begin{cases} x^* = \mathrm{prox}_{\tau f}(y) \\ y = y + \lambda \Big[\mathrm{prox}_{\tau g}(2\, x^* - y) - x^* \Big]. \end{cases} \tag{1.8}$$

Voici donc un algorithme proposé pour résoudre le problème (1.7) basée sur la relation (1.8) :

$$x_0 \in X, \qquad \text{et} \qquad \forall\, k \in \mathbb{N}, \qquad \begin{cases} x_{k+1} &= \mathrm{prox}_{\tau f}(y_k) \\ y_{k+1} &= y_k + \lambda_k \Big[\mathrm{prox}_{\tau g}(2\, x_{k+1} - y_k) - x_{k+1} \Big] \end{cases}$$

où λ_k est choisi dans l'intervalle $[\,\varepsilon\,;2-\varepsilon\,]$, avec $\varepsilon > 0$.

1.4.3 Itérations de Bregman

Enfin, signalons la possibilité d'utiliser des distances de Bregman [3] pour proposer une variante des algorithmes proximaux. On peut en effet voir dans la définition du point proximal la norme au carré comme une distance ; les auteurs de [5] ont proposé de remplacer cette distance par une classe de pseudo-distances, étudiée par Bregman [4], on peut appliquer les méthodes proximales en remplaçant chaque calcul de point proximal par une itération dite de Bregman, mais en gagnant en vitesse de convergence.

Distance de Bregman Soit H une fonction strictement convexe et différentiable. On définit la pseudo-distance suivante, qu'on appellera désormais *distance de* Bregman *associée à la fonction* H :

$$\forall\, (x,y) \in E^2, \qquad D_x^H(x,y) = H(y) - H(x) - \langle \nabla H(x), y - x \rangle.$$

Notons que cette distance n'est pas symétrique par rapport à x et y. Puisque la fonction H est supposée strictement convexe, on montre que la quantité $D_x^H(x,y)$ est toujours positive quels que soient x et y. Par ailleurs, il est immédiat que $D_x^H(x,x) = 0$.

On peut aisément vérifier que, si $H = \|\cdot\|^2$, alors $D_x^H(x,y) = \|x-y\|^2/2$.

Convergence des itérations de Bregman L'intérêt des distances de Bregman [14] réside dans le fait qu'il est possible de trouver un réel $\alpha > 0$ tel que

$$\forall\, (x,y) \in E^2, \qquad D_x^{\alpha\, H}(x,y) \geq \frac{1}{2}\, \|x-y\|^2.$$

On supposera donc désormais que H est telle que l'inégalité précédente soit vraie pour $\alpha = 1$. Une conséquence de cette hypothèse est l'existence d'un unique minimum pour la fonction $y \mapsto F(y) + D_x^H(x,y)$ quelle que soit F une fonction convexe. Il est donc possible de définir une version généralisée de l'opérateur proximal, en remplaçant dans sa définition la norme euclidienne par une distance de Bregman :

$$\underset{y \in E}{\mathrm{argmin}} \Big\{ F(y) + D_x^H(x,y) \Big\}.$$

On a de plus le résultat suivant :

Théorème 9 *Soient F une fonction s.c.i., convexe et propre et $x \in X$. Si*

$$x^* = \operatorname*{argmin}_{y \in X} \Big\{ F(y) + D_y^H(y,x) \Big\}$$

alors on a

$$\forall\, y \in X, \qquad F(y) + D_y^H(y,x) \geq F(x^*) + D_{x^*}^H(x^*,x) + D_y^H(y,x^*).$$

Cette propriété permet généralement de remplacer dans les algorithmes proximaux l'évaluation du point proximal par la minimisation de la fonction $y \mapsto F(y) + D_y^H(y,x)$ (lorsque celle-ci est calculable), tout en conservant la validité des preuves de convergence. Dans [5] par exemple, les auteurs ont démontré la convergence de l'algorithme du point proximal utilisant une distance de Bregman. L'intérêt de les utiliser peut être multiple [13] : il peut permettre de s'affranchir de certaines hypothèses de régularité (sur ∇F par exemple) dans les méthodes de gradient ; l'évaluation de l'opérateur proximal peut être plus simple lorsque la distance de Bregman est adaptée au problème (voir le paragraphe suivant).

Exemples de distances de Bregman Un premier exemple simple de distances de Bregman est donné par

$$H(x) = \|x\|_M = \sqrt{\langle Mx,x \rangle}$$

où M est une matrice symétrique définie positive. La fonction H définit dans ce cas une norme, et on vérifie que $D_x^H(x,y) = \|x-y\|_M^2/2$ est une distance de Bregman.

Considérons un second exemple. Commençons par introduire la fonction dite *d'entropie*, définie par

$$\forall\, x = (x_i) \in \mathbb{R}^d, \qquad H(x) = \begin{cases} \displaystyle\sum_{i=0}^{d-1} x_i \ln x_i & \text{si } x \in \Sigma \\ +\infty & \text{sinon} \end{cases}$$

avec $0 \ln 0 = 0$ par convention, avec Σ le simplexe de $\mathbb{R}^d$, défini par l'ensemble des vecteurs x de $\mathbb{R}^d$ à coefficients positifs et de somme 1. Posons $h(t) = t \ln t$ pour tout réel positif t. La fonction h est strictement convexe (car dérivable sur $\mathbb{R}_+^*$, de dérivée strictement croissante). On en déduit la stricte convexité de H, ce qui implique que H définit une distance de Bregman. Cette version est en particulier utilisée pour minimiser certaines fonctions (supposées différentiables) sur le simplexe, qui consiste à résoudre pour $x \in \mathbb{R}^d$

$$\operatorname*{argmin}_{y \in \Sigma} f(y) = \operatorname*{argmin}_{y \in \mathbb{R}^d} \Big\{ f(y) + \chi_\Sigma(y) \Big\}.$$

En posant $F = f + \chi_\Sigma$, écrivons les itérations de la méthode du sous-gradient implicite pour ce problème :

$$x_0 \in \mathbb{R}^d \qquad \text{et} \qquad \forall\, k \in \mathbb{N}, \qquad x_{k+1} = \operatorname{prox}_{\tau F}(x_k)$$

où la mise-à-jour de x_{k+1} s'écrit explicitement

$$x_{k+1} = \operatorname*{argmin}_{y \in \mathbb{R}^d} \Big\{ \tau\, F(y) + \frac{1}{2} \|x_k - y\|^2 \Big\}.$$

Si on remplace l'évaluation de cet opérateur proximal par une itération de Bregman, alors on est amené à résoudre à la place

$$x_{k+1} = \operatorname*{argmin}_{y\in\mathbb{R}^d} \Big\{ \tau\, F(y) + D^H_{x_k}(x_k,y) \Big\}.$$

La définition de H assure que ce dernier problème s'écrit

$$x_{k+1} = \operatorname*{argmin}_{y\in\mathbb{R}^d} \Big\{ \tau\, f(y) + H(y) - H(x_k) - \langle \nabla H(x_k), y - x_k \rangle \Big\}.$$

La fonctionnelle est différentiable sur Σ et on peut résoudre explicitement ce problème à l'aide d'un multiplicateur de Lagrange pour la contrainte (égalité) sur la somme des coefficients de x_{k+1}. Plus précisément, x_{k+1} est défini comme la solution du problème de minimisation

$$\min_{y\in\mathbb{R}^d} \left\{ \tau\, f(y) + \sum_{i=0}^{d-1} y_i \ln y_i - \sum_{i=0}^{d-1} (x_k)_i \ln(x_k)_i - \sum_{i=0}^{d-1} \Big(1 + \ln(x_k)_i\Big)(y_i - (x_k)_i) \right\}$$

sous la contrainte égalité $\displaystyle\sum_{i=0}^{d-1} y_i = 1.$

Les conditions de Kuhn–Tucker s'écrivent pour ce problème

$$\forall\, i \in [\![\, 0\,;\, d-1\,]\!]\,, \qquad \tau\, \frac{\partial f}{\partial y_i}\Big((x_{k+1})_i\Big) + 1 + \ln(x_{k+1})_i - (1 + \ln(x_k)_i) + \lambda = 0$$

On en déduit que

$$\forall\, i \in [\![\, 0\,;\, d-1\,]\!]\,, \qquad \tau\, \frac{\partial f}{\partial y_i}\Big((x_{k+1})_i\Big) + \ln(x_{k+1})_i = \ln(x_k)_i - \lambda.$$

Ainsi, si les $\displaystyle\tau\, \frac{\partial f}{\partial y_i} + \ln$ sont inversibles, on a

$$\forall\, i \in [\![\, 0\,;\, d-1\,]\!]\,, \qquad (x_{k+1})_i = \left(\tau\, \frac{\partial f}{\partial y_i} + \ln\right)^{-1} \Big(\ln(x_k)_i - \lambda\Big)$$

et la contrainte égalité assure que λ doit être solution de l'équation

$$\sum_{i=0}^{d-1} \left(\tau\, \frac{\partial f}{\partial y_i} + \ln\right)^{-1} \Big(\ln(x_k)_i - \lambda\Big) = 1.$$

Cette méthode peut être envisagée dans le cas où $f(x) = \langle a,x\rangle$ par exemple.

Pour une revue récente sur les algorithmes de projection sur le simplexe, on pourra également se référer à [7].

Références

[1] Grégoire Allaire. *Analyse numérique et optimisation : Une introduction à la modélisation mathématique et à la simulation numérique.* Éditions École polytechnique, 2005.

[2] Heinz H. Bauschke and Patrick L. Combettes. *Convex analysis and monotone operator theory in* Hilbert *spaces.* Springer Science & Business Media, 2011.

[3] Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3) :167–175, 2003.

[4] Lev M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3) :200–217, 1967.

[5] Yair Censor and Stavros Andrea Zenios. Proximal minimization algorithm with D-functions. *Journal of Optimization Theory and Applications*, 73(3) :451–464, 1992.

[6] Patrick L. Combettes and Jean-Christophe Pesquet. Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer, 2011.

[7] Laurent Condat. Fast projection onto the simplex and the ℓ_1 ball. *Mathematical Programming*, pages 1–11, 2014.

[8] Damek Davis and Wotao Yin. Faster convergence rates of relaxed Peaceman–Rachford and ADMM under regularity assumptions. *arXiv preprint arXiv :1407.5210*, 2014.

[9] Jim Douglas and Henry H. Rachford. On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American mathematical Society*, 82(2) :421–439, 1956.

[10] Ivar Ekeland and Roger Témam. *Convex Analysis and Variational Problems*, volume 28. SIAM, 1999.

[11] Bernard Martinet. Brève communication. régularisation d'inéquations variationnelles par approximations successives. *Revue Française d'Informatique et de Recherche Opérationnelle, série rouge*, 4(3) :154–158, 1970.

[12] Jean-Jacques Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société Mathématique de France*, 93 :273–299, 1965.

[13] Van Quang Nguyen. *Méthodes d'éclatement basées sur les distances de* Bregman *pour les inclusions monotones composites et l'optimisation.* PhD thesis, Paris 6, 2015.

[14] Paul Tseng. On accelerated proximal gradient methods for convex-concave optimization. *Submitted to SIAM J. Optim*, 2008.

Chapitre 2

Comment voit-on en relief ?

Introduction

La capacité d'un humain à percevoir le relief repose principalement sur sa vision binoculaire, appelée *stéréoscopie*. Grâce à ses yeux, il voit le monde depuis deux points de vue légèrement différents, desquels son cerveau extrait une vue unique en relief. Cette propriété a été observée dès le X^{e} siècle, notamment par un savant nommé Alhazen [22]. Dans un traité de François d'Aiguilon, publié en 1613, une illustration de Rubens dépeint ainsi un vieil homme borgne appréciant mal les distances à cause de sa monophtalmie [1] [22, page 16]. Cette particularité de la vision humaine reste cependant peu exploitée, jusqu'à l'invention de la photographie et des premiers stéréogrammes. Ces derniers sont composés de deux photographies d'une même scène, légèrement décalées l'une par rapport à l'autre. Au début du XX^{e} siècle, ils connaissent un succès important, sous le nom de cartes stéréoscopiques. Grâce à un stéréoscope, dont le premier modèle est inventé dès 1838 par Charles Wheatstone, ces cartes offrent une

1. Le fait de ne voir que d'un seul œil.

vue *en relief* des scènes photographiées, à partir de développements photographiques *plans*. Plus récemment, la stéréoscopie connaît un regain de popularité avec les films dits *en 3 dimensions* : le spectateur, équipé de lunettes spéciales, expérimente une projection du film où personnages, objets et décors semblent posséder volume et profondeur réalistes. Les constructeurs de consoles de jeu et de téléviseurs ne sont pas en reste et ont conçu des écrans offrant un rendu en relief des images affichées. Dans tous les cas, le principe est le même : ce que voit l'œil gauche diffère de ce que voit l'œil droit, ce qui permet au cerveau de reconstituer une information de relief.

Dans le domaine du traitement de l'image et de la vision par ordinateur, la stéréovision binoculaire est depuis des décennies une branche très active, notamment depuis la mise en ligne en 2001 du banc d'essai MIDDLEBURY [2] par Daniel SCHARSTEIN, Richard SZELISKI et Heiko HIRSCHMÜLLER [36]. L'objectif est reconstruire le relief d'une scène à partir de deux photographies de celle-ci, prises de deux points de vue différents, connaissant les paramètres des systèmes optiques impliqués dans la prise de vue. Nous verrons qu'il s'agit fondamentalement d'un problème de *mise en correspondance* (section 2.1). Du fait d'un phénomène appelé *occultation*, il est malheureusement mal posé (section 2.2) et, par ailleurs, difficile à résoudre. De nombreuses stratégies ont été explorées ces dernières années (section 2.3), mais nous nous pencherons dans ce mémoire sur une classe de méthodes dites *globales*, avec d'une part une approche reposant sur une relaxation convexe du problème variationnel sous-jacent (chapitre 3) et d'autre part une méthode de coupures de graphes (*graph cuts*) tirant parti de l'efficacité des algorithmes de flot maximal (chapitre 4).

2.1 La mise en correspondance stéréoscopique

2.1.1 Modèle de formation des images

Modèle sténopé Commençons par présenter le modèle de formation des images classiquement choisi en stéréovision binoculaire. Il s'agit du modèle dit *sténopé* [3]. Dans ce modèle, le système optique (l'appareil photographique) est caractérisé par son *plan image* et son *centre optique*, la distance entre ces deux éléments étant appelée *distance focale*. On appelle alors *scène* le demi-espace délimité par le plan image et ne contenant pas le centre optique. La prise de vue est ainsi modélisée : tout point physique de la scène est visible par ce système optique s'il existe une droite (qui modélise la trajectoire du rayon lumineux) reliant sans obstacle le point physique au centre optique. Son *image* par ce système optique est alors l'intersection de cette droite avec le plan image [4]. On pourra se reporter à la figure 2.1 pour mieux visualiser le modèle décrit. On utilisera par ailleurs désormais l'anglicisme *caméra* pour désigner l'appareil photographique.

Cadre et champ d'une caméra En pratique, les photographies ont un domaine fini et rectangulaire, appelé *cadre* de la caméra. Les points physiques de la scène dont la projection sur le plan image est située à l'intérieur du cadre de la caméra forment le *champ* de la caméra. Les autres points sont dit *hors-champ*. Sauf mention contraire, nous ne considérons désormais plus que les points physiques du champ de la caméra,

2. http ://vision.middlebury.edu/stereo/
3. Appelé *pin-hole model* en anglais
4. On considère à des fins de clarté l'image virtuelle des points de la scène, car elle n'est pas inversée (le haut et le bas sont en particulier préservés), contrairement à l'image réelle, qui se trouve elle sur le *plan image* du système, symétrique du plan image par rapport au centre optique.

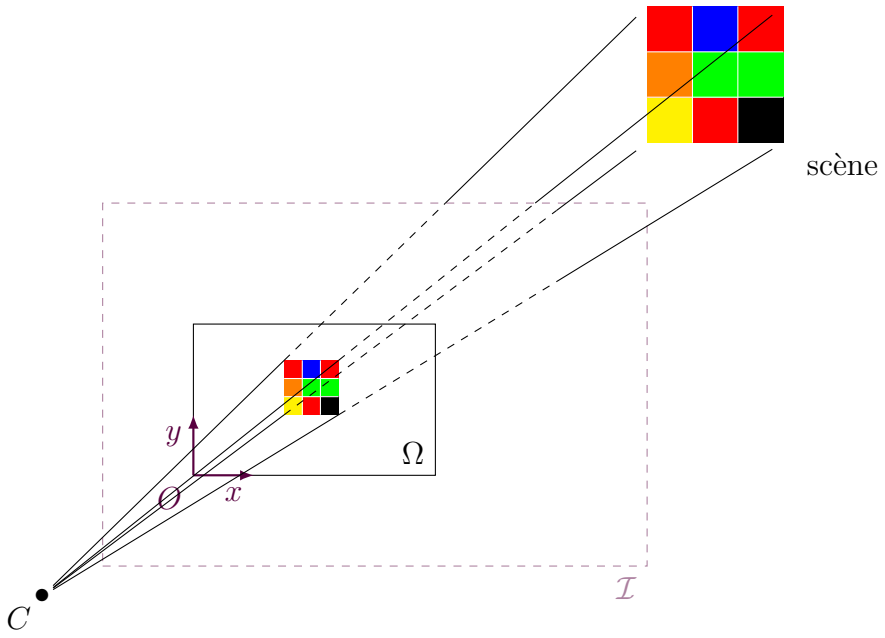


FIGURE 2.1 – Modèle sténopé de formation des images. Le système optique est modélisé par son centre optique C, son plan image $\mathcal{I}$ et sa distance focale f. La scène correspond au demi-espace ne contenant pas C et délimité par le plan $\mathcal{I}$. Le domaine Ω de l'image est matérialisé par le rectangle en trait plein, qui est muni d'un repère orthonormé. On choisit par convention de placer l'origine de ce repère au coin inférieur gauche de ce domaine, les deux directions du repère étant données par les côtés du rectangle.

et le terme *domaine de l'image* désignera la restriction du plan image au cadre de la caméra. Notons que le fait d'être situé dans le champ de la caméra n'assure pas à un point d'être visible par celle-ci.

Paramètres intrinsèques On munit le plan image d'un repère orthonormé. Son origine est le coin inférieur gauche du cadre de la caméra et les deux axes sont portés par les deux côtés issus de l'origine. Appelons *point principal* le projeté du centre optique sur le plan image. La distance focale et les coordonnées dans le repère précédemment introduit du point principal sont appelées *paramètres intrinsèques* de la caméra. La donnée des paramètres intrinsèques d'une caméra et de son cadre est suffisante pour en déduire toutes les caractéristiques du système optique étudié. On notera que, dans le cas des caméras réelles, le centre optique se projette généralement sur le centre du cadre[5], auquel cas on parlera de *caméra parfaite*.

Intensité d'un pixel On distinguera le *point physique* de la scène $M \in \mathbb{R}^3$, de coordonnées (X,Y,Z) dans un repère donné de l'espace, de sa projection (si elle existe) $m \in \mathbb{R}^2$ sur le plan image, de coordonnées (x,y) dans le repère de l'image, que l'on appellera *pixel*. Une *image* I désigne une fonction qui, à tout pixel du cadre de la caméra, associe son intensité, enregistrée par la caméra. L'intensité désigne de manière générique le niveau de gris dans le cas des images en niveaux de gris ou la couleur dans le cas des images couleurs. On choisit comme système de représentation des couleurs le système RGB (*red, green, blue*). L'image I est donc une fonction, définie sur le domaine rectangulaire $\Omega \subset \mathbb{R}^2$, et à valeurs dans $\mathbb{R}$ ou dans $\mathbb{R}^3$. En l'absence de bruit ou d'aberration

5. C'est pourquoi le point principal est parfois appelé *centre de l'image*.

chromatique et sauf cas particulier (surface réfléchissante, par exemple), l'intensité d'un pixel ne dépend que du point M correspondant, et ne varie donc pas selon le point de vue.

2.1.2 Géométrie épipolaire et rectification des images

Paire stéréoscopique Supposons maintenant que la scène est photographiée par deux caméras, caractérisées par leur plan image, leur centre optique, leur distance focale et le domaine de leur image. On supposera ce dernier de dimension identique pour les deux caméras. La scène est alors définie comme l'intersection des champs associés aux deux caméras. On impose pour le moment les contraintes suivantes :

- les deux centres optiques sont distincts ;
- chaque centre optique n'appartient pas à la scène de l'autre caméra ;
- on écarte le cas trivial où la scène est vide.

La première condition élimine le cas d'une simple rotation de la caméra autour de son centre optique. La seconde évite en particulier que l'une des deux caméras soit visible par l'autre (et notamment que les deux caméras se fassent face).

En pratique, les images sont capturées soient par la même caméra, qui se déplace dans l'espace, soit par deux caméras simultanément. Dans le premier cas, les paramètres intrinsèques de la caméra restent inchangés, mais les objets de la scène peuvent avoir bougé entre deux prises de vue (par exemple : des voitures pour les vues aériennes). Dans le second cas, les paramètres intrinsèques des deux caméras peuvent être différents.

Droites épipolaires, plan épipolaire Soit M un point de la scène. Les contraintes présentées plus haut assurent que le point M et les deux centres optiques, notés O_{L} et O_{R}, ne peuvent être alignés, car la droite $(O_{\mathrm{L}},O_{\mathrm{R}})$ ne peut être dans le champ des deux caméras à la fois. Ils définissent donc un plan, que l'on appelle *plan épipolaire* associé au point M. Ce plan coupe le plan image de la caméra de gauche selon une droite, appelée *droite épipolaire* de l'image de gauche associée au point M et notée $\ell_L(M)$ et coupe de la même manière le plan image de la caméra de droite selon la droite épipolaire de l'image de droite associée au point M et notée $\ell_R(M)$. Le pixel m_{L}, image du point M par la caméra de gauche, appartient à la droite épipolaire $\ell_L(M)$, tandis que l'image m_{R} du point M par la caméra de droite, appartient à la droite épipolaire $\ell_R(M)$.

Déplacement fronto-parallèle de la caméra Dans le cas général, pour un point M donné, les droites épipolaires associées ont des directions totalement arbitraires. On va à présent imposer certaines contraintes sur les droites épipolaires et en déduire les conditions nécessaires sur les deux systèmes optiques que cela entraîne.

On demande dans un premier temps que, pour tout point M, les droites épipolaires soient confondues dans les deux images. Ces droites appartenant chacune au plan image de sa caméra associée, on en déduit que les deux plans image doivent être confondus.

On souhaite dans un second temps contraindre toutes les droites épipolaires à être horizontales, c'est-à-dire parallèles à l'axe horizontal du repère de leur image respective. Supposons donc que c'est le cas. Soient M et M' deux points dont les droites épipolaires ℓ et ℓ' (qui sont maintenant les mêmes dans les deux images) sont distinctes et horizontales, situées dans le plan image commun des deux caméras. Les deux plans épipolaires associés contiennent par définition les centres optiques C_{L} et C_{R}, ils se

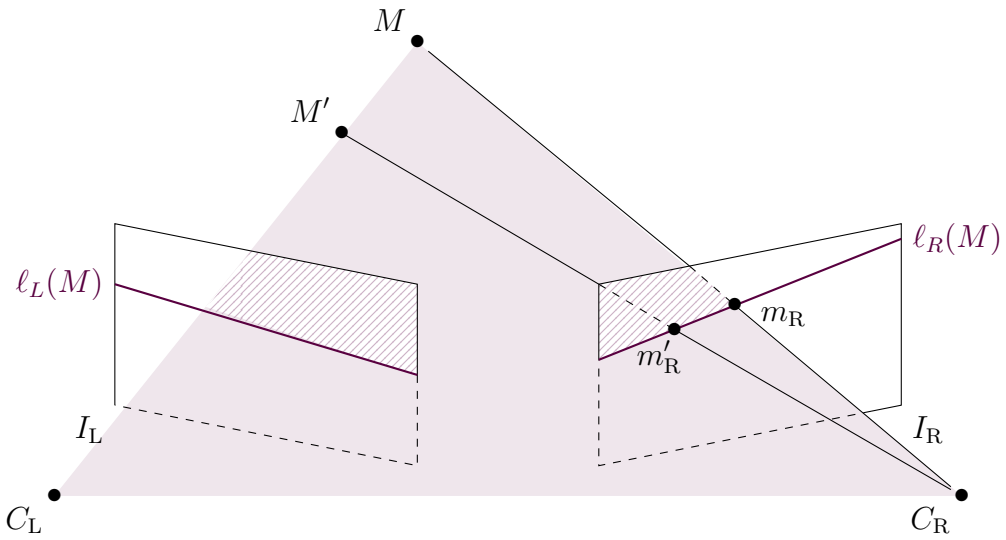


FIGURE 2.2 – Géométrie épipolaire. Le plan épipolaire, représenté ici par un triangle plein, est le plan passant par les trois points non alignés C_{L}, M et C_{R}. Il coupe chacun des deux plans images selon une droite, $\ell_L(M)$ et $\ell_R(M)$, appelées droites épipolaires. Tous les points de la scène appartenant à la droite (MC_{R}) (resp. (MC_{L})) ont pour projection un point de la droite épipolaire $\ell_L(M)$ (resp. $\ell_R(M)$).

coupent donc selon la droite $(C_{\mathrm{L}}C_{\mathrm{R}})$, appelée *baseline*. Or, les deux plans épipolaires sont parallèles par hypothèse aux droites épipolaires ℓ et ℓ', d'où l'on en conclut que c'est également le cas de leur intersection. La *baseline* est donc parallèle au plan image commun, ce qui implique que les deux systèmes optiques ont même distance focale. On en déduit également que la *baseline* est parallèle à l'axe horizontal commun du repère de chacune des images.

Lorsque les deux caméras sont dans cette configuration particulière, leurs paramètres intrinsèques (distance focale et coordonnées du point principal) sont identiques. On parle alors de *déplacement fronto-parallèle* de la caméra (cf. figure 2.3). En effet, si la scène est statique, on peut considérer qu'il s'agit de la même caméra que l'on a translatée selon la direction horizontale du repère associé à son image. Réciproquement, on montre que, lorsque les deux caméras ont mêmes paramètres intrinsèques et que le repère associé à l'image de droite est la translatée horizontale du repère associé à l'image de gauche, alors les droites épipolaires sont confondues dans les deux images et sont horizontales.

Rectification épipolaire Dans le cas général, il est possible de se ramener au cas où les droites épipolaires sont confondues d'une image à l'autre et horizontales, *via* une étape de *rectification épipolaire*. Cette opération consiste à déterminer deux homographies [30], qui permettent de transformer les deux images afin d'aligner les droites épipolaires. Cela revient à simuler deux nouvelles caméras et leur image respective. Les homographies sont estimées en mettant en correspondance des points SIFT [25] des deux images.

Il faut cependant noter que la rectification épipolaire est stable par translation horizontale et par translation verticale *simultanée* des deux images. En d'atures termes, l'abcisse des points principaux des caméras simulées est arbitraire, de même que leur ordonnée (commune). Ainsi, bien qu'elles aient même distance focale, on ne peut plus parler de déplacement fronto-parallèle, car les paramètres intrinsèques de deux caméras

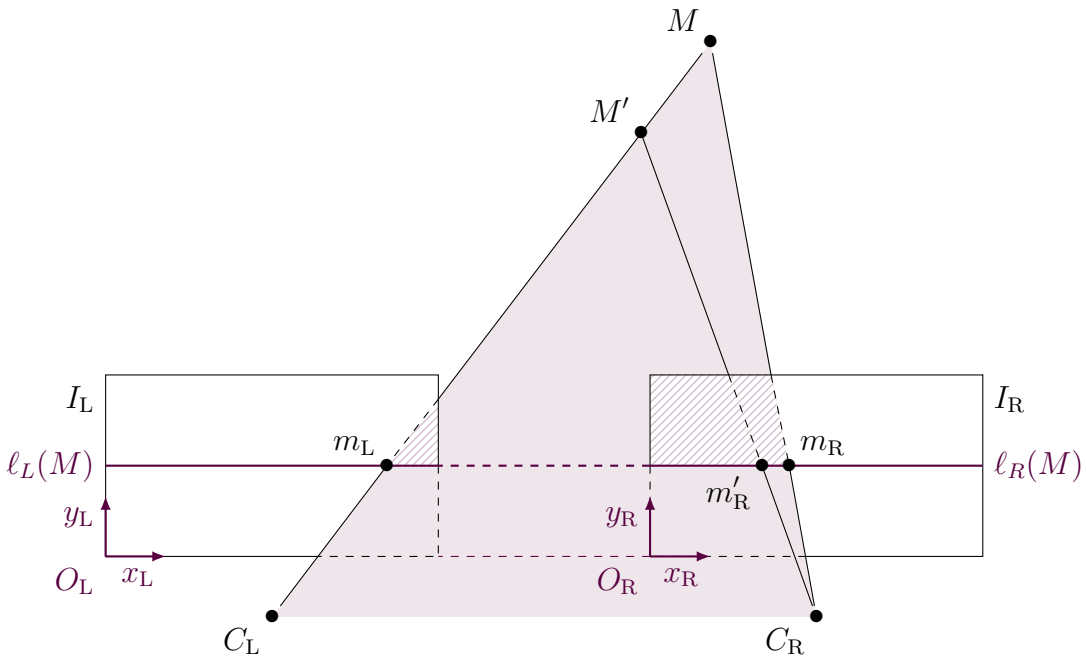


FIGURE 2.3 – Déplacement fronto-parallèle de la caméra. Les deux droites épipolaires $\ell_L(M)$ et $\ell_R(M)$ sont confondues, parallèles à l'axe horizontal (commun) des deux repères respectifs $(O_{\mathrm{L}}; \vec{x_{\mathrm{L}}}, \vec{y_{\mathrm{L}}})$ et $(O_{\mathrm{R}}; \vec{x_{\mathrm{R}}}, \vec{y_{\mathrm{R}}})$ des deux images. Tout pixel de l'image de droite situés sur la droite épipolaire $\ell_R(M)$ est la projection d'un point appartenant à la droite $(C_{\mathrm{L}}M)$, et réciproquement, la projection de tout point de la scène appartenant à la droite $(C_{\mathrm{L}}M)$ est située sur $\ell_R(M)$. Les paramètres intrinsèques des deux caméras étant identiques, on peut se ramener au cas d'une unique caméra, translatée du vecteur $\vec{C_{\mathrm{L}}C_{\mathrm{R}}}$ parallèle à l'axe $(O_{\mathrm{L}}; \vec{x_{\mathrm{L}}})$.

simulées sont potentiellement différents. Il est néanmoins facile de se ramener dans ce cas grâce à une translation (horizontale) du repère d'une des caméras simulées.

La contrainte de déplacement fronto-parallèle de la caméra est naturelle. Elle correspond d'une part à la configuration de la vision humaine (où la paire d'images est obtenue grâce à nos deux yeux). Un rendu en relief naturel tel que ceux proposés par l'industrie cinématographique suppose que les caméras sont en déplacement fronto-parallèle l'une par rapport à l'autre, avec un écartement équivalent à celui des yeux. D'autre part, ainsi qu'on va le voir dans le paragraphe suivant, cette configuration simplifie la reconstruction du relief. C'est pourquoi la plupart des algorithmes de stéréovision suppose que les images sont rectifiées au préalable. Nous en ferons de même dans tout ce qui suit.

2.1.3 Disparité et mise en correspondance

Le principe central de la vision stéréoscopique est la parallaxe, c'est-à-dire le mouvement apparent des objets lorsqu'ils sont vus depuis des points de vue différents. Explicitons ce phénomène. On rappelle que l'on se place désormais dans le cas d'un déplacement fronto-parallèle de la caméra (avec éventuellement une translation horizontale du point principal).

Quelques remarques préliminaires Les droites épipolaires sont confondues dans les deux images (on ne précisera donc plus l'image concernée) et sont horizontales. En particulier, les axes horizontaux des deux images sont confondus. Il s'ensuit que tout

(a) (b) (c) (d)

FIGURE 2.4 – Rectification épipolaire d'une paire stéréoscopique. (a) et (b) : Paire originale. (c) et (d) : Paire rectifiée. Les droites épipolaires sont maintenant horizontales et confondues, mais le point principal a été translaté horizontalement. (Code : [30])

point du plan image a même ordonnée dans chacun des deux repères des images. On ne précisera donc plus le repère lorsque l'on mentionnera l'ordonnée des points dans le plan image. Par ailleurs, puisque la projection (lorsqu'elle existe) d'un point physique est située sur la droite épipolaire de la caméra associée, on en déduit que tout point visible de la scène se projette sur la même ligne dans les deux images.

Soit M un point de la scène. On suppose qu'il est visible depuis les deux caméras. Notons $m_{\mathrm{L}}(x_{\mathrm{L}},y_{\mathrm{L}})$ sa projection sur le plan image de gauche, et $m_{\mathrm{R}}(x_{\mathrm{R}},y_{\mathrm{R}})$ sa projection sur le plan image de droite. On note e la droite épipolaire associée au point M. Les remarques d'introduction assurent que $y_{\mathrm{L}} = y_{\mathrm{R}}$, qui est également l'ordonnée de la droite épipolaire. Plaçons-nous à présent dans le plan épipolaire (cf. figure 2.5). Celui-ci coupe le plan image selon la droite e. Il contient par définition les deux centres optiques C_{L} et C_{R}, et en particulier la *baseline*. Cette dernière est parallèle à la droite épipolaire (car parallèle à la fois au plan épipolaire et au plan image d'après l'analyse menée dans le paragraphe précédent), et la distance entre la *baseline* et la droite épipolaire vaut exactement la distance focale f. Notons b la distance entre les deux centres optiques. L'intersection entre le plan image et le plan épipolaire est la droite $(m_{\mathrm{L}}m_{\mathrm{R}})$, dont l'intersection avec le domaine de chaque image est un segment. Notons o_{L} et o_{R} les extrémités gauche de ces deux segments. Il s'agit des pixels des coordonnées $(0,y_{\mathrm{L}})$ dans chacun des deux repères. Le vecteur $\overrightarrow{o_{\mathrm{L}}m_{\mathrm{L}}}$ est donc un vecteur horizontal, d'abscisse x_{L}, tandis que le vecteur (horizontal également) $\overrightarrow{o_{\mathrm{R}}m_{\mathrm{R}}}$ a pour abscisse x_{R}.

Disparité Désormais, les deux caméras n'auront plus un rôle symétrique. On choisit la caméra de gauche comme la caméra de *référence*. On définit alors la *disparité* du point M (ou, indifféremment, du pixel m_{L} dans l'image de référence), notée $u(M)$ (ou $u(m_{\mathrm{L}})$), comme le déplacement apparent de sa projection entre la vue de droite et la vue de gauche, lorsqu'il est visible depuis les deux caméras. Plus précisément, avec les notations introduites ici, on choisit la convention

$$u(M) = u(m_{\mathrm{L}}) = \begin{pmatrix} x_{\mathrm{L}} - x_{\mathrm{R}} \\ y_{\mathrm{L}} - y_{\mathrm{R}} \end{pmatrix} \in \mathbb{R}^2.$$

Les remarques qui précèdent assurent que la disparité est un vecteur horizontal :

$$u(M) = u(m_{\mathrm{L}}) = \begin{pmatrix} x_{\mathrm{L}} - x_{\mathrm{R}} \\ 0 \end{pmatrix} \in \mathbb{R}^2$$

et on confondra désormais le vecteur disparité et sa coordonnée horizontale.

Distance à la caméra Montrons que la disparité ne dépend que de la distance du point M à la *baseline*. On note h cette distance, et on l'appelle, par abus de langage,

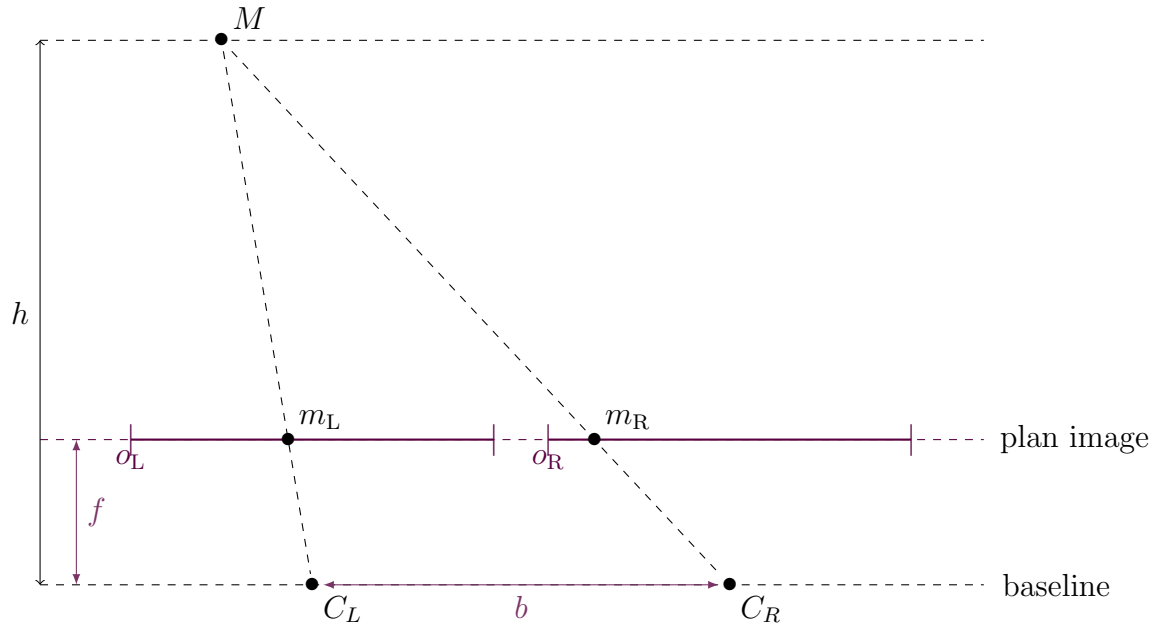


FIGURE 2.5 – Disparité et distance à la caméra. On se place ici dans le plan épipolaire. On note m_{L} et m_{R} les projections respectives du point M par chacune des deux caméras, de centres optiques C_{L} et C_{R}. L'intersection des domaines des images avec le plan épipolaire est réduit à deux segments, matérialisés ici par deux traits pleins, dont les extrémités gauche sont notés o_{L} et o_{R}. La distance du point M à la *baseline*, appelée par abus de langage *distance à la caméra*, est notée h. En utilisant le théorème de THALÈS, on peut exprimer la distance entre les deux images m_{L} et m_{R} en fonction de la distance entre les deux caméras b, la distance focale f et la distance à la caméra h. On en déduit alors la valeur de la *disparité* du point M, qui vaut par définition $|o_{\mathrm{L}}m_{\mathrm{L}}| - |o_{\mathrm{R}}m_{\mathrm{R}}|$.

distance du point M à la caméra. On commence par remarquer que la quantité $x_\mathrm{L} - x_\mathrm{R}$ vaut $|o_\mathrm{L}m_\mathrm{L}| - |o_\mathrm{R}m_\mathrm{R}|$. Le point m_L étant toujours situé entre les points o_L et m_R, on en déduit que

$$|o_\mathrm{L}m_\mathrm{L}| = |o_\mathrm{L}m_\mathrm{R}| - |m_\mathrm{L}m_\mathrm{R}|;$$

de manière similaire, le point o_R est toujours situé entre les deux points o_L et m_R, donc

$$|o_\mathrm{L}m_\mathrm{R}| = |o_\mathrm{L}o_\mathrm{R}| - |o_\mathrm{R}m_\mathrm{R}|.$$

Par conséquent, la disparité du pixel m_L vaut

$$u(m_\mathrm{L}) = |o_\mathrm{L}o_\mathrm{R}| - |m_\mathrm{L}m_\mathrm{R}|.$$

Si les deux caméras sont en déplacement fronto-parallèle, la position du centre optique par rapport au cadre est la même dans les deux caméras (car la distance focale et la position du point principal dans le cadre sont les mêmes). On en déduit que les droites $(C_\mathrm{L}o_\mathrm{L})$ et $(C_\mathrm{R}o_\mathrm{R})$ sont parallèles, ce qui entraîne que la distance $|o_\mathrm{L}o_\mathrm{R}|$ vaut exactement la distance entre les deux centres optiques $|C_\mathrm{L}C_\mathrm{R}| = b$. Si les caméras n'ont pas le même point principal (après une rectification épipolaire par exemple), la distance $|o_\mathrm{L}o_\mathrm{R}|$ vaut une valeur positive arbitraire que l'on peut calculer si on connaît les paramètres intrinsèques des deux caméras. Elle ne dépend cependant pas du point M et est donc considérée comme constante. Calculons à présent la longueur $|m_\mathrm{L}m_\mathrm{R}|$. En appliquant le théorème de Thalès dans les triangles $C_\mathrm{L}MC_\mathrm{R}$ et $o_\mathrm{L}Mo_\mathrm{R}$, on obtient la relation de proportionnalité suivante :

$$\frac{|m_\mathrm{L}m_\mathrm{R}|}{b} = \frac{h-f}{h}$$

d'où l'on déduit l'expression suivante de la disparité :

$$u(m_\mathrm{L}) = |o_\mathrm{L}o_\mathrm{R}| - b + \frac{f\,b}{h}. \tag{2.1}$$

Autrement dit, si les deux caméras sont en déplacement fronto-parallèle, alors la disparité du point M est inversement proportionnelle à sa distance à la caméra ; si ce n'est pas le cas, il y a un terme (constant) $|o_\mathrm{L}o_\mathrm{R}|-b$ qui s'ajoute. Ainsi, dans tous les cas, si on connaît la disparité d'un point M, on peut retrouver grâce aux paramètres intrinsèques des deux caméras sa distance à la caméra en inversant la formule précédente :

$$h = \frac{f\,b}{u(m_\mathrm{L}) + b - |o_\mathrm{L}o_\mathrm{R}|}.$$

En d'autres termes, si on parvient à calculer la disparité de tous les points de la scène, on en connaît précisément le relief.

Mise en correspondance Déterminer le relief d'une scène donnée par deux caméras repose donc sur deux problèmes complémentaires :

1. calculer ou mesurer les paramètres intrinsèques des caméras réelles et/ou virtuelles, dans le cas d'une rectification épipolaire ;
2. calculer la carte de disparité de la vue de référence dans la paire d'images rectifiée.

Ce dernier problème revient à trouver, pour chaque point M de la scène visible depuis les deux caméras, ses deux images m_L et m_R. De manière équivalente, cela revient à déterminer, pour chaque pixel m_L de l'image de référence, le pixel m_R, appelé *pixel homologue* du pixel m_L, tel que les deux pixels m_L et m_R soient images du même point physique. Ce processus est appelé *mise en correspondance*.

Les droites épipolaires étant confondues dans les deux images, le pixel homologue, s'il existe, d'un pixel de l'image de gauche est situé sur la même ligne que celui-ci. La recherche d'un pixel homologue se fait donc sur une seule ligne de l'image de droite. Elle peut même être restreinte à un intervalle appelé *intervalle de disparité* si des mesures préalables ont permis d'estimer la disparité minimale et maximale. La formule (2.1) assure en effet que la disparité est une fonction décroissante de la distance à la caméra. Elle est en particulier minimale pour les objets à l'infini, et vaut alors $|o_{\mathrm{L}}o_{\mathrm{R}}| - b$. Si l'objet le plus proche de la caméra est situé à une distance h_0, alors la disparité est par ailleurs majorée par la quantité $|o_{\mathrm{L}}o_{\mathrm{R}}| - b + f\,b/h_0$. Ainsi, si on est capable d'estimer la distance (ou la disparité) de l'objet le plus proche de la caméra, il est possible d'obtenir un intervalle de disparité $\mathrm{I}_{\mathrm{disp}} = [\,u_{\min}\,;u_{\max}\,]$. Le pixel homologue de tout pixel $m_{\mathrm{L}}(x_{\mathrm{L}},y_{\mathrm{L}})$ de l'image de gauche est alors à rechercher parmi les pixels de l'image de droite de coordonnées (x,y_{L}), avec $x \in x_{\mathrm{L}} - [\,u_{\min}\,;u_{\max}\,]$.

2.1.4 Difficultés à surmonter

Le problème de mise en correspondance est intrinsèquement difficile à résoudre, car il s'agit d'associer deux pixels issus du même point physique sans autre information que leurs intensités respectives (il n'y a pas de modèle de la scène). À cela, il faut ajouter des difficultés supplémentaires qui rendent la tâche encore plus ardue.

Mouvement des objets La scène doit être supposée statique pour que le lien entre disparité et distance à la caméra établi au paragraphe précédent soit vrai. Or, si les images ne sont pas prises simultanément (par exemple, lorsqu'il s'agit d'un satellite qui prend une image par passage au-dessus d'un certain point du sol), il y a de fortes chances pour des objets aient bougé entre-temps (avec de plus un mouvement apparent qui ne soit pas dans la direction épipolaire). Même si la mise en correspondance est correctement réalisée, la disparité des objets qui ont bougé n'est plus inversement proportionnelle à leur distance à la caméra.

Changement d'illumination ou de contraste Si les images sont prises simultanément, cela suppose qu'elles sont prises par deux caméras différentes. L'étape de rectification permet de simuler des caméras aux mêmes paramètres intrinsèques (à une translation horizontale du point principal près), mais ne peut pas corriger les différences de qualité ou de dynamique entre les deux images. Par exemple, certaines caméras adaptent automatiquement le contraste ou la balance des blancs. Cette opération ne peut être réalisée exactement de la même manière sur les deux caméras (même s'il s'agit du même modèle). Dans ce cas, les deux images peuvent présenter des aspects très différents, ce qui rend la mise en correspondance plus difficile.

C'est également le cas lorsque les images n'ont pas été prises au même moment : le soleil peut avoir changé de position dans le ciel, celui-ci peut s'être couvert. La scène, même en restant immobile, change alors visuellement d'aspect.

Reflets Certaines surfaces comme les vitres ou des métaux brillants renvoient partiellement la lumière qu'elles reçoivent. Cela a deux conséquences sur la mise en correspondance stéréoscopique. Tout d'abord, une surface complètement réfléchissante donne l'illusion d'une apparente profondeur. C'est ce que l'on observe par exemple en plaçant un miroir sur un mur : le regard porte au-delà du mur et on a l'illusion que la scène se prolonge *dans le mur*. Dans ce cas également, cela implique que la disparité n'est plus

une information pertinente sur la distance à la caméra : le miroir semblera plus éloigné qu'il ne l'est réellement. Ensuite, si la surface n'est pas plate, elle renvoie la lumière différemment selon la direction sous laquelle on l'observe. En d'autres termes, les reflets ne sont pas situés au même endroit suivant l'image. Or, sans information permettant d'interprêter le reflet comme tel, la mise en correspondance dicte d'associer les deux reflets, alors qu'ils ne correspondent pas au même point physique.

Végétation La végétation et les objets fins peuvent également présenter des aspects très différents suivant le point d'observation. Ils sont en effet composés de surfaces (parfois réfléchissantes) de très petites tailles (comme le feuillage), qui sont orientés dans de nombreuses directions. Ce sont donc généralement des zones difficiles à mettre en correspondance, car il est difficile d'identifier le pixel homologue qui change beaucoup d'apparence.

Régions plates et effet de Strobes Dans le cas des régions peu ou pas texturées (d'une couleur unie par exemple), le problème est inverse : il est difficile de sélectionner le pixel homologue car, visuellement, il y a beaucoup de candidats possibles (on parle de problème d'ouverture). Le cerveau rencontre parfois ce problème : lorsqu'on regarde de près un mur blanc, lisse (mais mat, donc sans reflets), il arrive que l'on se mette à loucher et à éprouver un léger vertige. Le cerveau ne parvient pas à mettre correctement en correspondance les deux images qu'il possède du mur. Il hésite entre plusieurs solutions, et c'est cette hésitation qui donne au mur un mouvement apparent (il semble avancer et reculer) qui donne le tournis. Il suffit alors de remarquer une petite aspérité dans le mur pour que le regard accroche et que le malaise cesse.

Ce phénomène se produit également lorsqu'une région présente une répétition de motifs identiques (comme par exemple des rayures). On parle alors d'effet de Strobes.

Bruit Enfin, il faut signaler que toute image numérique présente du bruit, ce qui signifie que l'intensité capturée n'est pas exactement celle du point physique. Il existe de nombreux types de bruit possibles, parmi lesquels on peut citer le bruit thermique (dû à l'agitation naturelles des électrons dans les capteurs), le bruit électronique (lorsque le nombre de photons est trop faible), le bruit de lecture (qui se produit pendant la conversion numérique du signal acquis), le bruit de quantification (dû à la discrétisation des valeurs du signal). Le bruit total est aléatoire, donc les deux images du même point ne sont pas affectées de la même manière.

On voit que les difficultés rencontrées peuvent être classées suivant trois catégories :

- la mise en correspondance est possible, mais la disparité ne donne aucune information significative sur la distance de l'objet à la caméra (mouvement dans la scène, reflets) ;
- la mise en correspondance n'est pas possible car les pixels homologues sont visuellement trop dissemblables (changement d'illumination ou de contraste, végétation, bruit important) ;
- la mise en correspondance n'est pas possible car il y a trop de pixels candidats et aucun moyen de les départager (régions plates, motifs répétés).

Dans ce qui suit, on supposera que les images sont prises simultanément et par la même caméra, ce qui revient à supposer que les objets n'ont pas bougé et que le contraste et l'illumination de la scène restent les mêmes. Les reflets ne seront pas spécifiquement

gérés, ni l'effet de STROBES, mais on verra que ce sont des difficultés atténuées par les approches globales. Malgré ces hypothèses simplificatrices, le prochain paragraphe montre que le problème est en réalité mal posé.

2.2 Le phénomène d'occultation

2.2.1 Occultation, désoccultation

Déterminer la disparité d'un point suppose que ce point est visible par les deux caméras. Or, dès que la scène possède un relief, certains points créent des obstacles entre d'autres points et au moins une des caméras. Il s'agit du phénomène d'*occultation*[6]. Plus précisément, on peut classer les points de la scène en quatre catégories :

- les points visibles depuis les deux caméras ;
- les points invisibles depuis les deux caméras ;
- les points uniquement visibles depuis la caméra de référence ;
- les points uniquement visibles depuis l'autre caméra.

Les points visibles depuis la vue de référence mais invisibles depuis l'autre vue (et, par extension, leur image dans la vue de référence) sont qualifiés d'*occultés*. Ceux qui, à l'inverse, ne sont visibles que depuis l'autre vue sont dits *désoccultés*. Les objets à l'origine de l'occultation seront appelés *occultants*.

Puisque la carte de disparité n'est calculée que sur l'image de référence, seuls les points occultés seront considérés. Ces points n'ont par définition aucun pixel homologue dans l'autre vue, ce qui implique que leur disparité n'est pas définie. Le problème de mise en correspondance est donc mal posé.

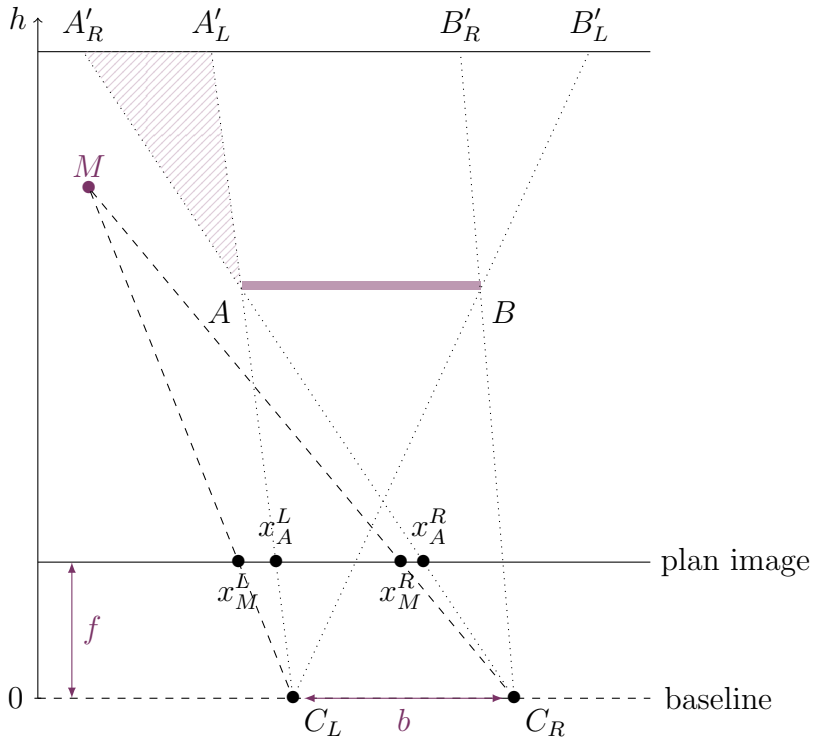
2.2.2 Préservation de l'ordre et largeur des objets

Largeur de l'objet Pour simplifier notre analyse, nous allons partir du cas simple d'un objet d'épaisseur nulle, par exemple un rectangle parallèle au plan image. Ce choix est motivé par le fait que la plupart des méthodes de stéréovision supposent que les objets de la scène ont une disparité constante, du moins localement. On supposera par ailleurs que l'objet est entièrement visible depuis les deux vues. On se place désormais dans un plan épipolaire coupant l'objet étudié, ce qui nous permet de nous ramener à des représentations planes.

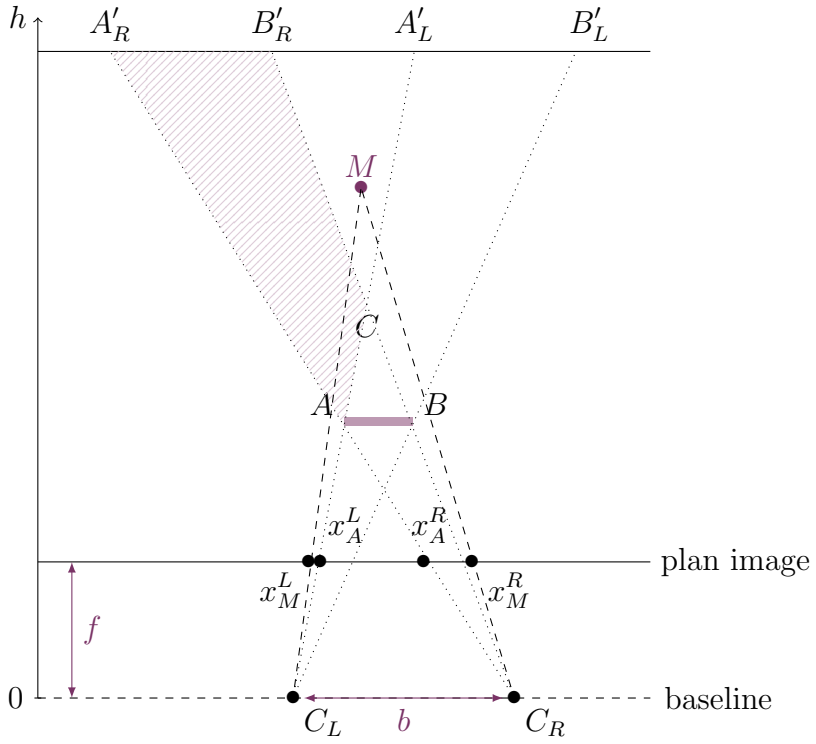
On définit à présent un objet *large* comme étant un objet dont la largeur (c'est-à-dire la taille selon l'axe horizontal) est supérieure ou égale à la distance b entre les centres optiques de deux caméras. Un objet *fin* est alors de largeur inférieure strictement à b.

Préservation de l'ordre Considérons la figure 2.6. On note $[AB]$ l'objet étudié. Plaçons-nous dans le cas où $|AB|$ est supérieur à b (figure 2.6(a)). Intéressons-nous tout d'abord aux points visibles par chacune des deux caméras. La région délimitée par les demi-droites $[AA'_L)$ et $[BB'_L)$ et le segment $[AB]$ est invisible depuis la vue de gauche, tandis que celle délimitée par les demi-droites $[AA'_R)$ et $[BB'_R)$ et le segment $[AB]$ est invisible depuis la vue de droite. Par conséquent, la région occultée (hachurée) est délimitée par les demi-droites $[AA'_L)$ et $[AA'_R)$. Considérons à présent un point non occulté M dans le voisinage de $[AB]$, d'images respectives x_M^L et x_M^R par les caméras

6. En anglais, l'occultation est nommée *occlusion*.



(a)



(b)

FIGURE 2.6 – Préservation de l'ordre. (a) Lorsque l'objet est large, le point M est vu à gauche du point A dans les deux vues. (b) Ce n'est plus le cas lorsque l'objet est fin : si M est situé dans un secteur très spécifique de la scène, alors il est vu à gauche de A dans la vue de gauche mais à droite de A dans la vue de droite.

de gauche et de droite. Si, dans la vue de référence, le point M est placé à gauche de l'objet $[AB]$, alors, nécessairement, il est situé dans la zone située à gauche du segment $[C_{\mathrm{L}}A]$ et de la demi-droite $[AA'_R)$. Or, M possède une image par la caméra de droite également située à gauche de $[AB]$, car la demi-droite $[C_{\mathrm{R}}M)$ est également située à gauche de $[C_{\mathrm{R}}A)$. On dit que l'*ordre est préservé* dans les deux images [7].

Lorsque l'objet est fin, comme c'est le cas dans la figure 2.6(b), il existe une zone située derrière l'objet qui est non occultée. Elle est délimitée par les deux demi-droites $[CA'_L)$ et $[CB'_R)$, où C est l'intersection des droites $(C_{\mathrm{L}}A)$ et $(C_{\mathrm{R}}B)$. Considérons M un point de cette zone. La demi-droite $[C_{\mathrm{L}}M)$ est située à gauche de $[C_{\mathrm{L}}A)$, donc depuis la vue de gauche, le point M est vu *à gauche* de l'objet $[AB]$. En revanche, la demi-droite $[C_{\mathrm{R}}M)$ est située à droite de $[C_{\mathrm{R}}A)$, donc, dans la vue de droite, le point M est vu *à droite* de l'objet $[AB]$: l'ordre est *inversé*. Cette configuration persiste tant que la zone délimitée par $[CA'_L)$ et $[CB'_R)$ existe, ce qui impose que $|AB|$ est strictement inférieur à b.

On voit donc que la largeur de l'objet joue sur la préservation de l'ordre des points dans les deux images. Il est à noter que, si les objets larges ne permettent pas une inversion de cet ordre, la présence d'objets fins ne garantit pas que cet ordre sera nécessairement inversé. Il faut pour cela qu'un point de la scène (donc visible depuis les deux vues) se situe dans le triangle $A'_L C B'_R$, ce qui n'est plus le cas si un autre objet occulte cette région.

Sauf mention contraire, on se placera désormais dans le cas où l'ordre est préservé dans les deux images.

2.2.3 Occultation et contrainte de visibilité

Comme on le verra dans la section 2.3, le phénomène d'occultation est généralement ignoré dans les méthodes de mise en correspondance stéréoscopique. Dans l'approche que nous proposons au chapitre 3, nous avons choisi de l'intégrer au modèle considéré. Nous développons donc ici une analyse préliminaire très fine de l'occultation. Commençons par étudier les conditions nécessaires à la présence d'occultation.

Si un point M est visible depuis les deux vues, alors seul le voisin de gauche de son image x^L_M dans l'image de référence peut être occulté, car l'occultation se produit toujours sur le bord gauche des objets dans la vue de gauche. Soit M' le point dont l'image $x^L_{M'}$ est un voisin à gauche de x^L_M (sur la même ligne). Démontrons que, s'il n'est pas occulté, alors sa disparité doit vérifier une certaine contrainte, appelée *contrainte de visibilité* [8].

Puisque M est visible depuis les deux scènes, on a $d(M) = x^L_M - x^R_M$ sa disparité. Par ailleurs, sur la droite $(C_{\mathrm{R}}M)$, seuls les points situés sur le segment $[Mx^R_M]$ sont visibles par la caméra de droite. Supposons que M' est également visible depuis les deux vues. Puisque son image $x^L_{M'}$ est située à gauche de celle de M dans la vue de gauche, on en déduit que le segment $[C_{\mathrm{L}}M']$ se situe à gauche de la droite $(C_{\mathrm{L}}M)$. Puisque l'ordre est supposé préservé, on en déduit que l'image $x^R_{M'}$ du point M' se situe également à gauche de l'image x^R_M du point M dans la vue de droite. Par conséquent, le segment $[C_{\mathrm{R}}M']$ est situé à gauche de la droite $(C_{\mathrm{R}}M)$. Cette première étude permet de situer le point M' dans le secteur délimité à droite par les droites $(C_{\mathrm{L}}M)$ et $(C_{\mathrm{R}}M)$.

7. La préservation de l'ordre est appelée *contrainte de monotonie* dans [17].
8. On utilise ici la terminologie de [31], même si la contrainte se traduit dans des termes différents.

Posons $\varepsilon = x_M^L - x_{M'}^L$, dont la composante horizontale est positive. Pour que M' reste visible depuis la vue de droite, sa distance à la caméra, notée h' est majorée par celle de l'intersection des droites $(C_{\mathrm{R}}M)$ et $(C_{\mathrm{L}}M')$, que l'on note C. On en déduit que la disparité du point M' est minorée par celle du point C, s'il était visible. Calculons-la. Puisque C se projette dans l'image de droite sur le pixel x_M^R, et dans l'image de gauche sur le pixel $x_{M'}^L$, on en déduit que

$$d(C) = x_{M'}^L - x_M^R = x_{M'}^L - x_M^L + x_M^L - x_M^R = d(M) - \varepsilon.$$

Puisque $d(M) = d(x_M^L)$ et que $d(M') = d(x_{M'}^L) = d(x_M^L - \varepsilon)$, on en déduit que

$$d(x_M^L) - d(x_M^L - \varepsilon) \leq \varepsilon.$$

En faisant tendre ε vers 0, on en déduit que, pour ne pas créer de l'occultation, les variations horizontales de la disparité d ne doivent pas atteindre ou excéder 1.

2.2.4 Analyse de l'occultation : lien avec la disparité

Nous allons à présent établir le lien entre la largeur d'une occultation et le saut de disparité correspondant. Une étude similaire a été proposée dans [17].

On suppose dans cette analyse que l'objet, appelé objet *occultant*, occulte partiellement un objet situé derrière lui, désigné sous le nom d'objet *occulté*. L'objet occulté sera supposé de distance à la caméra constante, notée h', donc parallèle au plan image. Sa disparité est donc calculable, même dans la région occultée. Il sera modélisé par un plan. Puisque l'occultation se produit sur le bord gauche des objets, appelons A le bord gauche de l'objet occultant, visible depuis les deux vues. Son image par chacune des caméras est notée respectivement x_A^L et x_A^R.

Calculons la longueur maximale que peut atteindre l'occultation dans la vue de gauche. Considérons pour cela la figure 2.7, dans laquelle O_{L} et O_{R} désignent les projetés des centres optiques sur l'intersection entre le plan image et le plan épipolaire. Calculer la largeur de l'occultation dans la carte de disparité revient à calculer la longueur du segment $[x_A^L x_{A'_L}^L]$. Remarquons que, par construction, les points A et A'_R ont même image dans l'image de droite, que l'on note x_A^R. En particulier, on en déduit que la longueur $|x_A^L x_{A'_L}^L|$ peut s'exprimer en fonction du point O_L, car O_L n'appartient pas au segment $[x_A^L x_{A'_L}^L]$,

$$|x_A^L x_{A'_L}^L| = \Big| |O_L x_A^L| - |O_L x_{A'_L}^L| \Big|$$

où on peut faire apparaître la longueur $|O_R x_A^R|$:

$$|x_A^L x_{A'_L}^L| = \Big| (|O_R x_A^R| - |O_L x_{A'_L}^L|) - (|O_R x_A^R| - |O_L x_A^L|) \Big|.$$

On reconnaît alors les disparités respectives des pixels x_A^L et $x_{A'_R}^L$, ce qui assure finalement que

$$|x_A^L x_{A'_L}^L| = |d(x_{A'_L}^L) - d(x_A^L)|.$$

Autrement dit, la largeur de l'occultation vaut la différence entre la disparité de l'objet occulté et la disparité de l'objet occultant, que l'on désignera désormais sous le nom de *saut de disparité* autour de la région occultée. Ce saut est positif, car l'objet occulté se situe derrière l'objet occultant. Par ailleurs, on a montré que, dans la vue de référence, l'occultation était positionnée immédiatement à gauche de l'image de l'objet occultant.

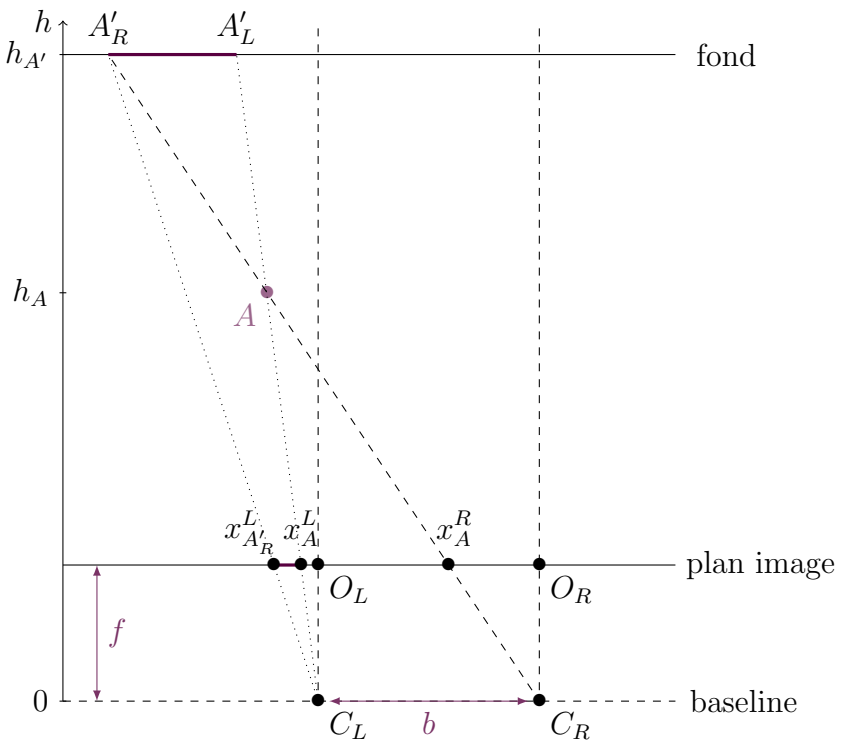


FIGURE 2.7 – Largeur de l'occlusion. La région du fond occultée par le point A est matérialisée par le segment $[A'_R A'_L]$. La largeur de l'occlusion dans la fonction de disparité est alors donnée par la longueur du segment $[x^L_A x^L_{A'_R}]$, où x^L_A (resp. $x^L_{A'_R}$) désigne l'image du point A (resp. A'_R) dans l'image de gauche.

2.3 L'état de l'art

Pour une revue plus complète des méthodes de stéréovision, on pourra se reporter à [36, 41]. Les mesures de dissimilarité sont évaluées plus spécifiquement dans [21] par exemple.

2.3.1 Mesurer la similarité de deux pixels

Deux pixels homologues étant les images d'un même point physique, la mise en correspondance stéréoscopique repose essentiellement sur des critères de similarité entre deux pixels. L'idée sous-jacente est que, sous l'hypothèse d'une absence de changement d'illumination ou de contraste, plus deux pixels sont visuellement ressemblants, plus il y a de chance pour qu'ils soient issus du même point.

Cette approche nécessite donc de définir des *mesures de dissimilarité*[9], qui quantifient la ressemblance visuelle de deux pixels. Une mesure de dissimilarité est une fonction D à valeurs positives. Si on note I_{L} l'image de référence et I_{R} l'image de droite, alors, pour tout pixel p dans l'image de gauche et tout pixel q dans l'image de droite, la quantité $D_{I_{\mathrm{L}},I_{\mathrm{R}}}(p,q)$, appelée *coût de corrélation*, est d'autant plus faible que les deux pixels p et q sont *semblables*.

Corrélation d'intensité Les mesures les plus faciles à définir sont les mesures de corrélation de niveaux de gris, car il s'agit de comparer les deux valeurs réelles que sont

9. Certains auteurs parlent de *mesure de corrélation*.

les intensités respectives de p dans I_{L} et de q dans I_{R}. Celles-ci sont notées respectivement $I_{\mathrm{L}}(p)$ et $I_{\mathrm{R}}(q)$. Toute distance sur $\mathbb{R}$ appliquée au couple $\big(I_{\mathrm{L}}(p),I_{\mathrm{R}}(q)\big)$ peut être utilisée. Les plus classiques [36] sont la distance qui découle de la norme euclidienne (aussi appelée mesure AD pour *absolute difference*) et le carré de celle-ci (appelée SD pour *squared difference*). Elles sont en effet simples à implémenter et peu coûteuses en calculs [18], tout en produisant des résultats raisonnables tant qu'il n'y a pas de changement d'illumination.

En réalité, les images numériques sont échantillonnées, ce qui signifie qu'un pixel n'est jamais l'image d'un point unique de la scène, mais plutôt une moyenne des images d'une petite région de l'espace. Ainsi, un point physique peut se projeter dans deux pixels qui, du fait de ce moyennage, apparaissent légèrement dissemblables, même en l'absence de bruit ou tout autre transformation chromatique. Pour réduire ce biais, Birchfield et Tomasi [3] propose d'utiliser une interpolation horizontale de l'image. Une variante exploitant l'information verticale est proposée dans [24]. L'interpolation choisie est l'interpolation bilinéaire, ce qui permet de rendre ce procédé peu coûteux en calculs.

Pour exploiter l'information supplémentaire contenue dans la couleur, une possibilité est de généraliser les mesures de corrélation de niveaux de gris à la couleur. Pour ce faire, [8] propose de combiner les coûts de corrélation obtenus pour chacun des canaux couleur, grâce à une fonction de *fusion*. Parmi les choix les plus classiques pour cette fonction, on peut citer la moyenne arithmétique, la valeur médiane, le minimum, le maximum, ainsi qu'une fonction proposée dans [2].

Exploitation des variations locales d'intensité La corrélation d'intensité est très sensible au bruit, ainsi qu'aux changements d'illumination et de contraste. Pour ajouter davantage d'informations, on peut exploiter les variations locales d'intensité.

Une première façon de procéder est d'appliquer sur les images une transformation, avant de les comparer avec une mesure de dissimilarité comme celles présentées au paragraphe précédent, si les images résultantes sont mono-valuées, ou avec des mesures plus adaptées dans les autres cas. Une transformation naturelle consiste à calculer le gradient des images[10]. Dans [35], une mesure de dissimilarité est ensuite définie pour comparer les vecteurs gradients en p et q. Cette mesure tient compte de la fiabilité de la comparaison, qui augmente avec la longueur des vecteurs comparés. Maar et Hildreth [26] proposent quant à eux d'appliquer le LoG (*laplacian of gaussian*) sur les images. Les mises en correspondance les plus précises concernent en effet les points où le gradient est le plus grand, d'où l'idée de localiser les maxima de la norme du gradient. Cela revient à localiser les zéros de la dérivée seconde. Comme celle-ci est sensible au bruit (hautes fréquences), un filtre passe-bas (convolution avec une gaussienne) est appliqué sur les images avant de calculer le laplacien. Zabih et Woodfill [45] définissent quant à eux deux transformées dites non paramétriques basées sur le *rang* du pixel. Dans une fenêtre autour de p (resp. de q), on identifie les pixels dont l'intensité est plus faible que celle de p (resp. de q). Puis, on peut soit les compter (*rank filter*), ce

10. L'œil humain interprète en réalité les couleurs à partir de leurs variations relatives plus qu'il ne les perçoit de manière absolue. Ainsi, la perception d'une couleur dépend fortement de son environnement. Cette observation a été énoncée dès 1839 par le chimiste Michel-Eugène Chevreul et est connue sous le nom de *loi du contraste simultané des couleurs*. Plus récemment, en 2015, une photographie de robe a fait le tour du monde et des réseaux sociaux car certaines personnes la voyaient blanche et dorée, d'autres la voyaient au contraire bleue et noire, tandis qu'une minorité de personnes pouvaient la voir des deux manières. Il s'agissait pourtant de la même image, mais dont les couleurs ont certainement été interprétées de manière différente par le cerveau.

qui revient à ordonner les pixels de la fenêtre suivant leur intensité, puis à déterminer le rang de p ; soit conserver la localisation de ces pixels, en la codant sous la forme d'un vecteur booléen (*census rank*). Dans le premier cas, on utilise sur les images transformées une mesure de dissimilarité pour image mono-valuée, dans le second cas, les vecteurs obtenus sont comparés grâce à la distance de Hamming (qui compte de nombre de coefficients différents). Ces deux transformations sont insensibles à tout changement de contraste.

Une seconde approche consiste à définir directement des mesures de dissimilarité comparant les variations d'intensité. Parmi ces approches, une méthode très populaire est le calcul de la NCC (*normalized cross correlation*) entre deux fenêtres, bien qu'elle soit plus coûteuse en calculs [18]. On considère deux fenêtres carrées autour des pixels p et q, que l'on normalise. Puis on en calcule le produit scalaire : plus celui-ci est grand, plus les deux voisinages sont semblables. Cette méthode gère les changements de contraste affine entre les deux images. On peut également mentionner l'utilisation de l'information mutuelle (MI) [43, 20] pour définir une mesure de corrélation. On considère que les deux images p et q (prises avec leur voisinage) sont des réalisations d'une certaine distribution. Dans ce cas, plus ces réalisations sont indépendantes, plus l'information mutuelle est faible.

Ces mesures, définies sur les images en niveaux de gris, sont plus robustes aux changements d'illumination et de contraste. Dans [4], les auteurs montrent qu'elles sont plus efficaces que les mesures de corrélation de couleur, et conseillent donc de les préférer à ces dernières.

Combinaison de l'intensité et des variations L'utilisation des variations est performante dans les régions texturées, mais dans les zones plates, en présence de bruit, elles peuvent se révéler désastreuses. L'idée est donc de combiner une corrélation d'intensité avec une corrélation basée sur les variations locales.

Une premier choix consiste à combiner la mesure AD et une mesure basée sur le gradient. Le gradient peut être utilisé dans son intégralité [23] ou seule sa composante horizontale est prise en compte [33]. La mesure résultante est une combinaison convexe (avec éventuellement un seuillage préalable des valeurs) des deux coûts de corrélation considérés.

Une autre méthode très plébiscitée est l'AD Census [28], qui combine la mesure AD et la corrélation obtenue après le filtrage Census. Le coût de dissimilarité est défini comme la somme des deux coûts initiaux, auxquels on a appliqué au préalable une certaine fonction croissante (avec des paramètres différents pour les deux coûts).

Ces méthodes ont montrées leur efficacité sur le banc d'essai Middlebury (voir paragraphe 2.3.4), d'où leur popularité. Elles donnent effectivement des résultats satisfaisants, tout en étant relativement peu coûteuses en calculs. Néanmoins, elles reposent sur un nombre de paramètres important (en général 3), qui sont difficiles à régler correctement.

2.3.2 Approches locales *versus* approches globales

Selon Scharstein et Szeliski [36], les méthodes de stéréovision peuvent être classées en deux grandes familles : les méthodes locales, qui mettent en correspondance les pixels à l'aide d'informations purement locales, et les méthodes globales, qui résolvent un problème d'optimisation global.

Méthodes locales : comparaison de fenêtres Le principe central est le suivant : pour tout pixel p de l'image de référence donné, le pixel homologue est le pixel q le plus ressemblant (dans l'intervalle de disparité) de l'image de droite. Cette ressemblance étant mesurée par la mesure de dissimilarité choisie, le pixel q devrait être le pixel minimisant le coût de corrélation $D_{I_\mathrm{L},I_\mathrm{R}}(p,q)$. Or, la corrélation pixellique (qui ne compare un pixel qu'avec un autre pixel) n'est pas suffisamment fiable (surtout en présence de bruit ou d'effet de Strobes par exemple). Cette remarque est déjà l'introduction des comparaisons basées sur les variations locales d'intensité.

En partant de l'hypothèse que la disparité est localement constante (ce qui est en réalité faux), on peut renforcer la fiabilité de ce critère en prenant également en compte le coût de corrélation $D_{I_\mathrm{L},I_\mathrm{R}}(p',q')$ des voisins p' (resp. q') du pixel p (resp. q), où $q-p$ et $q'-p'$ sont égaux (ce qui revient à tester la même valeur de disparité sur tout le voisinage). Ce procédé est appelé *agrégation des coûts*. L'idée sous-jacente est de comparer le voisinage de p avec le voisinage de q : on parle de *block-matching* (le *block* désignant le voisinage).

La manière la plus simple pour agréger les coûts de corrélation est de les moyenner dans un voisinage du pixel considéré. Si la mesure de dissimilarité choisie est AD et que la moyenne est une moyenne arithmétique, alors il s'agit de l'agrégation SAD (pour *sum of absolute differences*) et pour SD, on parle de SSD (pour *sum of squared differences*). En effet, si le voisinage considéré est le même pour tous les pixels, alors la moyenne arithmétique est équivalente à une somme.

Le choix de ce voisinage est crucial : les deux paramètres possibles en sont la taille et la forme (et éventuellement la position relative par rapport au pixel considéré).

La taille du voisinage est délicat. S'il est trop petit, alors il y a trop peu d'informations à exploiter et la corrélation reste trop incertaine. S'il est trop grand, alors l'hypothèse de disparité constante dans le voisinage devient fausse. Dans ce dernier cas, apparaît un phénomène dit d'*adhérence*. Au voisinage d'une discontinuité, entraînant nécessairement occultation ou désoccultation, c'est l'objet occultant qui va imposer sa disparité. Dans le cas d'images aériennes par exemple, cela se traduit par un épaississement des immeubles (c'est pourquoi on parle également de *fattening effect*). Pour une taille et une forme de voisinage données, une manière d'éviter l'adhérence est de décentrer le voisinage d'agrégation pour éviter d'y inclure une discontinuité. C'est ce que fait le *MinFilter* [16]. Une autre manière d'éviter cet écueil est d'utiliser des voisinages de tailles variables [1].

Les formes de voisinage les plus simples sont les fenêtres carrées, car faciles à implémenter. Néanmoins, des variantes ont été proposées depuis deux décennies. L'idée est que le meilleur voisinage est le voisinage le plus grand dans lequel la disparité reste constante. Le modèle de scène classiquement retenu étant que les objets sont en réalité des surfaces planes parallèles au plan image, il suffit de considérer comme voisinage l'objet auquel appartient le point considéré. Cela conduit à segmenter la scène. En l'absence de textures, l'intensité reste une méthode fiable pour segmenter une image, avec des méthodes comme le *Mean Shift* [14, 10]. Les segments obtenus sont alors utilisés comme voisinages [5]. Cette procédure reste coûteuse, c'est pourquoi [48] choisit de segmenter l'image en permettant à deux segments (initialisés par des pixels) de fusionner suivant des critères sur la taille des voisins et la proximité des couleurs. Une autre méthode populaire consiste à construire le voisinage en déployant une croix [46] (*cross-based regions*) autour du pixel, qui forme le squelette du voisinage. Ensuite, pour chaque pixel de la branche verticale, on agrandit le voisinage en déployant des branches horizontales de part et d'autre de la branche verticale. À nouveau, l'ajout d'un pixel

au voisinage est conditionné par son intensité.

Cependant, les méthodes les plus efficaces consistent à considérer des voisinages *non opaques*, c'est-à-dire où chaque pixel n'a pas le même poids dans l'agrégation. Cela revient à agréger les coûts en utilisant une moyenne pondérée. Ils sont généralement connus sous le nom de *fenêtres adaptatives*. L'une des méthodes les plus réputées et les plus efficaces est celle proposée par Yoon et Kweon [44, 13] où la pondération associée au voisin p' dépend à la fois de la distance dans l'espace des couleurs entre les intensités $I_{\mathrm{L}}(p)$ et $I_{\mathrm{L}}(p')$ et de la distance spatiale entre les deux pixels. Il s'agit en réalité du filtre bilatéral (voir [40] pour une revue plus détaillée sur les filtres bilatéraux). Une variante moins coûteuse en calculs [19] consiste à utiliser un filtre guidé [33, 40].

Après l'étape d'agrégation de coût, le pixel homologue retenu est celui qui minimise le coût de corrélation agrégé : cette étape est appelée WTA (*winner-take-all*).

Méthodes globales : minimisation d'une énergie Les méthodes globales choisissent d'exploiter la régularité de la scène. Pour ce faire, elles introduisent une fonctionnelle d'énergie qui pénalise la non-régularité de toute carte de disparité, tout en incitant l'algorithme à mettre en correspondance des pixels semblables. Un minimum de cette fonctionnelle est alors calculé, qui est la fonction satisfaisant au mieux les critères pénalisés. La difficulté réside principalement dans l'étape d'optimisation : les fonctionnelles d'énergie considérées ne sont généralement pas convexes, ce qui n'assure pas l'existence d'un minimum global. Par ailleurs, cela conduit à des algorithmes très coûteux en calculs. C'est pourquoi les modèles de régularité choisis dépendent essentiellement de leur compatibilité avec des algorithmes d'optimisation existants.

Les fonctionnelles d'énergie possèdent classiquement plusieurs termes, chaque terme étant dédié à une propriété particulière recherchée pour la carte de disparité. Un premier terme est le terme d'*attache aux données* ou de *fidélité*, qui mesure à quel point les pixels mis en correspondance sont semblables. Il est donc défini à l'aide de mesures de dissimilarité. Puisque l'estimation de la disparité ne repose plus uniquement sur la corrélation, mais est renforcée par d'autres termes que nous allons présenter, il n'est plus nécessaire de choisir une mesure de dissimilarité très performance. C'est pourquoi ce terme est généralement défini à partir des mesures AD ou SD, qui sont les moins coûteuses en calculs. Un second terme classique est le terme de *régularité*. Comme son nom l'indique, il mesure la régularité de la carte de disparité, qui reflète celle de la scène, composée d'objets de surfaces généralement lisses par morceaux. Il est donc généralement défini sur les variations de la carte de disparité. Celles-ci peuvent être pénalisées dès qu'elles existent [24] ou la pénalisation peut dépendre de l'amplitude des variations : c'est le cas par exemple de la régularisation quadratique ou de la régularisation TV (variation totale) [32]. La régularisation TV présente l'avantage de mieux préserver les discontinuités, car elle conduit à des cartes constantes par morceaux, alors que la régularisation quadratique conduit à des cartes très lisses. Une variante plus régulière de TV (régularisation Huber, [32]) permet d'obtenir des cartes avec des discontinuités nettes, tout en autorisant un léger gradient. Le critère de régularisation peut également concerner les segments de l'image [48, 5, 23, 47]. Si l'on retrouve systématique les termes de fidélité et de régularité, certains auteurs ajoutent un ou plusieurs autres termes. Il est ainsi naturel d'introduire un terme forçant l'*injectivité* de la mise en correspondance (*uniqueness term*, [24]). Un terme d'*occultation* ([24]) permettant de tenir compte de ce phénomène peut aussi être ajouté. Enfin, Papadakis et Caselles [31] ont proposé un terme gérant les contraintes de *visibilité*.

Le choix de la fonctionnelle d'énergie est fortement lié à la méthode d'optimisation

utilisée pour la minimiser. Compte tenu de la taille du problème, celle-ci est choisie pour son efficacité. Les premières méthodes globales se sont donc appuyées sur les méthodes d'optimisation 1D basées sur la programmation dynamique. Pour pouvoir utiliser ce genre de méthodes, Bobick et Intille [6] choisissent de définir une fonctionnelle composée d'énergies indépendantes définies sur les lignes de la disparité. Ainsi, le problème se réécrit comme un ensemble de problèmes de dimension 1, qu'ils résolvent indépendemment. Pour incorporer une régularisation verticale tout en tirant parti de l'efficacité des méthodes 1D, Hirschmüller [20] propose d'alterner des minimisations dans différentes directions, tandis que Veksler [42] transforme le problème en un problème sur un arbre, sur lequel elle peut utiliser la programmation dynamique.

Les résultats les plus satisfaisants restent ceux obtenus avec une régularisation 2D. L'utilisation des *graph cuts* [24] permettent de donner une solution approchée en alternant les α-*expansion moves* ou les $\alpha\beta$-*swap* qui font décroître l'énergie. Les premiers consistent à agrandir à chaque itération l'ensemble de niveau α de la disparité (cf. chapitre 4) tandis que les seconds considèrent deux ensembles de niveaux α et β dont ils échangent les éléments. Les approches bayésiennes basées sur le *belief propagation* (BP) ont été également connu un succès important [39]. L'idée est de reformuler le problème avec des champs de Markov, où les différents termes de l'énergie se traduisent par des interactions entre les nœuds du réseau. Il s'agit ensuite de calculer le *maximum a priori* (MAP) en utilisant un algorithme de BP qui met à jour la disparité en faisant passer des messages à travers le réseau. Enfin, Pock et coll. [32] ont proposé une méthode de relaxation convexe de l'énergie, ce qui leur permet d'exploiter les outils d'optimisation convexe. Cette méthode, sur laquelle se base le chapitre 3, possède l'avantage de s'appliquer à une classe très large de fonctionnelles d'énergie. Enfin, on pourra citer les travaux de [29] et [9], qui exploitent également des outils d'optimisation convexe en considérant une version approchée mais convexe du problème initial non convexe qu'ils souhaitent résoudre, en utilisant une linéarisation d'une des images autour d'une première estimation de la disparité. Contrairement à la méthode de Pock et coll., cette approche ne constitue donc pas une relaxation convexe *exacte* du problème.

2.3.3 Occultation, correspondances non fiables

Du fait du phénomène d'occultation et des nombreuses difficultés soulevées dans le paragraphe 2.1.4 qui rendent la mise en correspondance difficile, la gestion des erreurs est un volet important de toute méthode de stéréovision. Il peut être utile de distinguer les erreurs dues à l'occultation des autres, qui peuvent résulter d'une multitude d'origines.

Gestion des occultations Dans les méthodes locales, les occultations sont généralement traitées comme des mises en correspondance non fiables, qui font l'objet du paragraphe suivant. Néanmoins, l'utilisation de mesures robustes aux occultations a été proposée dans [7].

Dans les méthodes globales, l'occultation peut être prise en compte grâce à l'introduction d'un terme dédié. Dans [6], les auteurs exploitent l'analyse de la section 2.2 pour classer les pixels en trois familles en exploitant les variations de chaque ligne de la disparité : ceux qui sont mis en correspondance (variations horizontales), ceux qui correspondant à des occultations (variations diagonales) et ceux qui correspondent à des désoccultations (variations verticales). Le désavantage majeur de cette méthode est qu'elle ne tient pas du tout compte de la régularité verticale de la scène car le problème

est traité indépendamment sur chaque ligne. Dans [24], le terme d'occultation compte le nombre de pixels non mis en correspondance avec un pixel homologue. Ce terme est pondéré par un paramètre choisi de sorte à contrôler le nombre de pixels occultés, ce qui implique de réussir à l'estimer de manière empirique ou heuristique.

Des approches plus complexes peuvent également être envisagées. Dans [38], on alterne estimation de la disparité sur les régions non occultées et estimation des régions occultées.

Détection et rejet des pixels non fiables Les cartes de disparité fournies par une méthode globale possèdent une cohérence globale grâce au terme de régularité et éventuellement au terme d'injectivité. Lorsque l'occultation est prise en compte, les zones occultées n'ont soit aucune disparité d'attribuée, soit celle-ci n'est pas significative. Dans les deux cas, leur localisation est connue et on peut facilement rejeter l'information dans ces régions.

Du fait de leur caractère local, les méthodes locales génèrent des cartes généralement moins fiables. Les erreurs sont principalement dues à l'occultation, à l'adhérence, à l'effet de Strobes ou au manque de textures. Contrairement aux méthodes globales, aucun critère de régularité ne permet de détecter ces erreurs. C'est pourquoi on définit des filtres de rejets que l'on applique une fois la carte de disparité estimée. Le filtre le plus populaire est le filtre LRRL (*left-right right-left*) [27, 33], qui traduit la contrainte d'injectivité. La carte de disparité sur la vue de référence et celle sur la vue de droite sont calculées. Ensuite, on évalue leur cohérence : tous les pixels de l'image de référence dont le pixel homologue dans la vue de droite n'est pas mis en correspondance avec lui dans l'estimation de la disparité de droite sont rejetés. Ce filtre nécessite de calculer deux cartes, donc de doubler le nombre d'opérations, mais ce n'est généralement pas un problème majeur car les méthodes locales sont peu coûteuses en calculs.

Un autre outil puissant est la validation *a contrario* [34]. Les méthodes *a contrario* reposent sur le principe d'Helmholtz qui assure que *dans le bruit, on ne voit rien.* Autrement dit, les structures détectables sont celles qui ont une faible de chance de se produire au hasard. Appliquée à la mise en correspondance stéréoscopique, ce principe permet de rejeter des mises en correspondance en mesurant la probabilité que la similarité entre les deux pixels (après agrégation des coûts) soit due au hasard. L'inconvénient majeur de ce filtre est qu'il rejette beaucoup de points.

Après l'application d'un ou de plusieurs de ces filtres ou l'extraction des zones d'occultation, on se retrouve avec des cartes de disparité incomplètes, dites *éparses* ou *non denses*. Pour obtenir une disparité définie partout, il faut ajouter une étape de *densification.*

Densification des cartes La densification des cartes de disparité consiste à compléter la carte de disparité où elle n'est pas connue. La stratégie choisie dépend de la raison pour laquelle le pixel a été rejeté. Dans [33], les auteurs considèrent que les pixels rejetés le sont principalement car ils sont occultés. Or, l'analyse menée dans 2.2 assure que, dans l'hypothèse la plus simple où les objets partiellement occultés ont une disparité constante (même au niveau de la partie occultée), la disparité de la région occultée est celle de l'objet occulté entier. Or, l'occultation étant située à gauche des discontinuités, la partie non occultée de l'objet occulté est située à gauche de la partie occultée. La densification se fait alors en diffusant la disparité vers la droite dans les régions occultées.

(a) Tsukuba (b) Venus (c) Teddy (d) Cones

FIGURE 2.8 – Vue de référence des quatre paires de la version 2 du banc d'essai Middlebury.

2.3.4 Le banc d'essai Middlebury

Suite à leur article de revue [36], SCHARSTEIN et SZELISKI ont mis en ligne le banc d'essai Middlebury [11]. Les chercheurs sont invités à tester leur algorithme sur les images proposées et à soumettre leurs résultats (mais pas leur algorithme). Leurs résultats sont classés suivant plusieurs critères (détaillés plus bas). La principale règle est que les paramètres ne doivent pas être adaptés manuellement à chaque scène.

Version 2 Pour la version 2 du banc d'essai, quatre paires sont proposées : Tsukuba, Venus [36], Teddy et Cones [37] (cf. figure 2.8). Les tailles des images varient entre 384×288 pour Tsukuba et 450×375 pour Teddy et Cones. Les caméras sont en mouvement fronto-parallèle. Les intervalles de disparité sont fournis : leur longueur varie entre 16 pixels (pour Tsukuba) et 60 pixels (pour Teddy et Cones). Les vérités-terrains sont disponibles pour les quatre paires.

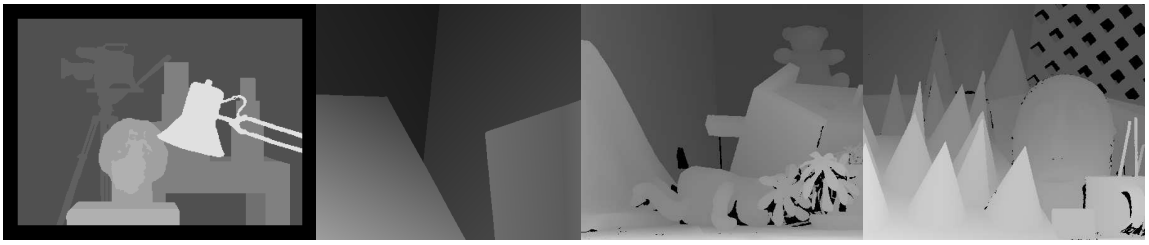
La version 2 n'est plus active depuis 2015. Elle a été remplacée par la version 3 [12] qui propose deux ensembles de paires : un ensemble avec vérité-terrains qui servent d'entraînement et un ensemble sans vérité-terrain pour des tests à l'aveugle. Les paires proposées dans la version 3 sont également beaucoup plus grandes (2880×1988 pour la paire Adirondack par exemple).

Vérité-terrain Les vérités-terrains (cf. figure 2.9) sont générées en projetant sur la scène des motifs réguliers [37] pour encoder chaque point de la scène, puis les disparités gauche et droite sont estimées pour N éclairages différents (ce qui permet en particulier de déplacer les ombres). Les cartes de disparité sont fusionnées et les zones d'occultation détectées. Un sous-échantillonnage des paires et de la carte de disparité finale est finalement effectué. La disparité de certains points de la scène reste inconnue.

Caractéristiques des scènes Ces scènes d'intérieur présentent peu d'ombre, peu de reflets (seule la paire Tsukuba en possède). Les images sont très texturées, les objets opaques. Il n'y a pas de changement d'illumination notable entre les deux vues.

La paire Tsukuba présente une vérité-terrain constante par morceaux et pixellique (alors que la lampe devrait logiquement être un peu bombée). La caméra et la lampe présentent des parties qui sont fines, mais seule la partie basse du fil de la lampe induit une inversion de l'ordre. La lampe et certaines parties métalliques de l'étagère au fond de la scène génèrent des reflets. La paire Venus est composée de panneaux texturés, non parallèles au plan image. Cela induit une vérité-terrain affine par morceaux. La

11. http ://vision.middlebury.edu/stereo
12. http ://vision.middlebury.edu/stereo/eval3



(a) Tsukuba (b) Venus (c) Teddy (d) Cones

FIGURE 2.9 – Vérité-terrain des quatre paires de la version 2 du banc d'essai Middlebury (vue de référence). Plus le pixel est clair, plus sa disparité est grande. En noir, les points dont la disparité n'est pas connue.

paire Teddy possède un intervalle de disparité très large (60 pixels), ce qui crée des zones d'occultation tout aussi larges. Par ailleurs, la texture du sol s'apparente à un bruit. La paire Cones possède également des zones d'occultation larges. De plus, les objets fins (pointes des cones, pinceaux) induisent une inversion de l'ordre.

Évaluation Pour chaque image, différents scores sont proposés : le pourcentage de disparité correcte dans l'image entière (*all*), près des zones de discontinuités de la scène (*nondisc*) et hors des régions occultées (*nonocc*). Pour cela, des masques sont proposés (cf. figure 2.11). Pour chaque masque, les erreurs sont mesurées à un seuil près (égal à 2, 1,5, 1, 0,75 ou 0,5 pixels). Pour chaque paire et chaque masque, le pourcentage d'erreur et le classement sont donnés. Par défaut, l'affichage correspond au classement moyen sur tous les masques pour un seuil de 1 pixel, mais le pourcentage moyen d'erreurs sur toutes les paires est également visible. Dans la figure 2.10 qui présente la tête du classement à la date de clôture de la version 2, on voit en particulier que la méthode PM-Forest est la meilleure pour le pourcentage moyen d'erreur (avec 2,64% d'erreur) mais n'arrive que neuvième au classement général.

2.3.5 Démos IPOL

La recherche reproductible En sciences, la reproductibilité d'une expérience permet de garantir sa pertinence et de valider les conclusions qui en découlent. Elle implique que tout résultat publié doit pouvoir être obtenu de manière identique si les conditions de l'expérience sont reproduites. Appliquée au domaine du traitement de l'image, la reproductibilité suppose que si l'on implémente la méthode décrite par un article, et qu'on l'applique aux mêmes données que celles testées par les auteurs, alors le résultat sera analogue. Si en théorie, cette condition semble facile à satisfaire (contrairement aux sciences pratiques, le nombre de paramètres influant l'expérience est limité, car un code informatique réagit systématiquement de la même manière), en pratique, elle implique d'avoir à disposition une implémentation analogue et les mêmes jeux de données. Or, ces deux éléments sont rarement disponibles.

Lorsqu'une méthode est publiée, l'article présente généralement une description (plus ou moins détaillée) de l'algorithme, ainsi qu'un pseudo-code présentant l'architecture du code. Malheureusement, ces informations sont loin d'être suffisantes pour réimplémenter l'algorithme proposé. Principalement à cause du nombre limité de pages, des éléments essentiels sont manquants : la valeur des paramètres sont rarement précisés (ou il en manque certains), d'éventuelles étapes de pré-traitement des données

Stereo Evaluation • Datasets • Code • Submit

Middlebury Stereo Evaluation - Version 2

Version 2 is no longer active. Please use the **Stereo Evaluation Version 3**

New features and main differences to version 1.

Open a new window for each link

Error Threshold = 1 (Error Threshold...) — Sort by nonocc ▼ — Sort by all ▼ — Sort by disc ▼

Algorithm	Avg. Rank ▼	Tsukuba ground truth nonocc ▼	Tsukuba all	Tsukuba disc	Venus ground truth nonocc	Venus all ▼	Venus disc	Teddy ground truth nonocc	Teddy all ▼	Teddy disc	Cones ground truth nonocc	Cones all ▼	Cones disc	Average percent of bad pixels (explanation)
IGSM [155]	10.4	0.93 10	1.37 12	5.05 12	0.07 2	0.17 5	1.04 2	4.08 20	5.98 10	11.4 21	2.14 9	6.97 14	6.27 8	3.79
TSGO [141]	13.4	0.87 4	**1.13** 1	4.66 6	0.11 10	0.24 14	1.47 13	5.61 46	8.09 21	13.8 39	1.67 2	6.16 3	4.95 2	4.06
JSOSP+GCP [149]	14.8	**0.74** 1	1.34 9	**3.98** 1	0.08 4	**0.16** 1	1.15 4	3.96 18	10.1 40	11.8 22	2.28 19	7.91 37	6.74 22	4.18
KADI [164]	15.2	1.02 16	1.23 4	5.51 17	0.08 3	0.20 8	1.11 3	5.16 37	9.43 35	13.0 33	2.07 4	7.16 19	5.97 4	4.33
SSCBP [157]	17.6	1.05 19	1.39 14	5.57 19	0.10 7	0.16 2	1.39 10	3.44 14	8.32 26	9.95 15	2.60 34	7.13 18	7.23 33	4.03
ADCensus [82]	18.2	1.07 23	1.48 21	5.73 26	0.09 5	0.25 18	1.15 4	4.10 21	6.22 11	10.9 18	2.42 25	7.25 21	6.95 26	3.97
AdaptingBP [16]	22.2	1.11 26	1.37 11	5.79 28	0.10 8	0.21 13	1.44 12	4.22 23	7.06 19	11.8 23	2.48 29	7.92 39	7.32 36	4.23
CoopRegion [39]	22.2	0.87 6	1.16 2	4.61 5	0.11 9	0.21 10	1.54 17	5.16 38	8.31 25	13.0 31	2.79 48	7.18 20	8.01 56	4.41
CCRADAR [150]	26.8	1.15 28	1.42 18	6.23 41	0.15 22	0.27 21	1.89 27	5.39 41	10.6 45	14.7 50	2.01 3	7.37 23	5.88 3	4.75
PM-Forest [162]	27.2	1.63 78	2.17 79	8.71 97	0.15 24	0.19 7	2.13 36	**1.91** 1	**2.29** 1	**5.47** 1	**1.32** 1	**2.02** 1	**3.69** 1	2.64
RDP [87]	28.7	0.97 12	1.39 15	5.00 11	0.21 46	0.38 38	1.89 27	4.84 29	9.94 39	12.6 28	2.53 33	7.69 29	7.38 37	4.57
MultiRBF [129]	28.7	1.33 53	1.56 27	6.02 37	0.13 15	0.17 4	1.84 24	5.09 35	6.36 12	13.4 37	2.90 58	6.76 11	7.10 31	4.39
DoubleBP [34]	29.0	0.88 8	1.29 7	4.76 9	0.13 16	0.45 56	1.87 26	3.53 17	8.30 24	9.63 11	2.90 57	8.78 69	7.79 48	4.19
OutlierConf [40]	30.0	0.88 7	1.43 19	4.74 8	0.18 35	0.26 20	2.40 45	5.01 31	9.12 33	12.8 30	2.78 47	8.57 58	6.99 27	4.60
SegAggr [144]	30.2	1.99 99	2.39 89	8.59 96	0.12 11	0.21 12	1.68 19	2.19 3	3.73 3	7.02 3	2.16 11	6.52 6	6.37 11	3.58
CVW-RM [146]	30.4	1.12 27	1.42 17	5.99 36	0.16 30	0.36 36	1.40 11	4.70 28	6.94 17	12.1 24	2.96 63	7.71 31	7.72 45	4.38
GC+LocalExp [158]	32.0	1.48 68	1.88 62	6.95 66	0.13 14	0.25 17	1.52 16	3.33 12	4.88 5	8.87 7	2.72 41	7.42 24	7.94 52	3.95
SOS [135]	35.0	1.45 64	1.63 33	7.83 84	0.21 44	0.32 29	2.29 44	3.13 11	8.45 28	9.74 12	2.43 26	7.10 17	7.02 28	4.30
SubPixSearch [109]	35.4	2.04 103	2.48 93	6.40 47	0.14 20	0.40 45	1.74 21	4.00 19	6.39 13	11.0 19	2.24 16	6.87 13	6.50 16	4.18

FIGURE 2.10 – Capture d'écran de la page de la version 2 du banc d'essai Middlebury (prise après sa désactivation). Les différents algorithmes sont classés suivant leur performance sur les quatre paires.

(a) Vérité-terrain (b) *all* (c) *disc* (d) *nonocc*

FIGURE 2.11 – Les masques utilisés par Middlebury pour évaluer les résultats soumis (exemple de la paire Tsukuba). En noir, les pixels dont la disparité n'est pas connue. En blanc, les points sur lesquels l'estimation de la disparité est évaluée : (b) partout où la vérité-terrain est connue ; (c) près des discontinuités de la scène ; (d) dans les régions non occultées.

ou post-traitement des résultats sont passées sous silence. Pourtant, ces détails jugés «techniques» sont nécessaires pour reproduire les résultats publiés.

Pour pallier ce manque, qui discrédite à terme la communauté scientifique, des plateformes ont vu le jour pour permettre aux auteurs de publier leurs codes. En 2011, le CNRS, HEC Paris et l'université d'Orléans lancent RunMyCode [13], qui permet d'associer à tout article de recherche (de tout domaine) une page web «compagnon» sur laquelle les auteurs postent leur code et les données sur lesquelles ils ont testé leur méthode. Le lien de cette page est alors cité dans l'article. L'intérêt est assez évident : la recherche redevient reproductible, puisque n'importe qui peut regénérer les résultats présentés par l'article. Il n'y a pas de paramètre ou d'étape cachée. On n'a pas besoin de réimplémenter la méthode, ce qui permet de gagner du temps. Enfin, on peut tester l'algorithme sur d'autres images que celles (forcément en nombre restreint) proposées par les auteurs.

Néanmoins, contrairement au contenu de l'article qui est expertisé par un comité de lecture, les codes publiés ne font l'objet d'aucune validation scientifique. En particulier, on n'y contrôle pas que le code correspond à l'algorithme décrit et que les résultats présentés sont bien obtenus grâce à ce code.

IPOL Le journal en ligne IPOL [14] (*Image Processing On Line*) est un journal lancé en 2009, à l'initiative de Nicolas Limare, Jean-Michel Morel et l'équipe Traitement d'Images et du Signal du CMLA (Centre de Mathématiques et leurs Applications), à l'ENS Cachan. Son objectif est de proposer des articles de traitement d'images, accompagnés d'un code qui est soumis à une expertise approfondie, ce qui permet de combler les limites de la plateforme RunMyCode.

Toute publication d'IPOL comporte trois composantes :

1. l'article à proprement parler ;
2. l'implémentation de l'algorithme présenté en ANSI C/C++ ou en Matlab (depuis mai 2015) ;
3. la partie *démo*, qui permet à l'utilisateur de faire tourner le code en ligne sur des images proposées par IPOL ou sur ses propres images.

L'article, qui n'est pas limité en nombre de pages, présente une méthode intéressante (non nécessairement originale). L'algorithme doit être décrit de manière exhaustive, en particulier les paramètres. Idéalement, cette description seule doit permettre à tout lecteur d'implémenter sa propre version de la méthode. Enfin, une analyse critique des résultats clôt l'article : les cas satisfaisants sont présentés, aussi bien que les mauvais, pour démontrer les apports et les limites de la méthode.

Un code (en ANSI C/C++ ou en Matlab) est également soumis, qui sera expertisé au même titre que l'article. Il doit être portable, c'est-à-dire pouvoir tourner sur toute machine standard (sous Windows, MacOS ou Linux). Il doit être suffisamment documenté pour permettre à tout lecteur de le comprendre. Enfin, puisqu'il a vocation à être diffusé et la démo maintenue par l'équipe IPOL, il doit être publié sous une licence de logiciel libre. Pour assurer sa diffusion la plus large possible, il doit utiliser des bibliothèques standards et stables. Les codes Matlab sont autorisés depuis 2015 car leur implémentation et leur utilisation sont plus souples que les codes ANSI C/C++. C'est par ailleurs l'un des langages les plus utilisés dans la communauté image.

13. http ://www.runmycode.org
14. http ://www.ipol.im

La partie *démo* constitue le dernier volet d'une publication IPOL. Elle se présente sous la forme d'une page dédiée, sur laquelle le *même* code que celui qui est publié peut être testé en ligne, sur les serveurs d'IPOL. Cela introduit une contrainte sur l'efficacité des méthodes publiées, qui doivent produire un résultat en moins de 30 secondes, quitte à restreindre la taille des données en entrée. Les implémentations exploitant le calcul parallèle sont donc encouragées. Le cas échéant, l'utilisateur peut changer les paramètres de la méthode, à l'aide de curseurs. Les algorithmes peuvent être testés sur les images proposées par les auteurs, mais également sur les images que l'utilisateur charge lui-même. Les résultats sur les images personnelles sont archivés et consultables en ligne (à moins que l'utilisateur ne l'interdise). Cette disposition permet de constituer une base de données plus importante qui montre comment se comporte l'algorithme sur des images variées.

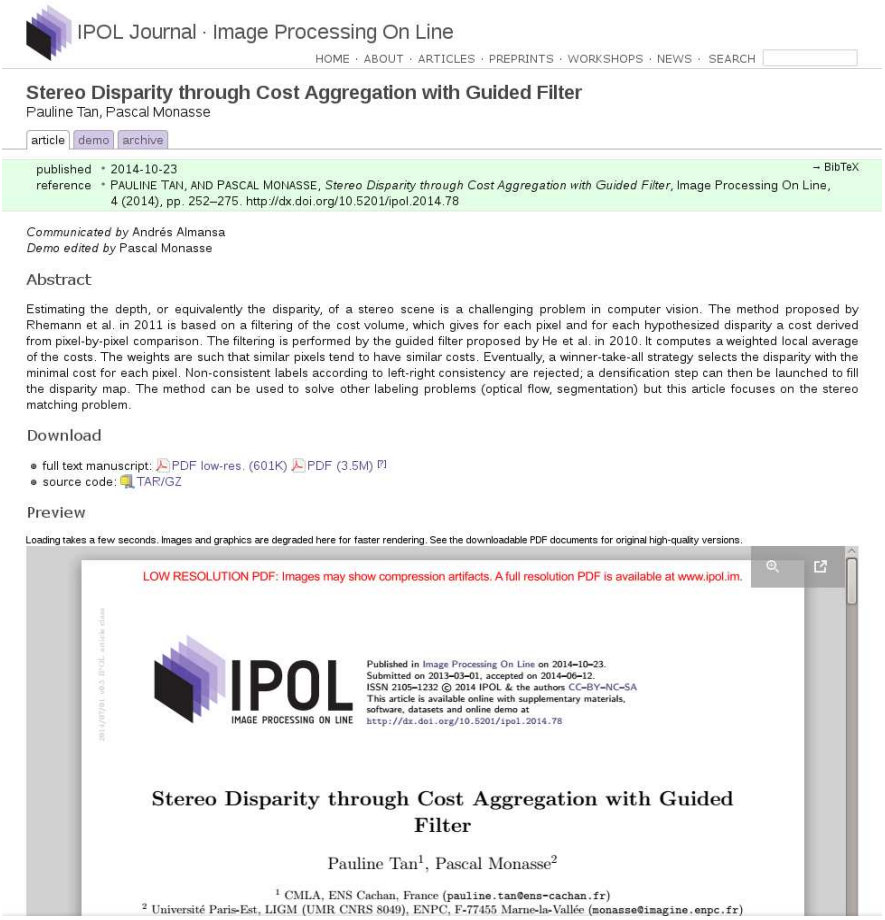
La politique très exigeante d'IPOL garantit une réelle reproductibilité des expériences publiées. Néanmoins, pour les auteurs, elle se traduit par une charge de travail plus lourde. Le code doit être lisible, documenté et maintenu. Ils doivent s'assurer de sa portabilité sur différentes plateformes. Or, ce travail est peu gratifiant, car généralement peu reconnu.

Le journal IPOL souhaite proposer dans chaque domaine du traitement d'images un maximum de méthodes constituant l'état de l'art. Actuellement (en 2016), six algorithmes ont été publiés dans la section stéréovision.

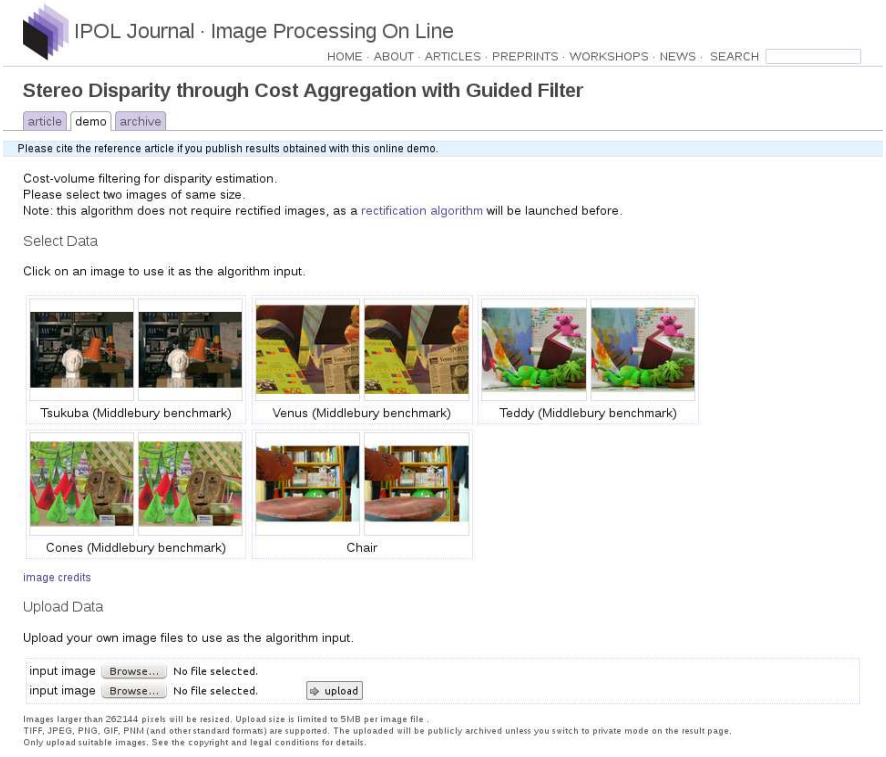
Algorithme de rectification épipolaire Un algorithme de rectification épipolaire [30] a été publié en 2011. Il estime deux homographies qui permettent de simuler deux vues en déplacement fronto-parallèle à partir d'une paire stéréoscopique quelconque. Elle est basée sur la méthode de Fusiello et Irsara [15], qui suppose que les deux caméras initiales sont parfaites (le point principal coïncide avec le centre du cadre) mais de (même) distance focale inconnue. La rectification est réalisée en minimisant le mouvement vertical de certains points mis en correspondance, sélectionnés par la méthode SIFT [25].

Algorithmes de mise en correspondance stéréoscopique Quatre articles ont été publiés sur IPOL à propos de la mise en correspondance stéréoscopique proprement dite. La première, publiée en 2014, est pour l'instant la seule méthode globale disponible sur IPOL. Elle est basée sur la méthode proposée en 2001 par Kolmogorov et Zabih [24]. Le code proposé est une variante du code de Kolmogorov (disponible sur sa page personnelle), plus adaptée aux standards d'IPOL. Malgré l'efficacité de l'implémentation des *graph cuts*, il a fallu découper les paires en bandes horizontales (avec recouvrement), afin de les traiter en parallèle, pour atteindre le temps d'exécution imposé par IPOL.

Les trois autres articles sont des méthodes locales. Le premier [12] présente une méthode qui permet d'agréger efficacement les coûts de corrélation, en s'appuyant sur une table de sommation, proposée par [11]. Ce même algorithme est à l'origine de la méthode présentée par [40], qui décrit l'algorithme initialement publié dans [33]. Il s'agit d'une méthode basée sur une implémentation efficace d'un filtre bilatéral. Enfin, l'article [13] propose une implémentation des célèbres fenêtres adaptatives (qui repose également sur un filtre bilatéral) de Yoon et Kweon [44], déjà évoquées plus haut dans ce chapitre.



(a)



(b)

FIGURE 2.12 – Les onglets d'une publication IPOL (captures d'écran), exemple de [40] : (a) l'article et le lien de téléchargement du code ; (b) la démo.

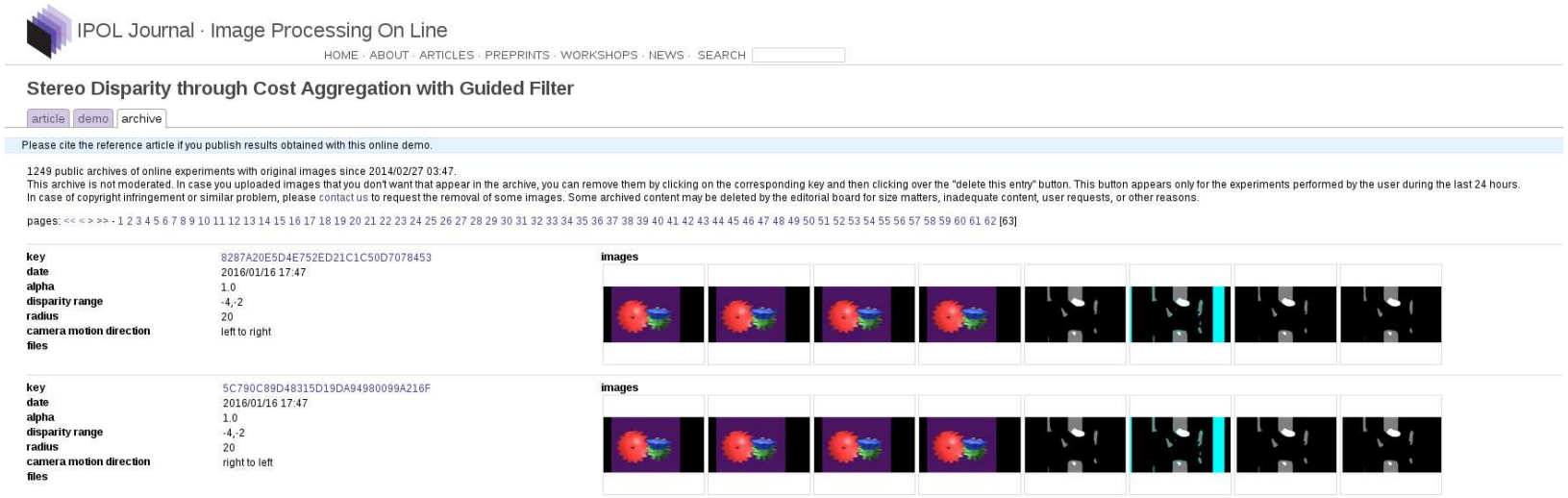


FIGURE 2.13 – Les onglets (suite et fin) d'une publication IPOL (captures d'écran), exemple de [40] : l'archive.

Références

[1] Satyajit Anil Adhyapak, Nasser Kehtarnavaz, and Mihai Nadin. Stereo matching via selective multiple windows. *Journal of Electronic Imaging*, 16(1) :013012, 2007.

[2] Thomas Belli, Matthieu Cord, and Sylvie Philipp-Foliguet. Colour contribution for stereo image matching. In *International Conference on Color in Graphics and Image Processing*, pages 317–322. Citeseer, 2000.

[3] Stan Birchfield and Carlo Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4) :401–406, 1998.

[4] Michael Bleyer and Sylvie Chambon. Does color really help in dense stereo matching ? In *International Symposium 3D Data Processing, Visualization and Transmission 2010*, pages 1–8, 2010.

[5] Michael Bleyer and Margrit Gelautz. A layered stereo algorithm using image segmentation and global visibility constraints. In *IEEE International Conference on Image Processing*, volume 5, pages 2997–3000. IEEE, 2004.

[6] Aaron F. Bobick and Stephen S. Intille. Large occlusion stereo. *International Journal of Computer Vision*, 33(3) :181–200, 1999.

[7] Sylvie Chambon. *Mise en correspondance stéréoscopique d'images couleur en présence d'occultations*. PhD thesis, Université Paul Sabatier-Toulouse III, 2005.

[8] Sylvie Chambon and Alain Crouzil. Color stereo matching using correlation measures. *Complex Systems Intelligence and Modern Technological Applications*, pages 520–525, 2004.

[9] Caroline Chaux, Mireille El-Gheche, Joumana Farah, Jean-Christophe Pesquet, and Béatrice Pesquet-Popescu. A parallel proximal splitting method for disparity estimation from multicomponent images under illumination variation. *Journal of mathematical imaging and vision*, 47(3) :167–178, 2013.

[10] Dorin Comaniciu and Peter Meer. Mean shift : A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5) :603–619, 2002.

[11] Franklin C. Crow. Summed-area tables for texture mapping. *SIGGRAPH Computer Graphics*, 18(3) :207–212, 1984.

[12] Gabriele Facciolo, Nicolas Limare, and Enric Meinhardt-Llopis. Integral images for block matching. *Image Processing On Line*, 4 :344–369, 2014.

[13] Laura Fernández Julià and Pascal Monasse. Bilaterally weighted patches for disparity map computation. *Image Processing On Line*, 5 :73–89, 2015.

[14] Keinosuke Fukunaga and Larry D. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1) :32–40, 1975.

[15] Andrea Fusiello and Luca Irsara. Quasi-euclidean uncalibrated epipolar rectification. In *IEEE International Conference on Pattern Recognition*, pages 1–4. IEEE, 2008.

[16] Andrea Fusiello, Vito Roberto, and Emanuele Trucco. Symmetric stereo with multiple windowing. *International Journal of Pattern Recognition and Artificial Intelligence*, 14(08) :1053–1066, 2000.

[17] Davi Geiger, Bruce Ladendorf, and Alan Yuille. Occlusions and binocular stereo. *International Journal of Computer Vision*, 14(3) :211–226, 1995.

[18] Marsha J. Hannah. Computer matching of areas in stereo images. Technical report, DTIC Document, 1974.

[19] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6) :1397–1409, 2013.

[20] Heiko Hirschmüller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2) :328–341, 2008.

[21] Heiko Hirschmüller and Daniel Scharstein. Evaluation of cost functions for stereo matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[22] Ian P. Howard and Brian J. Rogers. *Binocular vision and stereopsis*. Oxford University Press, 1995.

[23] Andreas Klaus, Mario Sormann, and Konrad Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *IEEE International Conference on Pattern Recognition*, volume 3, pages 15–18. IEEE, 2006.

[24] Vladimir Kolmogorov and Ramin Zabih. Computing visual correspondence with occlusions using graph cuts. In *IEEE International Conference on Computer Vision*, volume 2, pages 508–515. IEEE, 2001.

[25] David G. Lowe. Object recognition from local scale-invariant features. In *IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157. IEEE, 1999.

[26] David Marr and Ellen Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London B : Biological Sciences*, 207(1167) :187–217, 1980.

[27] Stefano Mattoccia, Federico Tombari, and Luigi Di Stefano. Stereo vision enabling precise border localization within a scanline optimization framework. In *Asian Conference on Computer Vision*, pages 517–527. Springer, 2007.

[28] Xing Mei, Xun Sun, Mingcai Zhou, Shaohui Jiao, Haitao Wang, and Xiaopeng Zhang. On building an accurate stereo matching system on graphics hardware. In *IEEE International Conference on Computer Vision Workshops*, pages 467–474. IEEE, 2011.

[29] Wided Miled, Jean-Christophe Pesquet, and Michel Parent. A convex optimization approach for depth estimation under illumination variation. *IEEE Transactions on Image Processing*, 18(4) :813–830, 2009.

[30] Pascal Monasse. Quasi-euclidean epipolar rectification. *Image Processing On Line*, 1, 2011.

[31] Nicolas Papadakis and Vicent Caselles. Multi-label depth estimation for graph cuts stereo problems. *Journal of Mathematical Imaging and Vision*, 38(1) :70–82, 2010.

[32] Thomas Pock, Daniel Cremers, Horst Bischof, and Antonin Chambolle. Global solutions of variational models with convex regularization. *SIAM Journal on Imaging Sciences*, 3(4) :1122–1145, 2010.

[33] Christoph Rhemann, Asmaa Hosni, Michael Bleyer, Carsten Rother, and Margrit Gelautz. Fast cost-volume filtering for visual correspondence and beyond. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3017–3024. IEEE, 2011.

[34] Neus Sabater. *Fiabilité et précision en stéréoscopie : application à l'imagerie aérienne et satellitaire à haute résolution.* PhD thesis, École normale supérieure de Cachan, 2009.

[35] Daniel Scharstein. Matching images by comparing their gradient fields. In *IAPR International Conference on Pattern Recognition*, volume 1, pages 572–575. IEEE, 1994.

[36] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3) :7–42, 2002.

[37] Daniel Scharstein and Richard Szeliski. High-accuracy stereo depth maps using structured light. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–195. IEEE, 2003.

[38] Jian Sun, Yin Li, Sing Bing Kang, and Heung-Yeung Shum. Symmetric stereo matching for occlusion handling. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 399–406. IEEE, 2005.

[39] Jian Sun, Nan-Ning Zheng, and Heung-Yeung Shum. Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7) :787–800, 2003.

[40] Pauline Tan and Pascal Monasse. Stereo disparity through cost aggregation with guided filter. *Image Processing On Line*, pages 252–275, 2014.

[41] Federico Tombari, Stefano Mattoccia, Luigi Di Stefano, and Elisa Addimanda. Classification and evaluation of cost aggregation methods for stereo correspondence. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[42] Olga Veksler. Stereo correspondence by dynamic programming on a tree. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 384–390. IEEE, 2005.

[43] Paul Viola and William M. Wells III. Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24(2) :137–154, 1997.

[44] Kuk-Jin Yoon and In So Kweon. Adaptive support-weight approach for correspondence search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (4) :650–656, 2006.

[45] Ramin Zabih and John Woodfill. Non-parametric local transforms for computing visual correspondence. In *European Conference on Computer Vision*, pages 151–158. Springer, 1994.

[46] Ke Zhang, Jiangbo Lu, and Gauthier Lafruit. Cross-based local stereo matching using orthogonal integral images. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(7) :1073–1079, 2009.

[47] C. Lawrence Zitnick and Sing Bing Kang. Stereo for image-based rendering using image over-segmentation. *International Journal of Computer Vision*, 75(1) :49–65, 2007.

[48] C. Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High-quality video view interpolation using a layered representation. In *ACM Transactions on Graphics*, volume 23, pages 600–608. ACM, 2004.

Deuxième partie

Calcul de cartes de disparité par méthode globale

Chapitre 3

Gérer les occultations par méthode variationnelle

Introduction

En stéréovision binoculaire, les méthodes globales traduisent le problème de mise en correspondance par une fonctionnelle d'énergie. Celle-ci encode les propriétés de la

carte de disparité recherchée, en pénalisant tout écart à ces conditions. Cette approche conduit à optimiser des problèmes de grande taille, ce qui est très coûteux en calculs. Le choix de la méthode d'optimisation est donc crucial. C'est pourquoi il est généralement prioritaire sur le choix de la fonctionnelle d'énergie. Toutes les fonctionnelles ne sont en effet pas minimisables par une méthode donnée. C'est donc *le choix de la méthode d'optimisation qui conditionne la forme de la fonctionnelle* et, de fait, le choix du modèle de scène sous-jacent. Toute méthode globale peut être évaluée sur ces deux points : la pertinence du modèle considéré et l'efficacité algorithmique de la minimisation.

Une méthode comme celle basée sur l'utilisation des *graph cuts* de Kolmogorov et Zabih [10], que nous étudierons en détails dans le chapitre suivant, présente une bonne efficacité algorithmique. En revanche, le modèle de régularité est biaisé par l'utilisation des graphes et la minimisation n'est pas exacte, mais seulement approchée. Néanmoins, c'est l'une des rares méthodes à gérer de manière raisonnable les occultations de la scène, en proposant à un pixel soit d'être mis en correspondance, soit d'être occulté. Toutefois, ce choix n'est basé que sur le coût de corrélation, et ne repose donc sur aucun modèle de scène. La méthode de programmation dynamique proposée par Bobick et Intille [1] repose quant à elle sur l'étude menée au chapitre précédent sur l'occultation. Lorsque l'ordre est préservé dans les deux images, on a montré que la pente horizontale de la disparité est strictement inférieure à 1, et que la largeur des occultations valait exactement le saut de disparité autour de l'occultation. Leur méthode repose sur ce modèle de régularité de la scène, dont on a montré la pertinence. Malheureusement, la méthode d'optimisation utilisée, qui par ailleurs est très efficace, n'est valable qu'en dimension 1, et leur méthode ne peut donc pas intégrer la régularité verticale. On voit ainsi que, dans ces deux exemples de méthodes gérant l'occultation, le modèle de régularité est défaillant, car il doit s'adapter à la méthode d'optimisation choisie.

Dans leur article de 2010, Pock et coll. proposent une méthode qui permet de transformer une classe très large de fonctionnelles en un problème convexe [12], par relaxation. La résolution de ce dernier problème fait alors appel à l'optimisation convexe, pour laquelle on dispose de davantage d'outils efficaces, du fait de certaines propriétés des fonctions convexes (dont en particulier l'existence d'un minimum). Parmi les fonctionnelles admissibles, se trouvent celles possédant un terme d'attache aux données quelconque (mais continu) et un terme de régularité convexe. Leur méthode est donc relativement peu contraignante pour la forme de la fonctionnelle, mais ne peut pas gérer un terme d'occultation comme celui que proposent Kolmogorov et Zabih [10]. D'où l'idée de reprendre le modèle de scène utilisé par Bobick et Intille [1] et d'y ajouter une régularité verticale.

Dans ce chapitre, nous proposons donc une nouvelle fonctionnelle d'énergie (section 3.1), qui est à la fois admissible par la méthode de relaxation convexe [12] et qui tienne compte du phénomène d'occultation de la même manière que dans [1], c'est-à-dire en intégrant le modèle de scène détaillé dans le chapitre précédent (paragraphe 3.1.2). Nous introduirons également un nouveau de terme d'attache aux données adaptatif (paragraphe 3.1.1) qui a pour but de limiter l'effet d'adhérence tout en bénéficiant de l'efficacité de la corrélation combinée de couleur et de gradient. Pour relaxer ce problème de manière convexe (section 3.2), nous reprenons une à une chaque étape de l'article [12]. Le problème convexe résultant peut s'écrire de manière primale-duale (paragraphe 3.2.3), et nous proposerons deux méthodes de résolution numérique (section 5.2). Puisque la fonctionnelle choisie permet de détecter les occultations dans la carte de disparité obtenue par relaxation convexe, une étape de densification est proposée (section 3.4). Nous présenterons enfin les résultats obtenus avec cette mé-

thode (section 3.5) : elle sera testée sur les images du banc d'essai Middlebury, et nous comparerons la détection des occultations avec la méthode des *graph cuts* de [10].

3.1 Fonctionnelle d'énergie

On rappelle que le domaine de image de référence est rectangulaire, noté $\Omega \subset \mathbb{R}^2$. La paire stéréoscopique considérée est donnée par $(I_\mathrm{L},I_\mathrm{R})$: ce sont des images en couleur RGB, dont les valeurs sur chaque canal sont comprises entre 0 et 255. Les caméras associées sont supposées en mouvement fronto-parallèle. La disparité u est donc une fonction définie sur Ω, à valeurs dans l'intervalle de disparité $\mathrm{I}_\mathrm{disp} \subset \mathbb{R}$, connu. On supposera dans un premier temps que u appartient à l'espace $\mathrm{W}^{1,2}(\Omega;\mathrm{I}_\mathrm{disp})$ des fonctions d'énergie finie, dérivables et dont la dérivée est intégrable sur Ω.

On adopte ici la méthode proposée par Pock et coll. dans [12]. Il s'agit d'une méthode globale : on définit une fonctionnelle d'énergie E sur l'ensemble des fonctions $\mathrm{W}^{1,2}(\Omega;\mathrm{I}_\mathrm{disp})$. Celle-ci encode les propriétés de la fonction de disparité que l'on recherche, en ce sens que $\mathrm{E}(u)$ sera d'autant plus grand que u ne satisfait pas ces propriétés. Si la fonctionnelle E est correctement choisie, la disparité de la scène la plus probable sera donnée par un minimiseur de E.

L'approche choisie par [12] est une approche variationnelle : la fonctionnelle d'énergie prend la forme

$$\mathrm{E}(u) = \int_\Omega f\Big(x,u(x),\nabla u(x)\Big)\,\mathrm{d}x.$$

Le lagrangien f est supposé continu en ses deux premières variables et convexe en sa troisième variable. Un choix particulier de f permet de séparer ces deux groupes de variables :

$$\forall\,(x,t) \in \Omega \times \mathbb{R}, \quad \forall\, p^x \in \mathbb{R}^2, \qquad f\Big(x,t,p^x\Big) = g\Big(x,t\Big) + r\Big(p^x\Big)$$

où g est continue sur $\Omega\times\mathbb{R}$ et r convexe. Ce choix conduit à considérer une fonctionnelle d'énergie à deux termes : un terme d'attache aux données (ou de fidélité)

$$\mathrm{E}_\mathrm{data}(u) = \int_\Omega g\Big(x,u(x)\Big)\,\mathrm{d}x$$

qui contraint l'algorithme à préférer mettre en correspondance des pixels semblables, et un terme de régularisation

$$\mathrm{E}_\mathrm{reg+vis}(u) = \int_\Omega r\Big(\nabla u(x)\Big)\,\mathrm{d}x$$

qui incite la fonction de disparité à être régulière. Nous intégrerons dans ce terme une contrainte de visibilité. Ces deux termes sont sommés et, quitte à remplacer g par g/μ, on peut faire apparaître un paramètre de pondération $\mu > 0$ qui permet de moduler l'importance relative des deux termes :

$$\mathrm{E}(u) = \mu\,\mathrm{E}_\mathrm{data}(u) + \mathrm{E}_\mathrm{reg+vis}(u).$$

L'objectif est de trouver un minimum de cette fonctionnelle, c'est-à-dire de résoudre le problème d'optimisation

$$\min_{u\in\mathrm{W}^{1,2}\left(\Omega;\mathrm{I}_\mathrm{disp}\right)} \mathrm{E}(u).$$

Détaillons dans ce qui suit le choix de chacun de ces deux termes.

3.1.1 Une corrélation adaptative pour limiter l'adhérence

Le rôle du terme d'attache aux données est de privilégier la mise en correspondance de pixels semblables. Il est donc naturel de le définir à l'aide d'une mesure de dissimilarité $D_{I_\mathrm{L},I_\mathrm{R}}$ (cf. section 2.3.1). Soit u une fonction de disparité candidate. Le pixel homologue de $x \in \Omega$ est alors le pixel $x - u(x)$ [1]. Le coût de corrélation pour cette mise en correspondance est donc donné par $g\big(x,u(x)\big)$, où g est défini par

$$\forall\,(x,t) \in \Omega \times \mathbb{R}, \qquad g(x,t) = D_{I_\mathrm{L},I_\mathrm{R}}(x,x-t)$$

Le coût total de corrélation pour la fonction u est donnée par l'intégrale de ces coûts individuels, ce qui nous permet de définir le terme de fidélité $\mathrm{E}_{\mathrm{data}}$.

Corrélation d'intensité combinée à la corrélation de gradient Puisque la combinaison de la corrélation d'intensité et la corrélation de gradient a montré son efficacité [8, 13], nous choisissons pour la mesure de dissimilarité une variante de la mesure utilisée dans [13, 14]. On rappelle que l'idée est de sommer la mesure AD et la corrélation des gradients dans une combinaison convexe, dont le coefficient α est choisi égal à 0,9 dans [13, 14]. L'intérêt est de tirer parti des performances des deux corrélations.

Malheureusement, l'utilisation d'un gradient revient essentiellement à comparer des voisinages. En effet, si le gradient de l'image est calculé à l'aide d'un opérateur linéaire de différences finies, alors il s'agit de comparer deux quantités $a(p,\{p'\})$ et $a(q,\{q'\})$, dépendant respectivement linéairement des voisins p' de p et des voisins q' de q. Or, on sait que la comparaison de voisinages provoque de l'adhérence près des discontinuités de scène. Nous proposons donc de modifier le poids de la corrélation de gradient dans le coût de corrélation, selon la position du pixel considéré : plus le pixel p est situé près d'une discontinuité de scène, moins la corrélation de gradient doit jouer. Cette approche implique de connaître la carte de disparité, ce qui est évidemment exclu. Une manière de procéder est donc de considérer les discontinuités *d'intensité*, qui, si elles ne coïncident pas avec les discontinuités de scène, les contiennent.

Coefficient variable Introduisons donc un coefficient variable $\alpha : \Omega \to [\,0\,;1\,]$, choisi de sorte que α soit petit près d'une discontinuité d'intensité et grand ailleurs. Une discontinuité d'intensité étant caractérisée par un fort gradient, nous définissons α comme une fonction décroissante de l'amplitude du gradient de l'image. Pour ne prendre en compte que des gradients significatifs, on effectue au préalable un débruitage ROF de l'image de référence (cf. chapitre 6), ce qui donne la version lissée $\tilde{I}_\mathrm{L}$. Le gradient au point x d'une image couleur $I = (I^\mathrm{r},I^\mathrm{g},I^\mathrm{b})$ est donné par une matrice de taille 2×3, définie par les dérivées horizontales et verticales (notées respectivement ∂_x et ∂_y) dans chacun des canaux couleur :

$$\forall\, x \in \Omega, \qquad \nabla I(x) = \begin{pmatrix} \partial_x I^\mathrm{r}(x) & \partial_x I^\mathrm{g}(x) & \partial_x I^\mathrm{b}(x) \\ \partial_y I^\mathrm{r}(x) & \partial_y I^\mathrm{g}(x) & \partial_y I^\mathrm{b}(x) \end{pmatrix}. \tag{3.1}$$

On peut alors définir la norme (ou l'amplitude) du gradient en x, notée $\|\nabla I(x)\|$, comme la norme euclidienne du vecteur de $\mathbb{R}^6$ associé au gradient (c'est-à-dire la norme de Frobenius de la matrice $\nabla I(x)$). Dans ce cas, α est défini par

$$\forall\, x \in \Omega, \qquad \alpha(x) = \frac{1}{1 + G \star \|\nabla \tilde{I}_\mathrm{L}\|^2(x)/a} \tag{3.2}$$

1. On note par abus de notation $x - t = x - {}^\mathrm{t}(t,0)$ pour tout réel t

avec $a > 0$ un paramètre que l'on choisira ultérieurement. L'amplitude du gradient est ici convolée avec une gaussienne, ce qui permet de diffuser dans un voisinage les forts gradients.

Corrélation de couleur et de gradient Pour la corrélation d'intensité, on choisit d'exploiter l'information contenue dans la couleur. Contrairement à ce qui est fait dans [13, 14], on utilise la distance induite par la norme euclidienne de $\mathbb{R}^3$, notée $\|\cdot\|$, pour comparer les vecteurs couleurs, ce qui conduit à considérer

$$\forall\,(x,t) \in \Omega \times \mathbb{R}, \qquad D^{\mathrm{AD}}_{I_\mathrm{L},I_\mathrm{R}}(x,x-t) = \|I_\mathrm{L}(x) - I_\mathrm{R}(x-t)\| \tag{3.3}$$

Pour la corrélation de gradient, on choisit à nouveau de considérer les variations horizontales et verticales de l'image dans les trois canaux couleurs donnée par la matrice (3.1), puis d'utiliser comme mesure de dissimilarité la distance issue de la norme de Frobenius :

$$\forall\,(x,t) \in \Omega \times \mathbb{R}, \qquad D^{\mathrm{grad}}_{I_\mathrm{L},I_\mathrm{R}}(x,x-t) = \|\nabla I_\mathrm{L}(x) - \nabla I_\mathrm{R}(x-t)\|. \tag{3.4}$$

Nouvelle mesure de dissimilarité Les deux coûts de corrélation (3.3) et (3.4) sont combinés à l'aide de la pondération variable α donnée par (3.2) :

$$\forall\,(x,t) \in \Omega \times \mathbb{R}, \qquad g(x,t) = \Big(1 - \alpha(x)\Big)\, D^{\mathrm{AD}}_{I_\mathrm{L},I_\mathrm{R}}(x,x-t) + \alpha(x)\, D^{\mathrm{grad}}_{I_\mathrm{L},I_\mathrm{R}}(x,x-t).$$

Même si la fonction g n'est théoriquement pas continue en (x,t) (car les images considérées ne le sont pas), il est possible de se ramener artificiellement à ce cas en considérant une interpolation régulière de cette fonction, qui est en pratique construite à partir d'un échantillonnage des images et de l'intervalle de disparité.

3.1.2 Une contrainte de pente pour gérer l'occultation

Le terme de régularisation $\mathrm{E}_{\mathrm{reg+vis}}$ que nous proposons possède en réalité deux composantes distinctes, une première forçant la régularisation proprement dite de la disparité et une seconde traduisant une contrainte de visibilité :

$$\mathrm{E}_{\mathrm{reg+vis}}(u) = \mathrm{E}_{\mathrm{reg}}(u) + \mathrm{E}_{\mathrm{vis}}(u).$$

Terme de régularisation On choisit d'utiliser la régularisation TV (variation totale), dont nous considérons ici la version isotrope :

$$\mathrm{E}^{\text{TV iso}}_{\mathrm{reg}}(u) = \int_\Omega \|\nabla u(x)\|\,\mathrm{d}x = \|\nabla u\|_1\,{}^{2}.$$

Cette régularisation a été introduite en 1992 par Rudin, Osher et Fatemi dans un modèle de débruitage qui porte depuis le nom de *modèle ROF*. L'intérêt principal de ce terme est le suivant : si on étend l'ensemble des fonctions admissibles u à un espace plus grand (voir le paragraphe **Extension aux fonctions BV**), alors ce terme produit des solutions *régulières*, tout en *préservant certaines discontinuités* [3]. Généralement, suivant le poids accordé à ce terme, les solutions sont constantes par morceaux.

2. Cette notation, bien qu'usuelle, est abusive dans le cas d'images multi-valuées. La variation totale, telle qu'elle est définie ici, correspond en réalité à la norme L^1 de la fonction $u \mapsto \|\nabla u(x)\|_2$. Néanmoins, pour éviter les notations trop lourdes, nous continuerons ici d'utiliser la notation standard.

Terme de visibilité Dans l'analyse proposée à la section 2.2, on a montré que, si l'ordre était préservé dans les images, alors la largeur de l'occultation dans l'image de référence est exactement égale au saut de disparité autour de l'occultation. Par ailleurs, nous avons établi que la disparité ne peut pas avoir une pente horizontale excédant 1 sans violer les contraintes de visibilité. De ces deux remarques, on déduit que, si on interpole les occultations de manière affine sur la ligne, alors la pente de la disparité interpolée vaut exactement 1 dans les zones occultées et est strictement inférieur à 1 dans les zones non occultées. Cela nous conduit à introduire le terme de visibilité suivant :

$$\mathrm{E}_{\mathrm{vis}}(u) = \int_{\Omega} \chi_{\{\partial_x u \leq 1\}}(x)\,\mathrm{d}x = \begin{cases} 0 & \text{si } \forall\, x \in \Omega, \quad \partial_x u(x) \leq 1 \\ +\infty & \text{sinon.} \end{cases}$$

qui contraint la pente horizontale de la disparité à être inférieure à 1.

Finalement, le terme de régularisation/visibilité peut s'écrire à l'aide de la fonction

$$\forall\, p^x = (p^x_x, p^x_y) \in \mathbb{R}^2, \qquad r(p^x) = \begin{cases} \|p^x\| & p^x_x \leq 1 \\ +\infty & \text{sinon.} \end{cases}$$

On vérifie aisément que la fonction r est bien convexe.

Extension aux fonctions BV Les fonctions de l'espace $\mathrm{W}^{1,2}(\Omega;\mathbb{R})$ ne peuvent présenter aucune discontinuité le long d'une ligne [3] : en d'autres termes, de telles fonctions ne peuvent pas modéliser des contours d'objets. Or, la scène étant composée d'objets distincts, les images et la carte de disparité présentent de telles ruptures de discontinuités (appelées *bords* ou *contours*). L'étude menée dans 2.2 assure qu'elles correspondent, dans la disparité, à des désocculatations. C'est pourquoi il est intéressant d'étendre le domaine de définition de la fonctionnelle E à l'ensemble $\mathrm{BV}(\Omega;\mathrm{I}_{\mathrm{disp}})$ des fonctions à variations bornées (voir par exemple [7] pour plus de précisions sur l'espace BV).

Le terme de fidélité reste bien défini pour ces fonctions. En revanche, le terme de régularisation/visibilité n'est *a priori* pas défini dans le cas où u n'est pas différentiable. On remarque alors que E_{vis} est une fonction convexe de ∇u, ce qui permet, grâce à [2], de l'appliquer à la mesure de Radon Du associée aux variations de u.

Nous sommes donc à présent amenés à étudier le problème

$$\min_{u \in \mathrm{BV}\left(\Omega;\mathrm{I}_{\mathrm{disp}}\right)} \left\{ \int_{\Omega} g\Big(x, u(x)\Big)\,\mathrm{d}x + \int_{\Omega} r\Big(Du\Big) \right\}. \tag{3.5}$$

3.2 Relaxation convexe du problème initial

La fonctionnelle d'énergie E n'est malheureusement pas convexe. L'idée est donc de transformer le problème initial non convexe (3.5) en un problème convexe, d'abord en transformant la fonctionnelle d'énergie E en une énergie d'interface convexe, puis en relaxant le problème. On verra (théorème 10) que les solutions du problème relaxé convexe et celles du problème initial non convexe sont reliées, si bien qu'il suffit de résoudre le problème relaxé pour obtenir des solutions du problème initial.

Nous allons reprendre les différentes étapes présentées dans [12], auquel le lecteur pourra se reporter pour les différentes démonstrations (non reproduites ici).

3.2.1 Énergie d'interface

Indicatrice du sous-graphe Commençons par montrer qu'il est possible de représenter de manière unique toute fonction BV par l'indicatrice d'une partie de $\mathbb{R}^3$, son *sous-graphe*. Définissons le sous-graphe d'une fonction u. Si u est une fonction à valeurs réelles définie sur l'ouvert Ω, alors on définit son sous-graphe comme l'ensemble des points $(x,t) \in \Omega \times \mathbb{R}$ vérifiant l'inégalité $u(x) \geq t$. On peut en définir l'indicatrice, notée $\mathbb{1}_u$, définie sur $\Omega \times \mathbb{R}$ et à valeurs dans $\{0,1\}$, par

$$\forall\,(x,t) \in \Omega \times \mathbb{R}, \qquad \mathbb{1}_u(x,t) = \begin{cases} 1 & \text{si } u(x) \geq t \\ 0 & \text{si } u(x) < t. \end{cases}$$

On notera que, pour tout $x \in \Omega$, la fonction $t \mapsto \mathbb{1}_u(x,t)$ est une fonction constante par morceaux, décroissante de 1 vers 0. On observe alors que

$$\forall\, x \in \Omega, \qquad u(x) = \sup\left\{ t \in \mathbb{R} \;\middle|\; \mathbb{1}_u(x,t) = 1 \right\}. \tag{3.6}$$

Autrement dit, il est suffisant de connaître $\mathbb{1}_u$ pour connaître entièrement la fonction u.

Cette remarque permet de motiver un changement de variable dans le problème (3.5). Au lieu de chercher à minimiser la fonctionnelle E définie sur la fonction u, on va minimiser une nouvelle énergie F définie sur l'indicatrice $\mathbb{1}_u$. Pour que les deux problèmes restent équivalents, il faut choisir F telle que $\mathrm{F}(\mathbb{1}_u) = \mathrm{E}(u)$ pour tout $u \in \mathrm{BV}\,(\Omega;\mathbb{R})$.

Notons que, puisque u est supposée à valeurs dans $\mathrm{I}_{\mathrm{disp}}$, alors pour tout $t > \max(\mathrm{I}_{\mathrm{disp}})$, le couple (x,t) n'appartient pas au sous-graphe de u, ce qui implique que $\mathbb{1}_u(x,t) = 0$. De même, pour tout $t < \min(\mathrm{I}_{\mathrm{disp}})$, on a $\mathbb{1}_u(x,t) = 1$.

Nouvelle fonctionnelle d'énergie Introduisons [12] le lagrangien h suivant, défini pour tout $(x,t) \in \Omega \times \mathbb{R}$ et tout $p = (p^x,p^t) \in \mathbb{R}^2 \times \mathbb{R}$ par :

$$h(x,t,p) = \begin{cases} |p^t|\, g(x,t) + |p^t|\, r\left(p^x/|p^t|\right) & \text{si } p^t < 0 \\ \lim\limits_{\lambda \to 0^+} \lambda\, r\,(p^x/\lambda) & \text{si } p^t = 0 \\ +\infty & \text{si } p^t > 0 \end{cases}$$

qui se réécrit ici

$$h(x,t,p) = \begin{cases} |p^t|\, g(x,t) + r(p^x) & \text{si } p^t \leq 0 \text{ et } p^x_x \leq -p^t \\ +\infty & \text{si } p^t > 0 \text{ ou } p^x_x > -p^t. \end{cases}$$

On peut décomposer ce lagrangien en deux termes h^{TV} et h^{vis}. Le premier terme est donné par

$$h^{\mathrm{TV}}(x,t,p) = \begin{cases} |p^t|\, g(x,t) + \|p^x\| & \text{si } p^t \leq 0 \\ +\infty & \text{si } p^t > 0 \end{cases}$$

et correspond au lagrangien associé à la régularisation TV sans contrainte de visibilité. Le second terme ne dépend que de la variable p et vaut

$$h^{\mathrm{vis}}(p) = \begin{cases} 0 & \text{si } p^x_x \leq -p^t \\ +\infty & \text{si } p^x_x > -p^t. \end{cases}$$

Dans ce terme, qui est convexe, est encodée la contrainte de visibilité. Si on pose pour tout $v \in \mathrm{BV}(\Omega \times \mathbb{R}; [0;1])$

$$\mathrm{F}(v) = \int_{\Omega\times\mathbb{R}} h(x,t,Dv) = \int_{\Omega\times\mathbb{R}} h^{\mathrm{TV}}(x,t,Dv) + \int_{\Omega\times\mathbb{R}} h^{\mathrm{vis}}(Dv) \tag{3.7}$$

alors $\mathrm{F}(\mathbb{1}_u) = \mathrm{E}(u)$ quelle que soit la fonction u [12]. En effet, notons que, si u est à variation bornée, alors l'indicatrice de son sous-graphe $\mathbb{1}_u$ est dans $\mathrm{BV}(\Omega \times \mathbb{R}; [0;1])$. En d'autres termes, le sous-graphe de toute fonction BV est à périmètre fini dans $\mathbb{R}^3$. Le théorème de Gauss-Green assure alors que

$$D\mathbb{1}_u = \nu_{\Gamma_u}(x,t)\ \mathrm{d}\mathcal{H}^2(x,t) \llcorner \Gamma_u$$

où on note Γ_u la frontière du sous-graphe de u [3] et ν_{Γ_u} la normale extérieure au sous-graphe de u, définie par

$$\forall\,(x,t) \in \Gamma_u, \qquad \nu_{\Gamma_u}(x,t) = \begin{pmatrix} \nabla u(x) \\ -1 \end{pmatrix}$$

si $u \in \mathrm{W}^{1,1}(\Omega;\mathbb{R})$, et $\mathcal{H}^2$ est la mesure de Haussdorff [17, chapitre V] de dimension 2. On en déduit l'identité souhaitée ($\llcorner$ symbolisant la restriction).

On notera que F est une fonctionnelle convexe. Par ailleurs, $\mathrm{F}(u)$ est s'écrit en fonction de la frontière du sous-graphe de u. C'est pourquoi on parle d'énergie d'*interface*.

Problème équivalent Le problème initial (3.5) est donc équivalent au problème

$$\min_{u\in\mathrm{BV}\left(\Omega;\mathrm{I}_{\mathrm{disp}}\right)} \left\{ \int_{\Omega\times\mathbb{R}} h(x,t,D\mathbb{1}_u) \right\} \tag{3.8}$$

qui s'écrit encore $\displaystyle\min_{\substack{v\in\mathrm{BV}(\Omega\times\mathbb{R};[0;1]) \\ v=\mathbb{1}_u, u\in\mathrm{BV}\left(\Omega;\mathrm{I}_{\mathrm{disp}}\right)}} \left\{ \int_{\Omega\times\mathbb{R}} h(x,t,Dv) \right\}.$

Le problème (3.8) est donc un problème de minimisation de l'énergie convexe F sur l'ensemble non convexe des fonctions qui sont indicatrices du sous-graphe d'une fonction à variation bornée de Ω.

3.2.2 Relaxation convexe

Pour rendre ce problème convexe, il suffit de procéder à une relaxation convexe, c'est-à-dire de remplacer l'ensemble des fonctions admissibles par un ensemble plus grand, mais convexe. Ainsi, le problème considéré devient la minimisation d'une énergie convexe sur un ensemble de fonctions admissibles convexes, c'est-à-dire un problème d'optimisation convexe. On pourra alors exploiter les outils de l'analyse convexe, qui assurent en particulier que le problème possède un minimum global.

Une manière de procéder est de considérer l'enveloppe convexe de l'ensemble des fonctions $\{v = \mathbb{1}_u \mid u \in \mathrm{BV}(\Omega;\mathbb{R})\}$, qui est l'ensemble des fonctions de $\mathrm{BV}(\Omega \times \mathbb{R}; [0;1])$ décroissantes selon leur seconde variable. On va toutefois suivre la démarche proposée par [12] et choisir une relaxation plus importante, en considérant l'ensemble suivant

$$\mathcal{C} = \left\{ v \in \mathrm{BV}(\Omega \times \mathbb{R}; [0;1]) \;\middle|\; \forall\, x \in \Omega, \begin{array}{l} \forall\, t \le \min(\mathrm{I}_{\mathrm{disp}}),\ v(x,t) = 1 \\ \forall\, t \ge \max(\mathrm{I}_{\mathrm{disp}}),\ v(x,t) = 0 \end{array} \right\} \tag{3.9}$$

3. Il s'agit du graphe de u quand u est continue.

qui est convexe et contient bien l'ensemble des indicatrices de sous-graphe de fonctions BV. Cette relaxation conduit donc à considérer le problème convexe

$$\min_{v\in\mathcal{C}} \left\{ \int_{\Omega\times\mathbb{R}} h(x,t,Dv) \right\}. \tag{3.10}$$

Quel est alors le lien entre les solutions du problème convexe (3.10) et celles (si elles existent) du problème initial (3.5) ? C'est l'objet du théorème suivant :

Théorème 10 ([12]) *Soit v^* un minimiseur global du problème relaxé (3.10). Alors, pour tout $s \in [\,0\,;1\,[$, la fonction caractéristique $\mathbb{1}_{\{v^*>s\}}$ est l'indicatrice du sous-graphe d'une fonction u^*, qui est un minimiseur du problème initial (3.5).*

Autrement dit, pour obtenir un minimiseur du problème initial, il suffit de résoudre le problème relaxé, puis d'en seuiller la solution avec n'importe quel $s \in [\,0\,;1\,[$. À l'aide de la formule de reconstruction (3.6), on obtient une solution du problème initial.

3.2.3 Formulation primale-duale

Le théorème 10 assurant que des minimiseurs globaux de la fonctionnelle E peuvent être obtenus à partir de minimiseurs globaux de l'énergie d'interface F sur le convexe $\mathcal{C}$, on s'attache à présent à résoudre le problème relaxé (3.10). On montre pour cela qu'il est possible de l'écrire sous une forme primale-duale.

Soit $v \in \mathcal{C}$. D'après le théorème 3.2 de [12], on a

$$\int_{\Omega\times\mathbb{R}} h^{\mathrm{TV}}(x,t,Dv) = \sup_{\phi\in\mathcal{K}} \int_{\Omega\times\mathbb{R}} \phi\, Dv,$$

où $\mathcal{K}$ est le convexe défini par

$$\mathcal{K} = \left\{ \phi \in \mathcal{C}^0\left(\Omega\times\mathbb{R};\mathbb{R}^3\right) \;\middle|\; \forall\,(x,t)\in\Omega\times\mathbb{R}, \phi^t(x,t) + \mu\, g(x,t) \geq 0 \text{ et } \|\phi^x(x,t)\| \leq 1 \right\}$$

avec $\phi = (\phi^x,\phi^t)$. On est donc amené à résoudre le problème de recherche de point-selle suivant

$$\min_{v\in\mathcal{C}} \sup_{\phi\in\mathcal{K}} \left\{ \int_{\Omega\times\mathbb{R}} \phi\, Dv + \int_{\Omega\times\mathbb{R}} h^{\mathrm{vis}}(Dv) \right\}. \tag{3.11}$$

3.3 Résolution numérique par algorithme primal-dual

Pour résoudre numériquement le problème primal-dual (3.11), en vue d'obtenir des solutions du problème initial (3.5), on commence par discrétiser les données du problème primal-dual. Nous proposerons ensuite deux algorithmes pour résoudre le problème discret qui en découle.

3.3.1 Discrétisation du problème

Images Notons I^h_{L} et I^h_{R} les deux images numériques de la paire stéréoscopique. Dans le cas d'images en niveaux de gris, ces deux images sont deux matrices de taille $N_x \times N_y$ et dans le cas d'images couleur RGB, on les représente comme deux triplets de matrices

de taille $N_x \times N_y$. On suppose que le domaine de l'image continue I_L est rectangulaire, donné par $\Omega =]\,0\,;A_x\,[\times]\,0\,;A_y\,[$. Sa discrétisation est alors donnée par la grille régulière

$$\Omega^h = \left\{ (i\,h_x, j\,h_y) \;\middle|\; (i,j) \in [\![\,0\,;\,N_x-1\,]\!] \times [\![\,0\,;\,N_y-1\,]\!] \right\}$$

avec $N_x = \lfloor A_x/h_x \rfloor + 1$ et $N_y = \lfloor A_y/h_y \rfloor + 1$ deux entiers positifs, correspondant respectivement à la largeur (horizontale) et à la hauteur (verticale) de l'image en pixels. Les deux réels positifs h_x et h_y sont les pas d'échantillonnage (horizontal et vertical) des images. Les pas de discrétisation de la grille d'échantillonnage sont désormais choisis égaux à $h_x = h_y = 1$, ce qui signifie que les images ne sont ni sous-échantillonnées ni sur-échantillonnées.

Soit $(i,j) \in [\![\,0\,;\,N_x-1\,]\!] \times [\![\,0\,;\,N_y-1\,]\!]$. Le pixel d'indice (i,j) de l'image de référence est donné par le coefficient (ou le vecteur, dans le cas d'image couleur) $(I_L^h)_{i,j}$, et, dans une première approximation, on supposera que $(I_L^h)_{i,j} = I_L(i,j)$ [4] (et de même pour l'image de droite).

Intervalle de disparité On discrétise ensuite l'intervalle de disparité $\mathrm{I}_{\mathrm{disp}} = [\,d_{\min}\,;d_{\max}\,]$:

$$\mathrm{I}_{\mathrm{disp}}^h = \left\{ d_{\min} + k\,h_t \;\middle|\; k \in [\![\,0\,;\,N_t-1\,]\!] \right\}$$

où l'entier naturel $N_t = \lfloor (d_{\max} - d_{\min})/h_t \rfloor + 1$ correspond au nombre de profondeurs différentes que compte la carte du relief recherchée et h_t est le pas de quantification de la carte de disparité. On note alors $h = (1,1,h_t)$ le vecteur des pas d'échantillonnage et $G^h = \Omega^h \times \mathrm{I}_{\mathrm{disp}}^h$ la grille d'échantillonnage du volume $\Omega \times \mathrm{I}_{\mathrm{disp}}$.

Terme d'attache aux données On introduit ensuite le *volume de coût* g^h, défini par

$$\forall\,(i,j,k\,h_t) \in G^h, \qquad g^h_{i,j,k} = g\Big((i,j),k\,h_t\Big).$$

Le volume de coût est un volume 3D tel que $g^h_{i,j,k}$ donne pour le pixel (i,j) le coût de corrélation associé à la disparité $k\,h_t$.

Dans le cas d'un pas unitaire dans la grille d'échantillonnage des images, le coefficient $g^h_{i,j,k}$ compare le pixel $(i,j) \in \mathbb{N}^2$ de l'image de référence avec le pixel $(i,j-k\,h_t)$ de l'image de droite. Supposons que la corrélation choisie est la corrélation d'intensité. Si $(i,j-k\,h_t)$ appartient à la grille d'échantillonnage de l'image de droite (c'est-à-dire si $k\,h_t$ est un entier), alors l'intensité de l'image en ce point est connue, et le coût $g^h_{i,j,k}$ est une fonction de $(I_L^h)_{i,j}$ et $(I_R^h)_{i,j-k\,h_t}$. Si $(i,j-k\,h_t)$ n'appartient pas à la grille d'échantillonnage de l'image de droite, alors une étape de sur-échantillonnage de l'image de droite est nécessaire pour interpoler l'intensité de I_R^h au point $(i,j-k\,h_t)$. L'interpolation que nous choisissons est l'interpolation B-spline d'ordre 5 [16, 15].

Convexes On peut ensuite discrétiser les convexes $\mathcal{C}$ et $\mathcal{K}$, en posant

$$\mathcal{C}^h = \left\{ v^h \in [\,0\,;1\,]^{N_x N_y N_t} \;\middle|\; \forall\,(i,j) \in \Omega^h, v^h_{i,j,0} = 1 \text{ et } v^h_{i,j,N_t-1} = 0 \right\}$$

(suite à la remarque, soulignée au paragraphe 3.2.1, concernant les valeurs prises par l'indicatrice $\mathbb{1}_u$ lorsque t est supérieur à $\max(\mathrm{I}_{\mathrm{disp}})$ et lorsque t est inférieur à $\min(\mathrm{I}_{\mathrm{disp}})$), et

$$\mathcal{K}^h = \left\{ \phi^h \in \mathbb{R}^{3N_xN_yN_t} \;\middle|\; \forall\,(i,j,k\,h_t) \in G^h, (\phi^h)^t_{i,j,k} + \mu g^h_{i,j,k} \geq 0 \text{ et } \left\| \left((\phi^h)^x_{i,j,k}, (\phi^h)^y_{i,j,k} \right) \right\| \leq 1 \right\}$$

4. En réalité, cette égalité est simpliste, car l'échantillonnage implique une étape préalable de filtrage de l'image, qui correspond à un filtre *anti-aliasing*, ou anti-recouvrement de spectre.

où on note cette fois $\phi^h_{i,j,k} = ((\phi^h)^x_{i,j,k},(\phi^h)^y_{i,j,k},(\phi^h)^t_{i,j,k}) \in \mathbb{R}^3$.

Opérateurs gradient et divergence On choisit pour la version discrète de l'opérateur gradient un opérateur linéaire $\nabla^h : \mathbb{R}^{N_xN_yN_t} \to (\mathbb{R}^3)^{N_xN_yN_t}$ construit à l'aide de différences finies avec des conditions de bord de type Neumann : pour tout $v^h \in \mathbb{R}^{N_xN_yN_t}$,

$$\forall\,(i,j,k\,h_t) \in G^h, \qquad (\nabla^h v^h)_{i,j,k} = \begin{pmatrix} (\delta^h_x v^h)_{i,j,k} \\ (\delta^h_y v^h)_{i,j,k} \\ (\delta^h_t v^h)_{i,j,k} \end{pmatrix}$$

où les différences finies $(\delta^h_x v^h)_{i,j,k}$, $(\delta^h_y v^h)_{i,j,k}$ et $(\delta^h_t v^h)_{i,j,k}$ sont données pour tout indice (i,j,k) par

$$(\delta^h_x v^h)_{i,j,k} = \begin{cases} v^h_{i+1,j,k} - v^h_{i,j,k} & \text{si } i < N_x - 1 \\ 0 & \text{si } i = N_x - 1 \end{cases}$$

$$(\delta^h_y v^h)_{i,j,k} = \begin{cases} v^h_{i,j+1,k} - v^h_{i,j,k} & \text{si } j < N_y - 1 \\ 0 & \text{si } j = N_y - 1 \end{cases}$$

et

$$(\delta^h_t v^h)_{i,j,k} = \begin{cases} \dfrac{v^h_{i,j,k+1} - v^h_{i,j,k}}{h_t} & \text{si } k < N_t - 1 \\ 0 & \text{si } k = N_t - 1. \end{cases}$$

Cet opérateur admet un adjoint, noté div^h, donné pour tout $\phi^h \in \mathbb{R}^{3N_xN_yN_t}$ par

$$\forall\,(i,j,k\,h_t) \in G^h, \qquad (\mathrm{div}^h\, v^h)_{i,j,k} = (\varepsilon^h_x(\phi^h)^x)_{i,j,k} + (\varepsilon^h_y(\phi^h)^y)_{i,j,k} + (\varepsilon^h_t(\phi^h)^t)_{i,j,k}$$

où les différences finies $(\varepsilon^h_x(\phi^h)^x)_{i,j,k}$, $(\varepsilon^h_y(\phi^h)^y)_{i,j,k}$ et $(\varepsilon^h_t(\phi^h)^t)_{i,j,k}$ sont données pour tout indice (i,j,k) par

$$(\varepsilon^h_x(\phi^h)^x)_{i,j,k} = \begin{cases} -(\phi^h)^x_{i,j,k} & \text{si } i = 0 \\ (\phi^h)^x_{i-1,j,k} - (\phi^h)^x_{i,j,k} & \text{si } 0 < i < N_x - 1 \\ (\phi^h)^x_{i-1,j,k} & \text{si } i = N_x - 1 \end{cases}$$

$$(\varepsilon^h_y(\phi^h)^y)_{i,j,k} = \begin{cases} -(\phi^h)^y_{i,j,k} & \text{si } j = 0 \\ (\phi^h)^y_{i,j-1,k} - (\phi^h)^y_{i,j,k} & \text{si } 0 < j < N_y - 1 \\ (\phi^h)^y_{i,j-1,k} & \text{si } j = N_y - 1 \end{cases}$$

et

$$(\varepsilon^h_t(\phi^h)^t)_{i,j,k} = \begin{cases} -\dfrac{(\phi^h)^t_{i,j,k}}{h_t} & \text{si } k = 0 \\ \dfrac{(\phi^h)^x_{i-1,j,k} - (\phi^h)^x_{i,j,k}}{h_t} & \text{si } 0 < k < N_t - 1 \\ \dfrac{(\phi^h)^x_{i-1,j,k}}{h_t} & \text{si } k = N_t - 1. \end{cases}$$

Problème discret Le problème primal-dual devient alors le problème de recherche de point-selle

$$\min_{v^h \in \mathcal{C}^h} \sup_{\phi^h \in \mathcal{K}^h} \left\{ \langle \phi^h, \nabla^h v^h \rangle + \sum_{(i,j,kh_t) \in G^h} \chi_{]-\infty;0]} \left((\delta^h_x v^h)_{i,j,k} + (\delta^h_t v^h)_{i,j,k} \right) \right\} \tag{3.12}$$

où $\chi_{]-\infty;0]}$ est la fonction caractéristique de l'intervalle $]-\infty\,;0\,]$. Elle vaut 0 sur cet intervalle et vaut une valeur infinie sur $]\,0\,;+\infty\,[$. On peut alors introduire un multiplicateur de Lagrange supplémentaire pour le second terme, car, pour tout $x \in \mathbb{R}$, on a

$$\chi_{]-\infty;0]}(x) = \sup_{\lambda \geq 0} \lambda\, x.$$

Le problème (3.12) se réécrit alors

$$\min_{v^h \in \mathcal{C}^h} \sup_{\substack{\phi^h \in \mathcal{K}^h \\ \lambda^h \geq 0}} \left\{ \langle \phi^h, \nabla^h v^h \rangle + \langle \lambda^h, \delta_x^h v^h + \delta_t^h v^h \rangle \right\} \tag{3.13}$$

où, pour simplifier les notations, on écrira $\lambda^h \geq 0$ à la place de $\lambda^h \in ([\,0\,;+\infty\,[)^{N_x N_y N_t}$.

3.3.2 Algorithme primal-dual

On peut résoudre le problème de recherche de point-selle (3.13) en utilisant un algorithme primal-dual analogue à celui proposé dans [12], mais accéléré selon [5].

Algorithme primal-dual sur-relaxé On commence par initialiser l'algorithme avec les variables $(v^h)_0 = (\bar{v}^h)_0 \in \mathcal{C}^h$, $(\phi^h)_0 \in \mathcal{K}^h$ et $(\lambda^h)_0 \geq 0$, puis on effectue pour tout $n \in \mathbb{N}$ les mises-à-jours suivantes

$$\begin{cases} (\hat{\phi}^h)_{n+1} &= \mathrm{proj}_{\mathcal{K}^h}\left((\phi^h)_n + \sigma\, \nabla^h (\bar{v}^h)_n\right) \\ (\hat{\lambda}^h)_{n+1} &= \mathrm{proj}_{([\,0\,;+\infty\,[)^{N_x N_y N_t}}\left((\lambda^h)_n + \sigma\, (\delta_x^h + \delta_t^h)(\bar{v}^h)_n\right) \\ (\hat{v}^h)_{n+1} &= \mathrm{proj}_{\mathcal{C}^h}\left((v^h)_n - \tau\, (\mathrm{div}^h (\phi^h)_{n+1} + (\varepsilon_x^h + \varepsilon_t^h)(\lambda^h)_{n+1})\right) \\ (\bar{v}^h)_{n+1} &= 2\,(\hat{v}^h)_{n+1} - (v^h)_n \end{cases}$$

où $\mathrm{proj}_{\mathcal{K}^h}$ (resp. $\mathrm{proj}_{([\,0\,;+\infty\,[)^{N_x N_y N_t}}$ et $\mathrm{proj}_{\mathcal{C}^h}$) désigne la projection sur le convexe $\mathcal{K}^h$ (resp. $([\,0\,;+\infty\,[)^{N_x N_y N_t}$ et $\mathcal{C}^h$). On ajoute ensuite une étape de sur-relaxation globale :

$$\begin{pmatrix} (v^h)_{n+1} \\ (\phi^h)_{n+1} \\ (\lambda^h)_{n+1} \end{pmatrix} = (1-\rho) \begin{pmatrix} (v^h)_n \\ (\phi^h)_n \\ (\lambda^h)_n \end{pmatrix} + \rho \begin{pmatrix} (\hat{v}^h)_{n+1} \\ (\hat{\phi}^h)_{n+1} \\ (\hat{\lambda}^h)_{n+1} \end{pmatrix}$$

de paramètre $\rho \in]\,0\,;2\,[$.

Il s'agit d'un algorithme effectuant dans un premier temps un pas de montée de gradient projeté en les variables duales (ϕ^h, λ^h) et un pas de descente de gradient projeté en la variable primale v^h, suivis d'une première sur-relaxation en v^h. Ensuite, une étape de sur-relaxation globale est réalisée sur les trois variables. On montre [4] que cet algorithme converge vers une solution du problème discret (3.13) sur les paramètres sont correctement choisis. Lorsque le paramètre de sur-relaxation ρ vaut 1, on retrouve l'algorithme proposé dans [11]. Lorsqu'il est proche de 2, on observe en pratique une accélération de la convergence, mais qui n'est pas expliquée de manière théorique.

Choix des paramètres Les pas de temps σ et τ doivent être choisis de sorte que $\sigma\tau L^2 < 1$, avec L la norme de l'opérateur $A = \nabla^h + \delta_x^h + \delta_t^h$. On montre de la même manière que dans [12] que L est majoré par $2\sqrt{4 + 1/h_t^2 + 1/h_t^4}$. On choisira par exemple $\tau = 0{,}1$ et $\sigma = 1/(\tau L^2)$. Par ailleurs, on choisira $\rho = 1{,}95$.

Calcul des projections Les trois projections en jeu dans cet algorithme peuvent se calculer de manière indépendante en chaque indice (i,j,k). La projection sur $\mathcal{K}^h$ est équivalente à une projection sur la boule unité de $\mathbb{R}^2$ [12], tandis que la projection sur $\mathcal{C}^h$ devient, suivant l'indice, une projection sur 0, 1 ou le segment $[\,0\,;1\,]$. Enfin, la projection sur $([\,0\,;+\infty\,[)^{N_xN_yN_t}$ est immédiate.

Remarque : C'est la simplicité de ces projections qui a principalement motivé l'introduction du multiplicateur de Lagrange λ^h. On pouvait en effet appliquer directement le résultat démontré dans [12] au lagrangien h, au lieu de le décomposer comme nous l'avons fait. Malheureusement, une telle démarche conduit à introduire un autre convexe $\mathcal{K}^h$, dont la forme plus complexe rend la projection difficile. De même, c'est la raison pour laquelle la relaxation choisie n'est pas celle où $\mathcal{C}$ est l'enveloppe convexe des fonctions admissibles du problème (3.10), car elle aurait conduit à une projection sur l'ensemble $\mathcal{C}^h$ des vecteurs décroissants selon la troisième dimension (à ce sujet, voir le pseudo-code et les références dans [6]).

Convergence Le thérorème 2 de [11] assure la convergence de l'algorithme proposé au début de ce paragraphe, mais ne donne aucune information quant au nombre d'itérations nécessaires pour s'approcher de la convergence.

Un outil généralement utilisé pour mesurer cette convergence est le *primal-dual gap*, défini pour le problème (3.13) et pour toute itération $n \in \mathbb{N}$ comme suit :

$$\begin{aligned}\mathcal{G}\Big((v^h)_n,(\phi^h)_n,(\lambda^h)_n\Big) = &\sup_{\substack{\phi^h\in\mathcal{K}^h\\ \lambda^h\geq 0}} \Big\{\langle\phi^h,\nabla^h(v^h)_n\rangle + \langle\lambda^h,\delta_x^h(v^h)_n + \delta_t^h(v^h)_n\rangle\Big\}\\ &- \min_{v^h\in\mathcal{C}^h}\Big\{\langle(\phi^h)_n,\nabla^h v^h\rangle + \langle(\lambda^h)_n,\delta_x^h v^h + \delta_t^h v^h\rangle\Big\}.\end{aligned}$$

Cette quantité est positive et tend vers 0 lorsque n tend vers $+\infty$. On peut alors s'en servir comme d'un critère d'arrêt de l'algorithme. Une première façon de procéder est de fixer une valeur de tolérance, et de stopper les itérations dès que le *gap* tombe endessous de cette valeur. Une autre démarche [12] consiste à arrêter l'algorithme dès que la valeur du *gap* a été divisée par une certaine valeur (1000 par exemple).

Malheureusement, dans notre cas précis, l'utilisation du *gap* n'est pas toujours appropriée. Les volumes $(v^h)_n$ n'étant pas explicitement contraints à satisfaire la condition

$$\forall\,(i,j,k\,h_t) \in G^h, \qquad (\delta_x^h(v^h)_n)_{i,j,k} + (\delta_t^h(v^h)_n)_{i,j,k} \leq 0$$

le premier terme dans $\mathcal{G}\Big((v^h)_n,(\phi^h)_n,(\lambda^h)_n\Big)$ prend généralement une valeur infinie, quel que soit le nombre de coefficients qui ne respectent pas cette contrainte. On peut alors soit ignorer ce terme dans le calcul du *gap*, soit affecter une pénalité (finie) à chaque coefficient qui ne satisfait pas cette majoration.

Seuillage de la solution Pour obtenir la carte de disparité u^h à partir de la solution v^h obtenue après N itérations de l'algorithme, on applique le théorème 10. On réalise un seuillage pour obtenir un niveau $s \in [\,0\,;1\,[$ de v^h , puis on reconstruit u^h à l'aide de la version discrète de la formule (3.6) :

$$\forall\,(i,j) \in \Omega^h, \qquad u_{i,j}^h = h_t\,\sup\Big\{k \in [\![\,0\,;\,N_t-1\,]\!] \;\Big|\; v_{i,j,k}^h > s\Big\}.$$

3.3.3 Accélération par convexification

Retour sur le problème continu On rappelle qu'on cherche à minimiser sur le convexe $\mathcal{C}$ la fonctionnelle F donnée par la formule (3.7). Le théorème 10 assure même qu'il suffit de trouver un ensemble de niveau d'un minimiseur v^* de F. Par ailleurs, la preuve du théorème 3.1 de [12] assure que, pour presque tout $s \in [\,0\,;1\,[$, l'indicatrice $\mathbb{1}_{\{v^*>s\}} \in \mathcal{C}$ de cet ensemble de niveau est elle-même minimiseur de la fonctionnelle F. En d'autres termes, l'ensemble $\{v^* > s\}$ est solution du problème convexe

$$\min_{\substack{E\subset\Omega\times\mathbb{R}\\ E \text{ satisfaisant } (\star)}} F(\mathbb{1}_E)$$

avec $\quad \Omega \times]-\infty\,;\min(\mathrm{I_{disp}})\,] \subset E \quad$ et $\quad \Omega \times [\,\max(\mathrm{I_{disp}})\,;+\infty\,[\,\cap E = \emptyset. \qquad (\star)$

Nous allons montrer que les solutions de ce problème sont liées à la solution d'un problème fortement convexe, que l'on cherchera donc à résoudre.

Formule de la co-aire D'après la preuve du théorème 3.1 de [12], la fonctionnelle F satisfait la *formule de la co-aire* : pour tout $v \in \mathrm{BV}\,(\Omega \times \mathbb{R})$,

$$F(v) = \int_{-\infty}^{+\infty} F(\mathbb{1}_{\{v>s\}})\,\mathrm{d}s.$$

Puisque la fonction v est supposée à valeurs dans $[\,0\,;1\,]$, les ensembles de niveaux supérieurs à 1 sont vides et les ensembles de niveaux inférieurs à 0 sont l'ensemble $\Omega\times\mathbb{R}$ tout entier. Ainsi, la formule de la co-aire se réécrit dans ce cas précis

$$F(v) = \int_{0}^{1} F(\mathbb{1}_{\{v>s\}})\,\mathrm{d}s.$$

Introduction d'un problème fortement convexe En reprenant la démarche du paragraphe 2.2.2 des notes [3], nous allons montrer que

Théorème 11 *Soit $M > 0$. La fonction $w^* \in \mathrm{BV}\,(\Omega \times \mathbb{R})$ est l'unique solution du problème fortement convexe*

$$\min_{w\in\mathcal{D}} F(w) + \frac{1}{2}\,\|w\|^2_{\mathrm{L}^2(\Omega\times\mathrm{I_{disp}})} \tag{3.14}$$

avec

$$\mathcal{D} = \left\{ w \in \mathrm{BV}\,(\Omega\times\mathbb{R};\mathbb{R}) \;\middle|\; \forall\, x\in\Omega,\ \begin{array}{l} \forall\, t \le \min(\mathrm{I_{disp}}),\ w(x,t) = +M \\ \forall\, t \ge \max(\mathrm{I_{disp}}),\ w(x,t) = -M \end{array} \right\} \tag{3.15}$$

si et seulement si ses ensembles de niveaux $\{(x,t) \in \Omega\times\mathbb{R} \mid w(x,t) > s\}$ sont solutions des problèmes

$$\min_{\substack{E\subset\Omega\times\mathbb{R}\\ E \text{ satisfaisant } (\star)}} F(\mathbb{1}_E) + s\,|E \cap (\Omega \times \mathrm{I_{disp}})| \tag{3.16}$$

pour tout $s \in [\,-M\,;M\,]$.

Pour cela, il suffit de montrer que les solutions du problème (3.16) sont des ensembles décroissants pour l'inclusion, et permet donc, à l'aide de la formule (3.6), de définir une fonction. Puis on montre que cette fonction est la solution du problème fortement convexe (3.14). L'unicité de cette solution prouve alors la réciproque du théorème.

Démonstration : Commençons par démontrer que le lemme 2.4 de [3]. Pour tout réel s, posons E_s une solution du problème (3.16). Montrons alors que pour tous $-M \leq s < s' \leq M$, on a l'inclusion $E_{s'} \subset E_s$. Il suffit pour cela de démontrer l'inégalité suivante pour tous ensembles A et B

$$F(\mathbb{1}_{A\cup B}) + F(\mathbb{1}_{A\cap B}) \leq F(\mathbb{1}_A) + F(\mathbb{1}_B).$$

On commence par remarquer que, géométriquement, il est aisé de vérifier que

$$F^{\text{vis}}(\mathbb{1}_{A\cup B}) + F^{\text{vis}}(\mathbb{1}_{A\cap B}) \leq F^{\text{vis}}(\mathbb{1}_A) + F^{\text{vis}}(\mathbb{1}_B)$$

car si le membre de droite est fini (le cas infini étant trivial), alors cela implique uniquement des conditions sur la frontière de chacun des deux ensembles A et B, conditions qui restent préservées par l'union et l'intersection de ces deux ensembles. Par ailleurs, si A n'est pas l'ensemble des ensembles de niveaux d'une fonction u, alors on peut montrer que $F^{\text{TV}}(\mathbb{1}_A) = +\infty$. À nouveau, cela nous conduit à ne considérer que le cas où $\mathbb{1}_A = \mathbb{1}_{u_A}$ et $\mathbb{1}_B = \mathbb{1}_{u_B}$ sont des indicatrices de sous-graphes. Dans ce cas, l'union $A \cup B$ correspond à la fonction $\max(u_A,u_B)$, tandis que l'intersection correspond à la fonction $\min(u_A,u_B)$. On a alors d'une part pour tout $x \in \Omega$

$$g\Big(x, \max(u_A(x),u_B(x))\Big) + g\Big(x, \min(u_A(x),u_B(x))\Big) = g\Big(x,u_A(x)\Big) + g\Big(x,u_B(x)\Big)$$

et d'autre part

$$\text{TV}\Big(\max(u_A,u_B)\Big) + \text{TV}\Big(\min(u_A,u_B)\Big) \leq \text{TV}\Big(u_A\Big) + \text{TV}\Big(u_B\Big).$$

On en déduit alors que

$$F^{\text{TV}}(\mathbb{1}_{A\cup B}) + F^{\text{TV}}(\mathbb{1}_{A\cap B}) \leq F^{\text{TV}}(\mathbb{1}_A) + F^{\text{TV}}(\mathbb{1}_B).$$

Pour tout $s \in \mathbb{R}$, l'ensemble E_s désigne désormais une solution du problème (3.16) (dont on peut montrer qu'elle est en réalité unique pour presque tout s). Montrons que

$$w^* : (x,t) \mapsto \begin{cases} \sup\left\{ s \in [\,-M\,;M\,] \;\middle|\; (x,t) \in E_s \right\} & \text{si } \exists\, s \in [\,-M\,;M\,], (x,t) \in E_s \\ -M & \text{sinon} \end{cases}$$

est solution de (3.14), en suivant la démonstration du lemme 2.5 de [3]. La fonction w^* est bien de carré intégrable, car elle est bornée sur un domaine bornée. Montrons que w^* est bien d'énergie minimale. Pour cela, on commence par remarquer que, pour tout $s \in [\,-M\,;M\,]$,

$$E_s^- = \bigcap_{s'>s} E_{s'} = \Big\{ w^* > s \Big\}$$

est la plus petite solution de (3.16). On peut alors reprendre la preuve du lemme 2.5 de [3] (avec $g = 0$ d'une part et en utilisant la formule de la co-aire pour F d'autre part).

Résolution numérique du problème fortement convexe La discrétisation du problème (3.14) conduit à considérer le problème suivant

$$\min_{w^h \in \mathcal{D}^h} \sup_{\substack{\phi^h \in \mathcal{K}^h \\ \lambda^h \geq 0}} \left\{ \langle \phi^h, \nabla^h w^h \rangle + \langle \lambda^h, \delta_x^h w^h + \delta_t^h w^h \rangle + \frac{1}{2} \|w^h\|_2^2 \right\} \tag{3.17}$$

où on utilise à nouveau le multiplicateur de Lagrange λ^h. La discrétisation du convexe $\mathcal{D}$ est donnée par

$$\mathcal{D}^h = \left\{ v^h \in \mathbb{R}^{N_x N_y N_t} \;\middle|\; \forall\, (i,j) \in \Omega^h, v^h_{i,j,0} = -M \text{ et } v^h_{i,j,N_t-1} = M \right\}$$

On résout ce problème à nouveau par algorithme primal-dual, ce qui nous amène à proposer l'algorithme suivant : on initialise la variable primale $(w^h)_0 = (\bar{w}^h)_0 \in \mathcal{D}^h$ et les variables duales $(\phi^h)_0 \in \mathcal{K}^h$ et $(\lambda^h)_0 \geq 0$, puis on effectue pour tout $n \in \mathbb{N}$ les mises-à-jours suivantes

$$\left\{\begin{array}{rcl} (\phi^h)_{n+1} & = & \mathrm{proj}_{\mathcal{K}^h}\Big((\phi^h)_n + \sigma\,\nabla^h(\bar{w}^h)_n\Big) \\ (\lambda^h)_{n+1} & = & \mathrm{proj}_{([0\,;+\infty[)^{N_x N_y N_t}}\Big((\lambda^h)_n + \sigma\,(\delta_x^h + \delta_t^h)(\bar{w}^h)_n\Big) \\ (w^h)_{n+1} & = & \mathrm{proj}_{\mathcal{D}^h}\left(\dfrac{(w^h)_n - \tau\Big(\mathrm{div}^h(\phi^h)_{n+1} + (\varepsilon_x^h + \varepsilon_t^h)(\lambda^h)_{n+1}\Big)}{1+\tau}\right) \\ (\bar{w}^h)_{n+1} & = & (1+\theta_{n+1})\,(w^h)_{n+1} - \theta_{n+1}(w^h)_n \end{array}\right.$$

qui alterne cette fois une montée de gradient projeté en (ϕ^h,λ^h) et une descente de gradient projeté en w^h. L'intérêt majeur de cet algorithme est qu'il résout un problème fortement convexe : d'une part, cela assure l'unicité de la solution, et d'autre part, la convergence est plus rapide.

Choix des paramètres On choisit d'initialiser les pas de temps avec $\tau_0 = 0{,}1$ d'une part et $\sigma_0 = 1/(\tau_0 L^2)$ d'autre part, puis de les mettre à jour de la manière suivante :

$$\forall\, n \in \mathbb{N}, \qquad \tau_{n+1} = \tau_n\,\theta_{n+1} \qquad \text{et} \qquad \sigma_{n+1} = \frac{\sigma_n}{\theta_{n+1}}.$$

Le paramètre de relaxation variable θ_n est quant à lui choisi vérifiant

$$\forall\, n \in \mathbb{N}, \qquad \theta_{n+1} = \frac{1}{\sqrt{1 + 1{,}2\,\tau_n}}.$$

Convergence Puisque c'est le niveau 0 de la solution w^h qui nous intéresse, on peut également choisir de stopper les itérations lorsque ce niveau se stabilise. Pour cela, toutes les dix itérations par exemple, on compte le nombre de coefficients de la variable primale w^h qui ont changé de signe ; si ce nombre tombe en dessous de certain seuil ($N_x \times N_y/10\,000$ par exemple), alors l'algorithme s'arrête.

Empiriquement, on remarque que ce critère n'est pertinent qu'après un certain nombre d'itérations (car les premières itérées de w^h varient peu), c'est pourquoi on impose également un minimum de 1 100 itérations avant d'arrêter la boucle.

Seuillage de la solution Pour obtenir la carte de disparité u^h à partir de la solution w^h obtenue après N itérations de l'algorithme proposé, on applique cette fois le théorème 11. On seuille pour obtenir le niveau 0 de w^h, puis on reconstruit u^h à l'aide de la version discrète de la formule (3.6) :

$$\forall\,(i,j) \in \Omega^h, \qquad u_{i,j}^h = h_t\,\sup\Big\{k \in [\![\,0\,;\,N_t - 1\,]\!] \;\Big|\; w_{i,j,k}^h > 0\Big\}.$$

3.4 Détection et gestion de l'occultation

On suppose à présent que la carte de la disparité u^h a été estimée par l'un de deux algorithmes présentés à la section précédente. La carte est dense, c'est-à-dire qu'une disparité est attribuée à chaque pixel, qu'il soit en réalité occulté ou non. Or, la fonctionnelle d'énergie a été conçue spécifiquement pour permettre de distinguer les

(a) Profils de gauche (haut) et de droite (bas)

(b) Deux discrétisations possibles

FIGURE 3.1 – Discrétisation et adhérence. Parce que les pixels et les disparités sont discrétisés, si la zone d'occultation n'est pas correctement placée (par rapport à la grille d'échantillonnage), deux profils sont possibles pour approcher le profil de gauche idéal recherché (points rouges pleins ou vides). Or, la première solution engendre un coût d'attache aux données potentiellement plus bas, car elle met en correspondance des pixels appartenant au même objet violet, donc *a priori* semblables (le coût de régularité étant le même dans les deux cas).

pixels occultés des autres. Nous allons donc décrire le traitement destiné à 1) détecter ces pixels ; 2) leur attribuer une disparité raisonnable.

3.4.1 Détection de l'occultation par saturation de la pente

On rappelle que la disparité n'est pas définie dans les zones occultées. Celle qui est estimée par la méthode étudiée dans ce chapitre n'est en réalité que le produit d'une interpolation affine (sur la ligne). L'analyse menée au chapitre 2 assure que ces pixels interpolés peuvent être distingués des autres. Les pixels de l'image de référence peuvent en effet être classés en deux groupes : ceux pour lesquels la dérivée horizontale de la disparité interpolée vaut 1, et les autres. Les premiers correspondent à des pixels occultés.

Ainsi, une manière de détecter l'occultation dans l'image de référence une fois que la disparité dense u^h a été estimée est de calculer la pente horizontale de la disparité, puis d'identifier tous les pixels pour lesquels cette pente vaut 1. Comme il s'agit de la pente maximale théorique, on parle de *saturation de la pente*. Le masque qui en résulte permet de localiser précisément les pixels occultés. Numériquement, on utilise comme approximation de la dérivée horizontale l'opérateur $\tilde{\delta}_x^h$ défini pour tout pixel $(i,j) \in \Omega^h$ par les différences finies

$$(\tilde{\delta}_x^h u^h)_{i,j} = \begin{cases} u^h_{i+1,j} - u^h_{i,j} & \text{si } i < N_x - 1 \\ 0 & \text{si } i = N_x - 1. \end{cases} \tag{3.18}$$

Dans le cas discret, une étude rapide montre que, si on utilise directement cette dérivée pour détecter les occultations, on introduit de manière systématique de l'adhérence sur les bords gauches des objets. Pour tenter de comprendre ce phénomène, intéressons-nous à la situation présentée à la figure 3.1. On y présente à gauche deux profils, qui correspondent aux profils de gauche (en haut) et de droite (en bas) d'une scène théorique. Pour distinguer les objets auxquels ils se réfèrent, chaque portion du profil est de couleur différente. Dans le profil de gauche, les pointillés correspondent à

des portions occultées (invisibles depuis la vue de droite), dans le profil de droite, les pointillés correspondent à une portion non visible sur l'image de droite. L'interpolation de la zone d'occultation du profil de gauche est proposé en rouge : il s'agit d'un segment de pente 1.

Sauf cas particulier, la zone d'occultation commence et termine en des points qui n'appartiennent pas à la grille Ω^h. Dans ce cas (figure 3.1, droite), puisque la disparité aussi est discrétisée, il existe deux moyens d'approcher le profil exact. Chacun de ces deux profils discrets induit une mise en correspondance différente des pixels. Dans le premier cas (points rouges pleins), tous les pixels de la zone occultée sont mis en correspondance avec le dernier point non occulté de l'objet violet (qui correspond ici à l'objet occulté) ; dans le second cas (points rouges vides), tous les pixels de la zone occultée sont mis en correspondance avec le premier point non occulté de l'objet bleu clair (qui correspond lui à l'objet occultant). Dans l'exemple étudié, les points occultés appartenant à l'objet violet, il semble raisonnable d'estimer que les pixels occultés ressemblent aux pixels non occultés de cet objet, et sont très différents des pixels de l'objet bleu clair. La première solution produit donc un coût de mise en corespondance *a priori* beaucoup plus bas que la seconde solution, qui met en correspondance tous les points de la région occultée avec un pixel très différent. Par ailleurs, dans les deux cas, le coût de régularisation est le même. La première solution sera donc préférentiellement choisie ; or, elle introduit de l'adhérence (d'une largeur 1).

Pour corriger ce biais pratiquement systématique, on choisit de translater artificiellement les zones occultées d'un pixel vers la droite. On est donc amené à définir le masque d'occultation de la manière suivante

$$\forall\,(i,j) \in \Omega^h, \qquad M^{\text{occ}}_{i,j} = \begin{cases} 1 & \text{si } (\tilde{\delta}^h_x u^h)_{i,j-1} \geq 1 \\ 0 & \text{si } (\tilde{\delta}^h_x u^h)_{i,j-1} < 1 \end{cases} \tag{3.19}$$

qui vaut 1 en cas d'occultation et 0 sinon.

3.4.2 Densification des zones occultées

Une fois les zones occultées localisées, on va y modifier la valeur de la disparité, dont on sait qu'en l'état, elle n'est pas significative. Évidemment, la disparité ne peut pas être connue dans ces régions, mais on peut toutefois en proposer une interpolation raisonnable.

On se base pour cela sur le modèle de scène que nous avons déjà utilisé pour analyser le phénomène d'occultation : un objet est partiellement occulté par un objet qui se situe devant lui, et on suppose que, localement, les objets ont une disparité constante. Sans d'autres informations, on peut donc faire l'hypothèse que la partie occultée est de disparité constante, qui se prolonge (au moins dans un voisinage) en dehors de la zone d'occultation.

L'occultation se produisant (dans l'image de référence) sur les bords *gauches* des objets, on en déduit que le premier point non occulté situé *à gauche* de la région occultée n'est pas sur l'objet occultant. Il s'agit donc *a priori* d'une partie non occultée de l'objet occulté. Par conséquent, la disparité de la région occultée la plus raisonnable est donnée par celle de ce premier point non occulté. On interpole ainsi la carte de disparité en diffusant (horizontalement) vers la droite la disparité connue dans les régions occultées. Plus précisément, l'interpolation $\tilde{u}^h$ s'écrit

$$\forall\,(i,j) \in \Omega^h, \qquad \tilde{u}^h_{i,j} = \begin{cases} u^h_{i,j} & \text{si } M^{\text{occ}}_{i,j} = 0 \\ u^h_{i_0,j} & \text{si } M^{\text{occ}}_{i,j} = 1 \text{ et } i_0 = \max\{i' \leq i \mid M^{\text{occ}}_{i',j} = 0\}. \end{cases} \tag{3.20}$$

On retrouve cette démarche dans [13, 14].

3.5 Résultats expérimentaux

On présente dans cette section les résultats obtenus avec d'une part l'algorithme sur-relaxé et d'autre part l'algorithme accéléré proposé dans la section précédente. Les tests sont effectués sur les paires du banc d'essai Middlebury (voir 2.3.4) et sont évalués à partir des vérités-terrains fournies, ainsi que des cartes d'occultation modifiées selon la méthode décrite au paragraphe suivant, le cas échéant.

3.5.1 Modification des cartes d'occultation

Les cartes d'occultation proposées par le banc d'essai Middlebury présentent deux défauts majeurs, qui sont

1. la suppression des occultations de largeur 1 ;
2. l'introduction des occultations hors-champ. En effet, certains pixels de l'image de référence ne possèdent pas de pixel homologue dans la vue de droite non pas parce qu'ils ont été occultés par un objet occultant, mais parce que leur pixel homologue est hors-cadre dans l'image de droite. Or ceux-ci n'obéissent pas aux mêmes lois que les pixels réellement occultés et ne devraient donc pas être pris en compte dans la carte des occultations (mais plutôt être considérés comme des pixels dont la disparité ne peut être évaluée).

Nous proposons donc ici de modifier les cartes proposées par Middlebury de sorte de supprimer ces deux écueils.

Réintroduction des occultations de largeur 1 Pour réintroduire les occultations fines qui ont été supprimées, on commence par supposer que la contrainte de préservation d'ordre est satisfaite pour les paires considérées. Cette hypothèse nous permet d'utiliser l'analyse du phénomène d'occultation proposée au chapitre 2 et de déduire de la vérité-terrain dense une première carte des occultations (figure 3.2, en rouge dans la colonne de gauche). En effet, chaque discontinuité de disparité entraîne sur sa gauche une occultation, dont la largeur est exactement égale au saut de disparité.

Malheureusement, les scènes de Middlebury ne vérifient pas toujours la contrainte de préservation de l'ordre, en particulier Tsukuba (bras de la lampe) et Cones (pinceaux et sommets des cones), ce qui conduit à une sur-détection des occultations. Pour raffiner ces premières cartes, on choisit alors de tester l'injectivité de la vérité-terrain dans les zones déclarées occultées à la première étape. En d'autres termes, on teste pour chacun des pixels déclarés occultés si, d'après la vérité-terrain, d'autres pixels ont été mis en corresdance avec son pixel homologue. Si c'est le cas, le pixel est définitivement considéré comme occulté (figure 3.2, en rouge dans la colonne de droite).

Suppression des occultations hors-champ Pour détecter les occultations qui sont dues à la restriction des images sur un domain fini Ω, on teste pour chaque pixel si son pixel homologue est bien dans le domaine de l'image de droite. Si ce n'est pas le cas, il est considéré comme les occultations hors-champ (figure 3.2, en cyan).

Les nouvelles cartes d'occultation obtenues sont présentées à la figure 3.2. Sauf mention contraire, ce sont elles qui sont dorénavant considérées comme la vérité-terrain des cartes d'occultation.

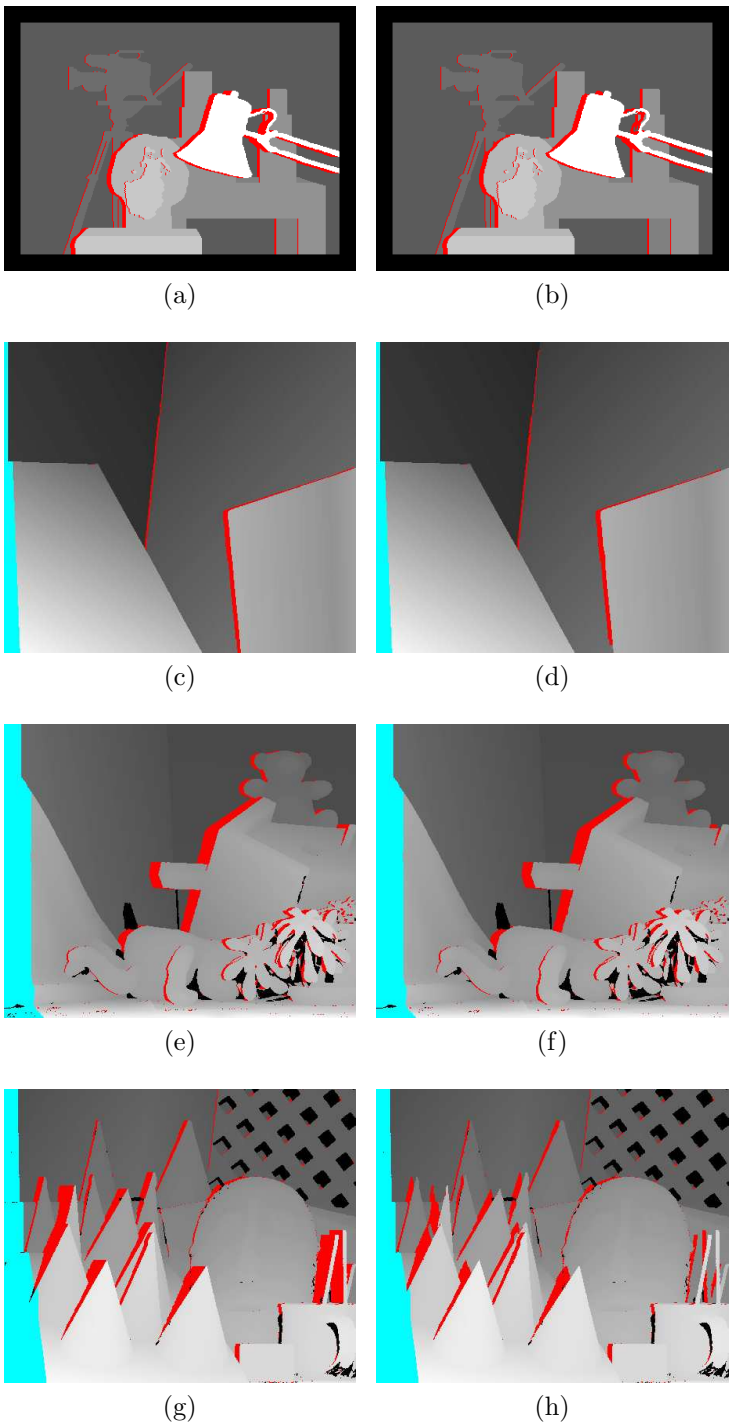


FIGURE 3.2 – Modification des cartes d'occultation. Colonne de gauche : premières cartes obtenues à partir des discontinuités de la vérité-terrain. Colonne de droite : raffinement calculé grâce à la consistence gauche-droite de la vérité-terrain. En rouge : les pixels occultés, en cyan les occultations hors-champ. De haut en bas : Tsukuba, Venus, Teddy et Cones.

3.5.2 Récapitulatif de l'algorithme

On rappelle que la méthode décrite plus haut se décompose en deux étapes :

1. résolution par algorithme primal-dual sur-relaxé du problème convexe (3.13) (désigné sous le nom de **algorithme sur-relaxé**), puis seuillage à $s = 0{,}9$; on obtient alors une première carte de disparité u^h ;

1bis ou résolution par algorithme primal-dual du problème fortement convexe (3.14) (désigné sous le nom de **algorithme accéléré**), puis seuillage à $s = 0$; on obtient également une première carte de disparité u^h ;

2. détection des zones occultées par saturation de la pente et densification par diffusion (vers la droite) de la disparité.

Dans ce qui suit, les paramètres de la méthode sont fixés comme suit :

- paramètre de pondération des termes d'attache aux données et de régularisation $\mu = 50/255$ [12] ;
- paramètres du terme d'attache aux données : $a = 100$ dans la définition du coefficient variable α^h, paramètre du lissage ROF $\lambda = 1/50$, convolution avec une gaussienne d'écart-type 8 et de support de taille 9×9 ;
- pas de discrétisation de la disparité $h_t \in \{1{,}0{,}5\}$.

Les cartes de disparités générées seront en particulier précises au pixel ou au demi-pixel près, suivant le choix de h_t.

3.5.3 Première carte de disparité

On présente dans ce paragraphe les résultats obtenus à l'issue de la première étape de l'algorithme, qui est l'estimation dense de la disparité obtenue par algorithme primal-dual.

Volume de coût Pour calculer le volume de coût g^h, il faut calculer pour chaque triplet d'indices (i,j,k) la quantité

$$g^h_{i,j,k} = (1 - \alpha^h_{i,j}) \, \|(I^h_{\mathrm{L}})_{i,j} - (I^{\mathrm{interp}}_{\mathrm{R}})_{i,j-kh_t}\| + \alpha^h_{i,j} \, \|(\tilde{\nabla}^h I^h_{\mathrm{L}})_{i,j} - (\tilde{\nabla}^h I^{\mathrm{interp}}_{\mathrm{R}})_{i,j-kh_t}\|$$

où l'opérateur $\tilde{\nabla}^h$ est défini par

$$\forall\, (i,j) \in \Omega^h, \qquad (\tilde{\nabla}^h \mathrm{I}^h)_{i,j} = \begin{pmatrix} (\tilde{\delta}^h_x \mathrm{I}^h)_{i,j} \\ (\tilde{\delta}^h_y \mathrm{I}^h)_{i,j} \end{pmatrix}$$

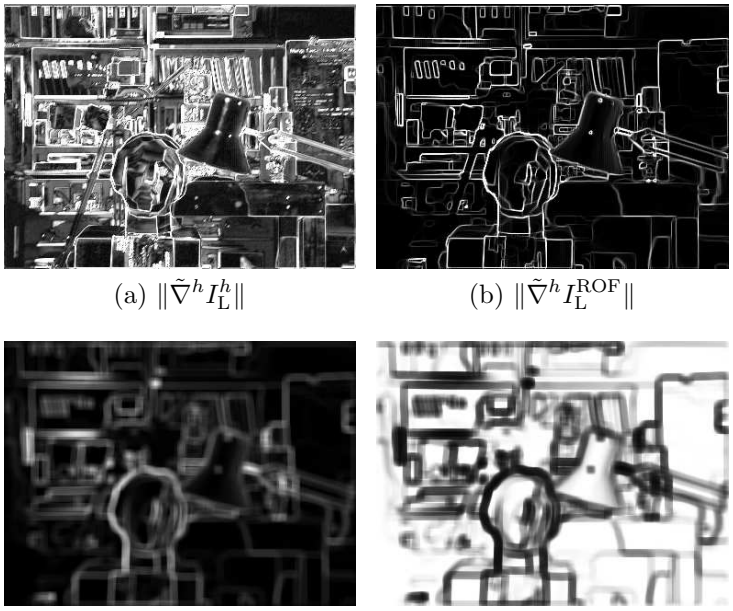
Les variations horizontales $\tilde{\delta}^h_x \mathrm{I}^h$ sont données par la formule (3.18) et les variations verticales $\tilde{\delta}^h_y$ sont définies de manière analogue par les différences finies

$$(\tilde{\delta}^h_y \mathrm{I}^h)_{i,j} = \begin{cases} \mathrm{I}^h_{i,j+1} - \mathrm{I}^h_{i,j} & \text{si } j < N_y - 1 \\ 0 & \text{si } j = N_y - 1. \end{cases}$$

L'image interpolée $I^{\mathrm{interp}}_{\mathrm{R}}$ est obtenue par interpolation B-spline d'ordre 5 [16, 15] de l'image échantillonnée I^h_{R}.

La pondération variable α^h est définie par

$$\forall\, (i,j) \in \Omega^h, \qquad \alpha^h_{i,j} = \frac{1}{1 + G \star \|(\tilde{\nabla}^h I^{\mathrm{ROF}}_{\mathrm{L}})\|^2_{i,j}/a}$$



(a) $\|\tilde{\nabla}^h I_{\mathrm{L}}^h\|$ (b) $\|\tilde{\nabla}^h I_{\mathrm{L}}^{\mathrm{ROF}}\|$

(c) $G \star \|\tilde{\nabla}^h I_{\mathrm{L}}^{\mathrm{ROF}}\|$ (d) α^h

FIGURE 3.3 – Amplitude du gradient dans la construction de α^h, exemple de la paire Tsukuba. (a) Amplitude du gradient de l'image initiale I_{L}^h : trop de détails sont apparents, qui sont principalement dus à la texture des objets. (b) Amplitude du gradient de l'image lissée $I_{\mathrm{L}}^{\mathrm{ROF}}$: le lissage ROF permet de supprimer les variations non significatives et de conserver celles qui correspondent plus vraisemblablement à des discontinuités de scène. (c) Filtrage de $\|\tilde{\nabla}^h I_{\mathrm{L}}^{\mathrm{ROF}}\|$ par une gaussienne : ce lissage permet d'étaler spatialement les forts gradients. Il est en effet essentiel que les voisinages de discontinuités soient traités comme des discontinuités. (d) Pondération α^h résultant : les discontinuités et leur voisinage correspondent à des valeurs faibles de α^h. La comparaison y sera donc majoritairement basée sur l'intensité, et non le gradient.

où l'image lissée $I_{\mathrm{L}}^{\mathrm{ROF}}$ est obtenue en résolvant le problème

$$\min_{\mathrm{I}^h \in \mathbb{R}^{3N_x N_y}} \frac{\lambda}{2} \|I_{\mathrm{L}}^h - \mathrm{I}^h\|_2^2 + \mathrm{TV}(I^h) \tag{3.21}$$

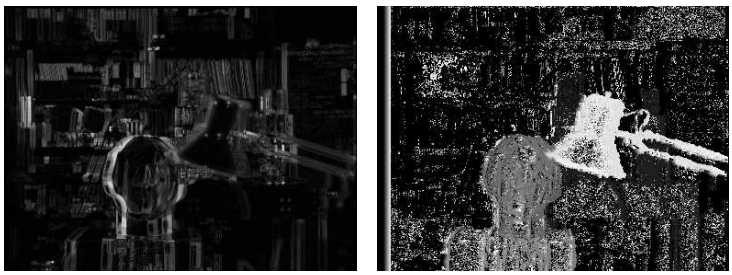
(voir chapitre 6 pour plus d'informations au sujet du débruitage ROF). On choisit $\lambda = 1/50$. L'intérêt d'utiliser une version lissée de l'image initiale est d'éliminer les variations non significatives. On choisit pour cela la régularisation ROF car elle préserve les discontinuités importantes. La gaussienne utilisée pour la convolution est quant à elle choisie centrée d'écart-type 8. Elle permet d'obtenir des valeurs de α^h faibles au voisinage des discontinuités. On pourra visualiser l'influence du lissage ROF et de la convolution par la gaussienne sur la construction de α^h à la figure 3.3.

On a ainsi construit le volume de coût g^h, de dimension $N_x \times N_y \times N_t$, dont on peut visualiser une tranche à la figure 3.4(a). Sans régularisation, la minimisation revient à effectuer un WTA, c'est-à-dire de prendre pour chaque pixel la disparité impliquant un coût de corrélation minimal :

$$\forall\,(i,j) \in \Omega^h, \qquad u_{i,j}^{\mathrm{WTA}} = h_t \underset{k \in [\![\,0\,;\,N_t\,]\!]}{\operatorname{argmin}} g_{i,j,k}^h$$

dont le résultat est présenté à la figure 3.4(b).

Nombre d'itérations Pour stopper les itérations, on choisit

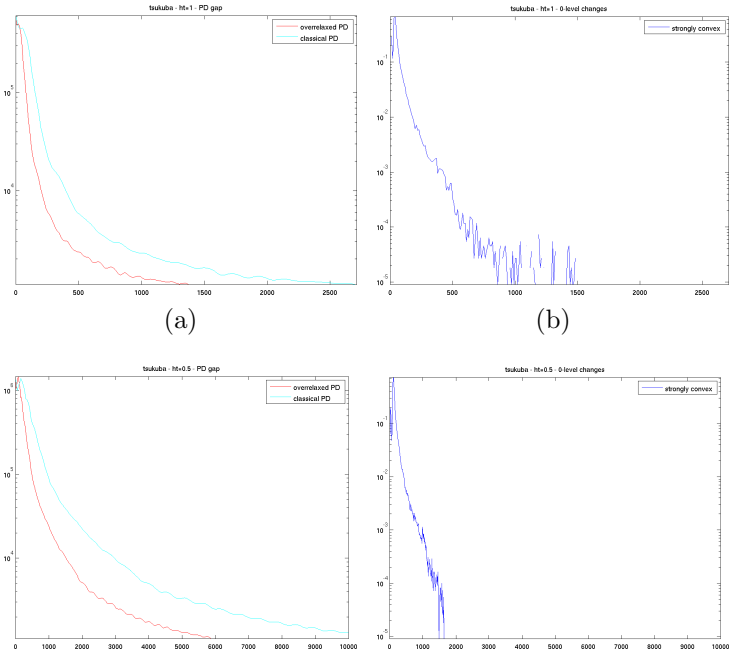


(a) $g^h_{i,j,1}$ (b) u^{WTA}

FIGURE 3.4 – Volume de coût g^h associé à la paire Tsukuba. (a) Tranche 1 du volume g^h. Plus le pixel est clair, plus le coût de corrélation est important. (B) WTA (*Winner-Takes-All*) : pour chaque pixel de l'image de référence, on retient la disparité qui minimise le coût de corrélation.

Paire	Tsukuba	Venus	Teddy	Cones
Algorithme sur-relaxé ($h_t = 1$)	700	1 390	3 770	4 020
Algorithme accéléré ($h_t = 1$)	1 490	1 520	3 350	2 370
Algorithme classique ($h_t = 1$)	1 350	2 710	7 350	7 830
Algorithme sur-relaxé ($h_t = 0{,}5$)	2 470	5 780	10 000+	10 000+
Algorithme accéléré ($h_t = 0{,}5$)	1 650	2 050	5 920	4 510
Algorithme classique ($h_t = 0{,}5$)	4 800	10 000+	10 000+	10 000+

FIGURE 3.5 – Nombre d'itérations obtenu grâce au critère d'arrêt. L'*algorithme classique* désigne une variante de l'algorithme sur-relaxé, sans la sur-relaxation globale ($\rho = 1$). Le nombre d'itérations maximal est limité à 10 000 pour chaque expérience.



(a) (b)

(c) (d)

FIGURE 3.6 – Comparaison des *gaps* pour les algorithmes primaux-duaux présentés dans ce chapitre (l'échelle est logarithmique pour l'ordonnée), exemple de la paire Tsukuba. (a) et (c) Algorithmes sur-relaxé et classique. (b) et (d) Algorithme accéléré. Ligne du haut : précision pixellique. Ligne du bas : précision sous-pixellique.

Paire	Tsukuba			Venus		
Précision pixellique ($h_t = 1$)						
Algorithme sur-relaxé	9,92%	2,44%	9,92%	4,08%	3,34%	22,44%
Algorithme accéléré	11,73%	3,40%	11,73%	5,01%	3,52%	23,01%
Précision sous-pixellique ($h_t = 0{,}5$)						
Algorithme sur-relaxé	6,26%	2,51%	23,33%	3,64%	2,80%	12,72%
Algorithme accéléré	7,18%	3,05%	23,21%	3,62%	2,83%	12,94%

Paire	Teddy			Cones		
Précision pixellique ($h_t = 1$)						
Algorithme sur-relaxé	14,38%	11,57%	42,66%	11,18%	8,91%	35,22%
Algorithme accéléré	14,80%	11,87%	42,73%	10,65%	8,38%	34,10%
Précision sous-pixellique ($h_t = 0{,}5$)						
Algorithme sur-relaxé	11,97%	9,60%	26,58%	8,01%	6,48%	19,14%
Algorithme accéléré	12,00%	9,51%	25,80%	8,18%	6,67%	18,15%

FIGURE 3.7 – Erreur d'estimation dans la première estimation (dans les zones non occultées). Pour chaque paire, on considère l'erreur pixellique, l'erreur Middlebury et l'erreur sous-pixellique.

1. pour l'algorithme sur-relaxé : d'arrêter l'algorithme lorsque le *primal-dual gap* tombe en-dessous de $N_x \times N_y \times N_t/1\,000$ (ce qui revient à tolérer un *gap* moyen de l'ordre du millième) ;
2. pour l'algorithme accéléré : de mesurer la stabilité du niveau zéro de la variable primale et d'arrêter l'algorithme lorsque le nombre de coefficients changeant de signe tombe en dessous de $N_x \times N_y/10\,000$.

Le tableau 3.5 affiche le nombre d'itérations nécessaires pour activer le critère d'arrêt pour chaque algorithme et pour deux précisions différentes ($h_t = 1$ pour la précision pixellique et $h_t = 0{,}5$ pour la précision sous-pixellique). La figure 6.6 présente quant à elle l'évolution du critère mesuré pour l'arrêt (*primal-dual gap* et nombre de coefficients changeant de signe).

Cartes obtenues On présente les résultats obtenus sur les paires de Middlebury aux figures 3.8 et 3.9. Grâce à la vérité terrain, on peut déterminer précisément les estimations erronées, qui sont répertoriés dans le tableau 3.7. On s'intéresse aux erreurs pixelliques (supérieures ou égales au pixel) localisées en dehors des zones occultées ou non renseignées de la scène, ainsi qu'aux erreurs strictement supérieure à 1 (erreur Middlebury) et aux erreurs sous-pixelliques (supérieure ou égale à 0,5 px). L'erreur Middlebury est naturellement la plus faible, et l'erreur sous-pixellique la plus grande.

3.5.4 Traitement des occultations

Une fois la première estimation de la disparité obtenue, on procède au traitement des occultations en suivant la méthode présentée dans la section 3.4.

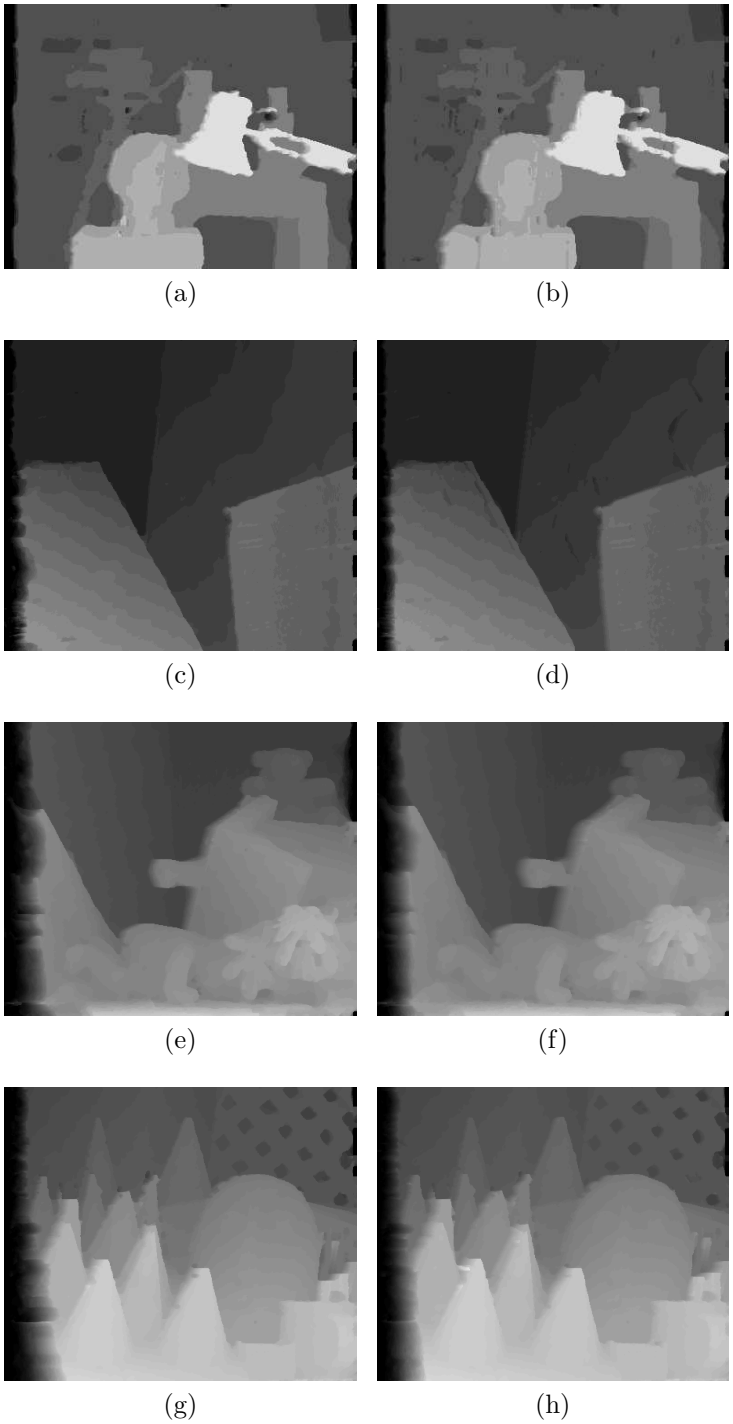


FIGURE 3.8 – Première estimation de la carte de disparité, précision **pixellique**. Colonne de gauche : algorithme sur-relaxé. Colonne de droite : algorithme accéléré. De haut en bas : Tsukuba, Venus, Teddy et Cones.

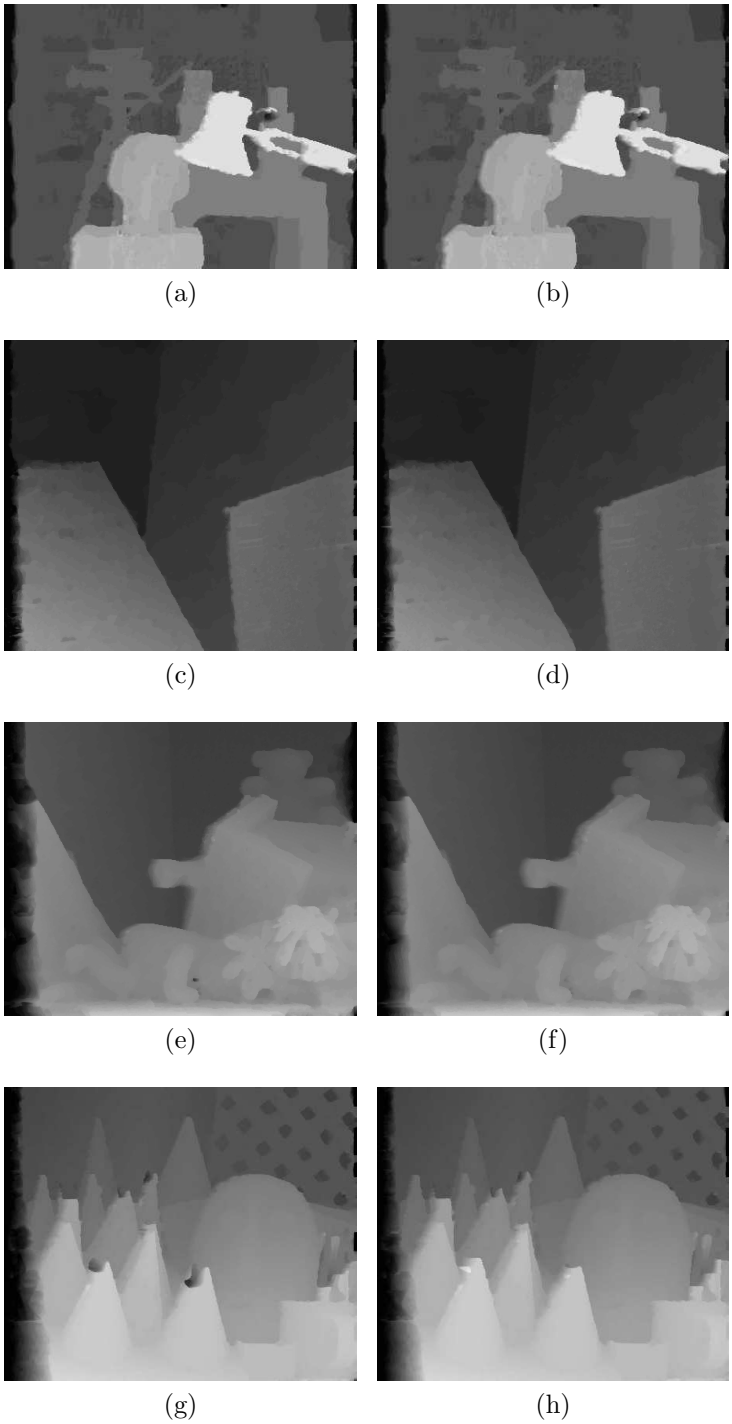


FIGURE 3.9 – Première estimation de la carte de disparité, précision **sous-pixellique**. Colonne de gauche : algorithme sur-relaxé. Colonne de droite : algorithme accéléré. De haut en bas : Tsukuba, Venus, Teddy et Cones.

Précision et rappel Pour évaluer la détection des occultations par notre méthode, on commence par mesurer le nombre de

1. vrais positifs (TP) : occultations correctement détectées comme telles ;
2. vrais négatifs (TN) : pixels non occultés d'après la vérité-terrain qui n'ont pas été détectés comme occultés ;
3. faux positifs (FP) : pixels non occultés selon la vérité-terrain mais détectés comme occultés par notre méthode ;
4. faux négatifs (FN) : pixels occultés selon la vérité-terrain mais non détectés comme tels.

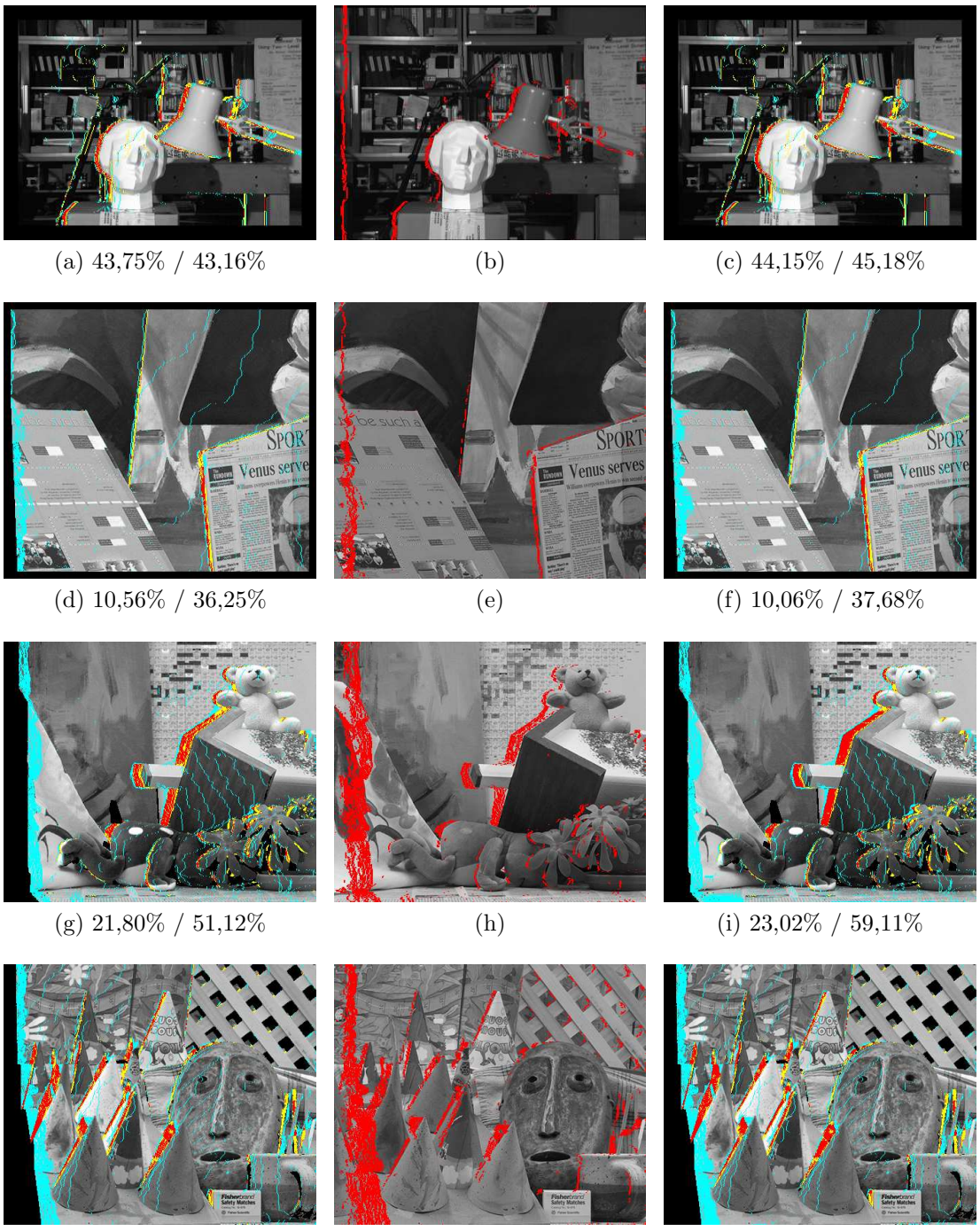
On rappelle qu'on utilise comme vérités-terrains les cartes d'occultation générées selon le paragraphe 3.5.1. On remarquera que la somme $\mathrm{TP} + \mathrm{FN}$ donne le nombre de pixels occultés d'après la vérité-terrain, tandis que la somme $\mathrm{TP} + \mathrm{FP}$ donne le nombre de pixels occultés d'après le détecteur.

Pour mesurer les performances de la détection, on utilise deux mesures appelées *précision* et *rappel*. La précision est définie par le rapport $\mathrm{TP}/(\mathrm{TP} + \mathrm{FP})$ tandis que le rappel est défini par le rapport $\mathrm{TP}/(\mathrm{TP} + \mathrm{FN})$. Plus ces deux taux sont proches de 1, meilleure est la détection. La précision mesure à quel point la détection est concentrée dans les zones à détecter : en effet, elle est élevée s'il y a peu de faux positifs. Ainsi, même si peu de détections sont faites, ce score peut être grand si, parmi elles, un grand nombre correspond à des détections correctes. Au contraire, si la méthode a tendance à sur-détecter les occultations, ce score peut être faible même si toutes les occultations ont été détectées. Le rappel ne s'intéresse quant à lui qu'à l'efficacité de la détection dans les zones occultées selon la vérité-terrain. Peu importe la détection en-dehors de ces zones, le rappel sera élevé si un grand nombre de détections sont correctes.

Détection par saturation de la pente On présente dans les figures 3.10 – 3.11 (colonne de gauche) les régions détectées par le masque (3.19), c'est-à-dire les pixels saturant la pente horizontale de la disparité. Chaque détection est marquée en rouge quand il s'agit d'un vrai positif et en cyan quand il s'agit d'un faux positif. Les points jaunes correspondent aux points non détectés, c'est-à-dire aux faux négatifs.

Densification Une fois les occultations détectées, on interpole la carte de disparité en diffusant (sur la ligne) la disparité des zones non occultées vers les zones occultées. Cette diffusion se fait vers la droite, en suivant la formule (3.20) (figures 3.12 – 3.13, colonne de gauche).

Amélioration de la détection La saturation de la pente peut s'avérer insuffisante pour détecter correctement les occultations. Parmi les causes des détections incorrectes, on peut citer le fait que, lorsque le bord de l'objet n'est pas vertical, mais incliné, localiser exactement les occultations implique d'introduire dans la carte de disparité des discontinuités qui suivent le bord de l'objet, ici incliné. Ce genre de discontinuité conduit à une variation totale plus importante, puisqu'il faut tenir compte à la fois des variations horizontales et verticales. Une manière de réduire le coût de régularisation consiste à déplacer localement les zones occultées, de sorte de les aligner d'une ligne à l'autre. On observe alors des motifs en escalier qui paraissent artificiels. Une autre cause d'erreur peut survenir dans le cas de la précision sous-pixellique. En effet, dans ce cas-là, l'occultation est détectée lorsqu'il y a un saut de disparité d'un pixel, alors que la carte peut présenter des sauts d'un demi-pixel. Un coût de mise en correspondance plus



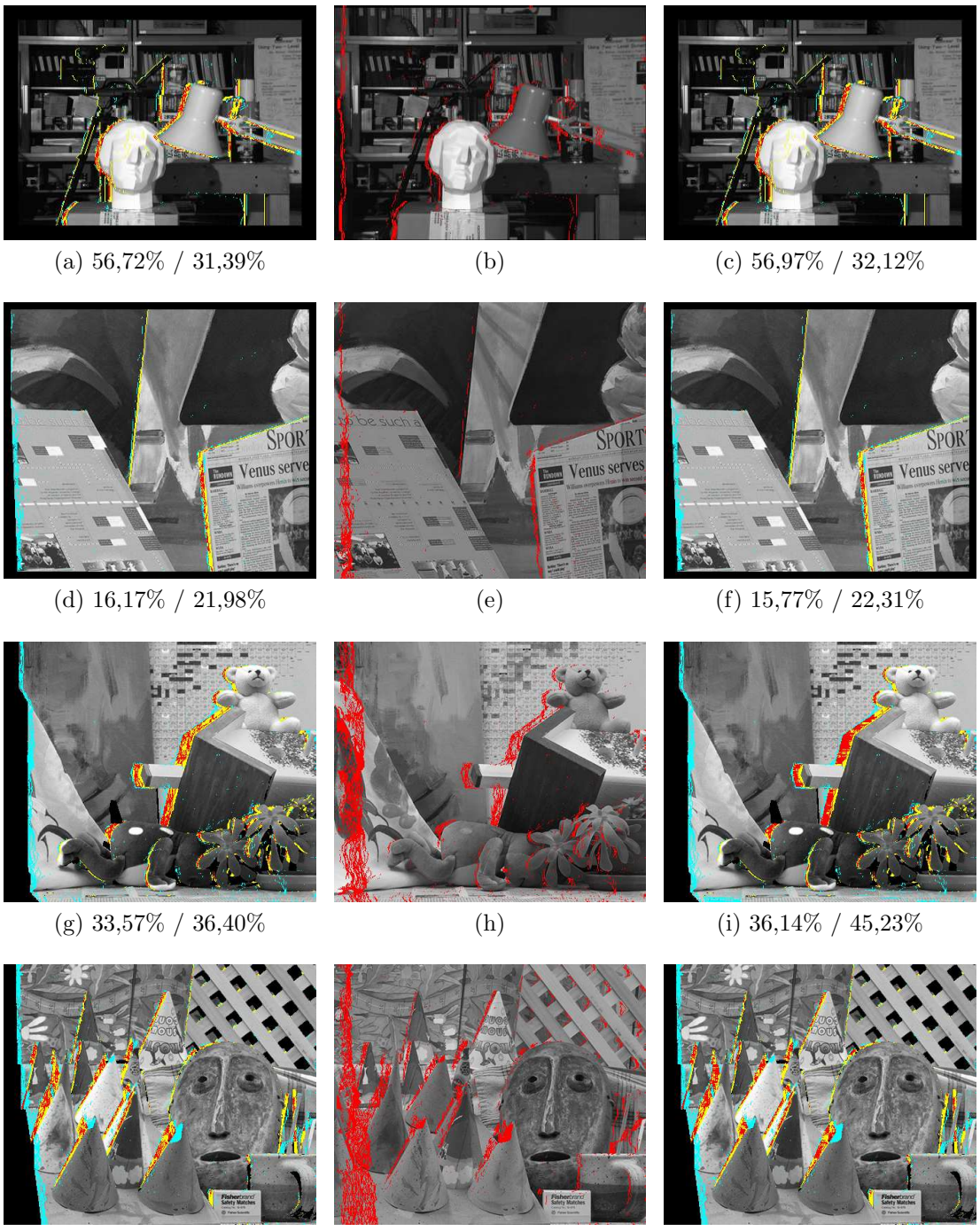
(a) 43,75% / 43,16% (b) (c) 44,15% / 45,18%

(d) 10,56% / 36,25% (e) (f) 10,06% / 37,68%

(g) 21,80% / 51,12% (h) (i) 23,02% / 59,11%

(j) 25,81% / 55,95% (k) (l) 26,12% / 58,49%

FIGURE 3.10 – Traitement des occultations, précision **pixellique**. Cas de l'algorithme **sur-relaxé**. Colonne de gauche : détection des occultations par saturation de la pente. Colonne du milieu : suppression des détections de largeur 1. Colonne de droite : amélioration des détections. En rouge, les détections correctes (vrais positifs, TP), en jaune, les détections manquantes (faux négatifs, FN) et en cyan, les détections incorrectes (faux positifs, FP). En légende : les taux de précision et de rappel. De haut en bas : Tsukuba, Venus, Teddy et Cones.



(a) 56,72% / 31,39% (b) (c) 56,97% / 32,12%

(d) 16,17% / 21,98% (e) (f) 15,77% / 22,31%

(g) 33,57% / 36,40% (h) (i) 36,14% / 45,23%

(j) 36,88% / 38,83% (k) (l) 38,04% / 41,40%

FIGURE 3.11 – Traitement des occultations, précision **sous-pixellique**. Cas de l'algorithme **sur-relaxé**. Colonne de gauche : détection des occultations par saturation de la pente. Colonne du milieu : suppression des détections de largeur 1. Colonne de droite : amélioration des détections. En rouge, les détections correctes (vrais positifs, TP), en jaune, les détections manquantes (faux négatifs, FN) et en cyan, les détections incorrectes (faux positifs, FP). En légende : les taux de précision et de rappel. De haut en bas : Tsukuba, Venus, Teddy et Cones.

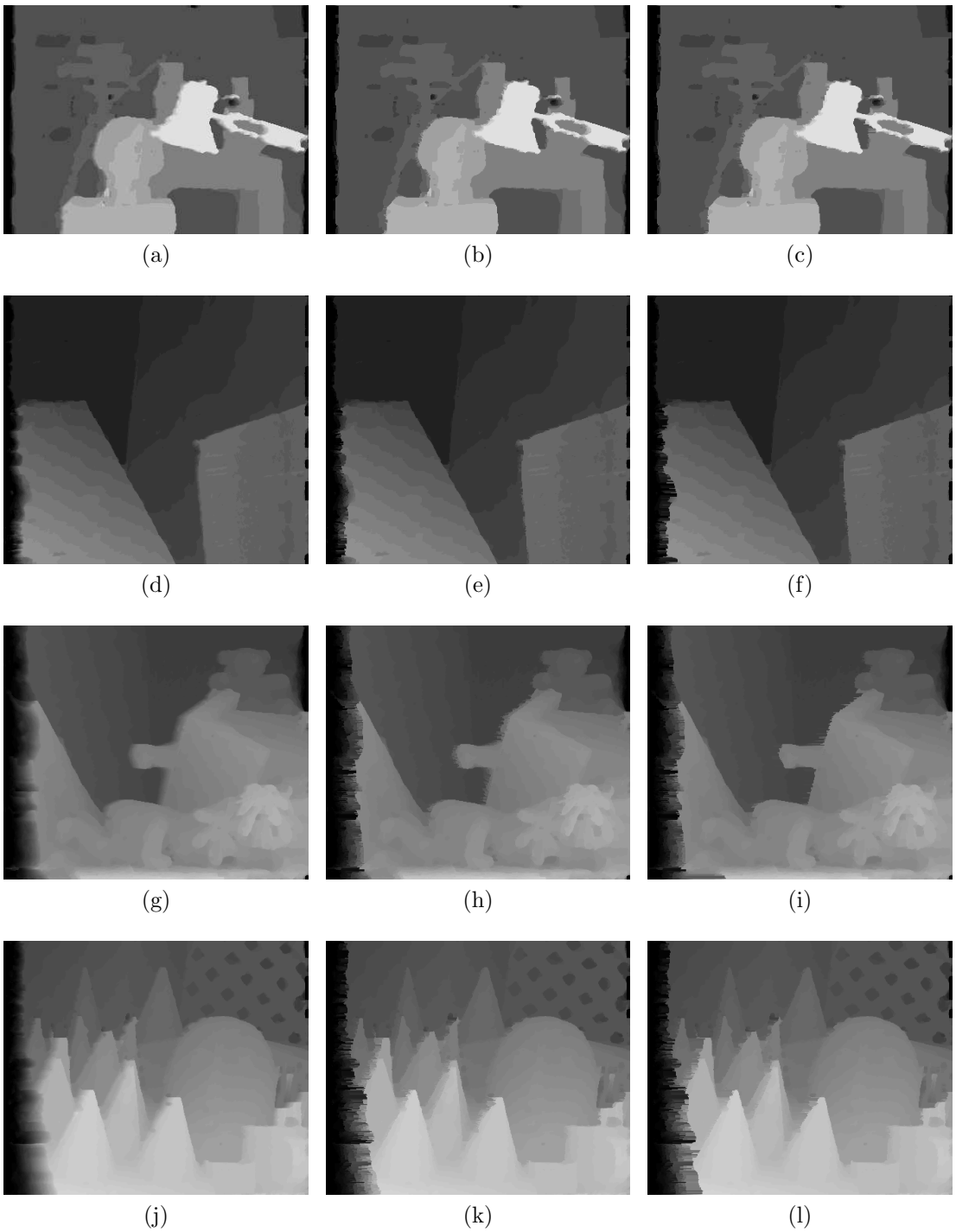


FIGURE 3.12 – Densification des occultations, précision **pixellique**. Cas de l'algorithme **surrelaxé**. Colonne de gauche : carte dense initiale. Colonne du milieu : densification à partir de la première détection des occultations par saturation de pente. Colonne de droite : amélioration des détections. De haut en bas : Tsukuba, Venus, Teddy et Cones.

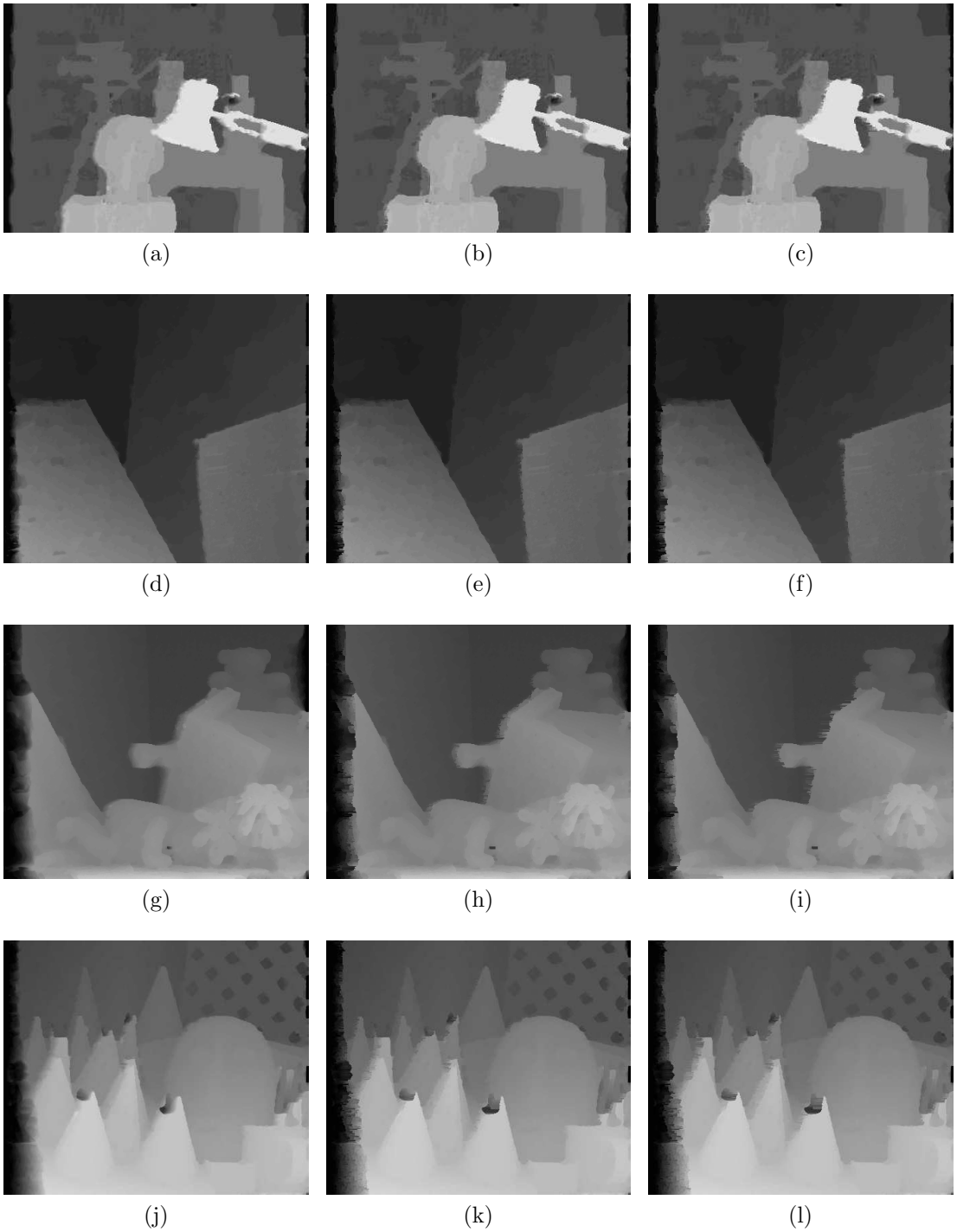


FIGURE 3.13 – Densification des occultations, précision **sous-pixellique**. Cas de l'algorithme **sur-relaxé**. Colonne de gauche : carte dense initiale. Colonne du milieu : densification à partir de la première détection des occultations par saturation de pente. Colonne de droite : amélioration des détections. De haut en bas : Tsukuba, Venus, Teddy et Cones.

favorable peut alors conduire à préférer dans les zones occultées deux sauts disjoints d'un demi-pixel à un saut plus franc d'un pixel, qui ne sont alors pas alors détectés comme occultation. Dans ces deux cas, ces artéfacts conduisent à créer, à l'intérieur des régions occultées, les zones où la disparité est constante. Les pixels correspondants ne saturant pas la constrainte de pente, ils ne sont pas détectés comme pixels occultés. Ainsi, les zones détectées comme occultées présentent parfois des trous : elles ne sont pas denses.

Pour densifier ces régions, on propose un post-traitement supplémentaire. Il consiste à déclarer occulté tout pixel qui serait situé dans un tel trou. Pour détecter ces pixels, on suppose que si un pixel est situé entre deux pixels occultés du même objet, alors il est vraisemblablement lui-même occulté. La difficulté est principalement de décider si deux points appartiennent au même objet, sans connaître au préalable la disparité de la scène. Nous proposons ici d'utiliser la version $I_{\mathrm{L}}^{\mathrm{ROF}}$ lissée par le modèle ROF de l'image de référence. Si la distance entre deux intensités $I_{\mathrm{L}}^{\mathrm{ROF}}(p)$ et $I_{\mathrm{L}}^{\mathrm{ROF}}(p')$ (pour la norme euclidienne de $\mathbb{R}^3$) est inférieure à un seuil $T = 8$, alors les deux pixels sont déclarés appartenant au même objet.

Le post-traitement proposé se décrit donc ainsi : pour tout pixel non occulté p, on recherche dans l'intervalle $[\,p - h_t(N_t - 1)\,;p + h_t(N_t - 1)\,]$ les voisins à gauche et à droite (sur la même ligne) occultés les plus proches, s'ils existent, de p :

$$p_1 = \underset{\substack{p' \in p - h_t [\![\, 1\,;\, N_t - 1 \,]\!] \\ M_{p'}^{\mathrm{occ}} = 1}}{\operatorname{argmin}} \|p - p'\| \qquad \text{et} \qquad p_2 = \underset{\substack{p' \in p + h_t [\![\, 1\,;\, N_t - 1 \,]\!] \\ M_{p'}^{\mathrm{occ}} = 1}}{\operatorname{argmin}} \|p - p'\|$$

étant entendu que la largeur d'une occultation ne peut excéder $h_t N_t$ pixels. On vérifie ensuite que p, p_1 et p_2 appartiennent au même objet, en définissant

$$M_{p,p'}^{\mathrm{objet}} = \begin{cases} 1 & \text{si } \|I_{\mathrm{L}}^{\mathrm{ROF}}(p) - I_{\mathrm{L}}^{\mathrm{ROF}}(p')\| \leq T \\ 0 & \text{sinon.} \end{cases}$$

Si c'est le cas, alors p est déclaré occulté :

$$\tilde{M}_p^{\mathrm{occ}} = \begin{cases} 1 & \text{si } M_p^{\mathrm{occ}} = 1 \\ 1 & \text{si } M_p^{\mathrm{occ}} = 0 \text{ et } (p_1,p_2) \text{ existent, avec } M_{p,p_1}^{\mathrm{objet}} M_{p,p_2}^{\mathrm{objet}} = 1 \\ 0 & \text{sinon.} \end{cases}$$

Dans le cas de la précision pixellique, on travaille sur une carte d'occultation où les occultations de largeur 1 ont été supprimées au préalable, car celles-ci sont généralement le fruit de la quantification des cartes de disparité (figures 3.10 – 3.11, colonne du milieu).

Le résultat de ce traitement est présenté aux figures 3.10 – 3.11 (colonne de droite). Une fois la carte des occultations améliorée, on peut à nouveau densifier la carte de disparité en diffusant la disparité vers la droite. Le résultat de cette densification est présentée dans la colonne de droite de la figure 4.12. Le tableau 3.14 réunit les taux de précision et de rappel pour chaque expérience, avant et après le traitement des détections des zones occultées. Le tableau 3.15 présente pour les deux détections les erreurs d'estimation dans les zones occultées d'après la vérité-terrain, obtenues à l'issue de la densification. Elles sont comparées aux scores initiaux.

Paire	Tsukuba	
Précision pixellique ($h_t = 1$)		
Algorithme sur-relaxé	43,75% / 43,16%	44,15% / 45,18%
Algorithme accéléré	29,11% / 33,90%	29,49% / 35,41%
Précision sous-pixellique ($h_t = 0{,}5$)		
Algorithme sur-relaxé	56,72% / 31,39%	56,97% / 32,12%
Algorithme accéléré	44,42% / 23,22%	44,58% / 23,71%

Paire	Venus	
Précision pixellique ($h_t = 1$)		
Algorithme sur-relaxé	10,56% / 36,25%	10,06% / 37,68%
Algorithme accéléré	6,68% / 23,48%	6,96% / 24,97%
Précision sous-pixellique ($h_t = 0{,}5$)		
Algorithme sur-relaxé	16,17% / 21,98%	15,77% / 22,31%
Algorithme accéléré	14,16% / 17,44%	14,41% / 18,29%

Paire	Teddy	
Précision pixellique ($h_t = 1$)		
Algorithme sur-relaxé	21,80% / 51,12%	23,02% / 59,11%
Algorithme accéléré	16,43% / 37,77%	17,95% / 46,08%
Précision sous-pixellique ($h_t = 0{,}5$)		
Algorithme sur-relaxé	33,57% / 36,40%	36,14% / 45,23%
Algorithme accéléré	29,02% / 30,46%	32,01% / 38,31%

Paire	Cones	
Précision pixellique ($h_t = 1$)		
Algorithme sur-relaxé	25,81% / 55,95%	26,12% / 58,49%
Algorithme accéléré	23,69%/ 47,70%	24,30% / 51,00%
Précision sous-pixellique ($h_t = 0{,}5$)		
Algorithme sur-relaxé	36,88% / 38,83%	38,04% / 41,40%
Algorithme accéléré	37,54% / 38,67%	38,64% / 41,65%

FIGURE 3.14 – Taux de précision et de rappel, avant (colonne de gauche) et après le traitement des occultations (colonne de droite).

Paire	Tsukuba			Venus		
Précision pixellique ($h_t = 1$)						
Algorithme sur-relaxé	86,52%	65,33%	61,66%	90,47%	68,55%	63,29%
Algorithme accéléré	91,83%	83,59%	79,96%	93,71%	87,48%	80,67%
Précision sous-pixellique ($h_t = 0{,}5$)						
Algorithme sur-relaxé	83,41%	71,72%	70,46%	93,71%	82,94%	81,97%
Algorithme accéléré	87,05%	79,54%	78,28%	95,53%	91,89%	89,56%

Paire	Teddy			Cones		
Précision pixellique ($h_t = 1$)						
Algorithme sur-relaxé	96,08%	79,92%	55,79%	95,08%	74,83%	70,24%
Algorithme accéléré	96,76%	94,59%	78,79%	96,08%	90,03%	84,12%
Précision sous-pixellique ($h_t = 0{,}5$)						
Algorithme sur-relaxé	96,44%	89,01%	76,87%	95,83%	85,68%	82,77%
Algorithme accéléré	96,88%	90,81%	79,40%	96,29%	87,54%	84,05%

FIGURE 3.15 – Erreur d'estimation dans les zones occultées, avant et après le traitement des occultations. Pour chaque expérience, on mesure l'erreur pixellique dans les zones occultées dans la première estimation de la disparité (colonne de gauche), après la détection des occultations par saturation de la pente (colonne du milieu) et après l'amélioration de ces détections (colonne de droite).

3.6 Discussion

3.6.1 Résultats

Estimation de la disparité En dehors des zones occultées, les cartes de disparité obtenues présentent les mêmes caractéristiques que celles obtenues grâce à une régularisation TV, c'est-à-dire des cartes constantes par morceaux, avec des contours simples et nets. Ainsi, certains détails de la scène ne peuvent être restitués, comme le bras de la lampe dans Tsukuba, dont les deux branches ont fusionné du fait de la régularité.

Avant le traitement des occultations, les bords gauches des objets présentent une rampe de disparité, qui est contrainte par la contrainte de visibilité. Cette contrainte produit également des artéfacts qui sont dus à la non-préservation de l'ordre dans la scène, qui est visible au niveau des pinceaux dans Cones ou le fil de la lampe de Tsukuba.

On notera également des effets de bords importants sur la partie gauche de la scène. Dans cette partie de l'image, l'occultation hors-champ conduit à des coûts de corrélation non significatifs (on les a choisis constants égaux à $g_{\max} = 100$). Néanmoins, on constate dans les figures 3.10 – 3.11 que ces erreurs sont diffusées en-dehors de cette région.

Détection des occultations Qualitativement, les occultations détectées présentent la même forme que celles de la vérité-terrain, ce qui tend à montrer que le modèle considéré est correct. En particulier, la carte dense produite induit des régions où la pente horizontale vaut 1, et celles-ci sont corrélées avec les zones occultées.

Néanmoins, les régions extraites par la saturation de la pente ne sont pas toujours correctement placées. C'est le cas en particulier de la paire Venus, où l'occultation du fond de la scène est légèrement décalée sur la gauche. On observe également des détections non significatives, car, les cartes étant quantifiées, un saut de disparité se produit même en cas de disparité variant de manière douce (comme les plans inclinés dans Venus ou le toit dans Cones) et ne correspondent pas à de réelles occultations. Or, dans le cas de la précision pixellique, celles-ci sont systématiquement détectées comme telles car toute discontinuité de la scène correspond à un saut d'au moins un pixel. Dans le cas de la précision sous-pixellique, il peut y avoir dans ces cas des sauts d'un demi-pixel, qui ne sont, eux, pas considérés comme des occultations.

Post-traitement des zones occultées On observe pour Teddy un autre phénomène déjà évoqué, qui est celui des alignements verticaux des lignes de niveaux. Ainsi, pour l'occultation induite par le toit de la maison, qui présente un bord oblique dans le repère de l'image, la régularité verticale tend à déplacer les discontinuités, et donc en particulier les lignes de saturation de la pente. La région affectée par ce phénomène est donc globalement plus large que la région effectivement occultée, tandis que la carte de saturation présente des trous. Dans ce cas, le post-traitement proposé permet d'améliorer de manière satisfaisante les détections. Plus précisément, le taux de précision gagne en moyenne +0,77% et le taux de rappel +3,33%. Il n'est pas étonnant de constater un gain plus important du rappel que de la précision, car le post-traitement vise à améliorer la densité de la détection dans les zones occultées, ce qui conduit en théorie (si la première détection est de bonne qualité) à rajouter des vrais positifs. On notera que, mis à part la paire Venus, les trois autres paires bénéficient, quelle que soit l'expérience considérée, de ce post-traitement.

Densification On peut vérifier les gains en terme d'estimation de la disparité de chacune de ces étapes de traitement dans le tableau 3.15 : pour la précision pixellique, on gagne en moyenne +12,74% dans les taux d'erreur pixellique après la première détection des occultations, puis +8,73% une fois ces détections améliorées, ce qui équivaut à un gain total de +21,47%. Pour la précision sous-pixellique, le gain est moins important (respectivement +9,46%, +4,47% et +13,94%). L'amélioration la plus importante concerne la paire Teddy (précision pixellique, carte obtenue par l'algorithme sur-relaxé), où l'on passe de 96,08% d'erreur pixellique à 55,79%, soit une erreur pratiquement divisée par deux.

3.6.2 Différences entre les deux algorithmes proposés

Équivalence des problèmes relaxé (3.13) et fortement convexe (3.17) Nous avons démontré l'équivalence entre le problème primal-dual continu (3.11)

$$\min_{v\in\mathcal{C}} \sup_{\phi\in\mathcal{K}} \left\{ \int_{\Omega\times\mathbb{R}} \phi\, Dv + \int_{\Omega\times\mathbb{R}} h^{\mathrm{vis}}(Dv) \right\}$$

et le problème fortement convexe (3.14)

$$\min_{w\in\mathcal{D}} F(w) + \frac{1}{2}\,\|w\|^2_{\mathrm{L}^2(\Omega\times \mathrm{I}_{\mathrm{disp}})}.$$

La discrétisation de ces deux problèmes n'en préserve visiblement pas l'équivalence. Dans la figure 3.8, on observe ainsi des différences dans le fond de la scène Venus.

(a) $\mu = 50/255$ (b) $\mu = 15/255$ (c) $\mu = 100/255$

FIGURE 3.16 – Influence du pamarètre μ sur la paire Tsukuba. Lorsque μ est trop faible (b), la disparité est trop régulière (voir par exemple sous la table). Lorsque μ est trop grand (c), la disparité paraît bruitée car elle n'est pas suffisamment régularisée. On retrouve en revanche certains détails perdus lorsque la régularisation est trop importante (comme le bras de la lampe).

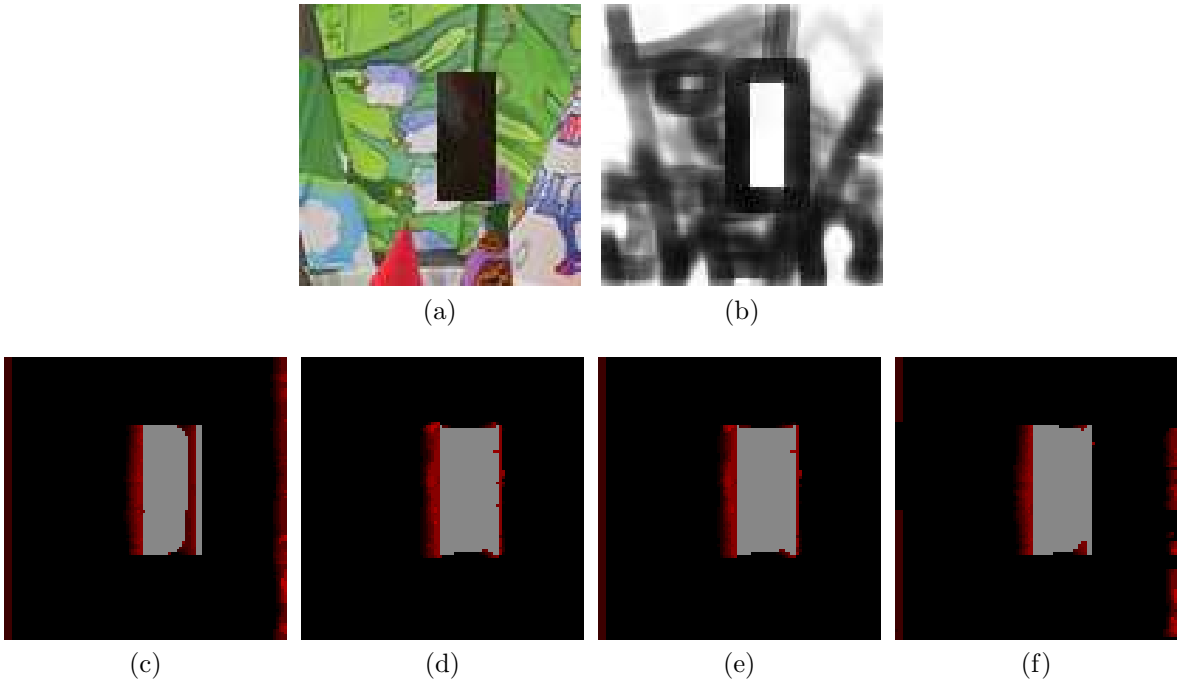
On observe également des différences dans les scores considérés pour mesurer la performance des algorithmes testés. Ainsi, qu'il s'agit de la carte de la disparité ou de la carte des occultations, les résultats diffèrent selon la méthode employée. Plus précisément, sauf pour quelques cas particuliers (comme la paire Cones dans le cas de la précision sous-pixellique), l'algorithme accéléré conduit à des erreurs d'estimation de la disparité plus importantes (jusqu'à une différence de 1,81%). Les taux de précision et de rappel sont également systématiques moins élevés, ce qui se traduit par une estimation moins bonne des occultations (à l'unique exception de Cones en précision sous-pixellique, où la différence pour le taux de précision est de 0,66% avant amélioration des détections). Il est en revanche indubitable que l'algorithme accéléré présente généralement une convergence plus rapide que l'algorithme sur-relaxé (à l'exception notable des paires Tsukuba et Venus pour la précision pixellique). Ainsi, le nombre d'itérations est réduit de moitié pour au moins deux expériences (Cones en précision pixellique et Venus en précision sous-pixellique). L'évolution du critère d'arrêt montre une allure globalement décroissante pour l'algorithme accéléré moins importance que celle de l'algorithme sur-relaxé, mais qui est localement oscillante, permettant d'atteindre plus rapidement le seuil de tolérance.

Influence de la sur-relaxation La sur-relaxation introduite dans l'algorithme primal-dual conduit expérimentalement à une convergence plus rapide de l'algorithme. Ainsi, le *primal-dual gap*, qui, rappelons-le, ne dépend pas de l'algorithme mais du problème considéré, montre une convergence vers zéro plus rapide lorsque $\rho = 1{,}95$ (voir courbe cyan dans la figure 6.6). Le nombre d'itérations est pratiquement réduit de moitié (sauf pour les paires Teddy et Cones en précision sous-pixellique, où nous avons arrêté les itérations au bout de 10 000 itérations). En outre, lorsque le critère d'arrêt est activé, les cartes obtenues sont dans les deux cas les mêmes.

3.6.3 Choix des paramètres

Influence du paramètre de pondération μ Le paramètre μ permet de choisir les poids respectifs du terme d'attache aux données et du terme de régularisation. Dans [12], il est suggéré de le choisir égal à 50/255, ce que nous faisons car les résultats obtenus nous paraissent convenables.

La valeur de ce paramètre est cruciale. Si μ est choisi trop faible, alors c'est la



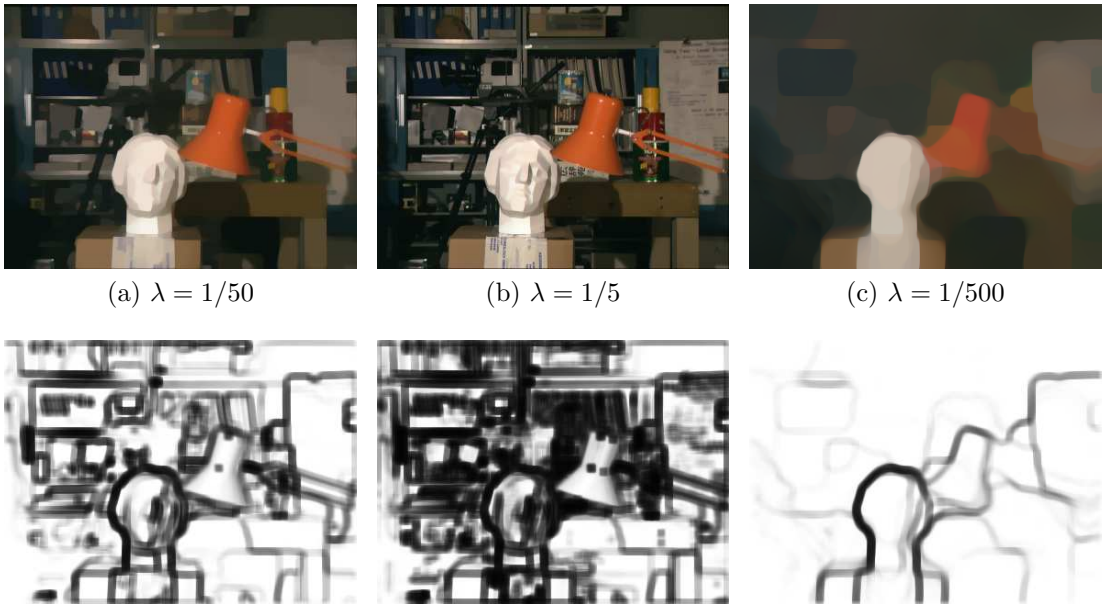
(a) (b) (c) (d) (e) (f)

FIGURE 3.17 – Limitation de l'adhérence à l'aide du coefficient variable α. Le masque transparent rouge montre les erreurs. (a) Image de référence. (b) Valeurs prises par le coefficient variable α^h. Plus le pixel est sombre, moins la corrélation de gradient compte. (c) Corrélation de couleurs. (d) Corrélation de gradient. (e) Combinaison convexe des deux corrélation, avec $\alpha = 0.9$ fixe [13, 14]. (f) Combinaison convexe des deux corrélation, avec α^h variant entre 0 et 1. On voit que lorsque le gradient est utilisé, de l'adhérence (au moins sur le bord droit du rectangle) apparaît.

régularisation qui l'emporte et les cartes obtenues sont trop régulières (figure 3.16(b)). On voit par exemple les effets de cette sur-régularisation sous la table, ou juste au-dessus du bras de la lampe. Le fond de la scène paraît en ravanche plus satisfaisant, car la disparité y présente moins de variations (dans la vérité-terrain, elle est constante). Au contraire, si μ est trop grand (figure 3.16(c)), alors la disparité n'est pas suffisamment régulière. Cela conduit à des cartes de disparité qui paraissent bruitées. Ce manque de régularité peut cependant améliorer l'estimation de la disparité dans des cas spécifiques, comme pour le bras de la lampe. Dès que la régularisation est un peu forte, alors l'espace entre deux barres disparaît.

On constate donc que le choix de ce paramètre est difficile et principalement empirique. La valeur optimale dépend de la scène considérée, mais elle ne sera pas toujours adaptée aux différentes parties de la scène. Il faut donc généralement trouver un équilibre entre une bonne régularisation et la perte de certains détails. Néanmoins, des expériences montrent que les résultats sont relativement peu sensibles à la valeur de μ, en ce sens qu'elle doit beaucoup varier avant que les résultats ne se dégradent de manière significative.

Influence de la pondération variable α Le terme d'attache aux données introduit dans le paragraphe 3.1.1 a été choisi pour tirer parti de l'efficacité de la corrélation combinée utilisée dans [13] et pour limiter le phénomène d'adhérence. Pour justifier cette dernière affirmation, on propose de comparer la performance de ces deux corrélations sur un exemple synthétique, composé d'un rectangle texturé sur un fond texturé. On



(a) $\lambda = 1/50$ (b) $\lambda = 1/5$ (c) $\lambda = 1/500$

(d) $\lambda = 1/50$ (e) $\lambda = 1/5$ (f) $\lambda = 1/500$

FIGURE 3.18 – Influence de la régularisation ROF de l'image sur la valeur de α^h. Ligne du haut : lissage de l'image pour différentes valeurs du paramètre λ. Ligne du bas : le coefficient α^h correspondant. On voit que quand (b) λ est trop grand, α est petit pratiquement partout (e), la corrélation sera donc principalement basée sur la couleur. Quand (c) λ est trop faible, alors α^h sera plutôt grand (f) et la corrélation reposera principalement sur le gradient, même près des discontinuités de la scène.

voit dans la figure 3.17 que, lorsque la corrélation de gradient est utilisée systématiquement sur toute la scène (3.17(d) et (e)), de l'adhérence apparaît, au moins sur le bord droit du rectangle. Ce n'est pas le cas ni avec la corrélation de couleurs (3.17(c)), ni avec le nouveau terme que nous avons introduit (3.17(f)). Près des bords du rectangle, le coefficient α^h est en effet choisi de sorte que la corrélation se fait principalement sur la couleur (figure 3.17(b)). En comparant la corrélation couleurs et de notre terme d'attache aux données sur cet exemple synthétique, on voit cependant que ce dernier conserve les performances de la corrélation de gradient sur la moitié droite du rectangle, où la disparité très mal estimée par la corrélation de couleurs.

La conception du coefficient variable α^h repose principalement sur deux composantes : la régularisation ROF de l'image de référence et le choix du paramètre a dans la définition de α^h. Nous allons observer l'influence de ces deux composantes. On rappelle que l'objectif est de concevoir un coefficient variable α^h qui soit proche de 0 près des discontinuités de la scène et proche de 1 ailleurs.

La régularisation ROF choisie dépend d'un paramètre, qui est le poids λ du terme d'attache aux données dans le modèle de débruitage ROF (3.21). La valeur de ce dernier détermine la force de la régularisation. Si la régularisation est trop importante (figure 3.18(b) et (e)), alors beaucoup de variations de l'image sont ignorées, dont celles de la scène, et c'est la comparaison de gradient qui est principalement utilisée, qui introduit de l'adhérence. Si elle est trop faible (figure 3.18(c) et (f)), trop de variations sont détectées et dans ce cas, c'est la comparaison d'intensité qui prime, dont on sait qu'elle est moins performante que la comparaison de gradient loin des discontinuités de scène. Le paramètre λ doit donc être choisi de manière à conserver les discontinuités significatives (c'est-à-dire celles qui coïncident avec les discontinuités de la scène) et

(a) $a = 1/100$ (b) $a = 1/10$ (c) $a = 1/1000$

FIGURE 3.19 – Influence du choix du paramètre a sur la valeur de α^h. On voit que quand (b) a est trop grand, α est petit pratiquement partout, la corrélation sera donc principalement basée sur la couleur. Quand (c) a est trop faible, alors α^h sera plutôt grand et la corrélation reposera principalement sur le gradient, même près des discontinuités de la scène.

éliminer les autres variations (qui correspondent plutôt à de la texture). Ce choix est évidemment difficile car certaines textures présentent des variations d'intensité très importantes alors que certaines discontinuités de scène ne sont pas très visibles.

De la même manière, le choix du paramètre a est également important, car il permet de régler la sensibilité du détecteur de discontinuité : à partir de quelles valeurs de l'amplitude du gradient α devient-il négligeable ? On voit dans la figure 3.19 qu'à nouveau, a doit être choisi ni trop petit, ni trop grand, afin que les gradients significatifs correspondent à des valeurs proches de zéro pour α^h et que les autres pixels conduisent à des valeurs plus grandes.

Influence du seuil s Dans la version continue du problème que l'on étudie, le choix du seuillage s pour l'algorithme sur-relaxé est arbitraire. Cependant, le cas discret ne conserve pas cette propriété. En particulier, on observe expérimentalement qu'un seuillage près de zéro introduit de l'adhérence (les objets apparaissent plus gros qu'ils ne le sont réellement). Ainsi, il semble préférable de privilégier un seuillage près de 1.

Précision sous-pixellique Mis à part le cas de la paire Tsukuba, qui présente une vérité-terrain pixellique, l'introduction de la précision sous-pixellique améliore de manière significative l'estimation de la disparité dans les zones non occultées. Ce gain est particulièrement visiblement lorsque l'on mesure l'erreur sous-pixellique, puisque celle-ci est pratiquement diminuée de moitié.

En revanche, si elle améliore le taux de précision (pour les quatre paires, les meilleurs scores sont obtenus pour l'algorithme sur-relaxé en précision sous-pixellique), la précision suous-pixellique dégrade le taux de rappel. Ces résultats s'expliquent principalement par deux phénomènes. Le premier, qui est le plus visible, concerne les effets de bords que l'on peut observer sur la partie gauche des scènes. Sur le bord gauche des images, la disparité est en effet nulle, puis, au fur et à mesure que l'on parcourt la scène vers la droite, l'attache aux données lui impose de croître horizontalement. La saturation de la pente détecte donc cette région comme occultée. Or, pour la précision pixellique, cette région dépasse celle – non prise en compte – les occultations hors-champ et ce, de manière plus importante qu'avec la précision sous-pixellique. Le taux de précision pour cette dernière est de fait plus élevée lorsque l'on retire ces détections incorrectes. Le taux de rappel plus faible indique quant à lui une sous-détection dans les zones occultées. Celle-ci s'explique par la fait qu'introduire une disparité sous-

pixellique permet de remplacer un saut (horizontal) d'un pixel par deux sauts d'un demi-pixel chacun. Or, le premier est détecté comme occultation, ce qui n'est pas le cas des seconds. C'est le terme d'attache aux données qui détermine le choix de l'un de ces deux scénarios. Ainsi, si l'utilisation d'un saut sous-pixellique permet d'éviter que des discontinuités de scènes dues à la quantification pixellique de la disparité ne soit interprétées comme des occultations, elle permet au contraire à des occultations de passer inaperçues en ne saturant pas la pente horizontale.

3.6.4 Comparaison avec d'autres algorithmes

Comparaison avec KZ2 Comparons la détection d'occultation avec celle de [10], qui, comme on le verra au chapitre suivant, propose une gestion différente de l'occultation. La figure 3.20 présente les occultations détectées par [10] (implémentation de [9]). Commençons par souligner que cette méthode est capable de détecter les occultations hors-champ. Cette bonne performance ne se lira pas sur les différentes statistiques présentées.

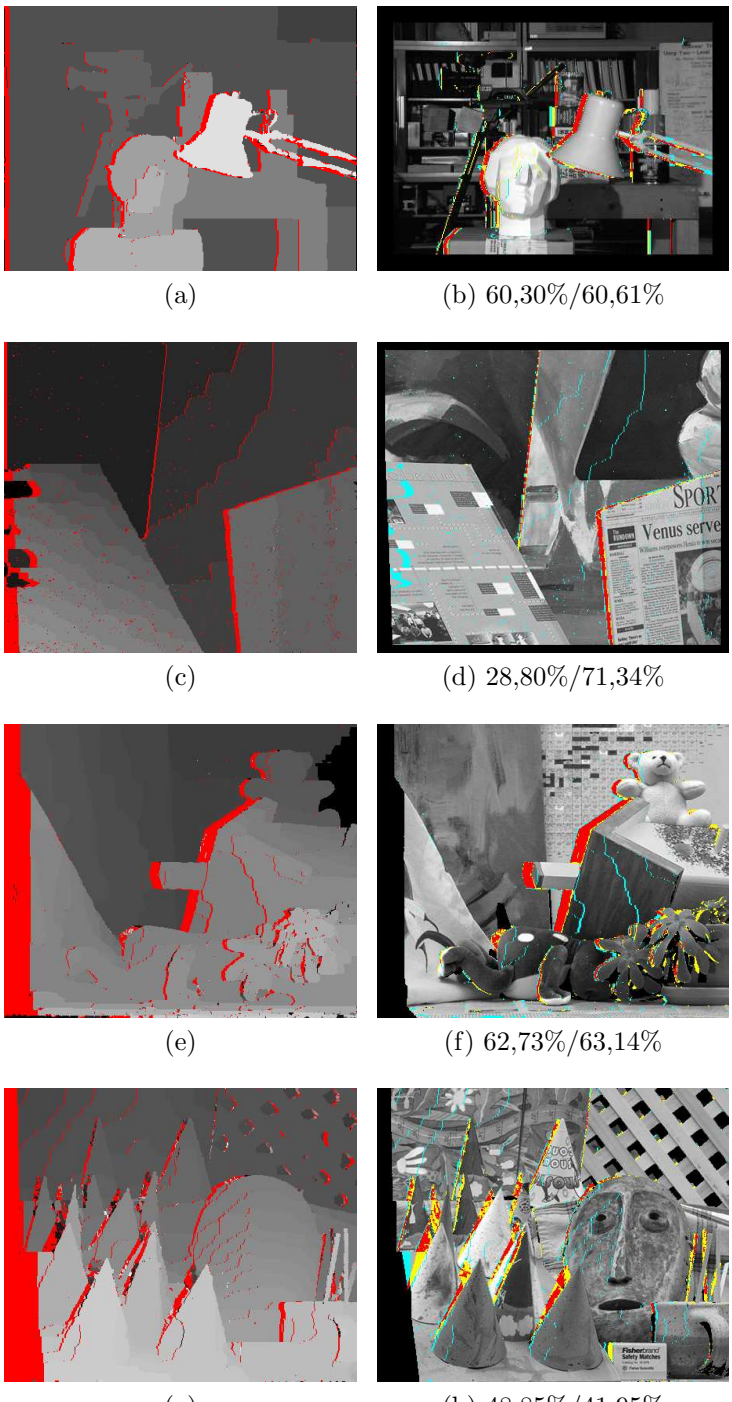
On voit que les occultations détectées par [10] sont globalement mieux localisées que celles que nous obtenons, en particulier sur la paire Venus. Les occultations dans la paire Teddy (notamment celles induites par la maison) sont également remarquablement détectées. Néanmoins, de nombreux faux négatifs peuvent être observés pour Cones. Ainsi, hormis pour la paire Cones, la méthode de Kolmogorov et de Zabih affiche des scores plus élevés.

Ainsi qu'on le verra au chapitre 4, la détection des occultations dans [10] repose sur le choix d'un certain paramètre, noté K. Celui-ci, s'il est choisi trop petit, conduit à détecter trop de pixels occultés, et au contraire, la méthode n'en détecte pas suffisamment s'il est trop grand. Il s'agit en effet du coût de «corrélation » attribué aux pixels occultés.

On voit donc que le choix de ce paramètre est crucial, alors que notre méthode est principalement basée sur la saturation de la pente horizontale, qui ne dépend elle d'aucun paramètre. Elle dépend en revanche du seuil T qui permet, dans l'étape d'amélioration des détections d'occultation, de décider si deux pixels appartiennent au même objet. Ce seuil est néanmoins plus facile à choisir, puisqu'il s'agit de déterminer en moyenne à partir de quel seuil deux couleurs sont significativement différentes. En particulier, il ne dépend pas de la scène, mais de la perception (ou plutôt, de la différentiation) des couleurs. Par ailleurs, cette étape ne corrige qu'une petite partie des occultations, ce qui signifie que notre méthode est moins sensible au choix de ce paramètre (tant qu'il reste raisonnable).

Il faut également noter que, dans la méthode de Kolmogorov et de Zabih, les pixels occultés ne paient qu'un coût de «corrélation» K fixe, alors, dans notre méthode, ceux-ci paient le coût de corrélation associé à la mise en correspondance avec un pixel qui ne peut être leur pixel homologue. Ce coût est donc potentiellement important et peut entraîner des solutions non satisfaisantes mais légèrement moins coûteuses (revoir par exemple l'analyse associée à la figure 3.1). Ainsi, on observe pour la paire Venus de nombreuses détections ponctuelles, qui n'ont aucune signification physique.

Comparaison avec la régularisation TV seule [12] On peut également comparer notre méthode à celle de [12], dans le cas de la régularisation TV. On choisit d'utiliser le même terme d'attache aux données. La fonctionnelle d'énergie est donc la même que celle considérée dans ce chapitre, à l'exception du terme de visibilité qui n'est pas pris



(a) (b) 60,30%/60,61%

(c) (d) 28,80%/71,34%

(e) (f) 62,73%/63,14%

(g) (h) 48,85%/41,95%

FIGURE 3.20 – Détections des occultations par la méthode [10]. Colonne de gauche : carte des occultations générées par [9] (en rouge). Colonne de droite : en rouge, les détections correctes (vrais positifs, TP), en jaune, les détections manquantes (faux négatifs, FN) et en cyan, les détections incorrectes (faux positifs, FP). En légende : les taux de précision et de rappel. De haut en bas : Tsukuba, Venus, Teddy et Cones.

en compte. En l'absence de ce terme, les occultations peuvent être détectées par un filtre LRRL qui vérifie la consistence des cartes de disparité calculées sur la vue de gauche et sur la vue de droite. Plus précisément, deux pixels mis en correspondance dans une des deux cartes devraient le rester dans l'autre carte. Si ce n'est pas le cas, ces pixels sont rejetés (et considérés généralement comme occultés).

La figure 3.21 propose les cartes de telles détections, ainsi que, en légende, les taux de précision et de rappel associés. Dans l'ensemble, ces taux sont meilleurs que ceux obtenus dans notre méthode. Visuellement, on observe des détections mal situées, comme celles localisées sur le buste de Tsukuba. Il faut noter que l'obtention de telles cartes requiert le calcul de deux cartes de disparité, ce qui revient à doubler le temps de calculs. Néanmoins, il est possible de les calculer en parallèle.

Conclusion

Nous avons proposé dans ce chapitre une méthode variationnelle pour la stéréovision binoculaire dont la fonctionnelle d'énergie, qui traduit le modèle de scène considéré, est relativement peu biaisée par la méthode d'optimisation. La régularisation choisie est en effet la régularisation TV, qui est classiquement choisie en traitement d'images car elle lisse tout préservant les discontinuités de la scène. Elle est donc généralement adaptée aux images rencontrées. L'attache aux données est quant à elle une version adaptative d'une corrélation qui a démontré son efficacité [13, 14], modifiée de sorte de limiter l'adhérence qu'elle introduit aux bords des objets. L'apport principal réside toutefois dans le terme de visibilité, que nous avons intégré au terme de régularité. Il permet d'une part de correctement gérer le phénomène d'occultation et d'autre part d'utiliser la méthode de relaxation convexe proposée par [12]. À l'issue d'une première étape d'optimisation convexe, on obtient une carte de disparité dans laquelle on peut détecter les occultations. Un post-traitement permet d'abord d'améliorer la détection de ces zones, puis de densifier la carte de disparité résultante. Par ailleurs, bien que la détection des occultations est légèrement moins performante que celle proposée par Kolmogorov et Zabih dans [10], elle est moins sensible au choix de son paramètre (qui ne dépend pas de la scène, mais du système de couleur choisi), alors que la méthode [10] dépend du choix d'un paramètre, qu'il faut savoir correctement estimer. Enfin, malgré l'introduction d'un algorithme fortement convexe, l'efficacité reste médiocre. La relaxation convexe ajoute en effet une dimension au problème, et la contrainte de pente empêche de mesurer de manière précise la convergence de l'algorithme, ce qui nécessite de recourir à des critères d'arrêt plus heuristiques. Cette complexité est un frein pour la précision sous-pixellique, alors que cette méthode présente l'avantage d'être, en théorie, capable de produire des cartes à cette précision.

Le modèle de scène considéré dans notre méthode est donc plus satisfaisant que la plupart des méthodes globales, car la régularisation est naturelle et les occultations sont traitées comme découlant de manière structurelle de la disparité. Néanmoins, les détections obtenues sont moins bonnes que celles que fournit [10]. Un travail futur consisterait à développer un moyen de faire coïncider les zones occultées et les discontinuités de la scène, en considérant par exemple une pondération spatiale de la régularisation (TV non locale). Les performances de [10] nous conduisent par ailleurs à l'utiliser dans le chapitre suivant pour améliorer la détection des occultations dans des méthodes qui n'en proposent pas mais qui fournissent des estimations (denses ou non) fiables par ailleurs, en adaptant l'algorithme pour la densification de cartes non denses et le raffinement sous-pixellique de cartes pixelliques, avec dans chaque cas une

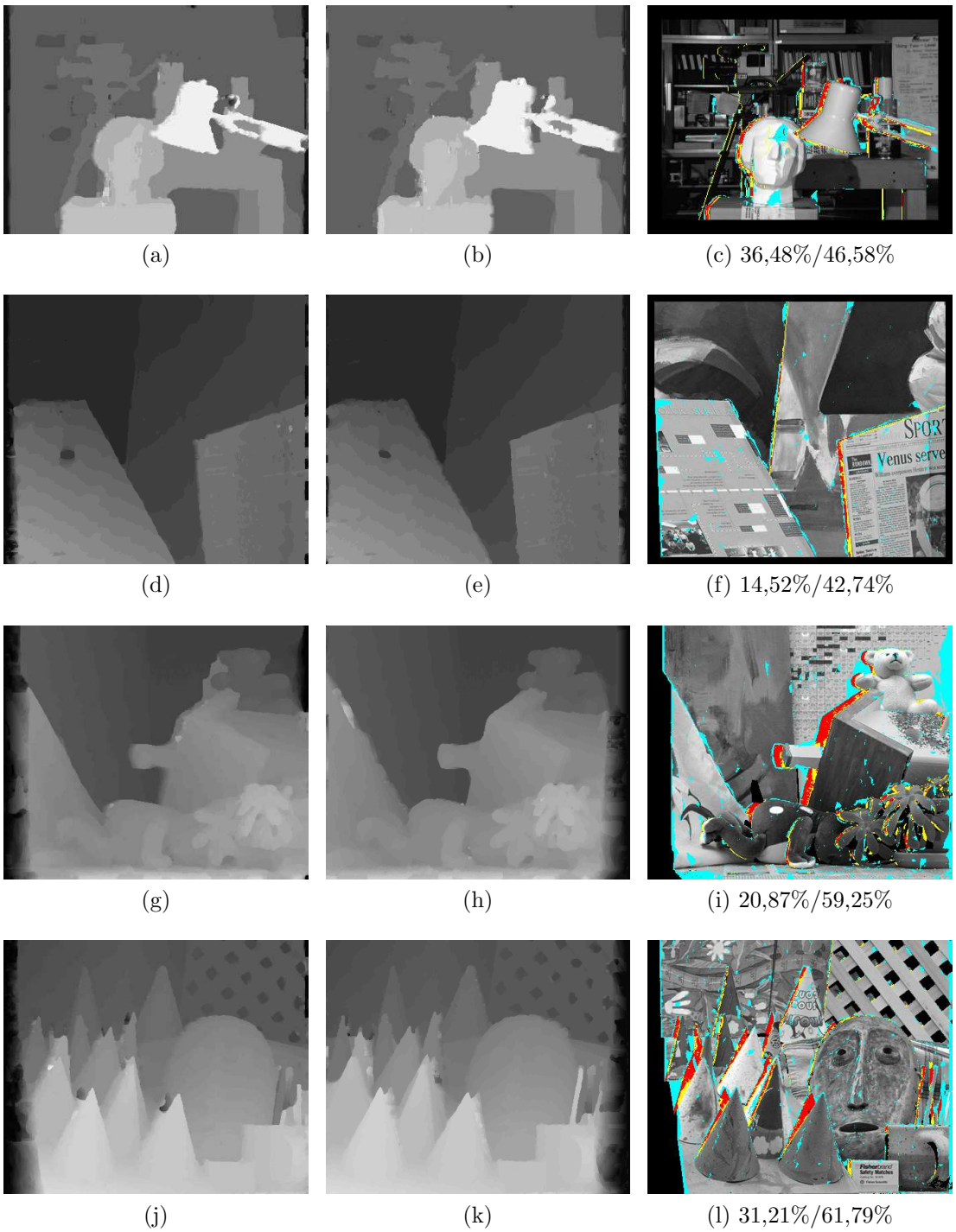


FIGURE 3.21 – Détections des occultations par le filtre LRRL dans le cas d'une carte obtenue par une régularisation TV du volume de coût g. Colonne de gauche : carte de disparité de la vue de gauche. Colonne du milieu : carte de disparité de la vue de droite. Colonne de droite : en rouge, les détections correctes (vrais positifs, TP), en jaune, les détections manquantes (faux négatifs, FN) et en cyan, les détections incorrectes (faux positifs, FP). En légende : les taux de précision et de rappel.De haut en bas : Tsukuba, Venus, Teddy et Cones.

détection de l'occultation.

Références

[1] Aaron F. Bobick and Stephen S. Intille. Large occlusion stereo. *International Journal of Computer Vision*, 33(3) :181–200, 1999.

[2] Guy Bouchitté and Michel Valadier. Integral representation of convex functionals on a space of measures. *Journal of Functional Analysis*, 80(2) :398–420, 1988.

[3] Antonin Chambolle, Vicent Caselles, Daniel Cremers, Matteo Novaga, and Thomas Pock. An introduction to total variation for image analysis. *Theoretical Foundations and Numerical Methods for Sparse Recovery*, 9(263-340) :227, 2010.

[4] Antonin Chambolle and Thomas Pock. On the ergodic convergence rates of a first-order primal–dual algorithm. *Mathematical Programming*, pages 1–35, 2015.

[5] Laurent Condat. A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms. *Journal of Optimization Theory and Applications*, 158(2) :460–479, 2013.

[6] Laurent Condat and Nelly Pustelnik. Segmentation d'image par optimisation proximale. In *XXVème colloque GRETSI (GRETSI 2015)*, 2015.

[7] Lawrence Craig Evans and Ronald F. Gariepy. *Measure theory and fine properties of functions*. CRC press, 2015.

[8] Andreas Klaus, Mario Sormann, and Konrad Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *IEEE International Conference on Pattern Recognition*, volume 3, pages 15–18. IEEE, 2006.

[9] Vladimir Kolmogorov, Pascal Monasse, and Pauline Tan. Kolmogorov and Zabih's graph cuts stereo matching algorithm. *Image Processing On Line*, 4 :220–251, 2014.

[10] Vladimir Kolmogorov and Ramin Zabih. Computing visual correspondence with occlusions using graph cuts. In *IEEE International Conference on Computer Vision*, volume 2, pages 508–515. IEEE, 2001.

[11] Thomas Pock, Daniel Cremers, Horst Bischof, and Antonin Chambolle. An algorithm for minimizing the Mumford-Shah functional. In *IEEE International Conference on Computer Vision*, pages 1133–1140. IEEE, 2009.

[12] Thomas Pock, Daniel Cremers, Horst Bischof, and Antonin Chambolle. Global solutions of variational models with convex regularization. *SIAM Journal on Imaging Sciences*, 3(4) :1122–1145, 2010.

[13] Christoph Rhemann, Asmaa Hosni, Michael Bleyer, Carsten Rother, and Margrit Gelautz. Fast cost-volume filtering for visual correspondence and beyond. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3017–3024. IEEE, 2011.

[14] Pauline Tan and Pascal Monasse. Stereo disparity through cost aggregation with guided filter. *Image Processing On Line*, pages 252–275, 2014.

[15] Philippe Thévenaz, Thierry Blu, and Michael Unser. Image interpolation and resampling. *Handbook of medical imaging, processing and analysis*, pages 393–420, 2000.

[16] Michael Unser. Sampling – 50 years after Shannon. *Proceedings of the IEEE*, 88(4) :569–587, 2000.

[17] Cédric Villani. Intégration et analyse de fourier. *Notes de cours pour l'ENS Lyon*, 6, 2005.

Chapitre 4

Spécification de l'intervalle de disparité par pixel dans la méthode des *graph cuts*

Introduction

En 2001, Vladimir Kolmogorov et Ramin Zabih proposent une méthode globale basée sur l'utilisation de graphes pour minimiser de manière approchée une fonctionnelle d'énergie. L'intérêt de cette méthode réside en deux principales caractéristiques. En premier lieu, elle ne découle pas d'une formulation variationnelle. Autrement dit, la disparité discrète n'est pas la discrétisation d'une fonction continue, elle peut donc en particulier prendre des valeurs non réelles (sans être infinie). Cette propriété permet de définir une étiquette qui marque les pixels comme *occultés*, sans devoir les mettre artificiellement en correspondance avec d'autres pixels (section 4.1). En second lieu, la stratégie de minimisation, qui repose sur l'algorithme des *expansion moves*, peut être gérée de manière efficace en la reformulant comme des recherches de coupures minimales dans un graphe (section 4.2). Ce dernier problème est bien connu depuis Ford et Fulkerson et de nombreux algorithmes existent qui le résolvent efficacement.

Cependant, on verra que les performances de la méthode reposent grandement sur la valeur de certains paramètres, dont le choix peut s'avérer délicat (section 4.6). En outre, l'utilisation des *expansion moves* ne permet pas de minimiser exactement la fonctionnelle, mais seulement de proposer un schéma de décroissance de l'énergie (section 4.3). Si la fonctionnelle ne possède pas de minima locaux, on peut raisonnablement espérer en trouver le minimum global de cette manière. Malheureusement, cette propriété n'est pas assurée car la fonctionnelle n'est pas convexe. Par ailleurs, la représentabilité de l'énergie (section 4.2) introduit des contraintes quant au choix de la fonctionnelle d'énergie. Néanmoins, l'algorithme donne des résultats très satisfaisants, notamment en terme de détection des occultations (voir le chapitre précédent), et dans un temps raisonnable.

L'objet principal de ce chapitre est de proposer une modification de la méthode de Kolmogorov et Zabih qui permette de définir pour chaque pixel un intervalle de disparité adapté (section 4.5). Une telle possibilité permet d'utiliser l'efficacité des coupures de graphes pour densifier des cartes éparses ou de raffiner des cartes à des précisions subpixelliques. On se propose donc dans un premier temps de décrire la méthode originale, dont les différents éléments ont été détaillées dans plusieurs articles [8, 9, 3], ainsi que dans la thèse de Kolmogorov [6]. Cette étude reprend en partie la publication IPOL [7] consacrée à cette méthode. L'objectif est d'offrir la compréhension nécessaire pour envisager les modifications souhaitées, dont les différents résultats expérimentaux sont proposés à la section 4.6.

4.1 Fonctionnelle d'énergie

Dans tout ce qui suit, on reprend l'approche de la méthode KZ2 de [8], déjà détaillée dans l'article IPOL [7]. Toutefois, nous reformulons de manière différente la fonctionnelle, pour faciliter la comparaison avec celle de la méthode proposée au chapitre précédent (dont la version discrète est donnée dans la section 3.3.1).

Plaçons-nous directement dans la formulation discrète. Les pas de discrétisation des images et de l'intervalle de disparité valent 1, ce qui conduit, en conservant les notations du chapitre précédent (paragraphe 3.3.1) à considérer le vecteur $h = (1,1,1)$. Contrairement à la méthode étudiée dans le chapitre précédent, la disparité u^h est à valeurs dans $\mathrm{I}^h_{\mathrm{disp}} \cup \mathtt{occ}$, où $\mathtt{occ}$ est une étiquette particulière destinée à déclarer un

pixel occulté. Le problème de mise en correspondance s'écrit donc

$$\min_{u^h \in (\mathrm{I}^h_{\mathrm{disp}} \times \{\mathtt{occ}\})^{N_x N_y}} \mathrm{E}(u^h)$$

où les images I_L et I_R sont de tailles $N_x \times N_y$. On rappelle que $\mathrm{I}^h_{\mathrm{disp}} = [\![d_{\min} ; d_{\max}]\!] \subset \mathbb{Z}$.

La fonctionnelle considérée possède trois termes :

$$\mathrm{E}(u^h) = \mathrm{E}_{\mathrm{data+occ}}(u^h) + \mathrm{E}_{\mathrm{reg}}(u^h) + \mathrm{E}_{\mathrm{inj}}(u^h).$$

Le premier terme correspond au terme d'attache aux données classique, auquel on ajoute un terme pour prendre en compte les pixels occultés ; le seond terme mesure la régularité de la disparité ; enfin, le dernier terme assure l'injectivité de la mise en correspondance.

4.1.1 Terme d'attache aux données et d'occultation

Ce terme attribue un coût à tout pixel, qui est soit le coût de corrélation si le pixel n'est pas occulté, soit un *coût d'occultation* sinon. Il se décompose donc en deux termes distincts :

$$\mathrm{E}_{\mathrm{data+occ}}(u^h) = \mathrm{E}_{\mathrm{data}}(u^h) + \mathrm{E}_{\mathrm{occ}}(u^h).$$

Terme d'attache aux données Ce terme somme les coûts de corrélation pour chaque pixel non occulté :

$$\mathrm{E}_{\mathrm{data}}(u^h) = \sum_{i=0}^{N_x-1} \sum_{j=0}^{N_y-1} g\Big((i,j),u^h_{i,j}\Big) \Big(1 - \mathbb{1}_{\{\mathrm{occ}\}}(u^h_{i,j})\Big). \tag{4.1}$$

La fonction de corrélation g est choisie de manière suivante pour la corrélation AD :

$$g_{\mathrm{AD}}(x,t) = N_{I_\mathrm{L},I_\mathrm{R}}(x,x-t),$$

tandis que pour la corrélation SD, on a

$$g_{\mathrm{SD}}(x,t) = N_{I_\mathrm{L},I_\mathrm{R}}(x,x-t)^2.$$

Ces deux corrélations sont définies à partir de la norme N, qui vaut dans le cas pixellique

$$N^{\mathrm{pix}}_{I_\mathrm{L},I_\mathrm{R}}(x,x-t) = \frac{1}{3} \sum_{c\in\{\mathrm{r,g,b}\}} \min\Big\{|I^c_\mathrm{L}(x) - I^c_\mathrm{R}(x-t)|,30\Big\}$$

ce qui revient à calculer dans chaque canal couleur une différence en valeur absolue seuillée, puis de fusionner les trois coûts obtenus à l'aide d'une moyenne. L'utilisation d'un seuil est fréquent (c'est le cas par exemple dans [10]) : lorsque la dissimilarité est importante pour tout couple de pixels, elle évite qu'une mise en correspondance soit privilégiée par rapport à une autre alors qu'aucune n'est pertinente. Dans le cas subpixellique, c'est une variante de la mesure de Birchfield et Tomasi [1] qui est utilisée :

$$N^{\mathrm{subpix}}_{I_\mathrm{L},I_\mathrm{R}}(x,x-t) = \frac{1}{3} \sum_{c\in\{\mathrm{r,g,b}\}} \min\Big\{(\|I^c_\mathrm{L}(x) - I^c_\mathrm{R}(x-t)\|_{\mathrm{BT}},30\Big\}.$$

La dissimilarité dans chaque canal est mesurée par la fonction

$$\|I^c_\mathrm{L}(x) - I^c_\mathrm{R}(y)\|_{\mathrm{BT}} = \max\Big\{0,I^c_\mathrm{L}(x) - (I^c_\mathrm{R})^{\max}(y),(I^c_\mathrm{R})^{\min}(y) - I^c_\mathrm{L}(x)\Big\}$$

qui calcule la distance de l'intensité $I_{\mathrm{L}}^c(x)$ au segment $\Big[\,(I_{\mathrm{R}}^c)^{\min}(y)\,;(I_{\mathrm{R}}^c)^{\max}(y)\,\Big]$, où les images $(I_{\mathrm{R}}^c)^{\max}$ et $(I_{\mathrm{R}}^c)^{\min}$ sont définies respectivement par

$$(I_{\mathrm{R}}^c)^{\max}(y) = \max_{r\in\{(0,0),(\pm1,0),(0,\pm1)\}} \Big\{ \tilde{I_{\mathrm{R}}}^c(y+r/2) \Big\}$$

et

$$(I_{\mathrm{R}}^c)^{\min}(y) = \min_{r\in\{(0,0),(\pm1,0),(0,\pm1)\}} \Big\{ \tilde{I_{\mathrm{R}}}^c(y+r/2) \Big\}$$

avec

$$\tilde{I_{\mathrm{R}}}^c(y+r/2) = \frac{1}{2}\Big(I_{\mathrm{R}}^c(y) + I_{\mathrm{R}}^c(y+r)\Big)$$

l'interpolation bilinéaire du canal couleur c de l'image I_{R}. Autrement dit, $(I_{\mathrm{R}}^c)^{\max}(y)$ (respectivement $(I_{\mathrm{R}}^c)^{\min}(y)$) correspond à la valeur maximale (resp. minimale) prise par l'interpolée $\tilde{I_{\mathrm{R}}}^c$ sur le carré de côté 1 centré en y. Dans [1], les auteurs montrent que l'utilisation de cette mesure de corrélation réduit le biais dû à l'échantillonnage (eux se contentant de considérer $r \in \{(0,0),(0,\pm 1)\}$). En effet, si le pixel homologue se trouve dans ce carré mais pas sur la grille d'échantillonnage, alors le coût de corrélation restera malgré tout faible.

Remarque : Le coût de corrélation présenté ici est celui proposé par les auteurs de l'article original [8]. Il a été repris et implanté dans l'article IPOL consacré à cette méthode [7]. C'est la raison pour laquelle il est utilisé dans ce chapitre. Toutefois, rien dans la méthode n'empêche l'utilication d'une autre mesure de dissimilarité.

Terme d'occultation Le terme d'occultation associe à chaque pixel occulté un coût d'occultation fixe $K \geq 0$

$$\mathrm{E}_{\mathrm{occ}}(u^h) = \sum_{i=0}^{N_x-1}\sum_{j=0}^{N_y-1} K\,\mathbb{1}_{\{\mathrm{occ}\}}(u_{i,j}^h). \tag{4.2}$$

Le choix de la pénalité K sera explicité ultérieurement.

4.1.2 Terme de régularisation

Le terme de régularisation pénalise toute variation de la disparité. Cette pénalité ne dépend pas de l'amplitude de cette variation, mais de la variation d'intensité au niveau de la variation de disparité :

$$\begin{aligned}\mathrm{E}_{\mathrm{reg}}(u^h) = &\sum_{i=1}^{N_x-1}\sum_{j=0}^{N_y-1} R_{\lambda_1,\lambda_2}\Big((i,j),(i,j-1),u_{i,j}^h\Big) + R_{\lambda_1,\lambda_2}\Big((i,j),(i,j-1),u_{i,j-1}^h\Big)\\ &+ \sum_{i=0}^{N_x-1}\sum_{j=1}^{N_y-1} R_{\lambda_1,\lambda_2}(u_{i,j}^h,u_{i,j-1}^h)\,\mathbb{1}_{\{0\}}\Big(\mathcal{N}(u_{i,j}^h,u_{i,j-1}^h)\Big)\end{aligned} \tag{4.3}$$

où la pénalité R_{λ_1,λ_2} est définie par

$$R_{\lambda_1,\lambda_2}(p,p',u) = \begin{cases} 0 & \text{si } u = \mathtt{occ} \\ \lambda_1 & \text{si } u \neq \mathtt{occ} \text{ et } N_{I_{\mathrm{L}},I_{\mathrm{L}}}^{\mathrm{pix}}(p,p') \leq 8 \\ & \text{et } N_{I_{\mathrm{R}},I_{\mathrm{R}}}^{\mathrm{pix}}(p+u,p'+u) \leq 8 \\ \lambda_2 & \text{sinon} \end{cases}$$

avec $0 \leq \lambda_2 \leq \lambda_1$. Autrement dit, la variation de disparité est moins pénalisée si elle s'accompagne d'une variation d'intensité importante dans la vue de gauche ou dans la vue de droite.

4.1.3 Terme d'injectivité

Enfin, le terme d'injectivité impose que deux pixels de l'image de référence possèdent deux pixels homologues différents. Cette contrainte se traduit sous la forme suivante :

$$\mathrm{E}_{\mathrm{inj}}(u^h) = \sum_{i=0}^{N_x-1} \sum_{j=0}^{N_y-1} \sum_{j'=0}^{N_y-1} \chi_{\mathbb{R}\setminus\{j-u^h_{i,j}\}}(j' - u^h_{i,j'}). \tag{4.4}$$

Remarque : La contrainte d'injectivité tient partiellement compte de l'analyse de l'occultation proposée dans le chapitre 2 en ce sens que si la contrainte de visibilité (sur la pente horizontale) est respectée, alors tous les pixels saturant la pente doivent être déclarés occultés à cause de ce terme. En effet, ces pixels sont tous mis en correspondance avec le même pixel, ce qui est ici interdit.

4.2 Représentation d'une énergie par un graphe

On va montrer dans cette section qu'il est possible d'utiliser les graphes pour résoudre de manière efficace certains problèmes d'optimisation d'énergies. Pour cela, on introduit le concept de représentation d'une énergie par un graphe.

4.2.1 Coupure minimale et flot maximal

Graphe orienté et pondéré Commençons par introduire quelques définitions utiles pour la suite. On appelle *graphe* $\mathcal{G}$ le couple $(\mathcal{V},\mathcal{E})$ où $\mathcal{V}$ est un ensemble dont les éléments sont appelés *sommets* (*vertices* en anglais) et $\mathcal{E} \subset \mathcal{V}^2$ est l'ensemble des arcs *orientés* et *pondérés* du graphe (*edges* en anglais). L'orientation des arcs implique de distinguer les deux arcs (a_1,a_2) et (a_2,a_1) si $a_1 \neq a_2 \in \mathcal{V}$ et la pondération consiste à leur attribuer une valeur $c_{\mathcal{G}}(a_1,a_2) \in]\,0\,;+\infty\,]$, appelé *capacité* de l'arc. On ignore les arcs de capacité nulle, ce qui revient à les supprimer de l'ensemble $\mathcal{E}$. Notons qu'on accepte les arcs de capacité infinie.

Pour plus d'informations sur la théorie des graphes, on pourra se reporter à [2].

Coupure d'un graphe Désormais, on suppose que le graphe $\mathcal{G}$ possède deux sommets distincts s et t, appelés respectivement *source* et *puits* du graphe. Une *coupure*[1] du graphe désigne alors toute partition $(\mathcal{V}^s,\mathcal{V}^t)$ des sommets telle que la source s (resp. le puits t) appartienne au sous-ensemble $\mathcal{V}^s$ (resp. $\mathcal{V}^t$). Le *coût* d'une coupure $(\mathcal{V}^s,\mathcal{V}^t)$ est donnée par la somme des capacités des arcs $(a_1,a_2) \in \mathcal{V}$ partant de $a_1 \in \mathcal{V}^s$ et aboutissant à $a_2 \in \mathcal{V}^t$. Elle est notée $C(\mathcal{V}^s,\mathcal{V}^t)$ et donnée par la formule

$$C_{\mathcal{G}}(\mathcal{V}^s,\mathcal{V}^t) = \sum_{\substack{(a_1,a_2)\in\mathcal{E} \\ a_1\in\mathcal{V}^s, a_2\in\mathcal{V}^t}} c_{\mathcal{G}}(a_1,a_2).$$

On appelle *coupure minimale* d'un graphe la coupure de coût minimal.

Flot d'un graphe On appelle *flot*[2] du graphe $\mathcal{G}$ toute fonction $\Phi : \mathcal{E} \to [\,0\,;+\infty\,]$ vérifiant les deux conditions suivantes, la première sur les arcs, appelée *contrainte de capacité* :

$$\forall\,(a_1,a_2) \in \mathcal{E}, \qquad 0 \leq \Phi(a_1,a_2) \leq c_{\mathcal{G}}(a_1,a_2)$$

1. *Cut* en anglais, si bien qu'on rencontre parfois le terme *coupe* également.
2. *Flow* en anglais.

et la seconde sur les sommets

$$\forall\, a \in \mathcal{V} \setminus \{s,t\}, \qquad \sum_{(a,a')\in\mathcal{E}} \Phi(a,a') = \sum_{(a',a)\in\mathcal{E}} \Phi(a',a).$$

Cette dernière propriété est appelée *loi de conservation du flot* (aussi connue sous le nom de loi de Kirchhoff), car elle assure que le flot aboutissant en a est égal au flot qui en est issu. La *valeur* d'un flot est alors défini par la valeur commune

$$V(\Phi) = \sum_{(s,a)\in\mathcal{E}} \Phi(s,a) = \sum_{(a,t)\in\mathcal{E}} \Phi(a,t).$$

On appelle *flot maximal* d'un graphe le flot de valeur maximale.

Max-Flow/Min-Cut Le problème de recherche de la coupure minimale est en réalité dual à celui de recherche du flot maximal. Le théorème de Ford-Fulkerson [4] assure que la valeur du flot maximal d'un graphe est égale au coût de sa coupure minimale. Par ailleurs, le flot maximal peut être obtenu grâce à l'algorithme de Ford-Fulkerson. Celui-ci est basé sur le principe des *chemins augmentants*. On commence par rechercher un chemin orienté dans le graphe $\mathcal{G}$ liant la source au puits, c'est-à-dire $m \in \mathbb{N}$ sommets $(a_{i_m}) \in \mathcal{V}^m$ tels que

$$(s,a_{i_1}) \in \mathcal{E}, \qquad \forall\, k \in [\![\, 1\,;\, m-1 \,]\!], \quad (a_{i_k},a_{i_{k+1}}) \in \mathcal{E} \qquad \text{et} \qquad (a_{i_m},t) \in \mathcal{E}$$

et que ces arcs soient de capacité strictement positive. Dorénavant, on dira qu'un chemin reliant un sommet à un autre est de capacité strictement positive si chacun des arcs qui le composent sont de capacité strictement positive. En choisissant le minimum de ces capacités $c_{\min}$, on peut alors définit un flot en posant

$$\forall\, (a_1,a_2) \in \mathcal{E}, \qquad \begin{cases} \Phi(a_1,a_2) = c_{\min} & \text{si } (a_1,a_2) = (s,a_{i_1}) \\ \Phi(a_1,a_2) = c_{\min} & \text{si } (a_1,a_2) = (a_{i_k},a_{i_{k+1}}), k \in [\![\, 1\,;\, m-1 \,]\!] \\ \Phi(a_1,a_2) = c_{\min} & \text{si } (a_1,a_2) = (a_{i_m},t) \\ \Phi(a_1,a_2) = 0 & \text{sinon.} \end{cases}$$

On construit ensuite un nouveau graphe, appelé *graphe résiduel* de $\mathcal{G}$ et noté $\mathcal{G}_\Phi$. Ce graphe possède les mêmes sommets et arcs que le graphe initial $\mathcal{G}$, mais ses arcs sont pondérés de la manière suivante :

$$\forall\, (a_1,a_2) \in \mathcal{V}^2, \qquad c_{\mathcal{G}_\Phi}(a_1,a_2) = \begin{cases} c_{\mathcal{G}}(a_1,a_2) - \Phi(a_1,a_2) & \text{si } (a_1,a_2) \in \mathcal{E} \\ \Phi(a_1,a_2) & \text{si } (a_2,a_1) \in \mathcal{E} \\ 0 & \text{sinon.} \end{cases}$$

Si la capacité d'un arc est définie deux fois par cette définition (si les deux arcs (a_1,a_2) et (a_2,a_1) existent), alors on somme les deux capacités. La contrainte de capacité assure que les capacités ainsi définies sont positives ou nulles. On recommence alors le procédé, en recherchant un nouveau chemin liant la source et le puits suivant des arcs de capacité $c_{\mathcal{G}_\Phi}$ strictement positive (chemin augmentant). Si un tel chemin existe, alors on définit le flot correspondant et on construit le graphe résiduel associé ; sinon, alors la somme des flots définis précédemment donne le flot maximal et la coupure minimale est donnée par $(\mathcal{V}^s,\mathcal{V}^t)$, où $\mathcal{V}^s$ est défini comme l'ensemble des sommets a de $\mathcal{G}$ pour lesquels il existe un chemin de capacité strictement positive $c_{\mathcal{G}_\Phi}$ reliant la source s au sommet a. L'ensemble $\mathcal{V}^t$ est alors donné par $\mathcal{V} \setminus \mathcal{V}^s$.

Les algorithmes basés sur le principe des flots augmentants sont généralement désignés sous le nom d'algorithmes de type Ford-Fulkerson [5]. Leur complexité dans le pire des cas est en $O(m\,n^2)$, où n est le nombre de sommets dans le graphe et m le nombre d'arcs [3].

4.2.2 Représentabilité d'une énergie

On reprend ici une partie de la théorie de la représentabilité des fonctions par un graphe, détaillée dans [9].

Représentation d'une fonction Soit E une fonction de n variables binaires $x = \{x_i\}_{i\in[\![1\,;n]\!]} \in \{0,1\}^n$. On dit que G est représentable par un graphe s'il existe un graphe $\mathcal{G} = (\mathcal{V},\mathcal{E})$ comportant une source s et un puits t, ainsi que n sommets $\{a_i\}_{i\in[\![1\,;n]\!]} \subset \mathcal{V}$ tels que, pour tout $x \in \{0,1\}^n$, la quantité $\mathrm{E}(x)$ soit égale à une constante C plus le coût minimal des coupures $(\mathcal{V}^s,\mathcal{V}^t)$ vérifiant $a_i \in \mathcal{V}^s$ si $x_i = 0$ et $a_i \in \mathcal{V}^t$ si $x_i = 1$. On dit alors que le graphe $\mathcal{G}$ représente la fonction E. Notons qu'il n'y a pas unicité de la représentation.

Si le nombre de sommets dans $\mathcal{G}$ est minimal, exactement égal à $n+2$, c'est-à-dire si $\mathcal{V} = \{s,t,\{a_i\}_{i\in[\![1\,;n]\!]}\}$, alors $\mathrm{E}(x)$ est égal à C plus le coût de *la* coupure $(\mathcal{V}^s,\mathcal{V}^t)$ vérifiant $a_i \in \mathcal{V}^s$ si $x_i = 0$ et $a_i \in \mathcal{V}^t$ si $x_i = 1$.

La classe $\mathcal{F}^2$ Soit n un entier naturel non nul. On définit la classe $\mathcal{F}^2$ comme étant l'ensemble des fonctions à n variables binaires s'écrivant comme la somme de fonctions à au plus deux variables binaires :

$$\forall\, x \in \{0,1\}^n, \qquad \mathrm{E}(x) = \sum_{i=1}^{n} \mathrm{E}^i(x_i) + \sum_{i=1}^{n}\sum_{j=1}^{i-1} \mathrm{E}^{i,j}(x_i,x_j)$$

Les fonctions à une variable binaire étant toujours représentables par un graphe, et la représentabilité étant une propriété additive, le théorème suivant donne une condition suffisante sur les termes à deux variables pour assurer la représentabilité de toute fonction E :

Théorème 12 (Théorème $\mathcal{F}^2$, [8, 9]) *Soit E une fonction de la classe $\mathcal{F}^2$, écrit sous la forme*

$$\forall\, x \in \{0,1\}^n, \qquad E(x) = \sum_{i=1}^{n} E^i(x_i) + \sum_{i=1}^{n}\sum_{j=1}^{i-1} E^{i,j}(x_i,x_j)$$

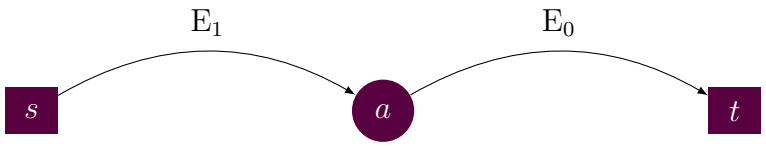
Si pour tout couple $(i,j) \in [\![1\,;n]\!]^2$ tel que $i<j$, le terme $E^{i,j}$ vérifie la condition de sous-modularité

$$E^{i,j}(0,0) + E^{i,j}(1,1) \leq E^{i,j}(0,1) + E^{i,j}(1,0),$$

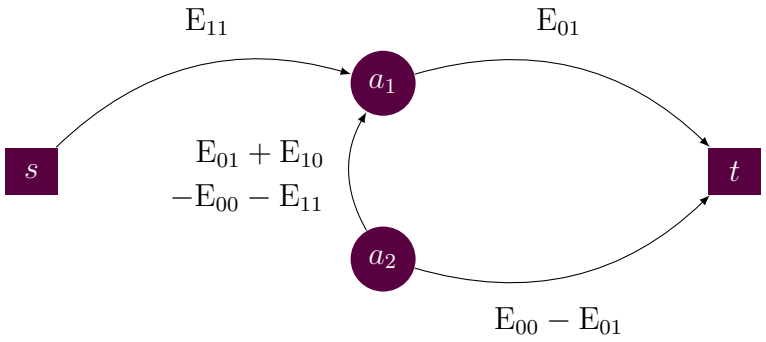
alors la fonction E est représentable par un graphe.

Pour démontrer ce théorème, on construit pour chaque terme un graphe le représentant. Dans [6], on montre que cette condition est en réalité nécessaire.

Représentation d'une fonction d'une variable Pour représenter la fonction E à une variable binaire, il suffit que de poser $\mathrm{E}_0 = \mathrm{E}(0)$ et $\mathrm{E}_1 = \mathrm{E}(1)$. Si ces deux valeurs sont positives, alors il suffit de considérer un sommet a (en plus des deux sommets s et t) et de construire les arcs (s,a) et (a,t), en leur attribuant respectivement les capacités positives E_0 et E_1 (voir figure 4.1(a)). On vérifie que ce graphe représente bien la fonction E, car si $x = 0$, alors $\mathrm{E}(x) = \mathrm{E}_0$ et la coupure associée est la coupure $(\{s,a\},\{t\})$, dont le coût vaut E_0 ; si $x = 1$, alors $\mathrm{E}(x) = \mathrm{E}_1$ et la coupure associée



(a) Fonction d'une variable



(b) Fonction de deux variables

FIGURE 4.1 – Représentation des fonctions à une et deux variables binaires. On suppose que, dans (b), la fonction vérifie la condition de sous-modularité.

est la coupure $(\{s\},\{t,a\})$, dont le coût vaut E_1 (voir figure 4.2(a)). La constante C de la définition est ici nulle.

Dans le cas général, si une des deux quantités E_0 ou E_1 est négative par exemple (mais cela reste valable si elles sont positives toutes les deux), il suffit de considérer la fonction positive $\mathrm{E} - \min\{\mathrm{E}_0,\mathrm{E}_1\}$, qui est représentable par un graphe d'après le paragraphe précédent. Les différentes coupures sont alors de coûts respectifs $\mathrm{E}(x) - \min\{\mathrm{E}_0,\mathrm{E}_1\}$, ce qui montre que la fonction E est représentable par ce même graphe, avec cette fois $C = -\min\{\mathrm{E}_0,\mathrm{E}_1\}$. On notera que, dans ce cas, l'un des arcs construits est de capacité nulle, ce qui permet de l'ignorer.

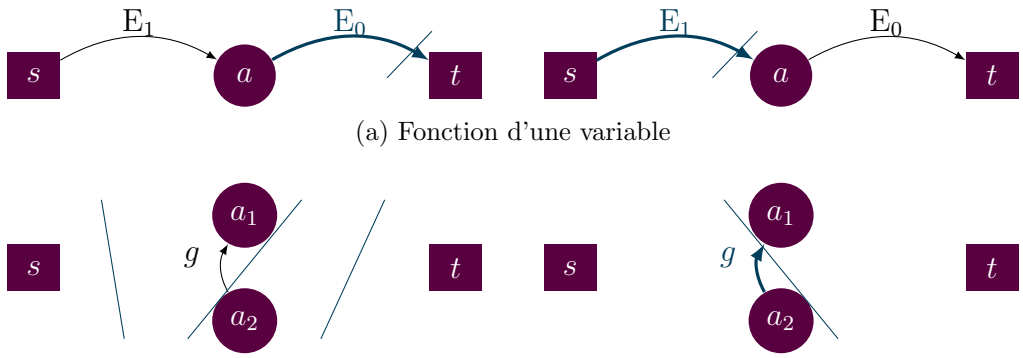
Représentation d'une fonction de deux variables sous-modulaire On suppose que la fonction à deux variables E vérifie la condition de sous-modularité. Pour tout $(a,b) \in \{0,1\}^2$, notons $\mathrm{E}_{ab} = \mathrm{E}(a,b)$. On a alors

$$\mathrm{E}_{00} + \mathrm{E}_{11} \leq \mathrm{E}_{01} + \mathrm{E}_{10}.$$

Remarquons ensuite que E peut se décomposer de la manière suivante :

$$\begin{aligned} \mathrm{E}(x_1,x_2) = &\begin{cases} \mathrm{E}_{01} & \text{si } x_1 = 0 \\ \mathrm{E}_{11} & \text{si } x_1 = 1 \end{cases} \\ &+ \begin{cases} \mathrm{E}_{00} - \mathrm{E}_{01} & \text{si } x_2 = 0 \\ 0 & \text{si } x_2 = 1 \end{cases} \\ &+ \begin{cases} \mathrm{E}_{01} + \mathrm{E}_{10} - \mathrm{E}_{00} - \mathrm{E}_{11} & \text{si } (x_1,x_2) = (1,0) \\ 0 & \text{sinon.} \end{cases} \end{aligned} \tag{4.5}$$

Le premier terme définit une fonction ne dépendant que de la variable binaire x_1, tandis que le second ne dépend que de la variable x_2. Aussi, d'après le point précédent,



(a) Fonction d'une variable

(b) Fonction de deux variables : construction du terme binaire

FIGURE 4.2 – Les différentes coupures possibles dans les graphes construits à la figure 4.1. Dans (b), on ne considère que la représentation du troisième terme dans 4.5. Les coupures sont symbolisées par les lignes bleues, tandis que les arcs bleus sont ceux contribuant au coût de la coupure.

ces deux termes sont représentables par un graphe. Le dernier terme est une fonction à deux variables binaires à valeurs positives d'après la condition de sous-modularité. Pour la représenter par un graphe, il suffit de construire l'arc (a_2,a_1) en la pondérant par la quantité positive $g = \mathrm{E}_{01} + \mathrm{E}_{10} - \mathrm{E}_{00} - \mathrm{E}_{11}$, où les sommets a_1 et a_2 sont respectivement associés aux variables x_1 et x_2. La construction d'un tel graphe est proposée dans la figure 4.1(b). On peut alors vérifier que la seule coupure de coût non nul est la coupure $(\{s,a_2\},\{t,a_1\})$, et qu'elle est de coût g (voir figure 4.2(b)).

4.3 Décroissance de l'énergie par *expansion move* optimal

La méthode de KOLMOGOROV et ZABIH ne cherche pas à minimiser la fonctionnelle d'énergie (non convexe) E car aucun algorithme efficace n'est connu pour cette tâche, le problème étant NP-difficile. À la place, elle essaie de la faire décroître, en introduisant des *expansion moves*. On montrera qu'une décroissance optimale peut être obtenue grâce à la représentation de l'énergie par un graphe.

4.3.1 *Expansion moves* et décroissance de l'énergie

Expansion moves Soit $u^h \in (\mathrm{I}_{\mathrm{disp}} \times \{\mathtt{occ}\})^{N_x N_y}$ une carte de disparité et $\alpha \in \mathrm{I}^h_{\mathrm{disp}}$ une valeur de disparité admissible. Alors on dit que $(u^h)' \in (\mathrm{I}_{\mathrm{disp}} \times \{\mathtt{occ}\})^{N_x N_y}\}$ est un α-*expansion move* de u^h si

$$\forall\,(i,j) \in \Omega^h, \qquad \begin{cases} (u^h)'_{i,j} = \alpha & \text{si } u^h_{i,j} = \alpha \\ (u^h)'_{i,j} \in \{u^h_{i,j},\alpha,\mathtt{occ}\} & \text{si } u^h_{i,j} \neq \alpha. \end{cases}$$

En particulier, u^h est un α-*expansion move* de lui-même quel que soit α. Pour un α donné, tout α-*expansion move* de u^h possède davantage de pixels de disparité α que la carte u^h. Chaque pixel de disparité différente de α peut soit devenir occulté, soit adopter la disparité α, soit conserver sa disparité. Enfin, chaque pixel occulté peut soit rester occulté, soit adopter la disparité α. Par conséquent, l'ensemble des pixels occultés peut croître, mais pas les ensembles des pixels de disparité différente de α.

Décroissance de l'énergie Soit u^h une carte de disparité donnée, d'énergie $\mathrm{E}(u^h)$ supposée finie. Pour tout $\alpha \in \mathrm{I}^h_{\mathrm{disp}}$ donné, on considère l'ensemble des α-*expansion move* de u^h. Puisque cet ensemble est de cardinal fini, il existe un élément d'énergie minimale, noté u^h_α. Par ailleurs, cet ensemble contient u^h, donc $\mathrm{E}(u^h_\alpha) \leq \mathrm{E}(u^h)$.

Le principe de la décroissance de l'énergie par *expansion moves* est de partir d'une carte de disparité u^h, puis, pour une valeur de $\alpha \in \mathrm{I}^h_{\mathrm{disp}}$, de la mettre à jour en la remplaçant par u^h_α l'α-*expansion move* de u^h d'énergie minimale

$$u^h_\alpha = \underset{(u^h)' \ \alpha\text{-expansion move de } u^h}{\operatorname{argmin}} \mathrm{E}\Big((u^h)'\Big).$$

Une *itération* de l'algorithme d'*expansion moves* consiste à répéter cette opération pour toutes les valeurs de $\mathrm{I}^h_{\mathrm{disp}}$, en mettant à jour la carte de départ entre chaque minimisation. Cela conduit à l'algorithme 1. Le critère d'arrêt est choisi de la manière suivante : lorsque l'énergie n'a pas décru sur une itération complète de tous les α(*expansion moves* appliqués successivement, alors l'algorithme s'arrête.

Algorithme 1: Décroissance de l'énergie par *expansion moves* : une itération

Entrée : u^h_0 une carte de disparité initiale, $\mathrm{I}^h_{\mathrm{disp}}$ l'intervalle de disparité
Sortie : u^h une carte de disparité d'énergie plus petite

```
1 begin
2     u^h ← u^h_0
3     foreach valeur de disparité α ∈ I^h_disp (dans un ordre arbitraire) do
4         u^h ← argmin_{(u^h)' α-expansion move de u^h} E((u^h)')
```

L'algorithme proposé ne minimise donc pas la fonctionnelle E, mais chaque pas de minimisation considéré est choisi de manière optimale. Si la fonctionnelle d'énergie ne possède pas de minima locaux, alors cette approche permet de retrouver un minimiseur global. Sinon, il est possible de trouver un minimiseur local. Néanmoins, ce risque est plus faible comparé à une approche de type descente de gradient, alors la décroissance de l'énergie se fait en considérant un sous-ensemble de cartes de disparité qui ne se situe pas nécessairement dans un voisinage de la carte de départ.

On se propose dans ce qui suit d'utiliser la représentabilité des énergies par un graphe pour résoudre de manière optimale la recherche de l'*expansion move* d'énergie minimale. Pour cela, on montre que l'on peut réécrire ce problème à l'aide d'une énergie représentable par un graphe, en effectuant un changement de variable. Cette section reprend l'étude proposée dans [7].

4.3.2 Assignements et configurations

On introduit ici un premier changement de variable qui permettra de rendre le problème de la décroissance d'énergie par *expansion moves* représentable par des graphes.

Assignements Au lieu de considérer les couples $\Big((i,j),u^h_{i,j}\Big)$ formés par les pixels de l'image de référence et leur disparité, on choisit une représentation symétrique sur les deux images de la paire en considérant les couples (p,q) de pixels, avec p appartenant à l'image de référence (dont l'ensemble des pixels est noté $\mathcal{I}_{\mathrm{L}}$) et q appartenant à

l'image de droite (dont l'ensemble des pixels est noté $\mathcal{I}_{\mathrm{R}}$). Parmi ces couples, on appelle *assignement* tout couple vérifiant $q - p \in \mathrm{I}^h_{\mathrm{disp}} \times \{0\}$. Un assignement est donc un couple de deux pixels situés sur la même ligne, dont la différence sur la ligne appartient à l'intervalle de disparité. Autrement dit, tout couple de pixels homologues est un assignement. L'ensemble des assignements est noté $\mathcal{A}$:

$$\mathcal{A} = \Big\{ a = (p,q) \in \mathcal{I}_{\mathrm{L}} \times \mathcal{I}_{\mathrm{R}} \mid q - p \in \mathrm{I}^h_{\mathrm{disp}} \times \{0\} \Big\}.$$

Notons en particulier que si $(p,p+\alpha)$ appartient à l'ensemble des assignements, alors α est une valeur de disparité admissible et $p+\alpha$ est un pixel de l'image de droite.

Configuration Pour toute carte de disparité u^h, si $p = (i,j) \in \mathcal{I}_{\mathrm{L}}$ n'est pas occulté, alors il est mis en correspondance avec le pixel $q = (i,j - u^h_{i,j}) \in \mathcal{I}_{\mathrm{R}}$. On dit alors que l'assignement $a = (p,q)$ est *actif*. Pour tout assignement (p',q'), si p' n'est pas mis en correspondance avec q', alors l'assignement est dit *inactif*.

On peut alors définir la *configuration* associée à la carte de disparité u^h, qui est une fonction f_{u^h} qui à tout assignement actif associe la valeur 1 et à tout assignement inactif associe la valeur 0 :

$$\forall\, a = (p,q) \in \mathcal{A}, \qquad f_{u^h}(a) = \begin{cases} 1 & \text{si } p = (i,j) \qquad \text{et} \qquad q = (i,j + u^h_{i,j}) \\ 0 & \text{sinon.} \end{cases}$$

Toute fonction $f : \mathcal{A} \to \{0,1\}$ définit alors une configuration, et on appelle *état* d'un assignement son image par la configuration f. On note $\mathcal{A}^\circ(f)$ l'ensemble des assignements actifs sous la configuration f :

$$\mathcal{A}^\circ(f) = \Big\{ a \in \mathcal{A} \mid f(a) = 1 \Big\}.$$

Disparité Enfin, pour tout assignement $a = (p,q) \in \mathcal{A}$ (actif ou inactif), on définit sa disparité $d(a) = q - p$, que l'on confondra avec son abscisse qui appartient par définition à $\mathrm{I}^h_{\mathrm{disp}}$ (son ordonnée est nulle). Si $a = (p,q)$ est un assignement actif, alors p et q sont des pixels homologues et la disparité de $p = (i,j)$ dans l'image de gauche vaut $u^h_{i,j} = u^h_{I_{\mathrm{L}}}(p) = d(a)$. Celle de q dans l'image de droite vaut $u^h_{I_{\mathrm{R}}}(q) = -d(a)$.

Notons que la carte de disparité est entièrement donnée par la configuration associée, grâce à la formule de reconstruction

$$\forall\, p = (i,j) \in \mathcal{I}_{\mathrm{L}} \qquad u^h_{i,j} = \begin{cases} d(a) & \text{si } \exists\, a = (p,q) \in \mathcal{A}^\circ(f_{u^h}) \\ \mathtt{occ} & \text{sinon.} \end{cases}$$

Cette formule de reconstruction est bien définie si la configuration satisfait la contrainte d'injectivité[3], c'est-à-dire si, pour tout pixel $p \in \mathcal{I}_{\mathrm{L}}$ (respectivement $q \in \mathcal{I}_{\mathrm{R}}$), il existe au plus un assignement actif de premier élément p (resp. de second élément q).

Pour toute valeur de disparité $\alpha \in \mathrm{I}^h_{\mathrm{disp}}$, on note $\mathcal{A}^\alpha$ l'ensemble des assignements de disparité α :

$$\mathcal{A}_\alpha = \Big\{ a \in \mathcal{A} \mid d(a) = \alpha \Big\}.$$

On notera que cet ensemble ne dépend pas de la configuration choisie. Par ailleurs, une remarque préliminaire assure que si $(p,p+\alpha) \in \mathcal{A}_\alpha$, alors $p+\alpha$ est un pixel de l'image de droite.

3. *Uniqueness constraint* dans [8, 7].

4.3.3 Énergie d'une configuration

On va à présent réaliser un changement de variables dans la définition de l'énergie E pour définir l'énergie d'une configuration. Pour cela, on va construire pour chaque terme de l'énergie E un terme correspondant défini sur l'ensemble des configurations, de sorte que $\mathrm{E}(f_{u^h}) = \mathrm{E}(u^h)$.

Terme d'attache aux données et d'occultation Le terme d'attache aux données (4.1) n'est défini que sur les pixels p non occultés, c'est-à-dire ceux pour lesquels il existe un assignement (p,q) actif. On verra par la suite que la contrainte d'injectivité assure que cet assignement est unique, aussi on peut définir le nouveau terme d'attache aux données sur les assignements actifs sous la configuration f :

$$\mathrm{E}_{\mathrm{data}}(f) = \sum_{a=(p,q)\in\mathcal{A}^\circ(f)} g\Big(p,d(a)\Big).$$

Pour redéfinir le terme d'occultation (4.2), il faut remarquer qu'un pixel p est occulté si tous les assignements de la forme (p,q) sont inactifs. Or, le terme d'injectivité assure qu'il existe au plus un tel assignement actif. Ainsi, un pixel est occulté si

$$\sum_{a=(p,q)\in\mathcal{A}} \mathbb{1}_{\{1\}}\Big(f(a)\Big) = 0$$

et sinon, cette quantité vaut 1. On en déduit le terme d'occultation suivant

$$\mathrm{E}_{\mathrm{occ}}(f) = \sum_{p\in\mathcal{I}_L} K\left(1 - \sum_{a=(p,q)\in\mathcal{A}} \mathbb{1}_{\{1\}}\Big(f(a)\Big)\right) = K\,N_x\,N_y - \sum_{a\in\mathcal{A}^\circ(f)} K.$$

Si on somme ces deux termes, on peut définir un coût $D(a)$ pour chaque assignement a en posant

$$\forall\, a = (p,q) \in \mathcal{A}, \qquad D(a) = g\Big(p,d(a)\Big) - K$$

de sorte que la somme du terme d'attache aux données et d'occultation s'écrive

$$\mathrm{E}_{\mathrm{data+occ}}(f) = \sum_{a\in\mathcal{A}^\circ(f)} D(a) + \text{constante}$$

où la constante vaut $K\,N_x\,N_y$. On remarque que ce terme est défini sur les assignements actifs seulement. On vérifie que

$$\mathrm{E}_{\mathrm{data+occ}}(f_{u^h}) = \mathrm{E}_{\mathrm{data+occ}}(u^h).$$

Terme de régularisation Définissons à présent le nouveau terme de régularisation. Dans (4.3), une pénalité est ajoutée à chaque fois que deux pixels voisins (tous deux non occultés) p et p' ont une disparité différente. Commençons par introduire la notation suivante pour tous assignements $a = (p,q)$ et $a' = (p',q')$

$$a \sim a' \qquad \text{si} \qquad d(a) = d(a') \qquad \text{et} \qquad \begin{cases} p' = p + (0, \pm 1) \\ p' = p + (\pm 1, 0) \end{cases}$$

pour des assignements qualifiés de *voisins*. Deux assignements sont voisins s'ils ont même disparité et si leurs premiers pixels le sont dans l'image de référence ou que leurs seconds pixels le sont dans l'image de droite. On notera que, puisque $d(a) = d(a')$, on obtient les mêmes conditions sur q et q'.

Soient deux pixels p et p' voisins dans l'image de gauche. Si les deux sont occultés ou qu'ils ont même disparité, alors le couple (p,p') ne contribue pas au terme de régularisation. Or, dans le premier cas, on a

$$\forall\, a = (p,q) \in \mathcal{A}, \quad \forall\, a' = (p',q') \in \mathcal{A}, \qquad f(a) = f(a') = 0.$$

En particulier, pour tous $a \sim a'$, on a $f(a) = f(a') = 0$. Dans le second cas, le seul assignement actif de la forme (p,q) et le seul assignement actif de la forme (p',q') ont même disparité, puisque p et p' ont même disparité (et les autres assignements de cette forme sont inactifs). On en déduit que $f(a) = f(a')$ pour tous assignements $a = (p,q)$ et $a' = (p',q')$.

Supposons maintenant que les deux pixels ne sont pas occultés, mais de disparité différente. Alors le couple (p,p') contribue deux fois au terme de régularisation, une fois pour la disparité $u^h_{I_\mathrm{L}}(p)$ et une seconde fois pour la disparité $u^h_{I_\mathrm{L}}(p')$. Or, cette situation impose que l'assignement actif de la forme (p,q) et l'assignement actif de la forme (p',q') soient de disparité différente. Plus précisément, les assignements $(p,p+u^h_{I_\mathrm{L}}(p))$ et $(p',p'+u^h_{I_\mathrm{L}}(p'))$ sont actifs mais pas les assignements $(p,p+u^h_{I_\mathrm{L}}(p'))$ et $(p',p'+u^h_{I_\mathrm{L}}(p))$. Autrement dit,

$$\Big(p,p+u^h_{I_\mathrm{L}}(p)\Big) \sim \Big(p',p'+u^h_{I_\mathrm{L}}(p)\Big) \qquad \text{et} \qquad f\Big(p,p+u^h_{I_\mathrm{L}}(p)\Big) \neq f\Big(p',p'+u^h_{I_\mathrm{L}}(p)\Big)$$

et

$$\Big(p',p'+u^h_{I_\mathrm{L}}(p')\Big) \sim \Big(p,p+u^h_{I_\mathrm{L}}(p')\Big) \qquad \text{et} \qquad f\Big(p',p'+u^h_{I_\mathrm{L}}(p')\Big) \neq f\Big(p,p+u^h_{I_\mathrm{L}}(p')\Big)$$

Supposons à présent que l'un des deux pixels (p' par exemple) est occulté (et que l'autre ne l'est pas). Alors le couple (p,p') contribue une fois au terme de régularisation (pour la disparité $u^h_{I_\mathrm{L}}(p)$ du pixel non occulté) et on a cette fois

$$\Big(p,p+u^h_{I_\mathrm{L}}(p)\Big) \sim \Big(p',p'+u^h_{I_\mathrm{L}}(p)\Big) \qquad \text{et} \qquad f\Big(p,p+u^h_{I_\mathrm{L}}(p)\Big) \neq f\Big(p',p'+u^h_{I_\mathrm{L}}(p)\Big)$$

puisque l'assignement $(p',p'+u^h_{I_\mathrm{L}}(p))$ est nécessairement inactif.

On en déduit que le couple de pixels voisins (p,p') contribue au terme de régularisation à chaque fois qu'il existe des assignements $a = (p,q)$ et $a' = (p',q')$ voisins qui n'ont pas le même état. Cela nous conduit à proposer le terme de régularisation suivant

$$\mathrm{E}_{\mathrm{reg}}(f) = \frac{1}{2} \sum_{\substack{(a,a')\in\mathcal{A}^2 \\ a\sim a'}} \tilde{R}_{\lambda_1,\lambda_2}\Big(a,a',f(a),f(a')\Big)$$

$$\text{avec} \quad \tilde{R}_{\lambda_1,\lambda_2}\Big(a,a',f(a),f(a')\Big) = \begin{cases} 0 & \text{si } f(a) = f(a') \\ R_{\lambda_1,\lambda_2}\Big(p,p',d(a)\Big) & \text{si } f(a) \neq f(a') \text{ et } f(a) = 1 \\ R_{\lambda_1,\lambda_2}\Big(p,p',d(a')\Big) & \text{si } f(a) \neq f(a') \text{ et } f(a') = 1 \end{cases}$$

Le facteur $1/2$ est nécessaire car les couples d'assignements (a,a') et (a',a) contribuent tous deux à ce terme. On a alors

$$\mathrm{E}_{\mathrm{reg}}(f_{u^h}) = \mathrm{E}_{\mathrm{reg}}(u^h).$$

Terme d'injectivité Le terme d'injectivite (4.4) impose que, si p est mis en correspondance avec le pixel q, alors il est le seul. Autrement dit, si l'assignement (p,q) est actif, alors tous les assignements (p',q) avec $p' \neq p$ doivent être inactifs. Par ailleurs, il est évident que tout pixel de l'image de gauche ne peut être mis en correspondance avec au plus un pixel. En termes d'assignements, cela se traduit de la manière suivante : si

l'assignement (p,q) est actif, alors tous les assignements (p,q') avec $q' \neq q$ doivent être inactifs. On en déduit le terme d'injectivité suivant

$$\mathrm{E}_{\mathrm{inj}}(f) = \sum_{a=(p,q)\in\mathcal{A}^\circ(f)} \sum_{\substack{a'=(p',q)\\ p'\neq p}} \chi_{\{0\}}\Big(f(a')\Big) + \sum_{a=(p,q)\in\mathcal{A}^\circ(f)} \sum_{\substack{a'=(p,q')\\ q'\neq q}} \chi_{\{0\}}\Big(f(a')\Big).$$

Ce terme est nul ou prend une valeur infinie et on vérifie à nouveau que

$$\mathrm{E}_{\mathrm{inj}}(f_{u^h}) = \mathrm{E}_{\mathrm{inj}}(u^h).$$

Finalement, on a construit une énergie sur l'ensemble des configurations

$$\mathrm{E}(f) = \mathrm{E}_{\mathrm{data+occ}}(f) + \mathrm{E}_{\mathrm{reg}}(f) + \mathrm{E}_{\mathrm{inj}}(f)$$

vérifiant l'identité $\mathrm{E}(u^h) = \mathrm{E}(f_{u^h})$ pour toute carte de disparité.
Remarque : En réécrivant l'énergie à l'aide d'assignements, on rétablit la symétrie du problème de mise en correspondance, en rendant les deux images (gauche et droite) indifférenciées. En particulier, ce traitement nécessite de manipuler des images échantillonnées sur une même grille, afin de conserver cette symétrie.

4.3.4 Énergie d'un *expansion move*

Expansion move Soit f une configuration d'énergie finie et $\alpha \in \mathrm{I}^h_{\mathrm{disp}}$ une valeur de disparité. On dit que f' est un α-*expansion move* de la configuration f si sa carte de disparité associée est un α-*expansion move* de la carte de disparité associée à la configuration f.

D'après la définition des α-*expansion moves* sur les cartes de disparité, on en déduit que

$$\forall\, a \in \mathcal{A}, \quad f'(a) = \begin{cases} 1 & \text{si } f(a) = 1 \text{ et } d(a) = \alpha \\ 0 & \text{si } f(a) = 0 \text{ et } d(a) \neq \alpha \end{cases}.$$

En d'autres termes, tout assignement actif de disparité α reste actif après un α-*expansion move* (tout pixel de disparité α conserve sa disparité) et tout assignement inactif de disparité différente de α le reste (les pixels occultés ou de disparité différente de α ne peuvent pas adopter une nouvelle disparité, mais peuvent devenir occultés ou adopter la disparité α).

Changement d'état Remarquons qu'il est possible de différencier le comportement d'un assignement pendant un α-*expansion move* suivant l'ensemble auquel il appartient :

1. l'ensemble $\mathcal{A}^\circ(f) \cap \mathcal{A}_\alpha$ des assignements actifs de disparité α ;
2. l'ensemble $\mathcal{A}_\alpha \setminus \mathcal{A}^\circ(f)$ des assignements inactifs de disparité α ;
3. l'ensemble $\mathcal{A}^\circ(f) \setminus \mathcal{A}_\alpha$ des assignements actifs de disparité différente de α ;
4. l'ensemble $\mathcal{A} \setminus (\mathcal{A}^\circ \cup \mathcal{A}_\alpha)$ des assignements inactifs de disparité différente de α.

Les assignements du premier et du dernier ensemble ne changent pas d'état après un α-*expansion move*. Plus précisément, on a

$$\forall\, a \in \mathcal{A}, \quad f'(a) = \begin{cases} 1 & \text{si } a \in \mathcal{A}^\circ \text{ et } a \in \mathcal{A}_\alpha \\ 0 & \text{si } a \notin \mathcal{A}^\circ \text{ et } a \notin \mathcal{A}_\alpha. \end{cases}.$$

Le second ensemble ne peut que croître tandis que le troisième ne peut que décroître.

Énergie Nous allons à présent exprimer l'énergie d'un α-*expansion move* f' en fonction de l'énergie de la configuration initiale f. Pour cela, on utilise les remarques précédentes pour distinguer les termes variables de ceux qui restent constants, car impliquant des assignements dont l'état reste invariable après α-*expansion move*. Commençons par le terme d'attache aux données et d'occultation. Celui-ci est défini sur les assignements actifs de f'. Or, les seuls assignements f' qui peuvent être actifs sont ceux qui l'étaient déjà dans f, et les assignements de disparité α, en gardant en tête que, parmi ces derniers, ceux déjà actifs dans la configuration initiale f le restent (leur contribution à ce terme ne change donc pas). On peut donc le réécrire en utilisant les trois premiers ensembles introduits plus haut :

$$\mathrm{E}_{\text{data+occ}}(f') = \sum_{\substack{a\notin\mathcal{A}^\circ(f)\\ a\in\mathcal{A}_\alpha}} D(a)\,\mathbb{1}_{\{1\}}\Big(f'(a)\Big) + \sum_{\substack{a\in\mathcal{A}^\circ(f)\\ a\notin\mathcal{A}_\alpha}} D(a)\,\mathbb{1}_{\{1\}}\Big(f'(a)\Big) + \text{constante}.$$

Pour le terme de régularisation, on commence par décomposer la somme sur les assignements voisins de disparité α et les autres, puisqu'on rappelle que deux assignements voisins ont par définition même disparité :

$$\mathrm{E}_{\text{reg}}(f') = \frac{1}{2}\sum_{\substack{(a,a')\in\mathcal{A}^2\\ a\sim a'\\ d(a)=\alpha}} \tilde{R}_{\lambda_1,\lambda_2}\Big(a,a',f'(a),f'(a')\Big) + \frac{1}{2}\sum_{\substack{(a,a')\in\mathcal{A}^2\\ a\sim a'\\ d(a)\neq\alpha}} \tilde{R}_{\lambda_1,\lambda_2}\Big(a,a',f'(a),f'(a')\Big).$$

Ensuite, pour chaque terme, on distingue selon l'état initial des assignements a et a'. On rappelle qu'un assignement a' actif de disparité α le reste, de même qu'un assignement a' inactif de disparité différente de α. Par ailleurs, un couple (a,a') ne contribue à ce terme que si les deux états associés sont différents. En particulier, is a et a' sont actifs sous la configuration f et de disparité α, alors le couple (a,a') ne contribue pas plus au terme de régularité pour la configuration f que pour la configuration f' (*idem* pour deux assignements voisins inactifs sous f de disparité différente de α). On peut donc les ignorer, et on obtient alors

$$\begin{aligned}\mathrm{E}_{\text{reg}}(f') = &\sum_{\substack{a\in\mathcal{A}^\circ(f)\\ a'\notin\mathcal{A}^\circ(f)\\ a\sim a'\\ d(a)=\alpha}} \tilde{R}_{\lambda_1,\lambda_2}\Big(a,a',1,f'(a')\Big) + \frac{1}{2}\sum_{\substack{a\notin\mathcal{A}^\circ(f)\\ a'\notin\mathcal{A}^\circ(f)\\ a\sim a'\\ d(a)=\alpha}} \tilde{R}_{\lambda_1,\lambda_2}\Big(a,a',f'(a),f'(a')\Big)\\ &+\frac{1}{2}\sum_{\substack{a\in\mathcal{A}^\circ(f)\\ a'\in\mathcal{A}^\circ(f)\\ a\sim a'\\ d(a)\neq\alpha}} \tilde{R}_{\lambda_1,\lambda_2}\Big(a,a',f'(a),f'(a')\Big) + \sum_{\substack{a\notin\mathcal{A}^\circ(f)\\ a'\in\mathcal{A}^\circ(f)\\ a\sim a'\\ d(a)\neq\alpha}} \tilde{R}_{\lambda_1,\lambda_2}\Big(a,a',0,f'(a')\Big).\end{aligned}$$

Notons que, dès que l'on supprime la symétrie dans le rôle des assignements a et a', le facteur $1/2$ disparaît. Enfin, le terme d'injectivité étant nul pour f car celle-ci est supposée d'énergie finie, il est soit nul pour f' si la contrainte d'injectivité est respectée, soit infini si l'activation d'un assignement viole cette contrainte. Or, les seuls assignements à pouvoir être activés sont les assignements (p,q) de disparité α inactifs pour f, et cette activation viole la contrainte d'injectivité que s'il existait auparavant un assignement actif (p,q') ou (p',q) de disparité différente de α, et que celui-ci n'est

pas désactivé dans f' :

$$\mathrm{E}_{\mathrm{inj}}(f') = \sum_{\substack{a=(p,q)\in\mathcal{A}^\circ(f)\\ a\notin\mathcal{A}_\alpha\\ (p,p+\alpha)\in\mathcal{A}_\alpha}} \chi_{\{(0,0),(0,1),(1,0)\}}\Big(f'(p,p+\alpha),f'(a)\Big) + \sum_{\substack{a=(p,q)\in\mathcal{A}^\circ(f)\\ a\notin\mathcal{A}_\alpha\\ (q-\alpha,q)\in\mathcal{A}_\alpha}} \chi_{\{(0,0),(0,1),(1,0)\}}\Big(f'(q-\alpha,q),f'(a)\Big).$$

4.3.5 Représentabilité de l'énergie des *expansion moves*

Soit f une configuration et $\alpha \in \mathrm{I}^h_{\mathrm{disp}}$ une valeur de disparité. On rappelle qu'on cherche à trouver l'α-*expansion move* de f d'énergie minimale, c'est-à-dire de résoudre le problème

$$f_\alpha = \underset{f' \ \alpha\text{-expansion move de } f}{\operatorname{argmin}} \mathrm{E}(f'). \tag{4.6}$$

Malheureusement, cette énergie n'est pas représentable par un graphe. On va donc introduire un nouveau changement de variable pour la rendre représentable. Ce changement de variable permettra en outre de supprimer la contrainte dans le problème (4.6), en encodant de manière naturelle les α-*expansion moves* de f.

Vecteur de changement d'état On choisit d'introduire le changement de variable suivant, qui encode tout changement d'état entre la configuration initiale f et l'α-*expansion move* f' :

$$\forall\, a \in (\mathcal{A}^\circ(f)\cup\mathcal{A}_\alpha)\setminus(\mathcal{A}^\circ(f)\cap\mathcal{A}_\alpha), \qquad g'(a) = \begin{cases} 1 & \text{si } f(a)\neq f'(a)\\ 0 & \text{sinon.}\end{cases} \tag{4.7}$$

On choisit d'ignorer les assignements dont l'état reste invariable. Tout α-*expansion move* f' définit un vecteur $g' \in \{0,1\}^N$, avec $N = \mathrm{card}((\mathcal{A}^\circ(f)\cup\mathcal{A}_\alpha)\setminus(\mathcal{A}^\circ(f)\cap\mathcal{A}_\alpha))$, tandis que tout tel vecteur g' définit de manière unique un α-*expansion move* f', grâce à la formule d'inversion suivante :

$$\forall\, a \in \mathcal{A}, \qquad f'(a) = \begin{cases} 1-g'(a) & \text{si } a\in\mathcal{A}^\circ(f) \text{ et } a\notin\mathcal{A}_\alpha\\ g'(a) & \text{si } a\notin\mathcal{A}^\circ(f) \text{ et } a\in\mathcal{A}_\alpha\\ f(a) & \text{sinon.}\end{cases} \tag{4.8}$$

Une nouvelle énergie On va à présent proposer une nouvelle énergie $\mathrm{E}^{\alpha,f}$ en appliquant le changement de variable proposé plus haut. Ainsi, l'identité $\mathrm{E}^{\alpha,f}(g') = \mathrm{E}(f')$ est vérifiée (à une constante près) pour tout α-*expansion move* f'. Le terme d'attache aux données et d'occultation est donné par :

$$\mathrm{E}^{\alpha,f}_{\mathrm{data+occ}}(g') = \sum_{\substack{a\notin\mathcal{A}^\circ(f)\\ a\in\mathcal{A}_\alpha}} D(a)\,\mathbb{1}_{\{1\}}\Big(g'(a)\Big) + \sum_{\substack{a\in\mathcal{A}^\circ(f)\\ a\notin\mathcal{A}_\alpha}} D(a)\,\mathbb{1}_{\{1\}}\Big(g'(a)\Big).$$

On obtient ensuite pour le terme de régularisation

$$\mathrm{E}_{\mathrm{reg}}^{\alpha,f}(g') = \sum_{\substack{a\in\mathcal{A}^\circ(f)\\ a'\notin\mathcal{A}^\circ(f)\\ a\sim a'\\ d(a)=\alpha}} \tilde{R}_{\lambda_1,\lambda_2}\Big(a,a',1,g'(a')\Big) + \sum_{\substack{a\notin\mathcal{A}^\circ(f)\\ a'\in\mathcal{A}^\circ(f)\\ a\sim a'\\ d(a)\neq\alpha}} \tilde{R}_{\lambda_1,\lambda_2}\Big(a,a',0,1-g'(a')\Big)$$
$$+ \frac{1}{2}\sum_{\substack{a\notin\mathcal{A}^\circ(f)\\ a'\notin\mathcal{A}^\circ(f)\\ a\sim a'\\ d(a)=\alpha}} \tilde{R}_{\lambda_1,\lambda_2}\Big(a,a',g'(a),g'(a')\Big) + \frac{1}{2}\sum_{\substack{a\in\mathcal{A}^\circ(f)\\ a'\in\mathcal{A}^\circ(f)\\ a\sim a'\\ d(a)\neq\alpha}} \tilde{R}_{\lambda_1,\lambda_2}\Big(a,a',1-g'(a),1-g'(a')\Big)$$

et enfin pour le terme d'injectivité :

$$\mathrm{E}_{\mathrm{inf}}^{\alpha,f}(g') = \sum_{\substack{a=(p,q)\in\mathcal{A}^\circ(f)\\ a\notin\mathcal{A}_\alpha\\ (p,p+\alpha)\in\mathcal{A}_\alpha}} \chi_{\{(0,0),(1,0),(1,1)\}}\Big(g'(p,p+\alpha),g'(a)\Big)$$
$$+ \sum_{\substack{a=(p,q)\in\mathcal{A}^\circ(f)\\ a\notin\mathcal{A}_\alpha\\ (q-\alpha,q)\in\mathcal{A}_\alpha}} \chi_{\{(0,0),(1,0),(1,1)\}}\Big(g'(q-\alpha,q),g'(a)\Big).$$

On vérifie alors que, si on pose

$$\mathrm{E}^{\alpha,f}(g') = \mathrm{E}_{\mathrm{data+occ}}^{\alpha,f}(g') + \mathrm{E}_{\mathrm{reg}}^{\alpha,f}(g') + \mathrm{E}_{\mathrm{inf}}^{\alpha,f}(g')$$

alors on a l'identité $\mathrm{E}^{\alpha,f}(g') = \mathrm{E}(f') + \text{constante}$ pour tout α-*expansion move* de f, où la constante ne dépend pas de f. Ainsi, le problème (4.6) est équivalent au problème *non contraint* suivant

$$g_\alpha = \operatorname*{argmin}_{g'\in\{0,1\}^N} \mathrm{E}^{\alpha,f}(g'). \tag{4.9}$$

Représentabilité de l'énergie Vérifions que l'énergie $\mathrm{E}^{\alpha,f}$ est représentable par un graphe. Pour cela, on utilise le théorème 12, qui stipule qu'il suffit de vérifier si chaque terme dépendant de deux variables satisfait la contrainte de sous-modularité.

Le terme d'attache aux données ne possède que des termes dépendant d'une variable binaire, donc il est représentable. Passons au terme de régularisation. Les seuls termes dépendant de deux variables binaires sont ceux de la forme

$$\mathrm{E}_1(x,x') = \tilde{R}_{\lambda_1,\lambda_2}(a,a',x,x')$$

lorsque a et a' sont inactifs sous la configuration initiale f, mais de disparité α et ceux de la forme

$$\mathrm{E}_2(x,x') = \tilde{R}_{\lambda_1,\lambda_2}(a,a',1-x,1-x')$$

où a et a' sont actifs sous la configuration initiale f, mais de disparité différente de α. La définition de $\tilde{R}_{\lambda_1,\lambda_2}$ assure que $\mathrm{E}_1(0,0) = \mathrm{E}_1(1,1) = \mathrm{E}_2(0,0) = \mathrm{E}_2(1,1) = 0$ et que G_1 et G_2 sont positives, ce qui permet de vérifier la contrainte de sous-modularité. Enfin, les termes dans le terme d'injectivité sont de la forme

$$\mathrm{E}_3(x,x') = \chi_{\{0\}}\Big(x\,(1-x')\Big)$$

qui est nul si $x = 0$ ou $x' = 1$, et infini si $(x,x') = (1,0)$, ce qui assure à nouveau la sous-modularité de ce terme.

Résolution du problème (4.9) On vient donc de démontrer que l'énergie $\mathrm{E}^{\alpha,f}$ est représentable par un graphe $\mathcal{G}^{\alpha,f} = (\mathcal{V},\mathcal{E})$. Par définition, si on choisit le graphe avec le nombre de sommets minimal, alors, si on ordonne les assignements de $(\mathcal{A}^{\circ}(f) \cup \mathcal{A}_{\alpha}) \setminus (\mathcal{A}^{\circ}(f) \cap \mathcal{A}_{\alpha}) = \{a_i\}_{i\in[\![1\,;N]\!]}$, on a

$$\mathcal{V} = \left\{s,t,\{a_i\}_{i\in[\![1\,;N]\!]}\right\}$$

et $\mathrm{E}^{\alpha,f}(g')$ est égal à C plus le coût de la coupure $(\mathcal{V}^s,\mathcal{V}^t)$ vérifiant $a_i \in \mathcal{V}^s$ si $g'(a_i) = 0$ et $a_i \in \mathcal{V}^t$ si $g'(a_i) = 1$. On en déduit en particulier que la valeur minimale de $\mathrm{E}^{\alpha,f}$, qui vaut $\mathrm{E}^{\alpha,f}(g_\alpha)$ est donnée par la coupure minimale du graphe $\mathcal{G}^{\alpha,f}$. Résoudre le problème (4.9) revient donc à trouver la coupure minimale du graphe $\mathcal{G}^{\alpha,f}$, et d'après le théorème de Ford-Fulkerson, cela revient à trouver le flot maximal dans ce même graphe.

Une fois la coupure minimale obtenue, les assignements associés aux sommets qui sont reliés à la source ne changent pas d'état, tandis que les autres changent d'état.

4.4 Résolution numérique par coupure de graphes

On présente ici les détails de l'implémentation de l'algorithme décrit dans les sections précédentes, dont le code initial a été écrit par Kolmogorov (et largement aménagé pour [7]). Il sera désigné par la suite sous le nom d'algorithme KZ2.

On rappelle que l'idée est de faire décroître l'énergie en calculant pour chaque α le meilleur α-*expansion move* de la configuration f obtenue à l'itération précédente (algorithme 1). Pour cela, on a montré qu'il suffisait de construire le graphe représentant l'énergie $\mathrm{E}^{\alpha,f}$ puis d'en trouver la coupure minimale en en déterminant le flot maximal. Cela conduit à l'algorithme 2.

Algorithme 2: Recherche de l'α-*expansion move* optimal de f

Entrée : f une configuration initiale, $\alpha \in \mathrm{I}^h_{\mathrm{disp}}$ une valeur de disparité
Sortie : f_α l'α-*expansion move* de f d'énergie la plus faible

$$f_\alpha = \underset{f' \ \alpha\text{-expansion move de } f}{\operatorname{argmin}} \mathrm{E}\left(f'\right)$$

1 **begin**
2 Construire le graphe $\mathcal{G}^{\alpha,f}$
3 Calculer le flot maximal du graphe $\mathcal{G}^{\alpha,f}$
4 En déduire la coupure maximale du graphe $\mathcal{G}^{\alpha,f}$
5 Construire le vecteur g_α à l'aide de la formule (4.4.3)
6 Obtenir f_α à partir de g_α, avec (4.8)

4.4.1 Construction du graphe

Soit $\alpha \in \mathrm{I}^h_{\mathrm{disp}}$ et f une configuration initiale donnée (associée à la carte de disparité u^h). Commençons par détailler la construction du graphe $\mathcal{G}^{\alpha,f}$ représentant l'énergie $\mathrm{E}^{\alpha,f}$. Pour des précisions plus techniques quant à cette construction, nous invitons le lecteur à se reporter à [7].

Sommets Les sommets du graphe $\mathcal{G}^{\alpha,f}$ doivent comprendre une source s et un puits t, ainsi qu'un sommet pour chaque assignement actif sous f de disparité différente de α et un sommet par assignement inactif sous f de disparité α. Ces derniers sont indexés par le premier pixel p, pour lequel on peut construire jusqu'à deux sommets (celui associé à l'assignement $(p,p+u^h)$ et celui associé à l'assignement, s'il est différent, $(p,p+\alpha)$). Ainsi, pour tout pixel de l'image de référence, on construit 0, 1 ou 2 sommets. On confondra dans ce qui suit la notation des assignements et celle des sommets associés.

En pratique, la construction des sommets du graphe se fait à l'aide de deux tableaux, nommés `vars0` et `varsA`, qui sont indexés par les pixels p de l'image de référence. S'ils existent, les coefficients `vars0(p)` et `varsA(p)` représentent respectivement l'assignement actif $(p,p+u^h(p))$ de premier élément p et l'assignement $(p,p+\alpha)$ de disparité α. Les éléments de ces deux tableaux sont soit invariables (`VAR_ACTIVE` ou `VAR_ABSENT`), soit variables (`o` ou `a`).

Les valeurs invariables permettent de localiser les assignements qui ne changeront par d'état (et donc pour lesquels le vecteur g de changement d'état n'est pas défini). La valeur `VAR_ACTIVE` est réservée aux assignements qui restent actifs, c'est-à-dire les assignements $(p,p+\alpha)$ initialement actifs. Ainsi, si l'assignement $a=(p,p+\alpha)$ est initialement actif, on affecte `VAR_ACTIVE` aux coefficients `vars0(p)` et `varsA(p)`. La valeur `VAR_ABSENT` est quant à elle destinée aux sommets qui ne sont pas construits car les assignements associés n'existent pas, ou qui resteront inactifs (car inactifs et de disparité différente de α). C'est le cas lorsque $p+\alpha$ n'appartient pas à l'ensemble des pixels de l'image de droite (car il sort du domaine), auquel cas il existe aucun assignement de la forme $(p,p+\alpha)$; on attribue alors la valeur `VAR_ABSENT` au coefficient `varsA(p)`. C'est également le cas s'il n'existe aucun assignement actif de la forme (p,q), c'est-à-dire si p est occulté ; dans ce cas, on affecte `VAR_ABSENT` au coefficient `vars0(p)`.

Lorsqu'un sommet est construit pour un assignement $a=(p,q)$, alors on crée une variable `o` dans le tableau `vars0` si a est un assignement actif de disparité différente de α, et on crée une variable `a` pour le coefficient `varsA(p)` si l'assignement a est disparité α. On désignera désormais ces sommets comme des sommets *variables*.

Arcs La construction des arcs suit celle proposée dans l'étude du paragraphe 4.2.2, en particulier les graphes représentés à la figure 4.1.

Le terme d'attache aux données ne comporte que des fonctions dépendant d'une seule variable binaire. On doit donc construire pour chaque sommet variable a un arc, relié à la source ou au puits, suivant le signe du coût $D(a)$. Si $D(a)$ est positif, alors on construit l'arc (s,a), en lui attribuant la capacité $D(a)$. Sinon, on construit l'arc (a,t), de capacité $-D(a)$.

Le terme de régularité est composé à la fois de fonctions dépendant d'une seule variable et de fonctions dépendant de deux variables. Dans les deux cas, les capacités des arcs construits dépendent des assignements voisins, qui sont localisés en comparant la disparité des assignements des pixels voisins dans les tableaux `vars0` et `varsA`. Soit p le pixel courant et p' un de ses voisins. Commençons par considérer le cas où les assignements voisins $a=(p,q)$ et $a'=(p',q')$ sont de disparité α. Si le sommet $a=(p,p+\alpha)$ associé à p n'est pas variable, alors il n'y a aucun terme variable associé à l'assignement a dans l'énergie $\mathrm{E}^{\alpha,f}$. On suppose a est un sommet variable. Si a' n'est pas variable, alors deux cas sont possibles. Si a' n'est pas variable mais actif sous la configuration initiale, alors il faut représenter le terme $\tilde{R}_{\lambda_1,\lambda_2}(a,a',x,1)$ qui ne dépend que de la seule variable a, avec

$$\tilde{R}_{\lambda_1,\lambda_2}(a,a',1,1)=0 \qquad \text{et} \qquad \tilde{R}_{\lambda_1,\lambda_2}(a,a',0,1)=R_{\lambda_1,\lambda_2}\Big(p,p',d(a')\Big)>0$$

avec ici $d(a) = \alpha$. Il faut donc construire l'arc (a',t) avec la capacité $R_{\lambda_1,\lambda_2}\big(p,p',d(a)\big)$. Si l'arc a déjà été construit à l'étape précédent, il suffit de mettre à jour sa capacité en lui ajoutant cette quantité. Si a' n'est pas variable car l'assignement associé n'existe pas ($p' + \alpha$ n'est pas dans l'image de droite), alors il n'y a rien à construire car le couple (a,a') ne contribue pas au terme de régularisation. Si les deux sommets sont variables, alors il faut représenter le terme $\tilde{R}_{\lambda_1,\lambda_2}(a,a',x,x')$. On utilise pour cela la décomposition (4.5), avec

$$\tilde{R}_{\lambda_1,\lambda_2}(a,a',0,0) = 0 \qquad \text{et} \qquad \tilde{R}_{\lambda_1,\lambda_2}(a,a',0,1) = R_{\lambda_1,\lambda_2}\big(p,p',d(a')\big) > 0$$

$$\tilde{R}_{\lambda_1,\lambda_2}(a,a',1,1) = 0 \qquad \text{et} \qquad \tilde{R}_{\lambda_1,\lambda_2}(a,a',1,0) = R_{\lambda_1,\lambda_2}\big(p,p',d(a)\big) > 0.$$

On procède de la même manière pour les assignements voisins de disparité différente de α (il faut alors distinguer le cas où $a' = (p',p' + u^h(p))$ est variable et le cas où il n'est pas actif). Enfin, pour représenter le terme d'injectivité, on construit pour les assignements $a = (p,p + u^h(p))$ et $a = (p,p + \alpha)$ (si les deux existent et sont variables) l'arc (a,a') en lui attribuant une capacité infinie (une valeur arbitrairement grande en pratique).

4.4.2 Recherche du flot maximal par chemins augmentants

La recherche du flot maximal est réalisée grâce à l'algorithme proposé dans [3].

Cet algorithme améliore ceux basés sur les chemins augmentants de type Ford-Fulkerson [5]. Les auteurs affirment en effet que la recherche de chemins augmentants classique est trop coûteuse en calculs, car, dans les graphes utilisés en traitement d'images, elle nécessite généralement de parcourir une grande partie des sommets (dont le nombre est de l'ordre de celui des pixels des images utilisées). Ils proposent donc de construire deux arbres, dont les racines sont la source s et le puits t, dans une étape de *croissance* des arbres. Lorsque ces deux arbres se touchent, alors un chemin augmentant est trouvé et le graphe résiduel construit. Ensuite, au lieu de repartir de la source ou du puits pour trouver un nouveau chemin augmentant, on utilise les arbres précédemment construits, en les mettant à jour : on retire les branches saturées par le flot précédent, puis on essaie de reconstituer chaque arbre en reliant les sous-arbres obtenus à l'arbre principal *via* d'autres arcs. Enfin, on poursuit la croissance des nouveaux arbres. L'intérêt majeur de cette approche est que, dans la plupart des cas, la construction des arbres reste suffisamment avancée pour gagner du temps dans la recherche d'un chemin augmentant.

4.4.3 α-*Expansion move* optimal

Une fois le flot maximal calculé, on en déduit la coupure minimale $(\mathcal{V}^s,\mathcal{V}^t)$ en construisant le graphe résiduel, puis en recherchant l'ensemble des sommets pour lesquels il existe un chemin les reliant à s. On en déduit la valeur des coefficients `vars0(p)` et `varsA(p)` pour chaque pixel p :

$$\text{Si } \texttt{vars0(p)} \notin \{\texttt{VAR\_ACTIVE},\texttt{VAR\_ABSENT}\}, \quad \texttt{vars0(p)} = \begin{cases} 1 & \text{si } \big(p,p + u^h(p)\big) \in \mathcal{V}^t \\ 0 & \text{si } \big(p,p + u^h(p)\big) \in \mathcal{V}^s \end{cases}$$

$$\text{Si } \texttt{varsA(p)} \notin \{\texttt{VAR\_ACTIVE},\texttt{VAR\_ABSENT}\}, \quad \texttt{varsA(p)} = \begin{cases} 1 & \text{si } \big(p,p + \alpha\big) \in \mathcal{V}^t \\ 0 & \text{si } \big(p,p + \alpha\big) \in \mathcal{V}^s. \end{cases}$$

On retrouve alors la valeur du vecteur de changement d'état g_α :

$$g\Big(p,p+u^h(p)\Big) = \texttt{vars0(p)} \qquad \text{et} \qquad g\Big(p,p+\alpha\Big) = \texttt{varsA(p)}.$$

Ensuite, la formule d'inversion (4.8) permet d'obtenir la configuration optimale f_α. On peut alors comparer son énergie avec celle de la configuration initiale f.

L'énergie décroît (au sens large) ; si elle n'a pas décru en une itération complète (c'est-à-dire lorsque chaque disparité $\alpha \in \mathrm{I}^h_{\mathrm{disp}}$ a été testée), alors on arrête l'algorithme. Par défaut, le programme limite le nombre maximal d'itérations à 4. Autrement dit, chaque disparité est testée au maximum 4 fois.

4.4.4 Paramètres

La méthode de Kolmogorov et Zabih repose essentiellement sur le choix de trois paramètres : K pour le terme d'occultation et λ_1 et λ_2 pour le terme de régularisation. Une heuristique pour choisir de manière automatique la valeur de ces paramètres est proposée.

Mis à part le terme d'injectivité, les termes de l'énergie E doivent être équilibrés. Sinon, l'un des trois critères risque de l'emporter sur les deux autres. Tout comme dans la méthode présentée au chapitre précédent, le choix des paramètres doit préserver cet équilibre.

Choix du paramètre d'occultation K Ce paramètre correspond au coût d'occultation payé par un pixel occulté. Il participe ainsi au terme d'attache aux données et d'occultation. Sa valeur permet donc principalement de pondérer l'influence relative de ces deux critères. Si K est choisi trop faible, alors il sera globalement plus faible que le coût de corrélation. Par conséquent, il sera globalement moins coûteux de déclarer un pixel occulté que de le mettre en correspondance avec un autre pixel. S'il est au contraire choisi trop grand, alors le phénomène inverse se produit : il est trop coûteux de rendre un pixel occulté, si bien que n'importe quelle mise en correspondance (même erronée, donc engendrant un coût de corrélation important) est préférable.

On voit donc qu'il est essentiel de régler correctement ce paramètre. Dans [8], les auteurs suggèrent de choisir K de sorte qu'en moyenne, pour un pixel, un quart des mises en correpondance seulement soit plus avantageuse que l'occultation.

Choix des paramètres de régularisation λ_1 et λ_2 Une fois que K est choisi, la valeur des deux paramètres λ_1 et λ_2 détermine l'influence relative du terme de régularité par rapport au terme d'attache aux données et d'occultation. Ils ne doivent donc pas être choisis trop grands, au risque d'augmenter de manière significative l'influence de l'attache aux données et de l'occultation, ni trop petits, car sinon les cartes obtenues seront trop régularisées.

Kolmogorov et Zabih proposent dans un premier temps de fixer $\lambda_1 = 3\,\lambda$ et $\lambda_2 = \lambda$, de sorte que la pénalité pour la régularisation soit trois fois plus importante lorsque le saut de disparité ne coïncide pas avec une discontinuité d'intensité dans l'une des deux vues. Reste alors à choisir la valeur de λ. Il est proposé dans [8] de le choisir égal à $K/5$.

4.5 Adapter l'intervalle de disparité au pixel

Nous proposons dans cette section une modification de l'algorithme de Kolmogorov et de Zabih qui permet d'adapter l'intervalle de disparité à chaque pixel. Cette modification permettra entre autre de proposer un algorithme de densification et une approche de raffinement subpixellique.

4.5.1 Intervalle adaptatif

Principe L'idée est de spécifier pour chaque pixel p de l'image de référence l'ensemble $\mathrm{I}^h_{\mathrm{disp}}(p) \subset \mathbb{R}$ des valeurs de disparité qu'il peut prendre, ainsi que de l'autoriser ou non à être déclaré occulté. Cela revient pour tout pixel $p = (i,j) \in \mathcal{I}_{\mathrm{L}}$ à imposer pour la carte de disparité de satisfaire

$$u^h_{i,j} \in \mathrm{I}^h_{\mathrm{disp}}(p) \cup \{\texttt{occ}\} \qquad \text{ou} \qquad u^h_{i,j} \in \mathrm{I}^h_{\mathrm{disp}}(p).$$

On notera désormais $\tilde{\mathrm{I}}^h_{\mathrm{disp}}(p)$ l'ensemble des valeurs (incluant éventuellement l'étiquette occ) que peut prendre la disparité $u^h_{i,j}$. On le supposera non vide pour tout pixel p.

On se propose alors de considérer le problème

$$\min_{u^h_{i,j} \in \tilde{\mathrm{I}}^h_{\mathrm{disp}}(p)} \mathrm{E}(u^h)$$

où la fonctionnelle d'énergie E est celle introduite dans la section 4.1. Au lieu de chercher à la minimiser, on la fait à nouveau décroître par *expansion moves*, en sélectionnant pour chaque $\alpha \in \mathbb{R}$ l'α-*expansion move* de plus faible énergie. Montrons que la recherche de l'α-*expansion move* optimal reste représentable par un graphe, à condition d'utiliser à nouveau le vecteur de changement d'état.

Avant cela, il est nécessaire de modifier la définition des assignements :

$$\mathcal{A} = \Big\{ a = (p,q) \in \mathcal{I}_{\mathrm{L}} \times \mathcal{I}_{\mathrm{R}} \mid q - p \in \mathrm{I}^h_{\mathrm{disp}}(p) \times \{0\} \Big\}.$$

Occultation permise Commençons par étudier le cas où seules les valeurs de disparité sont spécifiées pour chaque pixel (tout pixel restant autorisé à être occulté). Il s'agit donc de considérer le problème

$$\min_{u^h_{i,j} \in \mathrm{I}^h_{\mathrm{disp}}(p) \times \{\texttt{occ}\}} \mathrm{E}(u^h).$$

Soit $\alpha \in \mathbb{R}$. Explicitons les α-*expansion moves* $(u^h)'$ d'une carte de disparité u^h. Il est tout d'abord clair que si α n'appartient à aucun des intervalles de disparité $\mathrm{I}^h_{\mathrm{disp}}(p)$, alors les seules modifications autorisées consistent à déclarer occultés certains pixels :

$$\forall\, (i,j) \in \Omega^h, \qquad (u^h)'_{i,j} \in \{u^h_{i,j}, \texttt{occ}\}.$$

Cette opération pouvant être réalisée pour n'importe quelle valeur de α, on va dorénavant ignorer ces α-*expansion moves*. Supposons donc qu'il existe au moins un intervalle $\mathrm{I}^h_{\mathrm{disp}}(p)$ contenant α. Les α-*expansion moves* $(u^h)'$ considérées doivent vérifier

$$\forall\, (i,j) \in \Omega^h, \qquad \begin{cases} (u^h)'_{i,j} = \alpha & \text{si } u^h_{i,j} = \alpha \\ (u^h)'_{i,j} \in \{u^h_{i,j}, \texttt{occ}\} & \text{si } \alpha \notin \mathrm{I}^h_{\mathrm{disp}}(p) \text{ et } u^h_{i,j} \neq \alpha \\ (u^h)'_{i,j} \in \{u^h_{i,j}, \alpha, \texttt{occ}\} & \text{si } \alpha \in \mathrm{I}^h_{\mathrm{disp}}(p) \text{ et } u^h_{i,j} \neq \alpha. \end{cases}$$

En termes de configurations et d'assignements, les configurations f' associées sont celles qui vérifient

$$\forall\, a \in \mathcal{A}, \quad f'(a) = \begin{cases} 1 & \text{si } f(a) = 1 \text{ et } d(a) = \alpha \\ 0 & \text{si } f(a) = 0 \text{ et } d(a) \neq \alpha \end{cases}$$

où f est la configuration associée à la carte de disparité initiale u^h. Ainsi, on voit que la définition des *expansion moves* reste inchangée, seul l'ensemble des assignements $\mathcal{A}$ sur lequel les configurations sont définies varie. Toute l'étude des sections précédentes reste donc valable (puisqu'aucune condition n'est requise sur l'ensemble $\mathcal{A}$). La recherche de l'α-*expansion move* de f d'énergie minimale est donc un problème représentable par un graphe, en effectuant le changement de variable (4.7).

Si $\tilde{\mathrm{I}}^h_{\mathrm{disp}}(p) = \mathrm{I}^h_{\mathrm{disp}} \cup \{\texttt{occ}\}$ pour tout p, alors on retrouve bien la méthode originale.

Occultation interdite Supposons à présent que certains pixels ne sont pas autorisés à être occultés. Les α-*expansion moves* s'écrivent alors

$$\forall\,(i,j) \in \Omega^h, \quad \begin{cases} (u^h)'_{i,j} = \alpha & \text{si } u^h_{i,j} = \alpha \\ (u^h)'_{i,j} = u^h_{i,j} & \text{si } \alpha \notin \mathrm{I}^h_{\mathrm{disp}}(p) \text{ et } u^h_{i,j} \neq \alpha \text{ et } \texttt{occ} \notin \tilde{\mathrm{I}}_{\mathrm{disp}}(p) \\ (u^h)'_{i,j} \in \{u^h_{i,j}, \texttt{occ}\} & \text{si } \alpha \notin \mathrm{I}^h_{\mathrm{disp}}(p) \text{ et } u^h_{i,j} \neq \alpha \text{ et } \texttt{occ} \in \tilde{\mathrm{I}}_{\mathrm{disp}}(p) \\ (u^h)'_{i,j} \in \{u^h_{i,j}, \alpha\} & \text{si } \alpha \in \mathrm{I}^h_{\mathrm{disp}}(p) \text{ et } u^h_{i,j} \neq \alpha \text{ et } \texttt{occ} \notin \tilde{\mathrm{I}}_{\mathrm{disp}}(p) \\ (u^h)'_{i,j} \in \{u^h_{i,j}, \alpha, \texttt{occ}\} & \text{si } \alpha \in \mathrm{I}^h_{\mathrm{disp}}(p) \text{ et } u^h_{i,j} \neq \alpha \text{ et } \texttt{occ} \in \tilde{\mathrm{I}}_{\mathrm{disp}}(p). \end{cases}$$

Les pixels ne changeant pas de disparité sont donc ceux qui sont de disparité α ou ceux qui ne peuvent ni adopter la disparité α, ni être occulté. Traduisons cette condition en termes de configurations :

$$\forall\, a = (p,q) \in \mathcal{A}, \quad f'(a) = \begin{cases} 1 & \text{si } f(a) = 1 \text{ et } d(a) = \alpha \\ 0 & \text{si } f(a) = 0 \text{ et } d(a) \neq \alpha \\ 1 & \text{si } f(a) = 1 \text{ et } \alpha \notin \mathrm{I}^h_{\mathrm{disp}}(p) \text{ et } \texttt{occ} \notin \tilde{\mathrm{I}}_{\mathrm{disp}}(p) \\ 1 & \text{si } f(a) = 1 \text{ et } \alpha \in \mathrm{I}^h_{\mathrm{disp}}(p) \text{ et } \texttt{occ} \notin \tilde{\mathrm{I}}_{\mathrm{disp}}(p) \\ & \quad d(a) \neq \alpha \text{ et } f(p,p+\alpha) = 0. \end{cases}$$

Les deux dernières conditions empêchent pour un pixel p interdit d'occultation que l'assignement $(p,p+u^h(p))$ (qui est nécessairement actif) ne soit désactivé si l'assignement $(p,p+\alpha)$ n'est activé en échange. Ainsi, si le pixel p ne peut être mis en correspondance avec le pixel $p+\alpha$ (soit que celui-ci n'est pas un pixel de l'image de droite, soit que la valeur α n'est pas une disparité admissible pour le pixel p), alors l'assignement $(p,p+u^h(p))$ doit rester actif. Par ailleurs, si les deux assignements $(p,p+u^h(p))$ et $(p,p+\alpha)$ existent, alors l'un des deux doit être actif.

Cette contrainte revient donc à introduire pour tous les assignements actifs, de la forme $a = (p,p+u^h(p))$, avec p non occultable, le terme suivant dans l'énergie $\mathrm{E}^{\alpha,f}$:

$$\chi_{\{(0,0),(0,1),(1,1)\}}\Big(g'(p,p+\alpha), g'(a)\Big).$$

En d'autres termes, on peut remplacer dans le terme d'injectivité le terme correspondant par

$$\chi_{\{(0,0),(1,1)\}}\Big(g'(p,p+\alpha), g'(a)\Big)$$

qui impose aux deux assignements (s'ils existent) a et $(p,p+\alpha)$ de changer d'état tous les deux ou de rester inchangés tous les deux ; lorsque l'assignement $(p,p+\alpha)$ n'existe

pas, l'état de l'assignement a reste inchangé. Ce nouveau terme est clairement sous-modulaire, puisque qu'il vaut 0 en (0,0) et (1,1) et prend une valeur infinie pour (0,1) et (1,0). L'énergie résultante est donc représentable.

Modification de l'algorithme KZ2 On voit donc que spécifier l'intervalle de disparité pour chaque pixel suppose de modifier l'ensemble des assignements $\mathcal{A}$, ce qui affecte la construction des sommets du graphe. Supposons dans un premier temps que le pixel p admette α comme valeur de disparité. Si l'occultation est autorisée, alors on ne change rien par rapport à l'algorithme original. Si l'occultation est interdite, il faut ajouter un arc entre les assignements $(p,p+u^h(p))$ et $(p,p+\alpha)$ pour interdire les changements d'état isolés ; plus précisément, on relie ces deux sommets par deux arcs opposés, tous deux de capacités infinies. Cela force les deux sommets à ne pas être séparés par une coupure.

Si $\alpha \notin \mathrm{I}^h_{\mathrm{disp}}(p)$, alors il faut distinguer deux cas. Si l'occultation est interdite, alors l'assignement $(p,p+u^h(p))$ reste actif pendant les α-*expansion moves* ; on ne construit donc pas de sommet pour cet assignement. Dans le code, cela revient à affecter les valeurs

`vars0(p) = VAR_ACTIVE` et `varsA(p) = VAR_ABSENT`.

Notons que, dans l'algorithme original, cette combinaison n'apparaît jamais. Si l'occultation est autorisée, alors il faut construire le sommet correspondant, mais aucun sommet pour le couple $(p,p+\alpha)$, puisque celui-ci n'est pas un assignement. On construit donc

`vars0(p) = o` et `varsA(p) = VAR_ABSENT`

ce qui revient à considérer que le pixel $p+\alpha$ n'existe pas dans l'image de droite. Il faut noter que l'introduction d'assignements actifs et non-désactivable, mais de disparité différente de α induit quelques modifications dans le code, car cette situation n'apparaît pas dans le code original. Elle conduit en particulier à ne pas construire de sommet pour l'assignement $(p,p+\alpha)$ lorsque $p+\alpha$ est mis en correspondance avec un pixel p' qui n'est pas amené à changer d'état pendant l'α-*expansion move*, sous peine de violer la contrainte d'unicité.

4.5.2 Densification de cartes de disparité

Motivation Les méthodes locales proposent des cartes de disparité obtenues de manière efficace (voir chapitre 2). Malheureusement, elles n'intègrent aucun modèle global de régularité, et génèrent donc beaucoup d'erreurs, notamment dans les zones d'occultation, les zones non texturées ou bruitées, les textures sujettes à l'effet de Strobes (cf. section 2.1.4). Ces erreurs peuvent être détectées par des tests simples comme le filtre LRRL qui vérifie la cohérence des cartes de disparité de la vue de droite et de la vue de gauche [10], ou bien encore par des procédés plus complexes comme ceux basés sur le principe *a contrario* [11]. Après ce traitement, les pixels mal estimés sont rejetés et la carte de disparité résultante devient éparse (la disparité n'est plus connue partout).

Il peut alors être intéressant d'exploiter un modèle de régularité pour densifier (c'est-à-dire interpoler) ces cartes. En d'autres termes, utiliser une méthode globale (et sa fonctionnelle d'énergie) pour interpoler la carte éparse. Si le nombre de pixels à interpoler est relativement faible, ou que la méthode globale utilisée est efficace,

cette étape de densification nécessitera un temps de calcul raisonnable. On se propose d'adapter l'algorithme KZ2 à cette fin.

Choix des intervalles Utiliser la méthode de Kolmogorov et Zabih pour densifier une carte éparse revient à rechercher une carte de disparité d'énergie minimale E, où seuls les pixels rejetés par le(s) filtre(s) d'erreur sont laissés libres de prendre n'importe quelle valeur de disparité (ou éventuellement d'être occultés). Soit u_0^h une carte de disparité éparse que l'on cherche à densifier. Les pixels dont la disparité n'est pas connue sont étiquetés par `unknown`. Ici, nous supposerons que les pixels non rejetés (dont la disparité est connue) ne sont pas occultés, et nous cherchons à densifier la carte tout en détectant les occultations. Aussi, la densification peut se traduire en termes d'intervalles de disparité adaptatifs. Pour les pixels p de disparité connue $u_0^h(p)$ dans la carte de disparité initiale éparse, on interdit tout changement, ce qui revient à poser

$$\forall\, p=(i,j)\in\mathcal{I}_{\mathrm{L}},\qquad (u_0^h)_{i,j}\neq \texttt{unknown}\qquad\Longrightarrow\qquad \tilde{\mathrm{I}}^h_{\mathrm{disp}}(p)=\left\{(u_0^h)_{i,j}\right\}.$$

Le pixel ne peut donc pas être déclaré occulté non plus. Pour les autres pixels, dont la disparité reste à estimer, on pose

$$\forall\, p=(i,j)\in\mathcal{I}_{\mathrm{L}},\qquad (u_0^h)_{i,j}=\texttt{unknown}\qquad\Longrightarrow\qquad \tilde{\mathrm{I}}^h_{\mathrm{disp}}(p)=\mathrm{I}^h_{\mathrm{disp}}\cup\{\texttt{occ}\}.$$

Ainsi, on ne contraint pas leur valeur et ils peuvent être occultés.

Algorithme La procédure pour une densification à l'aide de la méthode proposée peut donc être décrite par l'algorithme 3.

Algorithme 3: Densification par *expansion moves*

Entrée : u_0^h une carte de disparité initiale éparse, $\mathrm{I}^h_{\mathrm{disp}}$ l'intervalle de disparité
Sortie : u^h une carte de disparité dense (avec occultations)

begin
  foreach *pixel* $p=(i,j)\in\mathcal{I}_L$ **do**
    if $(u_0^h)_{i,j}\neq$ ***unknown*** **then** $\tilde{\mathrm{I}}^h_{\mathrm{disp}}(p)=\left\{(u_0^h)_{i,j}\right\}$
    else $\tilde{\mathrm{I}}^h_{\mathrm{disp}}(p)=\mathrm{I}^h_{\mathrm{disp}}\cup\{\texttt{occ}\}$
  $u^h\leftarrow u_0^h$
  while *l'énergie E décroît* **do**
    foreach *valeur de disparité* $\alpha\in\mathrm{I}^h_{\mathrm{disp}}$ **do**
      $u^h\leftarrow \underset{\substack{(u^h)'\ \alpha\text{-}\textit{expansion move}\ \text{de}\ u^h\\ (u_h)'_{i,j}\in\tilde{\mathrm{I}}^h_{\mathrm{disp}}(p)}}{\operatorname{argmin}} \mathrm{E}\big((u^h)'\big)$

Utilisation de points fiables Cette méthode permet également d'incorporer des informations que l'on possède *a priori* sur la carte de disparité. Il peut s'agit de points dont une autre méthode (de mise en correspondance par corrélation ou de mesure directe) a permis d'établir de manière fiable la disparité.

Estimation des occultations à partir d'une carte dense Il est également possible d'utiliser cet algorithme pour estimer les occultations dans une estimation dense de la carte de disparité. Pour cela, on affecte à chaque pixel p de disparité $d(p)$ le singleton $\{d(p)\}$ comme intervalle de disparité (avec éventuellement une quantification préalable si l'estimation initiale est subpixellique) et on autorise le pixel à devenir occulté.

4.5.3 Raffinement subpixellique

Motivation La précision des cartes de disparité peut être un sujet important. Malheureusement, augmenter la précision peut être très coûteux. Pour une précision au demi-pixel par exemple, la taille de l'intervalle de disparité est doublée. La mise en correspondance passe ensuite généralement par un sur-échantillonnage (de facteur 2 dans le cas de la précision au demi-pixel) horizontal d'au moins une des images. Le volume de coût double de volume. Pour les méthodes locales, on a pour chaque pixel de l'image de référence deux fois plus de corrélations à calculer. Pour une méthode globale comme celle proposée au chapitre précédent, on doit manipuler 4 (ou 5) volumes de taille deux fois plus grands que pour la précision pixellique (ce qui implique en autres deux fois plus de projections à calculer). La convergence, qui est liée à la taille du problème, est également plus lente. Pour l'algorithme KZ2, il y a deux fois plus de disparités α à tester dans les *expansion moves*. Calculer une carte de disparité à précision subpixellique peut donc très vite devenir techniquement limité.

Une stratégie pour limiter la complexité du problème est de *raffiner* une carte de disparité pixellique. L'idée est de partir d'une carte de précision pixellique, puis de relâcher pour chaque pixel l'intervalle de recherche autour de la disparité pixellique précédemment estimée, en incluant cette fois des valeurs subpixelliques. Dans le cas idéal où la carte pixellique serait correcte, un raffinement subpixellique d'un demi pixel consisterait pour tout pixel non occulté $p = (i,j)$ de disparité pixellique $(u_0^h)_{i,j}$ à tester les disparités $\{(u_0^h)_{i,j}, (u_0^h)_{i,j} + 0{,}5, (u_0^h)_{i,j} - 0{,}5\}$ et à retenir la carte d'énergie minimale sous cette contrainte. Ainsi, pour chaque pixel non occulté, seul un nombre faible (généralement $2\,N + 1$) de disparité α est testé.

Choix des intervalles Supposons que l'on cherche à estimer la disparité avec une précision de $h_t = 1/N$ pixel, où N est un entier non nul (avec $h_t = 1/2$ dans le cas de la précision au demi-pixel, $h_t = 1/4$ pour le quart de pixel par exemple). Si on a une carte de disparité pixellique u_0^h obtenue par une méthode quelconque, alors on définit pour chaque pixel $p = (i,j)$ non occulté

$$(u_0^h)_{i,j} \neq \texttt{occ} \qquad \Longrightarrow \qquad \tilde{\mathrm{I}}^h_{\mathrm{disp}}(p) = \Big\{(u_0^h)_{i,j} + h_t \, [\![-N+1 \,;\, N-1]\!] \Big\} \cup \Big\{\texttt{occ}\Big\}.$$

Il est possible d'interdire l'occultation (si la carte des occultations est fiable par exemple), ou au contraire, si la carte initiale n'est pas suffisamment fiable, de relâcher légèrement l'intervalle de disparité en remplaçant $N-1$ par un entier plus grand. Pour les pixels occultés (s'il y en a), on peut soit interdire toute modification (si la carte des occultations est vraiment fiable), ou plus simplement de ne pas contraindre la disparité

$$(u_0^h)_{i,j} = \texttt{occ} \qquad \Longrightarrow \qquad \tilde{\mathrm{I}}^h_{\mathrm{disp}}(p) = \Big\{\mathrm{I}^h_{\mathrm{disp}} + h_t \, [\![-N+1 \,;\, N-1]\!] \Big\} \cup \Big\{\texttt{occ}\Big\}.$$

Il y a donc théoriquement N fois plus d'α-*expansion moves* à tester dans une itération, mais chacun d'entre eux n'affecte qu'une petite partie des assignements (ceux dont le

premier élément $p = (i,j)$ a pour disparité pixellique $(u_0^h)_{i,j} \in \{\lfloor\alpha\rfloor, \lceil\alpha\rceil\}$ et ceux qui sont initialement occultés).

En pratique, on utilisera le raffinement pour doubler la précision. Si la disparité d^N a été estimée à une précision $1/N$, alors chaque pixel p non occulté se voit affecté l'intervalle $d^N(p) + \{,-1,0,1\}/(2N)$. Ainsi, pour obtenir une estimation d'une précision au quart de pixel par exemple, il faut commencer par raffiner au demi-pixel la carte pixellique, puis raffiner au quart de pixel la carte raffinée obtenue.

Algorithme Le raffinement subpixellique d'une carte de disparité de précision pixellique est décrit par l'algorithme 4.

Algorithme 4: Raffinement sous-pixellique par *expansion moves*

Entrée : u_0^h une carte de disparité initiale de précision pixellique,
$\mathrm{I}_{\mathrm{disp}}^h$ l'intervalle de disparité pixellique, $h_t = 1/K$ la précision souhaitée
Sortie : u^h une carte de disparité sous-pixellique de précision h_t

begin
  foreach *pixel* $p = (i,j) \in \mathcal{I}_L$ **do**
    if $(u_0^h)_{i,j} \neq \texttt{occ}$ **then** $\tilde{\mathrm{I}}_{\mathrm{disp}}^h(p) = \left\{(u_0^h)_{i,j} + h_t \,[\![-K+1\,;\, K-1]\!]\right\} \cup \left\{\texttt{occ}\right\}$
    else $\tilde{\mathrm{I}}_{\mathrm{disp}}^h(p) = \left\{\mathrm{I}_{\mathrm{disp}}^h + h_t \,[\![-K+1\,;\, K-1]\!]\right\} \cup \left\{\texttt{occ}\right\}$
  $u^h \leftarrow u_0^h$
  while *l'énergie E décroît* **do**
    foreach *valeur de disparité* $\alpha \in \mathrm{I}_{\mathrm{disp}}^h + h_t \,[\![-K+1\,;\, K-1]\!]$ **do**
      $u^h \leftarrow \underset{\substack{(u^h)' \ \alpha\text{-expansion move de } u^h \\ (u_h)'_{i,j} \in \tilde{\mathrm{I}}_{\mathrm{disp}}^h(p)}}{\operatorname{argmin}} \mathrm{E}\left((u^h)'\right)$

4.6 Résultats expérimentaux

On présente dans cette section les résultats obtenus avec la méthode étudiée dans ce chapitre. Le code utilisé est soit celui proposé dans la publication IPOL [7] pour l'algorithme original, soit une version modifiée de celui-ci pour l'intervalle de disparité adaptatif (modifié suivant les remarques de la section 4.5).

4.6.1 Algorithme original

Commençons par présenter les résultats obtenus avec l'algorithme original sur les paires du banc d'essai Middlebury.

Paramètres automatiques La figure 4.3 présente les cartes de disparité (colonne de gauche) obtenues en choisissant les paramètres suivant l'heuristique proposée dans le paragraphe 4.4.4. Leurs valeurs sont indiquées pour chaque paire, ainsi que les taux d'erreur pixellique (supérieure à un pixel), Middlebury (strictement supérieure au pixel) et subpixellique (supérieure au demi-pixel). Le masque affiché en cyan (colonne du milieu) correspond à l'erreur pixellique. Dans tous les cas, seuls les pixels non occultés

(d'après la vérité-terrain) sont pris en compte. Enfin, on présente (colonne de droite) également la carte des détections d'occultations avec les taux de précision et de rappel (voir Chapitre 3).

Nombre d'itérations, temps de calcul Le temps de calcul est donné dans le tableau 4.4, ainsi que le nombre d'itérations nécessaires. Par défaut, le nombre d'itérations est limité à 4. La figure 4.5 permet de comparer les cartes obtenues pour moins de 4 itérations et celles obtenues pour moins de 8 itérations. À titre d'information, on donne pour chaque paire la taille du volume de coût (voir chapitre 3), qui est un volume de taille $N_x \times N_y \times N_t$, où $N_x \times N_y$ correspond aux dimensions de l'image de référence et N_t le nombre de disparité dans l'intervalle de disparité $\mathrm{I}^h_{\mathrm{disp}}$.

Choix des paramètres On rappelle que la valeur automatique des paramètres suit les lois suivantes :

$$\lambda = \frac{K}{5}, \qquad \lambda_1 = 3\,\lambda \qquad \text{et} \qquad \lambda_2 = \lambda$$

où K est le coût d'occultation, λ_1 le coût de régularité lorsque les variations de couleur sont faibles et λ_2 le coût de régularité lorsqu'elles sont importantes. Il est donc possible de modifier manuellement la valeur de ces paramètres de différentes manières.

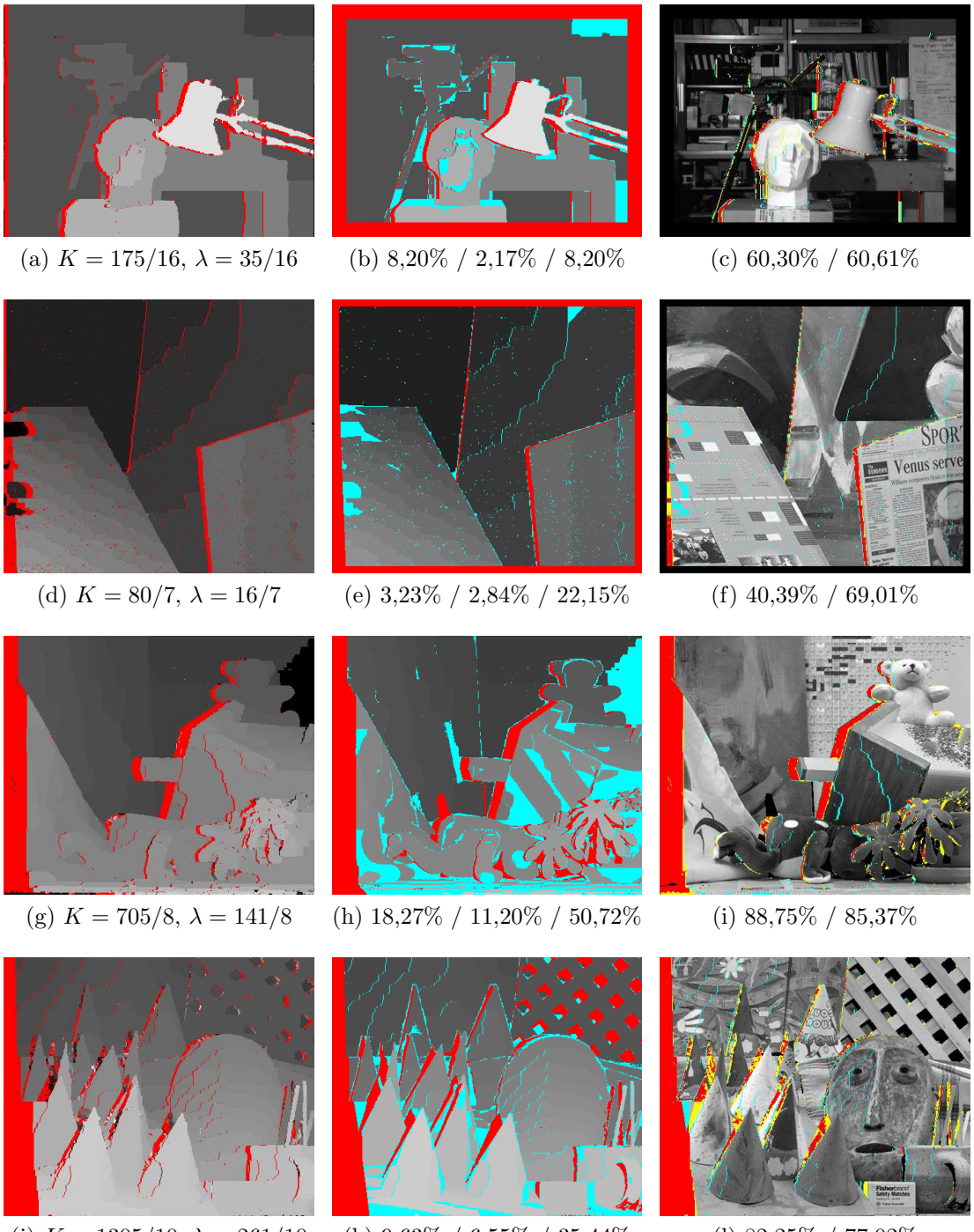
On commence par modifier la valeur de K, en conservant toutes les dépendances listées plus haut. Ainsi, plus K est grand, plus l'occultation est coûteuse, mais c'est également le cas des discontinuités de disparité. Au contraire, si K est plus faible, alors l'occultation tout comme les sauts de disparité sont moins coûteux. On choisit de modifier la valeur de K de deux façons différentes : soit en doublant (ou en divisant par 2) la valeur déterminée automatiquement par l'heuristique de Kolmogorov et Zabih, soit en modifiant directement cette heuristique, en demandant à ce que K soit choisi de sorte qu'en moyenne un tiers (resp. un cinquième) des corrélations soient plus coûteuses que l'occultation (au lieu d'un quart dans l'algorithme original), ce qui a pour effet d'augmenter (resp. diminuer) la valeur de K. Les résultats obtenus par cette procédure sont présentés à la figure 4.6.

Il est ensuite possible de modifier le rapport entre la valeur de K et celle de λ (en conservant les définitions de λ_1 et de λ_2). Par défaut, ce rapport est de 5, on choisit donc de le doubler ou de le diminuer de moitié. Dans cette expérience, la valeur de K reste celle déterminée automatiquement par l'algorithme. Cette modification a pour effet de changer le poids relatif du terme d'attache aux données + occultation et du terme de régularité. La figure 4.7 montre les résultats obtenus suite à cette modification.

On peut également modifier le rapport entre λ_1 et λ_2. Plus précisément, on conserve la valeur de K, celle de λ et de λ_1, et on change le facteur dans la définition de λ_1, en le diminuant ou en l'augmentant. Cela revient à pénaliser davantage les discontinuités de disparité qui ne coïncident pas avec une discontinuité de couleur. Les cartes obtenues sont présentées dans la figure 4.8.

On peut enfin modifier le seuil qui détermine si le coût de régularité est λ_1 ou λ_2. Par défaut, il vaut 8. Ce seuil correspond à la sensibilité du détecteur de bords dans l'image : plus il est important, plus il y a de bords détectés. Les résultats obtenus à la suite de ces modifications sont présentés à la figure 4.9.

Pour terminer, on notera que le calcul du paramètre K dépend de l'estimation de l'intervalle de disparité. La figure 4.10 présente les cartes obtenues pour deux (sur-)estimations différentes de cet intervalle. Les paramètres sont alors calculés automatiquement (en suivant l'heuristique décrite plus haut).



(a) $K = 175/16$, $\lambda = 35/16$ (b) 8,20% / 2,17% / 8,20% (c) 60,30% / 60,61%

(d) $K = 80/7$, $\lambda = 16/7$ (e) 3,23% / 2,84% / 22,15% (f) 40,39% / 69,01%

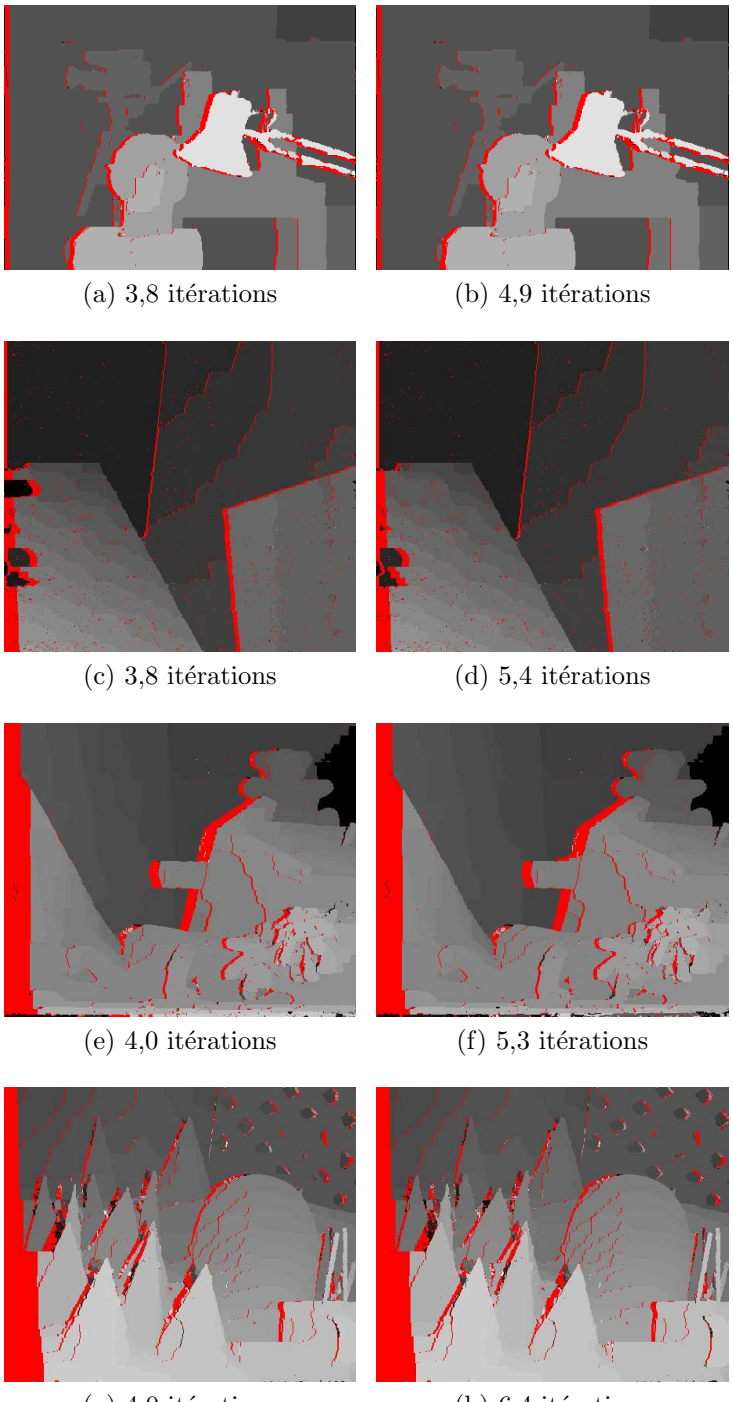
(g) $K = 705/8$, $\lambda = 141/8$ (h) 18,27% / 11,20% / 50,72% (i) 88,75% / 85,37%

(j) $K = 1305/10$, $\lambda = 261/10$ (k) 9,63% / 6,55% / 35,44% (l) 82,25% / 77,02%

FIGURE 4.3 – Résultats obtenus à partir de l'algorithme original KZ2 (précision **pixellique**). Colonne de gauche : carte de disparité, avec en rouge les occultations détectées. En légende : les paramètres estimés automatiquement. Colonne du milieu : erreur d'estimation. En rouge, le masque des points dont la disparité n'est pas connue (d'après la vérité-terrain fournie par Middlebury). En cyan, les disparités mal estimées pour l'erreur pixellique (supérieure ou égale à 1). En légende : le pourcentage d'erreur pixellique / erreur Middlebury (strictement supérieure à 1) / erreur sous-pixellique (supérieure ou égale à 0,5) dans les zones occultées (d'après la vérité-terrain). Les pixels déclarés occultés par KZ2 alors qu'ils ne le sont pas sont compabilisés comme des erreurs dans tous les cas. Colonne de droite : erreur dans la détection des occultations (en rouge les détections correctes, en jaune les faux négatifs et en cyan les faux positifs). En légende : le taux de précision et de rappel. De haut en bas : Tsukuba, Venus, Teddy et Cones.

Paire	Tsukuba	Venus	Teddy	Cones
Taille du volume de coût (en mégapixels)	1,8	3,3	10,1	10,1
Temps d'exécution (en secondes)	5	12	32	28
Nombre d'itérations	3,8	3,8	4,0	4,0

FIGURE 4.4 – Temps d'exécution et nombre d'itérations pour la méthode proposée. La taille du volume de coût est donnée par $N_x \times N_y \times N_t$, où $N_X \times N_y$ est la taille de l'image de référence et N_t la taille de l'intervalle de disparité.



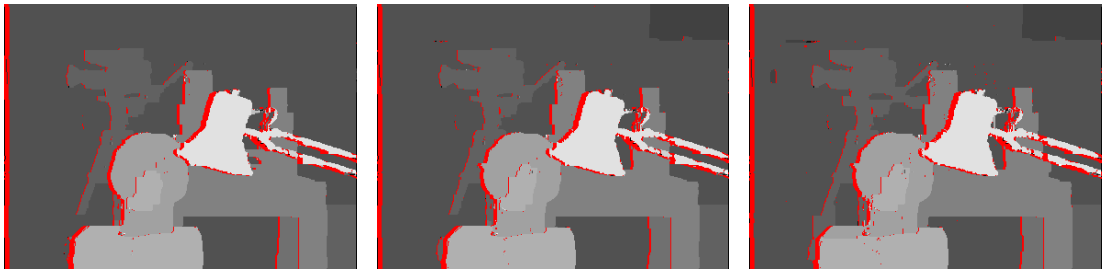
(a) 3,8 itérations (b) 4,9 itérations

(c) 3,8 itérations (d) 5,4 itérations

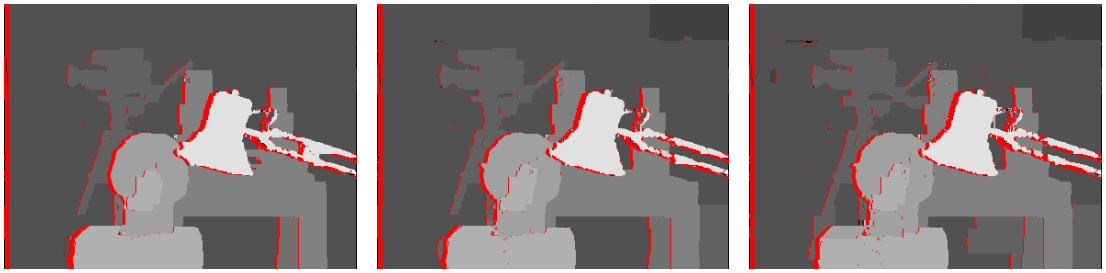
(e) 4,0 itérations (f) 5,3 itérations

(g) 4,0 itérations (h) 6,4 itérations

FIGURE 4.5 – Résultats obtenus à partir de l'algorithme KZ2, pour deux nombres d'itérations maximaux différents (4 pour la colonne de gauche, 8 pour celle de droite). En légende : le nombre d'itérations effectif. De haut en bas : Tsukuba, Venus, Teddy, Cones.

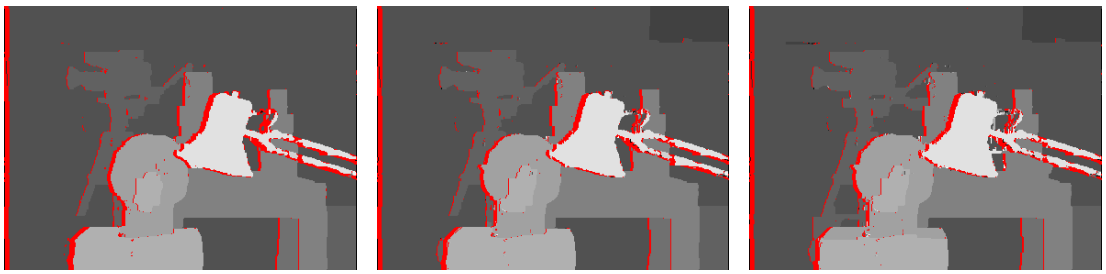


(a) $K = 22$, $\lambda = K/5$ (b) $K = 175/16$, $\lambda = K/5$ (c) $K = 11/2$, $\lambda = K/5$



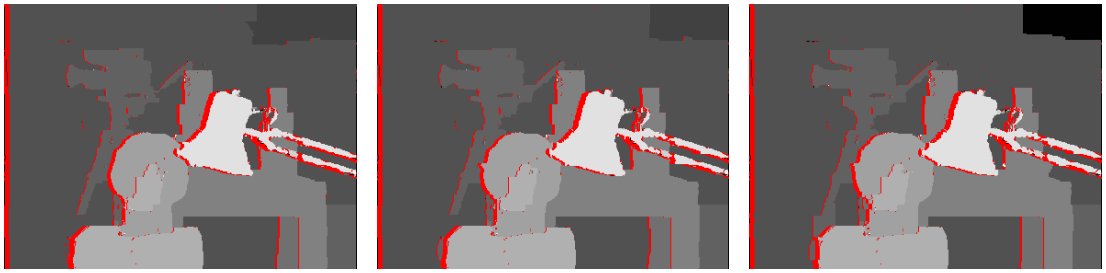
(d) $K = 430/13$, $\lambda = K/5$ (e) $K = 175/16$, $\lambda = K/5$ (f) $K = 79/16$, $\lambda = K/5$

FIGURE 4.6 – Choix de K. On modifie la valeur de K de deux manières : en doublant/réduisant de moitié la valeur calculée automatiquement par l'algorithme (ligne du haut, colonne de gauche et colonne de droite respectivement) ; en choisissant K de sorte qu'en moyenne, un tiers (resp. un cinquième) des corrélations soit plus avantageux que l'occultation (colonne de gauche et de droite, respectivement). Au milieu : résultat obtenu avec K par défaut.



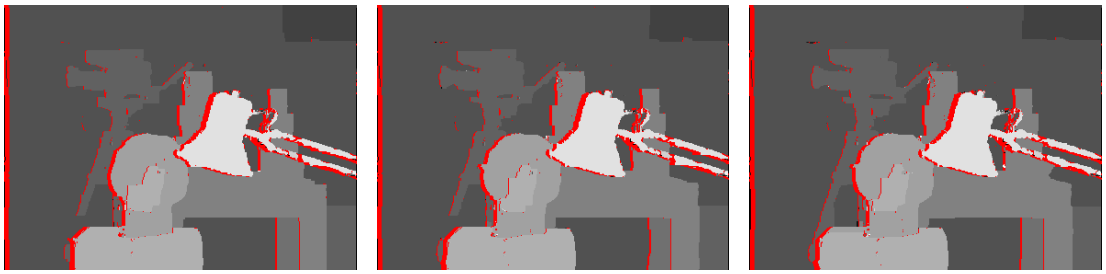
(a) $K = 175/16$, $\lambda = 2K/5$ (b) $K = 175/16$, $\lambda = K/5$ (c) $K = 175/16$, $\lambda = K/10$

FIGURE 4.7 – Rapport entre le coût d'occultation K et le coût de régularité λ. On modifie le facteur de proportionnalité entre ces deux paramètres (en laissant les autres paramètres dépendant de la nouvelle valeur de λ) : en doublant la valeur de λ, ce qui revient à considérer $\lambda = 2K/5$ (colonne de gauche), soit en la réduisant de moitié, en choisissant $\lambda = K/10$ (colonne de droite). Le paramètre K reste inchangé pour les trois expériences. La colonne du milieu correspond au résultat obtenu avec les paramètres par défaut.



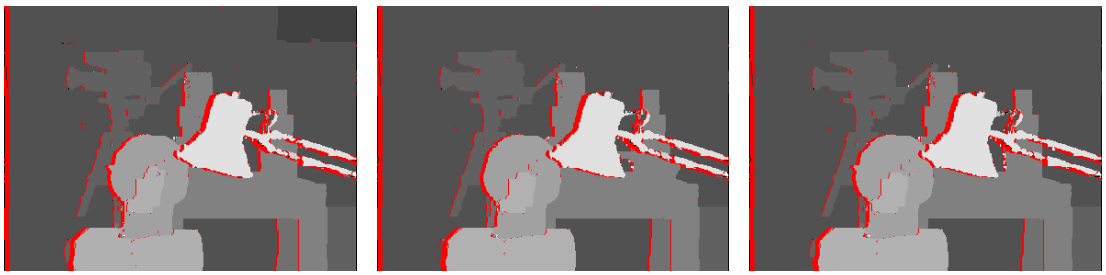
(a) $\lambda_1 = 6\lambda$, $\lambda_2 = \lambda$ (b) $\lambda_1 = 3\lambda$, $\lambda_2 = \lambda$ (c) $\lambda_1 = 1{,}5\lambda$, $\lambda_2 = \lambda$

FIGURE 4.8 – Rapport entre le coût de régularité λ_1 (sans discontinuité de couleur) et le coût de régularité λ_2 (avec discontinuité de couleur). On modifie le facteur de proportionnalité entre ces deux paramètres : en doublant la valeur de λ_1, ce qui revient à considérer $\lambda_1 = 6\lambda$ (colonne de gauche), soit en la réduisant de moitié, en choisissant $\lambda = 1{,}5\lambda$ (colonne de droite). Les paramètres K et λ_2 restent inchangés pour les trois expériences. La colonne du milieu correspond au résultat obtenu avec les paramètres par défaut.



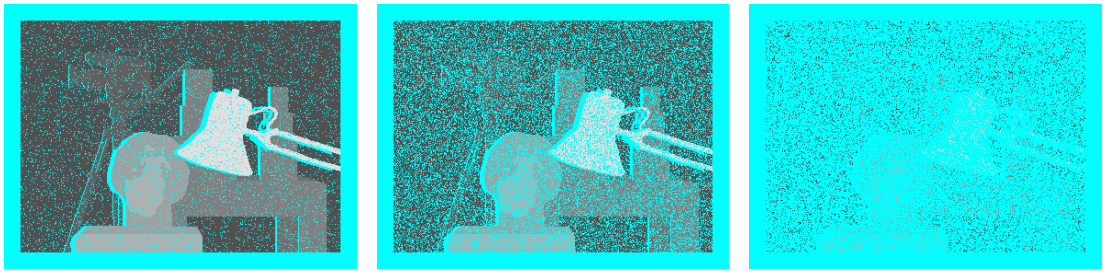
(a) seuil 16 (b) seuil 8 (c) seuil 4

FIGURE 4.9 – Sensibilité du détecteur de bords. On modifie le seuil dans la définition du coût de régularité, qui décide si la pénalité est de λ_1 ou de λ_2. Il vaut par défaut 8. Les paramètres K, λ_1 et λ_2 restent inchangés pour les trois expériences. La colonne du milieu correspond au résultat obtenu avec les paramètres par défaut.



(a) $\mathrm{I}_{\mathrm{disp}} = [\![0\,;\, 16]\!]$ (b) $\mathrm{I}_{\mathrm{disp}} = [\![0\,;\, 18]\!]$ (c) $\mathrm{I}_{\mathrm{disp}} = [\![0\,;\, 20]\!]$

FIGURE 4.10 – Influence de l'estimation de l'intervalle de disparité. La colonne de gauche correspond au résultat obtenu avec les paramètres par défaut (estimation optimale de l'intervalle de disparité).



(a) (b) (c)

FIGURE 4.11 – Exemples de cartes éparses. À gauche, on a injecté 70% de la vérité-terrain comme information connue. Au milieu, ce taux est de 50%. À droite, il est de 10%.

(a) 70% d'informations (b) 0,14% / 0,12% / 0,14% (c) 96,56% / 91,10%

(d) 50% d'informations (e) 0,74% / 0,80% / 0,74% (f) 83,34% / 84,22%

(g) 10% d'informations (h) 2,44% / 1,64% / 2,44% (i) 67,68% / 69,31%

FIGURE 4.12 – **Densification** de la vérité-terrain. Colonne de gauche : carte de disparité, avec en rouge les occultations détectées. Colonne du milieu : erreur d'estimation. En rouge, le masque des points dont la disparité n'est pas connue (d'après la vérité-terrain fournie par Middlebury). En cyan, les disparités mal estimées pour l'erreur pixellique (supérieure ou égale à 1). En légende : le pourcentage d'erreur pixellique / erreur Middlebury (strictement supérieure à 1) / erreur sous-pixellique (supérieure ou égale à 0,5) dans les zones occultées (d'après la vérité-terrain). Les pixels déclarés occultés par KZ2 alors qu'ils ne le sont pas sont compabilisés comme des erreurs dans tous les cas. Colonne de droite : erreur dans la détection des occultations (en rouge les détections correctes, en jaune les faux négatifs et en cyan les faux positifs). En légende : le taux de précision et de rappel. De haut en bas : résultat obtenu à partir de 70%, 50% et 10% d'informations.

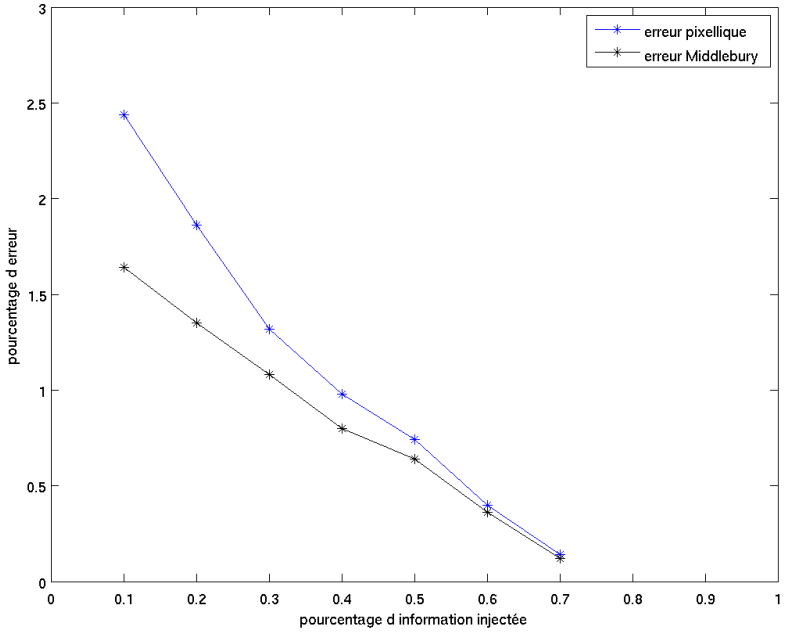


FIGURE 4.13 – Évolution des erreurs en fonction de la quantité d'information connue injectée.

4.6.2 Densification de cartes éparses

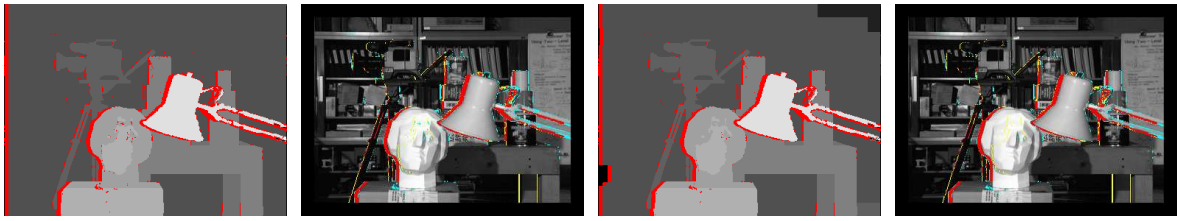
Densification sans occultation possible Pour tester cet algorithme, on utilise la vérité-terrain de la paire Tsukuba. On densifie alors les cartes éparses ainsi obtenues comme spécifié dans l'algorithme 3, en attribuant pour chaque pixel p dont la disparité $d(p)$ est connue le singleton $\{d(p)\}$ comme intervalle de disparité, et l'intervalle initial entier pour les autres pixels. On choisit ici de ne pas autoriser les pixels déjà estimés à devenir occultés.

On a choisi (figure 4.12) de laisser les paramètres estimés par défaut (qui correspondent à ceux de la paire Tsukuba). Les taux d'erreur pixellique et Middlebury en fonction de la quantité d'information conservée sont proposés dans la figure 4.13.

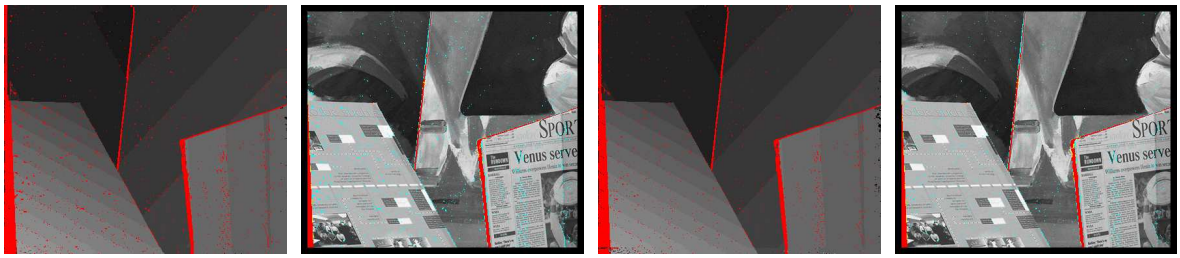
Estimation des occultations à partir d'une carte dense Dans la figure 4.14, on montre par ailleurs les résultats obtenus lorsque l'on utilise cet algorithme pour estimer les occultations à partir de la vérité-terrain dense. Pour cela, on affecte à chaque pixel p le singleton $\{d(p)\}$ comme intervalle de disparité et on les autorise à devenir occulté durant le processus. On propose les résultats obtenus avec les paramètres automatiques, puis obtenus en personnalisant les paramètres, où K est deux fois plus importants et λ deux fois plus petit.

4.6.3 Précision subpixellique

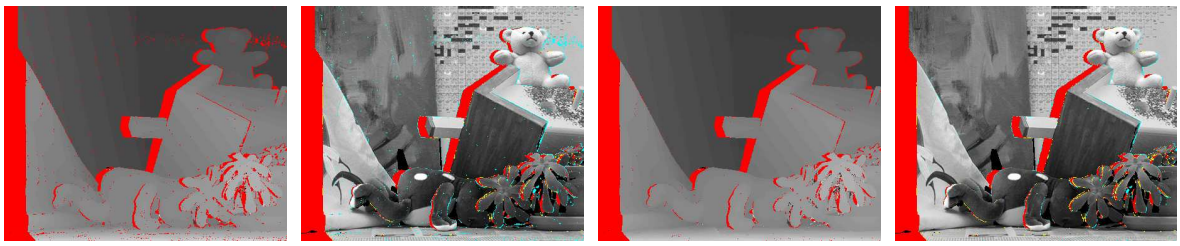
La méthode de KOLMOGOROV et ZABIH est théoriquement adaptable à la précision subpixellique, mais en pratique, il faut se ramener à un cadre pixellique. Ainsi, pour obtenir des cartes subpixelliques, on peut sur-échantillonner les images (horizontalement) d'un facteur $k = 2^n$, avec $n \in \mathbb{N}^*$, puis d'appliquer l'algorithme original avec un intervalle de disparité $\mathrm{I}_{\mathrm{disp}}^{h_k} = [\![\, k\, d_{\min}\, ;\, k\, d_{\max}\,]\!]$. Si $n = 1$, alors la précision atteinte est celle du demi-pixel. De manière générale, on obtient une précision de $1/k$ pixel. En procédant



(a) $K = 175/16$, $\lambda = 35/16$ (b) 64,53% / 78,21% (c) $K = 350/16$, $\lambda = 35/32$ (d) 69,01% / 75,56%



(e) $K = 80/7$, $\lambda = 16/7$ (f) 49,76% / 94,49% (g) $K = 160/7$, $\lambda = 8/7$ (h) 61,15% / 93,82%



(i) $K = 970/11$, $\lambda = 194/11$ (j) 87,03% / 97,03% (k) $K = 1\ 940/11$, $\lambda = 97/11$ (l) 93,88% / 95,64%

(m) $K = 261/2$, $\lambda = 261/10$ (n) 85,95% / 96,39% (o) $K = 261$, $\lambda = 261/20$ (p) 91,01% / 94,16%

FIGURE 4.14 – Estimation des occultations à partir la vérité-terrain. Les deux premières colonnes : paramètres automatiques. Les deux dernières colonnes : paramètres personnalisés. En légende : colonnes 1 et 3 : paramètres de la méthode ; colonnes 2 et 4 : taux de précision et de rappel. De haut en bas : Tsukuba, Venus, Teddy et Cones.

de la sorte, on obtient une estimation de la disparité de la scène sur-échantillonnée. Pour se ramener à l'échelle initiale, il faut identifier pour chaque pixel de l'image de référence (à l'échelle pixellique) l'ensemble des disparités associées aux pixels issus de celui-ci. En d'autres termes, si $p = (x,y)$ est un pixel de l'image de référence à l'échelle initiale, alors un sur-échantillonnage d'un facteur de $k = 2^n$ le décompose en k pixels, de coordonnées (à l'échelle subpixellique) $(x,k\,y + [\![\,0\,;\,k-1\,]\!])$. L'ensemble des disparités d^n (à cette échelle) de ces points est alors donné par $\{d^{h_k}(p') \mid p' = (x,k\,y + [\![\,0\,;\,k-1\,]\!])\}$. Pour les ramener à la scène pixellique, il suffit de les diviser par k. On choisit alors d'attribuer au pixel p la valeur médiane des disparités lorsqu'au moins la moitié des pixels p' n'est pas occultée à l'échelle subpixellique, et de déclarer p occulté sinon.

On teste tout d'abord cet algorithme sur une rampe horizontale articifielle, où la paire est générée à partir d'une image quelconque et une disparité subpixellique affine $u(x,y) = 0.01x + 1.5$ (figure 4.15). On teste ensuite sur les quatre paires de Middlebury (figures 4.16 et 4.17). On notera que les paramètres, toujours calculés automatiquement, sont différents de ceux obtenus pour la précision pixellique, puisque la paire est sur-échantillonnée. Les temps de calculs sont présentés dans le tableau 4.22 pour différentes précisions. Un comparatif des scores est donné dans le tableau 4.21.

4.6.4 Raffinement subpixellique

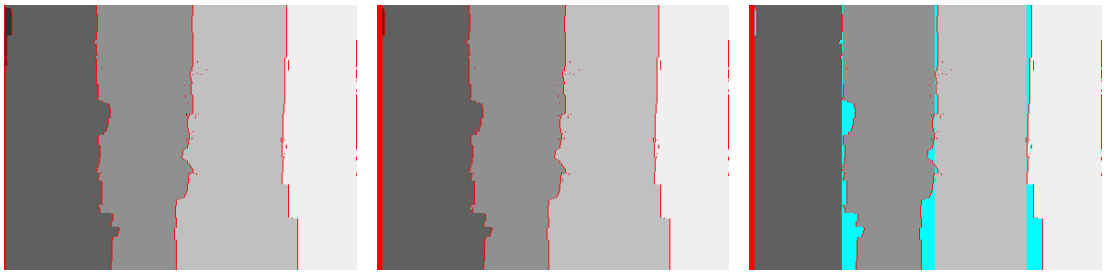
Pour le raffinement subpixellique, on reprend l'expérience précédente, dans laquelle on spécifie cette fois l'intervalle de disparité en chaque pixel. Si le pixel p est occulté, alors on affecte aux pixels associés $\{p' = (x,k\,y + [\![\,0\,;\,k-1\,]\!])\}$ l'intervalle de disparité complet $\mathrm{I}^{h_k}_{\mathrm{disp}} = [\![\,k\,d_{\min}\,;\,k\,d_{\max}\,]\!]$. Si p est non occulté, de disparité pixellique $d(p)$, alors on attribue au pixel $p' = (x,k\,y)$ situé sur la grille pixellique l'intervalle de disparité $\mathrm{I}^{h_k}_{\mathrm{disp}}(p') = k\,d(p) + [\![\,-k+1\,;\,k-1\,]\!]$, tandis que tout pixel interpolé $p' = (x,k\,y+i)$ (avec $i \in [\![\,0\,;\,k-1\,]\!]$) se voit attribuer l'union des intervalles de disparité de ces deux plus proches voisins situés sur la grille pixellique. Pour repasser à l'échelle initiale après estimation de la disparité, on procède comme dans le cas précédent. On choisit ici de tester des raffinements successifs permettant de doubler la précision à chaque étape.

On commence par tester cette procédure sur la vérité-terrain des paires Middlebury : après une quantification de ces cartes théoriques (qui donnent des cartes à valeurs entières), on les raffine à l'aide de l'algorithme décrit au paragraphe précédent, en choisissant de doubler la précision. Les résultats obtenus sont visibles à la figure 4.18.

On présente ensuite les résultats obtenus en estimant dans un premier temps la carte pixellique à l'aide de l'algorithme original, puis en raffinant peu à peu, en partant à chaque nouvelle estimation de l'estimation précédente. Les résultats obtenus à la précision du demi-pixel et du quart de pixel sont respectivement présentés dans les figures 4.19 et 4.20, tandis que les temps de calculs cumulés pour différentes précisions peuvent être consultés dans le tableau 4.22. Les scores sont par ailleurs stockés dans le tableau comparatif 4.21.

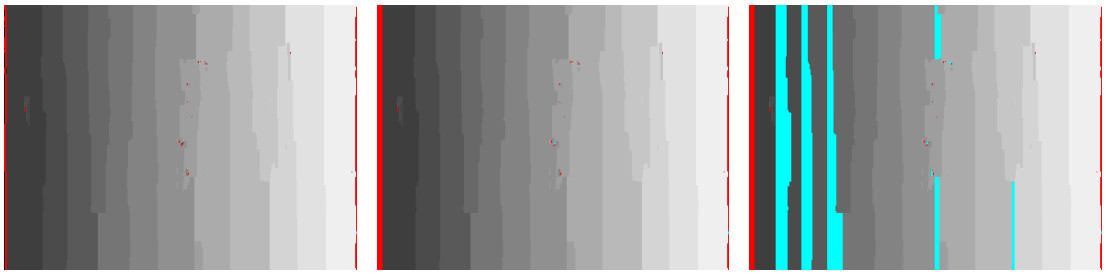
4.6.5 Discussion

Algorithme original Les performances de l'algorithme original sont très satisfaisantes, tant en termes d'estimation de la carte de disparité qu'en termes de détection de l'occultation. L'erreur pixellique (la plus pertinente pour une méthode fournissant une carte de précision pixellique) varie entre 1,42% pour la paire Venus et 17,18% pour la paire Teddy. Cette dernière paire semble particulièrement souffrir de la quantifica-



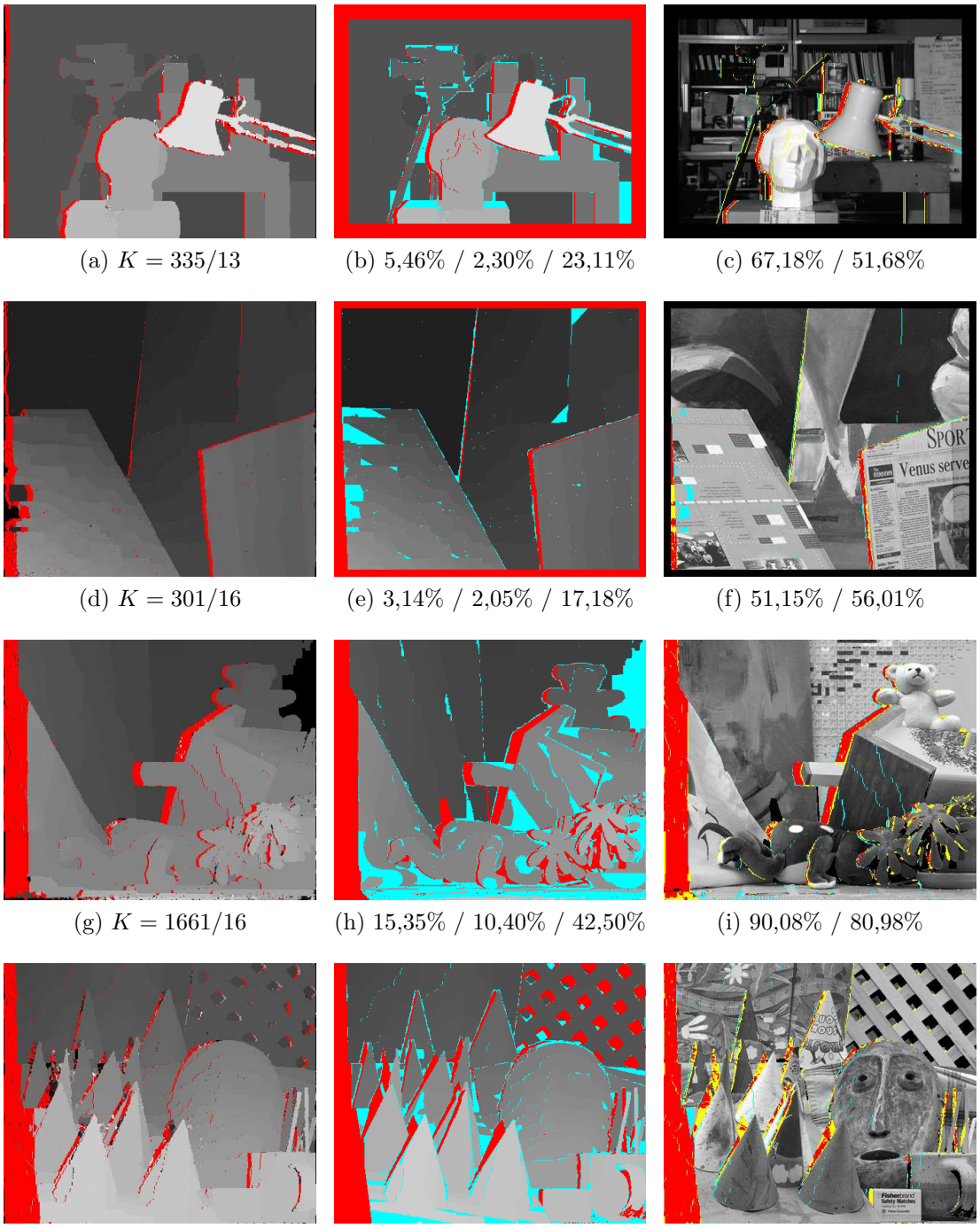
(a) $h_t = 1$ (b) 0,00% (c) 4,85%

(d) $h_t = 0{,}5$ (e) 0,00% (f) 3,44%



(g) $h_t = 0{,}25$ (h) 0,01% (i) 10,15%

FIGURE 4.15 – Précision sous-pixellique. Exemple de la rampe articifielle. La paire synthétique est construite à partir de l'image de gauche dans la paire Tsukuba et une rampe de disparité u qui permet de générer la vue de droite. À gauche : carte de disparité, avec en rouge les occultations détectées. En légende : la précision de l'estimation h_t. Au milieu : erreur d'estimation pixellique. En rouge, le masque des points dont la disparité n'est pas connue (d'après la vérité-terrain fournie par Middlebury) ou détectés occultés. En cyan, les disparités mal estimées pour l'erreur pixellique (supérieure ou égale à 1). En légende : le pourcentage d'erreur pixellique sur l'ensemble des pixels non occultés (d'après la vérité-terrain ou l'estimation KZ2). À droite : erreur subpixellique (inférieure ou égale à 0.5) avec en légende le pourcentage correspondant.



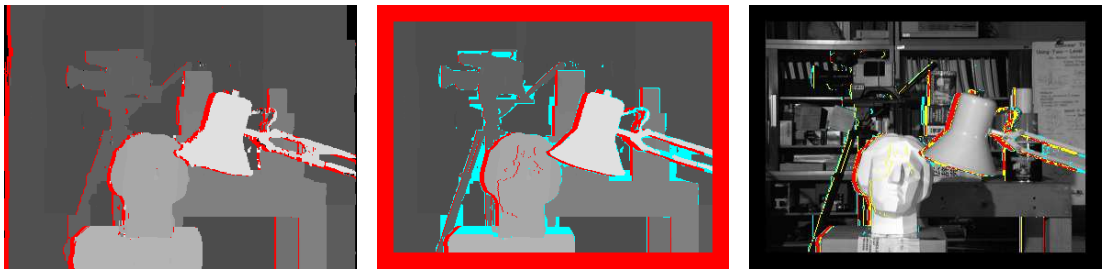
(a) $K = 335/13$ (b) 5,46% / 2,30% / 23,11% (c) 67,18% / 51,68%

(d) $K = 301/16$ (e) 3,14% / 2,05% / 17,18% (f) 51,15% / 56,01%

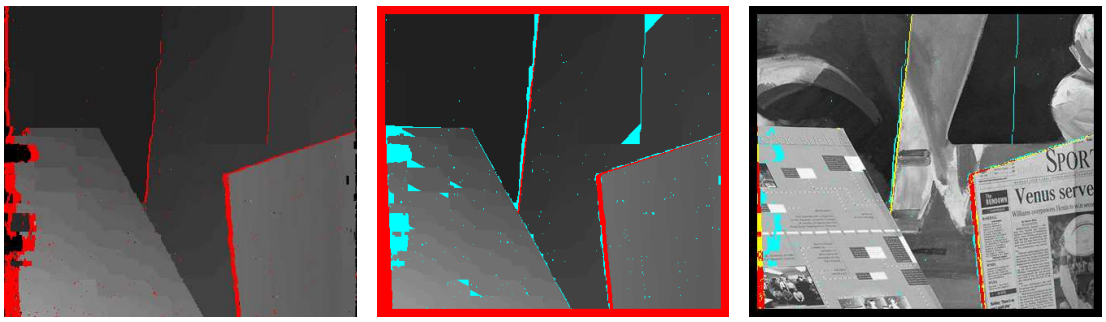
(g) $K = 1661/16$ (h) 15,35% / 10,40% / 42,50% (i) 90,08% / 80,98%

(j) $K = 1420/9$ (k) 8,52% / 6,33% / 29,33% (l) 86,70% / 72,80%

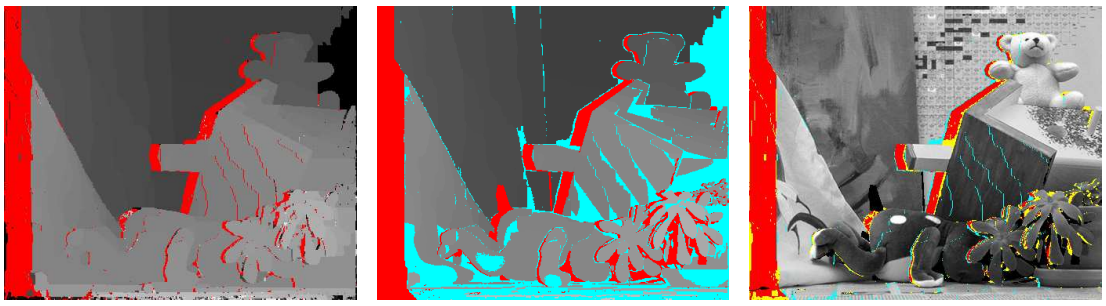
FIGURE 4.16 – **Précision sous-pixellique** ($h_t = 0{,}5$). Colonne de gauche : carte de disparité, avec en rouge les occultations détectées. En légende : les paramètres estimés automatiquement. Colonne du milieu : erreur d'estimation. En rouge, le masque des points dont la disparité n'est pas connue (d'après la vérité-terrain fournie par Middlebury). En cyan, les disparités mal estimées pour l'erreur pixellique (supérieure ou égale à 1). En légende : le pourcentage d'erreur pixellique / erreur Middlebury (strictement supérieure à 1) / erreur sous-pixellique (supérieure ou égale à 0,5) dans les zones occultées (d'après la vérité-terrain). Les pixels déclarés occultés par KZ2 alors qu'ils ne le sont pas sont compabilisés comme des erreurs dans tous les cas. Colonne de droite : erreur dans la détection des occultations (en rouge les détections correctes, en jaune les faux négatifs et en cyan les faux positifs). En légende : le taux de précision et de rappel. De haut en bas : Tsukuba, Venus, Teddy et Cones.



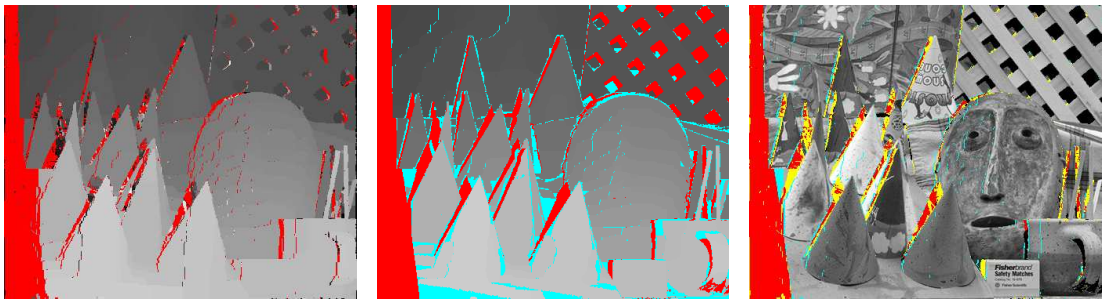
(a) $K = 425/11$ (b) 4,93% / 2,43% / 13,64% (c) 61,64% / 53,63%



(d) $K = 336/15$ (e) 3,42% / 2,71% / 15,56% (f) 42,77% / 55,98%

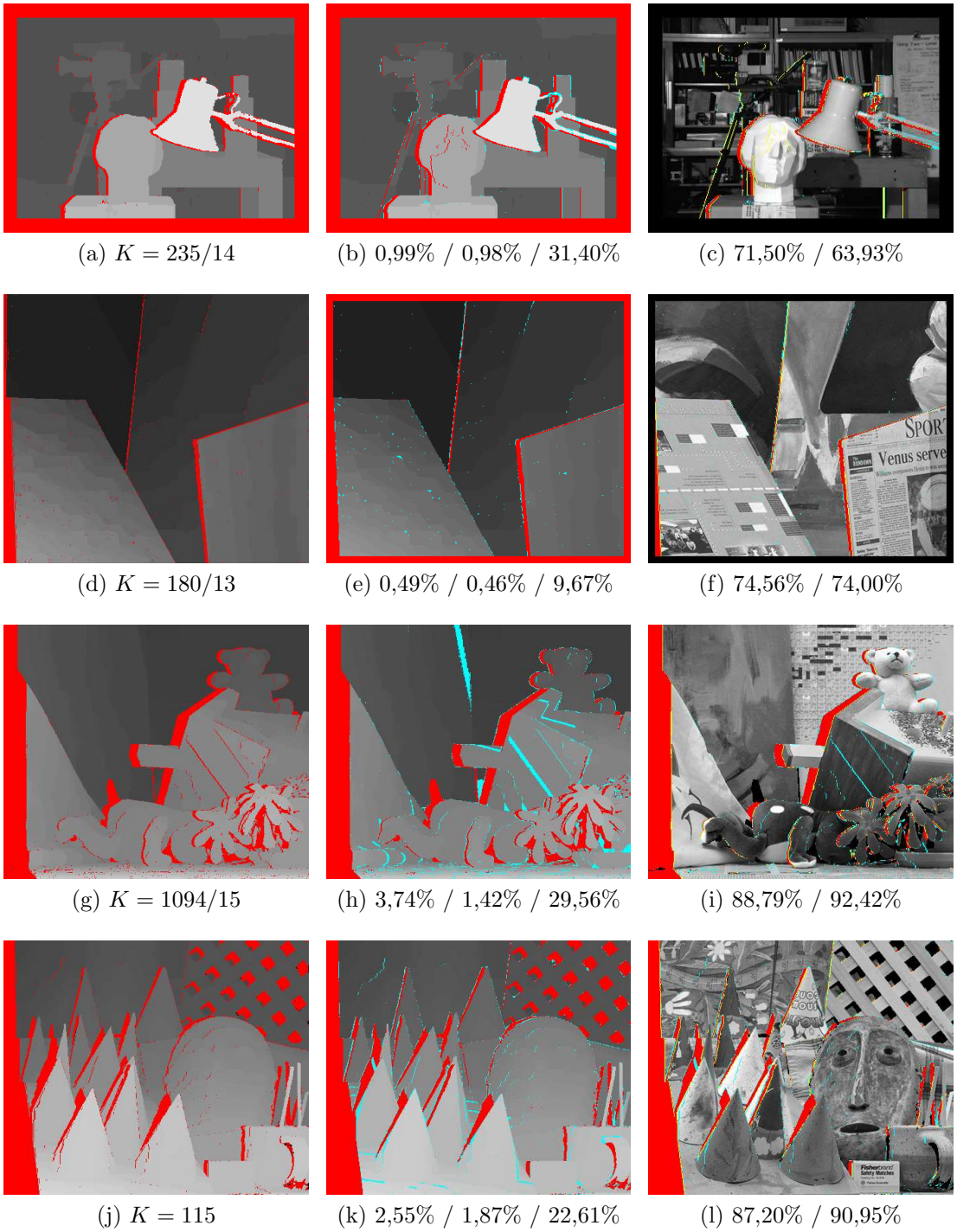


(g) $K = 1105/10$ (h) 17,32% / 11,44% / 41,61% (i) 88,22% / 81,50%



(j) $K = 2719/16$ (k) 8,19% / 25,56% / 2,03% (l) 83,60% / 73,72%

FIGURE 4.17 – **Précision sous-pixellique** ($h_t = 0{,}25$). Colonne de gauche : carte de disparité, avec en rouge les occultations détectées. En légende : les paramètres estimés automatiquement. Colonne du milieu : erreur d'estimation. En rouge, le masque des points dont la disparité n'est pas connue (d'après la vérité-terrain fournie par Middlebury). En cyan, les disparités mal estimées pour l'erreur pixellique (supérieure ou égale à 1). En légende : le pourcentage d'erreur pixellique / erreur Middlebury (strictement supérieure à 1) / erreur sous-pixellique (supérieure ou égale à 0,5) dans les zones occultées (d'après la vérité-terrain). Les pixels déclarés occultés par KZ2 alors qu'ils ne le sont pas sont compabilisés comme des erreurs dans tous les cas. Colonne de droite : erreur dans la détection des occultations (en rouge les détections correctes, en jaune les faux négatifs et en cyan les faux positifs). En légende : le taux de précision et de rappel. De haut en bas : Tsukuba, Venus, Teddy et Cones.



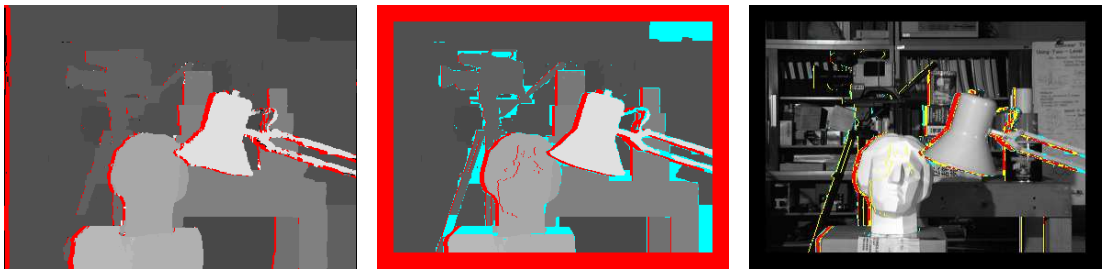
(a) $K = 235/14$ (b) 0,99% / 0,98% / 31,40% (c) 71,50% / 63,93%

(d) $K = 180/13$ (e) 0,49% / 0,46% / 9,67% (f) 74,56% / 74,00%

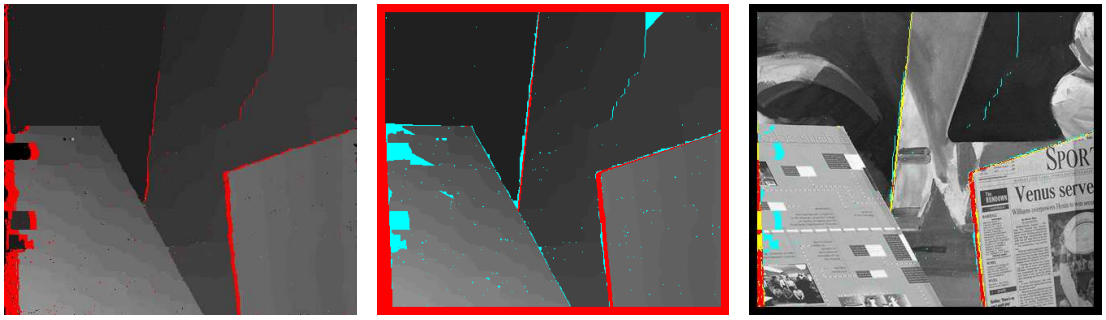
(g) $K = 1094/15$ (h) 3,74% / 1,42% / 29,56% (i) 88,79% / 92,42%

(j) $K = 115$ (k) 2,55% / 1,87% / 22,61% (l) 87,20% / 90,95%

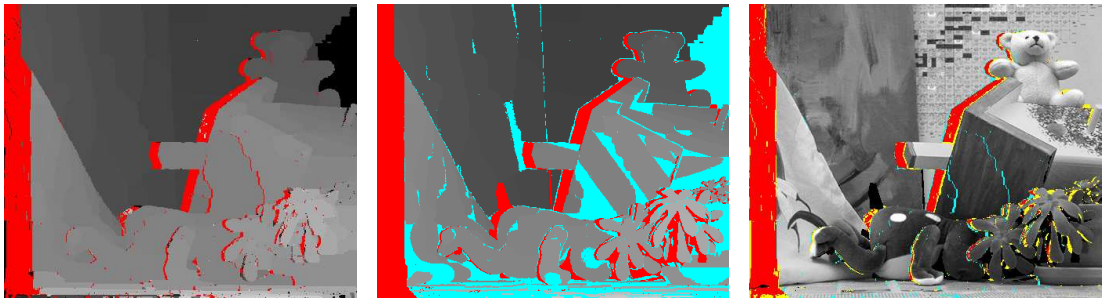
FIGURE 4.18 – **Raffinement sous-pixellique ($h_t = 0{,}5$) à partir de la vérité-terrain quantifiée**. Colonne de gauche : carte de disparité, avec en rouge les occultations détectées. Colonne de gauche : carte de disparité, avec en rouge les occultations détectées. En légende : les paramètres estimés automatiquement. Colonne du milieu : erreur d'estimation. En rouge, le masque des points dont la disparité n'est pas connue (d'après la vérité-terrain fournie par Middlebury). En cyan, les disparités mal estimées pour l'erreur pixellique (supérieure ou égale à 1). En légende : le pourcentage d'erreur pixellique / erreur Middlebury (strictement supérieure à 1) / erreur sous-pixellique (supérieure ou égale à 0,5) dans les zones occultées (d'après la vérité-terrain). Les pixels déclarés occultés par KZ2 alors qu'ils ne le sont pas sont compabilisés comme des erreurs dans tous les cas. Colonne de droite : erreur dans la détection des occultations (en rouge les détections correctes, en jaune les faux négatifs et en cyan les faux positifs). En légende : le taux de précision et de rappel. De haut en bas : Tsukuba, Venus, Teddy et Cones.



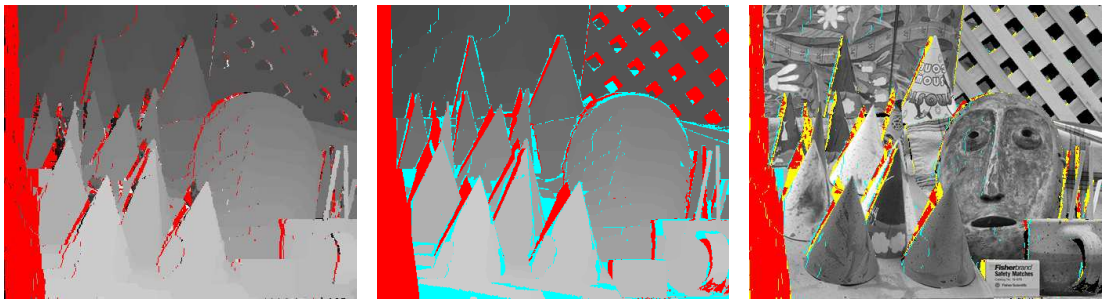
(a) $K = 235/14$ (b) 7,03% / 2,31% / 23,66% (c) 68,69% / 51,78%



(d) $K = 180/13$ (e) 1,61% / 14,43% / 0,09% (f) 50,12% / 56,13%

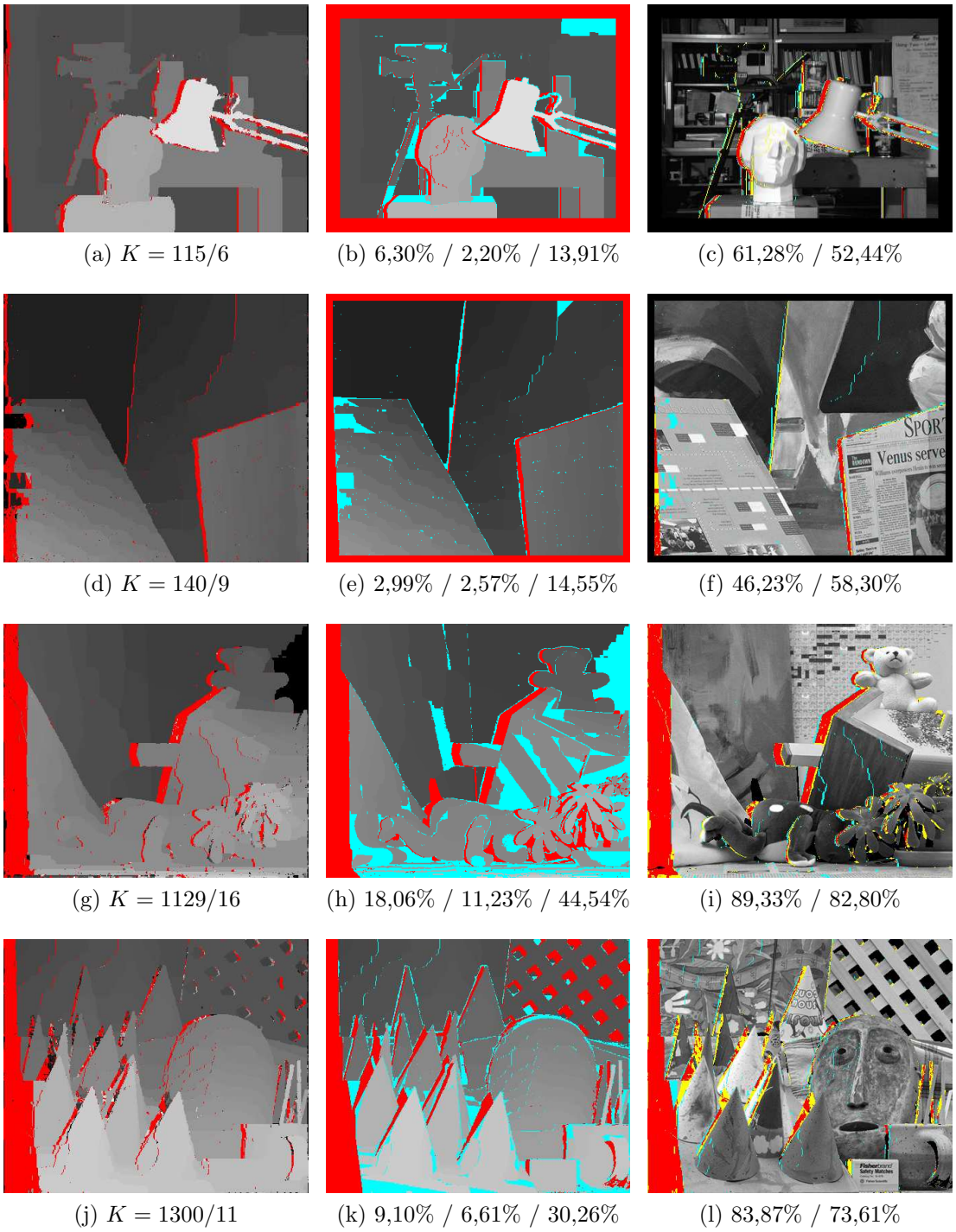


(g) $K = 1094/15$ (h) 17,15% / 10,73% / 43,98% (i) 91,01% / 81,63%



(j) $K = 115$ (k) 8,19% / 5,78% / 31,48% (l) 86,32% / 72,36%

FIGURE 4.19 – **Raffinement sous-pixellique** ($h_t = 0{,}5$). Colonne de gauche : carte de disparité, avec en rouge les occultations détectées. En légende : les paramètres estimés automatiquement. Colonne du milieu : erreur d'estimation. En rouge, le masque des points dont la disparité n'est pas connue (d'après la vérité-terrain fournie par Middlebury). En cyan, les disparités mal estimées pour l'erreur pixellique (supérieure ou égale à 1). En légende : le pourcentage d'erreur pixellique / erreur Middlebury (strictement supérieure à 1) / erreur sous-pixellique (supérieure ou égale à 0,5) dans les zones occultées (d'après la vérité-terrain). Les pixels déclarés occultés par KZ2 alors qu'ils ne le sont pas sont compabilisés comme des erreurs dans tous les cas. Colonne de droite : erreur dans la détection des occultations (en rouge les détections correctes, en jaune les faux négatifs et en cyan les faux positifs). En légende : le taux de précision et de rappel. De haut en bas : Tsukuba, Venus, Teddy et Cones.



(a) $K = 115/6$ (b) 6,30% / 2,20% / 13,91% (c) 61,28% / 52,44%

(d) $K = 140/9$ (e) 2,99% / 2,57% / 14,55% (f) 46,23% / 58,30%

(g) $K = 1129/16$ (h) 18,06% / 11,23% / 44,54% (i) 89,33% / 82,80%

(j) $K = 1300/11$ (k) 9,10% / 6,61% / 30,26% (l) 83,87% / 73,61%

FIGURE 4.20 – **Raffinement sous-pixellique** ($h_t = 0{,}25$). Colonne de gauche : carte de disparité, avec en rouge les occultations détectées. En légende : les paramètres estimés automatiquement. Colonne du milieu : erreur d'estimation. En rouge, le masque des points dont la disparité n'est pas connue (d'après la vérité-terrain fournie par Middlebury). En cyan, les disparités mal estimées pour l'erreur pixellique (supérieure ou égale à 1). En légende : le pourcentage d'erreur pixellique / erreur Middlebury (strictement supérieure à 1) / erreur sous-pixellique (supérieure ou égale à 0,5) dans les zones occultées (d'après la vérité-terrain). Les pixels déclarés occultés par KZ2 alors qu'ils ne le sont pas sont compabilisés comme des erreurs dans tous les cas. Colonne de droite : erreur dans la détection des occultations (en rouge les détections correctes, en jaune les faux négatifs et en cyan les faux positifs). En légende : le taux de précision et de rappel. De haut en bas : Tsukuba, Venus, Teddy et Cones.

Paire	Tsukuba	Venus	Teddy	Cones
Estimation pixellique ($h_t = 1$)	8,20%	3,23%	18,27%	9,63%
Estimation sous-pixellique ($h_t = 0{,}5$)	5,46%	3,14%	15,35%	8,52%
Raffinement sous-pixellique ($h_t = 0{,}5$)	7,03%	2,63%	17,15%	8,19%
Estimation sous-pixellique ($h_t = 0{,}25$)	4,93%	3,42%	17,32%	8,74%
Raffinement sous-pixellique ($h_t = 0{,}25$)	6,30%	2,99%	18,06%	7,30%

(a) Erreur pixellique

Paire	Tsukuba	Venus	Teddy	Cones
Estimation pixellique ($h_t = 1$)	2,71%	2,84%	11,20%	6,55%
Estimation sous-pixellique ($h_t = 0{,}5$)	2,30%	2,05%	10,40%	6,33%
Raffinement sous-pixellique ($h_t = 0{,}5$)	2,31%	2,23%	10,73%	5,78%
Estimation sous-pixellique ($h_t = 0{,}25$)	2,43%	2,71%	11,44%	6,36%
Raffinement sous-pixellique ($h_t = 0{,}25$)	2,20%	2,57%	11,23%	6,61%

(b) Erreur Middlebury

Paire	Tsukuba	Venus	Teddy	Cones
Estimation pixellique ($h_t = 1$)	8,20%	22,15%	50,72%	35,44%
Estimation sous-pixellique ($h_t = 0{,}5$)	23,11%	17,18%	42,50%	29,33%
Raffinement sous-pixellique ($h_t = 0{,}5$)	23,66%	15,34%	43,98%	31,48%
Estimation sous-pixellique ($h_t = 0{,}25$)	13,64%	15,56%	41,61%	27,05%
Raffinement sous-pixellique ($h_t = 0{,}25$)	13,91%	14,55%	44,54%	31,48%

(c) Erreur sous-pixellique

FIGURE 4.21 – Erreurs d'estimation pour les trois méthodes testées (estimation pixellique, estimations sous-pixelliques et raffinement sous-pixellique). Les raffinements sous-pixelliques s'entendent respectivement comme le raffinement au demi-pixel de la carte pixellique et le raffinement au quart de pixel de ce premier raffinement. Les scores en rouge correspondent pour chaque erreur et chaque paire aux scores les plus faibles.

Paire	Tsukuba	Venus	Teddy	Cones
Taille du volume de coût (en mégapixels)	3,6	6,6	20,2	20,2
Temps d'exécution (en secondes) pour la méthode sous-pixellique	20	59	139	151
Temps d'exécution (en secondes) pour le raffinement sous-pixellique	8	23	74	75

(a) $h_t = 0{,}5$

Paire	Tsukuba	Venus	Teddy	Cones
Taille du volume de coût (en mégapixels)	7,2	13,2	40,4	40,4
Temps d'exécution (en secondes) pour la méthode sous-pixellique	104	271	711	633
Temps d'exécution (en secondes) pour le raffinement sous-pixellique	42	111	285	292

(b) $h_t = 0{,}25$

FIGURE 4.22 – Temps d'exécution pour chaque niveau de précision : (a) demi-pixel, (b) quart de pixel. La taille du voulume de coût est donnée par $N_x \times N_y \times N_t/h_t$, où $N_X \times N_y$ est la taille de l'image de référence et N_t la taille de l'intervalle de disparité. Pour le raffinement, il s'agit du temps cumulé (on comptabilise le temps de calculs de chaque échelle).

tion de la disparité, en particulier au niveau de la maison. Il s'agit en effet de surfaces dont la disparité varie de manière subpixellique, et la valeur pixellique retenue par l'algorithme ne coïncide pas toujours avec la disparité pixellique la plus proche de la disparité réelle. Notons que la paire Tsukuba est la seule qui possède une vérité-terrain pixellique, ce qui explique que, pour une estimation pixellique de la disparité, l'erreur pixellique et l'erreur subpixellique soient égales.

La détection des occultations est également très bonne, puisqu'on atteint un taux de 82,25% pour la précision et de 85,37% pour le rappel avec Teddy (ces scores seront améliorés par la suite). Il faut toutefois souligner que les occultations peuvent très isolées (un pixel), comme dans la paire Venus, ce qui ne correspond pas à une réalité pertinente (voir l'analyse sur l'occultation réalisée dans le chapitre 2). Enfin, on notera la grande variabilité des valeurs du paramètre K, qui est compris entre 10 et 130,5. Vu la sensibilité de l'algorithme à la valeur de ses paramètres, il semble illusoire d'espérer obtenir des résultats globalement corrects en fixant une valeur unique pour K.

L'algorithme est très rapide, puisqu'il fournit des résultats en quelques secondes. Cela est principalement dû à l'efficacité de l'algorithme de FORD-FULKERSON utilisé pour calculer la coupure minimale. Pour un nombre d'itérations donné, la complexité est de l'ordre du nombre de sommets et d'arcs construits dans le graphe. Or, on construit au plus 5 arcs par sommets, et le nombre de sommets est majoré par le double du nombre de pixels (on ne construit au plus que deux sommets par pixel). Le nombre d'α-*expansion moves* par itération est donné par le nombre de disparité dans l'intervalle de disparité. on en déduit que la complexité est proportionnelle à la taille du volume de coût, ce que l'on vérifie dans le tableau 4.4. On voit par ailleurs dans la figure 4.5 qu'en augmentant le nombre d'itérations maximal autorisé, les résultats varient peu qualitativement. Seule la paire Venus semble en bénéficier (un des trous sur la gauche est comblé). Le choix de ce nombre semble donc justifié.

Lorsqu'on modifie la valeur de K, en conservant les dépendances des autres paramètres (figure 4.6), on constate qu'effectivement, plus K est grand, moins la carte

présente d'occultation et plus elle est régulière. Cela est visible en particulier dans le fond de la scène. En revanche, cette sur-régularisation a pour effet d'introduire des artéfacts sur le bord droit de la lampe, en introduisant une mauvaise estimation des contours de boîtes sur la table. Dans l'exemple (d), pour la valeur de K vaut approximativement 33 (et est donc environ trois fois plus grand que par défaut), tout le fond de la scène est bien estimé (sur le bord droit des pieds de la table en particulier), mais les deux barres dans le bras de la lampe fusionnent. Au contraire, si K prend une valeur plus faible, alors l'occultation tout comme les sauts de disparité sont plus présentes. On remarque en particulier l'apparition d'occultations isolées au dessus de la lampe et à gauche de la caméra. Si on modifie la valeur de λ par rapport à celle de K (figure 4.7), on change le poids relatif du terme d'attache aux données + occultation et du terme de régularité. Si λ est choisi plus faible (égal à $K/10$ par exemple), alors il devient moins coûteux d'introduire des discontinuités ; au contraire, s'il est plus grand ($\lambda = K/2.5$ par exemple), alors la régularité est privilégiée, quitte à introduire davantage d'occultation. Une fois encore, l'effet le plus visible de ce changement de régularité s'observe sur le fond de la scène (coin en haut à droite). L'effet du rapport entre λ_1 et λ_2 est plus subtil (figure 4.8). Lorsque λ_1 est choisi grand ($\lambda_1 = 6\lambda$), alors il devient encore plus coûteux d'introduire une discontinuité de disparité qui ne coïncide pas avec une discontinuité de couleur. L'algorithme a donc tendance à aligner les discontinuités de couleur et de disparité, ce que l'on voit de manière très nette dans le coin en haut à gauche de la scène (à comparer avec l'image de référence). La discontinuité que l'on observe suit exactement un contour présent dans l'image Enfin, la modification du seuil qui détermine si le coût de régularité est λ_1 ou λ_2 (figure 4.9) ne semble pas avoir d'effet très visible sur les résultats.

On voit ainsi que le paramètre le plus critique est le coût d'occultation K. Kolmogorov et Zabih ont proposé une heuristique qui semble fournir une valeur raisonnable pour ce paramètre, mais il est utile de signaler que ce calcul, qui dépend d'un pourcentage sur les assignements, est en réalité très sensible au choix de l'intervalle de disparité. Si celui-ci est mal estimé, par exemple s'il est beaucoup plus grand que sa véritable valeur, alors la valeur de K augmente (puisqu'on ajoute des corrélations possibles).

Densification Dans la figure 4.13, on observe que, de manière attendue, les pourcentages d'erreurs décroissent à mesure que la quantité d'information (c'est-à-dire de disparité connue) augmente dans les cas considérés. En particulier, introduire 10% d'informations permet de passer d'une erreur pixellique de 8,20% à 2,44%, soit une erreur divisée par trois.

La détection des occultations dans les vérités-terrains donne dans l'ensemble de bons taux de rappel, allant jusqu'à 97,03%. Le taux de précision est moins bon lorsque les paramètres sont choisis automatiquement, car on observe dans ce cas de nombreuses détections isolées (1 pixel). En réglant manuellement ces paramètres, on parvient à les éliminer en partie, mais en perdant quelques détections initialement correctes (comme le montre la baisse du taux de rappel dans la seconde expérience). Ainsi, si la disparité est correctement estimée, la méthode KZ2 fournit un bon détecteur d'occultation, à condition d'en modifier les paramètres.

Précision subpixellique L'estimation pixellique de la disparité réalisée à partir des paires sur-échantillonnées montre des résultats variables. Pour le cas de la rampe (figure 4.15), on observe ainsi que le résultat est visuellement plus satisfaisant à mesure que l'on augmente la précision, car la scène présente une disparité subpixellique. La

quantification introduite par la méthode KZ2 introduit dans ce cas des artéfacts gênants pour l'interprétation visuelle. Cependant, on observe dès cette expérience que l'erreur d'estimation calculée à partir de la vérité-terrain ne reflète pas toujours le gain apporté par l'augmentation de la précision. Ainsi, la carte précise au quart de pixel est visuellement plus proche de la rampe affine, mais présente des erreurs plus importantes (+6,71% pour l'erreur subpixellique dans le pire des cas). Notons que l'erreur pixellique est la même pour chaque précision.

Sur des scènes plus complexes, le gain apporté par la précision subpixellique dépend de la scène et de l'erreur mesurée. À part pour Tsukuba, on observe une amélioration des résultats avec la précision subpixellique (qu'elle soit directe ou obtenue par raffinement). Ce comportement peut être expliqué par le fait qu'il s'agit de la seule carte pixellique des quatre paires considérées, alors que, visiblement, la scène n'est pas constante par morceaux (on pourra considérer le cas de la lampe par exemple). Aussi, il est raisonnable de penser que, dans ce cas précis, la fiabilité de la précision subpixellique de la vérité-terrain est à mettre en cause. L'erreur subpixellique la plus faible est par ailleurs obtenue – pour les trois paires à vérité-terrain subpixellique – est obtenue avec la précision du quart de pixel, avec un gain maximal de 9,11 atteint pour la paire Teddy.

Considérons ensuite le raffinement pixellique. On commence par analyser le raffinement réalisé à partir d'une version quantifiée de la vérité-terrain des paires de Middlebury. Commençons par noter que l'erreur pixellique n'est pas systématiquement nulle, et ce, bien que la carte raffinée ne diffère au plus que d'un demi-pixel de la carte initiale quantifiée. Cela est dû au fait que l'écart de la disparité raffinée avec la vérité-terrain subpixellique peut être égale à 1 tout en conduisant à un écart à la version quantifiée égal à 0,5. En effet, si la disparité correcte (subpixellique) vaut 1,5 par exemple, sa quantification conduit à la valeur 1 (suivant la convention choisie) et une erreur d'un demi-pixel dans l'étape de raffinement peut aboutir à la valeur 0,5. D'autres erreurs pixelliques, observées au niveau des discontinuité de la scène, s'expliquent quant à elle par la convention choisie pour passer d'une carte sur-échantillonnée à une carte à l'échelle initiale et par le choix des intervalles de disparité adaptatifs. Les pixels situés dans de telles régions sont associés dans la scène sur-échantillonnée à des pixels interpolés dont l'intervalle de disparité est plus grand que celui du pixel considéré (car son voisin possédant une disparité pixellique différente, l'union de leurs deux intervalles est nécessairement plus large). Ainsi, l'erreur commise peut excéder le pixel. Néanmoins, on observe que l'erreur Middlebury est nulle partout sauf pour la paire Cones, ce qui s'explique par le fait que, pour cette erreur, seules discontinuités de la scène peuvent être impliquées (l'erreur de quantification étant inférieure ou égale à 1). Le raffinement des vérité-terrains montrent, dans trois cas sur quatre, un gain pour l'erreur subpixellique. À nouveau, la seule paire qui échoue à ce test est la paire Tsukuba. L'erreur double entre l'estimation pixellique (par l'algorithme original) et le raffinement, pourtant réalisé à partir d'une information fiable.

Intéressons-nous enfin à l'expérience de raffinement, réalisée à partir de l'estimation pixellique. Les résultats sont naturellement moins bons que ceux obtenus à partir de la vérité-terrain quantifiée, car le raffinement dépend fortement de l'initialisation. En effet, si une erreur est commise à une échelle, alors elle ne peut être corrigée aux échelles suivantes, à moins que le pixel ne soit déclaré occulté. Néanmoins, pour certaines paires, l'erreur pixellique reste proche (voire plus fiable) que celles obtenues en estimant directement la disparité subpixellique. Pour terminer, comparons en termes d'efficacité l'estimation subpixellique et le raffinement. Le gain apporté par la stratégie

de raffinement est nettement visible dans les tableaux de la figure 4.22. Le temps de calcul est ainsi divisé (en moyenne) par 2,3.

Conclusion

La méthode originale de Kolmogorov et Zabih reste l'une des rares méthodes à produire des détections satisfaisantes de cartes d'occultations, tout en étant d'une complexité algorithmique raisonnable. Cette performance repose sur deux particularités de la méthode. D'une part, la fonctionnelle d'énergie n'est pas minimisée exactement, mais décroît selon une règle de descente (les *expansion moves*), dont chaque pas est choisi de manière optimale. D'autre part, l'optimisation du pas de descente repose entièrement sur la représentabilité par un graphe de l'énergie sous-jacente, qui découle du choix particulier de la fonctionnelle initiale. La résolution se fait donc par coupure de graphes, dont les algorithmes sont connus pour être particulièrement efficaces grâce à la dualité avec le problème de recherche de flot maximal. Malgré les contraintes liées à la représentabilité des énergies considérées, l'algorithme reste suffisamment souple pour permettre des modifications importantes. Nous avons ainsi pu spécifier l'intervalle de disparité de chaque pixel. Cette possibilité permet d'exploiter l'efficacite des coupures de graphes au profit d'autres problèmes, tels que la densification de cartes éparses ou le raffinement subpixellique de cartes pixelliques. Une telle approche peut en particulier améliorer les résultats obtenus avec des méthodes plus fiables (pour produire des cartes éparses ou pixelliques) mais incapables de densifier leurs résultats ou trop coûteux en termes de calcul pour envisager une précision subpixellique.

Néanmoins, la fonctionnelle d'énergie considérée dans cette méthode ne repose sur aucune modélisation explicite du phénomène d'occultation. Celle-ci n'est considérée que comme une alternative moins coûteuse que la mise en correspondance. Or, le chapitre précédent a démontré que l'occultation obéissait à des règles précises. Celles-ci ne sont partiellement que prises en compte (et de manière implicite) dans le terme d'injectivité. Ainsi, bien qu'expérimentalement, la détection d'occultation reste satisfaisante, elle introduit parfois des occultations isolées (de la taille d'un pixel) qui sont tout à fait artificielles et ne correspondent pas à de véritables occultations. Par ailleurs, ce terme repose sur le choix d'un paramètre qui n'est pour l'instant pas rigoureusement justifié et ni compris. Lorsque l'on utilise la méthode comme simple détecteur d'occultations, on observe en particulier que la valeur proposée par défaut de ce paramètre n'est visblement plus adaptée au problème.

Références

[1] Stan Birchfield and Carlo Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4) :401–406, 1998.

[2] Adrian Bondy and Uppaluri SR Murty. *Graph Theory (Graduate Texts in Mathematics 244)*. Springer, 2008.

[3] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of Min-Cut/Max-Flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9) :1124–1137, 2004.

[4] Lester R. Ford and Delbert R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8(3) :399–404, 1956.

[5] Lester R. Ford and Delbert R. Fulkerson. *Flows in networks*. Princeton University Press, 2015.

[6] Vladimir Kolmogorov. *Graph based algorithms for scene reconstruction from two or more views*. PhD thesis, Cornell University, 2004.

[7] Vladimir Kolmogorov, Pascal Monasse, and Pauline Tan. Kolmogorov and Zabih's graph cuts stereo matching algorithm. *Image Processing On Line*, 4 :220–251, 2014.

[8] Vladimir Kolmogorov and Ramin Zabih. Computing visual correspondence with occlusions using graph cuts. In *IEEE International Conference on Computer Vision*, volume 2, pages 508–515. IEEE, 2001.

[9] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts ? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2) :147–159, 2004.

[10] Christoph Rhemann, Asmaa Hosni, Michael Bleyer, Carsten Rother, and Margrit Gelautz. Fast cost-volume filtering for visual correspondence and beyond. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3017–3024. IEEE, 2011.

[11] Neus Sabater. *Fiabilité et précision en stéréoscopie : application à l'imagerie aérienne et satellitaire à haute résolution*. PhD thesis, École normale supérieure de Cachan, 2009.

Troisième partie

Accélération d'algorithmes d'optimisation convexe

Chapitre 5

Convergence d'algorithmes primaux-duaux : application à l'ADMM

Introduction

La méthode des directions alternées (en anglais ADMM pour *Alternating Direction Method of Multipliers*) a été initialement proposée en 1976 par Gabay et Mercier [9] et en 1975 par Glowinski et Marrocco [10]. C'est l'une des nombreuses méthodes [7] proposées pour résoudre les problèmes composites de la forme

$$\min_{x\in X}\Big\{G(x)+F(Kx)\Big\}.$$

Cette méthode a fait l'objet d'une étude intensive ces dernières années, en particulier à propos de sa convergence. Les résultats de convergence proposés se limitent généralement à des hypothèses restrictives sur le rang de la matrice K ainsi que sur la régularité des fonctions F et G. L'ADMM est par ailleurs reliée à d'autres méthodes classiquement rencontrées en optimisation convexe, comme les méthodes Douglas-Rachford ou Peaceman-Rachford, ainsi qu'on le verra au paragraphe 5.1.3, ce qui a permis en particulier d'en proposer de nombreuses variantes.

Dans ce chapitre, on commencera par proposer une courte revue de la méthode des directions alternées classique, ainsi que ses liens avec d'autres méthodes, et certaines des variantes récemment proposées dans la littérature (section 5.1). On montrera ensuite dans la section 5.2 que l'algorithme classique peut se réécrire comme un algorithme primal-dual proximal. On propose alors à une preuve de la convergence de ce dernier dans le cas régulier, d'où on peut dériver un résultat de convergence sur la méthode des directions alternées. Enfin, la section 5.3 présente une variante simple de l'ADMM dont on démontre qu'elle présente un meilleur taux de convergence théorique.

5.1 La méthode des directions alternées

On présente dans cette section le problème étudié dans ce chapitre et on introduit la méthode des multiplicateurs, dont on rappelle quelques résultats connus. Une courte revue d'algorithmes existants liés à cette méthode est enfin présentée.

5.1.1 Position du problème

Problème composite Nous étudions dans ce chapitre les problèmes composites de la forme

$$\min_{x\in X}\Big\{G(x)+F(Kx)\Big\}. \tag{5.1}$$

Pour l'instant, on ne fait aucune hypothèse particulière sur les fonctions convexes et propres $G : X \to \mathbb{R}\cup\{+\infty\}$ et $F : Z \to \mathbb{R}\cup\{+\infty\}$, ni sur l'opérateur linéaire continu $K : X \to Z$ (de norme finie), si ce n'est l'existence d'une solution au problème (5.1). Dans ce qui suit, X et Z désignent des espaces de Hilbert de dimensions finies. Le problème sans contrainte (5.1) peut également s'écrire sous la forme d'un problème sous contrainte linéaire :

$$\min_{\substack{(x,z)\in X\times Z\\ Kx=z}}\Big\{G(x)+F(z)\Big\}. \tag{5.2}$$

Cette écriture permet en outre de rendre le lagrangien du problème séparable en x et z, le couplage se faisant dans la contrainte linéaire.

De nombreuses méthodes ont été proposées pour résoudre ce problème. Elles reposent généralement sur une stratégie d'éclatement qui vise à découpler les deux termes de la fonction, puis elles exploitent des hypothèses de régularité sur les fonctions ou de rang sur l'opérateur. On s'intéressera dans cette section à l'ADMM (et ses variantes), tandis qu'une autre méthode sera présentée dans la section suivante.

Lagrangien augmenté Pour résoudre le problème (5.2), la méthode ADM propose de considérer le lagrangien augmenté, défini pour tout $(x,y,z) \in X \times Z \times Z$ par

$$L_\lambda(x,y,z) = G(x) + F(z) + \langle y, Kx - z\rangle + \frac{1}{2\lambda}\|Kx - z\|^2$$

où le paramètre λ est supposé strictement positif. Le lagrangien augmenté possède les mêmes points selles que le lagrangien

$$L(x,y,z) = G(x) + F(z) + \langle y, K\,x - z\rangle$$

où y est un multiplicateur de Lagrange permettant de tenir compte de la contrainte égalité $Kx = y$. L'ajout du terme quadratique permet d'accélérer la recherche de ces derniers. On peut ainsi s'intéresser au problème primal-dual

$$\min_{(x,z)\in X\times Z} \sup_{y\in Z} L_\lambda(x,y,z) \tag{5.3}$$

dont une solution (x^*,y^*,z^*) fournit une solution x^* au problème (5.1) (avec $Kx^* = z^*$).

5.1.2 L'algorithme ADMM

La méthode des multiplicateurs Pour résoudre le problème de recherche de point-selle (5.3), une première méthode, appelée *méthode des multiplicateurs*, a été proposée. Elle consiste à alterner une minimisation partielle du lagrangien augmenté en les variables primales (x,z) et une étape de montée de gradient en la variable duale y. Cela conduit à l'algorithme suivant : on initialise avec $y_0 \in Z^*$, puis on effectue pour tout $n \in \mathbb{N}$ les mises-à-jours suivantes

$$\left\{\begin{array}{rcl} (x_{n+1},z_{n+1}) & = & \displaystyle\operatorname*{argmin}_{(x,z)\in X\times Z} L_\lambda(x,y_n,z) \\ \\ y_{n+1} & = & \displaystyle y_n + \sigma\,\frac{\partial L_\lambda}{\partial y}(x_{n+1},y_n,z_{n+1}) \end{array}\right.$$

qui s'écrit encore

$$\left\{\begin{array}{rcl} (x_{n+1},z_{n+1}) & = & \displaystyle\operatorname*{argmin}_{(x,z)\in X\times Z} \left\{G(x) + F(z) + \langle y_n, K\,x - z\rangle + \frac{1}{2\,\lambda}\,\|K\,x - z\|^2\right\} \\ \\ y_{n+1} & = & y_n + \sigma\,(K\,x_{n+1} - z_{n+1}). \end{array}\right.$$

On notera que la méthode des multiplicateurs, telle qu'elle est écrite ici, suppose l'existence d'un minimum unique pour la fonction $(x,z) \mapsto L_\lambda(x,y_n,z)$, ce qui n'est *a priori* pas le cas.

Directions alternées La méthode des multiplicateurs transforme le problème de minimisation du problème composite (5.1) en une série de problèmes de minimisations où les variables sont à nouveau couplées, donc plus difficiles à résoudre. Pour s'affranchir de cette difficulté, on peut choisir de remplacer la minimisation partielle du lagrangien en (x,z) en deux problèmes de minimisations séparées en x et z. Les deux minimisations peuvent se faire en parallèle à partir de l'itération (x_n,y_n,z_n) :

$$\left\{\begin{array}{rcl} x_{n+1} & = & \displaystyle\operatorname*{argmin}_{x\in X} L_\lambda(x,y_n,z_n) \\ \\ z_{n+1} & = & \displaystyle\operatorname*{argmin}_{z\in Z} L_\lambda(x_n,y_n,z) \end{array}\right.$$

ou bien, dans le cas de l'ADMM, de manière successive, en mettant à jour l'une de deux variables primales entre les deux minimisations partielles, par exemple :

$$\left\{\begin{array}{rcl} x_{n+1} & = & \displaystyle\operatorname*{argmin}_{x\in X} L_\lambda(x,y_n,z_n) \\ \\ z_{n+1} & = & \displaystyle\operatorname*{argmin}_{z\in Z} L_\lambda(x_{n+1},y_n,z). \end{array}\right.$$

L'ADMM nous amène donc à considérer l'algorithme suivant :

$$\left\{\begin{array}{rcl} x_{n+1} & = & \displaystyle\operatorname*{argmin}_{x\in X}\left\{G(x)+\langle y_n,K\,x\rangle+\frac{1}{2\,\lambda}\,\|K\,x-z_n\|^2\right\}\\ z_{n+1} & = & \displaystyle\operatorname*{argmin}_{z\in Z}\left\{F(z)-\langle y_n,z\rangle+\frac{1}{2\,\lambda}\,\|K\,x_{n+1}-z\|^2\right\}\\ y_{n+1} & = & \displaystyle y_n+\frac{1}{\lambda}\,(K\,x_{n+1}-z_{n+1}). \end{array}\right.$$

où, pour plus de lisibilité, on a supprimé les termes constants superflus. Ainsi qu'on le montrera à la section suivante, le choix $\sigma = 1/\lambda$ pour le pas de montée est un choix naturel pour cet algorithme. Dans [1], les auteurs montrent par ailleurs que cette valeur apparaît lorsque l'on construit l'algorithme ADMM en écrivant les conditions d'optimalité pour les points selles du lagrangien.

Une fois encore, l'algorithme proposé suppose l'unicité des solutions des deux problèmes de minimisations partielles, ce qui n'est pas assuré pour le premier problème (le second est un problème fortement convexe donc strictement convexe). Il faut pour cela introduire d'autres hypothèses, généralement sur la forte convexité de G ou une hypothèse sur le rang de K, pour assurer la forte convexité du premier problème.

Convergence La plupart des résultats de convergence, dont certains seront présentés plus bas, repose sur des hypothèses plus ou moins restrictives concernant l'opérateur K ou les fonctions G et F. Les cas les plus intéressants sont ceux qui transforment les deux problèmes de minimisations partielles en des problèmes fortement convexes, soit grâce à la forte convexité des fonctions, soit grâce au rang de la matrice K (qui rend le terme quadratique $x \mapsto \|K\,x - z_n\|^2$ fortement convexe). Ce sont naturellement les conditions les plus restrictives.

Un premier résultat très général [3] assure que

Théorème 13 *Si le lagrangien L admet un point-selle, alors on a trois résultats de convergence pour la suite des itérées de l'algorithme ADMM*

1. *convergence du résiduel : si on pose $r_n = K\,x_n - z_n$ pour tout entier n, la suite $(r_n)_{n\in\mathbb{N}}$ converge vers 0 ;*
2. *convergence vers le minimum : la suite $(G(x_n)+F(z_n))_{n\in\mathbb{N}}$ tend vers le minimum de la fonction $x \mapsto G(x) + F(Kx)$;*
3. *convergence de la variable duale : la suite $(y_n)_{n\in\mathbb{N}}$ converge vers y^*.*

On notera que, sans hypothèse additionnelle, la convergence des variables primales n'est pas assurée.

Il faut par ailleurs distinguer les résultats de convergence dit *ergodiques*, qui s'appliquent à la suite définie comme une moyenne arithmétique (pondérée ou non) des variables générées par l'algorithme, des résultats non ergodiques, qui concernent directement la suite des variables $(x_n,y_n,z_n)_{n\in\mathbb{N}}$.

Généralisation La méthode des directions alternées a été en réalité introduite pour résoudre le problème plus général

$$\min_{\substack{(x,z)\in X\times Z\\ Ax=Bz}}\Big\{G(x)+F(z)\Big\}.$$

On se ramène au cas étudié ici lorsque $A = K$ et $B = I_Z$ (l'identité). La raison pour laquelle on se place dans ce chapitre dans ce cas particulier tient au fait que nous ferons plus tard des hypothèses de régularité qui se traduiront dans le cas général par la forte convexité de la fonction $y \mapsto F^*(B^*(y))$. Cette hypothèse se traduit en particulier par l'inversibilité de l'opérateur B, qui permet de se ramener dans le cas particulier en réécrivant la contrainte linéaire $B^{-1}Ax = z$.

5.1.3 Liens avec d'autres algorithmes

On a déjà vu le lien entre l'ADMM et la méthode des multiplicateurs. Dans ce paragraphe, on va expliciter ses liens avec d'autres méthodes fréquemment rencontrées en optimisation convexe, ainsi que quelques unes des variantes proposées dans la littérature.

Méthode Douglas-Rachford Montrons que l'ADMM n'est rien d'autre que l'algorithme de Douglas-Rachford appliqué au problème dual. Celui-ci est donné par

$$\max_{y\in Z} \Big\{ -G^*(-K^*y) - F^*(y) \Big\} \tag{5.4}$$

qui est équivalent au problème

$$\min_{y\in Z} \Big\{ g(y) + f(y) \Big\}$$

avec $g(y) = G^*(-K^*y)$ et $f(y) = F^*(y)$. La méthode de Douglas-Rachford résout ce genre de problème à l'aide de l'algorithme

$$\left\{ \begin{array}{rcl} p_{n+1} & = & \mathrm{prox}_{g/\lambda}(2\,y_n - q_n) \\ q_{n+1} & = & q_n + p_{n+1} - y_n \\ y_{n+1} & = & \mathrm{prox}_{f/\lambda}(q_{n+1}) \end{array} \right.$$

pour un certain $\lambda > 0$.

Montrons que cet algorithme, appliqué au problème dual (5.4), devient la méthode des directions alternées. Commençons par considérer la mise-à-jour de la variable duale y. L'identité de Moreau assure que

$$y_{n+1} = q_{n+1} - \frac{1}{\lambda}\,\mathrm{prox}_{\lambda f^*}(\lambda\, q_{n+1}). \tag{5.5}$$

Intéressons-nous ensuite à la mise-à-jour de la première variable auxiliaire p. Par définition de l'opérateur proximal, puis de la conjuguée convexe de la fonction G, on a

$$\begin{aligned} p_{n+1} &= \operatorname*{argmin}_{p\in Z} \left\{ \frac{\lambda}{2}\,\|2\,y_n - q_n - p\|^2 + G^*(-K^*p) \right\} \\ &= \operatorname*{argmin}_{p\in Z} \left\{ \sup_{x\in X} \frac{\lambda}{2}\,\|2\,y_n - q_n - p\|^2 + \langle -K^*p, x\rangle - G(x) \right\} \end{aligned}$$

Aussi, le problème que l'on cherche à résoudre est le problème de recherche de point-selle

$$\min_{p\in Z}\sup_{x\in X} \left\{ \frac{\lambda}{2}\,\|2\,y_n - q_n - p\|^2 + \langle -K^*p, x\rangle - G(x) \right\}$$

qui est équivalent à

$$\max_{x\in X}\inf_{p\in Z}\left\{\frac{\lambda}{2}\,\|2\,y_n-q_n-p\|^2+\langle -K^*p,x\rangle-G(x)\right\}$$

Or, pour tout $x\in X$ et $p\in Z$ donnés, on a

$$\begin{aligned}\frac{\lambda}{2}\,\|2\,y_n-q_n-p\|^2+\langle -K^*p,x\rangle &= \frac{\lambda}{2}\,\left\|2\,y_n-q_n+\frac{Kx}{\lambda}-p\right\|^2\\ &\quad-\frac{\lambda}{2}\,\left\|2\,y_n-q_n+\frac{Kx}{\lambda}\right\|^2+\frac{\lambda}{2}\,\|2\,y_n-q_n\|^2\end{aligned}$$

Cette quantité est minimale par rapport à p lorsque le premier terme s'annule, ce qui est le cas pour $p_{n+1}=2\,y_n-q_n+K\,x/\lambda$. On obtient alors, pour tout $x\in X$ donné,

$$\begin{aligned}&\inf_{p\in Z}\left\{\frac{\lambda}{2}\,\|2\,y_n-q_n-p\|^2+\langle -K^*p,x\rangle-G(x)\right\}\\ &\qquad=-\frac{\lambda}{2}\,\left\|2\,y_n-q_n+\frac{Kx}{\lambda}\right\|^2+\frac{\lambda}{2}\,\|2\,y_n-q_n\|^2-G(x).\end{aligned}$$

Finalement, la mise-à-jour de la variable auxiliaire p devient

$$p_{n+1}=2\,y_n-q_n+\frac{1}{\lambda}\,K\,x_{n+1}$$

où la variable x_{n+1} est donnée par

$$\begin{aligned}x_{n+1}&=\operatorname*{argmax}_{x\in X}\left\{-\frac{\lambda}{2}\,\left\|2\,y_n-q_n+\frac{Kx}{\lambda}\right\|^2+\frac{\lambda}{2}\,\|2\,y_n-q_n\|^2-G(x)\right\}\\ &=\operatorname*{argmin}_{x\in X}\left\{\frac{\lambda}{2}\,\left\|2\,y_n-q_n+\frac{Kx}{\lambda}\right\|^2+G(x)\right\}.\end{aligned}\tag{5.6}$$

L'introduction de la nouvelle variable x permet de réécrire la mise-à-jour de la seconde variable auxiliaire q :

$$q_{n+1}=y_n+\frac{1}{\lambda}\,K\,x_{n+1}.\tag{5.7}$$

Revenons à la mise-à-jour de la variable duale y, donnée par (5.5), et que l'on réécrit à l'aide de la nouvelle variable x :

$$y_{n+1}=y_n+\frac{1}{\lambda}\left(K\,x_{n+1}-\operatorname{prox}_{\lambda f^*}(\lambda\,y_n+K\,x_{n+1})\right).\tag{5.8}$$

Posons $z_{n+1}=\operatorname{prox}_{\lambda f^*}(\lambda\,y_n+K\,x_{n+1})$; la définition de l'opérateur proximal assure que

$$\begin{aligned}z_{n+1}&=\operatorname*{argmin}_{z\in Z}\left\{\frac{1}{2\,\lambda}\,\|z-(\lambda\,y_n+K\,x_{n+1})\|^2+F(z)\right\}\\ &=\operatorname*{argmin}_{z\in Z}\left\{\frac{1}{2\,\lambda}\,\|z-K\,x_{n+1}\|^2-\langle y_n,z\rangle+F(z)\right\}.\end{aligned}\tag{5.9}$$

L'introduction de cette nouvelle variable primale permet de réécrire (5.8) et (5.7) respectivement :

$$y_{n+1}=y_n+\frac{1}{\lambda}\,(K\,x_{n+1}-z_{n+1})\qquad\text{et}\qquad q_{n+1}=y_{n+1}+\frac{1}{\lambda}\,z_{n+1}\tag{5.10}$$

ce qui assure que la mise-à-jour (5.6) de la variable duale x se lit

$$\begin{aligned} x_{n+1} &= \operatorname*{argmin}_{x\in X} \left\{ \frac{\lambda}{2} \left\| y_n + \frac{Kx - z_n}{\lambda} \right\|^2 + G(x) \right\} \\ &= \operatorname*{argmin}_{x\in X} \left\{ \frac{1}{2\,\lambda} \|Kx - z_n\|^2 + \langle K\,x, y_n \rangle + G(x) \right\}. \end{aligned} \tag{5.11}$$

Finalement, le calcul de y (donné par la première équation dans (5.10)) ne nécessite que les calculs des deux variables primales x et z, donc les mises-à-jours sont données par (5.11) et (5.9). C'est précisément l'algorithme ADMM.

ADMM proximal (PADMM) Au lieu de calculer exactement les deux minimisations dans l'algorithme ADMM, on peut se contenter d'une minimisation approchée, en ajoutant un terme quadratique dans les deux lagrangiens considérés [1] :

$$\left\{ \begin{array}{rcl} x_{n+1} &=& \operatorname*{argmin}_{x\in X} \left\{ G(x) + \langle y_n, K\,x \rangle + \dfrac{1}{2\,\lambda} \|K\,x - z_n\|^2 + \dfrac{1}{2} \|x - x_n\|^2 \right\} \\ z_{n+1} &=& \operatorname*{argmin}_{z\in Z} \left\{ F(z) - \langle y_n, z \rangle + \dfrac{1}{2\,\lambda} \|K\,x_{n+1} - z\|^2 + \dfrac{1}{2} \|z - z_n\|^2 \right\} \\ y_{n+1} &=& y_n + \dfrac{1}{\lambda} (K\,x_{n+1} - z_{n+1}). \end{array} \right.$$

On voit alors que cela revient à introduire deux opérateurs proximaux pour les mises-à-jours des deux variables primales : c'est pourquoi on parle d'*ADMM proximal*. Un intérêt majeur de l'ajout de ces termes quadratiques est de rendre les deux problèmes associés fortement convexes, donc d'assurer l'unicité de la solution.

Utilisation des distances de Bregman On peut remplacer dans l'algorithme PADMM les opérateurs proximaux par des itérations de Bregman :

$$\left\{ \begin{array}{rcl} x_{n+1} &=& \operatorname*{argmin}_{x\in X} \left\{ G(x) + \langle y_n, K\,x \rangle + \dfrac{1}{2\,\lambda} \|K\,x - z_n\|^2 + D_{x_n}(x, x_n) \right\} \\ z_{n+1} &=& \operatorname*{argmin}_{z\in Z} \left\{ F(z) - \langle y_n, z \rangle + \dfrac{1}{2\,\lambda} \|K\,x_{n+1} - z\|^2 + D_{z_n}(z, z_n) \right\} \\ y_{n+1} &=& y_n + \dfrac{1}{\lambda} (K\,x_{n+1} - z_{n+1}). \end{array} \right.$$

Les plus courantes consistent à considérer deux matrices positives P et Q pour définir la distance de Bregman $\|\cdot\|_P = \langle P\,\cdot, \cdot \rangle$ (*idem* pour $\|\cdot\|_Q$). On choisit généralement les deux matrices P et Q de la forme $P = I - \alpha\ {}^{\mathrm{t}}MM$ (voir le chapitre suivant).

ADMM relaxé Dans un paragraphe précédent, on a montré que l'ADMM est en réalité la méthode de Douglas-Rachford appliquée au problème dual (5.4). Or, une variante de cet algorithme, appelée Peaceman-Rachford relaxé, est donnée par

$$\left\{ \begin{array}{rcl} p_{n+1} &=& \operatorname{prox}_{g/\lambda}(2\,y_n - q_n) \\ q_{n+1} &=& q_n + 2\,\beta_n\,(p_{n+1} - y_n) \\ y_{n+1} &=& \operatorname{prox}_{f/\lambda}(q_{n+1}) \end{array} \right.$$

où la suite $(\beta_n)_n$ est à valeurs dans $]\,0\,;1\,]$. On retrouve la méthode de Douglas-Rachford lorsque cette suite est identiquement égale à $1/2$, tandis que le cas particulier $\beta_n = 1$ est appelé méthode de Peaceman-Rachford. L'idée est donc d'appliquer

la méthode de Peaceman-Rachford relaxé au problème dual pour obtenir une version relaxée d'ADMM. On obtient alors l'algorithme suivant :

$$\begin{cases} x_{n+1} &= \underset{x\in X}{\operatorname{argmin}}\, L_\lambda(x,y_n,z_n) \\ y_{n+1/2} &= y_n + \dfrac{2\,\beta_n - 1}{\lambda}\dfrac{\partial L_\lambda}{\partial y}(x_{n+1},y_n,z_n) \\ z_{n+1} &= \underset{z\in Z}{\operatorname{argmin}}\, L_\lambda(x_{n+1},y_{n+1/2},z) \\ y_{n+1} &= y_{n+1/2} + \dfrac{1}{\lambda}\dfrac{\partial L_\lambda}{\partial y}(x_{n+1},y_{n+1/2},z_{n+1}) \end{cases}$$

que l'on peut interpréter comme des minimisations partielles, avec après chaque mise-à-jour d'une variable primale un pas de montée (ou éventuellement de descente lorsque $\beta_n < 1/2$) de gradient (soit une de plus que dans le cas non relaxé, où $\beta_n = 1/2$).

5.2 Algorithme PDHG : cas régulier

On présente dans cette section une nouvelle preuve de convergence de l'algorithme primal-dual PDHG sur-relaxé, dans le cas particulier où les fonctions considérées sont fortement convexes. La preuve est inspirée de celle proposée dans [4], mais où les conditions sur les différents paramètres ont été choisies les moins contraignantes possibles. On obtient ainsi des taux de convergence optimaux meilleurs, qui sont ceux présentés dans [5] avec une preuve différente.

5.2.1 Algorithme PDHG

Soient X et Y deux espaces de Hilbert réels de dimension finie. On considère à nouveau dans cette section le problème composite

$$\min_{x\in X} f(Ax) + g(x) \tag{5.12}$$

où $f : X \to \mathbb{R}\cup\{+\infty\}$ et $g : Y \to \mathbb{R}\cup\{+\infty\}$ sont des fonctions propres, convexes et s.c.i.. L'opérateur linéaire $A : X \to Y$ est supposé continu, d'adjoint A^*. On suppose par ailleurs qu'il est borné, de norme L

$$L = \|A\| = \sup_{x\in X,\|x\|\leq 1} \|Ax\|.$$

En utilisant la conjuguée convexe $f^* : Y \to \mathbb{R}\cup\{+\infty\}$ de la fonction f, on obtient une nouvelle formulation primale-duale du problème (5.12), donnée par le problème de recherche de point-selle

$$\min_{x\in X}\sup_{y\in Y}\; g(x) + \langle Ax,y\rangle - f^*(y). \tag{5.13}$$

On pose $\mathcal{L}(x,y) = g(x) + \langle Ax,y\rangle - f^*(y)$ le lagrangien du problème (5.13).

Pour résoudre le problème primal-dual (5.13), on se propose d'étudier l'algorithme suivant :

$$\begin{cases} y_0 \in Y \\ x_0 \in X \\ \bar{x}_0 = x_0 \end{cases} \quad \text{et} \quad \forall\, n\in\mathbb{N}, \quad \begin{cases} y_{n+1} = \operatorname{prox}_{\sigma f^*}(y_n + \sigma A\bar{x}_n) \\ x_{n+1} = \operatorname{prox}_{\tau g}(x_n - \tau A^* y_{n+1}) \\ \bar{x}_{n+1} = x_{n+1} + \theta\,(x_{n+1} - x_n) \end{cases} \tag{5.14}$$

où les pas de temps $\sigma > 0$ et $\tau > 0$ et le paramètre de relaxation $0 < \theta \leq 1$ seront spécifiés plus tard. Quand $\theta = 0$, cet algorithme est connu sous le nom PDHG (pour *Primal Dual Hybrid Gradient*) [14]. L'étape de sur-relaxation a été introduite par [13] (pour la minimisation de la fonctionnelle Mumford-Shah), puis étudiée dans un cadre plus large dans [8], puis plus récemment dans [5]. La mise-à-jour de la variable duale y peut être interprétée comme un pas de montée de gradient proximale, tandis que la mise-à-jour de la variable primale est un pas de descente de gradient proximale.

5.2.2 Résultats de convergence dans le cas régulier

Désormais, les fonctions f^* et g sont supposées fortement convexes, de paramètre respectif $\delta > 0$ et $\gamma > 0$. On rappelle que l'opérateur A est de norme L. On définit alors le *conditionnement* de la fonction $g + f(A\cdot)$ comme étant la quantité $\kappa = L^2/(\gamma\delta)$ [1]. On a le résultat de convergence ergodique suivant :

Théorème 14 *Supposons que le problème (5.13) admet une solution, notée (x^*,y^*). Si on choisit les paramètres $\tau > 0$, $\sigma > 0$ et $0 < \theta \leq 1$ tels que*

$$\max\left\{\frac{1}{\tau\gamma+1}, \frac{1}{\sigma\delta+1}\right\} \leq \theta \leq \frac{1}{L^2\tau\sigma} \tag{5.15}$$

alors, pour tout ω vérifiant

$$\max\left\{\frac{1}{\tau\gamma+1}, \frac{\theta+1}{\sigma\delta+2}\right\} \leq \omega \leq \theta \tag{5.16}$$

on a la majoration suivante pour tout $N \in \mathbb{N}$ et pour tout $(x,y) \in X \times Y$:

$$\begin{aligned} 0 \leq \frac{1}{2\tau}\|x_N - x\|^2 + (1-\omega L^2\tau\sigma)\frac{1}{2\sigma}\|y_N - y\|^2 \\ + \sum_{n=1}^{N} \frac{1}{\omega^{n-1}}\Big(\mathcal{L}(x_n,y) - \mathcal{L}(x,y_n)\Big) \\ \leq \frac{\omega^N}{2\tau}\|x_0 - x\|^2 + \frac{\omega^N}{2\sigma}\|y_0 - y\|^2. \end{aligned}$$

Posons maintenant $\displaystyle T_N = \sum_{n=1}^{N} \frac{1}{\omega^{n-1}} = \frac{1-\omega^N}{\omega^{N-1}(1-\omega)}$

et définissons $\displaystyle X_N = \frac{1}{T_N}\sum_{n=1}^{N} \frac{1}{\omega^{n-1}} x_n \quad \text{et} \quad Y_N = \frac{1}{T_N}\sum_{n=1}^{N} \frac{1}{\omega^{n-1}} y_n.$

Alors on a l'encadrement suivant pour tout $(x,y) \in X \times Y$

$$\begin{aligned} 0 \leq \frac{1-\omega}{\omega(1-\omega^N)}\frac{1}{2\tau}\|x - x_N\|^2 + \frac{1-\omega}{\omega(1-\omega^N)}(1-\omega L^2\tau\sigma)\frac{1}{2\sigma}\|y - y_N\|^2 \\ + \mathcal{L}(X_N,y) - \mathcal{L}(x,Y_N) \\ \leq \frac{1}{T_N}\frac{1}{2\tau}\|x - x_0\|^2 + \frac{1}{T_N}\frac{1}{2\sigma}\|y - y_0\|^2. \end{aligned}$$

1. Cette définition recouvre celle usuellement utilisée pour les fonctions fortement convexes et de gradient lipschitizien, qui est alors définie comme étant le rapport de la constante de Lipschitz sur le paramètre de forte convexité. Dans ce cas, le conditionnement est nécessaire supérieur ou égal à 1. Dans le cas plus général considéré ici, le conditionnement peut être inférieur à 1.

Démonstration : Commençons par considérer une itération de l'algorithme dans sa forme la plus générale : pour tous $(\bar{y},\tilde{y}) \in Y^2$ et $(\bar{x},\tilde{x}) \in X^2$, on pose

$$\begin{cases} \hat{y} = \operatorname{prox}_{\sigma f^*}(\bar{y} + \sigma A\tilde{x}) \\ \hat{x} = \operatorname{prox}_{\tau g}(\bar{x} - \tau A^*\tilde{y}). \end{cases}$$

Optimalité Par définition de l'opérateur proximal, le point $\hat{x}$ est le minimiseur d'une fonction fortement convexe :

$$\hat{x} = \operatorname*{argmin}_{x\in X} \left\{ \frac{1}{2\tau} \|\bar{x} - \tau A^*\tilde{y} - x\|^2 + g(x) \right\}.$$

Les conditions nécessaires d'optimalité du premier ordre assurent que

$$-\frac{1}{\tau}(\hat{x} - \bar{x}) - A^*\tilde{y} \in \partial g(\hat{x}).$$

On obtient de manière similaire (en considérant la définition du point $\hat{y}$)

$$-\frac{1}{\sigma}(\hat{y} - \bar{y}) + A\tilde{x} \in \partial f^*(\hat{y}).$$

Forte convexité Exploitons à présent les hypothèses de régularité. En utilisant la définition de la forte convexité, on obtient (en développant les produits scalaires)

$$g(x) + \frac{1}{2\tau}\|x-\bar{x}\|^2 \geq g(\hat{x}) + \langle A(\hat{x}-x),\tilde{y}\rangle + \frac{1}{2\tau}\|\hat{x}-\bar{x}\|^2 + \frac{1}{2\tau}\|x-\hat{x}\|^2 + \frac{\gamma}{2}\|x-\hat{x}\|^2 \quad (5.17)$$

et

$$f^*(y) + \frac{1}{2\sigma}\|y-\bar{y}\|^2 \geq f^*(\hat{y}) - \langle A\tilde{x},\hat{y}-y\rangle + \frac{1}{2\sigma}\|\hat{y}-\bar{y}\|^2 + \frac{1}{2\sigma}\|y-\hat{y}\|^2 + \frac{\delta}{2}\|y-\hat{y}\|^2. \quad (5.18)$$

Sommons les deux inégalités (5.17) et (5.18). On obtient après un réarrangement des termes

$$\begin{aligned} \mathcal{L}(\hat{x},y) - \mathcal{L}(x,\hat{y}) \leq{}& \frac{1}{2\tau}\|x-\bar{x}\|^2 - \frac{1+\tau\gamma}{2\tau}\|x-\hat{x}\|^2 - \frac{1}{2\tau}\|\bar{x}-\hat{x}\|^2 \\ &+ \frac{1}{2\sigma}\|y-\bar{y}\|^2 - \frac{1+\sigma\delta}{2\sigma}\|y-\hat{y}\|^2 - \frac{1}{2\sigma}\|\bar{y}-\hat{y}\|^2 \\ &+ \langle A(\hat{x}-x),\hat{y}-\tilde{y}\rangle - \langle A(\hat{x}-\tilde{x}),\hat{y}-y\rangle. \end{aligned}$$

Choix de $\hat{x}$, $\bar{x}$, $\tilde{x}$ et $\hat{y}$, $\bar{y}$, $\tilde{y}$ On spécifie maintenant les différents points impliqués dans l'itération. On choisit $\hat{x} = x_{n+1}$, $\bar{x} = x_n$, $\tilde{x} = x_n + \theta(x_n - x_{n-1})$ pour $1 \geq \theta > 0$ (pour l'instant laissé libre), et $\hat{y} = y_{n+1}$, $\bar{y} = y_n$ et $\tilde{y} = y_{n+1}$. Après simplification, on obtient

$$\begin{aligned} \mathcal{L}(x_{n+1},y) - \mathcal{L}(x,y_{n+1}) \leq{}& \frac{1}{2\tau}\|x-x_n\|^2 + \frac{1}{2\sigma}\|y-y_n\|^2 \\ &- \frac{1+\tau\gamma}{2\tau}\|x-x_{n+1}\|^2 - \frac{1+\sigma\delta}{2\sigma}\|y-y_{n+1}\|^2 \\ &- \frac{1}{2\tau}\|x_n-x_{n+1}\|^2 - \frac{1}{2\sigma}\|y_n-y_{n+1}\|^2 \\ &+ \theta\langle A(x_{n-1}-x_n),y-y_{n+1}\rangle - \langle A(x_n-x_{n+1}),y-y_{n+1}\rangle. \end{aligned} \quad (5.19)$$

Posons $\tau\gamma = \mu > 0$ et $\sigma\delta = \mu' > 0$. Pour tout $n \in \mathbb{N}$, on définit

$$\Delta_n = \frac{1}{2\tau} \|x - x_n\|^2 + \frac{1}{2\sigma} \|y - y_n\|^2.$$

On peut alors réécrire l'inégalité (5.19) avec Δ_n, μ et μ', ce qui donne

$$\begin{aligned}\mathcal{L}(x_{n+1},y) - \mathcal{L}(x,y_{n+1}) \leq \Delta_n - (1+\mu)\,\Delta_{n+1} - \frac{1}{2\tau}\|x_n - x_{n+1}\|^2 - \frac{1}{2\sigma}\|y_n - y_{n+1}\|^2 \\ + \theta\,\langle A(x_{n-1} - x_n), y - y_{n+1}\rangle - \langle A(x_n - x_{n+1}), y - y_{n+1}\rangle \\ + \frac{\mu - \mu'}{2\sigma}\|y - y_{n+1}\|^2.\end{aligned} \tag{5.20}$$

Produits scalaires dans (5.20) Majorons les produits scalaires dans l'inégalité (5.20). On commence par écrire pour tout $0 < \omega \leq \theta$ la décomposition

$$\begin{aligned}\theta\,\langle A(x_{n-1} - x_n), y - y_{n+1}\rangle &= \omega\,\langle A(x_{n-1} - x_n), y - y_n\rangle \\ &\quad + \omega\,\langle A(x_{n-1} - x_n), y_n - y_{n+1}\rangle \\ &\quad + (\theta - \omega)\,\langle A(x_{n-1} - x_n), y - y_{n+1}\rangle.\end{aligned}$$

Bornons chacun des deux derniers termes. En utilisant l'inégalité de Cauchy- Schwarz puis la définition de la norme L, on a pour tout $\alpha > 0$ (puisque $\omega \geq 0$)

$$\begin{aligned}\omega\,\langle A(x_{n-1} - x_n), y_n - y_{n+1}\rangle &\leq \omega\,L\,\|x_{n-1} - x_n\| \cdot \|y_n - y_{n+1}\| \\ &\leq \omega\,L\left(\frac{\alpha}{2}\|x_{n-1} - x_n\|^2 + \frac{1}{2\alpha}\|y_n - y_{n+1}\|^2\right).\end{aligned} \tag{5.21}$$

De même, puisque $\theta - \omega \geq 0$, on a pour tout $\alpha' > 0$

$$(\theta - \omega)\,\langle A(x_{n-1} - x_n), y - y_{n+1}\rangle \leq (\theta - \omega)\,L\left(\frac{\alpha'}{2}\|x_{n-1} - x_n\|^2 + \frac{1}{2\alpha'}\|y - y_{n+1}\|^2\right). \tag{5.22}$$

En injectant dans (5.20) les deux bornes (5.21) et (5.22), on obtient après simplification

$$\begin{aligned}\mathcal{L}(x_{n+1},y) - \mathcal{L}(x,y_{n+1}) \leq \Delta_n - (1+\mu)\,\Delta_{n+1} \\ + \left(\omega\,\frac{\alpha}{2} + (\theta - \omega)\,\frac{\alpha'}{2}\right) L\,\|x_{n-1} - x_n\|^2 - \frac{1}{2\tau}\|x_n - x_{n+1}\|^2 \\ + \left(\frac{\omega\,L}{2\alpha} - \frac{1}{2\sigma}\right)\|y_n - y_{n+1}\|^2 \\ + \omega\,\langle A(x_{n-1} - x_n), y - y_n\rangle - \langle A(x_n - x_{n+1}), y - y_{n+1}\rangle \\ + \left(\frac{(\theta - \omega)\,L}{2\alpha'} + \frac{\mu - \mu'}{2\sigma}\right)\|y - y_{n+1}\|^2.\end{aligned}$$

Choisissons $\alpha = \alpha' = \omega L\sigma$. De cette manière, on a $\omega L/\alpha = 1/\sigma$, de sorte que le terme en $\|y_n - y_{n+1}\|^2$ s'annule, ce qui entraîne :

$$\begin{aligned}\mathcal{L}(x_{n+1},y) - \mathcal{L}(x,y_{n+1}) \leq \Delta_n - (1+\mu)\,\Delta_{n+1} \\ + \omega\,\frac{\theta L^2\tau\sigma}{2\tau}\,\|x_{n-1} - x_n\|^2 - \frac{1}{2\tau}\,\|x_n - x_{n+1}\|^2 \\ + \omega\,\langle A(x_{n-1} - x_n), y - y_n\rangle - \langle A(x_n - x_{n+1}), y - y_{n+1}\rangle \\ + \left(\frac{\theta-\omega}{\omega} + \mu - \mu'\right)\frac{1}{2\sigma}\,\|y - y_{n+1}\|^2.\end{aligned} \tag{5.23}$$

Puisque $1 + \mu = 1/\omega + 1 + \mu - 1/\omega$, on a

$$-(1+\mu)\,\Delta_{n+1} = -\frac{1}{\omega}\,\Delta_{n+1} + \left(\frac{1}{\omega} - \mu - 1\right)\left(\frac{1}{2\tau}\,\|x - x_{n+1}\|^2 + \frac{1}{2\sigma}\,\|y - y_{n+1}\|^2\right)$$

ce qui implique que le membre de gauche dans (5.23) devient

$$\begin{aligned}\Delta_n - \frac{1}{\omega}\,\Delta_{n+1} + \omega\,\frac{\theta L^2\tau\sigma}{2\tau}\,\|x_n - x_{n-1}\|^2 - \frac{1}{2\tau}\,\|x_n - x_{n+1}\|^2 \\ + \omega\,\langle A(x_{n-1} - x_n), y - y_n\rangle - \langle A(x_n - x_{n+1}), y - y_{n+1}\rangle \\ + \left(\frac{1}{\omega} - \mu - 1\right)\frac{1}{2\tau}\,\|x - x_{n+1}\|^2 + \left(\frac{\theta-\omega}{\omega} + \frac{1}{\omega} - \mu' - 1\right)\frac{1}{2\sigma}\,\|y - y_{n+1}\|^2.\end{aligned} \tag{5.24}$$

Conditions sur ω, θ, τ et σ L'idée est à présent d'imposer des contraintes sur les valeurs des paramètres de sorte de simplifier l'inégalité précédente en la majorant par une expression plus simple. On commence par imposer que θ, τ et σ vérifient $\theta L^2\tau\sigma \leq 1$. Ensuite, on choisit θ de sorte que $1/\omega - \mu - 1$ et $(\theta-\omega)/\omega + 1/\omega - \mu' - 1$ soient tous deux négatifs, ce qui entraîne que

$$\frac{1}{\mu+1} \leq \omega \qquad \text{et} \qquad \frac{\theta+1}{\mu'+2} \leq \omega.$$

Ainsi, on peut majorer le terme de droite dans (5.24) par

$$\begin{aligned}\Delta_n - \frac{1}{\omega}\,\Delta_{n+1} + \omega\,\frac{1}{2\tau}\,\|x_{n-1} - x_n\|^2 - \frac{1}{2\tau}\,\|x_n - x_{n+1}\|^2 \\ + \omega\,\langle A(x_{n-1} - x_n), y - y_n\rangle - \langle A(x_n - x_{n+1}), y - y_{n+1}\rangle.\end{aligned}$$

Finalement, si on revient à l'inégalité (5.23), on obtient

$$\begin{aligned}\mathcal{L}(x_n,y) - \mathcal{L}(x,y_n) \leq \Delta_n - \frac{1}{\omega}\,\Delta_{n+1} \\ + \omega\,\frac{1}{2\tau}\,\|x_{n-1} - x_n\|^2 - \frac{1}{2\tau}\,\|x_n - x_{n+1}\|^2 \\ + \omega\,\langle A(x_{n-1} - x_n), y - y_n\rangle - \langle A(x_n - x_{n+1}), y - y_{n+1}\rangle.\end{aligned} \tag{5.25}$$

Convergence ergodique Multiplions (5.25) par $1/\omega^n$ puis sommons pour n entre 0 et $N-1$. Si on pose $x^{-1} = x^0$, certains se télescopent, et on obtient alors

$$\sum_{n=1}^{N} \frac{1}{\omega^{n-1}}\Big(\mathcal{L}(x_n,y) - \mathcal{L}(x,y_n)\Big) \leq \Delta_0 - \frac{1}{\omega^N}\,\Delta_N - \frac{1}{2\tau\omega^{N-1}}\,\|x_{N-1} - x_N\|^2 - \frac{1}{\omega^{N-1}}\,\langle A(x_{N-1} - x_N), y - y_N\rangle. \tag{5.26}$$

On majore à nouveau le produit scalaire à l'aide de $\beta > 0$,

$$-\frac{1}{\omega^{N-1}}\,\langle A(x_{N-1} - x_N), y - y_N\rangle \leq \frac{L}{\omega^{N-1}}\left(\frac{\beta}{2}\,\|x_{N-1} - x_N\|^2 + \frac{1}{2\beta}\,\|y - y_N\|^2\right).$$

L'inégalité (5.26) devient alors

$$\sum_{n=1}^{N} \frac{1}{\omega^{n-1}}\Big(\mathcal{L}(x_n,y) - \mathcal{L}(x,y_n)\Big) \leq \Delta_0 - \frac{1}{\omega^N}\,\Delta_N + \left(\frac{L\beta}{2\omega^{N-1}} - \frac{1}{2\tau\omega^{N-1}}\right)\|x_{N-1} - x_N\|^2 + \frac{L}{\omega^{N-1}}\frac{1}{2\beta}\,\|y_N - y^*\|^2.$$

Si on choisit $\beta = 1/(L\tau)$, le terme en $\|x_{N-1} - x_N\|^2$ disparaît, et on obtient l'inégalité suivante

$$\sum_{n=1}^{N} \frac{1}{\omega^{n-1}}\Big(\mathcal{L}(x_n,y) - \mathcal{L}(x,y_n)\Big) \leq \Delta_0 - \frac{1}{\omega^N}\,\Delta_N + \frac{L^2\tau\sigma}{\omega^{N-1}}\frac{1}{2\sigma}\,\|y - y_N\|^2.$$

Maintenant, on remplace Δ_0 et Δ_n par leurs définitions respectives :

$$\sum_{n=1}^{N} \frac{1}{\omega^{n-1}}\Big(\mathcal{L}(x_n,y) - \mathcal{L}(x,y_n)\Big) \leq \frac{1}{2\tau}\,\|x - x_0\|^2 + \frac{1}{2\sigma}\,\|y - y_0\|^2 - \frac{1}{\omega^N}\,\frac{1}{2\tau}\,\|x - x_N\|^2 - \frac{1}{\omega^N}\,(1 - \omega L^2\tau\sigma)\frac{1}{2\sigma}\,\|y - y_N\|^2.$$

Puisque $\omega L^2\tau\sigma \leq \theta L^2\tau\sigma \leq 1$ et $\mathcal{L}(x_n,y) - \mathcal{L}(x,y_n) \geq 0$ pour tout $n \in \mathbb{N}$, on obtient après un réarrangement des termes

$$\begin{aligned} 0 \leq \frac{1}{\omega^N}\,\frac{1}{2\tau}\,\|x - x_N\|^2 + \frac{1}{\omega^N}\,(1 - \omega L^2\tau\sigma)\frac{1}{2\sigma}\,\|y - y_N\|^2 \\ + \sum_{n=1}^{N} \frac{1}{\omega^{n-1}}\Big(\mathcal{L}(x_n,y) - \mathcal{L}(x,y_n)\Big) \\ \leq \frac{1}{2\tau}\,\|x - x_0\|^2 + \frac{1}{2\sigma}\,\|y - y_0\|^2. \end{aligned}$$

Divisons maintenant par $T_N \neq 0$, ce qui assure que

$$\begin{aligned} 0 \leq \frac{1-\omega}{\omega(1-\omega^N)}\,\frac{1}{2\tau}\,\|x - x_N\|^2 + \frac{1-\omega}{\omega(1-\omega^N)}\,(1 - \omega L^2\tau\sigma)\frac{1}{2\sigma}\,\|y - y_N\|^2 \\ + \frac{1}{T_N}\sum_{n=1}^{N} \frac{1}{\omega^{n-1}}\Big(\mathcal{L}(x_n,y) - \mathcal{L}(x,y_n)\Big) \\ \leq \frac{1}{T_N}\,\frac{1}{2\tau}\,\|x - x_0\|^2 + \frac{1}{T_N}\,\frac{1}{2\sigma}\,\|y - y_0\|^2. \end{aligned} \tag{5.27}$$

Utilisons la convexité du lagrangien $\mathcal{L}$ en sa première variable, et sa concavité en sa seconde variable, pour écrire :

$$0 \leq \mathcal{L}(X_N,y) - \mathcal{L}(x,Y_N) \leq \frac{1}{T_N} \sum_{n=1}^{N} \frac{1}{\omega^{n-1}} \Big(\mathcal{L}(x_n,y) - \mathcal{L}(x,y_n) \Big).$$

Ainsi, l'encadrement (5.27) devient

$$\begin{aligned} 0 \leq \frac{1-\omega}{\omega(1-\omega^N)} \frac{1}{2\tau} \|x - x_N\|^2 + \frac{1-\omega}{\omega(1-\omega^N)} (1 - \omega L^2 \tau\sigma) \frac{1}{2\sigma} \|y - y_N\|^2 \\ + \mathcal{L}(X_N,y) - \mathcal{L}(x,Y_N) \\ \leq \frac{1}{T_N} \frac{1}{2\tau} \|x - x_0\|^2 + \frac{1}{T_N} \frac{1}{2\sigma} \|y - y_0\|^2 \end{aligned} \tag{5.28}$$

ce qui achève la démonstration. ■

Remarque : L'encadrement (5.28) permet de démontrer la convergence linéaire des variables x_n et, si $1 \neq \omega L^2 \tau\sigma$, celle des y_n, de taux ω. Mieux encore, écrivons la définition

$$\mathcal{L}(x_{n+1},y^*) - \mathcal{L}(x^*;y_{n+1}) = g(x_{n+1}) - g(x^*) + f^*(y_{n+1}) - f^*(y^*) + \langle Ax_{n+1},y^* \rangle - \langle Ax^*,y_{n+1} \rangle.$$

La forte convexité de g et de f^* et les conditions d'optimalité pour x^* et y^* impliquent

$$g(x_{n+1}) - g(x^*) \geq \langle -A^*y^*, x_{n+1} - x^* \rangle + \frac{\gamma}{2} \|x_{n+1} - x^*\|^2$$

et

$$f^*(y_{n+1}) - f^*(y^*) \geq \langle Ax^*, y_{n+1} - y^* \rangle + \frac{\delta}{2} \|y_{n+1} - y^*\|^2.$$

Ainsi, on obtient

$$\frac{\gamma}{2} \|x_{n+1} - x^*\|^2 + \frac{\delta}{2} \|y_{n+1} - y^*\|^2 \leq \mathcal{L}(x_{n+1},y^*) - \mathcal{L}(x^*,y_{n+1})$$

puisque la somme des produits scalaires s'annule. Ainsi, si on choisit d'abandonner le contrôle sur le *prima-dual gap*, l'inégalité (5.19) devient

$$\begin{aligned} 0 \leq \frac{1}{2\tau} \|x^* - x_n\|^2 + \frac{1}{2\sigma} \|y^* - y_n\|^2 \\ - \frac{1 + 2\tau\gamma}{2\tau} \|x^* - x_{n+1}\|^2 - \frac{1 + 2\sigma\delta}{2\sigma} \|y^* - y_{n+1}\|^2 \\ - \frac{1}{2\tau} \|x_n - x_{n+1}\|^2 - \frac{1}{2\sigma} \|y_n - y_{n+1}\|^2 \\ + \theta \langle A(x_{n-1} - x_n), y^* - y_{n+1} \rangle - \langle A(x_n - x_{n+1}), y^* - y_{n+1} \rangle. \end{aligned} \tag{5.29}$$

Il s'ensuit que tous les calculs depuis (5.19) jusqu'à (5.28) restent valables, avec cette fois μ et μ' remplacés par $\tilde{\mu} = 2\mu$ et $\tilde{\mu}' = 2\mu'$ et sans les termes en $\mathcal{L}$, de même que dans les conditions sur les paramètres. En d'autres termes, des calculs similaires prouvent le corollaire suivant :

Corollaire 1 *Supposons que le problème (5.13) admet une solution, notée (x^*,y^*). Si on choisit les paramètres $\tau > 0$, $\sigma > 0$, $0 < \theta \leq 1$ tels que*

$$\max\left\{\frac{1}{2\tau\gamma+1},\frac{1}{2\sigma\delta+1}\right\} \leq \theta \leq \frac{1}{L^2\tau\sigma} \tag{5.30}$$

alors, pour tout $\tilde{\omega}$ vérifiant

$$\max\left\{\frac{1}{2\tau\gamma+1},\frac{\theta+1}{2\sigma\delta+2}\right\} \leq \tilde{\omega} \leq \theta \tag{5.31}$$

on a la majoration suivante pour tout $N \in \mathbb{N}$

$$\|x^* - x_N\|^2 \leq \tilde{\omega}^N \left(\|x^* - x_0\|^2 + \frac{\tau}{\sigma}\|y^* - y_0\|^2\right). \tag{5.32}$$

De plus, si $\tilde{\omega}L^2\tau\sigma \neq 1$, alors on a également

$$\|y^* - y_N\|^2 \leq \frac{\tilde{\omega}^N}{1-\tilde{\omega}L_K^2\tau\sigma}\left(\frac{\sigma}{\tau}\|x^* - x_0\|^2 + \|y^* - y_0\|^2\right). \tag{5.33}$$

Pour tous τ, σ et θ, les bornes inférieures $1/(2\tau\gamma+1)$ et $(\theta+1)/(2\sigma\delta+2)$ pour $\tilde{\omega}$ sont plus petites que celles obtenues pour ω. Ainsi, le nouveau taux $\tilde{\omega}$, qui n'est *a priori* valable que pour la convergence des itérées, est meilleur le taux global ω (qui lui est également valable pour la convergence du *gap*). Notons par ailleurs que tout choix de paramètres satisfaisant (5.15) est compatible avec les contraintes (5.30), ce qui signifie qu'un choix de paramètres (τ,σ,θ) donné par le théorème 14 assure à la fois la convergence linéaire du *gap* au taux ω donné par (5.16) et celle des itérées au taux $\tilde{\omega}$ donné par (5.31).

5.2.3 Choix des paramètres

Le théorème 14 est valable tant qu'on parvient à choisir les pas de temps τ et σ et le paramètre de relaxation θ qui satisfont les contraintes imposées. La valeur minimale des ω correspondant donne alors le taux de convergence de l'algorithme. On va étudier ici plusieurs choix possibles, ainsi que les taux de convergence associés. On procèdera de la manière suivante :

1. On fixe $\tau > 0$.
2. On cherche les conditions sur σ pour que les inégalités (5.15) existent pour au moins un θ (*i.e.* pour que le membre de gauche soit inférieur au membre de droite).
3. On minimise la quantité $(\theta+1)/(\sigma\delta+2)$ par rapport à θ vérifiant (5.15) et par rapport à σ vérifiant les conditions déterminées à l'étape précédente.
4. On compare ce minimum à $1/(\tau\gamma+1)$ puis on en déduit une borne inférieure $\omega^*(\tau)$ pour ω grâce à (5.16).
5. On minimise enfin $\omega^*(\tau)$ par rapport à τ puis on en déduit le taux optimal ω^* ainsi que le taux $\tilde{\omega}$ correspondant sur la convergence des itérées.

On adaptera évidemment cette procédure lorsqu'un ou plusieurs paramètres seront fixés au préalable.

Cas $\theta = 1$ Fixons $\tau > 0$. En remplaçant $\theta = 1$ dans (5.15), on obtient la contrainte suivante sur le pas σ

$$1 \leq \frac{1}{L^2\tau\sigma} \tag{5.34}$$

ce qui implique que $\sigma \leq 1/(L^2\tau)$. La valeur de θ étant fixée, on cherche maintenant à minimiser $1/(\sigma\delta/2+1)$ par rapport à σ vérifiant les conditions (5.34). Puisque la fonction $\sigma \mapsto 1/(\sigma\delta/2+1)$ est décroissante sur $\mathbb{R}^+$, son minimum est atteint lorsque σ est maximal, et vaut

$$\min_{\sigma \text{ vérifiant (5.34)}} \left\{ \frac{1}{\sigma\delta/2+1} \right\} = \frac{1}{\delta/(2L^2\tau)+1}.$$

Comparons cette valeur à $1/(\tau\gamma+1)$. Il est clair que $1/(\delta/(2L^2\tau)+1)$ est supérieur à $1/(\tau\gamma+1)$ dès que $\tau \geq \sqrt{\delta/(2\gamma L^2)}$. Ainsi, la borne inférieure $\omega^*(\tau)$, qui vaut le maximum entre ces deux quantités, est donnée par

$$\omega^*(\tau) = \max\left\{ \frac{1}{\tau\gamma+1}, \frac{1}{\delta/(2L^2\tau)+1} \right\} = \begin{cases} \dfrac{1}{\tau\gamma+1} & \text{if } 0 < \tau < \sqrt{\delta/(2\gamma L^2)} \\ \dfrac{1}{\delta/(2L^2\tau)+1} & \text{if } \tau \geq \sqrt{\delta/(2\gamma L^2)}. \end{cases}$$

Celle-ci est minimale pour $\tau^* = \sqrt{\delta/(2\gamma L^2)}$ et conduit au taux de convergence optimal

$$\omega^* = \omega^*(\tau^*) = \frac{1}{\sqrt{(\gamma\delta)/(2L^2)}+1} = \frac{1}{\sqrt{1/(2\kappa)}+1}.$$

Ce taux est atteint pour

$$\tau = \tau^* = \sqrt{\frac{\delta}{2\gamma L^2}} \qquad \text{et} \qquad \sigma = \frac{1}{L^2\tau^*} = \sqrt{\frac{2\gamma}{\delta L^2}}.$$

On vérifie par ailleurs que le taux de convergence sur les itérées $\tilde{\omega}$ associé à ce choix de paramètres vérifie dans ce cas $\tilde{\omega} = \omega^*$ et que cette valeur est minimale étant donnée la contrainte $\theta = 1$.

Meilleur taux de convergence ($\theta < 1$) On cherche à présent à déterminer les valeurs des paramètres conduisant au taux de convergence minimal.

Théorème 15 *Le meilleur taux de convergence dans le théorème 14 est obtenu en choisissant*

$$\tau = \frac{\delta}{2L^2}\left(1+\sqrt{1+\frac{4L^2}{\gamma\delta}}\right) \qquad \textit{et} \qquad \sigma = \frac{\gamma}{2L^2}\left(1+\sqrt{1+\frac{4L^2}{\gamma\delta}}\right)$$

pour les pas de temps, ainsi que, pour le paramètre de relaxation,

$$\theta = \frac{\sqrt{1+\dfrac{4L^2}{\gamma\delta}}-1}{\sqrt{1+\dfrac{4L^2}{\gamma\delta}}+1} = \frac{\sqrt{1+4\kappa}-1}{\sqrt{1+4\kappa}+1} < 1.$$

Ces valeurs satisfont $\tau\gamma = \sigma\delta$ et $\theta L^2\tau\sigma = 1$. On a par ailleurs $\omega^* = \theta$.

Démonstration : Fixons $\tau > 0$ et déterminons les condtions que σ doit vérifier pour assurer l'existence de θ satisfaisant (5.15). Il existe de telles valeurs de θ si

$$\frac{1}{\tau\gamma + 1} \leq \frac{1}{L^2\tau\sigma} \qquad \text{et} \qquad \frac{1}{\sigma\delta + 1} \leq \frac{1}{L^2\tau\sigma}.$$

qui s'écrit également

$$\sigma \leq \frac{1}{L^2\tau} + \frac{\gamma}{L^2} \qquad \text{et} \qquad (L^2\tau - \delta)\,\sigma \leq 1.$$

Déterminons ensuite les conditions sur σ pour que ces inégalités existent. Si $L^2\tau - \delta$ est négatif, *i.e.* $\tau \leq \delta/L^2$, la seconde inéqualité est toujours vérifiée. La première inégalité fournit donc une borne supérieure pour σ. Ainsi, étudions le cas $L^2\tau - \delta$ est strictement positif, *i.e.* $\tau > \delta/L^2$. Le pas σ doit alors vérifier les deux majorations

$$\sigma \leq \frac{1}{L^2\tau} + \frac{\gamma}{L^2} \qquad \text{et} \qquad \sigma \leq \frac{1}{L^2\tau - \delta}. \tag{5.35}$$

Comparons ces deux majorants. Puisque

$$\frac{1}{L^2\tau} + \frac{\gamma}{L^2} - \frac{1}{L^2\tau - \delta} = \frac{\gamma L^2\tau^2 - \gamma\delta\tau - \delta}{L^2\tau(L^2\tau - \delta)}$$

avec $L^2\tau(L^2\tau - \delta)$ positif, $1/(L^2\tau) + \gamma/L^2$ est supérieur à $1/(L^2\tau - \delta)$ si et seulement si $\gamma L^2\tau^2 - \gamma\delta\tau - \delta \geq 0$, c'est-à-dire si et seulement si $\tau \geq \tau^*$, donné par

$$\tau^* = \frac{\delta}{2L^2}\left(1 + \sqrt{1 + \frac{4L^2}{\gamma\delta}}\right) > \frac{\delta}{L^2}.$$

Ainsi, pour tout $\delta/L^2 < \tau \leq \tau^*$, les conditions (5.35) deviennent $\sigma \leq 1/(L^2\tau) + \gamma/L^2$. Si $\tau > \tau^*$, alors les conditions (5.35) se lisent $\sigma \leq 1/(L^2\tau - \delta)$. En conclusion, le paramètre σ doit vérifier

$$\sigma \leq \begin{cases} \dfrac{1}{L^2\tau} + \dfrac{\gamma}{L^2} & \text{si } 0 < \tau \leq \tau^* \\ \dfrac{1}{L^2\tau - \delta} & \text{si } \tau^* < \tau. \end{cases} \tag{5.36}$$

Maintenant, fixons σ vérifiant (5.36) et minimisons la quantité $(\theta + 1)/(\sigma\delta + 2)$ avec θ vérifiant les contraintes (5.15). La fonction $\theta \mapsto (\theta + 1)/(\sigma\delta + 2)$ atteint son minimum quand θ est minimal. Or, la borne inférieure de θ est donnée par

$$\max\left\{\frac{1}{\tau\gamma + 1}, \frac{1}{\sigma\delta + 1}\right\}.$$

Calculons explicitement cette quantité. Remarquons tout d'abord que, si $\tau > \delta/L^2$, alors

$$\frac{\delta}{L^2\tau - \delta} \leq \tau\gamma \quad \Longleftrightarrow \quad \gamma L^2\tau^2 - \gamma\delta\tau - \delta \geq 0 \quad \Longleftrightarrow \quad \tau \geq \tau^*. \tag{5.37}$$

Supposons que $\tau > \tau^*$, ce qui entraîne $\tau > \delta/L^2$. Puisque σ est majoré par $1/(L^2\tau - \delta)$, on en déduit que $\sigma\delta \leq \tau\gamma$, ce qui implique la borne inférieure de θ est donnée dans ce cas par

$$\theta^*(\sigma) = \max\left\{\frac{1}{\tau\gamma + 1}, \frac{1}{\sigma\delta + 1}\right\} = \frac{1}{\sigma\delta + 1} \quad \text{si} \quad 0 < \sigma \leq \frac{1}{L^2\tau - \delta}. \tag{5.38}$$

Considérons à présent le cas $\tau \leq \tau^*$. Puisque

$$\frac{1}{L^2\tau} + \frac{\gamma}{L^2} \geq \frac{\tau\gamma}{\delta} \quad \Longleftrightarrow \quad \gamma L^2\tau^2 - \gamma\delta\tau - \delta \leq 0 \quad \Longleftrightarrow \quad \tau \leq \tau^*,$$

on en déduit que la borne inférieure de θ vaut

$$\theta^*(\sigma) = \max\left\{\frac{1}{\tau\gamma+1}, \frac{1}{\sigma\delta+1}\right\} = \begin{cases} \dfrac{1}{\sigma\delta+1} & \text{si} \quad 0 < \sigma \leq \dfrac{\tau\gamma}{\delta} \\ \dfrac{1}{\tau\gamma+1} & \text{si} \quad \dfrac{\tau\gamma}{\delta} < \sigma \leq \dfrac{1}{L^2\tau} + \dfrac{\gamma}{L^2}. \end{cases} \tag{5.39}$$

Minimisons $(\theta+1)/(\sigma\delta+2)$ par rapport à σ, quand θ est égal à sa borne inférieure $\theta^*(\sigma)$, donnée par (5.38) lorsque $\tau > \tau^*$ et (5.39) sinon. Cela revient à minimiser la quantité suivante par rapport à σ :

$$\frac{\theta^*(\sigma)+1}{\sigma\delta+2} = \begin{cases} \dfrac{1}{\sigma\delta+1} & \text{si} \quad \tau > \tau^* \text{ ou } \left(\tau \leq \tau^* \text{ et } 0 < \sigma \leq \dfrac{\tau\gamma}{\delta}\right) \\ \dfrac{1}{\tau\gamma+1} \times \dfrac{\tau\gamma+2}{\sigma\delta+2} & \text{si} \quad \left(\tau \leq \tau^* \text{ et } \dfrac{\tau\gamma}{\delta} < \sigma \leq \dfrac{1}{L^2\tau} + \dfrac{\gamma}{L^2}\right). \end{cases}$$

Dans les deux cas, le minimum est atteint quand σ est maximal, égal à sa borne supérieure donnée par (5.36). Ainsi,

$$\min_{\substack{\sigma \text{ vérifiant (5.36)} \\ \theta \text{ vérifiant (5.15)}}} \left\{\frac{\theta+2}{\sigma+1}\right\} = \begin{cases} 1 - \dfrac{\delta}{L^2\tau} & \text{si } \tau > \tau^* \\ \min\left\{\dfrac{1}{\tau\gamma+1}, \dfrac{1}{\tau\gamma+1} \times \dfrac{\tau\gamma+2}{\delta/(L^2\tau) + \delta\gamma/L^2 + 2}\right\} & \text{si } \tau \leq \tau^*. \end{cases}$$

Comparons cette valeur à $1/(\tau\gamma+1)$, pour en déduire la borne inférieure de $\omega^*(\tau)$:

$$\omega^*(\tau) = \max\left\{\frac{1}{\tau\gamma+1}, \min_{\substack{\sigma \text{ vérifiant (5.36)} \\ \theta \text{ vérifiant (5.15)}}} \left\{\frac{\theta+2}{\sigma+1}\right\}\right\}.$$

Grâce à (5.37), on obtient que

$$\omega^*(\tau) = \begin{cases} 1 - \dfrac{\delta}{L^2\tau} & \text{si } \tau > \tau^* \\ \dfrac{1}{\tau\gamma+1} & \text{si } \tau \leq \tau^* \end{cases}$$

qui est minimal pour $\tau = \tau^*$ ce qui conduit au taux optimal :

$$\omega^* = 1 - \frac{\delta}{L^2\tau^*} = \frac{\sqrt{1 + (4L^2)/(\gamma\delta)} - 1}{\sqrt{1 + (4L^2)/(\gamma\delta)} + 1}$$

obtenu pour $\tau = \tau^*$ et $\sigma = \tau^*\gamma/\delta$. On peut par ailleurs vérifier que le taux $\tilde{\omega}$ valable pour la convergence de la variable primale est dans ce cas donné par

$$\tilde{\omega} = \frac{1}{\tau^*\gamma/2 + 1} = \frac{\sqrt{1 + (4L^2)/(\gamma\delta)} - 1}{\sqrt{1 + (4L^2)/(\gamma\delta)} + 3} = \frac{\sqrt{1+4\kappa} - 1}{\sqrt{1+4\kappa} + 3}. \blacksquare$$

5.2.4 Relaxation sur la variable duale

La démonstration précédente peut être adaptée lorsque l'on inverse les rôles des variables primale et duale, c'est-à-dire lorsque l'on considère le problème équivalent

$$\min_{y\in Y}\sup_{x\in X}\Big\{F^*(y)+\langle(-K^*)y,x\rangle-G(x)\Big\}.$$

Dans ce cas, la variable primale est mise-à-jour avant la variable duale, et la relaxation est faite sur la variable duale y au lieu de la variable primale x. Cela conduit à l'algorithme dual

$$\begin{cases} x_{n+1}=\mathrm{prox}_{\tau G}(x_n-\tau K^*\bar{y}_n)\\ y_{n+1}=\mathrm{prox}_{\sigma F^*}(y_n+\sigma K x_{n+1})\\ \bar{y}_{n+1}=y_{n+1}+\theta\,(y_{n+1}-y_n). \end{cases}$$

On obtient alors dans ce cas un résultat similaire :

Théorème 16 *Supposons que le problème (5.13) admet une solution, notée (x^*,y^*). Si on choisit $\tau>0$, $\sigma>0$ et $0<\theta\leq 1$ vérifiant*

$$\max\left\{\frac{1}{\tau\gamma+1},\frac{1}{\sigma\delta+1}\right\}\leq\theta\leq\frac{1}{L^2\tau\sigma} \tag{5.40}$$

alors, pour tout ω tel que

$$\max\left\{\frac{\theta+1}{\tau\gamma+2},\frac{1}{\sigma\delta+1}\right\}\leq\omega\leq\theta$$

alors on a la majoration suivante pour tout $N\in\mathbb{N}$ et pour tout $(x,y)\in X\times Y$:

$$\begin{aligned} 0\leq(1-\omega L^2\tau\sigma)\frac{1}{2\tau}\|x_N-x\|^2+\frac{1}{2\sigma}\|y_N-y\|^2&\\ +\sum_{n=1}^{N}\frac{1}{\omega^{n-1}}\Big(\mathcal{L}(x_n,y)-\mathcal{L}(x,y_n)\Big)&\\ \leq\frac{\omega^N}{2\tau}\|x_0-x\|^2+\frac{\omega^N}{2\sigma}\|y_0-y\|^2.& \end{aligned}$$

Posons pour tout $N\geq 1$

$$T_N=\sum_{n=1}^{N}\frac{1}{\omega^{n-1}}=\frac{1-\omega^N}{\omega^{N-1}(1-\omega)}$$

et définissons

$$X_N=\frac{1}{T_N}\sum_{n=1}^{N}\frac{1}{\omega^{n-1}}\,x_n\qquad\text{et}\qquad Y_N=\frac{1}{T_N}\sum_{n=1}^{N}\frac{1}{\omega^{n-1}}\,y_n.$$

Alors on a l'encadrement suivant pour tout $(x,y)\in X\times Y$

$$\begin{aligned} 0\leq\frac{1-\omega}{\omega(1-\omega^N)}(1-\omega L^2\tau\sigma)\frac{1}{2\tau}\|x-x_N\|^2+\frac{1-\omega}{\omega(1-\omega^N)}\frac{1}{2\sigma}\|y-y_N\|^2&\\ +\mathcal{L}(X_N,y)-\mathcal{L}(x,Y_N)&\\ \leq\frac{1}{T_N}\frac{1}{2\tau}\|x-x_0\|^2+\frac{1}{T_N}\frac{1}{2\sigma}\|y-y_0\|^2.& \end{aligned}$$

5.3 Application à l'ADMM

On commence par rappeler que les itérations de l'ADMM sont équivalentes aux itérations de l'algorithme PDHG avec sur-relaxation étudié dans la section précédente, mais appliqué à un autre problème. Ainsi, on pourra utiliser les résultats de convergence pour en déduire le taux de convergence ergodique optimal d'ADMM. Par ailleurs, on remarquera qu'une légère relaxation d'un des pas dans l'ADMM permet théoriquement d'en accélerer la convergence.

5.3.1 Lien entre PDHG avec sur-relaxation et ADMM

On rappelle que les itérations de l'algorithme ADMM sont données par

$$\begin{cases} z_0 \in X \\ y_0 \in Y \end{cases} \quad \text{et} \quad \forall\, n \in \mathbb{N}, \quad \begin{cases} x_{n+1} = \operatorname*{argmin}_{x\in X} \left\{ G(x) + \langle Kx, y_n\rangle + \dfrac{1}{2\,\lambda} \|Kx - z_n\|^2 \right\} \\ z_{n+1} = \operatorname*{argmin}_{z\in X} \left\{ F(z) - \langle z, y_n\rangle + \dfrac{1}{2\,\lambda} \|Kx_{n+1} - z\|^2 \right\} \\ y_{n+1} = y_n + \dfrac{1}{\lambda}\,(Kx_{n+1} - z_{n+1}). \end{cases}$$

Posons $\xi_{n+1} = Kx_{n+1}$ et introduisons la fonction convexe

$$G_K(\xi) = \inf_{x\in X, Kx=\xi} G(x).$$

Par optimalité, on a pour tout $x \in X$

$$G(x_{n+1}) + \langle Kx_{n+1}, y_n\rangle + \frac{1}{2\,\lambda} \|Kx_{n+1} - z_n\|^2 \leq G(x) + \langle Kx, y_n\rangle + \frac{1}{2\,\lambda} \|Kx - z_n\|^2.$$

On en déduit en particulier que, si $Kx = Kx_{n+1}$, alors

$$G(x_{n+1}) + \langle Kx_{n+1}, y_n\rangle + \frac{1}{2\,\lambda} \|Kx_{n+1} - z_n\|^2 \leq G(x) + \langle Kx_{n+1}, y_n\rangle + \frac{1}{2\,\lambda} \|Kx_{n+1} - z_n\|^2.$$

Il s'ensuit que

$$\begin{aligned} G(x_{n+1}) + \langle Kx_{n+1}, y_n\rangle + \frac{1}{2\,\lambda} \|Kx_{n+1} - z_n\|^2 &= G_K(\xi_{n+1}) + \langle \xi_{n+1}, y_n\rangle + \frac{1}{2\,\lambda} \|\xi_{n+1} - z_n\|^2 \\ &= \min_{\xi\in Y} \left\{ G_K(\xi) + \langle \xi, y_n\rangle + \frac{1}{2\,\lambda} \|\xi - z_n\|^2 \right\}. \end{aligned}$$

Il vient en particulier que

$$\xi_{n+1} = \operatorname*{argmin}_{\xi\in Y} \left\{ G_K(\xi) + \langle \xi, y_n\rangle + \frac{1}{2\,\lambda} \|\xi - z_n\|^2 \right\} = \operatorname{prox}_{\lambda G_K}(z_n - \lambda\, y_n).$$

La montée de gradient en y s'écrit alors

$$\begin{aligned} y_{n+1} = y_n + \frac{1}{\lambda}\,(\xi_{n+1} - z_{n+1}) \quad &\Longleftrightarrow \quad z_{n+1} - \lambda y_{n+1} = \xi_{n+1} - \lambda\,(2\,y_{n+1} - y_n) \\ &\Longleftrightarrow \quad \xi_{n+1} + \lambda\, y_n = z_{n+1} + \lambda y_{n+1}, \end{aligned}$$

et implique d'une part que

$$\xi_{n+1} = \operatorname{prox}_{\lambda G_K}(\xi_n - \lambda\,(2\,y_n - y_{n-1}))$$

et d'autre part, en utilisant l'identité de Moreau, que

$$z_{n+1} = \operatorname{prox}_{\lambda F}(\xi_{n+1} + \lambda\, y_n) = z_{n+1} + \lambda y_{n+1} - \lambda \operatorname{prox}_{F_*/\lambda}(y_n + \xi_{n+1}/\lambda).$$

On peut alors réécrire les mises-à-jour de x_{n+1}, y_{n+1} et z_{n+1} de la manière suivante

$$\begin{cases} \xi_0 \in Y^* \\ y_0 \in Y \\ \bar{y}_0 = y_0 \end{cases} \quad \text{et} \quad \forall\, n \in \mathbb{N}, \quad \begin{cases} \xi_{n+1} = \mathrm{prox}_{\lambda G_K}\left(\xi_n - \lambda\, \bar{y}_n\right) \\ y_{n+1} = \mathrm{prox}_{F^*/\lambda}(y_n + \xi_{n+1}/\lambda) \\ \bar{y}_{n+1} = y_{n+1} + (y_{n+1} - y_n). \end{cases} \tag{5.41}$$

D'après la section précédente, il s'agit de l'algorithme PDHG avec sur-relaxation sur la variable duale, qui résout le problème de recherche de point-selle

$$\min_{\xi \in X} \sup_{y \in Y} \Big\{ G_K(\xi) + \langle \xi, y \rangle - F^*(y) \Big\}$$

où on a $g = G_K$, $f = F$ et $A = \mathrm{Id}$ (de norme $\tilde{L} = 1$). Le paramètre de relaxation vaut $\theta = 1$, les pas de temps valent respectivement $\tau = \lambda$ et $\sigma = 1/\lambda$. On posera par ailleurs $\tilde{\mathcal{L}}(\xi, y) = G_K(\xi) + \langle \xi, y \rangle - F^*(y)$.

5.3.2 Taux de convergence de l'ADMM classique

Utilisons le théorème 16 pour calculer le taux de convergence optimal de l'algorithme (5.41). Pour pouvoir l'appliquer, il nous faut tout d'abord vérifier que g est fortement convexe. Or, si G est fortement convexe (de paramètre γ), alors G_K est fortement convexe, de paramètre γ/L^2. En effet, il est suffisant de montrer G_K^* est différentiable et que son gradient ∇G_K^* est lipschitzien de constante L^2/γ (voir chapitre 1). C'est bien le cas ici car

$$G_K^*(y+h) = G^*\Big(K^*(y+h)\Big) = G^*(K^*y + K^*h) = G^*(K^*y) + \langle \nabla G^*(K^*y), K^*h \rangle + o(\|K^*h\|).$$

Par conséquent, g est fortement convexe, de paramètre $\tilde{\gamma} = \gamma/L^2$. On a par ailleurs par hypothèse que f^* est fortement convexe, de paramètre δ. On pose dans ce cas le conditionnement $\kappa = (1/\delta)/\tilde{\gamma} = L^2/(\gamma\delta)$.

On peut ainsi appliquer le théorème 16. La condition (5.40) est toujours vérifiée, puisque

$$\max\left\{ \frac{1}{\lambda\gamma/L^2 + 1}, \frac{1}{\delta/\lambda + 1} \right\} \leq 1.$$

Ainsi, pour tout $\lambda > 0$, Le taux de convergence vérifie

$$\max\left\{ \frac{1}{(\lambda\gamma)/(2L^2) + 1}, \frac{1}{\delta/\lambda + 1} \right\} \leq \omega \leq 1.$$

La borne inférieure de ce taux vaut $1/((\lambda\gamma)/(2L^2) + 1)$ lorsque $\lambda \leq \sqrt{2\delta L^2/\gamma}$ et vaut $1/(\delta/\lambda + 1)$ sinon. Le taux optimal est donc donné par

$$\omega^* = \frac{1}{\sqrt{(\gamma\delta)/(2L^2)} + 1} = \frac{1}{\sqrt{1/(2\kappa)} + 1} = \tilde{\omega} \quad \text{atteint pour} \quad \lambda = \sqrt{\frac{2\delta L^2}{\gamma}}.$$

Ainsi, si on pose $\Xi_N = \dfrac{1}{T_N}\sum_{n=1}^{N}\dfrac{1}{\omega_{n-1}}\,\xi_n$, on a pour tout $(\xi,y)\in Y^*\times Y$

$$\begin{aligned}0 \leq \frac{1-\omega^*}{\omega^*(1-(\omega^*)^N)}(1-\omega^*)\sqrt{\frac{\gamma}{8\,\delta L^2}}\,\|\xi-\xi_N\|^2 + \frac{1-\omega^*}{\omega^*(1-(\omega^*)^N)}\sqrt{\frac{\delta L^2}{2\gamma}}\,\|y-y_N\|^2 \\ + \tilde{\mathcal{L}}(\Xi_N,y) - \tilde{\mathcal{L}}(\xi,Y_N) \\ \leq \frac{1}{T_N}\sqrt{\frac{\gamma}{8\,\delta L^2}}\,\|\xi-\xi_0\|^2 + \frac{1}{T_N}\sqrt{\frac{\delta L^2}{2\gamma}}\,\|y-y_0\|^2\end{aligned}$$

avec $T_N = (1-(\omega^*)^N)/((\omega^*)^{N-1}(1-\omega^*))$, qui s'écrit également pour tout $\xi = Kx$

$$\begin{aligned}0 \leq \frac{\gamma/L^2}{4(\sqrt{(\gamma\delta)/(2L^2)}+1)}\,\|Kx-Kx_N\|^2 + \frac{\delta}{2\sqrt{(\gamma\delta)/(2L^2)}}\,\|y-y_N\|^2 \\ + \frac{\omega^*(1-(\omega^*)^N)}{1-\omega^*}\left(\tilde{\mathcal{L}}(KX_N,y) - \tilde{\mathcal{L}}(Kx,Y_N)\right) \\ \leq (\omega^*)^N\left(\sqrt{\frac{\gamma}{8\,\delta L^2}}\,\|Kx-Kx_0\|^2 + \sqrt{\frac{\delta L^2}{2\gamma}}\,\|y-y_0\|^2\right).\end{aligned}$$

Cet encadrement assure la convergence linéaire des itérées y_n et Kx_n lorsqu'elle est appliquée à $(x,y) = (x^*,y^*)$ (mais non nécessairement celle de x_n seul). À nouveau, il est possible, en choisissant de perdre le contrôle sur le *primal-dual gap*, d'obtenir comme taux de convergence $\tilde{\omega}$ à la place de ω^*. Par ailleurs, si on pose $E = G + F(K\cdot)$ et $\xi = Kx$, on a par définition,

$$E(x) = G_K(\xi) + F(\xi) = \sup_{y\in Y}\tilde{\mathcal{L}}(\xi,y) = \sup_{y\in Y}\tilde{\mathcal{L}}(Kx,y)$$

On en déduit que $E(x^*) = \tilde{\mathcal{L}}(Kx^*,y^*) \geq \tilde{\mathcal{L}}(Kx^*,Y_N)$, ce qui implique que l'inégalité précédente devient lorsque $x = x^*$ (en ignorant les deux premiers termes quadratiques)

$$\begin{aligned}0 \leq \frac{\omega^*(1-(\omega^*)^N)}{1-\omega^*}\left(\tilde{\mathcal{L}}(KX_N,y) - E(x^*)\right) \\ \leq (\omega^*)^N\left(\sqrt{\frac{\gamma}{8\,\delta L^2}}\,\|Kx-Kx_0\|^2 + \sqrt{\frac{\delta L^2}{2\gamma}}\,\|y-y_0\|^2\right).\end{aligned}$$

Si on connaît par ailleurs un ouvert borné $\mathcal{B}_Y$ contenant y^*, alors on peut prendre la borne supérieure sur $y\in\mathcal{B}_Y$ dans cet encadrement, et obtenir

$$\begin{aligned}0 \leq \frac{\omega^*(1-(\omega^*)^N)}{1-\omega^*}\left(E(X_N) - E(x^*)\right) \\ \leq (\omega^*)^N\left(\sqrt{\frac{\gamma}{8\,\delta L^2}}\,\|Kx^*-Kx_0\|^2 + \sqrt{\frac{\delta L^2}{2\gamma}}\,\sup_{y\in\mathcal{B}_Y}\|y-y_0\|^2\right)\end{aligned}$$

ce qui implique une convergence linéaire (ergodique) de l'énergie primale, de taux ω^*.

5.3.3 Variante proposée de l'ADMM

On se propose maintenant de relaxer le choix du pas λ dans les mises-à-jour z et de y dans les itérations de l'ADMM. Plus précisément, si λ est remplacé par $\lambda' \leq \lambda$ dans chacune de ces deux mises-à-jour :

$$\begin{cases} x_{n+1} = \underset{x \in X}{\operatorname{argmin}} \left\{ G(x) + \langle Kx, y_n \rangle + \dfrac{1}{2\,\lambda} \|Kx - z_n\|^2 \right\} \\ z_{n+1} = \underset{z \in X}{\operatorname{argmin}} \left\{ F(z) - \langle z, y_n \rangle + \dfrac{1}{2\,\lambda'} \|Kx_{n+1} - z\|^2 \right\} \\ y_{n+1} = y_n + \dfrac{1}{\lambda'} \left(Kx_{n+1} - z_{n+1} \right) \end{cases}$$

alors on peut montrer que ces nouvelles itérations sont équivalentes à celles de l'algorithme suivant

$$\begin{cases} \xi^0 \in Y^* \\ y^0 \in Y \\ \bar{y}_0 = y_0 \end{cases} \qquad \text{et} \qquad \forall\, n \in \mathbb{N}, \qquad \begin{cases} \xi^{n+1} = \operatorname{prox}_{\lambda G_K} \left(\xi^n - \lambda\, \bar{y}^n \right) \\ y^{n+1} = \operatorname{prox}_{F^*/\lambda'}(y^n + \xi^{n+1}/\lambda') \\ \bar{y}^{n+1} = y^{n+1} + \dfrac{\lambda'}{\lambda} \left(y^{n+1} - y^n \right) \end{cases}$$

où le paramètre de sur-relaxation vaut cette fois $\theta = \lambda'/\lambda \leq 1$. On a alors $\tau = \lambda$ et $\sigma = 1/\lambda'$. Déterminons le meilleur taux de convergence que cet algorithme peut atteindre, afin d'en déduire la valeur optimale des paramètres.

D'après le théorème 16, les pas λ et λ' doivent être liés par la relation

$$\max \left\{ \frac{1}{\lambda\gamma/L^2 + 1}, \frac{1}{\delta/\lambda' + 1} \right\} \leq \frac{\lambda'}{\lambda} \leq 1.$$

Fixons $\lambda' > 0$. Cherchons les conditions que doit vérifier λ pour que l'encadrement précédent soit valable. De manière équivalente, on cherche λ satisfaisant au moins un des deux ensembles de conditions suivants

$$\frac{1}{\lambda\gamma/L^2 + 1} \leq \frac{1}{\delta/\lambda' + 1} \qquad \text{et} \qquad \frac{1}{\delta/\lambda' + 1} \leq \frac{\lambda'}{\lambda} \qquad \text{et} \qquad \frac{\lambda'}{\lambda} \leq 1 \tag{5.42}$$

ou

$$\frac{1}{\lambda\gamma/L^2 + 1} \geq \frac{1}{\delta/\lambda' + 1} \qquad \text{et} \qquad \frac{1}{\lambda\gamma/L^2 + 1} \leq \frac{\lambda'}{\lambda} \qquad \text{et} \qquad \frac{\lambda'}{\lambda} \leq 1. \tag{5.43}$$

Suivant les valeurs de λ', on va établir si ces deux ensembles de conditions admettent des éléments. Commençons par les conditions (5.42), qui sont équivalentes à

$$\max \left\{ \frac{\delta L^2}{\gamma\lambda'}, \lambda' \right\} \leq \lambda \leq \lambda' + \delta. \tag{5.44}$$

Pour que de tels λ existent, il faut (puisqu'on a nécessairement $\lambda' \leq \lambda' + \delta$) que

$$\frac{\delta L^2}{\gamma\lambda'} \leq \lambda' + \delta. \tag{5.45}$$

Il est aisé de vérifier que tout réel positif λ' vérifiant (5.45) est supérieur à $(\lambda')^*$ (strictement si on considère également l'inégalité stricte dans (5.45)), donné par

$$(\lambda')^* = \frac{\delta}{2}\left(\sqrt{1+\frac{4L^2}{\gamma\delta}}-1\right).$$

Aussi, l'ensemble des réels λ strictement positifs vérifiant les conditions (5.42) est un intervalle non vide si et seulement si $\lambda' \geq (\lambda')^*$. La borne inférieure est alors donnée par λ' si $\lambda' > \sqrt{\delta L^2/\gamma}$ et par $(\delta L^2)/(\gamma\lambda')$ sinon. Intéressons-nous à présent aux conditions (5.43). On peut vérifier qu'elles s'écrivent de manière équivalente

$$\lambda' \leq \lambda \leq \frac{\delta L^2}{\gamma\lambda'} \qquad \text{et} \qquad \lambda\left(1-\frac{\gamma\lambda'}{L^2}\right) \leq \lambda'. \tag{5.46}$$

Le premier encadrement entraîne que $\lambda' \leq (\delta L^2)/(\gamma\lambda')$, soit $\lambda' \leq \sqrt{\delta L^2/\gamma}$. Aussi, on peut d'ores et déjà affirmer que, si λ' ne vérifie pas cette inégalité, alors aucun réel λ ne peut satisfaire les conditions (5.43). On supposera donc dans la suite de cette analyse que le premier encadrement peut être satisfait. Pour interpréter la seconde relation, il est nécessaire de tenir compte du signe du terme entre parenthèses. Deux cas de figure sont envisageables. Si ce terme est négatif ou nul, c'est-à-dire si $\lambda' \geq L^2/\gamma$, alors cette inégalité est toujours vraie. Les conditions (5.46) impliquent alors

$$\frac{L^2}{\gamma} \leq \lambda' \leq \sqrt{\frac{\delta L^2}{\gamma}}.$$

Ainsi, si $L^2/\gamma \leq \delta$, alors il existe des réels λ vérifiant les conditions (5.43) lorsque λ' vérifie l'encadrement précédent. Si $L^2/\gamma > \delta$ ou si $\lambda' < L^2/\gamma$, alors $1-\gamma\lambda'/L^2$ est positif strictement, et les conditions (5.46) deviennent

$$\lambda' \leq \lambda \leq \min\left\{\frac{\delta L^2}{\gamma\lambda'}, \frac{\lambda'}{1-(\gamma\lambda')/L^2}\right\}.$$

Notons que, puisque L^2/γ vérifie (strictement) la relation (5.45), alors il est strictement supérieur à $(\lambda')^*$. On montre de même que $\sqrt{\delta L^2/\gamma} > (\lambda')^*$. Calculons ensuite le membre de droite de l'encadrement précédent. Puisque

$$\frac{\delta L^2}{\gamma\lambda'} \leq \frac{\lambda'}{1-(\gamma\lambda')/L^2} \quad \Longleftrightarrow \quad 0 \leq \frac{\gamma}{L^2}(\lambda')^2 + \frac{\gamma\delta}{L^2}\lambda' - \delta \quad \Longleftrightarrow \quad \lambda' \geq (\lambda')^*$$

on en déduit que, si $\lambda' < (\lambda')^*$, alors le pas λ est encadré par

$$\lambda' \leq \lambda \leq \frac{\lambda'}{1-(\gamma\lambda')/L^2}. \tag{5.47}$$

Si $\sqrt{\delta L^2/\gamma} \geq \lambda' \geq (\lambda')^*$, nous ignorons la seconde condition dans (5.46) et λ est contraint par le premier encadrement. Reprenons à présent l'analyse complète.

1. On a montré que, si $\lambda' < (\lambda')^*$, alors les conditions (5.42) ne peuvent être satisfaites. En revanche, les conditions (5.43) conduisent dans ce cas à l'encadrement (5.47). On a en effet $\lambda' < \sqrt{\delta L^2/\gamma}$ et $\lambda' < L^2/\gamma$.

2. On a ensuite établi que, si $\lambda' > \sqrt{\delta L^2/\gamma}$, alors ce sont cette fois-ci les conditions (5.43) qui ne peuvent être satisfaites. Néanmoins, cette hypothèse implique que $\lambda' \geq (\lambda')^*$; les conditions (5.42) prennent alors la forme (5.44), où la borne inférieure vaut λ' :
$$\lambda' \leq \lambda \leq \lambda' + \delta. \tag{5.48}$$
3. Enfin, si λ' est compris entre $(\lambda')^*$ et $\sqrt{\delta L^2/\gamma}$, alors les deux ensembles de conditions (5.42) et (5.43) peuvent être satisfaites. Elles s'écrivent respectivement
$$\frac{\delta L^2}{\gamma\lambda'} \leq \lambda \leq \lambda' + \delta \qquad \text{et} \qquad \lambda' \leq \lambda \leq \frac{\delta L^2}{\gamma\lambda'}$$
ces deux encadrements pouvant être réunis en un seul, donné par (5.48).

On voit donc finalement que les conditions sur λ ne dépendent que de la valeur relative de λ' par rapport à $(\lambda')^*$.

Maintenant qu'on a établi les valeurs admissibles pour les pas λ et λ', on peut minimiser le taux de convergence ω. On rappelle qu'il est contraint par
$$\max\left\{\frac{\lambda'/\lambda + 1}{\lambda\gamma/L^2 + 2}, \frac{1}{\delta/\lambda' + 1}\right\} \leq \omega \leq \frac{\lambda'}{\lambda}.$$
On cherche donc à minimiser la borne inférieure de cet encadrement. Calculons le minimum de la fonction $\lambda \mapsto (\lambda'/\lambda + 1)(\lambda\gamma/L^2 + 2)$ pour λ satisfaisant la contrainte (5.47) si $\lambda' < (\lambda')^*$ et la contrainte (5.48) sinon. Puisqu'il s'agit d'une fonction strictement décroissante, son minimum est atteint lorsque λ est maximal. Il s'ensuit que
$$\min_{\substack{\lambda \text{ satisfaisant} \\ (5.48) \text{ et } (5.47)}} \left\{\frac{\lambda'/\lambda + 1}{\lambda\gamma/L^2 + 2}\right\} = \begin{cases} 1 - \dfrac{\gamma\lambda'}{L^2} & \text{si } \lambda' < (\lambda')^* \\ \dfrac{1}{\delta/\lambda' + 1}\,\dfrac{\delta/\lambda' + 2}{(\delta/\lambda' + 1)\gamma\lambda'/L^2 + 2} & \text{si } \lambda' \geq (\lambda')^*. \end{cases}$$
Comparons maintenant ce mimimum à $1/(\delta/\lambda' + 1)$. Pour ce faire, on montre d'une part que
$$1 - \frac{\gamma\lambda'}{L^2} \geq \frac{1}{\delta/\lambda' + 1} \quad \Longleftrightarrow \quad \frac{\delta}{\lambda'} - \frac{\delta}{\lambda'}\frac{\gamma\lambda'}{L^2} - \frac{\gamma\lambda'}{L^2} \geq 0 \quad \Longleftrightarrow \quad \lambda' \leq (\lambda')^*$$
et d'autre part que
$$\begin{aligned} \frac{1}{\delta/\lambda' + 1}\,\frac{\delta/\lambda' + 2}{(\delta/\lambda' + 1)\gamma\lambda'/L^2 + 2} \leq \frac{1}{\delta/\lambda' + 1} \quad &\Longleftrightarrow \quad \frac{\delta}{\lambda'} \leq \left(\frac{\delta}{\lambda'} + 1\right)\frac{\gamma\lambda'}{L^2} \\ &\Longleftrightarrow \quad \lambda' \geq (\lambda')^*. \end{aligned}$$
Nous avons donc prouvé que ω est minoré par
$$\omega^*(\lambda') = \begin{cases} 1 - \dfrac{\gamma\lambda'}{L^2} & \text{si } \lambda' < (\lambda')^* \\ \dfrac{1}{\delta/\lambda' + 1} & \text{si } \lambda' \geq (\lambda')^* \end{cases}$$
qui est alors minimal pour $\lambda' = (\lambda')^*$. On en déduit le taux
$$\omega^* = \frac{\sqrt{1 + (4L^2)/(\gamma\delta)} - 1}{\sqrt{1 + (4L^2)/(\gamma\delta)} + 1} = \frac{\sqrt{1 + 4\kappa} - 1}{\sqrt{1 + 4\kappa} + 1}$$

obtenu lorsque les pas de temps λ et λ' sont choisis comme suit :

$$\lambda' = \frac{\delta}{2}\left(\sqrt{1+\frac{4L^2}{\gamma\delta}}-1\right) \qquad \text{et} \qquad \lambda = \lambda'+\delta = \frac{\delta}{2}\left(\sqrt{1+\frac{4L^2}{\gamma\delta}}+1\right).$$

Le lecteur pourra vérifier que cette valeur est strictement inférieure au taux ω^* obtenu dans le cas de l'ADMM classique. Finalement, en reprenant les mêmes calculs que pour l'ADMM classique, on montre que pour tout $N \geq 1$

$$\begin{aligned} 0 \leq \frac{\lambda'}{2}\|y-y_N\|^2 + \frac{\omega^*(1-(\omega^*)^N)}{1-\omega^*}\left(\tilde{\mathcal{L}}(\Xi_N,y)-\tilde{\mathcal{L}}(\xi,Y_N)\right) \\ \leq (\omega^*)^N\left(\frac{1}{2\lambda}\|\xi-\xi_0\|^2+\frac{\lambda'}{2}\|y-y_0\|^2\right) \end{aligned}$$

car, ici, les paramètres optimaux vérifient $\omega^* = \theta = 1/(\tau\sigma)$. On voit ainsi que choisir les paramètres λ et λ' de manière à obtenir la plus petite valeur de ω conduit à perdre le contrôle sur la convergence de la variable x. Cette convergence reste néanmoins assurée par la forte convexité (cf. proposition 7). Pour la convergence de la variable duale, le taux

$$\tilde{\omega} = \frac{\sqrt{1+(4L^2)/(\gamma\delta)}-1}{\sqrt{1+(4L^2)/(\gamma\delta)}+3} = \frac{\sqrt{1+4\kappa}-1}{\sqrt{1+4\kappa}+3}$$

est à nouveau valable.

Remarque : Si l'on tient à conserver ce contrôle, une manière simple de procéder est de considérer un $\tilde{L} \neq L$ et de poser

$$\lambda' = \frac{\delta}{2}\left(\sqrt{1+\frac{4\tilde{L}^2}{\gamma\delta}}-1\right) \qquad \text{et} \qquad \lambda \geq \left(\frac{L}{\tilde{L}}\right)^2\frac{\delta}{2}\left(\sqrt{1+\frac{4\tilde{L}^2}{\gamma\delta}}+1\right)$$

de sorte d'avoir $\lambda\gamma/L^2+1 \geq \delta/\lambda'+1$. La condition (5.40) du théorème 16 s'écrit alors

$$\frac{\sqrt{1+(4\tilde{L}^2)/(\gamma\delta)}-1}{\sqrt{1+(4\tilde{L}^2)/(\gamma\delta)}+1} \leq \left(\frac{\tilde{L}}{L}\right)^2\frac{\sqrt{1+(4\tilde{L}^2)/(\gamma\delta)}-1}{\sqrt{1+(4\tilde{L}^2)/(\gamma\delta)}+1} \leq 1$$

On en déduit $\tilde{L}$ doit être compris entre L et $L\sqrt{1+\sqrt{\gamma\delta/L^2}}$. Sous cette condition, le théorème 16 assure que le taux de convergence ω est minoré par

$$\max\left\{\frac{\lambda'/\lambda+1}{\lambda\gamma/L^2+2}, \frac{1}{\delta/\lambda+1}\right\}$$

dont on montre qu'il vaut dans le cas présent $1/(\delta/\lambda+1)$. Ainsi, le meilleur taux est donné par

$$\omega^* = \frac{\sqrt{1+(4\tilde{L}^2)/(\gamma\delta)}-1}{\sqrt{1+(4\tilde{L}^2)/(\gamma\delta)}+1} \qquad \text{avec} \qquad 1-\frac{\omega^*}{\theta} = 1-\left(\frac{L}{\tilde{L}}\right)^2 \leq \frac{\sqrt{\gamma\delta/L^2}}{1+\sqrt{\gamma\delta/L^2}}$$

et on a pour tout $N \geq 1$

$$\begin{aligned} 0 \leq \frac{1-(L/\tilde{L})^2}{2\lambda}\|\xi-\xi_N\|^2 + \frac{\lambda'}{2}\|y-y_N\|^2 + \frac{\omega^*(1-(\omega^*)^N)}{1-\omega^*}\left(\tilde{\mathcal{L}}(\Xi_N,y)-\tilde{\mathcal{L}}(\xi,Y_N)\right) \\ \leq (\omega^*)^N\left(\frac{1}{2\lambda}\|\xi-\xi_0\|^2+\frac{\lambda'}{2}\|y-y_0\|^2\right). \end{aligned}$$

5.4 Exemples numériques

5.4.1 Algorithme FISTA

On peut comparer les taux obtenus pour l'ADMM modifié avec ceux obtenus pour un autre algorithme de descente proximale accéléré, proposé par Nesterov [11] et Beck et Teboulle [2]. Il s'agit de considérer la forme sans contrainte du problème primal, donnée par

$$\min_{x\in X}\Big\{F(Kx)+G(x)\Big\}$$

et de le résoudre par méthode de gradient explicite-implicite (voir chapitre 1), en ajoutant un pas de sur-relaxation :

$$x_0=\bar{x}_0\in X \qquad \text{et} \qquad \begin{cases} x_{k+1}=\operatorname{prox}_{\tau G}\Big(\bar{x}_k-\tau\,K^*\nabla\Big(F(K\bar{x}_k)\Big)\Big)\\ \bar{x}_{k+1}=x_{k+1}+\beta_{k+1}\,(x_{k+1}-x_k)\end{cases}$$

où les paramètres variables de la sur-relaxation sont donnés par

$$t_{k+1}=\frac{1-q\,{t_k}^2+\sqrt{(1-q\,{t_k}^2)^2+4{t_k}^2}}{2} \qquad \text{et} \qquad \beta_k=(1+\tau\,\gamma-t_{k+1}\tau\gamma)\,\frac{t_k-1}{t_{k+1}}$$

avec $q=\tau\gamma/(1+\tau\gamma)$, pour $\tau\in\left]0\,;\delta/L^2\right]$. Dans le cas où F^* et G sont supposées fortement convexes, de modules respectifs δ et γ, on peut montrer [6] que la convergence de cet algorithme est linéaire. Lorsque les pas sont choisis constants ($t_k=t=1/\sqrt{\tau\gamma/(1+\tau\gamma)}$), ce taux vaut alors $1-\sqrt{q}$ [6, Remark B.2], ce qui conduit au taux minimal

$$\omega^*_{\text{FISTA}}=1-\sqrt{\frac{\tau\gamma}{1+\tau\gamma}}=1-\sqrt{\frac{\gamma\delta/L^2}{1+\gamma\delta/L^2}}=1-\sqrt{\frac{1}{1+\kappa}}$$

et on a la majoration suivante pour tout $k\in\mathbb{N}$

$$E(x_k)-E(x^k)\leq(\omega^*_{\text{FISTA}})^k\left(E(x_0)-E(x^k)+\gamma\,\frac{\|x_0-x^*\|^2}{2}\right).$$

L'intérêt de cet algorithme réside dans le cas où ∇F et prox_G sont calculables.

5.4.2 Comparaison théorique avec FISTA et PDHG

On peut comparer les taux théoriques obtenus pour chacun des quatre algorithmes considérés dans cette section, qui sont l'ADMM, l'ADMM modifié, PDHG sur-relaxé et FISTA (cas fortement convexe). Ainsi, la figure 5.1 affiche la valeur des taux théoriques minimaux en fonction du conditionnement. Ainsi, il apparaît que, comme prévu, l'ADMM classique affiche le taux le moins bon, que l'ADMM modifié et PDHG offrent le même taux, qui est légèrement moins bon que celui de FISTA. Tous les taux tendent vers 1 lorsque le problème est mal conditionné.

5.4.3 Premier test quadratique

On commence dans ce paragraphe par comparer numériquement les performances de l'ADMM et de l'ADMM modifié sur un exemple simple.

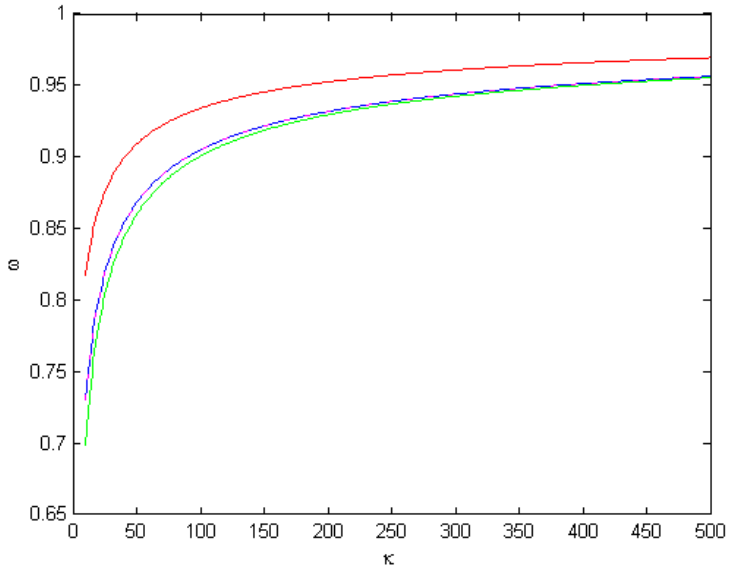


FIGURE 5.1 – Comparaison des taux de convergence ω théoriques en fonction du conditionnement κ, pour l'ADMM classique (rouge), l'ADMM modifié (magenta), PDHG sur-relaxé (bleu) et FISTA (vert). L'ordonnée sont en échelle logarithmique.

Problème considéré Soit N un entier naturel. Considérons le problème suivant sous contrainte égalité :

$$\min_{\substack{x=(x_i)_{i\in[\![0\,;\,N-1]\!]}\in\mathbb{R}^N \\ x_0=1}} \left\{ \frac{M-m}{2} \|K_N x\|^2 + \frac{m}{2} \|x\|^2 \right\} \tag{5.49}$$

où l'opérateur linéaire $K_N : \mathbb{R}^N \to \mathbb{R}^{N-1}$ est défini par $(K_N x)_i = (x_{i+1} - x_i)/2$ pour tout $i \in [\![0\,;\,N-2]\!]$, avec $\|K_N\| \leq 1$. Le conditionnement de ce problème vaut M/m, de sorte que, si l'on choisit M très grand devant m, alors ce problème est mal conditionné. On peut donc poser $F(z) = (M-m)\|z\|^2/2$ pour tout $z \in \mathbb{R}^{N-1}$ et $G(x) = m\|x\|^2/2 + \chi_{\{1\}}(x_0)$ pour tout $x = (x_i)_{i\in[\![0\,;\,N-1]\!]} \in \mathbb{R}^N$. Ces deux fonctions sont fortement convexes, de modules respectifs $M-m$ et $\gamma = m$. On vérifie que $F^*(y) = \|y\|^2/(M-m)/2$, qui est fortement convexe, de module $\delta = 1/(M-m)$. Appliquons à ce problème les deux versions de l'ADMM présentées dans cette section.

Résolution explicite Le minimiseur du problème (5.49) peut être calculé de manière explicite. Il suffit pour cela d'introduire le sous-vecteur $\tilde{x}$ de x, défini comme suit :

$$\forall\, i \in [\![0\,;\,N-2]\!], \qquad \tilde{x}_i = x_{i+1}$$

de sorte que $x = (1,\tilde{x})$. Le problème sous contrainte (5.49) peut alors s'écrire sous la forme non contrainte

$$\min_{\tilde{x}=(\tilde{x}_i)_{i\in[\![0\,;\,N-2]\!]}\in\mathbb{R}^{N-1}} \left\{ \frac{M-m}{2} \left(\|K_{N-1}\tilde{x}\|^2 + \frac{(\tilde{x}_0-1)^2}{4} \right) + \frac{m}{2} \left(\|\tilde{x}\|^2 + 1 \right) \right\}$$

en remarquant que $(\tilde{K_N x}) = K_{N-1}\tilde{x}$. Le minimiseur $\tilde{x}^*$ de ce problème fortement convexe est donné par l'équation d'EULER, qui assure que $\tilde{x}^* = A^{-1}b$, avec

$$A = m\,\mathrm{I}_{N-1} + (M-m)\,{}^{\mathrm{t}}K_{N-1}K_{N-1} + \frac{M-m}{4}\,e_{0,0}$$

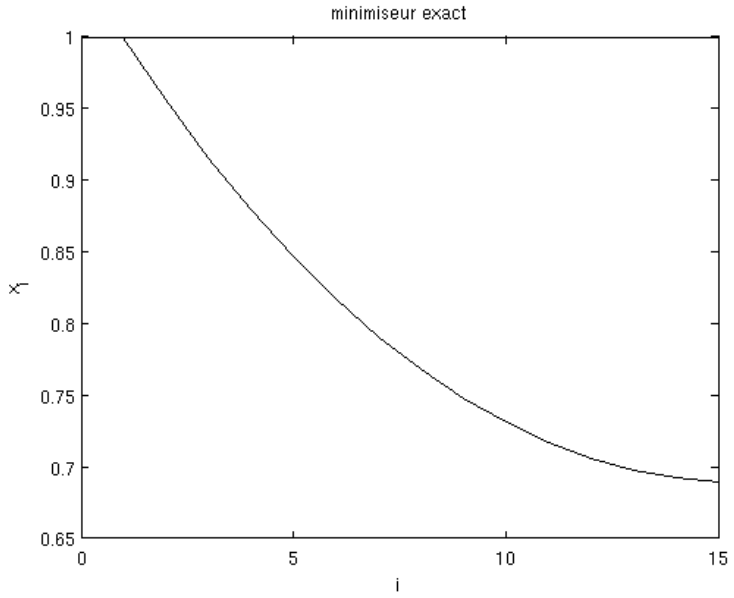


FIGURE 5.2 – Minimiseur théorique du problème (5.49).

avec $e_{0,0}$ la matrice carrée de taille $N-1$ ayant tous ses coefficients nuls à l'exception du coefficient d'indice (0,0), qui vaut 1, et $b = (M-m)\,e_0/4$, où e_0 est le premier vecteur de la base canonique de $\mathbb{R}^{N-1}$. Le minimiseur du problème initial (5.49) est alors donné par $x^* = (1,\tilde{x}^*)$. Pour $M = 1000$ et $m = 1$, la figure 5.2 affiche l'allure de x^*.

Application de l'ADMM L'ADMM modifié s'écrit pour ce problème

$$\begin{cases} x_{n+1} = \underset{\substack{x=(x_i)_i\in\mathbb{R}^N \\ x_0=1}}{\operatorname{argmin}} \left\{ \dfrac{m}{2}\,\|x\|^2 + \langle Kx, y_n\rangle + \dfrac{1}{2\,\lambda}\,\|Kx - z_n\|^2 \right\} \\ z_{n+1} = \underset{z\in\mathbb{R}^{N-1}}{\operatorname{argmin}} \left\{ \dfrac{M-m}{2}\,\|z\|^2 - \langle z, y_n\rangle + \dfrac{1}{2\,\lambda'}\,\|Kx_{n+1} - z\|^2 \right\} \\ y_{n+1} = y_n + \dfrac{1}{\lambda'}\,(Kx_{n+1} - z_{n+1}). \end{cases}$$

La mise-à-jour de z s'écrit grâce à l'équation d'EULER

$$z_{n+1} = \frac{y_n + Kx_{n+1}/\lambda'}{M - m + 1/\lambda'}.$$

Pour calculer explicitement la mise-à-jour de x, on utilise à nouveau les sous-vecteurs introduits au paragraphe précédent. Le problème à résoudre s'écrit alors sans contrainte (en ignorant les termes constants)

$$\min_{\tilde{x}=(\tilde{x}_i)_i\in\mathbb{R}^{N-1}} \left\{ \frac{m}{2}\,\|\tilde{x}\|^2 + \langle \tilde{x}, {}^{\mathrm{t}}K_{N-1}\tilde{y}_n\rangle + \frac{1}{2\,\lambda}\left(\|K_{N-1}\tilde{x} - \tilde{z}_n\|^2 + \left(\frac{\tilde{x}_0 - 1}{2} - (z_n)_0 \right)^2 \right) \right\}.$$

L'équation d'EULER assure cette fois que $\tilde{x}_{n+1} = A_n^{-1} b_n$, avec

$$A_n = m\,\mathrm{I}_{N-1} + \frac{1}{\lambda}\,{}^{\mathrm{t}}K_{N-1}K_{N-1} + \frac{1}{4\lambda}\,e_{0,0}$$

et $$b_n = -\,{}^{\mathrm{t}}K_{N-1}\tilde{y}_n + \frac{1}{\lambda}\,{}^{\mathrm{t}}K_{N-1}\tilde{z}_n + \left(-\frac{(y_n)_0}{2} + \frac{1}{2\lambda}\,(z_n)_0 + \frac{1}{4\lambda} \right) e_0.$$

On a alors $x_{n+1} = (1,\tilde{x}_{n+1})$.

Choix des paramètres On teste dans cette expérience deux jeux de paramètres différents pour λ et λ'. Le premier (désigné sous le nom d'ADMM classique) correspond au paramètre optimal pour l'ADMM classique déterminé dans cette section, qui vaut dans ce cas

$$\lambda = \lambda' = \sqrt{\frac{2L^2}{m(M-m)}}.$$

On prendra $L = 1$. Le second jeu de paramètres (ADMM modifié) correspond aux paramètres optimaux dans l'ADMM modifié, conduisant au meilleur taux de convergence :

$$\lambda = \frac{1}{2(M-m)}\left(\sqrt{1+\frac{4L^2(M-m)}{m}}+1\right)$$

et

$$\lambda' = \lambda - \frac{1}{M-m} = \frac{1}{2(M-m)}\left(\sqrt{1+\frac{4L^2(M-m)}{m}}-1\right).$$

Mesure de la convergence Pour comparer la convergence des trois jeux de paramètres, on utilisera trois outils :

1. la distance quadratique au minimiseur $\|x_n - x^*\|^2$;
2. la différence absolue entre l'énergie à l'itération n et l'énergie minimale ;
3. un *primal-dual gap*.

Concernant le dernier outil, le *gap* qui devrait être pris en compte est celui correspondant au problème primal-dual résolu, qui est la recherche de point-selle du lagrangien augmenté associé au problème initial (5.49). Malheureusement, l'énergie primale donnée par

$$\sup_{y\in\mathbb{R}^{N-1}}\left\{\frac{m}{2}\|x_n\|^2 + \langle Kx_n - z_n, y_n\rangle + \frac{M-m}{2}\|z_n\|^2 + \frac{1}{2\,\lambda'}\|Kx_n - z_n\|^2\right\}$$

est généralement infinie, car Kx_n n'est pas contraint à être égal à z_n. De ce fait, utiliser ce *gap* n'est pas pertinent. C'est pourquoi on se propose ici d'utiliser la conjuguée convexe de $x \mapsto F(Kx)$, ce qui permet de réécrire le problème primal (5.49) sous la forme primale-duale

$$\min_{\substack{x=(x_i)_{i\in[\![0\,;N-1]\!]}\in\mathbb{R}^N \\ x_0=1}} \sup_{z'\in\mathbb{R}^{N-1}}\left\{\frac{m}{2}\|x\|^2 + \langle Kx, z'\rangle - \frac{1}{2(M-m)}\|z'\|^2\right\}$$

Ce problème admet au moins une solution, donnée par $(x^*,(z')^*)$, avec

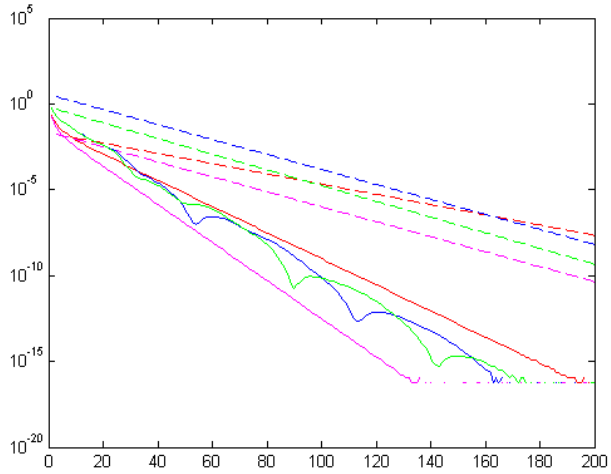
$$(z')^* = \operatorname*{argmax}_{z\in\mathbb{R}^{N-1}}\left\{\frac{m}{2}\|x^*\|^2 + \langle Kx^*, z'\rangle - \frac{1}{2(M-m)}\|z'\|^2\right\} = (M-m)\,Kx^*$$

(on résout l'équation d'Euler). Le *gap* pour ce problème vaut

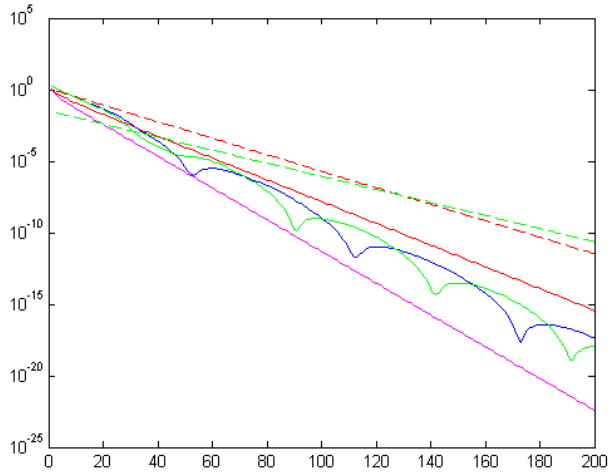
$$\mathcal{G}(x_n, z'_n) = \mathcal{P}(x_n) - \mathcal{D}(z_n)$$

où l'énergie primale en x_n vaut $\mathcal{P}(x_n) = m\,\|x_n\|^2/2 + (M-m)\,\|Kx_n\|^2/2$, tandis que l'énergie duale en z'_n vaut

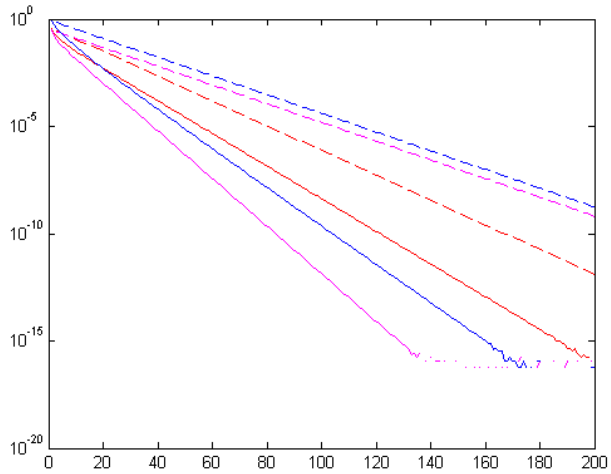
$$\mathcal{D}'(z'_n) = \frac{m}{2}\|x'\|^2 + \langle Kx', z'_n\rangle - \frac{1}{2(M-m)}\|z'_n\|^2$$



(a) Énergie $E(x_n) - E(x^*)$



(b) Distance au minimiseur $\|x_n - x^*\|^2$



(c) *Primal-dual gap*

FIGURE 5.3 – Mesure de la convergence pour l'ADMM (rouge), l'ADMM modifié (magenta), PDHG sur-relaxé (bleu) et FISTA (vert) pour le problème quadratique. En trait plein, les mesures empiriques, en pointillé, les valeurs théoriques (décroissance linéaire en ω pour l'énergie et le *gap* et en $\tilde{\omega}$ pour la convergence de la variable primale, sauf pour FISTA). Les ordonnées sont en échelle logarithmique.

avec $\tilde{x}' = -{}^{\mathrm{t}}K_{N-1} z'_n/m$. On sait que le *gap* $\mathcal{G}(x_n; z'_n)$ doit tendre vers 0 si $(x_n)_n$ tend vers x^* et $(z'_n)_n$ tend vers $(M-m)\,Kx^* = (M-m)\,z^*$. On en déduit en particulier que la suite $(\mathcal{G}(x_n; (M-m)\,z_n))_n$ doit tendre vers 0. C'est ce *gap*-là que l'on considèrera dans les expériences réalisées.

La figure 5.3 présente l'évolution des trois mesures décrites plus haut, pour un même nombre d'itérations. On affiche par ailleurs l'évolution de ces mêmes mesures obtenues lorsque l'on applique le PDHG sur-relaxé avec les paramètres optimaux déterminés au paragraphe 5.2.3 et lorsque l'on applique FISTA (cas fortement convexe) avec des pas constants (voir paragraphe précédent). Dans ce dernier cas, on ne considère que les deux premières quantités (le *gap* étant mal défini).

5.4.4 Application au débruitage TV-Huber

Problème considéré Soit $u^h \in \mathbb{R}^{3N_xN_y}$ une image couleur RGB (éventuellement bruitée). On cherche à résoudre

$$\min_{v^h \in \mathbb{R}^{3N_xN_y}} \left\{ \frac{1}{2\mu} \|v^h - u^h\|^2 + \frac{1}{2} \mathrm{TV}^h_{\mathrm{Huber}}(v^h) \right\} \tag{5.50}$$

où l'opérateur linéaire gradient $\nabla^h : \mathbb{R}^{3N_xN_y} \to \mathbb{R}^{3N_xN_y} \times \mathbb{R}^{3N_xN_y}$ est défini pour toute image v^h par $\nabla^h v^h = {}^{\mathrm{t}}(\delta^h_x v^h, \delta^h_y v^h)$, où les différences finies sont données pour tout indice $(i,j) \in [\![\, 0\,;\, N_x - 1 \,]\!] \times [\![\, 0\,;\, N_y - 1 \,]\!]$ par

$$(\delta^h_x v^h)_{i,j} = \begin{cases} v^h_{i+1,j} - v^h_{i,j} & \text{si } i < N_x - 1 \\ 0 & \text{sinon} \end{cases} \quad \text{et} \quad (\delta^h_y v^h)_{i,j} = \begin{cases} v^h_{i,j+1} - v^h_{i,j} & \text{si } j < N_y - 1 \\ 0 & \text{sinon} \end{cases}$$

et avec $\mathrm{TV}^h_{\mathrm{Huber}}(v^h)$ défini par

$$\mathrm{TV}^h_{\mathrm{Huber}}(v^h) = \sum_{i=0}^{N_x-1} \sum_{j=0}^{N_y-1} h\Big(\|(\nabla^h v^h)i,j\|\Big) \qquad \text{où} \qquad h(t) = \begin{cases} \dfrac{|t|^2}{2} & \text{si } |t| \leq 1 \\ |t| - \dfrac{1}{2} & \text{si } |t| > 1. \end{cases}$$

Ce problème peut être interprété comme un problème de débruitage, avec un terme d'attache aux données quadratique et un terme de régularisation défini par une version régularisée de TV (voir figure 5.4), qui agit comme une régularisation quadratique lorsque les variations sont faibles et comme la régularisation TV sinon. Ce modèle peut donc être rapproché du modèle ROF étudié plus en détail dans le chapitre suivant. Le paramètre $\mu > 0$ permet de pondérer l'importance relative du terme d'attache aux données par rapport au terme de régularisation.

Application de l'ADMM Posons $G = \|\cdot - u^h\|^2/(2\mu)$ et

$$F(\phi) = \sum_{i=0}^{N_x-1} \sum_{j=0}^{N_y-1} h\Big(\|\phi_{i,j}\|\Big) \qquad \text{avec} \qquad F^*(\xi) = \sum_{i=0}^{N_x-1} \sum_{j=0}^{N_y-1} \frac{1}{2} \|\xi_{i,j}\|^2 + \chi_{[\,0\,;1\,]}(\|\xi_{i,j}\|).$$

L'ADMM modifié s'écrit pour ce problème

$$\begin{cases} v^h_{n+1} = \displaystyle\operatorname*{argmin}_{v^h \in \mathbb{R}^{3N_xN_y}} \left\{ \frac{1}{2\mu} \|v^h - u^h\|^2 + \langle \nabla^h v^h, \xi^h_n \rangle + \frac{1}{2\,\lambda} \|\nabla^h v^h - \phi^h_n\|^2 \right\} \\ \phi^h_{n+1} = \displaystyle\operatorname*{argmin}_{\phi^h \in (\mathbb{R}^{3N_xN_y})^2} \left\{ F(\phi^h) - \langle \phi^h, \xi^h_n \rangle + \frac{1}{2\,\lambda'} \|\nabla^h v^h_{n+1} - \phi^h\|^2 \right\} \\ \xi^h_{n+1} = \xi^h_n + \dfrac{1}{\lambda'} (\nabla^h v^h_{n+1} - \phi^h_{n+1}). \end{cases}$$

(a) Image originale u^h (b) Image bruitée g^h (c) Image débruitée v^*

FIGURE 5.4 – Image originale u^h (à gauche), image bruitée g^h (au milieu) et image débruitée v^* (à droite). Le bruit ajouté est un bruit blanc gaussien additif, de variance 100 (les images sont à valeurs entre 0 et 255). L'image débruitée v^* est obtenue en appliquant une régularisation quadratique avec une attache aux données quadratique. Source : détail de l'image *Hepatica nobilis flowers*, par Archenzo.

Chacune des deux minimisations partielles se résolvent grâce à l'équation d'EULER, qui assure que la mise-à-jour de l'image v^h est calculée par

$$v_{n+1}^h = \left(\frac{1}{\mu}\,\mathrm{I} + \frac{1}{\lambda}\,(\nabla^h)^*\nabla^h\right)^{-1}\left(\frac{1}{\mu}\,u^h + \frac{1}{\lambda}\,(\nabla^h)^*\phi_n^h - (\nabla^h)^*\xi_n^h\right)$$

tandis que la mise-à-jour de la seconde variable primale ϕ^h est donnée par

$$\phi_{n+1}^h = \frac{\tau'(\xi_n)_{i,j} + (\nabla v_{n+1})_{i,j}}{|\tau'(\xi_n)_{i,j} + (\nabla v_{n+1})_{i,j}|}\,|(\phi_{n+1})_{i,j}|$$

avec

$$|(\phi_{n+1})_{i,j}| = \begin{cases} \dfrac{\tau'|(\xi_n)_{i,j} + (\nabla v_{n+1})_{i,j}|}{\tau'+1} & \text{si } |\tau'(\xi_n)_{i,j} + (\nabla v_{n+1})_{i,j}| \leq \tau'+1 \\ |\tau'(\xi_n)_{i,j} + (\nabla v_{n+1})_{i,j}| - \tau' & \text{si } |\tau'(\xi_n)_{i,j} + (\nabla v_{n+1})_{i,j}| > \tau'+1. \end{cases} \tag{5.51}$$

Choix des paramètres On va à nouveau tester les deux jeux de paramètres (λ,λ') correspondant à l'ADMM classique et à l'ADMM modifié.

Commençons par déterminer la régularité du problème. Les fonctions F^* et G sont fortement convexes, de modules respectifs $\delta = 1$ et $\gamma = 1/\mu$. L'opérateur gradient est borné, de norme $L \leq 2\sqrt{2}$, avec L tendant vers cette borne supérieure lorsque N_x ou N_y tendent vers $+\infty$.

Les trois jeux de paramètres sont donc donnés par

1. ADMM classique : $\lambda = \lambda' = 4\sqrt{\mu}$;
2. ADMM modifié :

$$(\lambda,\lambda') = \left(\frac{1}{2}\left(\sqrt{1+\frac{32}{\mu}}+1\right), \frac{1}{2}\left(\sqrt{1+\frac{32}{\mu}}-1\right)\right).$$

Mesure de la convergence Pour mesurer la convergence, on utilise cette fois deux outils distincts : la différence entre l'énergie primale et son minimum et la distance quadratique en v_n^h et le minimiseur v^* du problème. La première mesure dépendant de la valeur de la solution v^*, cette dernière est obtenue expérimentalement en faisant tourner sur un temps long la méthode PDHG sur-relaxé.

Résultats expérimentaux La figure 5.4 présente l'image initiale bruitée u^h ainsi que le débruitage v^* réalisé en résolvant le problème (5.50). On a choisi $\mu = 2$ pour cette expérience. On compare ensuite les performances de l'ADMM classique, l'ADMM modifié, le PDHG sur-relaxé et FISTA. La figure 5.5 présente quant à elle la convergence de l'énergie et celle de la variable primale pour chaque expérience. Dans chaque cas, on affiche également la courbe de décroissance théorique, qui, en échelle logarithmique, est une droite de pente $\ln(\tilde{\omega})$ pour la distance au minimiseur.

5.4.5 Discussion

ADMM classique *versus* ADMM modifié Les deux expériences présentées aux paragraphes précédents montrent dans les deux cas que la version modifiée de l'ADMM proposée dans ce chapitre conduit à de meilleures performances que l'ADMM classique. Pour une comparaison juste, on a choisi pour l'ADMM classique et une des versions de l'ADMM modifié les paramètres conduisant théoriquement au meilleur taux de convergence. On observe alors que, tant pour l'énergie primale et pour le *primal-dual gap*, l'introduction d'un paramètre $\lambda' \neq \lambda$ permet effectivement d'accélérer la décroissance.

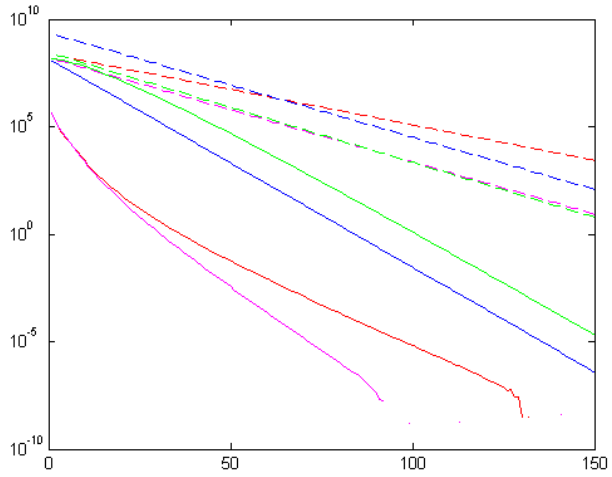
Comparaison avec les taux théoriques Dans le premier exemple du problème quadratique, les performances observées empiriquement sont bien meilleures que celles attendues (qui sont affichées en pointillés). Cela peut expliquer par le fait que le problème est beaucoup plus régulier que ceux concernés par les différentes preuves de convergence démontrées dans ce chapitre.

Dans la régularisation TV-Huber, on observe pour l'ADMM, l'ADMM modifié et PDHG une bonne corrélation entre les courbes théoriques et les courbes empiriques dans le cas de la convergence de la variable primale (même si cette convergence n'a pas été formellement démontrée dans le cas des ADMM). En revanche, il apparaît que la convergence de l'énergie et du *gap* soit meilleure en pratique qu'en théorie, puisqu'elle semble afficher le même taux que celle de la variable primale.

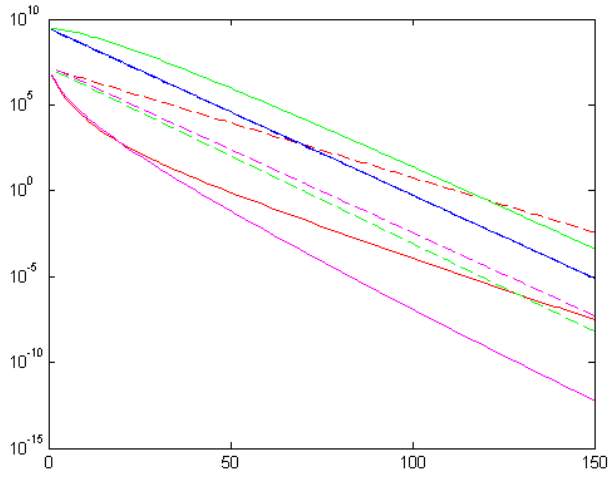
Oscillations de FISTA et de PDHG accéléré Pour FISTA et PDHG sur-relaxé, on observe des oscillations dans le problème quadratique. Dans le cas de FISTA, ce phénomène a déjà été observé [12] pour des problèmes similaires. Ces oscillations sont dues à l'étape de sur-relaxation, et apparaissent lorsque θ est choisi trop grand en comparaison des valeurs propres de l'opérateur $m\,\mathrm{I}_N + (M-m)K_N^*K_N$.

Dans le cas de l'algorithme primal-dual, de telles oscillations n'ont pas été rapportées à notre connaissance. Il est vraisemblable qu'elles soient également dues à la sur-relaxation, pour des raisons similaires à celles intervenant dans FISTA. Cependant, à l'heure où ce manuscrit est rédigé, l'étude théorique de ce phénomène n'a pas encore abouti.

Mis à part l'apparition d'oscillations dans le cas quadratique mal conditionné, qui ralentissent localement la convergence des algorithmes PDHG et FISTA, il doit être noté que ces deux algorithmes ne font pas intervenir l'inversion d'une matrice (qui, dans le cas du débruitage TV-Huber, peut s'avérer être d'une taille très importante). Ainsi, une itération dans ces deux algorithmes est bien moins coûteuse en temps de calculs. En d'autres termes, même si la convergence en termes de nombre d'itérations peut être meilleure pour l'ADMM modifié, celle est très inférieure lorsqu'elle est mesurée en temps total de calculs.



(a) Énergie $E(x_n) - E(x^*)$



(b) Distance au minimiseur $\|x_n - x^*\|^2$

FIGURE 5.5 – Mesure de la convergence pour l'ADMM (rouge), l'ADMM modifié (magenta), PDHG sur-relaxé (bleu) et FISTA (vert) pour le débruitage TV-HUBER. En trait plein, les mesures empiriques, en pointillé, les valeurs théoriques (décroissance linéaire en ω pour l'énergie et en $\tilde{\omega}$ pour la convergence de la variable primale). Les ordonnées sont en échelle logarithmique.

Conclusion

On a proposé dans ce chapitre une version modifiée de l'algorithme ADMM en relaxant légèrement le choix du paramètre de la version classique de l'algorithme, en introduisant un second paramètre. Dans le cas fortement convexe, on a démontré, en écrivant les itérations de l'ADMM sous la forme d'un algorithme primal-dual connu sous le nom de PDHG, la convergence de ce nouvel algorithme, sous certaines conditions sur les deux paramètres.

L'introduction de ce second paramètre, s'il est correctement choisi, conduit à une meilleure convergence théorique que l'ADMM classique. On a déterminé pour ces deux versions les paramètres conduisant au meilleur taux de convergence théorique, ainsi que, pour la version modifiée, des paramètres associés à un taux sous-optimal mais permettant de conserver un contrôle sur la convergence des deux variables.

Des tests expérimentaux, dont l'un sur le débruitage, confirment qu'en pratique, la version modifiée de l'ADMM présente de meilleures performances que la version classique. Dans un cas simple mais mal-conditionné, elle affiche par ailleurs un meilleur comportement que deux autres algorithmes accélérés classiques, qui sont l'algorithme primal-dual PDHG sur-relaxé et FISTA (accéléré dans le cas fortement convexe). Ces deux derniers présentent en effet des oscillations qui dégradent leur vitesse de convergence. Néanmoins, dans des cas moins pathologiques, même si les deux versions d'ADMM étudiées offrent des convergences comparables à ces deux algorithmes en termes de nombre d'itérations, l'inversion d'un opérateur nécessaire pour l'ADMM le rend moins compétitif lorsque la convergence est mesurée en termes de temps de calcul.

Références

[1] Hedy Attouch and Mohamed Soueycatt. Augmented lagrangian and proximal alternating direction methods of multipliers in Hilbert spaces. applications to games, PDE's and control. *Pacific Journal of Optimization*, 5(1) :17–37, 2008.

[2] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1) :183–202, 2009.

[3] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1) :1–122, 2011.

[4] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1) :120–145, 2011.

[5] Antonin Chambolle and Thomas Pock. On the ergodic convergence rates of a first-order primal–dual algorithm. *Mathematical Programming*, pages 1–35, 2015.

[6] Antonin Chambolle and Thomas Pock. An introduction to continuous optimization for imaging. *Acta Numerica*, 25 :161–319, 5 2016.

[7] Patrick L. Combettes and Jean-Christophe Pesquet. Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer, 2011.

[8] Ernie Esser, Xiaoqun Zhang, and Tony Chan. A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. *SIAM Journal on Imaging Sciences*, 3(4) :1015–1046, 2010.

[9] Daniel Gabay and Bertrand Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1) :17–40, 1976.

[10] Roland Glowinski and A. Marrocco. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires. *Revue Française d'Automatique, Informatique, Recherche Opérationnelle. Analyse Numérique*, 9(2) :41–76, 1975.

[11] Yurii Nesterov. *Introductory lectures on convex optimization : A basic course*, volume 87. Springer Science & Business Media, 2013.

[12] Brendan O'Donoghue and Emmanuel Candes. Adaptive restart for accelerated gradient schemes. *Foundations of computational mathematics*, 15(3) :715–732, 2015.

[13] Thomas Pock, Daniel Cremers, Horst Bischof, and Antonin Chambolle. An algorithm for minimizing the Mumford-Shah functional. In *IEEE International Conference on Computer Vision*, pages 1133–1140. IEEE, 2009.

[14] Mingqiang Zhu and Tony Chan. An efficient primal-dual hybrid gradient algorithm for total variation image restoration. *UCLA CAM Report*, pages 08–34, 2008.

Chapitre 6

Alterner les descentes proximales : application au modèle ROF

Introduction

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

Parmi les méthodes les plus connues de débruitage d'images, la méthode ROF repose sur une formulation variationnelle du problème. Elle consiste à définir une fonctionnelle d'énergie à deux termes, dont l'un est la norme TV du signal. Ce modèle est populaire car la norme TV est une bonne norme pour les images, en ce sens que les images naturelles ont généralement une faible norme TV. Ainsi, rechercher le signal le plus proche du signal observé et de norme TV la plus faible possible est une manière naturelle de débruiter. L'intérêt principal de l'utilisation de la norme TV est qu'elle préserve les

discontinuités des images, ce que ne parviennent pas toujours à faire les autres méthodes de débruitage. Malheureusement, le débruitage d'une image est généralement une étape préliminaire à un autre traitement, et ne constitue pas une fin en soi. Or, la minimisation de la fonctionnelle ROF n'est pas toujours très efficace. Il a été montré que ce problème se résout en $O(N)$ pour des signaux 1D, mais de tels algorithmes ne sont pas généralisables pour des dimensions supérieures. Néanmoins, des stratégies dites d'éclatement ont été introduites pour exploiter cette efficacité du cas 1D dans le problème 2D (qui est le cas des images en niveaux de gris). Ces algorithmes sont en particulier parallélisables, car ils décomposent le problème 2D en plusieurs problèmes 1D indépendants.

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

6.1 Le modèle Rudin-Osher-Fatemi

6.1.1 Position du problème

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

$$g = u + n$$

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

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

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

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

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

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

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

6.1.2 Cas 1D : complexité en $O(N)$

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

Problème discret La version discrète du modèle ROF est donnée par

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

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

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

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

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

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

Algorithme de Condat Proposé en 2013 par Laurent Condat [2], cette méthode repose sur la caractérisation des *sur-solutions* et *sous-solutions* du problème (6.3), c'est-à-dire des majorants et des minorants de la solution, qui satisfont un certain nombre de conditions. Celles-ci découlent de l'équation d'Euler associé au problème étudié. En faisant des hypothèses de constance locale de la solution, on construit progressivement des sur- et sous-solutions, jusqu'à ce qu'une impossibilité apparaisse (typiquement, la sur-solution passant sous la solution ou la sous-solution passant au-dessus de la solution). On peut alors identifier le point jusqu'auquel les hypothèses sont correctes, puis reprendre l'estimation de la solution à partir de ce point. La complexité de cet algorithme est de $O(N)$ dans la majorité des cas, mais est moins bonne dans le pire des cas.

Algorithme de Johnson L'algorithme proposé par Nicholas Johnson en 2010 [5] repose quant à lui sur la programmation dynamique. Le principe est de transformer le problème global (6.3) en une série de problèmes locaux (définis sur les i premiers coefficients du vecteur). L'idée est de considérer l'estimation d'un vecteur de i points comme la minimisation d'une énergie à trois termes : l'énergie du dernier point i, l'énergie du vecteur privé de ce point, et l'énergie d'interaction entre ces deux composantes du vecteur initial, qui ne dépendant que du i-ème et du $i-1$-ème point. Si l'énergie optimale du vecteur composé des $i-1$ premiers points peut être rapidement calculée quelle que soit la configuration testée, alors le problème est simplifié, car il s'agit uniquement de trouver le i-ème point, les $i-1$ premiers points étant alors pris optimaux pour ce choix. Johnson montre que ce schéma de résolution repose en pratique entièrement sur la recherche du zéro d'une fonction affine par morceaux qui est progressivement mise à jour. Or, l'encodage d'une telle fonction est peu coûteuse, car il s'agit uniquement d'en stocker les points de rupture. La complexité de l'algorithme est en $O(N)$.

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

6.1.3 Cas 2D : stratégies d'éclatement

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

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

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

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

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

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

où les différences finies sont données par

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

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

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

$$\mathrm{TV}^h_{\mathrm{iso}}(v^h) = \sum_{i=0}^{N_x-1} \sum_{j=0}^{N_y-1} \sqrt{(\delta^h_x v^h)^2_{i,j} + (\delta^h_y v^h)^2_{i,j}}$$

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

$$\mathrm{TV}^h_{\mathrm{aniso}}(v^h) = \sum_{i=0}^{N_x-1} \sum_{j=0}^{N_y-1} |(\delta^h_x v^h)_{i,j}| + |(\delta^h_y v^h)_{i,j}|.$$

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

$$\min_{v^h \in \mathbb{R}^{N_x \times N_y}} \left\{ \frac{1}{2\lambda} \|v^h - g^h\|_2^2 + \mathrm{TV}^h_{\mathrm{h}}(v^h) + \mathrm{TV}^h_{\mathrm{v}}(v^h) \right\} \tag{6.5}$$

où on pose $\mathrm{TV}^h_{\mathrm{h}}(v^h) = \|\delta^h_x v^h\|_1$ et $\mathrm{TV}^h_{\mathrm{v}}(v^h) = \|\delta^h_y v^h\|_1$. Le lagrangien de ce problème est composé de trois termes, un terme différentiable et deux termes dont l'opérateur proximal peut être évalué, puisqu'il ne s'agit ni plus ni moins que de normes TV 1D (horizontale et verticale). On peut donc appliquer la méthode d'éclatement de Dykstra. Il s'agit de considérer le problème dual équivalent

$$\min_{x^h, y^h \in \mathbb{R}^{N_x \times N_y}} \left\{ (\mathrm{TV}^h_{\mathrm{h}})^*(y^h) + (\mathrm{TV}^h_{\mathrm{v}})^*(x^h) + \frac{1}{2\lambda} \left\| \lambda(x^h + y^h) - g^h \right\|_2^2 \right\}. \tag{6.6}$$

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

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

On résout le problème dual par minimisations partielles alternées :

$$\begin{cases} x^h_{n+1} = \underset{x \in \mathbb{R}^{N_x \times N_y}}{\operatorname{argmin}} \left\{ (\mathrm{TV}^h_{\mathrm{h}})^*(x) + \frac{1}{2\lambda} \left\| \lambda(x + y^h_n) - g^h \right\|_2^2 \right\} \\ y^h_{n+1} = \underset{y \in \mathbb{R}^{N_x \times N_y}}{\operatorname{argmin}} \left\{ (\mathrm{TV}^h_{\mathrm{v}})^*(y) + \frac{1}{2\lambda} \left\| \lambda(x^h_{n+1} + y) - g^h \right\|_2^2 \right\}. \end{cases} \tag{6.7}$$

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

Éclatement carrés pairs/impairs Restons pour l'instant dans le cas de la TV anisotrope $k = 1$. Remarquons qu'il est possible de décomposer la norme TV en deux termes, l'un portant sur les carrés de coin supérieur gauche de coordonnées paires et l'autre portant sur les carrés de coin supérieur gauche de coordonnées impaires. Plus précisément, commençons par définir le carré de coin supérieur gauche (i,j) en posant

pour $(i,j) \in [\![-1\,;\, N_x - 1]\!] \times [\![-1\,;\, N_y - 1]\!]$ l'opérateur d'extraction $S_{i,j}$, qui est défini par

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

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

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

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

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

Enfin, dans les autres cas, on a

$$S_{-1,-1} v^h = (v^h_{0,0}), \qquad S_{N_x-1,-1} v^h = (v^h_{N_x-1,0}),$$

$$S_{-1,N_y-1} v^h = (v^h_{0,N_y-1}) \qquad \text{et} \qquad S_{N_x-1,N_y-1} v^h = (v^h_{N_x-1,N_y-1}).$$

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

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

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

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

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

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

$$\mathrm{TV}^h_{\mathrm{aniso}}(v^h) = \underbrace{\sum_{\substack{i=0 \\ i \text{ pair}}}^{N_x-1} \sum_{\substack{j=0 \\ j \text{ pair}}}^{N_y-1} \|D(Sv^h)_{i,j}\|_1}_{= \mathrm{TV}^h_{\mathrm{e}}(v^h)} + \underbrace{\sum_{\substack{i=-1 \\ i \text{ impair}}}^{N_x-1} \sum_{\substack{j=-1 \\ j \text{ impair}}}^{N_y-1} \|D(Sv^h)_{i,j}\|_1}_{= \mathrm{TV}^h_{\mathrm{o}}(v^h)}$$

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

$$\min_{x^h, y^h \in \mathbb{R}^{N_x \times N_y}} (\mathrm{TV}^h_{\mathrm{e}})^*(x^h) + (\mathrm{TV}^h_{\mathrm{o}})^*(y^h) + \frac{1}{2\lambda} \left\| \lambda(x^h + y^h) - g^h \right\|_2^2 \tag{6.8}$$

par minimisations partielles alternées :

$$\begin{cases} x^h_{n+1} = \displaystyle\operatorname*{argmin}_{x \in \mathbb{R}^{N_x \times N_y}} \left\{ (\mathrm{TV}^h_{\mathrm{e}})^*(x) + \frac{1}{2\lambda} \left\| \lambda(x + y^h_n) - g^h \right\|_2^2 \right\} \\ y^h_{n+1} = \displaystyle\operatorname*{argmin}_{y \in \mathbb{R}^{N_x \times N_y}} \left\{ (\mathrm{TV}^h_{\mathrm{o}})^*(y) + \frac{1}{2\lambda} \left\| \lambda(x^h_{n+1} + y) - g^h \right\|_2^2 \right\}. \end{cases} \tag{6.9}$$

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

$$(\mathrm{TV}^h_{\mathrm{e}})^*(x) = \sup_{v \in \mathbb{R}^{N_x \times N_y}} \left\{ \langle v, x \rangle - \sum_{\substack{i=0 \\ i \text{ pair}}}^{N_x} \sum_{\substack{j=0 \\ j \text{ pair}}}^{N_y} \|D(Sv)_{i,j}\|_1 \right\}.$$

On peut facilement vérifier que

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

Ainsi, pour calculer la conjuguée convexe $(\mathrm{TV}_{\mathrm{e}}^h)^*(x)$, il suffit de savoir résoudre

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

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

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

et

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

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

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

Puisque

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

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

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

et pour tout (i,j) impairs,

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

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

$$\min_{s\in\mathbb{R}^4} \Big\{ \chi_{\mathcal{K}}(s) + \frac{1}{2} \|s - s_0\|_2^2 \Big\} \qquad \text{avec } s_0 \in \mathbb{R}^4,$$

qui s'écrit aussi

$$\min_{\substack{\xi\in\mathbb{R}^4\\ \|\xi\|_\infty\leq 1}} \Big\{ \frac{1}{2} \|D^*\xi - s_0\|_2^2 \Big\}.$$

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

6.2 Théorie des descentes alternées multiples

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

6.2.1 Position du problème

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

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

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

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

6.2.2 Algorithme

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

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

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

où M et P sont deux matrices symétriques. Si M et P sont l'identité, alors cet algorithme alterne donc K descentes proximales en x puis L descentes proximales en y, avec des moyennages entre chaque ensemble de descentes. Dans le cas général, les descentes proximales sont remplacées par des itérations de Bregman (voir chapitre 1). L'intérêt

de cet algorithme se manifeste évidemment si les descentes proximales peuvent être évaluées de manière exacte. Nous allons montrer que cet algorithme converge bien vers la solution du problème (6.10).

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

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

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

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

On pose alors $\qquad \tilde{x} = \dfrac{1}{K}\sum_{k=1}^{K} \hat{x}_k \qquad$ et $\qquad \tilde{y} = \dfrac{1}{L}\sum_{\ell=1}^{L} \hat{y}_\ell.$

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

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

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

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

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

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

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

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

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

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

Le produit scalaire restant se décomposant des deux manières suivantes

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

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

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

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

Ainsi, en revenant à (6.13),

$$\begin{aligned} f(x) + \frac{1}{2}\,\|Ax + B\bar{y} - c\|^2 + \frac{1}{2}\,\|x - \hat{x}_k\|_M^2 \geq f(\hat{x}_{k+1}) + \frac{1}{2}\,\|A\hat{x}_{k+1} + B\bar{y} - c\|^2 \\ + \frac{1}{2}\,\|x - \hat{x}_{k+1}\|_M^2 + \frac{1}{2}\,\|A(x - \hat{x}_{k+1})\|^2 + \frac{1}{2}\,\|\hat{x}_{k+1} - \hat{x}_k\|_M^2. \end{aligned} \tag{6.14}$$

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

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

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

$$\begin{aligned} f(x) + \frac{1}{2}\|Ax + B\bar{y} - c\|^2 + \frac{1}{2K}\|x - \hat{x}_0\|_M^2 \\ \geq f(\tilde{x}) + \frac{1}{2}\|A\tilde{x} + B\bar{y} - c\|^2 + \frac{1}{2K}\|x - \hat{x}_K\|_M^2 \\ + \frac{1}{2}\|A(x - \tilde{x})\|^2 + \frac{1}{2K}\sum_{k=0}^{K-1}\|\hat{x}_{k+1} - \hat{x}_k\|_M^2. \end{aligned} \tag{6.15}$$

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

$$\begin{aligned} g(y) + \frac{1}{2}\|A\tilde{x} + By - c\|^2 + \frac{1}{2L}\|y - \hat{y}_0\|_P^2 \\ \geq g(\tilde{y}) + \frac{1}{2}\|A\tilde{x} + B\tilde{y} - c\|^2 + \frac{1}{2L}\|y - \hat{y}_L\|_P^2 \\ + \frac{1}{2}\|B(y - \tilde{y})\|^2 + \frac{1}{2L}\sum_{\ell=0}^{L-1}\|\hat{y}_{\ell+1} - \hat{y}_\ell\|_P^2. \end{aligned} \tag{6.16}$$

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

$$\begin{aligned} f(x) + g(y) + \frac{1}{2K}\|x - \hat{x}_0\|_M^2 + \frac{1}{2L}\|y - \hat{y}_0\|_P^2 \\ + \frac{1}{2}\|Ax + B\bar{y} - c\|^2 + \frac{1}{2}\|A\tilde{x} + By - c\|^2 \\ \geq f(\tilde{x}) + g(\tilde{y}) + \frac{1}{2K}\|x - \hat{x}_K\|_M^2 + \frac{1}{2L}\|y - \hat{y}_L\|_P^2 \\ + \frac{1}{2}\|A\tilde{x} + B\bar{y} - c\|^2 + \frac{1}{2}\|A\tilde{x} + B\tilde{y} - c\|^2 \\ + \frac{1}{2}\|A(x - \tilde{x})\|^2 + \frac{1}{2}\|B(y - \tilde{y})\|^2 \\ + \frac{1}{2K}\sum_{k=0}^{K-1}\|\hat{x}_{k+1} - \hat{x}_k\|_M^2 + \frac{1}{2L}\sum_{\ell=0}^{L-1}\|\hat{y}_{\ell+1} - \hat{y}_\ell\|_P^2. \end{aligned} \tag{6.17}$$

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

$$\begin{aligned}
f(x)+g(y)+\frac{1}{2}\|Ax-By-c\|^2 & \\
+\frac{1}{2K}\|x-\hat{x}_0\|_M^2+\frac{1}{2L}\|y-\hat{y}_0\|_P^2+\frac{1}{2}\|B(y-\bar{y})\|^2 & \\
\geq f(\tilde{x})+g(\tilde{y})+\frac{1}{2}\|A\tilde{x}+B\tilde{y}-c\|^2 & \\
+\frac{1}{2K}\|x-\hat{x}_K\|_M^2+\frac{1}{2L}\|y-\hat{y}_L\|_P^2+\frac{1}{2}\|B(y-\tilde{y})\|^2 & \\
+\frac{1}{2K}\sum_{k=0}^{K-1}\|\hat{x}_{k+1}-\hat{x}_k\|_M^2+\frac{1}{2L}\sum_{\ell=0}^{L-1}\|\hat{y}_{\ell+1}-\hat{y}_\ell\|_P^2 & \\
+\frac{1}{2}\|A(x-\tilde{x})\|^2+\frac{1}{2}\|B(y-\bar{y})\|^2 & \\
-\frac{1}{2}\|Ax+B\bar{y}-c\|^2-\frac{1}{2}\|A\tilde{x}+By-c\|^2 & \\
+\frac{1}{2}\|Ax+By-c\|^2+\frac{1}{2}\|A\tilde{x}+B\bar{y}-c\|^2. &
\end{aligned} \tag{6.18}$$

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

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

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

$$\begin{aligned}
f(x)+g(y)+\frac{1}{2}\|Ax+By-c\|^2 & \\
+\frac{1}{2K}\|x-\hat{x}_0\|_M^2+\frac{1}{2L}\|y-\hat{y}_0\|_P^2+\frac{1}{2}\|B(y-\bar{y})\|^2 & \\
\geq f(\tilde{x})+g(\tilde{y})+\frac{1}{2}\|A\tilde{x}+B\tilde{y}-c\|^2 & \\
+\frac{1}{2K}\|x-\hat{x}_K\|_M^2+\frac{1}{2L}\|y-\hat{y}_L\|_P^2+\frac{1}{2}\|B(y-\tilde{y})\|^2. &
\end{aligned} \tag{6.20}$$

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

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

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

télescopent, ce qui entraîne

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

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

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

alors on obtient quatre sommes télescopiques

$$\mathcal{E}(x,y) + \frac{1}{2KN}\|x - x_K^0\|_M^2 + \frac{1}{2LN}\|y - y_L^0\|_P^2 + \frac{1}{2N}\|B(y-y^0)\|^2$$
$$\geq \frac{1}{N}\sum_{n=1}^{N}\mathcal{E}(x^n,y^n) + \frac{1}{2KN}\|x - x_K^N\|_M^2 + \frac{1}{2LN}\|y - y_L^N\|_P^2 + \frac{1}{2N}\|B(y-y^N)\|^2. \tag{6.21}$$

On introduit alors les deux moyennes

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

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

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

qui se lit également

$$\mathcal{E}(X^N,Y^N) - \mathcal{E}(x^*,y^*) \leq \frac{1}{N}\left\{\frac{\|x^* - x_K^0\|_M^2 - \|x^* - x_K^N\|_M^2}{2K} + \frac{\|y^* - y_L^0\|_P^2 - \|y^* - y_L^N\|_P^2}{2L}\right.$$
$$\left. + \frac{\|B(y^*-y^0)\|^2 - \|B(y^*-y^N)\|^2}{2}\right\} \tag{6.22}$$

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

6.2.3 Accélération

On propose maintenant une variante de l'algorithme, en introduisant une accélération de type FISTA pour obtenir une convergence plus rapide, donné par les mises-à-

jours suivantes pour $n \geq 0$:

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

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

$$\forall\, n \geq 0, \qquad t_n^2 \geq t_{n+1}(t_{n+1} - 1). \tag{6.23}$$

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

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

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

Si on utilise par ailleurs la convexité de $\mathcal{E}$ et qu'on multiplie l'inégalité résultante par t_{n+1}^2, on obtient

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

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

on obtient $$(t_{n+1} - 1)\,y^n + y^* - t_{n+1}\,\bar{y} = (t_n - 1)\,y^{n-1} + y^* - t_n\,y^n.$$

Si on choisit (avec par convention $x^{-1} = x^0$ et $y^{-1} = y^0$)

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

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

on a alors d'une part

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

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

d'autre part. Ainsi, (6.20) devient

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

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

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

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

avec

$$\begin{aligned}
C_N = &\frac{\|B(y^* - t_0\,y^0 + (t_0-1)\,y^{-1})\|^2 - \|B(y^* - t_N\,y^N + (t_N-1)\,y^{N-1})\|^2}{2}\\
&+ \frac{\|x^* - t_0\,x_K^0 + (t_0-1)\,x^{-1}\|_M^2 - \|x^* - t_N\,x_K^N + (t_N-1)\,x^{N-1}\|_M^2}{2K}\\
&+ \frac{\|y^* - t_0\,y_L^0 + (t_0-1)\,y^{-1}\|_P^2 - \|y^* - t_N\,y_L^N + (t_N-1)\,y^{N-1}\|_P^2}{2L}.
\end{aligned} \tag{6.24}$$

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

6.2.4 Application au modèle ROF

Montrons à présent comment cet algorithme peut être utilisé dans une approche d'éclatement pour résoudre le problème ROF dans le cas des images couleurs.

Version discrète de TV 2D couleur Dans le cas des images couleurs (on se restreint ici au cas des images à $m = 3$ canaux couleurs RGB, mais la démarche est généralisable au cas multi-spectral avec $m > 3$), l'image u peut se décomposer en trois images mono-valuées

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

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

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

alors que la version (spatialement) isotrope donne

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

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

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

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

$$\begin{aligned}(\nabla^h v^h)_{i,j} &= \Big((\nabla^h(v^h)^{\mathrm{R}})_{i,j},(\nabla^h(v^h)^{\mathrm{G}})_{i,j},(\nabla^h(v^h)^{\mathrm{B}})_{i,j}\Big)\\ &= \begin{pmatrix}(\delta_x^h(v^h)^{\mathrm{R}})_{i,j} & (\delta_x^h(v^h)^{\mathrm{G}})_{i,j} & (\delta_x^h(v^h)^{\mathrm{B}})_{i,j}\\ (\delta_y^h(v^h)^{\mathrm{R}})_{i,j} & (\delta_y^h(v^h)^{\mathrm{G}})_{i,j} & (\delta_y^h(v^h)^{\mathrm{B}})_{i,j}\end{pmatrix}\end{aligned}$$

$$(\nabla^h v^h)_{i,j} = \begin{pmatrix}(\delta_x^h v^h)_{i,j}\\ (\delta_y^h v^h)_{i,j}\end{pmatrix} \quad \text{avec} \quad \begin{cases}(\delta_x^h v^h)_{i,j} = \Big((\delta_x^h(v^h)^{\mathrm{R}})_{i,j} & (\delta_x^h(v^h)^{\mathrm{G}})_{i,j} & (\delta_x^h(v^h)^{\mathrm{B}})_{i,j}\Big)\\ (\delta_y^h v^h)_{i,j} = \Big((\delta_y^h(v^h)^{\mathrm{R}})_{i,j} & (\delta_y^h(v^h)^{\mathrm{G}})_{i,j} & (\delta_y^h(v^h)^{\mathrm{B}})_{i,j}\Big)\end{cases}$$

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

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

Les différentes normes TV couleur proposées plus haut correspondent donc respectivement à la définition de $\|(\nabla v^h)_{i,j}\|_{1,1}$, $\|(\nabla v^h)_{i,j}\|_{2,1}$ et $\|(\nabla v^h)_{i,j}\|_{2,2}$. On peut également choisir d'intervertir l'ordre de traitement respectif de la coordonnée spatiale et du canal

couleur, en définissant pour tous entiers α et β la norme $\|\cdot\|^{\alpha,\beta}$ en posant pour toute matrice $A = (a_{i,j})_{\substack{1\leq i\leq 2\\ 1\leq j\leq 3}}$

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

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

$$\|(\nabla^h v^h)_{i,j}\|^{2,1} = \Big\|\Big(\|(\delta_x^h v^h)_{i,j}\|_2, \|(\delta_y^h v^h)_{i,j}\|_2\Big)\Big\|_1 = \|(\delta_x^h v^h)_{i,j}\|_2 + \|(\delta_y^h v^h)_{i,j}\|_2. \tag{6.28}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

$$S_{i,j}(D^x \xi_x^h) = D^*(S_{i,j}\xi_x^h)$$

puis, pour i et j impairs, $\qquad S_{i,j}(D^y \xi_y^h) = D^*(S_{i,j}\xi_y^h)$

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

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

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

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

On en déduit en particulier que

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

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

$$\begin{cases} (\xi_x^h)_0^{n+1} = (\xi_x^h)_K^n \\ \begin{cases} \forall\, k \in [\![\, 0\,;\, K-1 \,]\!] \\ \qquad (\xi_x^h)_{k+1}^{n+1} = \displaystyle\operatorname*{argmin}_{\substack{\xi_x^h \in \mathbb{R}^{3\times N_x \times N_y} \\ \|(\xi_x^h)_{i,j}\|_2 \leq 1}} \left\{ \frac{1}{2} \left\| D^x \xi_x^h + (D^y \xi_y^h)^n - g^h/\lambda \right\|^2 + \frac{1}{2\,\tau} \left\| \xi_x^h - (\xi_x^h)_k^{n+1} \right\|^2 \right. \\ \qquad\qquad\qquad\qquad \left. - \frac{1}{2} \left\| D^x \xi_x^h - D^x (\xi_x^h)_k^{n+1} \right\|^2 \right\} \end{cases} \\ (\xi_x^h)^{n+1} = \dfrac{1}{K} \displaystyle\sum_{k=1}^{K} (\xi_x^h)_k^{n+1} \qquad \text{et} \qquad (\xi_y^h)_0^{n+1} = (\xi_y^h)_L^n \\ \begin{cases} \forall\, \ell \in [\![\, 0\,;\, L-1 \,]\!] \\ \qquad (\xi_y^h)_{\ell+1}^{n+1} = \displaystyle\operatorname*{argmin}_{\substack{\xi_y^h \in \mathbb{R}^{3\times N_x \times N_y} \\ \|(\xi_y^h)_{i,j}\|_2 \leq 1}} \left\{ \frac{1}{2} \left\| D^x (\xi_x^h)^{n+1} + D^y \xi_y^h - g^h/\lambda \right\|^2 + \frac{1}{2\,\tau} \left\| \xi_y^h - (\xi_y^h)_\ell^{n+1} \right\|^2 \right. \\ \qquad\qquad\qquad\qquad \left. - \frac{1}{2} \left\| D^y \xi_y^h - D^y (\xi_y^h)_\ell^{n+1} \right\|^2 \right\} \end{cases} \\ (\xi_y^h)^{n+1} = \dfrac{1}{L} \displaystyle\sum_{\ell=1}^{L} (\xi_y^h)_\ell^{n+1} \end{cases}$$

qui, après simplification, s'écrit

$$\begin{cases} (\xi_x^h)_0^{n+1} = (\xi_x^h)_K^n \\ \begin{cases} \forall\, k \in [\![\, 0\,;\, K-1\,]\!], \forall\, (i,j) \text{ pair} \\ \qquad S_{i,j}(\xi_x^h)_{k+1}^{n+1} = \mathrm{proj}_{B(0,1)}\Big(S_{i,j}(\xi_x^h)_k^{n+1} + \tau\, D\Big(D^*(S_{i,j}\xi_x^h)_k^{n+1} + S_{i,j}(D^y(\xi_y^h)^n - g^h/\lambda)\Big)\Big) \end{cases} \\ (\xi_x^h)^{n+1} = \dfrac{1}{K}\displaystyle\sum_{k=1}^{K}(\xi_x^h)_k^{n+1} \qquad \text{et} \qquad (\xi_y^h)_0^{n+1} = (\xi_y^h)_L^n \\ \begin{cases} \forall\, \ell \in [\![\, 0\,;\, L-1\,]\!], \forall\, (i,j) \text{ impair} \\ \qquad S_{i,j}(\xi_y^h)_{\ell+1}^{n+1} = \mathrm{proj}_{B(0,1)}\Big(S_{i,j}(\xi_y^h)_\ell^{n+1} + \tau\, D\Big(D^*(S_{i,j}\xi_y^h)_\ell^{n+1} + S_{i,j}(D^x(\xi_x^h)^{n+1} - g^h/\lambda)\Big)\Big) \end{cases} \\ (\xi_y^h)^{n+1} = \dfrac{1}{L}\displaystyle\sum_{\ell=1}^{L}(\xi_y^h)_\ell^{n+1} \end{cases}$$

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

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

$$\mathrm{TV}^h_{\text{pseudo-iso}}(v^h) = \underbrace{\sum_{\substack{i=0 \\ i \text{ pair}}}^{N_x-1}\sum_{\substack{j=0 \\ j \text{ pair}}}^{N_y-1}\|D(Sv^h)_{i,j}\|^{2,2}}_{=\mathrm{TV}^h_{\mathrm{e}}(v^h)} + \underbrace{\sum_{\substack{i=-1 \\ i \text{ impair}}}^{N_x-1}\sum_{\substack{j=-1 \\ j \text{ impair}}}^{N_y-1}\|D(Sv^h)_{i,j}\|^{2,2}}_{=\mathrm{TV}^h_{\mathrm{o}}(v^h)}$$

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

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

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

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

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

(a) Image originale u (b) Image bruitée g (c) Image débruitée V^*

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

et l'énergie duale :

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

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

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

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

6.3 Résultats expérimentaux

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

6.3.1 Choix de la référence

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

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

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

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

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

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

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

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

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

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

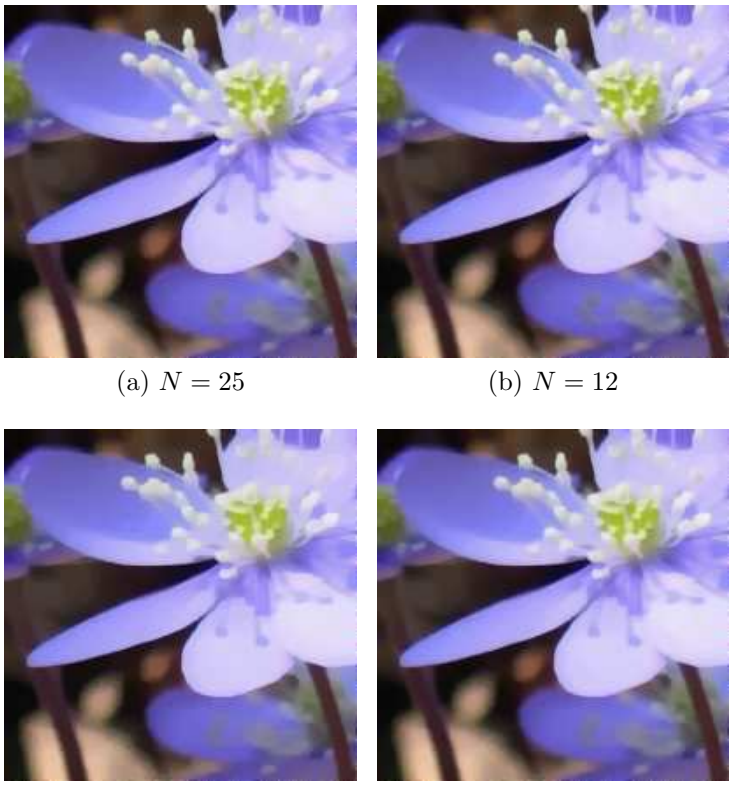
où le domaine de l'image g^h est de taille $N_x \times N_y$. Plus cette erreur est faible, plus v^* est proche du miminiseur V^*.

6.3.2 Descentes alternées pour l'éclatement sur carrés

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

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

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



(a) $N = 25$ (b) $N = 12$

(c) $N = 114$ (d) $N = 48$

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

Itérations locales $K = L$	1	2	3	4	5	6	7	8	9	10
Itérations globales N	37	29	25	23	22	21	20	20	20	19
Temps total (en secondes)	0,090	0,086	0,089	0,095	0,102	0,110	0,115	0,127	0,138	0,141
Temps par descente (en millisecondes)	1,216	0,741	0,590	0,516	0,463	0,437	0,411	0,397	0,383	0,371

(a)

Itérations locales $K = L$	1	2	3	4	5	6	7	8	9	10
Itérations globales N	20	14	12	12	12	12	12	12	12	12
Temps total (en secondes)	0,081	0,065	0,061	0,068	0,074	0,081	0,088	0,095	0,101	0,110
Temps par descente (en millisecondes)	2,025	1,161	0,847	0,708	0,617	0,563	0,495	0,495	0,468	0,458

(b)

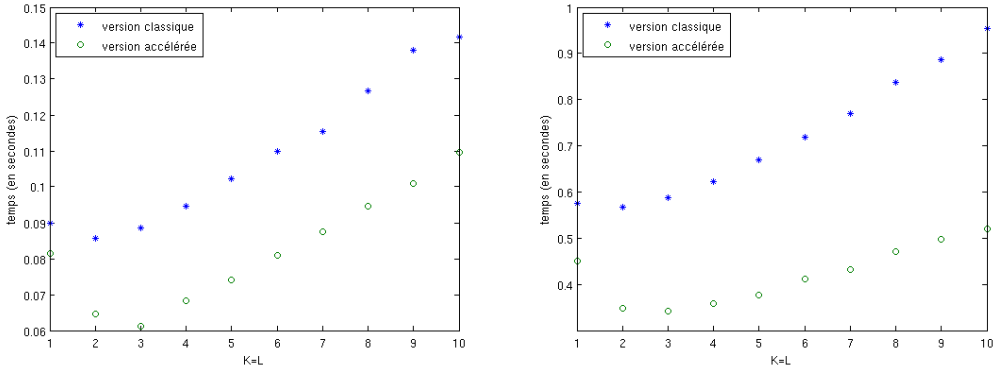
Itérations locales $K = L$	1	2	3	4	5	6	7	8	9	10
Itérations globales N	189	138	114	100	92	86	82	80	77	76
Temps total (en secondes)	0,575	0,566	0,588	0,622	0,670	0,718	0,770	0,838	0,886	0,953
Temps par descente (en millisecondes)	1,521	1,026	0,860	0,778	0,728	0,696	0,671	0,655	0,639	0,627

(c)

Itérations locales $K = L$	1	2	3	4	5	6	7	8	9	10
Itérations globales N	90	58	48	44	41	40	38	38	37	36
Temps total (en secondes)	0,451	0,348	0,341	0,358	0,377	0,411	0,432	0,472	0,497	0,521
Temps par descente (en millisecondes)	2,506	1,500	1,184	1,017	0,920	0,856	0,812	0,776	0,746	0,724

(d)

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



(a) TV pseudo-isotrope (b) TV anisotrope

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

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

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

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

6.3.3 Comparaison avec d'autres méthodes

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

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

6.3.4 Discussion

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

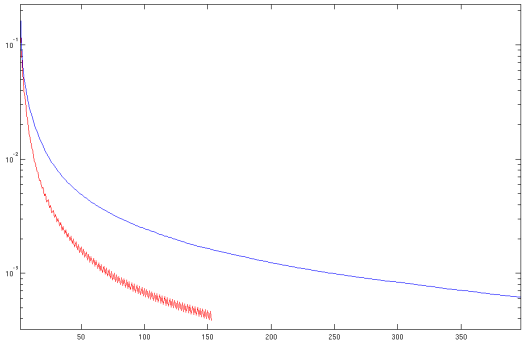
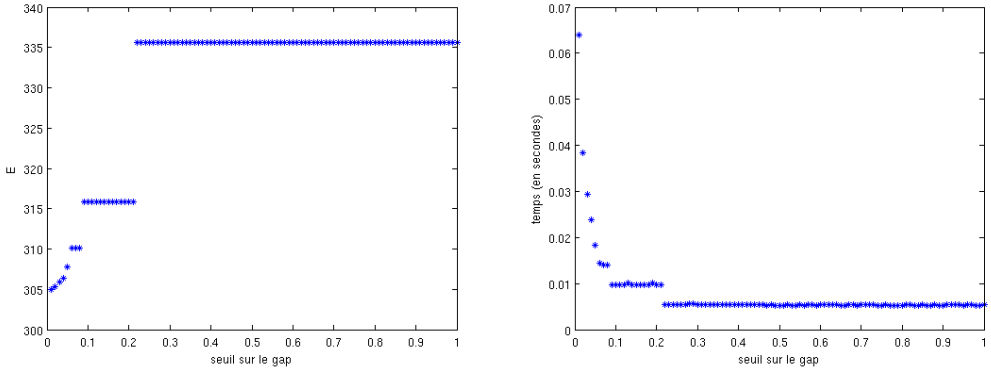


FIGURE 6.5 – Allure des *gaps* moyens (échelle logarithmique en ordonnée) au cours des itérations. En bleu, la méthode classique et en rouge, la version accélérée (TV pseudo-isotrope, $K = L = 3$). Remarquer les oscillations du *gap* dans la version accélérée.



(a) Influence sur l'énergie (b) Influence sur le temps de calcul

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

(a) (b)

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

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

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

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

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

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

car il comprend le temps dédié à la sur-relaxation.

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

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

Conclusion

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

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

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

Références

[1] Antonin Chambolle and Thomas Pock. A remark on accelerated block coordinate descent for computing the proximity operators of a sum of convex functions. *SMAI Journal of Computational Mathematics*, 1 :29–54, 2015.

[2] Laurent Condat. A direct algorithm for 1D total variation denoising. *IEEE Signal Processing Letters*, 20(11) :1054–1057, 2013.

[3] Joan Duran, Michael Moeller, Catalina Sbert, and Daniel Cremers. On the implementation of collaborative TV regularization : Application to cartoon+ texture decomposition. *Image Processing On Line*, 6 :27–74, 2016.

[4] Ernie Esser, Xiaoqun Zhang, and Tony Chan. A general framework for a class of first order primal-dual algorithms for TV minimization. *UCLA CAM Report*, pages 09–67, 2009.

[5] Nicholas Johnson. *Efficient models and algorithms for problems in genomics*. PhD thesis, Stanford University, 2010.

[6] Leonid I. Rudin. Images, numerical analysis of singularities and shock filters. 1987.

[7] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D : Nonlinear Phenomena*, 60(1) :259–268, 1992.

université PARIS-SACLAY

ÉCOLE DOCTORALE
INTERFACES
Approches Interdisciplinaires : Fondements, Applications et Innovation

Titre : Précision de modèle et efficacité algorithmique : exemples du traitement de l'occultation en stéréovision binoculaire et de l'accélération de deux algorithmes en optimisation convexe

Mots clés : stéréovision binoculaire, occultation, méthodes variationnelles, vision par ordinateur, optimisation convexe, calcul parallèle

Résumé : Le présent manuscrit est composé de deux parties relativement indépendantes.
La première partie est consacrée au problème de la stéréovision binoculaire, et plus particulièrement au traitement de l'occultation. En partant d'une analyse de ce phénomène, nous en déduisons un modèle de régularité qui inclut une contrainte convexe de visibilité. La fonctionnelle d'énergie qui en résulte est minimisée par relaxation convexe. Les zones occultées sont alors détectées grâce à la pente horizontale de la carte de disparité avant d'être densifiées. Une autre méthode gérant l'occultation est la méthode des graph cuts proposée par Kolmogorov et Zabih. L'efficacité de cette méthode justifie son adaptation à deux problèmes auxiliaires rencontrés en stéréovision, qui sont la densification de cartes éparses et le raffinement subpixellique de cartes pixelliques.
La seconde partie de ce manuscrit traite de manière plus générale de deux algorithmes d'optimisation convexe, pour lequels deux variantes accélérées sont proposées. Le premier est la méthode des directions alternées (ADMM). On montre qu'un léger relâchement de contraintes dans les paramètres de cette méthode permet d'obtenir un taux de convergence théorique plus intéressant. Le second est un algorithme de descentes proximales alternées, qui permet de paralléliser la résolution approchée du problème Rudin-Osher-Fatemi (ROF) de débruitage pur dans le cas des images couleurs. Une accélération de type FISTA est également proposée.

Title : Model accuracy and algorithmic efficiency : examples of occlusion handling in binocular stereovision and the acceleration of two convex optimization algorithms

Keywords : binocular stereovision, occlusion, variational methods, computer vision, convex optimization, parallel computing

Abstract : This thesis is splitted into two relatively independant parts.
The first part is devoted to the binocular stereovision problem, specifically to the occlusion handling. An analysis of this phenomena leads to a regularity model which includes a convex visibility constraint. The resulting energy functional is minimized by convex relaxation. The occluded areas are then detected thanks to the horizontal slope of the disparity map and densified. Another method with occlusion handling was proposed by Kolmogorov and Zabih. Because of its efficiency, we adapted it to two auxiliary problems encountered in stereovision, namely the densification of sparse disparity maps and the subpixel refinement of pixel-accurate maps.
The second part of this thesis studies two convex optimization algorithms, for which an acceleration is proposed. The first one is the Alternating Direction Method of Multipliers (ADMM). A slight relaxation in the parameter choice is shown to enhance the convergence rate. The second one is an alternating proximal descent algorithm, which allows a parallel approximate resolution of the Rudin-Osher-Fatemi (ROF) pure denoising model, in color-image case. A FISTA-like acceleration is also proposed.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin,
France