Rapport de recherche sur l'optimisation numérique et le débruitage en imagerie radar à synthèse d'ouverture

Xavier Rondeau

2010D022

Septembre 2010

Département Traitement du Signal et des Images
Groupe TII : Traitement et Interprétation des Images

Rapport de recherche sur l'optimisation numérique et le débruitage en imagerie radar à synthèse d'ouverture.

Xavier Rondeau

Institut Télecom, Télecom ParisTech, LTCI UMR 5141
46 rue Barrault, 75013 Paris
Département du Traitement du Signal et des Images

Financement DGA
Projet REI 2008.34.0042

Septembre 2010

Ce rapport présente les comparaisons que j'ai effectuées dans le cadre du projet REI sur le débruitage multiplicatif et non-stationnaire des images radar par synthèse d'ouverture (R.S.O.), sous l'encadrement de Florence Tupin, et en collaboration avec Loïc Denis, Eric Thiébaut, Jean-Marie Nicolas, Jean-François Aujol, Marc Sigelle et Jérôme Darbon. Après une brève introduction sur l'imagerie RSO haute résolution et sur les problématiques de débruitage qui lui sont associées, je décris les modélisations statistiques considérées pour le débruitage ainsi que les différents estimateurs envisagés dans un cadre Bayésien avec des a priori Markovien. Dans une deuxième section, j'introduis ensuite les algorithmes d'optimisation mis en œuvre ou développés pour calculer ces différents estimateurs. Puis je présente la base de données ainsi que les critères retenus pour effectuer les comparaisons. Je détaille dans la dernière partie la mise en œuvre pratique des algorithmes de débruitage envisagés, avant de présenter les résultats et les conclusions des comparaisons, que ce soit en terme d'optimisation ou de qualité du débruitage opéré.

Table des matières

1 Modélisation et estimation statistiques. 3

1.1 Bruit en imagerie radar. . . . 3

1.2 Estimation bayésienne 3

1.3 Densité de probabilité expérimentale (attache aux données) 5

1.4 Densité de probabilité a priori (régularisation) 7

1.4.1 Champs de Gibbs-Markov 7

1.4.2 Discrétisation 8

1.4.3 Fonction de potentiel 9

1.5 Débruitage par moyennes non-locales 11

1.6 Débruitage par ondelettes 11

2 Calcul des estimateurs 13

2.1 Calcul de $\mathbf{a}^{\mathrm{MAP}}$ par optimisation différentiable 13

2.1.1 Optimisation convexe 13

2.1.1.1 Algorithme de premier ordre 14

2.1.1.2 Algorithmes du second ordre 15

2.1.1.3 Gradients conjugués non-linéaires 16

2.1.2 Optimisation non-convexe 16

2.1.3 Optimisation contrainte par des bornes simples. . . . 18

2.2 Calcul de $\mathbf{a}^{\mathrm{MAP}}$ par coupure sur graphe 22

2.2.1 Coupure minimale sur un graphe 23

2.2.2 Maximisation de flot 24

2.2.3 Graphe d'Ishikawa 25

2.2.4 Coupures itératives 26
2.3 Optimisation stochastique 29
2.3.1 Echantillonnage Monte-Carlo mono-dimensionnel 29
2.3.2 Echantillonnage en grande dimension par chaînes de Markov 31
2.3.2.1 Algorithme de Metropolis-Hastings 31
2.3.2.2 Algorithme de Gibbs 32
2.3.2.3 Echantillonnage dynamique et algorithme de Monte-Carlo Hybride 33
2.3.2.4 Recuit simulé 35

3 Base de données et critères de comparaisons 37
3.1 Données 37
3.2 Critères de comparaisons 42
3.2.1 Poids de la régularisation 42
3.2.2 Critères globaux 42
3.2.3 Critères locaux 44
3.2.4 Performances des algorithmes d'optimisation 44

4 Résultats des comparaisons et conclusions. 46
4.1 Comparaison entre les différents algorithmes d'optimisation MAP 46
4.1.1 Minimisation de $f_{\text{data}}(\mathbf{a}) + \beta\,\Phi^{\ell_1}_{\text{diff}}(\mathbf{a})$ 46
4.1.1.1 Optimisation combinatoire 46
4.1.1.2 Optimisation différentiable 49
4.1.1.3 Optimisation par échantillonnage MCMC 51
4.1.2 Minimisation de $f_{\text{data}}(\mathbf{a}) + \beta\,\Phi^{\ell_1}_{\text{grad}}(\mathbf{a})$ 53
4.1.2.1 Optimisation différentiable 54
4.1.2.2 Optimisation par échantillonnage MCMC 56
4.1.3 Conclusion concernant les algorithmes d'optimisation MAP 56
4.2 Calcul de la moyenne a posteriori 57
4.3 Comparaison entre les différents estimateurs 58
4.3.1 Données synthétiques 58
4.3.2 Données réelles 65
4.4 Conclusion 65

Bibliographie 70

Introduction

Le radar est une modalité de télédétection initialement monodimensionnelle qui permet de mesurer la distance d'une cible, grâce à l'émission d'une onde électromagnétique (radiofréquences par distinction avec le lidar en optique) puis à la détection de l'onde rétrodiffusée par cette cible. C'est donc un système actif d'imagerie de type écholocation, où le système d'émission est généralement confondu avec le système de réception.

Pour obtenir une image 2D, l'imagerie radar combine le déplacement du radar (rotation ou translation) avec cette mesure en distance. Plus précisément, on s'intéressera ici à l'imagerie radar de type satellitaire qui est effectuée en embarquant les radars dans des avions ou des satellites. Un traitement du signal basé sur les mesures effectuées par le capteur radar en différents points et dans différentes directions, permet alors de recontruire typiquement une carte 2D des amplitudes ou intensités rétrodiffusées par les éléments répartis à la surface du sol de la zone d'intérêt (carte de réflectance). L'imagerie satellite radar fournit en principe une résolution spatiale plus faible que l'imagerie satellite optique, qui est basée sur la rétrodiffusion de la lumière diurne, à cause de la plus grande longueur d'onde de rayonnement mise en œuvre dans le cas monochromatique (hyperfréquences au lieu du visible). Cependant, les systèmes d'imagerie satellite radar présentent l'avantage par rapport aux systèmes optiques de ne pas dépendre des conditions d'ensoleillement, de ne pas être affectés par la propagation des ondes électromagnétiques dans l'atmosphère (interaction turbulente du rayonnement optique et absorption nébuleuse), et enfin d'utiliser un rayonnement cohérent, ce qui permet de pouvoir effectuer la synthèse d'ouverture RSO notamment. De plus, les informations de rétrodiffusion radar sont complémentaires à celles de l'imagerie optique, parce que le rayonnement n'est pas le même et parce que la formation du signal donne lieu à des phénomènes d'ombres et de recouvrements différents de l'imagerie satellite optique.

La résolution d'un système d'imagerie radar est en principe d'une part limitée par la résolution temporelle selon la dimension *en distance*. Elle est d'autre part limitée par les dimensions de l'antenne selon la dimension *en azimuth*, en vertu de la diffraction et du rayonnement d'antenne à l'émission. L'amélioration de la résolution en distance se heurte aux limitations technologiques liées à l'émission d'un signal très puissant dans un intervalle de temps de l'ordre de la nanoseconde. Les systèmes radars haute résolution ont ainsi recours à des techniques artificielles de traitement du signal, comme notamment le filtrage adapté d'un signal modulé en fréquence. De même pour contourner les limitations physiques liées aux dimensions de l'antenne embarquée et ainsi améliorer la résolution en azimuth, les systèmes radar *à synthèse d'ouverture* (RSO) ont également recours à ces mêmes techniques de traitement du

signal (le signal émis en différentes positions de la trajectoire du porteur pouvant alors être considére comme un signal modulé en fréquences spatiales). Le lecteur souhaitant trouver une introduction approfondie sur l'imagerie radar pourra par exemple se référer à [1]. Actuellement les résolutions des systèmes radars civils sont de l'ordre du m (TerraSAR-X par exemple, situé sur une orbite à 500 km du sol) pour les radars embarqués dans des satellites, voire de l'ordre de quelques cm pour les radars embarqués dans des avions.

La forte amélioration de la résolution spatiale des systèmes radars au cours de ces dernières années est donc le principal facteur motivant cette étude sur le débruitage en imagerie RSO. Cela nous conduit en effet à considérer de nouveaux estimateurs et de nouvelles modélisations statistiques pour le débruitage RSO, d'autant que les techniques de débruitage (non-convexes) préservant les contours et les contrastes ont également progressé parallèlement.

Il convient de noter qu'il peut être bénéfique de paramétrer les contenus de l'image pour la débruiter, et qu'on peut gagner à incorporer directement les traitements d'image de type débruitage au sein de l'application considérée (segmentation, détection de contours ou détection de motifs par filtrage adapté par exemple). Néanmoins on se positionne ici dans un cadre de modélisation de plus bas niveau, car on souhaite souvent pouvoir effectuer différents types de traitements sur les images d'une part, et car ceux-ci ne sont d'autre part pas toujours exactement délimités dans un premier temps. Il reste donc en pratique intéressant de considérer le débruitage sur les pixels comme un traitement à part entière, indépendamment même de l'intérêt qu'on peut trouver dans les méthodologies mises en œuvre pour optimiser des critères multi-modaux. Par ailleurs, étant donné qu'aucun modèle ne fait l'unanimité pour modéliser le bruit radar à haute résolution, il est également intéressant de mener ce type d'étude.

Chapitre 1

Modélisation et estimation statistiques.

1.1 Bruit en imagerie radar.

Les images RSO ont une apparence tavelée (petits speckles), ce qui est la conséquence de la sommation cohérente (interférences) des ondes rétrodiffusées dans une même cellule de résolution, ainsi que de la formation du signal RSO. Ainsi l'image RSO en amplitude (racine carrée de la réflectance) d'une portion de terrain homogène $\mathcal{I}$ et échantillonnée selon les pixels i :

$$a_i* = c \in \mathbb{R}_{>0} \quad \forall\, i \in \mathcal{I}\,, \tag{1.1}$$

est une portion d'amplitudes hétérogènes d_i, qui sont aléatoirement réparties autour d'une valeur moyenne c', et qui fluctuent également aléatoirement au cours du temps pour une même position $i{\in}\mathcal{I}$. Il convient ainsi de *débruiter* les amplitudes initiales de l'image d_i à l'aide d'une modélisation statistique appropriée. D'une façon générale on peut aussi noter que le bruit n_i affectant les images RSO est plutôt de nature multiplicative ($d_i \simeq a_i * \times n_i$), contrairement au bruit additif des caméras de détection CCD affectant les images optiques par exemple ($d_i \simeq a_i * + n_i$).

1.2 Estimation bayésienne

En première approximation — i.e. en négligeant la réponse impulsionnelle du capteur notamment — on peut considérer que les pixels d'une image radar RSO en amplitude $\mathbf{d}$, sont des réalisations de variables aléatoires *indépendantes* de paramètre a_i* (avec en principe $a_i* \neq a_j *$ lorsque $i{\neq}j$), ce qui est modélisé mathématiquement à travers la densité de probabilité conditionnelle des données sachant les paramètres (vraisemblance) qui est donc ici

séparable :

$$\mathcal{V}_{\text{data}}(\mathbf{d}|\mathbf{a}) = \prod_{i=1}^{N_a} \Pr\left(d_i|a_i\right) \,. \tag{1.2}$$

Cette vraisemblance correspond à une fonction de coût, également appelée terme d'attache aux données, après transformation monotone de type anti-logarithme :

$$f_{\text{data}}(\mathbf{d}|\mathbf{a}) = -\log\left[\mathcal{V}_{\text{data}}(\mathbf{d}|\mathbf{a})\right] \,. \tag{1.3}$$

Ainsi on dispose d'une unique réalisation aléatoire d_i pour estimer chaque a_i*, ce qui conduit à un problème d'estimation très mal posé étant donnée la fluctuation aléatoire qui affecte chaque donnée d_i. Cette instabilité de l'estimateur par rapport aux données est d'ailleurs d'autant plus problématique que la modélisation

$$\bar{\mathbf{d}} = \mathcal{M}(\mathbf{a}) \,, \tag{1.4}$$

est séparable ($\mathcal{M}$=cste$\times\mathcal{I}d$), et qu'elle ne permet donc même pas d'introduire de corrélations entre les paramètres a_i estimés, contrairement à un problème de reconstruction comme la déconvolution ou la tomographie. Néanmoins on peut tout de même remédier à ce caractère mal posé dans le cadre général de l'estimation bayésienne, en introduisant des corrélations a priori entre les paramètres à estimer. Les paramètres a_i sont ainsi reliés aux données à travers la densité de probabilité conditionnelle a posteriori :

$$\begin{aligned} \mathcal{L}_{\text{post}}(\mathbf{a};\beta|\mathbf{d}) &= \frac{\mathcal{L}_{\text{post}}(\mathbf{a};\beta;\mathbf{d})}{\mathcal{L}(\mathbf{d})} \\ &= \frac{\mathcal{V}_{\text{data}}(\mathbf{d}|\mathbf{a}) \times \mathcal{L}_{\text{prior}}(\mathbf{a}|\beta) \times \mathcal{L}_{\text{hyperprior}}(\beta)}{\iint_{\mathrm{D}_{\mathbf{a}},\mathrm{D}_\beta} \mathcal{L}_{\text{post}}(\mathbf{a};\beta;\mathbf{d})\,\mathrm{d}\mathbf{a}\mathrm{d}\beta} \,. \end{aligned} \tag{1.5}$$

où β est un hyper-paramètre permettant de régler la force des a priori, c'est à dire typiquement de régler la quantité de lissage spatial souhaitée dans l'estimation. Autrement dit β sert à régler un compromis biais/variance de l'estimée $\hat{\mathbf{a}}$ par rapport aux paramètres $\mathbf{a}*$, en supposant que les modélisations statistiques d'attache aux données $\mathcal{V}_{\text{data}}(\mathbf{d}|\mathbf{a})$ et d'a priori $\mathcal{L}_{\text{prior}}(\mathbf{a}|\beta)$ sont suffisamment pertinentes (elles sont détaillées un peu plus dans les deux sections suivantes).

On supposera dans un premier temps que l'hyperparamètre de régularisation β est fixé par l'utilisateur, plutôt que déduit de la relation a posteriori conditionnelle aux données 1.5. Ainsi on s'intéresse à l'estimation des paramètres $\mathbf{a}$ basée sur la loi a posteriori conditionnelle

$$\mathcal{L}_{\text{post}}(\mathbf{a}|\mathbf{d};\beta) \propto \mathcal{V}_{\text{data}}(\mathbf{d}|\mathbf{a}) \times \mathcal{L}_{\text{prior}}(\mathbf{a}|\beta) \,. \tag{1.6}$$

Dans ces conditions les estimateurs bayésiens sont généralement définis comme les solutions du problème intégral

$$\min_{\hat{\mathbf{a}}} \int_{\mathbf{a}} C(\mathbf{a};\hat{\mathbf{a}})\, \mathcal{V}_{\text{data}}(\mathbf{d}|\mathbf{a})\, \mathcal{L}_{\text{prior}}(\mathbf{a}|\beta)\, \mathrm{d}\mathbf{a} \,, \tag{1.7}$$

où $C(\mathbf{a};\hat{\mathbf{a}})$ est une fonction de coût de l'estimateur $\hat{\mathbf{a}}$ par rapport aux paramètres $\mathbf{a}$.

On s'intéressera en particulier dans cette étude au calcul de l'estimateur de maximum a posteriori $\hat{\mathbf{a}}^{\mathrm{MAP}}$, qui correspond au cas $C(\mathbf{a};\hat{\mathbf{a}})=\delta\left(\|\mathbf{a}-\hat{\mathbf{a}}\|_{\mathbf{2}}\right)$, et se réexprime comme la solution du problème :

$$\begin{aligned}\hat{\mathbf{a}}^{\mathrm{MAP}} &= \max_{\mathbf{a}} \ \mathcal{L}_{\mathrm{post}}(\mathbf{a}|\mathbf{d};\beta) && (1.8)\\ &= \min_{\mathbf{a}} \ -\log\left[\mathcal{L}_{\mathrm{post}}(\mathbf{a}|\mathbf{d};\beta)\right] && (1.9)\\ &= \min_{\mathbf{a}} \ f_{\mathrm{post}}(\mathbf{a}|\mathbf{d};\beta) = \min_{\mathbf{a}} \ f_{\mathrm{data}}(\mathbf{d}|\mathbf{a}) + \beta f_{\mathrm{prior}}(\mathbf{a}|\beta)\,. && (1.10)\end{aligned}$$

On cherchera à résoudre le problème MAP 1.10, à l'aide de techniques d'optimisation numérique différentiable, d'optimisation combinatoire ou d'optimisation stochastique par chaînes de Markov (MCMC), qui seront détaillées dans le chapitre suivant.

On s'intéressera d'autre part à l'estimateur de moyenne a posteriori (PM), qui correspond à la fonction de coût $C(\mathbf{a};\hat{\mathbf{a}})=\|\mathbf{a}-\hat{\mathbf{a}}\|_2^2$, et se réexprime comme la solution du problème intégral :

$$\mathbf{a}^{\mathrm{PM}} = \int_{D_{\mathbf{a}}} \mathbf{a} \times \mathcal{V}_{\mathrm{data}}(\mathbf{d}|\mathbf{a})\, \mathcal{L}_{\mathrm{prior}}(\mathbf{a}|\beta)\ \mathrm{d}\mathbf{a}\,, \tag{1.11}$$

On cherchera à résoudre le problème PM 1.11 à l'aide d'une technique d'intégration de Monte-Carlo par Chaînes de Markov basée sur un échantillonneur de type Metropolis-Hastings. La moyenne a posteriori est en général un estimateur bayésien statistiquement plus robuste que le maximum puisqu'il correspond à minimiser le risque bayésien de remplacer $\mathbf{a}*$ par $\mathbf{a}$ avec une fonction de coût décrivant l'ensemble du domaine de définition $D_{\mathbf{a}}$, au lieu d'une fonction ponctuelle pour l'estimateur MAP. La différence entre les deux estimateurs sera ici d'autant plus sensible que la vraisemblance a posteriori considérée est fortement disymétrique.

1.3 Densité de probabilité expérimentale (attache aux données)

En général on considère en imagerie RSO que les données en amplitude sont *identiquement* distribuées selon la loi de Rayleigh :

$$\Pr\left(d_i|a_i\right) = \frac{2\,d_i}{a_i^2} \times \exp\left[-\left(\frac{d_i}{a_i}\right)^2\right]\,, \tag{1.12}$$

en adoptant le formalisme du chatoiement pleinement développé de Goodman [3]. On suppose pour cela qu'une cellule de résolution (i.e. la zone imagée par un pixel) est constituée d'un grand nombre de réflécteurs aléatoirement distribués dans la cellule de résolution et que ces réflecteurs sont petits devant la longueur d'onde (points brillants réfléchissant de façon isotrope). La distribution de Rayleigh est une distribution à queue lourde contrairement à la distribution gaussienne, c'est à dire que sa queue n'a pas une décroissance exponentielle. Cela a notamment pour conséquence que l'anti-log vraisemblance de Rayleigh

$$f_{\mathrm{data}}(d_i|a_i) = \frac{d_i^2}{a_i^2} + 2\log(a_i) \quad (+cste)\,, \tag{1.13}$$

n'est pas convexe sur $\mathbb{R}_{>0}$ (voir la figure 1.1). Toutefois cette fonction reste unimodale sur $\mathbb{R}_{>0}$ et est strictement convexe sur $]0;\sqrt{3}d_i[$.

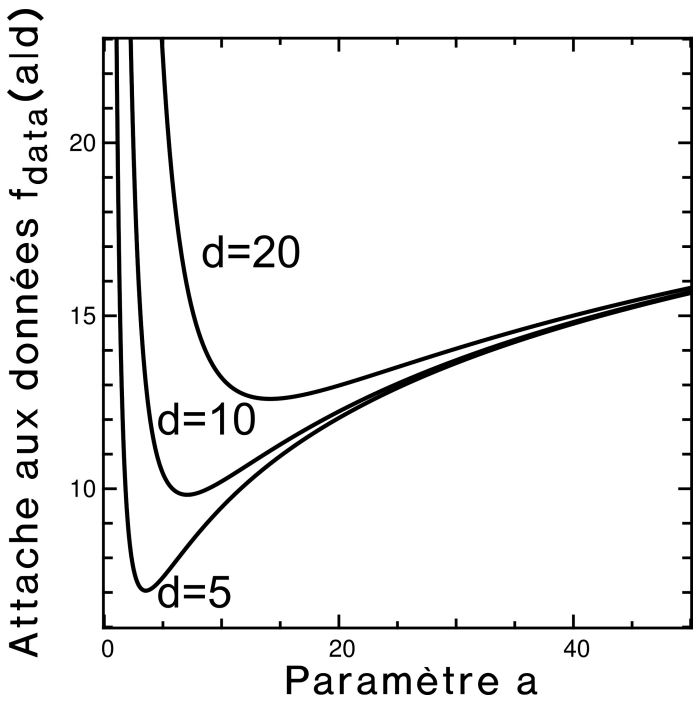


FIG. 1.1: Fonction d'attache aux données de Rayleigh f_{data}

L'amélioration de la résolution des capteurs RSO remet en cause le choix de cette unique distribution pour toutes les données d_i, car le nombre de réflecteurs diminue significativement dans certaines cellules d'une part et car la dépendance de la réflectance du sol avec la nature des cibles qui le constitue est par ailleurs exacerbée (composition, humidité, rugosité par rapport à la longueur d'onde). Une approche sans doute prometteuse consisterait à classifier les pixels de l'image en fonction de leur texture sous-jacente et à analyser la distribution effective des données sur des zones homogènes. La distribution de Fisher

$$\Pr\left(d_i|a_i;L_i;M_i\right) \propto \sqrt{\frac{L_i}{M_i}} \times \frac{2\,\Gamma(L_i+M_i)}{a_i\,\Gamma(L_i)\Gamma(M_i)} \times \frac{\left(\sqrt{\frac{L_i}{M_i}} \times \frac{d_i}{a_i}\right)^{2L_i-1}}{\left(1+\left(\sqrt{\frac{L_i}{M_i}} \times \frac{d_i}{a_i}\right)^2\right)^{L_i+M_i}}\,, \tag{1.14}$$

où $\Gamma(x)=\int_0^{+\infty} t^{x-1}\exp[-t]\mathrm{d}t$ est la fonction gamma, semble un bon candidat pour couvrir la diversité de distributions envisageables (lois à queue lourde et/ou à tête lourde) par le biais de l'ajustement de ses paramètres de forme M et L [2]. Toutefois cette approche est plus délicate et nécessite des données adaptées pour pouvoir mener une première étude dans ce sens. On se contentera donc dans cette étude de supposer que les données sont identiquement distribuées selon la loi de Rayleigh.

1.4 Densité de probabilité a priori (régularisation)

1.4.1 Champs de Gibbs-Markov

La prise en compte de la spécificité des différentes parties de l'image à débruiter est sans doute bénéfique lorsqu'on se concentre sur un traitement d'image spécifique. Cependant on se place ici dans un cadre plus bas niveau où l'on cherche une modélisation statistique a priori de l'image $\mathcal{L}_{\text{prior}}(\mathbf{a}|\beta)$ qui soit à la fois la plus adaptée possible à l'ensemble du contenu d'une image radar et également assez simple à mettre en œuvre. Dans ce contexte le formalisme des champs de Markov est sans doute l'un des plus pratiques, car il permet d'exprimer les interactions entre pixels voisins ou similaires sous une forme probabiliste et localisée. En effet la probabilité conditionnelle en chaque site — i.e. chaque pixel i — ne dépend alors que de la configuration des pixels appartenant au voisinage (ou ensemble de similarité) $\mathcal{V}_i$:

$$\Pr(\mathbf{a}|\beta) = \frac{\sum_{i=1}^{N_a} \exp\left[-\beta\,\Phi\left(a_i, \mathbf{a}_{\mathcal{V}_i}\right)\right]}{\mathcal{Z}}, \tag{1.15}$$

où $\mathcal{Z}$ est une fonction de normalisation :

$$\mathcal{Z} = \int_{D_{\mathbf{a}}} \sum_{i=1}^{N_a} \exp\left[-\beta\,\Phi\left(a_i, \mathbf{a}_{\mathcal{V}_i}\right)\right] \mathrm{d}\mathbf{a}\,, \tag{1.16}$$

également appelée fonction de partition, par emprunt à la description statistique d'un système thermodynamique en équilibre. $\mathcal{Z}$ est le plus souvent difficile à calculer, mais ne nécéssite pas de l'être pour les estimateurs auxquels on s'intéressera par la suite.

On privilégiera en particulier dans cette étude un voisinage $\mathcal{V}_i$ localisé, et restreint aux pixels dont la distance est strictement inférieure à deux pixels. Ainsi en prenant en compte les symétries des énergies considérées on se restreindra à un voisinage de 3 ou 4 pixels. Pour simplifier la présentation des fonctions de potentiel Φ considérées, on note pour la suite : A la valeur prise par le pixel correspondant à un site a_i, puis on représente par B et C les valeurs de ses deux plus proches voisins (horizontaux et verticaux) ainsi que par D et E celles de ses voisins diagonaux (voir figure 1.2).

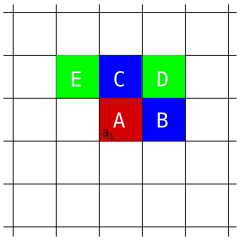


FIG. 1.2: Voisinage local considéré pour les corrélations a priori entre pixels

1.4.2 Discrétisation

On considérera alors principalement deux types d'approches, la première consistant à pénaliser les différences entre pixels :

$$\Phi_a(A,B,C,D,E) = \psi(A-B) + \psi(A-C) + \psi\left(\frac{A-D}{\sqrt{2}}\right) + \psi\left(\frac{A-E}{\sqrt{2}}\right), \tag{1.17}$$

où le facteur $1/\sqrt{2}$ est une normalisation qui correspond à la distance de A avec les pixels diagonaux E et D.

La deuxième approche consiste à pénaliser la norme euclidienne du gradient spatial :

$$\Phi_a(A,B,C,D) = \psi\left(\|\nabla a\|_2\right). \tag{1.18}$$

Bien que la seconde formulation soit plus isotrope et en particulier plus indépendante par rapport au choix des axes verticaux et horizontaux de l'image, ce qui favorise en principe la qualité de l'estimation si les poids sur les différences ne sont pas déterminés d'une façon probabiliste plus fine, la première approche reste toutefois attractive car sa simplicité (interactions entre 2 pixels seulement) permet de concevoir des algorithmes combinatoires puissants pour le cas de l'optimisation MAP multi-modale, comme nous le verrons dans la section 2.2. La deuxième approche est quand à elle mieux adaptée pour le cas de l'optimisation différentiable car la présence de corrélations plus étendues (dérivées secondes moins parcimonieuses) entre les pixels permet de concevoir des algorithmes de descente plus rapide. Pour distinguer les régularisations s'appliquant sur le gradient de celles s'appliquant sur les différences avec les plus proches voisins, nous utiliserons par la suite la notation indicée Φ_{diff} et Φ_{grad}.

Différentes discrétisations de la norme (carrée) du gradient

$$\|\nabla a(x,y)\|_2^2 = \left(\frac{\partial a}{\partial x}\right)^2 + \left(\frac{\partial a}{\partial y}\right)^2, \tag{1.19}$$

où x et y représentent les axes horizontaux et verticaux, sont envisageables à partir des pixels voisins. La plus simple et la plus répandue est basée sur une discrétisation de type différence finie :

$$\|\nabla a\|_2^2 = (A-B)^2 + (A-C)^2. \tag{1.20}$$

Pour gagner en isotropie nous considérerons toutefois plutôt la moyenne de la norme (carrée) du gradient dans une boîte 2×2 (A, B, C et D), en se basant sur une interpolation bilinéaire de la fonction

$$a(x,y) = (1-x)(1-y)\times A + x(1-y)\times B + (1-x)\,y\times C + xy\times D, \tag{1.21}$$

ce qui conduit à

$$\begin{aligned}\|\nabla a\|_2^2 &= \int_0^1\int_0^1 \|\nabla a(x,y)\|_2^2 \,\mathrm{d}x\mathrm{d}y \\ &= \frac{1}{2}\left[(A-D)^2 + (B-C)^2\right] + \frac{1}{6}(A-B-C+D)^2.\end{aligned} \tag{1.22}$$

1.4.3 Fonction de potentiel

Les fonctions de potentiel les plus utilisées sur les différences locales ou sur la norme du gradient spatial sont la fonction carrée

$$\psi^{\ell_2}(p) = p^2 \,, \tag{1.23}$$

qui favorise un lissage continu des paramètres ; et la fonction valeur absolue

$$\psi^{\ell_1}(p) = |p| \ . \tag{1.24}$$

Cette dernière favorise un lissage constant par morceaux des paramètres grâce à l'attaque franche de la fonction de potentiel proche de zéro, tout en favorisant une préservation plus nette des contours de l'image que la fonction ψ^{ℓ_2} grâce à une pénalisation plus faible des forts gradients (ou fortes différences entre les paramètres). Bien que ψ^{ℓ_1} ne soit pas différentiable en 0 elle est cependant sous-différentiable et l'on peut sélectionner au besoin la valeur nulle parmi les sous-gradients de ψ^{ℓ_1} en 0.

Un des inconvénients de la fonction de potentiel ψ^{ℓ_1} est que l'attaque raide en zéro favorise des images que l'on peut trouver trop constantes par morceaux. Il peut alors être avantageux de chercher à favoriser un lissage plus doux des paramètres dans les zones de faibles gradients, tout en conservant les discontinuités sur les contours de l'image à l'aide d'une pénalisation ℓ_1 sur les gradients de forte amplitude ; ce qui peut notamment être réalisé à l'aide de la fonction $\ell_2|\ell_1$ de Hubert :

$$\begin{array}{rcll} \psi_\epsilon^{\text{Hubert}}(p) & = & |p| & \text{si} \quad |p| > \epsilon^2 \\ & = & \frac{p^2}{2\times\epsilon^2} + \frac{\epsilon^2}{2} & \text{si} \quad |p| \leq \epsilon^2 \end{array} \ , \tag{1.25}$$

où le seuil ϵ est généralement fixé en fonction de la dynamique globale ou locale des contours. Ces observations concernant les propriétés renforcées par les fonctions de potentiel envisagées, sont souvent valables dans le cas des estimateurs bayésiens de type maximum a posteriori (MAP) avec un terme d'attache aux données classique (Gauss,Poisson, Rayleigh). Elles ne s'appliquent cependant pas au cas des estimateurs de type moyenne a posteriori (PM) : en effet, on observe par exemple que l'attaque linéaire en zéro de la fonction $\psi^{\ell_1}(p)$ ne favorise pas plus que cela les images constantes par morceaux pour le cas de l'estimateur de type moyenne a posteriori (il faut considérer un départ plus abrupte pour cela).

Pour pallier le fait que la fonction de Hubert n'est pas deux fois continuement différentiable (C^2) en son raccordement, ce qui peut se révéler gênant pour l'emploi d'algorithmes d'optimisation MAP différentiable performants, on utilisera plutôt la fonction

$$\psi_\epsilon^{\ell_2|\ell_1}(p) \quad = \quad |p| - \epsilon^2 \times \log\left(1 + \frac{|p|}{\epsilon^2}\right) \ . \tag{1.26}$$

Lorsque ϵ est petit devant la dynamique des paramètres, les estimateurs obtenus en utilisant $\psi_\epsilon^{\ell_2|\ell_1}(p)$ constituent notamment une bonne approximation de ceux obtenus avec $\psi^{\ell_1}(p)$. De plus, la fonction peut être utilisée pour obtenir rapidement une approximation de l'estimateur MAP composé de $\psi^{\ell_1}(p)$, en diminuant progressivement la valeur du seuil ϵ (car cela permet

notamment de renforcer la convexité du critère pour les premiers temps de l'optimisation). La fonction de régularisation

$$\Phi^{\mathrm{TV}}(\mathbf{a}) = \Phi^{\ell_1}_{\mathrm{grad}}(\mathbf{a}) \ , \tag{1.27}$$

plus connue sous le nom de variation totale de l'image [4], présente également l'inconvénient (dans le cas de l'estimateur MAP) d'entraîner une perte de contraste ϱ entre un objet homogène et le fond dans lequel il est plongé, qui est inversement proportionnelle à l'échelle de l'objet, c'est à dire le rayon de sa plus grande boule ouverte inscrite [5, 6]. Cela est particulièrement gênant pour le débruitage RSO car on souhaite pouvoir conserver les cibles de petite échelle sans trop atténuer leur contraste. D'autre part de nombreuses cibles radar brillantes présentent une granulosité que l'on souhaite conserver en n'introduisant pas de lissage excessif sur ces cibles brillantes. Cela peut être réalisé en imposant un taux de croissance de la fonction de potentiel qui soit suffisamment lent voire nul pour les forts gradients d'intensité par rapport à la croissance de Φ^{TV} (on pourrait aussi choisir de ne pas régulariser les cibles brillantes bien que cela n'ait pas été retenu).

D'un autre côté, il n'est pas non plus prouvé par la pratique que la transition de type ℓ_1 proche de zéro soit la plus adaptée pour le cas de l'estimateur PM, d'autant que cela dépend également du contenu des images considérées. Afin d'introduire une flexibilité permettant de répondre à tous ces enjeux on considérera donc la fonction de potentiel paramétrée

$$\begin{aligned} \psi^{\ell_\alpha \to \ell_0}_{\alpha,\epsilon,\zeta}(p) &= |p|^\alpha && \text{si} \quad |p| < \epsilon \\ &= c_1 \times \exp\left[-\frac{\zeta}{|p|+\zeta/2}\right] + c_2 && \text{si} \quad |p| \geq \epsilon \, , \end{aligned} \tag{1.28}$$

avec $c_1 = \alpha\epsilon^{\alpha-1} \times \frac{(\epsilon+\zeta/2)^2}{\zeta} \times \exp\left[\frac{\zeta}{\epsilon+\zeta/2}\right]$ et $c_2 = \epsilon^\alpha - \alpha\epsilon^{\alpha-1} \times \frac{(\epsilon+\zeta/2)^2}{\zeta}$. En effet α permet de ré-

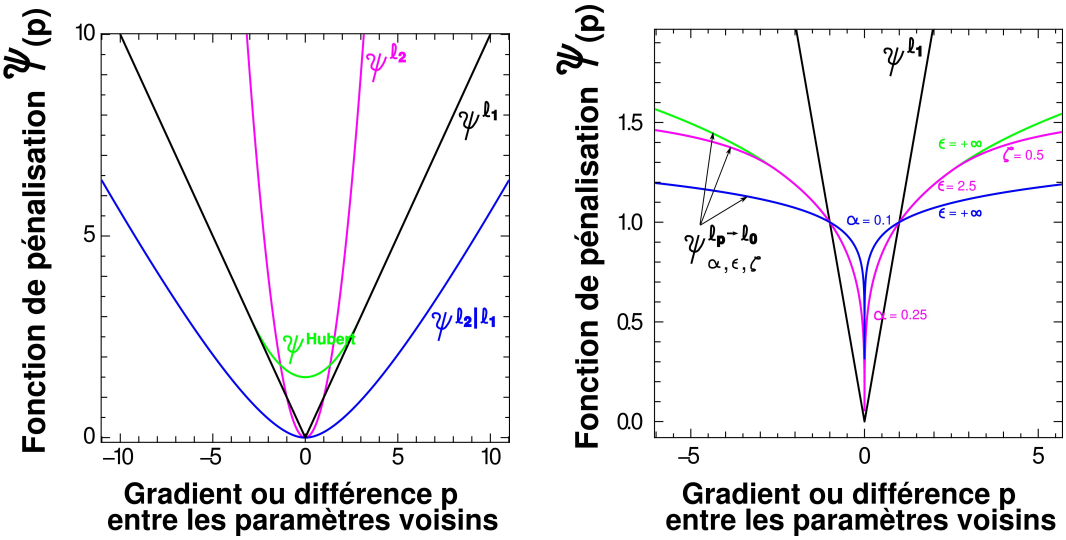


FIG. 1.3: Fonctions de pénalisation envisagées

gler la vitesse (norme) de pénalisation des faibles gradients, tandis que ζ sert à régler celle des

forts gradients de paramètres. Le seuil de transition entre les deux régimes peut également être réglé à l'aide du paramètre ϵ.

Pour récapituler ces choix, les différentes fonctions de pénalisation envisagées (avec des comportements intermédiaires entre ℓ^2 , ℓ^1 et ℓ^0) ont été représentées sur la figure 1.3.

1.5 Débruitage par moyennes non-locales

Bien que ce travail soit davantage focalisé sur les approches de débruitage opérant un lissage entre les voisinages locaux, il convient de mentionner l'approche consistant à considérer des a priori non-locaux. Cela paraît notamment tout particulièrement bien adapté pour le cas du débruitage d'images contenant de nombreux motifs et textures répétitifs. Buades et al. [7] proposent ainsi un algorithme de débruitage additif gaussien qui consiste à moyenner les pixels similaires :

$$\hat{a}_i = \frac{\sum_j w_{i,j} \times d_j}{\sum_j w_{i,j}}\,, \tag{1.29}$$

où $w_{i,j}$ dépend de la similarité entre les données i et j. La mesure de similarité est effectuée par une comparaison entre les pixels des données localisées dans des patches $\mathcal{F}_i$ et $\mathcal{F}_j$, lesquels correspondent typiquement à des fenêtres 5×5 pixels centrées sur les pixels i et j respectivement. Dans le cas du débruitage multiplicatif, Deledalle [8] propose d'étendre cette approche en utilisant un estimateur de maximum de vraisemblance pondéré itératif, basé sur un critère a posteriori de similarité entre patches :

$$w_{i,j}^{(k)} = \Pr(\mathbf{a}_{\mathcal{F}_i}{=}\mathbf{a}_{\mathcal{F}_j}|\mathbf{d}, \hat{\mathbf{a}}^{(k-1)})^h\,. \tag{1.30}$$

On comparera donc les résultats obtenus par notre approche locale à ceux donnés par cette approche de référence pour le débruitage multiplicatif.

On peut par ailleurs vouloir définir une norme de gradient non-local :

$$\|\nabla \mathbf{a}_i\|_2 = \sqrt{\sum_{j\in\mathcal{V}_i} w_{i,j}(a_i - a_j)^2}\,, \tag{1.31}$$

en utilisant le critère de similarité non-itératif $w_{i,j}$ de Deledalle ; puis s'intéresser à l'estimateur moyen que l'on obtient en appliquant la fonction de pénalisation $\psi_{\alpha,\epsilon,\zeta}^{\ell_\alpha\to\ell_0}(p)$ sur le gradient non-local au lieu de l'appliquer sur le gradient local. On incluera donc également cette nouvelle approche non-locale dans nos comparaisons.

1.6 Débruitage par ondelettes

Une autre famille d'algorithmes de débruitage très employée repose sur la représentation parcimonieuse des images dans des bases orthonormales ou redondantes (comme les algorithmes de poursuite dans des dictionnaires [9] par exemple). Ces approches découlent des travaux

menés sur la décomposition des signaux non-stationnaires dans des bases d'ondelettes à la fin du XX$^{\text{ième}}$ siècle [10, 11, 12, 13, 14], ainsi que sur les techniques de seuillage de coefficients d'ondelettes [15, 16]. Ces dernières supposent que le bruit se répartit à toutes les échelles de l'analyse en ondelettes tandis que le signal est plus localisé dans la base d'ondelettes, ce qui est une hypothèse parfaitement justifiée pour le cas d'un bruit blanc gaussien, et reste souvent valable même lorsqu'on ne dispose pas de connaissance concernant le bruit affectant le signal d'intérêt. L'avantage de la représentation parcimonieuse des images dans des dictionnaires est de permettre la réduction des artefacts qui sont souvent associés à la décomposition des images dans une base d'ondelettes ou de paquets d'ondelettes (comme l'introduction de structures périodiques annulaires localisées par example).

Grâce à une localisation espace-échelle et à une variation de la résolution spatiale adaptée (augmentation de la résolution spatiale proportionelle à l'augmentation de la fréquence), les décompositions de l'image en ondelettes voire en paquets d'ondelettes (si on gagne à affiner la résolution à certaines fréquences) permettent une représentation des éléments périodiques et localisés de l'image qui est plus parcimonieuse et donc plus informative que la transformée stationnaire de Fourier et que la transformée de Gabor. En effet la transformée en ondelettes effectue une décomposition d'un signal $x(t)$ monodimensionnel sur des fonctions de base passe-bandes $h_{c,\tau}(t)=\frac{1}{\sqrt{c}}h(\frac{t-\tau}{c})$:

$$\mathrm{CWT}_x(c,\tau) = \int_{\mathbb{R}} x(t) h^*_{c,\tau}(t)\,\mathrm{d}t\,, \tag{1.32}$$

et représente donc schématiquement l'évolution du signal en fonction de l'échelle (dilatation/compression c) et de la position (translation τ). En pratique on considère le plus souvent des discrétisations dyadiques ($c_j=2^j$ et $\tau_{j,k}=k\mathrm{T}\times c_j$) pour les applications numériques de la transformée en ondelettes. Celle-ci est alors implémentée, selon un codage pyramidal ou un codage en banc de filtres avec des filtres passe-bas et passe-haut.

L'extension de la transformée d'ondelettes mono-dimensionnelle au cas bi-dimensionnel des images peut être formalisé à l'aide d'un troisième paramètre de rotation θ de l'ondelette mère :

$$\mathrm{CWT}^{2D}_{\vec{f}}(c,\vec{\tau},\theta) = \int_{\mathbb{R}^2} \vec{f}(\mathbf{x}) h^*_{c,\vec{\tau},\theta}(\mathbf{x})\,\mathrm{d}(\mathbf{x})\,. \tag{1.33}$$

Dans le cas des analyses isotropes les plus simples on considère souvent une décomposition multi-résolution selon une composante passe-bas et trois composantes de détail (verticale, horizontale et diagonale). Néanmoins des transformées directionnelles comme les courbelettes [17] ont été proposées pour permettre de mieux représenter les contenus directionnels (anisotropes) et étendus des images de type bords.

Dans le cadre du débruitage multiplicatif radar, Achim [18] propose un algorithme de seuillage bayésien par ondellettes qui consiste à estimer les coefficients d'ondelettes à l'aide d'un estimateur de type maximum a posteriori. Le modèle qui est considéré pour le bruit est log-normal tandis que l'a priori est formé à partir de distributions $\alpha-$stables. Les résultats obtenus seront donc également comparés avec la méthode de référence de Achim ainsi qu'avec une technique plus classique de seuillage dur (et invariant par translation [19]) dans une base d'ondelettes.

Chapitre 2

Calcul des estimateurs

2.1 Calcul de $\mathbf{a}^{\mathrm{MAP}}$ par optimisation différentiable

Je présente ici un algorithme d'optimisation différentiable que j'ai développé pour minimiser une fonction $f_{\mathrm{post}}(\mathbf{a})$ supposée non-linéaire, non-convexe, à valeurs réelles, et deux fois continuement différentiable (C^2) sur l'orthant strictement positif $\mathbb{R}^{N_a}_{>0}$:

$$\mathbf{a}^{\mathrm{MAP}} = \min_{\mathbf{a}\in\mathbb{R}^{N_a}_{>0}} f_{\mathrm{post}}(\mathbf{a})\,, \tag{2.1}$$

Je commence tout d'abord par rappeler quelques résultats et algorithmes dédiés à la minimisation d'une fonction non-linéaire et convexe f de grande dimension $N_a \ggg 1$, avant de présenter l'algorithme d'optimisation non-conxexe développé pour minimiser localement $f_{\mathrm{post}}(\mathbf{a})$, étant donné que celle-ci n'est pas convexe sur l'ensemble de son domaine de définition $D_{\mathbf{a}}$, comme nous venons de le voir.

2.1.1 Optimisation convexe

La convexité d'une fonction $f(\mathbf{a})$ de plusieurs variables et supposée C^2 :

$$\sum_{i,j} b_i \frac{\partial^2 f(\mathbf{a})}{\partial a_i \partial a_j} b_j \geq 0 \quad \forall\, \mathbf{a} \in D_{\mathbf{a}}\,,\, \forall\, \mathbf{b} \in \mathbb{R}^{N_a}\,, \tag{2.2}$$

où $D_{\mathbf{a}}$ est le domaine de définition de f, est une condition suffisante mais pas nécessaire pour garantir l'existence d'un minimiseur $\mathbf{a}^+$ de $f(\mathbf{a})$ sur $D_{\mathbf{a}}$ lorsque $D_{\mathbf{a}}$ est compact et lorsque le gradient — $\mathbf{g}(\mathbf{a}) \triangleq \left[\frac{\partial f(\mathbf{a})}{\partial a_i}\right]_{i\in[1,N_a]}$ — de f s'annule en $\mathbf{a}^+$:

$$\frac{\partial f(\mathbf{a}^+)}{\partial a_i} = 0 \quad \forall\, i \in [1, N_a]\,. \tag{2.3}$$

De surcroît, la stricte-convexité sur $D_{\mathbf{a}}$:

$$\sum_{i,j} b_i \frac{\partial^2 f(\mathbf{a})}{\partial a_i \partial a_j} b_j > 0 \quad \forall\, \mathbf{a} \in D_{\mathbf{a}}\,, \forall\, \mathbf{b} \in \mathbb{R}^{N_a}\,, \tag{2.4}$$

permet de montrer que ce minimiseur est unique sur $D_{\mathbf{a}}$.

Lorsque l'ensemble de définition de f n'est pas compact, comme par exemple $D_{\mathbf{a}}$=$\mathbb{R}^{N_a}_{>0}$, il suffit par ailleurs de montrer que f est bornée inférieurement sur son ensemble de définition pour obtenir les mêmes résultats. C'est pourquoi la convexité délimite le cadre d'application de nombreux algorithmes d'optimisation numérique rapides et performants [21, 22, 23].

Il est cependant intéressant de bien comprendre que la convexité n'est pas une condition nécessaire à l'existence, ni même à l'unicité d'un minimiseur sur $D_{\mathbf{a}}$. La condition nécessaire impliquant la convexité concerne en réalité les propriétés locales de minimisation de f : à savoir qu'il est nécessaire et suffisant que f soit convexe et que son gradient s'annule en un point $\mathbf{a}^+$, pour que ce point soit un minimiseur local de f (la fonction étant C^2 et bornée, la convexité s'étend en fait sur un voisinage de $\mathbf{a}^+$) . C'est la raison pour laquelle il est tout à fait raisonnable de considérer l'emploi d'algorithmes d'optimisation différentiable pour minimiser localement la fonction $f_{\text{post}}(\mathbf{a})$ dans le cadre du débruitage MAP RSO, moyennant quelques adaptations pour s'adapter à la non-convexité des fonctions $f_{\text{post}}(\mathbf{a})$ envisagées. La pertinence de l'estimation effectuée – en supposant que les modélisations retenues sont pertinentes —, dépendra alors du fait que la fonction $f_{\text{post}}(\mathbf{a})$ est multi-modale ou pas, et le cas échéant de l'initialisation de l'algorithme d'optimisation locale ainsi que du fait que les minima locaux sont plus ou moins éloignés (différents) du minimum global.

Les algorithmes standards d'optimisation convexe dédiés à la minimisation de $f(\mathbf{a})$ lorsque $N_a \ggg 1$ et lorsque la solution de cette minimisation n'est pas explicite — à cause de la non-linéarité de f notamment —, consistent en général à génèrer une suite de points

$$\begin{aligned} \mathbf{a}^{(k+1)} &= \mathbf{a}^{(k)} + \alpha^{(k)} \mathbf{s}^{(k)}\,, \\ &= \mathbf{a}^{(k)} + \delta\mathbf{a}^{(k)}\,, \end{aligned} \tag{2.5}$$

en se basant sur un modèle différentiable simplifié $m_k(\delta\mathbf{a})$ de $f(\mathbf{a}+\delta\mathbf{a})$, censé permettre la décroissance monotone du critère. Ils diffèrent ainsi essentiellement les uns des autres dans le choix de la direction de descente $\mathbf{s}^{(k)}$ et de la longueur de pas $\alpha^{(k)}$>0.

2.1.1.1 Algorithme de premier ordre

Un développement limité de Taylor de f à l'ordre 1 et au point $\mathbf{a}^{(k)}$:

$$f\left(\mathbf{a}^{(k)} + \delta\mathbf{a}\right) = f\left(\mathbf{a}^{(k)}\right) + \mathbf{g}^{(k)\top}.\delta\mathbf{a} + O\left(\|\delta\mathbf{a}\|\right) \tag{2.6}$$

conduit notamment au choix de la direction de descente de gradient

$$\mathbf{s}^{(k)} = -\mathbf{g}^{(k)}\,. \tag{2.7}$$

L'algorithme de *plus grande pente* consiste en particulier dans ce contexte à calculer

$$\alpha^{(k)} = \min_{\alpha>0} f\left(\mathbf{a}^{(k)} - \alpha\,\mathbf{g}^{(k)}\right) , \tag{2.8}$$

ce qui nécessite de pouvoir résoudre rapidement le problème de minimisation monodimensionnel par rapport à α. Dans le cas contraire il faut trouver une longueur de pas $\alpha^{(k)}$ qui permette une décroissance suffisante du critère à chaque itération.

Plus généralement, on démontre que la convergence globale peut notamment être obtenue pourvu que la direction de descente vérifie

$$\mathbf{g}^{(k)\top}.\mathbf{s}^{(k)}<0 \tag{2.9}$$

et que la longueur de pas vérifie les deux conditions de Wolfe [24, 25] :

$$\begin{array}{rcl} f\left(\mathbf{a}^{(k)}+\alpha^{(k)}\mathbf{s}^{(k)}\right) & \leq & f\left(\mathbf{a}^{(k)}\right)+c_1\,\alpha^{(k)}\mathbf{g}^{(k)\top}.\mathbf{s}^{(k)} \\ \nabla f\left(\mathbf{a}^{(k)}+\alpha^{(k)}\mathbf{s}^{(k)}\right)^{\top}.\mathbf{s}^{(k)} & \geq & c_2\,\mathbf{g}^{(k)\top}.\mathbf{s}^{(k)} \end{array} \tag{2.10}$$

où $0<c_1<c_2<1$. Pour trouver un $\alpha^{(k)}$ vérifiant ces deux conditions on peut alors recourir à différents types d'algorithmes de recherche en ligne comme la dichotomie ou bien utiliser une approximation quadratique majorante du critère monodimensionnel [26].

2.1.1.2 Algorithmes du second ordre

Un développement limité de Taylor de f à l'ordre 2

$$f\left(\mathbf{a}^{(k)}+\delta\mathbf{a}\right) = f\left(\mathbf{a}^{(k)}\right)+\mathbf{g}^{(k)\top}.\delta\mathbf{a}+\delta\mathbf{a}^{\top}.\mathbf{H}^{(k)}.\delta\mathbf{a}+O\left(\|\delta\mathbf{a}\|^2\right) \tag{2.11}$$

conduit au pas de Newton :

$$\delta\mathbf{a}^{(k)}_{\text{Newton}} = -\mathbf{H}^{(k)^{-1}}.\mathbf{g}^{(k)} \tag{2.12}$$

qui peut permettre de minimiser la fonction $f(\mathbf{a})$ beaucoup plus rapidement que les algorithmes de premier ordre grâce à l'information de courbure du critère, pourvu que le calcul de l'inverse de la matrice Hessienne

$$\mathbf{H}^{(k)} = \left[\frac{\partial^2 f\left(\mathbf{a}^{(k)}\right)}{\partial a_i \partial a_j}\right]_{i,j} \tag{2.13}$$

ne soit pas trop coûteux. En pratique le pas de Newton est calculé en grande dimension à l'aide d'un algorithme itératif de type Krylov (gradients conjugués linéaires) pour éviter l'inversion explicite de $\mathbf{H}^{(k)}$ en $O(N_a^3)$ opérations ainsi que son stockage. Malheureusement dès que la matrice $\mathbf{H}^{(k)}$ n'a pas une structure simple de type diagonale ou Toeplitz, il est souvent trop coûteux de calculer le pas de Newton en très grande dimension.

Les approches de *quasi-Newton* [27, 28, 29] consistent dans ce cas là à chercher des approximations de la matrice hessienne à l'aide de mises à jour basées sur la variation du gradient entre les itérations successives (équation de la sécante) :

$$\mathbf{g}^{(k+1)}-\mathbf{g}^{(k)} = \mathbf{H}^{(k)}.\left(\mathbf{a}^{(k+1)}-\mathbf{a}^{(k)}\right)+o\left(\left\|\mathbf{a}^{(k+1)}-\mathbf{a}^{(k)}\right\|\right) , \tag{2.14}$$

ou à l'aide d'une approximation parcimonieuse de $\mathbf{H}^{(k)}$ comme la décomposition de Cholesky par exemple. Les approches de *Newton inexact* [30, 31] consistent alternativement à tronquer le calcul itératif du pas de Newton ou à remplacer $\mathbf{H}^{(k)}$ par une matrice approchée $\mathbf{B}^{(k)}$ mieux structurée, et dont la répartition des valeurs propres est notamment plus regroupée (meilleur conditionnement) :

$$\mathbf{s}^{(k)}_{\text{Newton_inexact}} = -\mathbf{B}^{(k)^{-1}}.\mathbf{g}^{(k)} \tag{2.15}$$

2.1.1.3 Gradients conjugués non-linéaires

L'algorithme des gradients conjugués non-linéaires, initialement proposé par Fletcher [32]

$$\mathbf{s}^{(k)}_{\text{GCNL}} = -\mathbf{g}^{(k)} + \frac{\mathbf{g}^{(k)\top}.\mathbf{g}^{(k)}}{\mathbf{g}^{(k-1)\top}.\mathbf{g}^{(k-1)}} \times \mathbf{a}^{(k-1)} \tag{2.16}$$

s'inspire de la résolution itérative des systèmes linéaires d'équations par gradients conjugués linéaires.

Cette famille d'algorithmes permet de minimiser une fonction f non-linéaire et convexe en limitant les produits matrice-vecteur et apporte en général un meilleur taux de convergence que l'algorithme de plus grande pente (notamment lorsqu'on y adjoint un bon préconditionneur). Tout comme dans le cas des algorithmes de descente de gradient, la longueur de pas $\alpha^{(k)}$ doit permettre une décroissance suffisante du critère à chaque itération pour obtenir la convergence vers le minimiseur global de f ; ce qui peut être réalisé en pratique à l'aide des méthodes de recherche en ligne ou de région de confiance. Bien que le taux de convergence reste en général inférieur par rapport aux méthodes d'optimisation différentiable du second-ordre, ces algorithmes sont cependant plus faciles à mettre en œuvre et permettent en principe de s'intéresser à des problèmes de plus grande dimension parce que la quantité de mémoire nécessaire est plus réduite.

2.1.2 Optimisation non-convexe

Lorsque la fonction $f(\mathbf{a})$ est non-convexe sur $D_{\mathbf{a}}$, et que ses matrices de dérivées secondes $\mathbf{H}(\mathbf{a})$ sont mal-conditionnées, singulières et ont des valeurs propres négatives, les méthodes de *région de confiance* restent néanmoins valables et performantes pour minimiser $f(\mathbf{a})$, en association avec des approches de type quasi-Newton ou Newton tronqué notamment[21, 31]. La direction de recherche est alors obtenue en résolvant le problème contraint :

$$\begin{aligned} \mathbf{s}_{\text{TR}} &= \min_{\mathbf{s}} \ \frac{1}{2}\mathbf{s}^{\top}.\mathbf{B}.\mathbf{s} + \mathbf{g}^{\top}.\mathbf{s} \qquad \text{tel que } \|\mathbf{s}\|_{\mathbf{M}} \leq \Delta \\ &= \min_{\mathbf{s}} \ \ Q(\mathbf{s}) \qquad\qquad\quad \text{tel que } \|\mathbf{s}\|_{\mathbf{M}} \leq \Delta \end{aligned} \tag{2.17}$$

où $\mathbf{M}$ est la métrique de la région de confiance et $Q(\mathbf{s}){=}\frac{1}{2}\mathbf{s}^{\top}.\mathbf{B}.\mathbf{s} + \mathbf{g}^{\top}.\mathbf{s}{\simeq}\frac{1}{2}\mathbf{s}^{\top}.\mathbf{H}.\mathbf{s} + \mathbf{g}^{\top}.\mathbf{s} + o\left(\|\mathbf{s}\|^2\right)$ est une approximation quadratique locale des variations de f, qui est valable lorsque

$\|\mathbf{s}\|$ est *suffisamment* petite et lorsque $\mathbf{B}$ est une bonne approximation de $\mathbf{H}$. La longueur du pas $\alpha^{(k)}>0$ peut éventuellement être affinée dans un second temps pour que

$$f\left(\alpha^{(k)}\mathbf{s}_{\mathrm{TR}}^{(k)}\right) < f\left(\mathbf{s}_{\mathrm{TR}}^{(k)}\right), \tag{2.18}$$

tandis que la taille de la région de confiance $\Delta^{(k)}$ est mise à jour à chaque itération k en fonction de l'accord

$$\rho^{(k)} = \frac{f\left(\mathbf{a}^{(k+1)}\right) - f\left(\mathbf{a}^{(k)}\right)}{Q\left(\delta\mathbf{a}_{\mathrm{TR}}^{(k)}\right)} \tag{2.19}$$

entre la réduction effective de la fonction de coût $f(.)$ et la modélisation quadratique de cette réduction $Q(.)$ ayant servi à dériver le pas $\delta\mathbf{a}_{\mathrm{TR}}^{(k)}$.

La résolution du problème quadratique contraint 2.17 est généralement effectuée à l'aide de l'algorithme standard de Moré et Sorensen [33], qui est basé sur la reformulation lagrangienne :

$$\mathbf{s}_{\mathrm{TR}} = \min_{\mathbf{s}} \frac{1}{2}\mathbf{s}^{\top}.\left(\mathbf{B}+\lambda\mathbf{M}\right).\mathbf{s} + \mathbf{g}^{\top}.\mathbf{s}\,. \tag{2.20}$$

Celle-ci est minimisée par décomposition spectrale de $\mathbf{B}$ et de $\mathbf{M}$, avec $\lambda^{+}\times\left(\Delta - \|\mathbf{s}\|_{\mathbf{M}}^{+}\right)=0$ et $\lambda^{+}\geq 0$.

La masse de calcul associée à l'algorithme de Moré-Sorensen, le rend toutefois inenvisageable pour des problèmes d'optimisation de grande dimension N_a. Pour remédier à cela, une première approche proposée par Powell [34, 21] et dite de *la jambe de chien*, consiste à résoudre approximativement le problème contraint en le projetant dans un sous-espace formé par le pas de Cauchy

$$\mathbf{s}_{\mathrm{C}} \;=\; -\left(\frac{\mathbf{g}^{\top}.\mathbf{g}}{\mathbf{g}^{\top}.\mathbf{B}.\mathbf{g}}\right)\times\mathbf{g}\,, \tag{2.21}$$

et par le pas de quasi-Newton

$$\mathbf{s}_{\mathrm{QN}} \;=\; -\mathbf{B}^{-1}.\mathbf{g}\,. \tag{2.22}$$

Lorsqu'il est trop coûteux d'inverser $\mathbf{B}$, il semble alors judicieux de chercher à résoudre le problème contraint dans le sous-espace constitué de la direction $\mathbf{s}_{\mathrm{C}}$ (permettant de garantir les propriétes de convergence globale) et de la direction de Newton tronquée $\mathbf{s}_{\mathrm{NT}}$ donnée par quelques itérations des gradients conjugués linéaires (préconditionnés) à la place de $\mathbf{s}_{\mathrm{QN}}$. Ce sous-espace est par exemple proposé par [35] dans le cadre de la minimisation d'une fonction non-linéaire convexe de très grande dimension, en y ajoutant quelques directions du sous-espace de Krylov accumulées lors du calcul du pas de Newton tronqué. Par ailleurs dans le cadre de la minimisation d'une fonction de type vraisemblance a posteriori, Skilling et Bryan [37] puis Pichon et Thiébaut [38] proposent de découpler les réductions selon f_{data} et selon f_{prior} dans le sous-espace, afin d'améliorer le taux de convergence avec un moindre surcoût.

Un deuxième type d'approche permettant de remplacer l'algorithme standard de Moré et Sorensen en grande dimension, consiste à résoudre le problème contraint (2.17) de façon approximative dans un sous-espace de dimension plus importante et variable en fonction des

différentes itérations k. Steihaug et Toint [39, 40] proposent ainsi de résoudre le problème contraint dans un sous-espace croissant de Krylov (à l'aide des gradients conjugués), ce qui permet de contrôler la norme croissante du pas effectué et de s'arrêter sur la frontière de la région de confiance. Lorsque la matrice $\mathbf{H}_k$ est semi-positive définie, cette stratégie permet une décroissance qui est au moins à moitié aussi bonne que la décroissance optimale issue de la résolution exacte du problème contraint[41]. Malheureusement, la solution obtenue par cette approche dans le cas non-convexe est trop largement sous-optimale, lorsqu'une direction de courbure négative est rencontrée. De plus, l'emploi d'un préconditionneur différent à chaque itération k gêne considérablement le principe de norme croissante sur lequel est basé cette approche. Des travaux récents dont notamment [36] semblent résoudre efficacement ces problèmes ; toutefois ayant une plus grande pratique du premier type d'approche [42] je me suis plutôt concentré sur celle-ci.

Je considère donc le sous-espace formé par les directions de recherche

$$\mathbf{s}_1 = -\mathbf{B}_{\text{post}}^{-1}.\mathbf{g}_{\text{data}} \,, \tag{2.23}$$

$$\mathbf{s}_2 = -\mathbf{B}_{\text{post}}^{-1}.\mathbf{g}_{\text{prior}} \,, \tag{2.24}$$

$$\mathbf{s}_3 = \mathbf{s}_{\text{NT}} \,, \tag{2.25}$$

où $\mathbf{B}_{\text{post}}^{-1}$ est un préconditionneur de type diagonal obtenu par seuillage de la diagonale de la hessienne $\mathbf{H}_{\text{post}}$. La projection de l'approximation quadratique locale 2.17 dans le sous-espace $\mathbf{S}=\{\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3\}$ conduit au problème de minimisation contraint

$$\min_{\boldsymbol{\beta}} \frac{1}{2}\boldsymbol{\beta}^\top.\mathbf{H}'.\boldsymbol{\beta} + \mathbf{g}'^\top.\boldsymbol{\beta} \qquad \text{tel que } \|\boldsymbol{\beta}\|_{\mathbf{M}} \leq \Delta \,, \tag{2.26}$$

avec $\mathbf{H}'=\mathbf{S}^\top.\mathbf{H}.\mathbf{S}$ et $\mathbf{g}'=\mathbf{S}.\mathbf{g}$. Le problème contraint 2.26 peut alors être résolu très rapidement à l'aide de l'algorithme de Moré et Sorensen puisque sa dimension est très petite (N_β=3).

2.1.3 Optimisation contrainte par des bornes simples.

Le dernier aspect du problème d'optimisation 2.1 jusqu'ici laissé en suspens, concerne la positivité des paramètres. En effet comme la fonction a posteriori n'est pas définie mathématiquement sur $\mathbb{R}^{N_a}_{\leq 0}$, et étant donné que les paramètres n'ont un sens physique que s'ils sont positifs et finis, il convient de prendre également en compte les contraintes

$$0 < a_i \leq M \quad \forall\, i \in [1, N_a] \,, \tag{2.27}$$

lors du calcul de $\mathbf{a}^{\text{MAP}}$. De plus, il est également nécessaire de borner l'ensemble des paramètres pour pouvoir résoudre la majorité des problèmes d'optimisation pratiques, à cause des erreurs de calcul qui sont introduites par l'implémentation des algorithmes sur un calculateur en précision finie. Par exemple, les overflows survenant lors de l'évaluation de la fonction de coût ou de ses dérivées sont un exemple typique d'erreurs invalidant les algorithmes d'optimisation qui ne prennent pas en compte des contraintes de type borne. Mais la question est parfois éludée soit parce que les démonstrations de convergence des algorithmes prenant en compte ces contraintes sont plus complexes, soit parce que la solution du problème est

"suffisamment" éloignée des bornes pour que son calcul ne pose pas ce problème (ce qui nécessite donc que le problème d'optimisation soit "suffisamment" bien initialisé par rapport aux contraintes).

Pour simplifier le problème de contrainte de type bornes sur les paramètres, il est plus simple de le formuler avec des inégalités non-strictes sous la forme générale :

$$\begin{aligned}
\mathbf{a}^{\mathrm{MAP}} &= \min_{\mathbf{a}\in D_{\mathbf{a}}} f_{\mathrm{post}}(\mathbf{a}) && (2.28)\\
&= \min_{\mathbf{a}\in\mathbb{R}^{N_a}} f_{\mathrm{post}}(\mathbf{a}) \;\; \mathrm{tq} \;\; u_i \leq a_i \leq v_i \;\; \forall\, i \in [1, N_a] && (2.29)\\
&= \min_{\mathbf{a}\in\mathbb{R}^{N_a}} f_{\mathrm{post}}(\mathbf{a}) \;\; \mathrm{tq} \;\; c_i(\mathbf{a}) \geq 0 \quad \forall\, i \in \mathcal{I}\,, && (2.30)
\end{aligned}$$

avec dans le cas présent $c_{2k}(a_k)=a_k-\epsilon_+$ et $c_{2k+1}(a_k)=M-a_k$, $\forall\, k \in [0, N_a-1]$. Le paramètre de seuil positif ϵ_+ est choisit très petit devant le niveau de quantification des données, et permet notammer d'éviter des problèmes d'overflows. L'ensemble faisable considéré pour le problème d'optimisation s'écrit donc finalement $D_{\mathbf{a}}=\mathbb{R}^{N_a}_{[\epsilon_+,M]}$.

Les solutions locales de (2.30) sont définies par les conditions nécessaires d'optimalité de Karush-Kuhn-Tucker (KKT) :

$$\begin{aligned}
\nabla \mathrm{L}_{f_{\mathrm{post}}}\left(\mathbf{a}^+, {\boldsymbol{\lambda}_{\mathbf{c}}}^+\right) &= \mathbf{0}\,,\\
\lambda_{c_i} &\geq 0 \;\; \forall\, i \in \mathcal{I}\,,\\
\lambda_{c_i} \times c_i(\mathbf{a}) &= 0 \;\; \forall\, i \in \mathcal{I}\,,
\end{aligned}$$

où $\mathrm{L}_{f_{\mathrm{post}}}(\mathbf{a}, \boldsymbol{\lambda}_{\mathbf{c}}) = f_{\mathrm{post}}(\mathbf{a}) - \sum_{i\in\mathcal{I}} \lambda_{c_i} \times c_i(\mathbf{a})$ est le Lagrangien associé aux contraintes $c_i(\mathbf{a})$; et où les multiplicateurs de Lagrange λ_{c_i} caractérisent la sensibilité de la position du minimiseur par rapport aux contraintes. Dans le cas particulier des contraintes de type borne, on a :

$$\mathrm{L}_{f_{\mathrm{post}}}(\mathbf{a}) = f_{\mathrm{post}}(\mathbf{a}) - \sum_{k=0}^{N_a-1} \lambda_{c_{2k}}(a_k-\epsilon_+) + \lambda_{c_{2k+1}}(M - a_k)\,. \tag{2.31}$$

Pour définir les conditions nécessaires et suffisantes d'optimalité locale — analogues à $\mathbf{g}(\mathbf{a}^+)=\mathbf{0}$ et $\mathbf{H}(\mathbf{a}^+)$ positive définie en optimisation non contrainte— il est nécessaire d'introduire quelques notions supplémentaires. L'ensemble actif $\mathcal{B}(\mathbf{a})$ désigne l'ensemble des indices i où la contrainte est nulle (i.e. $a_{2k}=\epsilon_+$ ou bien $a_{2k+1}=M$) au point faisable $\mathbf{a}\in D_{\mathbf{a}}$:

$$\mathcal{B}(\mathbf{a}) = \left\{ i \in \mathcal{I} \,\middle|\, c_i(\mathbf{a})=0 \right\} . \tag{2.32}$$

L'ensemble des directions faisables de premier ordre est noté

$$\mathcal{P}(\mathbf{a}) = \left\{ \mathbf{b} \in \mathbb{R}^{N_a} \,\middle|\, \mathbf{b}^\top.\nabla c_i(\mathbf{a}) \geq 0\,,\, \forall i \in \mathcal{B}(\mathbf{a}) \right\} . \tag{2.33}$$

On définit finalement le cône critique :

$$\mathcal{C}(\mathbf{a}^+, {\boldsymbol{\lambda}_{\mathbf{c}}}^+) = \left\{ \mathbf{b} \in \mathcal{P}(\mathbf{a}^+) \,\middle|\, \mathbf{b}^\top.\nabla c_i(\mathbf{a}^+)=0\,,\; \forall\, i \in \mathcal{B}(\mathbf{a}^+) \text{ tq } \lambda_{c_i}>0 \right\} , \tag{2.34}$$

qui correspond aux directions adhérant aux contraintes d'inégalités actives. Le cône critique correspond ainsi à l'ensemble des directions pour lesquelles il reste indéterminé de dire si

f augmente ou diminue à partir des seules conditions de KKT. Grâce à ces définitions on montre qu'il est nécessaire et suffisant que $\left(\mathbf{a}^+, {\boldsymbol{\lambda}_\mathbf{c}}^+\right)$ vérifient les conditions de KKT et que

$$\mathbf{b}^\top.\nabla^2_{\mathbf{aa}}\mathrm{L}_f\left(\mathbf{a}^+, {\boldsymbol{\lambda}_\mathbf{c}}^+\right).\mathbf{b} > 0 \quad \forall\, \mathbf{b}{\neq}\mathbf{0} \in \mathcal{C}(\mathbf{a}^+, {\boldsymbol{\lambda}_\mathbf{c}}^+)\,, \tag{2.35}$$

pour que $\mathbf{a}^+$ soit un minimiseur local strict du problème d'optimisation contraint avec des inégalités. Dans le cas particulier des contraintes de type borne, on a $\nabla^2_{\mathbf{aa}}\mathrm{L}_f(\mathbf{a}){=}\nabla^2_{\mathbf{aa}}f_{\mathrm{post}}(\mathbf{a})$. L'ensemble des directions faisables du premier ordre s'écrit :

$$\mathcal{P}(\mathbf{a}) = \left\{\mathbf{b} \in \mathbb{R}^{N_a} \,\middle|\, b_i \geq 0 \text{ si } a_i = u_i \text{ et } b_i \leq 0 \text{ si } a_i = v_i\right\}\,, \tag{2.36}$$

et correspond à l'ensemble des directions qui déplacent localement $\mathbf{a}$ dans l'ensemble faisable. De plus, le cône critique correspond à

$$\mathcal{C}(\mathbf{a}^+, {\boldsymbol{\lambda}_\mathbf{c}}^+) = \left\{\mathbf{b} \in \mathbb{R}^{N_a} \,\middle|\, b_i = 0 \;\forall\, i \in \mathcal{B}(\mathbf{a}^+) \text{ tq } \lambda_{c_i}{>}0\right\}\,, \tag{2.37}$$

c'est à dire l'ensemble des directions qui ne déplacent pas $\mathbf{a}^+$ sur ses composantes où les contraintes sont actives et non-dégénérées.

De nombreuses approches algorithmiques ont été proposées pour vérifier les conditions d'optimalité en optimisation contrainte avec des inégalités, et leur emploi dépend notamment de la dimension du problème, de la précision souhaitée de la solution, ainsi que de la sensibilité du minimiseur par rapport aux contraintes. Mais leur efficacité dépend aussi très largement de l'ensemble de la trajectoire de descente effectuée ce qui rend la sélection d'une méthode pour un problème donné peu évidente.

On peut par exemple séparer ces approches selon que l'on cherche ou non à prendre en compte les contraintes lors de l'optimisation. Les méthodes d'*ensemble actif* consistent notamment à chercher à prédire l'ensemble des indices actifs $\mathcal{B}(\mathbf{a}^+)$ de façon à pouvoir effectuer une minimisation non-contrainte sur l'ensemble des indices inactifs. L'ensemble des paramètres inactifs est réactualisé à chaque itération pour permettre une résolution accrue du problème contraint. Malheureusement ces approches sont peu pratiques en grande dimension étant donné la complexité combinatoire qui leur est associée. Néanmoins la méthode des gradients projetés est une extension de ce type d'approche qui est particulièrement efficace en grande dimension pour des contraintes d'inégalité de type borne. Originellement la méthode du gradient projeté consiste à effectuer une descente de gradient non contrainte sur l'ensemble des indices inactifs (i.e. dont les composantes courantes sont strictement à l'intérieur du domaine borné), ainsi que sur l'ensemble des indices actifs dont la composante d'anti-gradient pointe vers une portion faisable :

$$\{i \in \mathcal{I} \;\mid\; a_i = u_i \text{ et } g_i(\mathbf{a}) < 0 \text{ ou } \quad a_i = v_i \text{ et } g_i(\mathbf{a}) > 0\}\,. \tag{2.38}$$

De plus, une fois que le pas est déterminé, les nouveaux paramètres sont reprojetés dans l'ensemble faisable, c'est à dire que les composantes exclues du domaine faisable par la mise à jour sont repositionnées sur les bornes. Plusieurs extensions peuvent être envisagées dans le cas de la minimisation non-linéaire avec une direction de descente différente du gradient. Une première approche consiste à calculer le pas de Cauchy contraint puis à fixer les composantes actives du point de Cauchy sur les bornes associées et à effectuer la minimisation

non-contrainte pour les autres indices (en reprojetant les composantes dépassant les bornes une fois le pas accompli). Cela est malheureusement trop coûteux en grande dimension et on se contente généralement de maintenir fixées les composantes actives dont l'anti-gradient pointe vers la sortie du domaine faisable. C'est cette approche que j'ai retenue pour imposer les contraintes de type borne. Bien que cette approche puisse poser des problèmes des cas de figures fortement dégénérés, elle fonctionne bien en pratique pour le débruitage radar, et les minimiseurs trouvés n'ont pas de contrainte active pourvu que la borne $\epsilon+$ soit inférieure au niveau de quantification. En effet cela peut se comprendre intuitivement par le fait que le modèle des données est séparable sur chaque pixel et tend vers l'infini sur sa borne inférieure constituant ainsi une barrière répulsive pour la borne inférieure. De plus la régularisation locale (lissage entre les paramètres voisins) introduit un biais vers l'intérieur du domaine faisable.

A l'inverse des méthodes d'ensemble actif, les approches de type *points intérieurs* et *barrières* consistent à prendre en compte explicitement les contraintes à l'aide d'une fonction de pénalisation sanctionnant le rapprochement vers la contrainte selon un coût croissant et tendant vers l'infini sur la frontière de la contrainte. Originellement ces approches consistaient à utiliser une pénalisation logarithmique sur les contraintes d'inégalité de façon à toujours rester à l'intérieur du domaine faisable, tout en décroissant progressivement le poids μ de pénalisation des contraites vers zéro en cours d'optimisation :

$$\min_{\mathbf{a},\mathbf{v}} \left[f(\mathbf{a}) - \mu \sum_{i=1}^{N_{\mathcal{I}}} log(v_i) \right] \quad \text{tq} \ , \ c_i(\mathbf{a}) - v_i = 0 \ , \ \forall \, i \in \mathcal{I} \, , \tag{2.39}$$

où les v_i sont des variables intermédiaires. Cependant les techniques modernes permettent désormais de considérer tout aussi bien des trajectoires passant par des points extérieurs au domaine faisable (lorsque la fonction y est bien définie).

Par ailleurs, lorsqu'on peut violer la contrainte et qu'il est bénéfique de l'encourager pour accélérer le cours de l'optimisation, on peut également envisager de ne pénaliser la violation de la contrainte qu'à l'extérieur du domaine faisable. Cela peut notamment être réalisé en ajoutant une fonction de pénalisation de type L1 à la fonction de coût initiale :

$$\psi(\mathbf{a}, \mu) = f(\mathbf{a}) + \mu \sum_{i \in \mathcal{I}} \max[0, -c_i(\mathbf{a})] \, . \tag{2.40}$$

Le poids μ est amené vers l'infini en cours d'optimisation de façon à ce que la solution satisfasse finalement la contrainte. On peut montrer qu'il existe un $\mu*$ associé aux paramètres de Lagrange au delà duquel les solutions locales du problème contraint sont des solutions locales de $\psi(\mathbf{a}, \mu)$. En pratique cette méthode est problématique à cause de sa non-différentiabilité en zéro, ce à quoi on peut notamment remédier avec une approche itérative de type Lagrangien augmenté :

$$\min_{\mathbf{a}} \left[f(\mathbf{a}) - \sum_{i \in \mathcal{I}} \lambda_i c_i(\mathbf{a}) + \frac{\mu}{2} \sum_{i \in \mathcal{I}} c_i^2(\mathbf{a}) \right] \quad \text{tq} \ \epsilon_+ \leq a_i \leq M \ , \forall i \in [1, N_a]. \tag{2.41}$$

comme proposé par Nocédal dans le package Lancelot pour les contraintes d'inégalités qui ne sont pas de simples bornes (celles-ci sont gérées plus simplement par gradients projetés).

La formulation primale-duale associée aux conditions KKT du problème d'optimisation contraint :

$$(\mathbf{a}^+, \boldsymbol{\lambda_c}^+) = \min_{\mathbf{a}\in\mathbb{R}^{N_a}} \left\{ \max_{\boldsymbol{\lambda_c}\in\mathbb{R}^{N_\mathcal{I}}_{>0}} \left[f(\mathbf{a}) - \sum_{i\in\mathcal{I}} \lambda_{c_i} \times c_i(\mathbf{a}) \right] \right\} \tag{2.42}$$

$$= \max_{\boldsymbol{\lambda_c}\in\mathbb{R}^{N_\mathcal{I}}_{>0}} \left\{ \min_{\mathbf{a}\in\mathbb{R}^{N_a}} [\mathrm{L}(\mathbf{a}, \boldsymbol{\lambda_c})] \right\}, \tag{2.43}$$

peut également servir à dériver des algorithmes d'optimisation avec contraintes d'inégalités. C'est par exemple le cas des approches de type points intérieurs suivant le chemin central primal-dual. Par ailleurs, Nocédal propose une adaptation de l'algorithme des Lagrangiens augmentés pour le cas des contraintes de type inégalités en utilisant l'approximation lisse et proximale suivante à chaque itération k :

$$\mathrm{L}'(\mathbf{a}, \boldsymbol{\lambda_c}, \hat{\boldsymbol{\lambda}}^{\boldsymbol{k}}_{\boldsymbol{c}}, \mu^k) = f(\mathbf{a}) - \sum_{i\in\mathcal{I}} \lambda_{c_i} \times c_i(\mathbf{a}) - \frac{1}{2\,\mu_k} \sum_{i\in\mathcal{I}} (\lambda_{c_i} - \hat{\lambda}^k_{c_i})^2 \tag{2.44}$$

où $\hat{\boldsymbol{\lambda}}^{\boldsymbol{k}}_{\boldsymbol{c}}$ est l'estimée des multiplicateurs de Lagrange à l'itération k. La maximisation du Lagrangien augmenté est quadratique et séparable en $\boldsymbol{\lambda_c}$ et peut donc être effectuée analytiquement à chaque itération k. Si bien que le problème contraint peut être réexprimé sous la forme non-contrainte "itérative" suivante dans $\mathbb{R}^{N_a}$:

$$\min_{\mathbf{a}\in\mathbb{R}^{N_a}} \left\{ f(\mathbf{a}) - \sum_{i\in\mathcal{I}_1} \left[\hat{\lambda}^k_{c_i} c_i(\mathbf{a}) + \frac{\mu^k}{2} c_i^2(\mathbf{a}) \right] - \sum_{i\in\mathcal{I}/\mathcal{I}_1} \left[\frac{1}{2\mu^k} \hat{\lambda}^{k\,2}_{c_i} \right] \right\}, \tag{2.45}$$

où $\mathcal{I}_1 = \left\{ i \in \mathcal{I} \,\middle|\, \frac{\hat{\lambda}^k_{c_i}}{\mu_k} \geq c_i(\mathbf{a}) \right\}$, et en remettant à jour les estimateurs de Lagrange par

$$\hat{\lambda}^{k+1}_{c_i} = \begin{cases} 0 & \text{si } \dfrac{\hat{\lambda}^k_{c_i}}{\mu_k} \leq c_i(\mathbf{a}) \\ \hat{\lambda}^k_{c_i} - \mu_k \times c_i(\mathbf{a}) & \text{sinon} \end{cases} \tag{2.46}$$

Pour conclure cette partie sur l'optimisation différentiable non-linéaire, il convient de rappeler que les algorithmes présentés ne fournissent que des solutions locales lorsque le critère à minimiser est multi-modal. Si la séparation des modes n'est pas trop compliquée on peut associer à ces approches des départs permettant d'échantillonner judicieusement l'espace des paramètres. Sinon, on peut recourir à des algorithmes d'optimisation combinatoire comme par exemple les coupures minimales sur graphe dans le cas de l'estimateur MAP, ou bien encore considérer des approches d'échantillonnage stochastique de type Monte-Carlo par Chaînes de Markov et recuit simulé.

2.2 Calcul de $\mathbf{a}^{\mathrm{MAP}}$ par coupure sur graphe

Dans la section précédente nous nous sommes intéressés au calcul de $\mathbf{a}^{\mathrm{MAP}}$ par des approches différentiables qui considèrent que $\mathbf{a}$ peut prendre ses valeurs dans un ensemble continu et

compact voire ouvert. Nous allons considérer ici des approches *combinatoires* se basant au contraire sur une discrétisation fermée de l'espace des paramètres, avec un pas de quantification δa_i que nous considérerons a priori constant pour l'ensemble des paramètres ($\delta a_i{=}q\ \forall\, i$). Le choix du q est déterminant puisque les solutions obtenues ne seront pas les mêmes, voire peu pertinentes si le q n'est pas suffisamment fin.

Par opposition avec les approches différentiables, les approches combinatoires présentent l'intérêt majeur de pouvoir trouver le minimum global discret (i.e. associé au pas de quantification q considéré) de la fonction à minimiser lorsque celle-ci possède plusieurs minima. Les critères MAP considérés pour le débruitage RSO étant non-convexes et donc possiblement multi-modaux, on voit donc tout l'intérêt d'utiliser ces techniques combinatoires pour le calcul de $\mathbf{a}^{\text{MAP}}$. Malheureusement, bien que le cas particulier du débruitage avec une fonction de pénalisation de type $\Phi^{\ell_1}_{\text{diff}}$ sur les différences d'amplitude entre pixels voisins, soit aujourd'hui parfaitement maîtrisé à l'aide de techniques combinatoires comme les algorithmes de *coupure sur graphe*, cela n'est pas le cas pour toutes les fonctions de pénalisation que nous souhaitons considérer. De plus on souhaite également pouvoir calculer des estimateurs de débruitage de type moyenne a posteriori. On s'intéressera donc également dans la section suivante aux algorithmes d'échantillonnage Monte-Carlo par Chaînes de Markov, qui permettent le calcul de tous les estimateurs et fonctions de régularisation que l'on souhaite envisager.

L'algorithme combinatoire le plus primitif, généralement désigné sous le nom d'*algorithme de la force brute*, consiste à tester et comparer toutes les valeurs que peut prendre la fonction à optimiser sur son ensemble de définition discret et fini. Si cet algorithme présente un intérêt en petite dimension ($N_a{<}5$), il devient cependant rapidement irréalisable avec l'augmentation du nombre de paramètres N_a et avec la diminution du pas de quantification $q{=}\delta a$ puisque sa complexité est proportionnelle à ${N_q}^{N_a}$ (où N_q est le nombre de niveaux envisagés). Etant donné le besoin qui existe pour optimiser des problèmes combinatoires de grande dimension (gestion de trafic routier, ferrovière ou aérien, de placement de composants électroniques, de paquets de bits sur internet ...), l'optimisation combinatoire a connu de nombreux développements au siècle précédent à travers la théorie des graphes notamment.

2.2.1 Coupure minimale sur un graphe

Je me restreins toutefois ici à la présentation des algorithmes de type coupure minimale sur un graphe, qui sont tout particulièrement adaptés pour le débruitage lorsque la fonction de pénalisation est un champ de Markov convexe (ce qui est le cas de $f_{\text{prior}}(\mathbf{a}){=}\beta \times \Phi^{\ell_1}_{\text{diff}}(\mathbf{a})$) et lorsque le terme d'attache aux données est séparable (i.e. $f_{\text{data}}(\mathbf{a}){=}\sum_{i=1}^{N_a} f_{\text{data}}(a_i)$). Pour reformuler le problème de minimisation de $f_{\text{post}}(\mathbf{a}){=}f_{\text{data}}(\mathbf{a}) + f_{\text{prior}}(\mathbf{a})$ en un problème de coupure minimale sur graphe, les pixels (sites $i{\in}[1, N_{\mathbf{a}}]$) et les valeurs possibles de ces pixels (labels $[q]_{l\in\mathcal{L}}$ tels que $a_i^l{=}q_l$) sont représentés par des noeuds $\{n_{i,l}\,,\ (i,l) \in [1, N_{\mathbf{a}}] \times\mathcal{L}\}$ sur un graphe $\mathcal{G}$. Ces noeuds sont reliés entre eux par des arcs valués positivement (capacités $c(n_{i,l}, n_{j,l'}){>}0$), qui dépendent directement des énergies considérées $f_{\text{data}}(\mathbf{a})$ et $f_{\text{prior}}(\mathbf{a})$ (cf figure 2.1). Le réseau de noeuds ainsi constitué est relié en ses deux extrémités à des noeuds terminaux appelés la source $\mathcal{S}$ et le puits $\mathcal{T}$. Une coupe sur ce graphe orienté est un ensemble minimal d'arcs séparant la source $\mathcal{S}$ et le puits $\mathcal{T}$, c'est à dire tel qu'il n'y ait plus de chemin orienté reliant $\mathcal{S}$ et $\mathcal{T}$. Le coût d'une coupe est alors égal à la somme des capacités des arcs

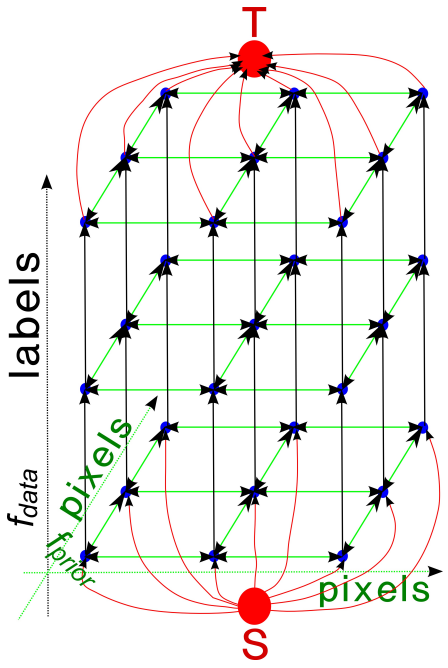


FIG. 2.1: Construction d'un graphe de type Ishikawa pour une fonction d'attache aux données séparable et une fonction de régularisation de type $\Phi^{\ell_1}_{\text{diff}}$.

constituant la coupe. Le problème de calcul de la coupe minimale, que l'on va par la suite associer au problème de minimisation de $f_{\text{post}}(\mathbf{a})$ à l'aide de la construction de graphe de Ishikawa, est résolu en pratique de façon équivalente par le calcul du flot maximum pouvant transiter sur le graphe $\mathcal{G}$.

2.2.2 Maximisation de flot

Le problème de maximisation de flot sur un graphe bénéficie d'algorithmes performants pour les applications de traitement du signal [43]. Pour comprendre le fonctionnement des algorithmes de maximisation de flot, je vais rappeler ici brièvement certains concepts qui leurs sont sous-jacents.

La valeur du flot est définie comme la quantité de flux sortant de la source $\mathcal{S}$ ou entrant dans le puits $\mathcal{T}$. La quantité de flot entrant dans un noeud est égale à la quantité de flot sortant de ce noeud à l'analogue des lois de Kirshoff en électricité. Un arc du réseau $\mathcal{G}$ est dit saturé lorsque le flot qui y transite est égal à la capacité de cet arc. Dans le cas contraire l'arc est caractérisé par sa capacité résiduelle c'est à dire la différence entre sa capacité et le flux qui y transite. Le flot maximum $F^{\max} \in \mathbb{R}_{>0}$ pouvant transiter entre la source et le puits est donc limité par les capacités des arcs du graphe $\mathcal{G}$. La capacité résiduelle d'un chemin $\mathcal{C}$ entre $\mathcal{S}$ et

$\mathcal{T}$ correspond au minimum des capacités résiduelles des arcs composant le chemin $\mathcal{C}$, c.a.d.

$$\min_{(n_{i,l},n_{j,l'})\in\mathcal{C}^2} \left[c(n_{i,l},n_{j,l'}) - F(n_{i,l},n_{j,l'})\right] . \tag{2.47}$$

On dit par ailleurs qu'on sature un chemin lorqu'on augmente le flot de ce chemin par sa capacité résiduelle (de telle sorte que la capacité résiduelle du chemin devient nulle). Lorsqu'on sature un chemin, la valeur du flot est donc accrue de cette même capacité résiduelle. Toutefois la valeur maximum du flot n'est pas forcément obtenue par saturation des chemins dans un ordre arbitraire, car la saturation d'un chemin dépend des saturations précédemment effectuées, et que certains chemins gagnent à être saturés plus tôt que d'autres.

Pour pallier ce problème on a recours aux concepts de réseau secondaire et de chemin augmentant. Le réseau secondaire $\mathcal{G}'$ associé à un flot F sur un réseau initial $\mathcal{G}$ permet de décroître le flot sur un des chemins $\mathcal{C}$ de $\mathcal{G}$ lorsque cela est nécessaire pour pouvoir augmenter le flot F sur $\mathcal{G}$. $\mathcal{G}'$ est définit en recopiant les noeuds du réseau initial $\mathcal{G}$ et en remplaçant les capacités de $\mathcal{G}$ par les capacités résiduelles $(c(n_{i,l},n_{j,l'}) - F(n_{i,l},n_{j,l'}))$. De plus, on ajoute sur $\mathcal{G}'$ des arcs inverses entre les mêmes noeuds, avec une capacité égale au flot transitant sur l'arc associé du réseau initial c.a.d. $F(n_{i,l},n_{j,l'})$.

Un chemin augmentant est un chemin dont la capacité n'est pas nulle sur le réseau résiduel. Si on définit un flot F' sur $\mathcal{G}'$ et que l'on ajoute à F, la composante de F' correspondant aux arcs communs tout en retranchant la composante correspondant aux arcs opposés, on montre en effet que F est augmenté. Cette fois-ci, il y a bien équivalence entre la saturation des chemins augmentant, la maximisation du flot F et le calcul de la coupe de coût minimum (cf. Ford et Fulkerson [44] pour la démonstration). Pour saturer les chemins augmentant et maximiser le flot il suffit donc de construire itérativement les réseaux secondaires résiduels, puis d'y trouver des chemins (à l'aide de l'algorithme de parcours en largeur "Breadth First Search" par exemple) et de les saturer.

2.2.3 Graphe d'Ishikawa

Pour associer les configurations de l'énergie a posteriori $f_{\text{post}}(\mathbf{a})$ à la valeur d'une coupe sur un graphe $\mathcal{G}$, j'ai utilisé la construction de graphe proposée par Ishikawa [45]. Celle-ci est une généralisation au cas multi-labels (i.e. avec plusieurs niveaux de gris) de la construction proposée en traitement d'images par Greig et al. [46] pour des labels binaires et des champs de Markov avec interactions du premier ordre entre voisins. La construction de graphe proposée garantit de trouver le minimum global à l'aide d'une coupe de coût minimale, lorsque le terme d'attache aux données est séparable (comme c'est le cas de la vraisemblance de Rayleigh) et lorsque la fonction de régularisation est convexe (comme c'est la cas de $\Phi^{\ell_1}_{\text{diff}}$).

La construction de graphe se présente de la façon suivante : le premier label de chaque pixel i (qui correspond au noeud $n_{i,l=1}$) est relié à la source $\mathcal{S}$ avec une capacité

$$c(\mathcal{S},n_{i,l=1}) = +\infty \ \ \forall\, i\,, \tag{2.48}$$

puis les noeuds correspondant à un même pixel i et à des labels adjacents l et l+1 sont reliés avec la capacité

$$c(n_{i,l},n_{i,l+1}) = f_{\text{data}}(q_l) \ \ \forall\, i \text{ et } l \in [1, n_l - 1] \tag{2.49}$$

Le noeud correspondant au dernier label $l{=}n_l$ de chaque pixel i est finalement relié au puits avec la capacité

$$c(n_{i,l=n_l}, \mathcal{T}) = f_{\text{data}}(q_{n_l}) \ \forall i. \tag{2.50}$$

Pour éviter que plusieurs noeuds correspondant au même pixel ne se trouvent dans la coupe et ainsi bien obtenir une configuration unique du champ de Markov, on ajoute les capacités de contraintes :

$$c(n_{i,l+1}, n_{i,l}) = +\infty \ \forall i \text{ et } l \in [1, n_l - 1]. \tag{2.51}$$

La maximisation de flot sur ce début de graphe permet d'obtenir les paramètres $\mathbf{a}^{\text{ML}}$ qui minimisent $f_{\text{data}}(\mathbf{a})$. Pour prendre en compte les interactions a priori entre pixels voisins et ainsi trouver les paramètres minimisant

$$f_{\text{post}}(\mathbf{a}) = f_{\text{data}}(\mathbf{a}) + \beta{\times}\Phi^{\ell_1}_{\text{diff}}(\mathbf{a}) \,, \tag{2.52}$$

on procède de la façon suivante : on ajoute des arcs bidirectionnels entre chaque pixel i et ses pixels voisins $j{\in}\mathcal{V}_i$ lorsque les labels l_i et l_j sont identiques (liens A/B, A/C, A/D et A/E selon les notations de la section 1.4) . Les capacités $c(n_{i,l}, n_{j\in\mathcal{V}_i,l})$ et $c(n_{j\in\mathcal{V}_i,l}, n_{i,l})$ de ces arcs valent donc $\beta{\times}\delta a$ pour les liens de type A/B ou A/C et valent $\beta{\times}\delta a/\sqrt{2}$ pour les liens diagonaux de type A/D ou A/E. De cette façon une coupe va également compter les capacités séparant deux configurations l^i et l^j de deux pixels voisins i et j appartenant à la coupe ; c'est à dire que la coupe compte à la fois f_{data} mais également la pénalité :

$$\begin{aligned} f_{\text{prior}}(a_i - a_j) &= \sum_{n=1}^{|l_i - l_j|} \beta{\times}\delta a \\ &= \beta \times |a_i - a_j| \end{aligned} \tag{2.53}$$

pour les liens de type A/B ou A/C par exemple.

Malheureusement les implémentations de maximisation de flot correspondant à cette construction de graphe sont très coûteuses en terme de mémoire vive (RAM) puisqu'elles nécessitent de stocker toutes les capacités établies entre les noeuds, ce qui représente un encombrement proportionnel à $N_{\text{a}}{\times}N_q$ avec l'algorithme de référence de Kolmogorov [43]. Pour des images de taille $N_{\text{a}}{=}500{\times}500$ pixels et un nombre de labels de $N_q{=}256$, cela implique par exemple de disposer d'une quantité de mémoire vive de l'ordre de 16×10^9 octets (Go), ce qui dépasse le standard équipant actuellement les calculateurs informatiques de type PC.

2.2.4 Coupures itératives

Pour remédier à ce problème d'encombrement mémoire, Loïc Denis a proposé de mettre en œuvre une approche de coupure sur graphe itérative, c'est à dire de considérer un nombre plus réduit de labels et d'itérer des coupures de type Ishikawa en affinant progressivement le pas de quantification $q{=}\delta a$. Après chaque itération (chaque nouvelle coupe) on construit ainsi un nouveau graphe, qui pour chaque pixel, prend en compte des labels équirépartis autour de la valeur optimale trouvée sur ce pixel lors de la coupure précédente. Le pas de quantification q est rétrécit progressivement (d'un facteur typiquement 2 à chaque nouvelle

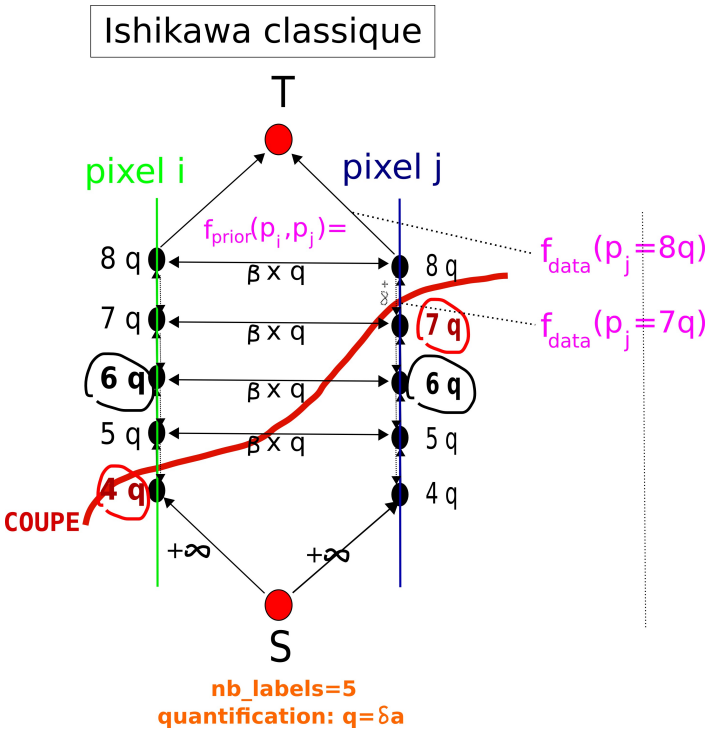


FIG. 2.2: Interactions entre pixels et labels pour un graphe de type Ishikawa classique.

itération). Cette approche n'est pour le coup plus exacte et l'on voit intuitivement qu'elle nécessite en principe pour fonctionner de ne pas se confronter au problème des minima locaux lors des premières itérations. C'est à dire que les minima locaux ne doivent pas avoir une échelle caractéristique trop large sauf à ce que le minimum global soit largement plus profond que les minima locaux. Cela dit il semble pertinent de tester cette approche pour les critères considérés car on s'attend à ce que ceux-ci ne soient pas très multi-modaux.

Pour mettre en œuvre l'algorithme d'Ishikawa itératif avec une fonction de type $\Phi^{\ell_1}_{\text{diff}}$, on voit qu'il devient nécessaire de prendre en compte les différences de valeur entre labels situés sur un même niveau l, lors de la construction du nouveau graphe à chaque itération. En effet chaque pixel est équiréparti autour d'une valeur en principe différente de son voisin contrairement à la construction originale de Ishikawa. La construction de graphe que j'ai mise en œuvre pour remédier à ce problème est représentée sur les figures 2.3 pour deux pixels voisins de type A/B ou A/C (cf. également la figure 2.2 pour la construction de graphe de Ishikawa classique).

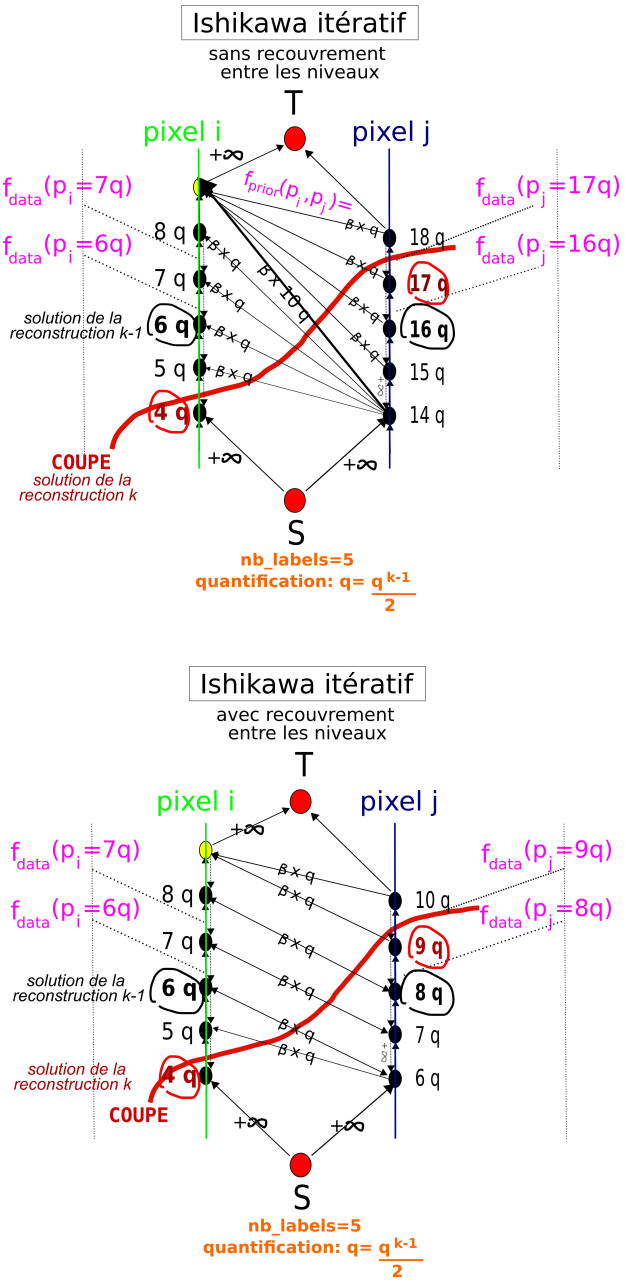


FIG. 2.3: Interactions entre pixels et labels pour un graphe de type Ishikawa itératif.

2.3 Optimisation stochastique

Les algorithmes de Monte-Carlo sont des algorithmes d'échantillonnage stochastique tout particulièrement dédiés à l'échantillonnage des distributions $\pi(\mathbf{a})$ de grande dimension N_a. Ils sont notamment mis en œuvre pour la résolution numérique de problèmes difficiles d'intégration (ne possédant pas de solution analytique) en exploitant la relation [1]

$$\mathcal{I} \triangleq \int_{D_\mathbf{a}} h(\mathbf{a}) \times \ell(\mathbf{a})\, \mathrm{d}\mathbf{a} = \lim_{n\to\infty} \frac{1}{n} \sum_{e=1}^{n} h(\mathbf{a}_e)\,, \tag{2.54}$$

lorsque les $\mathbf{a}_e$ sont des réalisations de la densité de probabilité

$$\pi(\mathbf{a}) = \frac{\ell(\mathbf{a})}{\int_{D_\mathbf{a}} \ell(\mathbf{a})\, \mathrm{d}\mathbf{a}} = \ell(\mathbf{a})/\mathcal{Z}\,. \tag{2.55}$$

On remarque que le calcul de $\mathbf{a}^{\mathrm{PM}}$ peut ainsi être effectué par un échantillonnage de type Monte-Carlo, en prenant tout simplement $h(\mathbf{a})$=$\mathbf{a}$ dans la relation (2.54), étant donné que l'on est capable d'évaluer ponctuellement $\pi(\mathbf{a})$=$\mathcal{D}_{\mathrm{post}}(\mathbf{a}|\mathbf{d};\beta)$ à une constante près $\mathcal{Z}$, $\forall \mathbf{a} \in D_\mathbf{a}$.

2.3.1 Echantillonnage Monte-Carlo mono-dimensionnel

La technique la plus primitive pour échantillonner une distribution monodimensionnelle $\pi(a)$ lorsqu'on est capable de calculer $\ell(a)$ $\forall a \in D_a$, consiste à échantillonner uniformément l'espace D_a puis à calculer $\ell(a_e)$ et $\hat{\mathcal{Z}}$=$\sum_{e=1}^{N_e} \ell(a_e)$, de façon à obtenir $\hat{\pi}(a_e)$=$\ell(a_e)/\hat{\mathcal{Z}}$. En se servant d'une source pseudo-aléatoire de bits on peut alors obtenir des échantillons de $\pi(a_e)$. On voit néanmoins que le coût de cette technique la rend inenvisageable en grande dimension N_a, puisqu'il faut alors effectuer $N_e^{N_a}$ évaluations de la fonction $\ell(\mathbf{a})$ pour estimer $\mathcal{Z}$ ($N_e \ggg 1$). Par ailleurs l'approximation effectuée en restreignant le nombre d'échantillons N_e considérés devient de plus en plus mauvaise avec l'augmentation de la dimension N_a, car l'ensemble typique contenant la masse principale de la distribution se raréfie exponentiellement par rapport à N_a.

Dans le cas où l'on dispose d'une distribution $\mathcal{Q}$, appelée distribution de proposition, que l'on peut d'une part évaluer ponctuellement à une constante près $\mathcal{Z}'$ telle que $\mathcal{Q}'(a)$=$\mathcal{Q}(a)/\mathcal{Z}'$, et dont on peut d'autre part facilement générer des échantillons $a_e^{\mathcal{Q}}$ semblant plus représentatifs de π que ceux donnés par la distribution uniforme, on peut néanmoins recourir à la technique d'échantillonnage d'importance. Celle-ci permet en effet d'estimer $\mathcal{I}$ plus rapidement que par

[1] L'erreur d'approximation de $\mathcal{I}$ pour n fini décroît proportionnellement à σ^2/n avec $\sigma^2 = \int_{\mathcal{D}_\mathbf{a}} (f(\mathbf{a}) - \mathcal{I})^2\, \ell(\mathbf{a})\, \mathrm{d}\mathbf{a}$ et est indépendante de la dimension de l'espace échantillonné (bien qu'il ne soit en pratique souvent pas aisé d'obtenir des échantillons indépendants en grande dimension).

échantillonnage uniforme en calculant

$$\hat{\mathcal{I}} = \frac{\sum_{e=1}^{N_e} \omega_e \times f(a_e^{\mathcal{Q}})}{\sum_{e=1}^{N_e} \omega_e}, \tag{2.56}$$

où les échantillons sont pondérés par $\omega_e = \ell(a_e^{\mathcal{Q}})/\mathcal{Q}'(a_e^{\mathcal{Q}})$. La convergence de $\hat{\mathcal{I}}$ vers $\mathcal{I}$ s'obtient de façon générale à condition que $\mathcal{Q}'(a)$ soit non nulle partout où $\ell(a)$ ne l'est pas, mais en pratique il est également nécessaire que $\mathcal{Q}$ ait la même allure que π sur l'ensemble de définition D_a pour que la technique soit efficace. Cette méthode n'est également pas envisageable en grande dimension car les différences entre $\mathcal{Q}(\mathbf{a})$ et $\pi(\mathbf{a})$ se répercutent également de façon exponentielle sur le nombre d'évaluations de fonctions.

Pour diminuer les ralentissements de calcul de $\mathcal{I}$ liés aux pondérations ω_e de l'échantillonnage d'importance, on peut utiliser la technique d'échantillonnage par réjection, lorsque la distribution de proposition $\mathcal{Q}$ peut être évaluée à une constante près $\mathcal{Z}$' telle que $\mathcal{Q}'(a) > \ell(a)$ $\forall a \in D_a$. En effet le fait que la distribution soit recouvrante permet d'échantillonner $\pi(a)$ en tirant un nombre aléatoire c_e uniformément sur $[0, \mathcal{Q}'(a_e)]$ et en acceptant l'échantillon si $c_e \leq \ell(a_e)$ (voir figure 2.4). La rapidité de cette technique d'échantillonnage dépend de la proximité du recouvrement de ℓ par $\mathcal{Q}'$. Cela se répercute une nouvelle fois de façon expo-

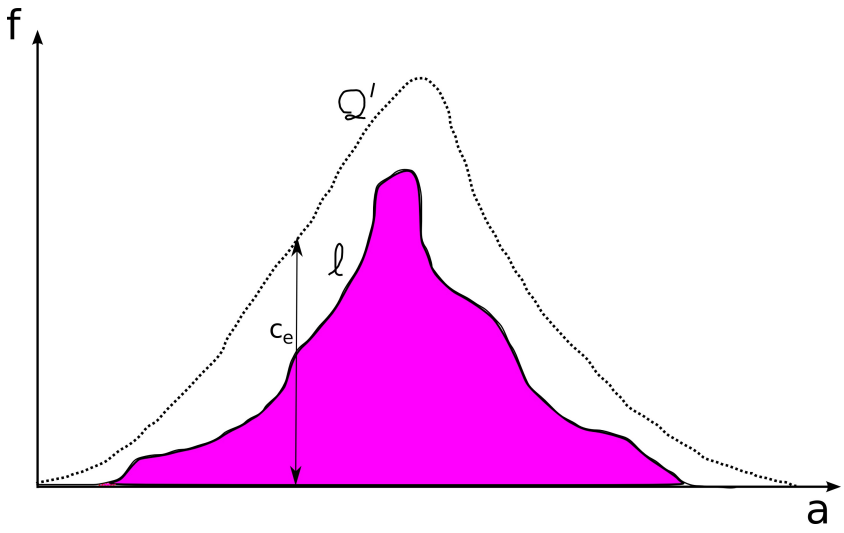


FIG. 2.4: Echantillonnage Monte-Carlo monodimensionnel par réjection

nentielle sur le nombre d'évaluations impliqué en grande dimension et rend la technique le plus souvent inappropriée pour ces cas-là, d'autant que la dimension augmentant, on aura tendance à trouver des approximations de plus en plus éloignées selon une même direction a_i pour pouvoir satisfaire la propriété de recouvrement.

2.3.2 Echantillonnage en grande dimension par chaînes de Markov

Comme on vient de le voir les méthodes d'échantillonage par importance et par réjection sont trop lourdement pénalisées en grande dimension par une proximité insuffisante entre la densité de proposition $\mathcal{Q}(\mathbf{a})$ et la densité cible $\pi(\mathbf{a})$. Pour y remédier les méthodes stochastiques de Monte Carlo par chaîne de Markov (MCMC) utilisent une densité de proposition $\mathcal{Q}(\mathbf{a})$ dépendant des états visités précédemment selon un processus conditionnel Markovien

$$\mathcal{Q}(\mathbf{a}^{(k)}|\mathbf{a}^{(k-1)},...,\mathbf{a}^{(2)},\mathbf{a}^{(1)}) = \mathcal{Q}(\mathbf{a}^{(k)}|\mathbf{a}^{(k-1)})\,. \tag{2.57}$$

La densité de probabilité que suit l'échantillon k est ainsi reliée à celle que suit l'échantillon $k-1$ par la relation de *chaîne de Markov* :

$$\Pr(\mathbf{a}^{(k)}) = \int_{D_{\mathbf{a}}} \mathcal{T}(\mathbf{a}^{(k)};\mathbf{a}^{(k-1)})\ \Pr(\mathbf{a}^{(k-1)})\,\mathrm{d}\mathbf{a}^{(k-1)}\,, \tag{2.58}$$

où $\mathcal{T}(\mathbf{a};\mathbf{a}')=\Pr(\mathbf{a}|\mathbf{a}')$ est la probabilité de transition caractéristique de la chaîne (probabilité conditionnelle par rapport à $\mathbf{a}'$).

D'une façon générale les méthodes d'échantillonnage MCMC consistent à construire une chaîne de Markov dont la probabilité de transition $\mathcal{T}(\mathbf{a};\mathbf{a}')=\Pr(\mathbf{a}|\mathbf{a}')$ accepte $\pi(\mathbf{a})$ pour invariant (propriété dite d'*invariance*) :

$$\pi(\mathbf{a}) = \int_{D_{\mathbf{a}}} \mathcal{T}(\mathbf{a};\mathbf{a}')\,\pi(\mathbf{a}')\,\mathrm{d}\mathbf{a}'\,. \tag{2.59}$$

Cette chaîne doit également être *ergodique* (moyenne temporelle égale à la moyenne d'ensemble statistique) avec $\pi(\mathbf{a})$ pour distribution stationnaire :

$$\lim_{k\to\infty}\ \Pr(\mathbf{a}^{(k)}) = \pi(\mathbf{a}) \quad \forall\ \Pr(\mathbf{a}^{(0)})\,, \tag{2.60}$$

et la probabilité de transition doit être positive et *irréductible* par rapport à $\pi(\mathbf{a})$, c'est à dire que tout état $\mathbf{x}$ de probabilité $\pi(\mathbf{x})$ non nulle doit à un moment pouvoir être atteint par la chaîne, quelle que soit son initialisation $\mathbf{y}$:

$$\forall(\mathbf{x},\mathbf{y}) \in D_{\mathbf{a}}^2\ \text{ tq }\ \pi(\mathbf{x})>0\,,\ \ \exists n\ \text{ tq }\ \mathcal{T}^{[n]}(\mathbf{x};\mathbf{y}) \triangleq \Pr(\mathbf{a}^{(n)}{=}\mathbf{x}|\mathbf{a}^{(0)}{=}\mathbf{y})>0\,. \tag{2.61}$$

En pratique les chaînes des algorithmes MCMC sont souvent réversibles, c'est à dire qu'elles vérifient la propriété dite d'*équilibre détaillé* :

$$\mathcal{T}(\mathbf{a}';\mathbf{a}) \times \pi(\mathbf{a}) = \mathcal{T}(\mathbf{a};\mathbf{a}') \times \pi(\mathbf{a}')\,, \tag{2.62}$$

ce qui implique la propriété d'invariance (2.59).

2.3.2.1 Algorithme de Metropolis-Hastings

Le premier algorithme de type MCMC est fréquemment attribué à Metropolis dans le cadre du projet américain de bombe nucléaire se déroulant lors de la seconde guerre mondiale. Cet

algorithme a été revisité plus en profondeur par Hastings, si bien qu'il est plus connu sous le nom d'algorithme de Metropolis-Hastings [48, 49]. L'échantillon candidat $\mathbf{a}^{\text{new}}$ proposé par la densité de proposition conditionnelle $\mathcal{Q}'(\mathbf{a}|\mathbf{a}^{(k-1)})$ à l'itération k est accepté (i.e. $\mathbf{a}^{(k)}$=$\mathbf{a}^{\text{new}}$), si le tirage aléatoire et uniforme d'un scalaire $c^{(k)}$ sur $[0;1]$ (i.e. $c^{(k)} \in \mathcal{U}_{[0;1]}$) est tel que

$$\zeta^{(k)}[\mathbf{a}^{\text{new}}, \mathbf{a}^{(k-1)}] \triangleq \frac{\pi\left(\mathbf{a}^{\text{new}}\right)}{\pi\left(\mathbf{a}^{(k-1)}\right)} \times \frac{\mathcal{Q}\left(\mathbf{a}^{(k-1)}|\mathbf{a}^{\text{new}}\right)}{\mathcal{Q}\left(\mathbf{a}^{\text{new}}|\mathbf{a}^{(k-1)}\right)} \geq \left(1 - c^{(k)}\right) . \tag{2.63}$$

Si au lieu de cela $c^{(k)} > \left(1-\zeta^{(k)}[\mathbf{a}^{\text{new}}, \mathbf{a}^{(k-1)}]\right)$, on accepte alors $\mathbf{a}^{(k)}$=$\mathbf{a}^{(k-1)}$ pour nouvel échantillon.

Contrairement à l'algorithme de réjection, l'algorithme de Métropolis-Hastings produit donc des échantillons corrélés (non-indépendants). Mais cet inconvénient est compensé par le fait que chaque évaluation de la fonction $\ell(\mathbf{a})$ conduise à accepter un nouvel échantillon par rapport à la méthode de rejection. Etant donné qu'il est nécessaire d'attendre un certain nombre d'itérations avant que les échantillons $\mathbf{a}^{(k)}$ soient distribués selon $\pi(\mathbf{a})$, on ne commence en pratique à sommer les échantillons pour calculer $\mathcal{I}$, qu'après un temps de chauffe k_{start}. De plus, il est parfois bénéfique de moyenner les échantillons avec un facteur de sous-échantillonnage, afin de diminuer la corrélation entre les échantillons visités par la chaîne et ainsi accélérer la convergence de $\mathcal{I}$.

La probabilité de transition caractéristique de la chaîne de Metropolis-Hastings s'écrit

$$\begin{aligned} \mathcal{T}^{\text{MH}}(\mathbf{a}^{(k)}; \mathbf{a}^{(k-1)}) \;&=\; \min\left\{1, \zeta^{(k)}[\mathbf{a}^{(\text{k})}, \mathbf{a}^{(k-1)}]\right\} \times \mathcal{Q}(\mathbf{a}^{(k)}|\mathbf{a}^{(k-1)}) \;+ \\ &\quad \delta_{\mathbf{a}^{(k-1)}}[\mathbf{a}^{(k)}] \times \textstyle\int_{D_{\mathbf{a}}} \max\left\{0, 1-\zeta^{(k)}[\mathbf{a}, \mathbf{a}^{(k-1)}]\right\} \mathcal{Q}(\mathbf{a}|\mathbf{a}^{(k-1)}) \;\text{d}\mathbf{a} \end{aligned} \tag{2.64}$$

où $\delta[.]$ représente la masse de Dirac. Il est possible de démontrer que la transition $\mathcal{T}^{\text{MH}}(\mathbf{a}; \mathbf{a}')$ est invariante, ergodique et irréductible par rapport à $\pi(\mathbf{a})$ pour un grand nombre de propositions $\mathcal{Q}(\mathbf{a}^{(\text{new})}|\mathbf{a})$, ce qui combiné avec sa simplicité de mise en œuvre pratique explique le large succès de l'algorithme de Metropolis-Hastings. Dans de nombreux cas la distribution de proposition est une distribution sphérique centrée sur l'échantillon courant (une gaussienne par example), car on ne dispose pas de proposition alternative semblant plus adaptée. L'échantillonneur résultant à un comportement de type marche aléatoire, ce qui présente l'avantage de permettre de ne pas s'immobiliser sur une zone densément probable comme par exemple lorsqu'il s'agit d'échantillonner une densité de probabilité fortement multimodale ; mais présente également l'inconvénient d'entraîner un très lent échantillonnage de l'espace des paramètres puisque l'échantillonneur ne se dirige pas préférentiellement dans les directions les plus probables.

2.3.2.2 Algorithme de Gibbs

Dans le cas particulier où les densités de probabilités conditionnelles $\pi\left(a_i|\mathbf{a}_{j\neq i}\right)$ associées à $\ell\left(\mathbf{a}\right)$ peuvent être échantillonnées facilement, il est plus efficace de recourir à l'algorithme d'échantillonnage de Gibbs. Celui-ci consiste à échantillonner successivement chacun des pa-

ramètres a_i lors d'une itération :

$$a_1^{(k)} \sim \pi(a_1|a_2^{(k-1)}, a_3^{(k-1)}, .., a_{N_a}^{(k-1)}) \quad (2.65)$$

$$a_2^{(k)} \sim \pi(a_2|a_1^{(k)}, a_3^{(k-1)}, .., a_{N_a}^{(k-1)}) \quad (2.66)$$

$$a_3^{(k)} \sim \pi(a_3|a_1^{(k)}, a_2^{(k)}, .., a_{N_a}^{(k-1)}) \ \ ..., \quad (2.67)$$

en acceptant donc tous les nouveaux échantillons par rapport à l'algorithme de Metropolis-Hastings.

2.3.2.3 Echantillonnage dynamique et algorithme de Monte-Carlo Hybride

Un des inconvénients généralement rencontré avec les algorithmes MCMC de type Metropolis-Hastings est la lenteur avec laquelle l'espace des paramètres est parcouru, ce qui est lié au comportement de type marche aléatoire, lequel ne permet pas de se diriger préférentiellement et donc rapidement vers les zones les plus densément probables. Pour remédier à cela Duane et al.[52] puis Neal [51] ont proposé d'exploiter la direction de gradient — qui pointe vers les zones plus densément probables — en s'inspirant des simulations statistiques qui sont effectuées en mécanique moléculaire pour décrire un ensemble de particules NVT (nombre de particules constant, volume constant et température constante) à l'aide d'une évolution NVE (nombre de particules constant, volume constant et énergie constante).

La dynamique moléculaire classique consiste à estimer les trajectoires (positions et vitesses) d'un ensemble de particules $\mathcal{S}=\{i\}$ en intégrant numériquement la relation fondamentale de la dynamique de Newton (seconde loi de Newton). Si on considère que les forces qui agissent sur les particules sont conservatives (pas de frottement), les équations dynamiques du mouvement se réécrivent dans l'espace des phases (espace permettant de décrire l'ensemble des trajectoires possibles et constitué des positions $\mathbf{p}_i=\{x, y, z\} \in \mathbb{R}^3$ ainsi que des quantités de mouvement ou impulsions $\mathbf{q}_i \in \mathbb{R}^3$) sous la forme :

$$\left.\begin{array}{rcl} \frac{\mathrm{d}\mathbf{p}_i(t)}{\mathrm{d}t} & = & \mathbf{q}_i(t)/m_i \\ \frac{\mathrm{d}\mathbf{q}_i(t)}{\mathrm{d}t} & = & -\frac{\partial \mathcal{V}[\mathbf{p}(t)]}{\partial \mathbf{p}_i(t)} \end{array}\right\} \ i \in \{1, ..., N_{\mathcal{S}}\} \quad (2.68)$$

où $\mathcal{V}[\mathbf{p}(t)]$ est la fonction d'énergie potentielle associée aux forces conservatives agissant sur $\mathcal{S}$, et où les masses m_i des particules ne dépendent pas du temps. Lorsque le système de particules $\mathcal{S}$ est isolé et que son énergie se conserve, l'équation 2.68 peut se réécrire de façon équivalente à l'aide du formalisme de Hamilton :

$$\left.\begin{array}{rcl} \frac{\mathrm{d}\mathbf{p}_i(t)}{\mathrm{d}t} & = & \frac{\partial \mathcal{H}[\mathbf{p}(t),\mathbf{q}(t),t]}{\partial \mathbf{q}_i(t)} \\ \frac{\mathrm{d}\mathbf{q}_i(t)}{\mathrm{d}t} & = & -\frac{\partial \mathcal{H}[\mathbf{p}(t),\mathbf{q}(t),t]}{\partial \mathbf{p}_i(t)} \end{array}\right\} \ i \in \{1, ..., N_{\mathcal{S}}\} \quad (2.69)$$

où l'énergie totale de $\mathcal{S}$ ou Hamiltonien $\mathcal{H}[\mathbf{p}(t), \mathbf{q}(t), t]$ est constante, et peut être décomposée selon la somme de son énergie potentielle et de son énergie cinétique :

$$\mathcal{H}[\mathbf{p}(t), \mathbf{q}(t), t] = \mathcal{V}[\mathbf{p}(t)] + \frac{1}{2}\mathbf{q}(t)^T.\mathrm{Diag}[\mathbf{m}]^{-1}.\mathbf{q}(t) \quad (2.70)$$

Les simulateurs numériques de dynamique moléculaire (intégrateurs de mouvement NVE sur un temps τ) sont généralement basés sur une succession de pas infinitésimaux δt permettant de garantir la conservation de l'énergie totale du système de particules $\mathcal{S}$, tout en conservant la métrique entremêlée de l'espace des phases (symplecticité). Le schéma de discrétisation le plus employé est la marche de Störmer-Verlet :

$$\mathbf{q}^{(j+\frac{1}{2})} = \mathbf{q}^{(j)} - \frac{\delta t}{2}\nabla_{\mathbf{p}}\mathcal{V}\left[\mathbf{p}^{(j)}\right], \tag{2.71}$$

$$\mathbf{p}^{(j+1)} = \mathbf{p}^{(j)} + \delta t \times \mathrm{Diag}[\mathbf{m}]^{-1}.\mathbf{q}^{(j+\frac{1}{2})}, \tag{2.72}$$

$$\mathbf{q}^{(j+1)} = \mathbf{q}^{(j+\frac{1}{2})} - \frac{\delta t}{2}\nabla_{\mathbf{p}}\mathcal{V}\left[\mathbf{p}^{(j+1)}\right], \tag{2.73}$$

qui correspond à une discrétisation de type *saute-mouton* et possède les propriétés de réversibilité et de symplecticité. Cette discrétisation ne garantit pas la conservation de l'énergie, ce qui est corrigé grâce à la règle d'acceptation de Metropolis.

Dans le cadre de l'estimation bayésienne et plus précisément de l'échantillonnage NVT de $f_{\mathrm{post}}(\mathbf{a})$, l'algorithme de Monte-Carlo hybride correspond à la marginalisation de la dynamique Hamiltonnienne par rapport aux vitesses initiales (quantité de mouvement). La position $\mathbf{p}$ correspond aux paramètres $\mathbf{a}$ que l'on cherche à estimer, et on introduit des variables auxiliaires $\mathbf{b}$ correspondant aux impulsions $\mathbf{q}$, de telle sorte que le Hamiltonien (énergie totale) s'écrit

$$\mathcal{H}(\mathbf{a},\mathbf{b}) = f_{\mathrm{post}}(\mathbf{a}) + \frac{1}{2}\mathbf{b}^T.\mathrm{Diag}[\mathbf{m}]^{-1}.\mathbf{b}\,, \tag{2.74}$$

où $f_{\mathrm{post}}(\mathbf{a})$ est l'anti-log vraisemblance du champ de Markov a posteriori considéré et correspond à la fonction d'énergie potentielle paramétrée en fonction du vecteur position, tandis que le second terme correspond à l'énergie cinétique dans le contexte Hamiltonien. Les équations décrivant l'évolution dynamique du sytème Hamiltonien (énergie totale constante) s'écrivent :

$$\frac{\partial\mathcal{H}}{\partial\mathbf{a}} = -\frac{\partial\mathbf{b}}{\partial t} \quad \text{et} \quad \frac{\partial\mathcal{H}}{\partial\mathbf{b}} = \frac{\partial\mathbf{a}}{\partial t}\,, \tag{2.75}$$

ce qui conduit à

$$\frac{\partial\mathbf{b}}{\partial t} = -g_{\mathrm{post}}(\mathbf{a}) \quad \text{et} \quad \frac{\partial\mathbf{a}}{\partial t} = \mathrm{Diag}[\mathbf{m}]^{-1}.\mathbf{b}\,. \tag{2.76}$$

En adoptant la discrétisation temporelle de type *saute-mouton* précédemment évoquée cela conduit à la dynamique approximativement Hamiltonienne

$$\mathbf{b}^{(j+\frac{1}{2})} = \mathbf{b}^{(j)} - \frac{\delta t}{2} \times \mathbf{g}_{\mathrm{post}}\left[\mathbf{a}^{(j)}\right], \tag{2.77}$$

$$\mathbf{a}^{(j+1)} = \mathbf{a}^{(j)} + \delta t \times \mathrm{Diag}[\mathbf{m}]^{-1}.\mathbf{b}^{(j+\frac{1}{2})}, \tag{2.78}$$

$$\mathbf{b}^{(j+1)} = \mathbf{b}^{(j+\frac{1}{2})} - \frac{\delta t}{2} \times \mathbf{g}_{\mathrm{post}}\left[\mathbf{a}^{(j+1)}\right]. \tag{2.79}$$

Cette discrétisation n'assure pas l'invariance de l'énergie totale, ce qui est corrigé en ayant recours à la règle d'acceptation de Metropolis [52].

Au final l'itération k de la chaîne consiste donc à tirer aléatoirement une quantité de mouvement ou impulsion $\mathbf{b}^{(k)}$ (pour effectuer la marginalisation par rapport à $\mathbf{b}$), puis à faire

évoluer le vecteur de quantité de mouvement $\mathbf{b}$ en fonction du gradient de $f_{\text{post}}(\mathbf{a})$ et à faire évoluer les paramètres $\mathbf{a}$ en fonction de $\mathbf{b}$, selon les principes de la mécanique Hamiltonienne 2.77. Après un petit nombre d'évolutions Hamiltoniennes n_j, la variation d'énergie totale $\Delta\mathcal{H}(\mathbf{a}, \mathbf{b})$ sert à décider de l'acceptation d'un nouvel échantillon, par comparaison avec le tirage aléatoire et uniforme d'un scalaire $c^{(k)}$ sur $[0; 1]$:

$$\mathbf{a}^{(k+1)} = \mathbf{a}^{\text{new}} \quad \text{si} \quad \exp\left[-\Delta\mathcal{H}(\mathbf{a}^{\text{new}}, \mathbf{b}^{\text{new}})\right] \geq \left(1 - c^{(k)}\right) \tag{2.80}$$

$$\mathbf{a}^{(k+1)} = \mathbf{a}^{(k)} \quad \text{sinon} \tag{2.81}$$

Pour augmenter les chances d'acceptation lorsque l'intégration du mouvement est plus longue, Chen et al. [53] ont proposé de sélectionner le candidat en fonction de l'ensemble de la trajectoire Hamiltonienne 2.77 effectuée, et d'utiliser la règle de Metropolis généralisée [54] de façon à satisfaire la propriété d'équilibre détaillé. Izaguirre et Hampton [55] puis Akhmatskaya et Reich [56] ont quand à eux proposé de considérer des énergies modifiées plus constantes par rapport à la discrétisation de Störmer-Verlet, et ont aussi proposé de modifier la mise à jour de la quantité de mouvement, de façon à augmenter les taux d'acceptation sur des temps d'intégration du mouvement plus long et pour les sytèmes contenant un plus grand nombre de particules. Bien que les approches d'échantillonnage dynamique soient séduisantes, elles restent cependant actuellement difficiles à mettre en œuvre sur des problèmes multi-modaux de grande dimension.

2.3.2.4 Recuit simulé

Le recuit simulé [63] est une technique d'optimisation basée sur l'échantillonnage MCMC et associée aux densités de probabilité s'écrivant sous la forme d'un potentiel de Gibbs :

$$\mathcal{D}(\mathbf{a}) = \frac{\exp\left[-\mathcal{V}(\mathbf{a})\right]}{\mathcal{Z}}, \tag{2.82}$$

ce qui est le cas de toutes les distributions a posteriori $f_{\text{post}}(\mathbf{a})$ que nous souhaitons envisager. Le recuit simulé s'inspire du procédé de recuit utilisé en métallurgie pour éliminer les impuretés de matériaux, par alternance de phases de recuit rapides avec des phases plus lentes de refroidissement. Il consiste à faire varier le paramètre de température T afin de rendre plus probables certaines transitions d'énergie, difficiles à effectuer pour la température T=1 :

$$\mathcal{D}_T(\mathbf{a}) = \frac{\exp\left[-f_{\text{post}}(\mathbf{a})/T\right]}{\mathcal{Z}(T)}, \tag{2.83}$$

où le terme de *température* est également employé par analogie avec la description statistique de la distribution de particules en fonction de leur énergie dans un système thermodynamique en équilibre.

Le recuit permet notamment de faciliter le calcul de $\mathbf{a}^{\text{MAP}}$ lorsque l'énergie $f_{\text{post}}(\mathbf{a})$ est non-convexe et multi-modale. Etant donné que le champ de Gibbs est localisé sur les minimiseurs globaux de l'énergie lorsque $T \to 0$, il est possible de démontrer la convergence asymptotique vers l'un de ces minimiseurs en faisant un échantillonnage exhaustif avec une décroissance

logarithmique de la température. Mais cela n'est pas d'une grande utilité pour le cas pratique de l'échantillonnage non-exhaustif en temps fini, si bien que le recuit simulé est souvent utilisé comme une heuristique, avec des schémas de décroissance plus rapide de la température, adaptés à chaque application ciblée ainsi qu'à la proximité tolérée par rapport au minimum global. De plus des phases de réaugmentation de la température gagnent parfois à être insérées dans le processus de refroidissement, lorsque l'énergie a posteriori est fortement multi-modale notamment.

D'autre part, le recuit peut également être utilisé comme une heuristique pour accélérer le calcul en temps fini des estimateurs de type moyenne a posteriori. En effet l'échantillonnage Monte Carlo par Chaînes de Markov avec T=1 risque notamment de rester bloqué dans certains puits d'énergie (le temps d'obtenir des taux d'acceptation suffisament élevés), tandis que des élévations locales de la température facilitent le franchissement des barrières d'énergie et une visite plus rapide de l'espace des paramètres. Néanmoins les propriétés théoriques des MCMC sont perdues étant données les ruptures périodiques dans la convergence de la chaîne de Markov vers la chaîne cible.

Pour pallier les insuffisances théoriques du recuit simulé, Marinari et Parisi [64] puis Geyer et Thompson [65] ont proposé d'utiliser la température comme une variable dynamique et statistique (recuit tempéré). Après mise à jour des paramètres, une nouvelle température $\mathrm{T_N}$ est proposée selon une distribution de proposition $q(\mathrm{T_N};\mathrm{T_C})$. Elle est acceptée en fonction de la règle de Metropolis sur le rapport de Hastings

$$\zeta = \frac{\pi_{\mathrm{post}}^{\mathrm{T_N}}(\mathbf{a}) \times h(\mathrm{T_N}) \times q(\mathrm{T_N};\mathrm{T_C})}{\pi_{\mathrm{post}}^{\mathrm{T_C}}(\mathbf{a}) \times h(\mathrm{T_C}) \times q(\mathrm{T_C};\mathrm{T_N})} \tag{2.84}$$

où $h(\mathrm{T})$ est un hyper a priori permettant de régler la probabilité de visite de chaque température (par rapport aux constantes d'énergie associées à chaque température). Neal [66] propose quand à lui d'utiliser une procédure de pondération par échantillonnage d'importance pour pallier l'emploi d'une distribution à une température différente. Pour remédier au manque de proximité entre les distributions, il considère une séquence de distributions où chaque membre diffère de son voisin seulement par une très légère variation de la température. Dans le cas de l'échantillonnage d'une distribution a posteriori, il préconise par ailleurs de remplacer la variation de la température par une variation du poids des a priori.

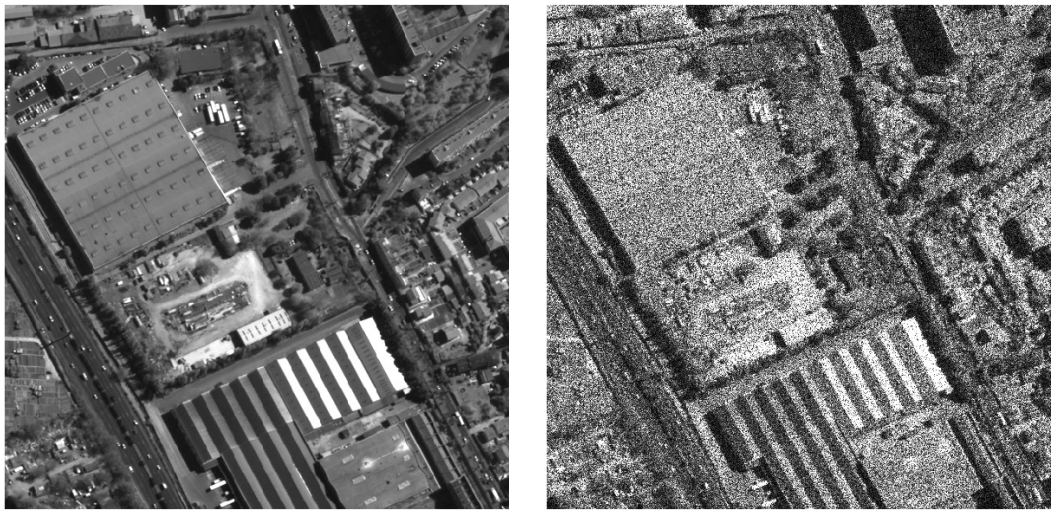
Chapitre 3

Base de données et critères de comparaisons

3.1 Données

Pour comparer les algorithmes de débruitage (i.e. les performances des différents algorithmes d'optimisation et estimateurs), on considère en premier lieu des données "synthétiques" auxquelles on incorpore artificiellement un bruit multiplicatif de Rayleigh. Cela nous permet en effet de pouvoir évaluer leurs performances sans être biaisé par l'inadéquation entre la modélisation statistique du bruit et le bruit effectivement présent dans un jeu particulier de données RSO. De plus, comme on dispose de la "vérité terrain" $\mathbf{a}^*$, on peut alors définir des critères quantitatifs de comparaison entre les différentes reconstructions, et également étudier leur variabilité lorsqu'on considère différentes réalisations de bruit. J'ai donc tout d'abord choisi une portion d'image optique haute résolution (600×600 pixels avec une résolution de l'ordre de 60 cm) acquise par le satellite Quickbird sur la ville de Londres (cf figures 3.1a, 3.1b). En effet les structures qui la composent sont en rapport avec le contexte satellite radar, bien que les deux types de données conservent des différences non négligeables qu'il ne faut pas oublier (dynamique des cibles brillantes et textures notamment).

Dans un deuxième temps on dispose de vraies données RSO multi-temporelles de Saint-Gervais acquises par le capteur Terra-Sar X. Il s'agit plus précisément de 10 acquisitions de la même portion de terrain (1024×1024 pixels avec une résolution de l'ordre de 1 m), qui ont été acquises à différentes dates entre Mai et Octobre pendant deux années de vol du satellite TerraSAR-X (figures 3.2c, 3.3c). On peut ainsi synthétiser une pseudo vérité terrain en moyennant l'intensité des images bruitées (figures 3.2a, 3.3a) ; ou bien en utilisant la technique de débruitage qui donne les meilleurs résultats et en prenant en compte les 10 réalisations bruitées de la même scène au lieu d'une seule dans la modélisation bayésienne(figures 3.2b, 3.3b). En effet le nombre de réalisations disponibles semble trop faible pour que leur moyenne suffise à supprimer entièrement le bruit. On dispose ainsi de vérités terrains approchées et complémentaires. En réalité la scène varie également entre les différentes réalisations tempo-



(a) image sans bruit (b) image bruitée

FIG. 3.1: Portions d'image satellite utilisées pour les comparaisons.

relles (présence de cibles ponctuelles comme les voitures, affaissement de terrain, évolution des glaciers et de la végétation), mais l'on négligera ces phénomènes dans un premier temps pour pouvoir mener des comparaisons qualitatives plus simples sur ces vraies données. On comparera les différents algorithmes de débruitage sur une réalisation bruitée effectivement acquise ainsi que sur une réalisation bruitée synthétique obtenue en incorporant du bruit multiplicatif à la pseudo vérité terrain.

On dispose par ailleurs pour effectuer les comparaisons, de vraies données RSO du CNES réechantillonnées en sous-bande. Le spectre d'une image radar RSO au format complexe n'est pas localisé autour de la fréquence nulle comme pour une image optique en amplitude et il contient l'information correspondant aux sous-ouvertures de l'antenne synthétique. On peut ainsi le découper en sous-bande contenant la même information, au prix d'une dégradation de la résolution spatiale. En réalité les images en sous-bande ne contiennent pas la même information pour les cibles dont la réfléctance n'est pas isotrope mais on négligera ce phénomène dans un premier temps pour mener des comparaisons qualitatives plus simples. Une image radar complexe RSO haute résolution (non disponible) a ainsi été découpée par le CNES en sous-bandes. On dispose d'une part de l'image des sous-bandes moyennées en intensité, qui constitue une sorte de vérité terrain (cf figure 3.4a). Et l'on dispose d'autre part d'une image en sous-bande (cf. figure 3.4b), qui constitue une réalisation bruitée de la scène à la même résolution ($\sim$1 m pour 409$\times$409 pixels). On comparera les différents algorithmes de débruitage sur la réalisation bruitée en sous-bande ainsi que sur une réalisation bruitée synthétique obtenue en incorporant du bruit multiplicatif à l'image moyennée (figure 3.4c).

(a) Moyenne en intensité des réalisations temporelles

(b) Image débruitée (PM) avec toutes les réalisations

(c) Réalisation temporelle

(d) Réalisation bruitée artificiellement

FIG. 3.2: Images radar RSO multi-temporelles de Saint-Gervais.

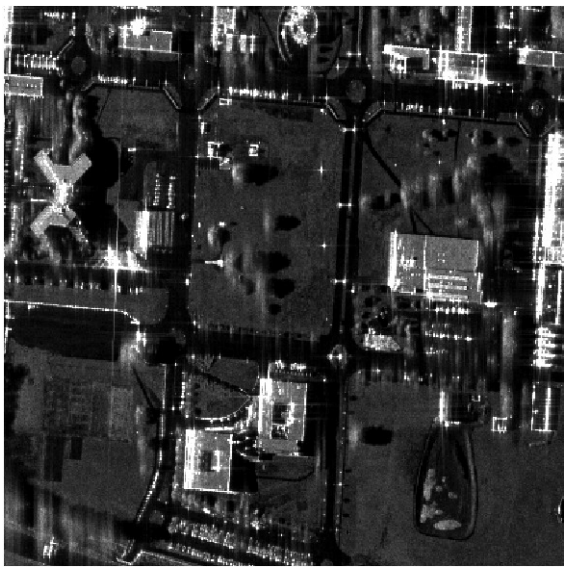
(a) Moyenne en intensité des réalisations temporelles

(b) Image débruitée (PM) avec toutes les réalisations

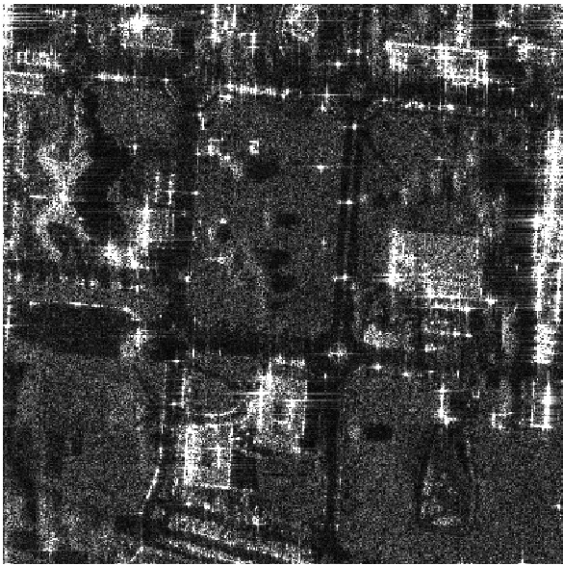
(c) Réalisation temporelle

(d) Réalisation bruitée artificiellement

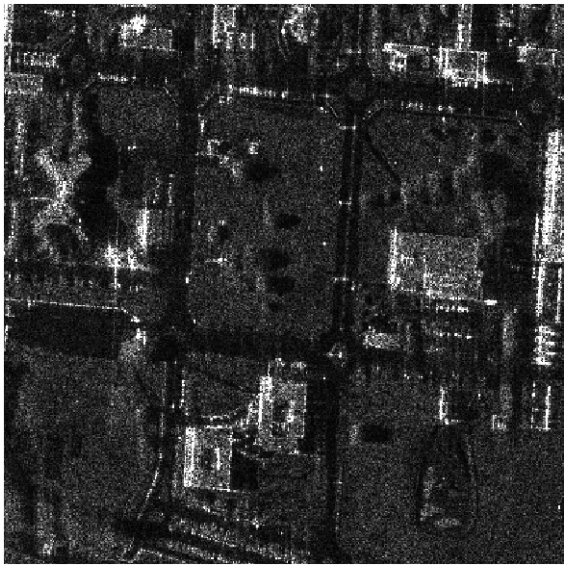
FIG. 3.3: Portions des images radar RSO multi-temporelles de Saint-Gervais.



(a) Moyenne en intensité des images sous-bandes



(b) Image en sous-bande



(c) Image bruitée artificiellement

FIG. 3.4: Images radar RSO de Toulouse.

3.2 Critères de comparaisons

3.2.1 Poids de la régularisation

Une des difficultés liée à la comparaison des algorithmes de débruitage envisagés est due au fait que l'hyper-paramètre β de réglage du poids des a priori, est un paramètre dont la valeur optimale (pour un niveau de bruit et pour un type d'image considéré) dépend également de la fonction de régularisation $\Phi(\mathbf{a})$ et du type d'estimateur $\mathbf{a}^{\mathrm{PM}}$ ou $\mathbf{a}^{\mathrm{MAP}}$ envisagés. Aussi pour choisir plus facilement β pour chacune des différentes méthodes, je me suis appuyé sur la mesure suivante

$$\vartheta = \sum_{i=1}^{N_a} \left(\frac{\hat{a}_i - d_i}{\hat{a}_i} \right)^2 , \tag{3.1}$$

qui permet d'estimer la quantité globale de lissage (résolution) de chaque estimateur $\hat{\mathbf{a}}$ par rapport aux mêmes données $\mathbf{d}$. En théorie les données de Rayleigh sont distribuées de telle sorte que

$$E\left[(a_i^* - d_i)^2\right] = \int_{\mathbb{R}_{>0}} (a_i^* - d_i)^2 \times \mathcal{V}_{\mathrm{data}}(d_i|a_i^*)\, \mathrm{d}d_i = \left(2 - \sqrt{\pi}\right) \times (a_i^*)^2 . \tag{3.2}$$

Mais vu que l'on ne dispose que d'une seule donnée d_i pour chaque a_i^*, on remplace le moyennage sur les réalisations par un moyennage spatial sur les pixels. Cela n'est pas du tout équivalent mais fournit en pratique un critère stable et fiable pourvu que le nombre de pixels soit suffisamment grand. J'ajuste donc β de telle sorte que

$$\vartheta = \eta \times \left[N_a \times \left(2 - \sqrt{\pi}\right) \right] , \tag{3.3}$$

où η est une constante proche de 1 qui peut au besoin être affinée pour chaque méthode de débruitage, en fonction du fait que la quantité de lissage optimale pour cette méthode semble plus ou moins importante.

3.2.2 Critères globaux

D'un point de vue assez général, on peut considérer que le débruitage RSO doit être capable de lisser les zones homogènes bruitées, tout en préservant les contours des objets et les cibles brillantes.

A ma connaissance il n'existe pas de critère idéal pour qualifier le débruitage multiplicatif (ni même additif), d'autant qu'il convient également de prendre en compte la perception et les biais du système visuel humain (sans parler de la subjectivité et de l'expérience du photo-interprète). Le critère de référence pour estimer les performances d'un estimateur bayésien $\hat{\theta}$, pour lequel le terme d'attache aux données $f_{\mathrm{data}}(\theta)$ est dédié à la minimisation de biais tandis que le terme d'a priori $f_{\mathrm{prior}}(\theta)$ est plus spécifiquement dédié à la minimisation de variance, est l'erreur quadratique moyenne

$$E\left[(\hat{\theta} - \theta^*)\right] = \mathrm{Biais}(\hat{\theta}, \theta^*) + \mathrm{Var}(\hat{\theta}) , \tag{3.4}$$

car elle permet de quantifier le compromis biais-variance effectivement opéré par l'estimateur bayésien. Dans le cadre du débruitage, ce critère semble donc pertinent puisqu'on souhaite justement améliorer le compromis biais-variance en diminuant la variance des données qui est liée au bruit, tout en conservant le plus possible d'information de l'image (biais). Malheureusement on ne dispose que d'une seule réalisation d_i pour estimer chaque paramètre a_i^*.

Il est alors tentant de remplacer la moyenne sur les réalisations par une moyenne sur les pixels, ce qui permet par la même occasion d'obtenir un critère moyen sur l'image. Néanmoins comme les données ne correspondent pas aux mêmes paramètres ($a_i^* \neq a_j^*$ en général lorsque $i \neq j$) et qu'elles correspondent d'autre part à des dégradations d'amplitude différentes à cause de la nature multiplicative du bruit, l'erreur quadratique calculée en moyennant les erreurs quadratique sur chaque pixels ne semble pas très pertinente. Une solution empirique consiste à calculer cette erreur sur chaque pixel pour différentes réalisations de bruit. Mais en pratique cette carte d'erreur dépend à la fois des données considérées et de l'estimateur considéré ce qui rend cette approche peu pratique. A défaut de critère idéal, il semble plus judicieux de normaliser l'erreur de reconstruction effectuée sur chaque pixel par la variance du bruit apparemment présente sur ce pixel. En effet on peut calculer la moyenne du bruit de Rayleigh sur chaque pixel

$$E\left[d_i\right] \triangleq \int_0^{+\infty} d_i \times \left(\frac{2\,d_i}{{a_i^*}^2} \times \exp\left[-\left(\frac{d_i}{a_i^*} \right)^2 \right] \right) \mathrm{d}d_i = \frac{\sqrt{\pi}}{2} \times a_i^* \tag{3.5}$$

puis sa variance

$$E\left[\left(d_i - E\left[d_i\right] \right)^2 \right] = \int_0^{+\infty} \left(d_i - \frac{\sqrt{\pi}}{2} a_i^* \right)^2 \times \left(\frac{2\,d_i}{{a_i^*}^2} \times \exp\left[-\left(\frac{d_i}{a_i^*} \right)^2 \right] \right) \mathrm{d}d_i = \left(1 - \frac{\pi}{4} \right) \times {a_i^*}^2 . \tag{3.6}$$

ce qui conduit au critère d'erreur suivant :

$$\nu_{\mathrm{g}}^{\mathrm{Err1}} = \sum_{i=1}^{N_a} \left(\frac{\hat{a}_i - a_i^*}{a_i^*} \right)^2 = \sum_{i=1}^{N_a} \left(1 - \frac{\hat{a}_i}{a_i^*} \right)^2 , \tag{3.7}$$

pour mesurer la qualité globale de la solution reconstruite $\hat{\mathbf{a}}$ dans le cas où on dispose de la vérité terrain $\mathbf{a}^*$. Etant donné que le paramètre de la distribution de Rayleigh décrivant les données en amplitude ne correspond pas à la moyenne de la distribution, contrairement au cas du paramètre de la distribution Gamma pour les données en intensité, on peut préférer utiliser un critère de type erreur quadratique normalisée en intensité :

$$\nu_{\mathrm{g}}^{\mathrm{Err2}} = \sum_{i=1}^{N_a} \left(\frac{\hat{a}_i^2 - {a_i^*}^2}{{a_i^*}^2} \right)^2 = \sum_{i=1}^{N_a} \left(1 - \frac{\hat{a}_i^2}{{a_i^*}^2} \right)^2 . \tag{3.8}$$

En pratique cela ne semble pas changer fondamentalement la donne car le paramètre est proportionnel à la moyenne dans le cas de la distribution de Rayleigh et que la constante associée semble compensée dans le rapport entre le paramètre et son estimée, mais vu que l'estimateur est non-linéaire on considérera néanmoins les deux critères.

Dans le cas du bruit additif, le rapport signal à bruit (PSNR) est souvent utilisé pour comparer différents algorithmes de débruitage :

$$\nu_{\mathrm{g}}^{\mathrm{PSNR}} = 10\times \log_{10}\left[\frac{e^2}{\sum_{i=1}^{N_a}(\hat{a}_i - a_i^*)^2}\right] . \tag{3.9}$$

où e correspond à la déviation standard ou au maximum de l'image (on prendra la déviation standard). Bien que ce critère ne semble pas particulièrement bien adapté au cas du bruit multiplicatif car le terme d'erreur quadratique au dénominateur ne tient pas compte du fait que la variance affectant chaque pixel varie en fonction du signal, on le considérera également en complément des autres critères. En effet le PSNR est tout de même censé être globalement d'autant plus élevé que le débruitage effectué est bon, même si la relation n'est sans doute pas exactement monotone.

Pour visualiser globalement l'information perdue et estimer la qualité effective du débruitage effectué, on représentera également le rapport $\mathbf{d}/\hat{\mathbf{a}}$ entre les données bruitées et l'image estimée. En effet, vu que l'on cherche à enlever un bruit multiplicatif dans les données, on s'attend à ce que ce rapport représente une image de bruit dans le meilleur des cas, ou bien des structures de l'image lorsque celles-ci n'ont pas été correctement débruitées.

3.2.3 Critères locaux

Dans le cas où on sait identifier une zone $\mathcal{Z}$ d'amplitude constante sur l'image on pourra calculer une estimation de l'erreur quadratique locale en moyennant les erreurs quadratiques sur cette portion d'image :

$$\nu_{\mathrm{loc}}^{\mathrm{Err}} = \sum_{i\in\mathcal{Z}}(\hat{a}_i - a_i^*)^2 . \tag{3.10}$$

Un des problèmes majeurs associé au débruitage statistique des images RSO est la perte de contraste qui est liée à l'emploi d'une fonction de régularisation. Aussi lorsqu'on dispose de l'image vérité terrain et qu'on identifie deux zones $\mathcal{Z}_1$ et $\mathcal{Z}_2$ qui sont constantes et connexes, on pourra comparer le contraste dans l'image non-bruitée

$$\rho_{\mathbf{a}^*}^{\mathcal{Z}_1,\mathcal{Z}_2} = 0.5 \times \frac{E\left[a^*_{\mathcal{Z}_1}\right]_{i\in\mathcal{Z}_1} - E\left[a^*_{\mathcal{Z}_2}\right]_{i\in\mathcal{Z}_2}}{E\left[a^*_{\mathcal{Z}_1}\right]_{i\in\mathcal{Z}_1} + E\left[a^*_{\mathcal{Z}_2}\right]_{i\in\mathcal{Z}_2}} , \tag{3.11}$$

avec celui de l'image débruitée $\rho_{\hat{\mathbf{a}}}^{\mathcal{Z}_1,\mathcal{Z}_2}$.

3.2.4 Performances des algorithmes d'optimisation

On a jusqu'à présent considéré des critères visant à qualifier la qualité du débruitage effectué. Néanmoins étant donné les difficultés mathématiques associées au calcul des différents estimateurs, il sera également intéressant de qualifier les performances des différents algorithmes

par rapport à leur objectif en terme d'optimisation. On s'intéressera notamment à la minimisation effective du critère $f_{\text{post}}(\mathbf{a})$ et aux coûts matériels impliqués (mémoire et temps de calcul).

Chapitre 4

Résultats des comparaisons et conclusions.

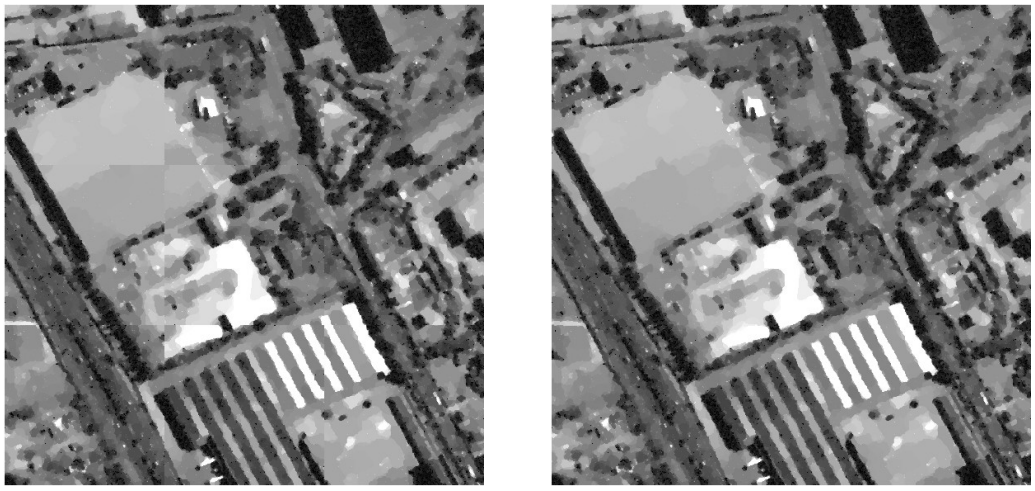
4.1 Comparaison entre les différents algorithmes d'optimisation MAP

4.1.1 Minimisation de $f_{\text{data}}(\mathbf{a}) + \beta\,\Phi_{\text{diff}}^{\ell_1}(\mathbf{a})$

4.1.1.1 Optimisation combinatoire

Comme on l'a vu dans la section 2.2, l'algorithme de coupure sur graphe de Ishikawa est tout particulièrement adapté à la minimisation de $f_{\text{post}}(\mathbf{a})=f_{\text{data}}(\mathbf{a}) + \beta\,\Phi_{\text{diff}}^{\ell_1}(\mathbf{a})$. Il garantit de trouver son minimum global pour le niveau de quantification q considéré. Mais comme la quantité de mémoire nécessaire pour échantillonner l'espace des paramètres avec 256 niveaux de gris sur une image 500×500 pixels s'élève à 16 Go (!), on s'attend cependant à devoir recourir à la technique de coupure sur graphe de Ishikawa itératif proposée à la sous-section 2.2.4, afin de pouvoir débruiter les images de notre base de données. Pour se convaincre que l'idée alternative de découper les images spatialement puis de raccorder les résultats pose de gros problèmes de continuité, il suffit en effet de regarder l'image 4.1, débruitée en considérant séparemment des portions 200×200 pixels de l'image satellite 3.1b (600×600 pixels).

Dans un premier temps j'ai donc étudié l'influence du choix du nombre de labels N_q sur la minimisation de $f_{\text{post}}(\mathbf{a})=f_{\text{data}}(\mathbf{a})+\beta\Phi_{\text{diff}}^{\ell_1}(\mathbf{a})$ effectivement obtenue en employant les méthodes de coupure sur graphe de Ishikawa classique puis de Ishikawa itératif. Pour pouvoir considérer un nombre de labels N_q tel que l'énergie $f_{\text{post}}(\mathbf{a})$ ne puisse plus diminuer d'une part et que les paramètres ne soient plus modifiés significativement d'autre part, j'ai considéré une portion 200×200 pixels de l'image satellite 3.1b seulement. Le pas de quantification q est déterminé pour la méthode de Ishikawa classique en effectuant le rapport entre la moyenne de l'image



(a) avec découpage spatial (b) sans découpage spatial

FIG. 4.1: Problèmes de continuité lors du raccordement de coupures sur graphes effectuées sur des portions d'image pour diminuer la quantité de mémoire employée.

bruitée + trois fois son écart-type avec le nombre de labels N_q considéré. Typiquement cela donne un pas de quantification de q=1.3 pour l'imagette considérée lorsque $N_q \simeq 2^{10}$ labels, puis q=5.2 pour $N_q \simeq 2^8$ labels et q=41.6 pour $N_q \simeq 2^5$ labels. Dans le cas des coupures sur graphe par la méthode de Ishikawa itératif, on itère les coupures en gardant N_q constant et avec un pas de quantification $q^{(k)}$ de plus en plus fin. Typiquement je prends $q^{(k)}=q^{(k-1)}/2$ en partant d'un premier découpage $q^{(0)}$ égal à celui considéré pour la méthode de Ishikawa classique. Le processus itératif est interrompu lorsque l'énergie et les paramètres ne varient plus significativement, ce qui correspond typiquement à $q^{(\text{last})} \sim 0.5$ sur les images considérées.

Les résultats obtenus sont reportés sur la figure 4.2. On constate tout d'abord que la technique de Ishikawa itératif proposée permet de diminuer la quantité de mémoire employée d'un facteur significatif de l'ordre de 10 sans perdre en qualité d'optimisation de $f_{\text{post}}(\mathbf{a})$. Ce bon comportement a également été vérifié sur d'autres portions d'images de la base de données, mais le fait de considérer cette méthode comme une référence sur des images de plus grande taille nécessite cependant d'extrapoler ce bon comportement, puisqu'on ne dispose alors pas de suffisamment de mémoire RAM pour effectuer une comparaison pratique d'une part, et parce que cette méthode ne possède pas de garantie théorique de convergence vers le minimum global de $f_{\text{post}}(\mathbf{a})$ d'autre part.

Etant donné que le pas de quantification final $q^{(\text{last})}$ considéré pour la méthode de Ishikawa itératif est plus fin que celui que l'on est en mesure de considérer ($q \simeq 1$ lorsque $N_q \simeq 1000$ labels) pour la méthode de Ishikawa classique, on constate même paradoxalement que la méthode itérative permet de minimiser légèrement mieux l'énergie $f_{\text{post}}(\mathbf{a})$ que celle de Ishikawa "exact". Toutefois cette différence n'est pas perceptible visuellement sur les images débruitées. En fait en y regardant de plus près les très légères différences ne sont pas simplement dues

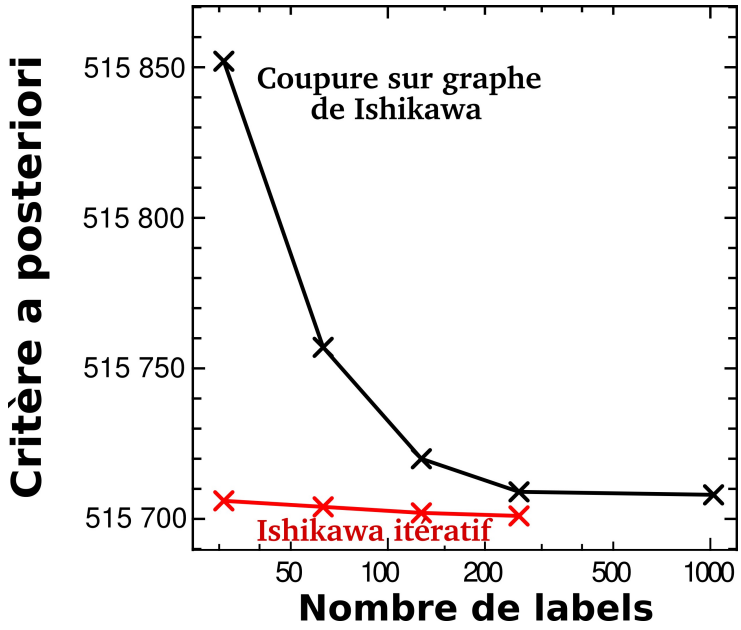


FIG. 4.2: Optimisation de $f_{\text{post}}(\mathbf{a})=f_{\text{data}}(\mathbf{a})+\beta\,\Phi^{\ell_1}_{\text{diff}}(\mathbf{a})$ par coupure sur graphe (Ishikawa classique et Ishikawa itératif) en fonction du nombre de labels N_q considérés.

au fait que la méthode itérative explore avec un niveau de quantification q plus fin, mais elles sont également liées au fait que la méthode itérative permet d'aller chercher des pixels dont la valeur optimale est plus élevée ou plus faible que ce qui peut être échantillonné par le graphe de Ishikawa classique.

On constate finalement que les variations de l'énergie $f_{\text{post}}(\mathbf{a})$ ne sont pas très importantes, même lorsqu'il existe des différences notables entre les paramètres reconstruits, ce qui suggère d'être numériquement vigilent dans la façon dont les fonctions constituant cette énergie et leurs dérivées sont calculées, quelle que soit la méthode d'optimisation envisagée.

D'un point de vue intuitif on s'attend à ce que la méthode de coupure sur graphe itérative ne fonctionne pas sur des critères multi-modaux dont les modes ont une échelle caractéristique très large dans l'espace des paramètres, sauf à ce que l'énergie de ces modes permette de distinguer les modes locaux du mode global quand l'échelle de quantification est large. En effet au fur et à mesure que le pas de quantification q est rétréci, certains labels et donc certaines portions de l'espace des paramètres ne peuvent plus être explorés, ce qui implique pour un bon fonctionnement de cette méthode d'optimisation (i.e. pour obtenir la convergence vers le mode global) que les étapes précédentes ne soient pas confrontées à des modes (minima) locaux, ou bien puissent déjà les discriminer facilement et se positionner proche du mode global. Le bon comportement pratique de l'algorithme de Ishikawa itératif sur l'ensemble des images considérées pour le critère $f_{\text{data}}(\mathbf{a}) + \beta\,\Phi^{\ell_1}_{\text{diff}}(\mathbf{a})$, nous amène donc à estimer que ce critère n'est pas très multi-modal au moins jusqu'à une petite échelle de variation des paramètres.

Pour effectuer la comparaison entre les résultats obtenus par les différentes techniques d'optimisation, j'ai donc retenu au final l'énergie calculée sur les trois images 3.1b, 3.3c, 3.4b par coupure sur graphe itérative ; en prenant un nombre confortable de $N_q \simeq 100$ labels. L'hyper-paramètre de régularisation β est ajusté sur chaque image comme cela est détaillé dans la sous-section 3.2.1. Il est bien évidemment fixé pour la comparaison avec les autres techniques d'optimisation. Les images obtenues sont représentées sur les figures (4.4a,4.4d,4.4g) et ont pour énergies respectives f_1^{gcut}=4.7107 e+06, f_2^{gcut}=1.8447 e+06 et f_3^{gcut}=3.9643 e+06. On peut se demander si tous les chiffres donnés sont significatifs, mais c'est effectivement le cas puisqu'une modification sensible de quelques pixels de l'image donne la même énergie à la précision considérée (plus il y a de pixels dans l'image plus il faut prendre en considération de chiffres après la virgule dans l'énergie a posteriori pour caractériser le minimum global sans ambiguïté ou distinguer des solutions pratiques proches de celui-ci).

4.1.1.2 Optimisation différentiable

Comme on l'a vu à la section 2.1, les méthodes d'optimisation différentiable ne garantissent pas de trouver le minimum global de la fonction non-convexe $f_{\text{data}}(\mathbf{a}) + \beta\, \Phi_{\text{diff}}^{\ell_1}(\mathbf{a})$ et visent uniquement à trouver un minimum local. Néanmoins puisqu'on a remarqué que la fonction ne semblait pas trop multi-modale, il reste probable que ces méthodes permettent de trouver un résultat assez proche de l'optimum global, comme cela a pu être calculé par optimisation combinatoire. Il convient toutefois de prendre certaines précautions pour parvenir à ce résultat étant donnée la non-convexité et la forme spécifique de $f_{\text{post}}(\mathbf{a})$.

Lorsque j'effectue une descente itérative de gradient (projeté) à pas constant, comme étudié théoriquement par [67], et que je sélectionne une longueur de pas suffisamment petite pour que $f_{\text{post}}(\mathbf{a})$ diminue à chaque itération, mais également pas trop petite de façon à ne pas faire du sur-place dans l'espace de grande dimension ($N_a \sim 10^6$) considéré, je constate que l'optimiseur atteint une solution assez éloignée de la solution calculée par coupure sur graphe (avec des énergies significativement plus élevées), et cela même après un très grand nombre d'itérations ($N^{\text{iter}} \sim 10^6$) de surcroît. Bien qu'un nombre encore plus grand d'itérations semble pouvoir encore très lentement diminuer la fonction $f_{\text{post}}(\mathbf{a})$, il n'est pas envisageable d'un point de vue pratique de devoir effectuer plusieurs semaines de calcul pour débruiter une petite image 500×500 pixels (avec des modèles creux ou séparables qui plus est !). Ce comportement est d'ailleurs observé quelle que soit l'initialisation considérée, à savoir une initialisation par les données, par la moyenne des données (image constante), ou bien par un filtrage gaussien des données.

Par ailleurs, lorsque j'effectue une recherche en ligne de la longueur de pas $\alpha > 0$ dans la direction de l'anti-gradient $-\mathbf{g}_{\text{post}}(\mathbf{a})$, de façon à obtenir une décroissance plus rapide de $f_{\text{post}}(\mathbf{a}^{(k-1)} - \alpha^{(k)} \times \mathbf{g}_{\text{post}}^{(k)})$ à chaque itération k, en me basant sur ce qui est préconisé pour l'optimisation convexe (cf. 2.1.1.1), je constate que l'optimiseur converge vers une solution très éloignée du minimum global calculé par coupure sur graphe, bien que cette solution soit légèrement plus proche du minimiseur global après $N^{\text{iter}} \sim 10^4$ itérations que celle obtenue à pas constant au bout de $N^{\text{iter}} \sim 10^6$ itérations. Et cela reste vrai quelle que soit la stratégie de recherche en ligne considérée : même pour une stratégie de type force brute consistant à échantillonner et tester toute une série de longueurs pas à chaque itération. Si on observe de

plus près les variations monodimensionnelles de $f_{\text{post}}(\mathbf{a}^{(k-1)} - \alpha^{(k)} \times \mathbf{g}^{(k)}_{\text{post}})$ dans la direction de l'anti-gradient $-\mathbf{g}^{(k)}_{\text{post}}(\mathbf{a})$ (par exemple sur la figure 4.3), on comprend mieux les difficultés rencontrées lorsqu'on utilise une stratégie de recherche en ligne. En effet, vu que le critère est constant par paliers voire multi-modal dans la direction de l'anti-gradient, les techniques usuelles de longueur de pas censées exploiter la convexité dans cette direction sont vouées à l'échec.

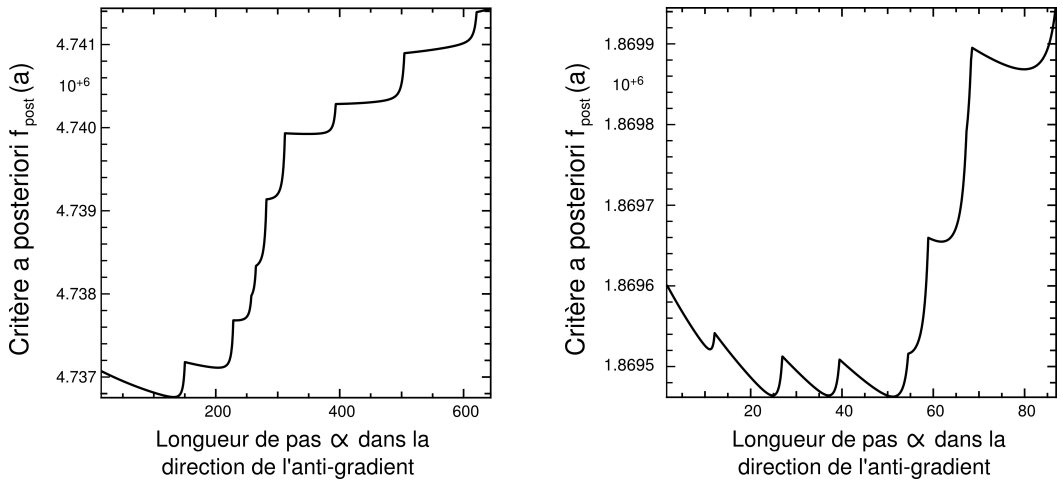


FIG. 4.3: Exemples de l'allure de $f_{\text{post}}\left[\mathbf{a}^{(k-1)} - \alpha^{(k)} \times \mathbf{g}^{(k)}\right]$ selon la direction d'anti-gradient et en fonction de α^k.

Puisque la vraisemblance $f_{\text{post}}(\mathbf{a})$ semble d'une part trop constante par paliers voire multi-modale dans certaines directions comme celle du gradient, et puisqu'il semble d'autre part difficile de construire une direction de descente plus pertinente qu'une direction de type anti-gradient pour cette vraisemblance a posteriori, il semble judicieux d'autoriser l'optimiseur à pouvoir effectuer des augmentations de $f_{\text{post}}(\mathbf{a})$ lors de certaines itérations effectuées dans la direction de l'anti-gradient, de façon à permettre un déplacement plus large dans l'espace des paramètres. D'un autre côté, il faut quand même contrôler l'amplitude et le nombre de ces remontées pour garantir une diminution globale du critère $f_{\text{post}}(\mathbf{a})$ considéré et la convergence vers un minimum local. De plus il semble intéressant d'exploiter le fait que la meilleure longueur de pas qui est trouvée à l'aide d'une stratégie de recherche en ligne de type force brute dans la direction du gradient, à tendance à décroître au fur et à mesure des itérations.

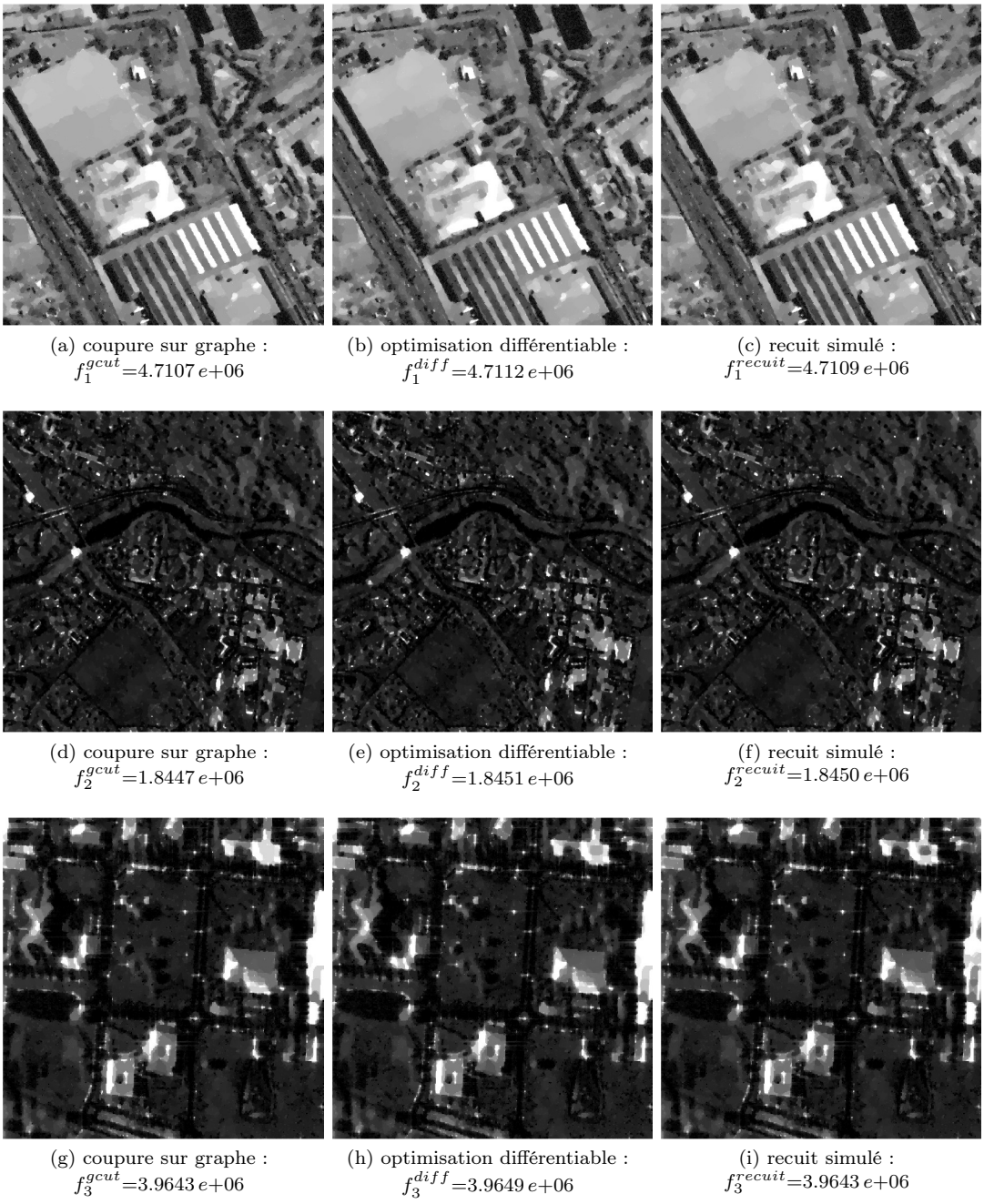
En me basant sur ces observations, j'ai donc mis en place une stratégie heuristique d'ajustement des longueurs de pas $\alpha^{(k)}$ qui consiste à considérer une longueur de pas initiale $\alpha^{(0)}$ très élevée et à la diminuer progressivement en fonction de la descente globalement observée sur un petit nombre d'itérations n_{obs} typiquement égal à 5. J'enregistre pour cela l'itération k_{best} correspondant à la meilleure minimisation $f_{\text{post}}(\mathbf{a}^{(k_{\text{best}})})$ jusqu'alors effectuée, et je diminue légèrement la longueur de pas courante $\alpha^{(k_{\text{best}})}$ tout en réinitialisant l'algorithme avec la meilleure solution trouvée $\mathbf{a}^{(k_{\text{best}})}$, lorsque la longueur de pas courante n'a pas per-

mis de trouver une solution $\mathbf{a}^{k'}$ telle que $k_{\text{best}}<k'\leq k_{\text{best}}+n_{\text{obs}}$ et $f_{\text{post}}(\mathbf{a}^{k'})<f_{\text{post}}(\mathbf{a}^{k_{\text{best}}})$ (i.e. lorsque la longueur de pas courante n'a pas permis de diminuer $f_{\text{post}}(\mathbf{a}^{k_{\text{best}}})$ après au plus n_{obs} itérations). Si au contraire tout se passe comme cela est souhaité, la longueur de pas courante $\alpha^{(k_{\text{best}})}$ est conservée. En pratique cette stratégie heuristique permettant localement de légères remontées dans la direction de l'anti-gradient est très efficace sur les critères considérés. A l'instar du recuit simulé j'autorise de temps en temps la longueur de pas à recroître. Les énergies obtenues au bout de $\sim 10^4$ itérations à l'aide de cette stratégie valent respectivement f_1^{diffg}=4.7113 e+06, f_2^{diffg}=1.8537 e+06 et f_3^{diffg}=3.9689 e+06 sur les 3 images considérées. Les images obtenues sont globalement assez semblables à celles obtenues par coupure sur graphe bien que des différences puissent être détectées visuellement sur certains pixels.

Une autre stratégie pour minimiser $f_{\text{post}}(\mathbf{a})=f_{\text{data}}(\mathbf{a})+\beta\,\Phi_{\text{diff}}^{\ell_1}(\mathbf{a})$ par optimisation différentiable consiste à approximer la fonction de régularisation $\Phi_{\text{diff}}^{\ell_1}$ par la fonction de régularisation plus lisse $\Phi_{\text{diff}}^{\ell_2|\ell_1}$, en considérant le même poids de régularisation β et en diminuant progressivement le seuil $\epsilon^{\ell_2|\ell_1}$ de façon à se rapprocher progressivement de $\Phi_{\text{diff}}^{\ell_1}$ (approche homotopique). En effet étant donné que $\Phi_{\text{diff}}^{\ell_2|\ell_1}$ possède des dérivées secondes qui ne sont pas toujours nulles, cela permet de considérer des directions de descente du second-ordre plus pertinentes que la direction de gradient pour minimiser plus rapidement les fonctions a posteriori successives associées. D'autre part, comme les fonctions de régularisation considérées sont plus lisses tout en s'approchant de plus en plus de la fonction de régularisation approximée, cela permet d'éviter certains minima locaux qui pourraient être rencontrés lors de la descente avec la fonction de régularisation $\Phi_{\text{diff}}^{\ell_1}$ uniquement. Je me suis en particulier servi d'un algorithme non-convexe du second-ordre de type sous-espace avec région de confiance (cf section 2.1.2) en prenant $\mathbf{s}_1=-\mathbf{B}_{\text{post}}^{-1}.\mathbf{g}_{\text{data}}$ et $\mathbf{s}_2=-\mathbf{B}_{\text{post}}^{-1}.\mathbf{g}_{\text{prior}}$, où $\mathbf{B}_{\text{post}}^{-1}$ est un préconditionneur diagonal. J'observe cependant que les critères considérés sont trop peu convexes et corrélés pour que le calcul de la direction de Newton-tronqué $\mathbf{s}_3=\mathbf{s}_{\text{NT}}$ soit bénéfique. Les énergies obtenues au bout de $\sim 10^4$ itérations, lesquelles sont réparties selon une vingtaine de valeur de seuil (reliées entre elles par un facteur de décroissance typique de 0.7) avant de considérer finalement la fonction de régularisation $\Phi_{\text{diff}}^{\ell_1}(\mathbf{a})$ et de l'optimiser à l'aide de l'algorithme de premier ordre précédemment décrit, valent respectivement f_1^{diff}=4.7112 e+06, f_2^{diff}=1.8451 e+06 et f_3^{diff}=3.9649 e+06 pour les 3 images considérées. Ces énergies et les solutions qui leur correspondent (cf. figures 4.4b,4.4e,4.4h) sont significativement plus proches de l'optimum calculé par coupure sur graphe, que les solutions obtenues à l'aide de la stratégie différentiable de premier ordre précédemment envisagée tout en nécessitant un temps de calcul comparable. Les différences par rapport aux solutions obtenues par coupure sur graphe sont globalement minimes bien que perceptibles sur certains pixels (petites zones constantes par morceaux et valant une constante visiblement différente, ce qui est lié en partie à l'interaction entre la fonction de pénalisation et un critère d'attache aux données qui est plat par endroits).

4.1.1.3 Optimisation par échantillonnage MCMC

Pour calculer $\mathbf{a}^{\text{MAP}}$ par optimisation stochastique, j'ai utilisé un algorithme de type recuit simulé que j'ai initialisé avec les mêmes images initiales que l'algorithme d'optimisation différentiable.



(a) coupure sur graphe : f_1^{gcut}=4.7107 e+06
(b) optimisation différentiable : f_1^{diff}=4.7112 e+06
(c) recuit simulé : f_1^{recuit}=4.7109 e+06
(d) coupure sur graphe : f_2^{gcut}=1.8447 e+06
(e) optimisation différentiable : f_2^{diff}=1.8451 e+06
(f) recuit simulé : f_2^{recuit}=1.8450 e+06
(g) coupure sur graphe : f_3^{gcut}=3.9643 e+06
(h) optimisation différentiable : f_3^{diff}=3.9649 e+06
(i) recuit simulé : f_3^{recuit}=3.9643 e+06

FIG. 4.4: Comparaison entre les images obtenues par les différentes techniques d'optimisation envisagées pour le modèle $f_{\text{data}}(\mathbf{a}) + \beta\, \Phi_{\text{diff}}^{\ell_1}(\mathbf{a})$.

Les échantillons à une température donnée sont obtenus à l'aide d' un algorithme MCMC de Metropolis-Hastings de type *marche aléatoire* avec *balayage aléatoire et séquentiel des pixels* [60], proposé notamment par Louchet [58] pour le débruitage gaussien des images avec un a priori de type variation totale (dans le cadre de l'estimation moyenne a posteriori). On ne modifie donc qu'un seul paramètre a_i à la fois, lequel est choisi aléatoirement parmi l'ensemble des paramètres $\mathbf{a}$. D'autre part la densité de proposition conditionnelle est une loi uniforme centrée sur la valeur courante du paramètre (pixel) choisi :

$$\mathcal{Q}(a_i|a_i^{(k-1)}) = \mathcal{U}_{\left[a_i^{(k-1)}-\gamma\,,\; a_i^{(k-1)}+\gamma\right]}\,, \tag{4.1}$$

où le paramètre γ est un paramètre d'échelle. γ est ajusté pour ne pas aller taper trop souvent dans des régions de faible probabilité tout en étant suffisamment grand pour permettre l'échantillonnage rapide de l'ensemble des zones d'intérêt (en principe raréfié en grande dimension). En pratique on choisit γ de sorte que le taux d'acceptation soit de l'ordre de 0.234 [61, 62].

Comme on l'a mentionné précédemment le recuit simulé est une heuristique ne garantissant pas la convergence vers un minimiseur global ou même local lorsque la température n'est pas décrue suffisamment lentement (ce qui nécessiterait un temps de calcul infini). Mais il facilite néanmoins la diminution de l'énergie dans le cas où le critère à optimiser est multimodal. Pour accélérer l'optimisation du critère a posteriori, j'ai donc combiné l'approche stochastique de recuit simulé avec l'algorithme de descente différentiable de premier ordre précédemment présenté. J'effectue ainsi de l'ordre de 10^3 itérations de descente avec l'algorithme différentiable avant chaque nouvelle phase de recuit (5 phases de recuit typiquement permettant d'atteindre une température quasi-nulle à l'issue de ces 5 phases). Cela correspond à effectuer de l'ordre de 10^5 évaluations de fonctions en tout et pour tout et permet d'obtenir les énergies f_1^{recuit}=4.7109 e+06, f_2^{recuit}=1.8450 e+06 et f_3^{recuit}=3.9643 e+06. Ces énergies sont quasiment identiques à celles obtenues par coupure sur graphe, et ne nécessitent qu'une quantité de mémoire RAM beaucoup plus réduite (d'un facteur $10 \times N_q$ typiquement) ainsi qu'un temps de calcul comparable à celui des coupes sur graphe itératives. Les images corrrespondantes sont représentées sur les figures (4.4c,4.4f,4.4i).

Notons que l'utilisation du recuit simulé est parfois couplée à des initialisations multiples afin d'atteindre des modes différents et de sélectionner la solution de plus basse énergie. Mais en pratique, je n'ai pas eu recours à ces initialisations multiples étant donné les énergies obtenues après une seule initialisation.

4.1.2 Minimisation de $f_{\text{data}}(\mathbf{a}) + \beta\,\Phi^{\ell_1}_{\text{grad}}(\mathbf{a})$

La régularisation $\Phi^{\ell_1}_{\text{grad}}(\mathbf{a})$ correspond à une discrétisation plus précise de la variation totale isotrope de l'image, par rapport à la régularisation $\Phi^{\ell_1}_{\text{diff}}(\mathbf{a})$ précédemment envisagée. D'autre part le fait que $\Phi^{\ell_1}_{\text{grad}}(\mathbf{a})$ contienne des corrélations entre pixels ainsi que le fait que ses dérivées secondes ne soient pas nulles, permettent de mettre en œuvre des algorithmes d'optimisation différentiable du second ordre plus performants pour minimiser l'énergie a posteriori correspondante $f_{\text{post}}(\mathbf{a})$=$f_{\text{data}}(\mathbf{a})$+$\beta\,\Phi^{\ell_1}_{\text{grad}}(\mathbf{a})$. Cela est également censé rendre l'énergie localement plus convexe voire globalement moins multi-modale qu'en utilisant $\Phi^{\ell_1}_{\text{diff}}(\mathbf{a})$.

Néanmoins l'énergie $f_{\text{post}}(\mathbf{a})$ ne peut alors plus être minimisée de façon exacte par les approches de coupure sur graphe précédemment envisagées. De plus, étant donné que $f_{\text{post}}(\mathbf{a})$ reste potentiellement non-convexe voire multi-modale, on ne peut pas garantir qu'elle soit minimisée de façon exacte ou quasi-exacte par des approches différentiables. Au vu des bons résultats précédemment obtenus sur l'optimisation d'une énergie plus difficilement optimisable (avec la régularisation $\Phi^{\ell_1}_{\text{diff}}(\mathbf{a})$), on s'attend toutefois à pouvoir bien optimiser la nouvelle énergie et à ce que cette optimisation soit plus rapide que lorsqu'on a considéré la fonction de régularisation moins convexe et moins corrélée $\Phi^{\ell_1}_{\text{diff}}(\mathbf{a})$.

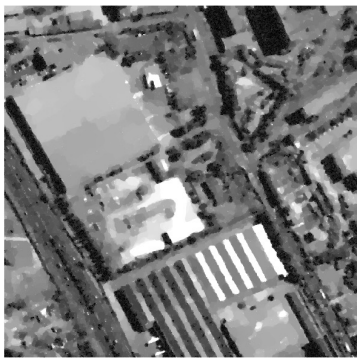
Je compare donc ici les résultats pratiques de l'optimisation du modèle $f_{\text{data}}(\mathbf{a}) + \beta\,\Phi^{\ell_1}_{\text{grad}}(\mathbf{a})$ que l'on obtient d'une part par optimisation différentiable, et d'autre part par optimisation stochastique. Bien qu'on ait vu lors des comparaisons effectuées avec l'énergie $f_{\text{data}}(\mathbf{a}) + \beta\,\Phi^{\ell_1}_{\text{diff}}(\mathbf{a})$, qu'on ne dispose pas d'un temps suffisant pour laisser l'approche stochastique converger vers le minimum global comme cela est prévu par la théorie, on s'attend toutefois au vu des résultats précédents à atteindre un résultat très proches de l'optimiseur global lorsqu'on combine l'approche stochastique avec une approche différentiable dans un temps plus raisonnable.

4.1.2.1 Optimisation différentiable

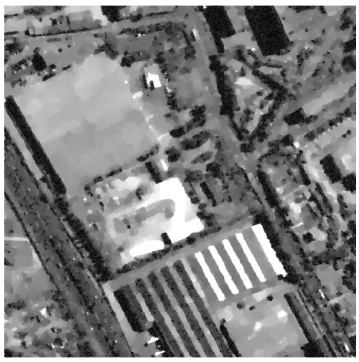
J'ai été confronté aux mêmes difficultés concernant la descente de gradient que celles rencontrées pour optimiser $f_{\text{data}}(\mathbf{a})+\beta\,\Phi^{\ell_1}_{\text{diff}}(\mathbf{a})$. C'est à dire que malgré le fait que la fonction de régularisation $\Phi^{\ell_1}_{\text{grad}}$ soit plus convexe que $\Phi^{\ell_1}_{\text{diff}}$, l'énergie globale reste trop constante par paliers voire multi-modale selon la direction de descente de l'anti-gradient pour pouvoir être simplement minimisée par une descente "classique" de gradient. J'ai donc tout d'abord optimisé $f_{\text{data}}(\mathbf{a})+\beta\,\Phi^{\ell_1}_{\text{grad}}(\mathbf{a})$ par une descente de gradient en me servant de la stratégie heuristique d'ajustement de longueur de pas présentée à la section 4.1.1.2. J'obtiens respectivement les énergies $f^{\text{diffg}}_{1'}$=4.7072 e+06, $f^{\text{diffg}}_{2'}$=3.9556 e+06 et $f^{\text{diffg}}_{3'}$=1.8472 e+06.

J'ai par la suite optimisé $f_{\text{data}}(\mathbf{a})+\beta\,\Phi^{\ell_1}_{\text{grad}}(\mathbf{a})$ en utilisant la fonction de régularisation $\Phi^{\ell_2|\ell_1}_{\text{grad}}$ et en diminuant progressivement le seuil de régularisation $\epsilon^{\ell_2|\ell_1}$ (avant de considérer $\Phi^{\ell_1}_{\text{grad}}$ pour les dernières itérations). Cette approche permet de simplifier la minimisation de l'énergie $f_{\text{data}}(\mathbf{a}) + \beta\,\Phi^{\ell_1}_{\text{grad}}(\mathbf{a})$, car celle-ci est fortement non-convexe voire multi-modale, tandis que les approximations successives considérées sont en principe moins multi-modales et plus lisses localement. Une façon alternative de s'approcher du minimum global au moyen d'approximations homotopiques plus lisses, consiste à utiliser un poids de régularisation initial β^0 plus élevé que celui souhaité, et à diminuer progressivement β jusqu'à sa valeur finale. Cela peut être réalisé facilement et très rapidement à l'aide de l'algorithme présenté dans [68, 42] par exemple, mais j'obtiens toutefois de meilleurs résultats avec la stratégie de décroissance de $\epsilon^{\ell_2|\ell_1}$ sur les énergies considérées.

Pour optimiser partiellement les énergies basées sur les approximations successives de la régularisation $\Phi^{\ell_1}_{\text{grad}}$ (ainsi que celle utilisant in fine la régularisation cible), j'ai utilisé l'algorithme non-convexe de type sous-espace avec région de confiance présenté à la section 2.1.2. J'ai ajouté la direction de Newton tronqué $\mathbf{s}_3$=$\mathbf{s}_{\text{NT}}$ dans le sous-espace par rapport à la section 4.1.1.2, puisque l'énergie considérée s'y prête davantage (dérivées secondes plus positives



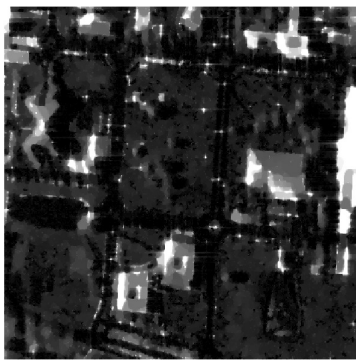
(a) optimisation différentiable : $f_{1'}^{\text{diff}}$=4.7067 e+06



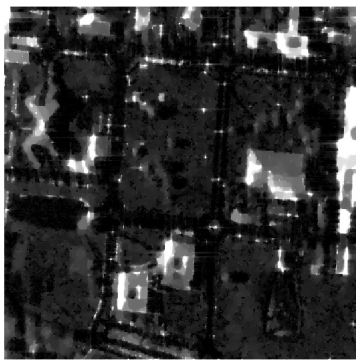
(b) recuit simulé : $f_{1'}^{\text{rec}}$=4.7067 e+06

(c) optimisation différentiable : $f_{2'}^{\text{diff}}$=1.8400 e+06

(d) recuit simulé : $f_{2'}^{\text{rec}}$=1.8400 e+06



(e) optimisation différentiable : $f_{3'}^{\text{diff}}$=3.9533 e+06



(f) recuit simulé : $f_{3'}^{\text{rec}}$=3.9535 e+06

FIG. 4.5: Comparaison entre les images obtenues par les différentes techniques d'optimisation envisagées pour le modèle $f_{\text{data}}(\mathbf{a}) + \beta\, \Phi_{\text{grad}}^{\ell_1}(\mathbf{a})$.

et moins localisées) et de façon à accélérer le taux de convergence de l'optimiseur. J'obtiens respectivement les énergies $f_{1'}^{\text{diff}}$=4.7067 e+06, $f_{2'}^{\text{diff}}$=3.9533 e+06 et $f_{3'}^{\text{diff}}$=1.8400 e+06. Les images correspondantes sont reportées sur les figures 4.5a, 4.5c et 4.5e. Les énergies sont significativement meilleures que celles obtenues par l'approche de type descente de gradient, tout en nécessitant un temps de calcul et une quantité de mémoire RAM équivalents.

Par ailleurs, on peut noter la rapidité de convergence de l'optimiseur non-convexe envisagé (de l'ordre de 10^2 itérations voire moins en fonction du degré de convergence souhaité), pour le cas où l'on cherche à minimiser un modèle avec une fonction de régularisation de type $\Phi_{\text{diff}}^{\ell_2|\ell_1}$ plutôt qu'avec une fonction de type variation totale (afin de ne pas trop favoriser les portions d'images constantes par morceaux et les effets de marches d'escalier notamment).

On peut également remarquer que l'emploi d'une région de confiance assure une décroissance monotone des énergies considérées, et qu'il pourrait être plus judicieux de relaxer cette propriété pour mieux s'adapter à la forme des énergies considérées, à l'instar de ce qui a été fait pour la recherche en ligne de longueur de pas.

4.1.2.2 Optimisation par échantillonnage MCMC

Pour limiter les temps de calcul, j'ai combiné l'approche stochastique de recuit simulé avec des descentes différentiables locales de type sous-espace, de la même manière que cela a été décrit à la section 4.1.1.3. J'obtiens les énergies $f_{1'}^{\text{rec}}$=4.7067 e + 06, $f_{2'}^{\text{rec}}$=3.9535 e + 06 et $f_{3'}^{\text{rec}}$=1.8400 e + 06, qui sont quasiment identiques à celles obtenues par optimisation différentiable pour le même modèle. Cela semble témoigner d'une meilleure qualité de minimisation de l'approche différentiable que lorsqu'on a considèré le modèle $\Phi_{\text{grad}}^{\ell_1}(\mathbf{a})$. Cette supputation est par ailleurs corroborée par le fait que la norme du gradient de la solution obtenue par l'approche différentiable est de l'ordre de 10 fois plus faible lorsqu'on considère le modèle $\Phi_{\text{grad}}^{\ell_1}(\mathbf{a})$ à la place du modèle $\Phi_{\text{diff}}^{\ell_1}(\mathbf{a})$.

Par ailleurs on remarque que les solutions obtenues avec le modèle $\Phi_{\text{grad}}^{\ell_1}(\mathbf{a})$ sont globalement assez proches de celle obtenues avec le modèle $\Phi_{\text{diff}}^{\ell_1}(\mathbf{a})$.

4.1.3 Conclusion concernant les algorithmes d'optimisation MAP

Les différents algorithmes d'optimisation MAP envisagés offrent des performances pratiques identiques en terme de minimisation d'énergie et similaires en terme de temps de calcul. On peut relever de légères différences entre les solutions trouvées par les différents optimiseurs, ce qui montre qu'aucun des algorithmes ne converge exactement vers le minimiseur global, et que des solutions sensiblement différentes peuvent être obtenues pour des énergies identiques et très proches de celle du minimum global. On peut donc parler de dégénérescence ou de quasi-dégénérescence des estimateurs envisagés.

Néanmoins ces différences restent négligeables devant les imperfections des modèles considérés ou devant l'information que l'on pourra sans doute extraire des images débruitées avec les modèles considérés, et la proximité par rapport au minimiseur global semble très bonne.

Seule l'approche de coupure sur graphe possède des propriétés de convergence théorique démontrées vers le minimiseur global. Celle-ci peut donc légitimement être considérée comme la plus fiable. Néanmoins on a vu qu'il est nécessaire de disposer d'une quantité de mémoire très importante pour la mettre en œuvre, ce qui peut être rédhibitoire lorsqu'on doit considérer des images de grande dimension. De plus les modèles a priori pour lesquels la convergence exacte est démontrée sont également relativement restreints. Il est donc également intéressant de constater que les approches différentiables et stochastiques peuvent fournir des solutions comparables dans le contexte de débruitage qui nous intéresse, moyennant quelques adaptations de ces algorithmes au cas non-convexe et multi-modal.

Deux types d'interactions a priori entre pixels voisins (différences isotropes entre paires $\Phi^{\ell_1}_{\text{diff}}$ et gradient isotropes $\Phi^{\ell_1}_{\text{grad}}$) ont été envisagées pour les comparaisons. Les différences observées entre les estimateurs associés sont du même ordre que celles observées lorsqu'on applique différents optimiseurs pour le même modèle. Le choix du type d'interactions peut donc plutôt être motivé par le choix de l'algorithme d'optimisation utilisé et par les performances qui en découlent. Ainsi on gagnera à considérer plutôt $\Phi^{\ell_1}_{\text{diff}}$ pour l'optimisation combinatoire par coupure sur graphe, et $\Phi^{\ell_1}_{\text{grad}}$ pour l'optimisation différentiable du deuxième ordre.

4.2 Calcul de la moyenne a posteriori

Etant donné la grande dimension des problèmes d'intégration ($N_a \sim 3\times10^5$) ainsi que le caractère multi-modal et constant par paliers des distributions a posteriori envisagées, il semble trop ambitieux voire peu réaliste de vouloir calculer très précisément la moyenne a posteriori, et de revendiquer une précision théorique ou pratique d'un algorithme MCMC en temps limité et en précision finie. On ne cherchera donc pas ici à comparer les performances de différents algorithmes d'échantillonnage stochastique, mais on présentera plus simplement un algorithme qui semble échantillonner efficacement l'espace des paramètres sur les problèmes envisagés (grâce au recours à des stratégies heuristiques de recuit simulé et de départs multiples).

Malgré la lenteur de convergence de l'échantillonneur MCMC de type marche aléatoire proposé par Louchet (cf. section 4.1.1.3), celui-ci est particulièrement simple à mettre en œuvre et dispose d'une flexibilité que l'on peut automatiser à l'aide d'un nombre restreint de paramètres (largeur de la loi de proposition, facteur de sous-échantillonnage, et temps de chauffe). J'ai donc retenu les grandes lignes de cet échantillonneur (qui est présenté plus en détail dans sa thèse de doctorat [59]) et y ai ajouté des variations de température et des départs multiples afin de l'adapter aux formes problématiques des distributions a posteriori envisagées. Dans le cadre envisagé par Louchet deux chaînes de Markov sont lancées en parallèle avec des initialisations différentes, afin d'évaluer la distance de la moyenne des échantillons par rapport à l'estimateur moyen. Dans le contexte multi-modal qui nous intéresse on utilise un plus grand nombre d'initialisations différentes (de l'ordre de 6 pour des raisons pratiques, bien qu'un nombre plus élevé semble fournir de meilleurs résultats), de façon à éviter que l'échantillonneur concentre la plus grande partie du temps qui lui est imparti sur une portion restreinte de l'espace des paramètres. Au final la moyenne est opérée sur de l'ordre de $\sim 3\times\sqrt{N_a}$ échantillons, chacun séparé par un facteur de sous-échantillonnage de $5\times N_a$. De

plus une très courte phase de recuit est également opérée tous les $\sim \sqrt{N_a}/10$ échantillons pour favoriser le franchissement d'éventuelles barrières d'énergie.

Le choix des départs aléatoires est bien entendu décisif et il est responsable d'une partie du biais de l'estimateur obtenu. Pour que ce biais soit favorable à la qualité de l'estimation on exploite le fait que l'hyperparamètre de réglage du poids des a priori β devrait idéalement pouvoir être ajusté spatialement — aussi pertinente que puisse être la fonction de potentiel envisagée. De plus la valeur nominale de β n'est sans doute pas non plus exactement optimale. En effet, bien qu'on puisse déterminer une valeur de poids d'a priori globalement pertinente, il reste que certaines structures de l'image semblent apparaître plus clairement pour des valeurs différentes de poids de régularisation β. A défaut de disposer d'une stratégie efficace d'ajustement spatial du poids des a priori, il semble donc judicieux d'effectuer des moyennes a posteriori sur un nombre plus réduit d'échantillons pour des valeurs de β encadrant la valeur nominale, puis de se servir de ces résultats comme initialisation des chaînes de Markov effectuées avec la valeur nominale de β.

4.3 Comparaison entre les différents estimateurs

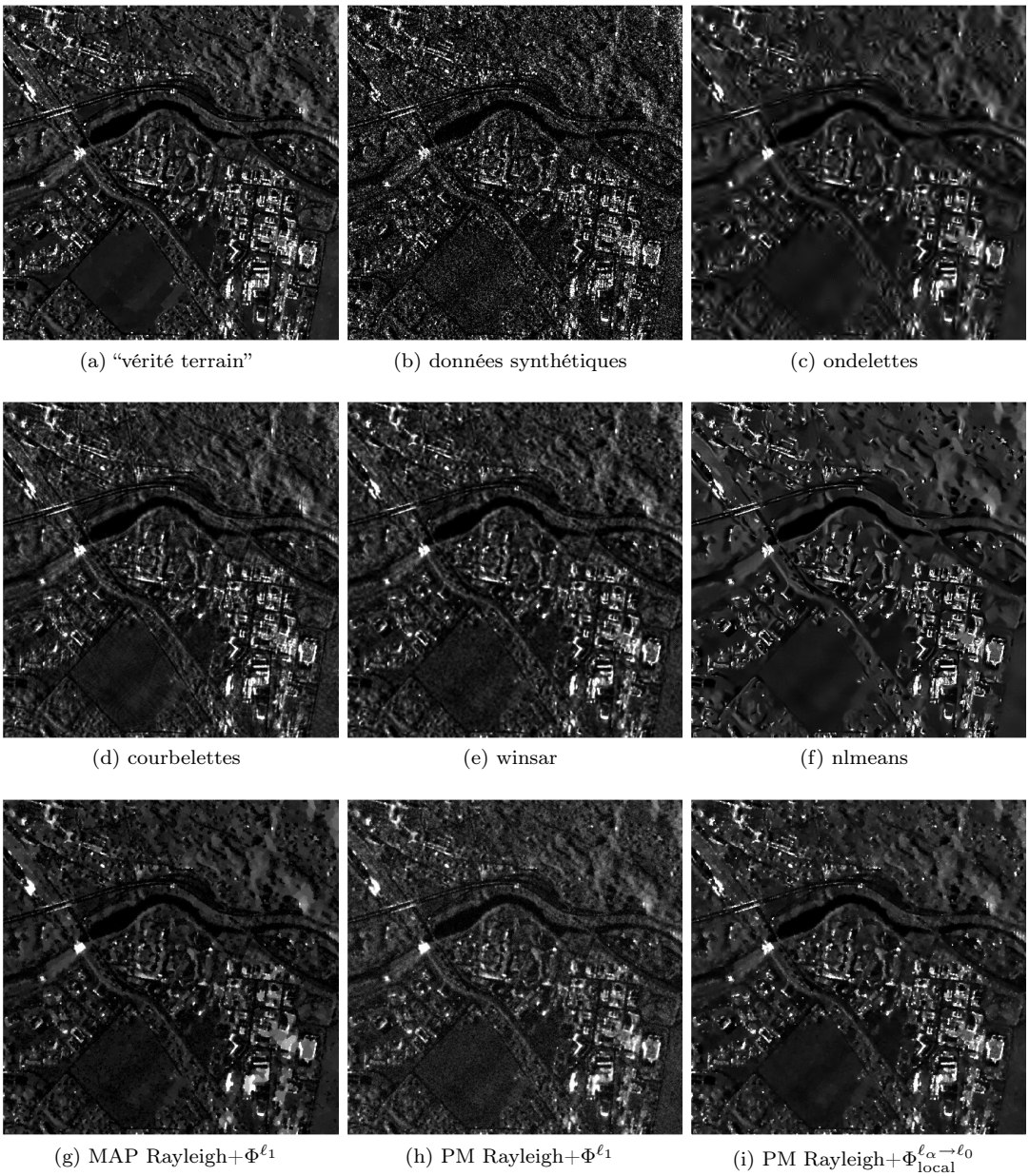
Etant donné le grand nombre d'algorithmes de débruitage envisagés et implémentés, je ne présente ici que les résultats les plus marquants. J'évalue tout d'abord les performances de différents estimateurs en utilisant les données synthétiques radar et les critères de comparaison présentés à la section 3.2.2. Puis je présente les résultats obtenus avec les vraies données radar RSO introduites à la section 3.1.

4.3.1 Données synthétiques

Les résultats obtenus par seuillage dur et invariance par translation dans des bases d'ondelettes et de courbelettes sont représentés sur les figures 4.6c, et 4.6d,4.8c respectivement. J'ai pour cela supposé que le logarithme des données est corrompu par un bruit additif gaussien de variance uniforme, puis j'ai effectué un débiaisage a posteriori sur l'image débruitée. Pour réduire les artefacts associés à la représentation de l'image par des courbelettes, j'ai appliqué un filtre médian sur les valeurs qui sont des extrema locaux et qui sont trop différentes par rapport à leur voisinage local ou par rapport aux données initiales. Ce traitement ne concerne donc que quelques points et permet d'améliorer légèrement les critères considérés tout en supprimant les points aberrants de l'image débruitée. Par ailleurs je présente également les résultats obtenus avec l'algorithme winsar (figure 4.6e), qui constitue une référence pour le débruitage multiplicatif radar par seuillage bayésien dans une base d'ondelettes. Une partie de ces résulats n'est donnée que pour l'image synthétique de Saint-Gervais pour des raisons d'implémentation des algorithmes en puissance de 2.

En ce qui concerne les approches de débruitage non-local, les résultats obtenus avec l'algorithme de Deledalle [1] dédié au bruit multiplicatif radar sont représentés sur les figures

[1] je remercie d'ailleurs au passage Charles Deledalle pour m'avoir fourni les résultats de son code de moyennes non-local ainsi que ceux de son implémentation de winsar

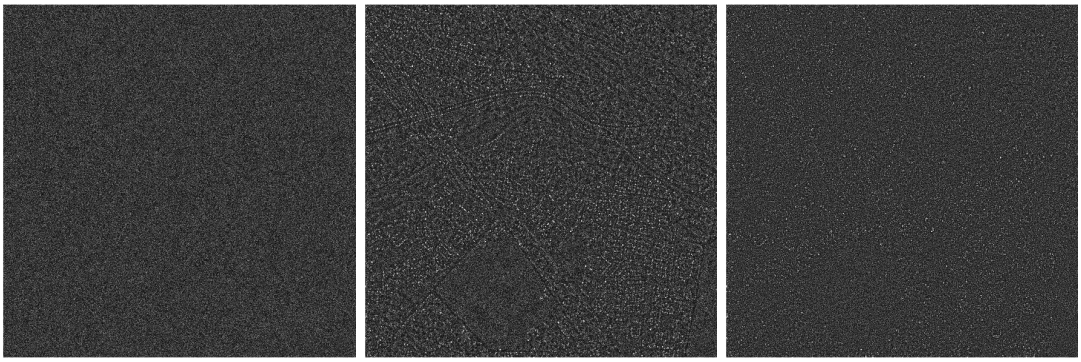


(a) “vérité terrain” (b) données synthétiques (c) ondelettes

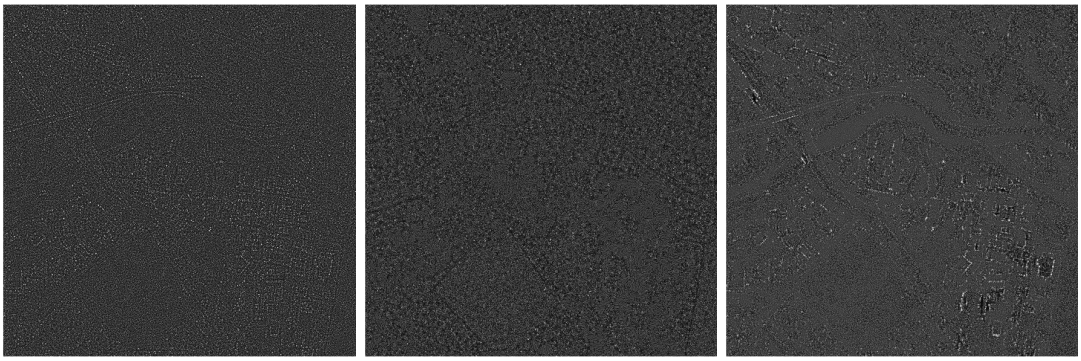
(d) courbelettes (e) winsar (f) nlmeans

(g) MAP Rayleigh+Φ^{ℓ_1} (h) PM Rayleigh+Φ^{ℓ_1} (i) PM Rayleigh+$\Phi_{\text{local}}^{\ell_\alpha \to \ell_0}$

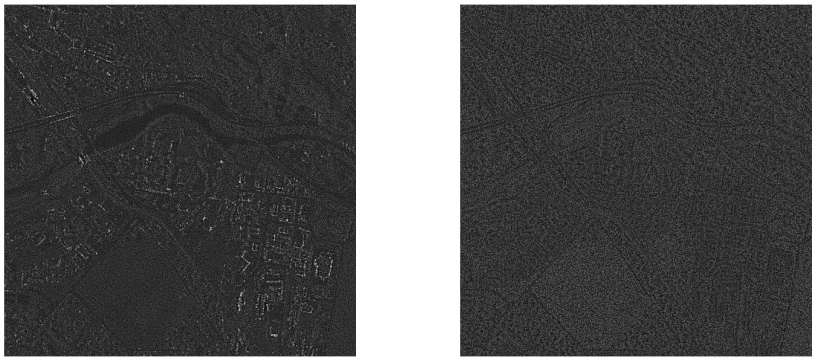
FIG. 4.6: Résultats des différents algorithmes de débruitage appliqués à l’image synthétique de Saint-Gervais.



(a) “vérité terrain” (b) ondelettes (c) courbelettes

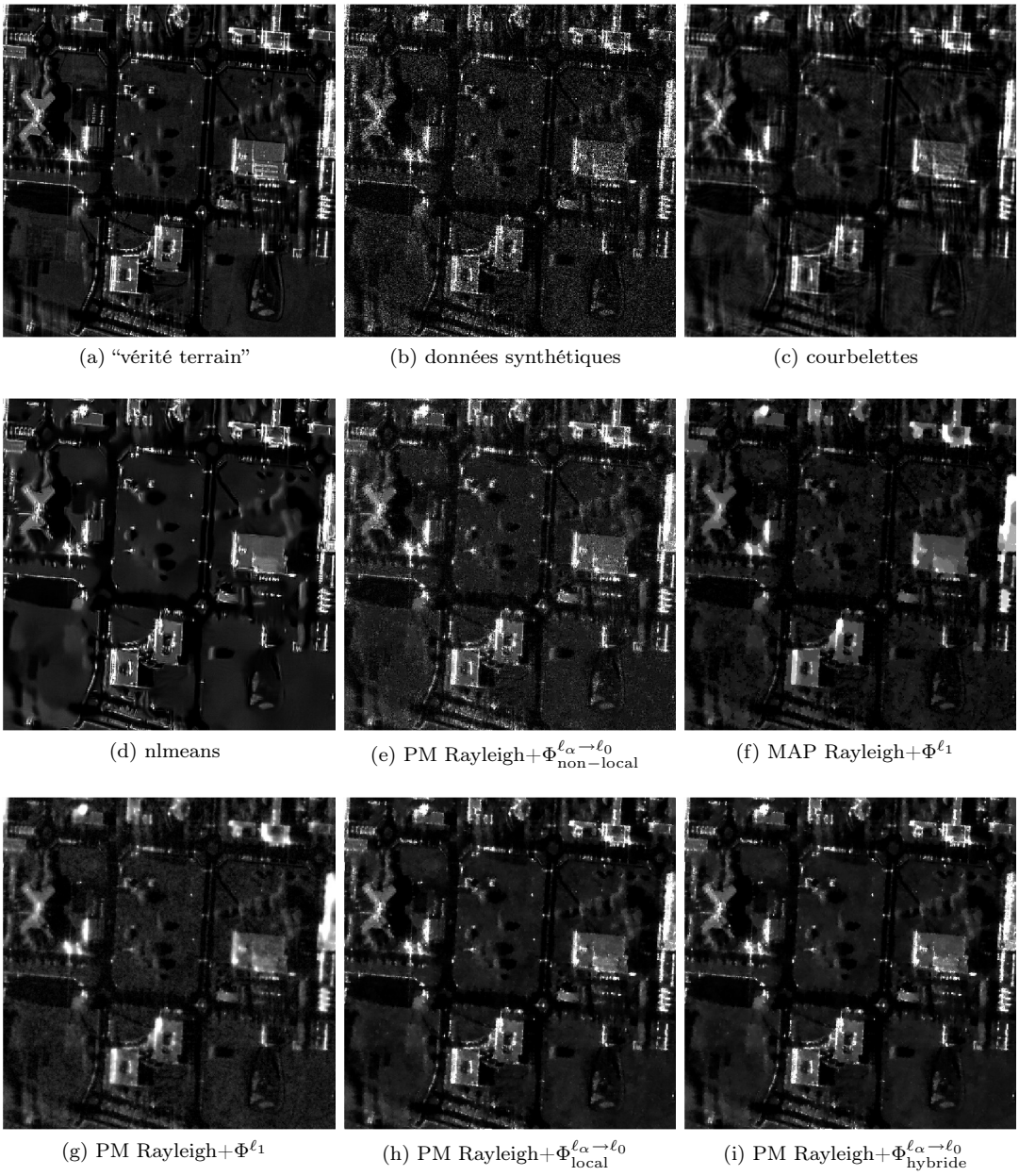


(d) winsar (e) nlmeans (f) MAP Rayleigh+Φ^{ℓ_1}



(g) PM Rayleigh+Φ^{ℓ_1} (h) PM Rayleigh+$\Phi_{\text{local}}^{\ell_\alpha \to \ell_0}$

FIG. 4.7: Rapports de bruit entre l’image synthétique de Saint-Gervais et les images débruitées par les différents algorithmes envisagés.

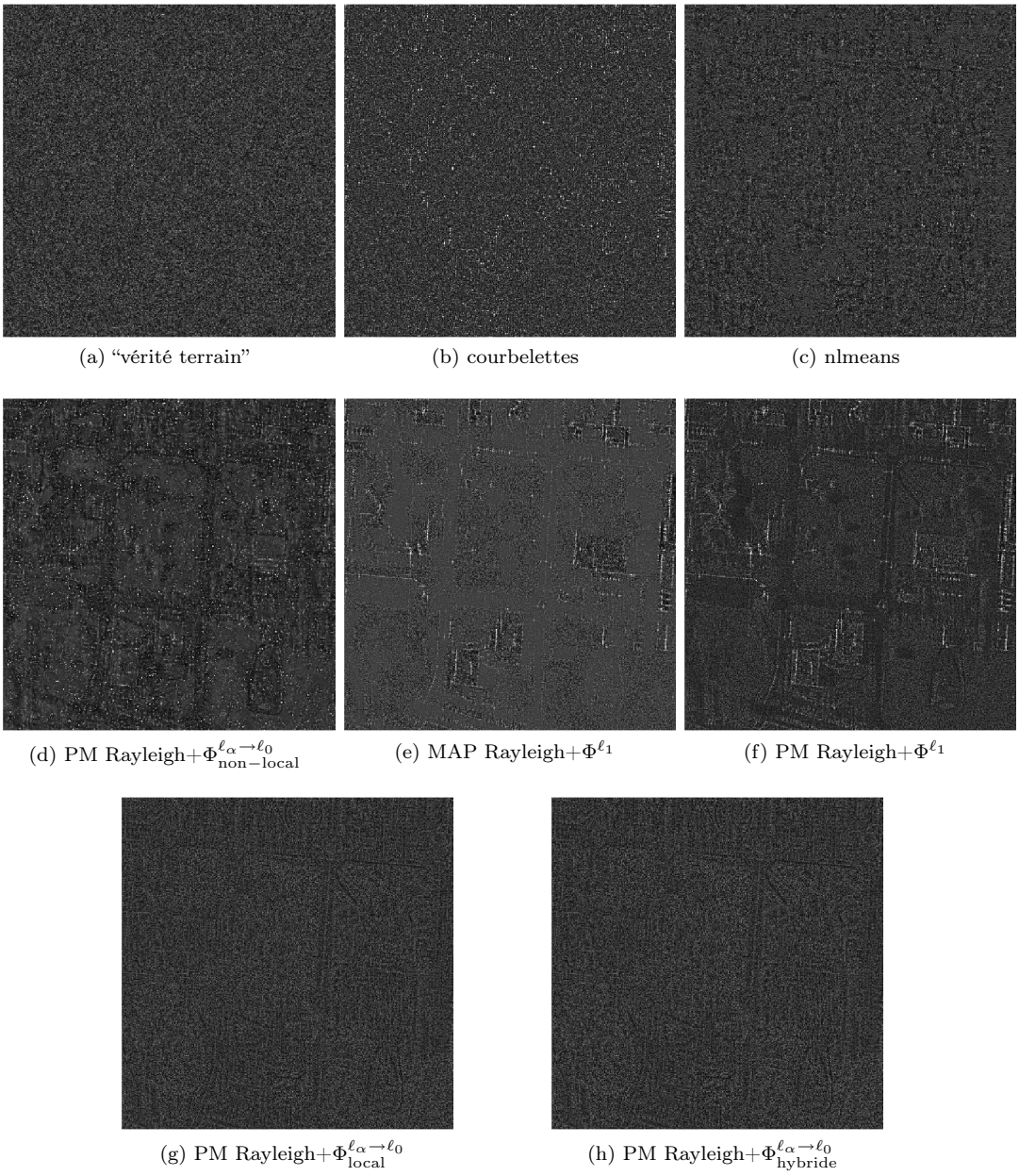


(a) “vérité terrain” (b) données synthétiques (c) courbelettes

(d) nlmeans (e) PM Rayleigh+$\Phi^{\ell_\alpha \to \ell_0}_{\text{non-local}}$ (f) MAP Rayleigh+Φ^{ℓ_1}

(g) PM Rayleigh+Φ^{ℓ_1} (h) PM Rayleigh+$\Phi^{\ell_\alpha \to \ell_0}_{\text{local}}$ (i) PM Rayleigh+$\Phi^{\ell_\alpha \to \ell_0}_{\text{hybride}}$

FIG. 4.8: Résultats des différents algorithmes de débruitage appliqués à l’image synthétique de Toulouse.



(a) "vérité terrain" (b) courbelettes (c) nlmeans

(d) PM Rayleigh+$\Phi^{\ell_\alpha \to \ell_0}_{\text{non-local}}$ (e) MAP Rayleigh+Φ^{ℓ_1} (f) PM Rayleigh+Φ^{ℓ_1}

(g) PM Rayleigh+$\Phi^{\ell_\alpha \to \ell_0}_{\text{local}}$ (h) PM Rayleigh+$\Phi^{\ell_\alpha \to \ell_0}_{\text{hybride}}$

FIG. 4.9: Rapports de bruit entre l'image synthétique de Toulouse et les images débruitées par les différents algorithmes envisagés.

4.6f,4.8d. Puis un résultat obtenu par moyenne a posteriori avec un a priori sur le gradient non-local est représenté sur la figure 4.8e.

Les résultats du débruitage par maximum a posteriori avec un modèle de bruit de Rayleigh et un a priori de type variation totale sur l'amplitude sont représentés sur les figures 4.6g,et 4.8f.

Finalement les résultats obtenus par moyenne a posteriori avec un a priori de type variation totale ($\Phi^{\ell_1}_{\text{diff}}$) puis un a priori plus adapté aux amplitudes radar ($\Phi^{\ell_\alpha \to \ell_0}_{\text{local}}$ avec α=0.005) sont représentés sur les figures 4.6h,4.8g puis 4.6i,4.8h respectivement.

Les comparaisons entre tous ces résultats sont effectuées à l'aide des critères précédemment présentés et sont reportés dans le tableau 4.1. De plus les images de "bruit" obtenues par le rapport entre les données et les images débruitées sont représentées sur les figures 4.7 et 4.9. Comme on l'a vu précédemment aucun de ces critères pris séparément ne permer de caractériser indiscutablement la qualité du débruitage effectué. Néanmoins une image idéalement débruitée minimise l'ensemble de ces critères. On s'attend donc quand même à ce que le recoupement des différents critères reflète à peu près la qualité de débruitage en complément de l'appréciation visuelle. Etant donné que les erreurs ne sont pas normalisées pour le critère $\nu_{\text{g}}^{\text{PSNR}}$, on s'attend d'autre part à ce que celui-ci soit plus sensible et donc représentatif des erreurs effectuées sur les cibles brillantes par rapport aux critères $\nu_{\text{g}}^{\text{Err1}}$ puis $\nu_{\text{g}}^{\text{Err2}}$ (lesquels sont respectivement normalisés par l'amplitude de la vérité terrain puis par son carré).

En considérant donc l'ensemble des critères ainsi que l'impression visuelle et les images de bruit, il apparaît indubitablement que les méthodes de débruitage par simple seuillage dur dans une base d'ondelettes ou par estimation MAP et PM avec un a priori Φ^{ℓ_1} sur la norme du gradient local ne sont pas les mieux adaptées pour le débruitage des images radar en amplitude. En effet les approches MAP et PM avec des a priori de type variation totale ne préservent pas bien les cibles brillantes et les textures radar, ce qui apparaît notamment sur les images de bruit 4.6g et 4.8f et dans les mauvaises valeurs de $\nu_{\text{g}}^{\text{PSNR}}$ par rapport aux valeurs des autres critères. De plus l'approche de débruitage par ondelettes présente de trop nombreux artefacts ainsi qu'une perte de contraste sur les cibles brillantes.

Il semble ensuite assez difficile de départager catégoriquement les autres méthodes (moyennes non-locales, winsar, courbelettes, PM avec un a priori $\Phi^{\ell_\alpha \to \ell_0}_{\text{local}}$) sans définir de critère spécifique associé à l'utilisation finale de ces images. Si l'on compare en détail les images et que l'on caricature les impressions visuelles, l'approche des moyennes non-locales semble sur certains motifs de l'image plus fidèle que les autres à la vérité terrain mais largement moins fidèle à d'autres endroits (détails avec faible contraste complêtement lissés notamment). L'approche par moyenne a posteriori semble quand à elle plus uniformément fidèle sur l'ensemble de l'image, exception faite de certains contours brillants, tout comme l'approche de seuillage bayésien winsar. Il apparaît ainsi plus judicieux de chercher à pénaliser une norme anisotrope de gradient le long de ces contours brillants ou bien de diminuer localement la valeur du poids des a priori. L'approche par courbelettes semble quand à elle relativement fidèle à la vérité terrain sur l'ensemble des images mais la présence excessive de motifs curvilignes et oscillants la pénalise trop lourdement. Il apparaît ainsi intéressant de chercher à modéliser plus finement le bruit comme cela est fait avec winsar par exemple, et à représenter l'image dans

	ν_g^{Err1} ($\searrow$)	ν_g^{Err2} ($\searrow$)	ν_g^{PSNR} ($\nearrow$)
image de Saint-Gervais bruitée	**0.23**	**0.99**	4.14
seuillage dur ondelettes	9.2 e-02	0.59	4.71
seuillage dur courbelettes	7.1 e-02	0.43	5.72
seuillage bayésien Winsar	**6.0 e-02**	**0.33**	5.67
Moyennes non-locales	9.5 e-02	0.80	6.34
MAP Rayleigh+Φ^{ℓ_1}	0.10	0.57	**3.18**
PM Rayleigh+Φ^{ℓ_1}	0.12	0.83	3.76
PM Rayleigh+$\Phi^{\ell_\alpha \to \ell_0}_{\text{local}}$	6.5 e-02	0.41	**7.62**
image de Toulouse bruitée	0.23	1.01	4.95
seuillage dur courbelettes	7.2 e-02	**0.42**	4.28
Moyennes non-locales	8.6 e-02	0.96	6.38
PM Rayleigh+$\Phi^{\ell_\alpha \to \ell_0}_{\text{non-local}}$	**0.26**	**2.5**	5.79
MAP Rayleigh+Φ^{ℓ_1}	0.11	0.42	**3.00**
PM Rayleigh+Φ^{ℓ_1}	0.12	0.77	3.10
PM Rayleigh+$\Phi^{\ell_\alpha \to \ell_0}_{\text{local}}$	6.6 e-02	0.49	6.56
PM Rayleigh+$\Phi^{\ell_\alpha \to \ell_0}_{\text{"hybride"}}$	**6.5 e-02**	0.47	**6.76**

TAB. 4.1: Valeurs des critères de comparaison en fonction des algorithmes de débruitage envisagés (les meilleures valeurs sont en rouge et les pires sont en vert).

un dictionnaire contenant les courbelettes et permettant de réduire les artéfacts associés. On note par ailleurs que certaines cibles brillantes sont fortement atténuées dans l'image obtenue avec les courbelettes ou winsar, ce à quoi il conviendrait également de remédier.

Finalement on remarque que l'approche de moyenne a posteriori avec une pénalisation non-locale du gradient n'est efficace que pour le débruitage des contours brillants. Cela se reflète d'ailleurs dans les valeurs des critères obtenus pour cette méthode. Il peut donc apparaître judicieux de pénaliser le gradient non-local pour les contours brillants et de pénaliser le gradient local pour le reste de l'image. En effet cela correspond grosso modo à imposer un lissage anisotrope le long des contours et un lissage isotrope pour le reste de l'image. De plus, la mesure de similarité entre patches peut sembler adaptée pour sélectionner l'anisotropie adéquate pourvu que l'on restreigne la taille des fenêtres de comparaisons entre patches et que l'on dispose d'une méthode adaptée pour sélectionner les pixels des contours brillants. Dans un premier temps j'ai utilisé une stratégie heuristique de seuillage sélectif pour détecter ces contours à partir des données. Les résultats de cette approche hybride sont donnés sur les figures 4.8i et 4.9h

4.3.2 Données réelles

Je présente finalement dans cette section les résultats de débruitage obtenus sur des vraies données radar RSO, pour les 3 meilleurs algorithmes de débruitage considérés (winsar exclu pour des raisons d'implémentation en puissance de 2). Les images de "vérité terrain" étant approchées (moyenne temporelle ou moyenne en sous-bande), il apparaît encore plus compliqué et arbitraire que précédemment d'établir un classement des résultats obtenus à l'aide de ces approches (figures 4.10,4.11,4.12,4.13)

4.4 Conclusion

L'objectif du projet REI était de comparer les performances de différents algorithmes d'optimisation pour le débruitage par minimisation de fonctionnnelles bayésiennes avec des a priori de type variation totale et un terme d'attache aux données non-convexe (bruit multiplicatif de Rayleigh). Comme on l'a vu précédemment toutes les approches d'optimisation envisagées (optimisation différentiable, optimisation combinatoire et échantillonnage MCMC) fournissent des résultats très similaires, moyennant quelques stratégies empiriques et ajustements permettant de s'accomoder de la non-convexité et de la multi-modalité des distributions a posteriori. Seule l'approche d'optimisation combinatoire par coupure sur graphe possède des preuves théoriques de convergence en temps et en précision finis, ce qui la rend tout particulièrement adaptée pour le problème considéré. Néanmoins étant donné la quantité de mémoire nécessaire pour mettre en pratique cette approche sur des images de grande dimension, il reste tout à fait intéressant de pouvoir recourir aux approches alternatives d'optimisation différentiable et stochastique sans perte notable de qualité.

Chemin faisant les résultats obtenus avec l'estimateur MAP pour un modèle de bruit de Rayleigh et un a priori de lissage local Φ^{ℓ_1} ont été comparés avec des techniques alternatives

(a) vérité terrain

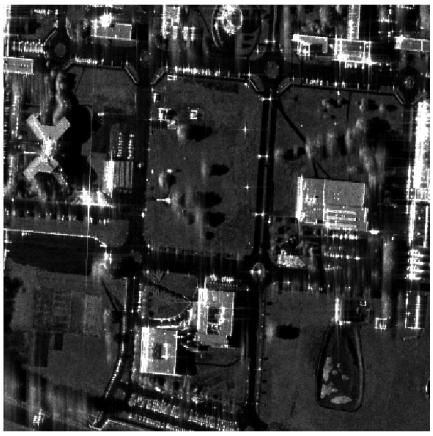
(b) réalisation temporelle

(c) moyennes non-locales

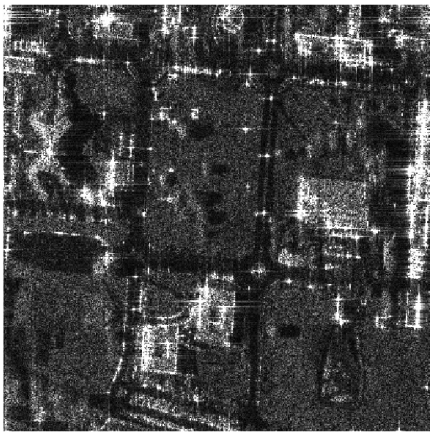
(d) seuillage dur courbelettes

(e) PM Rayleigh+$\Phi^{\ell_\alpha \to \ell_0}_{\text{local}}$

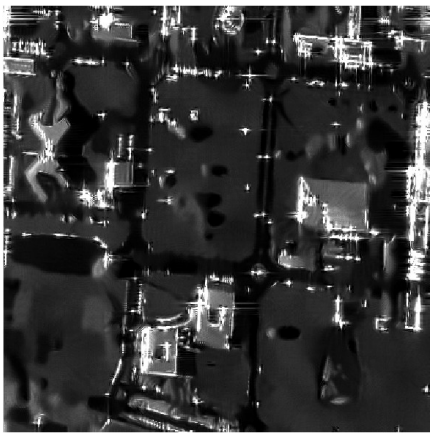
FIG. 4.10: Résultats des meilleures techniques de débruitage envisagées sur une image radar RSO de Saint-gervais (capteur TerraSAR-X).



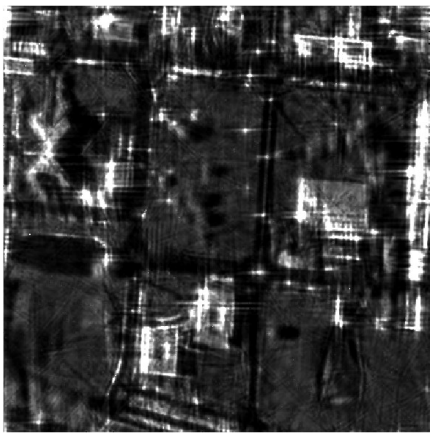
(a) vérité terrain



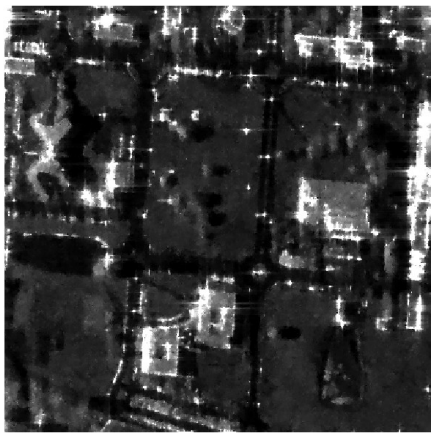
(b) réalisation en sous-bande



(c) moyennes non-locales

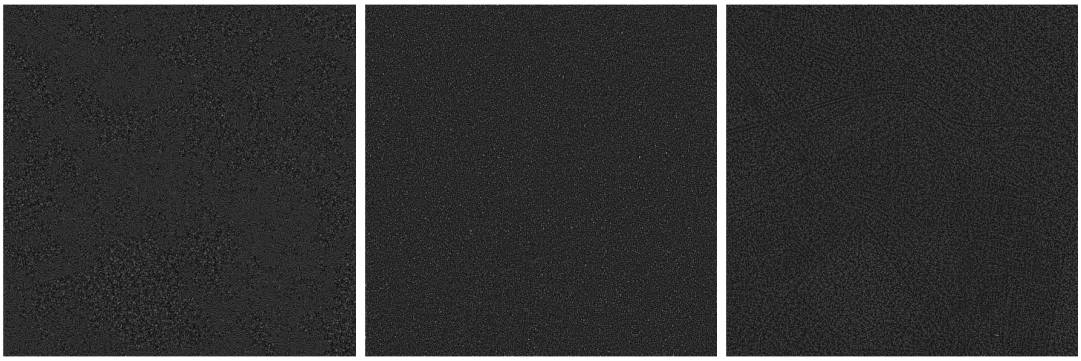


(d) seuillage dur courbelettes



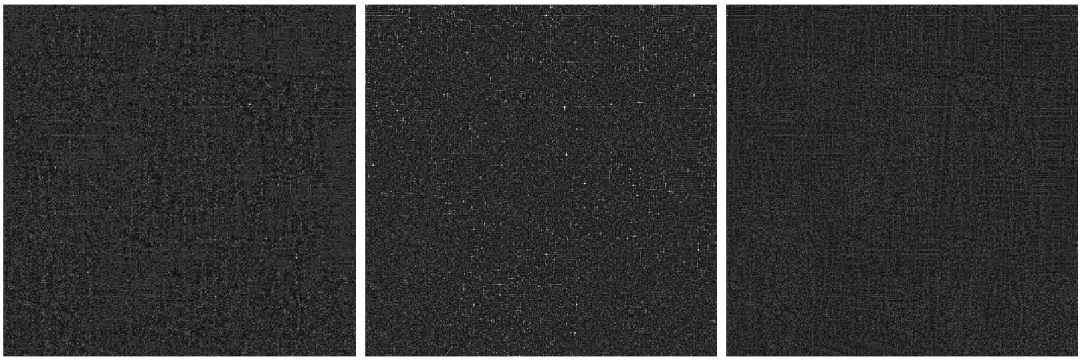
(e) PM Rayleigh+$\Phi_{\text{local}}^{\ell_\alpha \to \ell_0}$

FIG. 4.11: Résultats des meilleures techniques de débruitage envisagées sur une image radar RSO en sous-bande de Toulouse (CNES).



(a) moyennes non-locales (b) seuillage dur courbelettes (c) PM Rayleigh+$\Phi^{\ell_\alpha \to \ell_0}_{\text{local}}$

FIG. 4.12: Images de bruit obtenues par le rapport entre la réalisation bruitée de Saint-Gervais et les images débruitées associées.



(a) moyennes non-locales (b) seuillage dur courbelettes (c) PM Rayleigh+$\Phi^{\ell_\alpha \to \ell_0}_{\text{local}}$

FIG. 4.13: Images de bruit obtenues par le rapport entre la réalisation bruitée de Toulouse et les images débruitées associées.

de débruitage multiplicatif non-stationnaire. Les résultats se sont révélés peu compétitifs par rapport aux approches de représentation parcimonieuse dans des trames et de moyennage non-local notamment, si bien que j'ai proposé de considérer l'estimateur bayésien de moyenne a posteriori (PM) plutôt que le maximum (MAP) des mêmes distributions, et d'utiliser un a priori de lissage $\Phi^{\ell_\alpha \to \ell_0}$ mieux adapté au contenu radar en amplitude (notamment pour la préservation des cibles et des textures brillantes). Les résultats obtenus sont tout à fait compétitifs avec l'état de l'art du débruitage radar RSO à ma connaissance, bien qu'ils ne les surclassent pas non plus. Il apparaît d'ailleurs difficile de sélectionner une technique de débruitage préférentiellement aux autres, sans critère spécifiquement et incontestablement bien adapté à l'utilisation finale des images RSO considérées. Si l'on souhaite fortement privilégier la préservation des cibles brillantes, il peut sembler plus judicieux de pénaliser une norme anisotrope le long des contours brillants au lieu d'une norme isotrope de gradient. Etant donné que la mesure de similarité entre patches reflète partiellement l'anisotropie des contours, j'ai proposé de remplacer la pénalisation locale par une pénalisation non-locale sur les structures brillantes de l'image. L'approche d'estimation par moyenne a posteriori ainsi développée avec un champ de Markov "hybride" améliore légèrement la préservation des cibles brillantes par rapport aux résultats obtenus avec un champ "isotrope" $\Phi^{\ell_\alpha \to \ell_0}$, et peut par exemple permettre d'imposer un lissage isotrope plus important du fond bruité de l'image sans perte de contraste sur les cibles brillantes.

De nombreuses améliorations de l'algorithme de débruitage finalement proposé sont encore envisageables, à commencer par la pénalisation anisotrope des contours brillants. Le fait de considérer des poids de lissage spatialement variables semble également particulièrement prometteur bien que la mise en œuvre ne soit pas triviale. D'autre part étant donné les bonnes performances obtenues à l'aide d'un simple seuillage dur dans une base de courbelettes, il semble tout aussi pertinent de poursuivre dans cette voie-là par une meilleure modélisation du bruit et par l'utilisation d'un dictionnaire permettant la réduction des artefacts observés.

Bibliographie

[1] "Processing of synthetic aperture radar images" ; par H. Maître ; dans digital signal and image processing series, 2008.

[2] "Application de la transformée de Mellin : Etude des lois statistiques de l'imagerie cohérente" ; par J.-M. Nicolas ; dans un rapport interne du groupe TSI de l'institut Télécom Paris, 2006.

[3] "Some fundamental properties of speckle" ; par J.W. Goodman ; dans J. Opt. Soc. Am., vol. 66, pp. 1145–1150, 1976.

[4] "Nonlinear total variation based noise removal algorithms" ; par L. Rudin, S. Osher et E. Fatemi ; dans Physica D, vol. 60, pp.259–268, 1992.

[5] "A note on nice-levable MRFs for SAR image denoising with contrast preservation" ; par J. Darbon, M. Sigelle et F. Tupin ; dans SPIE Electronic Imaging, 2007.

[6] "Recent developments in Total Variation image restoration" ; par T. Chan, S. Esedoglu, F. Park et A. Yip ; dans Mathematical Models in Computer Vision : The Handbook, Dec. 2004.

[7] "A review of image denoising algorithms, with a new one" ; par A. Buades, B. Coll et J.M. Morel ; dans SIAM Multiscale Model. Simul., vol. 4, no. 2, pp. 490–530, 2005.

[8] "Iterative Weighted Maximum Likelihood Denoising with Probabilistic Patch-Based Weights" ; par C.-A. Deledalle, L. Denis et F. Tupin ; dans IEEE Trans. on Im. Proc.Journal of Math. Imaging and Vision, vol. 18, no. 12, pp. 2661–2672, 2009.

[9] "Atomic Decomposition by basis pursuit" ; par S. Chen, D. Donoho et M. Sauders ; dans SIAM J. Sci. Comp., vol. 20, no. 1, pp. 33–61.

[10] "Wave propagation and sampling theory" ; par J. Morlet et al. ; dans Geophysics, vol. 47, 1982.

[11] "Ondelettes et opérateurs" ; par Y. Meyer ; édition Hermann, 1988.

[12] "Orthonormal bases of compactly supported wavelets ; par I. Daubechies ; dans Communications in pure and applied mathematics, vol. 41, pp. 909–996, 1988.

[13] "A theory for multiresolution signal decomposition : the wavelet representation" ; par S. Mallat ; dans IEEE trans. on PAMI, vol. 11, no. 7, 1989.

[14] "Wavelets and signal processing" ; par O. Rioul et M. Vetterli ; dans IEEE SP magazine, pp. 14–38, Oct. 1991.

[15] "Ideal spatial adaptation by wavelet shrinkage" ; par D.L. Donoho et I.M. Johnstone ; dans Biometrika, vol. 81, no. 3, pp. 425–455, 1994.

[16] "Adapting to unknown smoothness via wavelet shrinkage" ; par D.L. Donoho et I.M. Johnstone ; dans J. Acoust. Soc. Am., vol. 90, 1995.

[17] "Recovering edges in ill-posed inverse problems : Optimality of curvelet frames" ; par E.J. Candès et D.L. Donoho ; dans Ann. Statist., vol. 30, pp. 784–842, 2002.

[18] "SAR image denoising via bayesian wavelet shrinkage based on heavy-tailed modeling" ; par A. Achim, P. Tsakalides, A. Bezerianos ; dans IEEE Trans. on Geoscience and remote sensing, vol. 41, no. 8, Août 2003.

[19] "Translation-invariant denoising" par R.R. Coifman et D.L. Donoho ; dans un rapport de recherche des universités de Yale et Stanford, 1995.

[20] "Multiplicative noise removal using L1 fidelity on frame coefficients" ; par S. Durand, J. Fadili, M. Nikolova ; dans Journal of Math. Imaging and Vision, 2010.

[21] "Numerical optimization, second edition" ; par J. Nocedal et S.J. Wright ; dans Springer series in operations research, 2006.

[22] "Convex optimization" ; par S. Boyd et L. Vandenberghe ; édité par Cambridge University Press ; 2004.

[23] "Numerical methods for large-scale nonlinear optimization" ; par N. Gould, D. Orban et Ph. Toint ; dans un rapport technique du CCLRC, Nov. 2004.

[24] "Convergence conditions for ascent methods" ; par P. Wolfe ; dans SIAM review, vol. 11, no. 2, pp. 226–235, 1969.

[25] "Convergence conditions for ascent methods : some corrections" ; par P. Wolfe ; dans SIAM review, vol. 13, no. 2, pp. 185–188, 1971.

[26] "Convergence of conjugate gradient methods with a closed-form stepsize formula" ; par C. Labat et J. Idier ; dans J. Opt. Th. Appl., 2007.

[27] "Variable metric methods for minimization" ; par W.C. Davidon ; dans un rapport technique de Argonne National Lab Report IL, 1959.

[28] "Updating quasi-newton matrices with limited storage" ; par J. Nocedal ; dans Mathematics of computation, vol. 35, no. 151, pp. 773–782, 1980.

[29] "Global convergence of a class of quasi-newton methods on convex problems" ; par R.H. Byrd, J. Nocedal et Y. Yuan ; dans SIAM Journal of Numerical Analysis, vol. 24, pp. 1171–1190, 1987.

[30] "Inexact newton methods" ; par R.S. Dembo, S.C. Eisenstat et T. Steihaug ; dans SIAM Journal of Numerical Analysis, vol. 19, pp. 400–408, 1982.

[31] "A survey of truncated-Newton methods" ; par S.G. Nash ; dans Journal of Computational and Applied Mathematics, vol. 124, pp. 45–59, 2000.

[32] "Function minimization by conjugate gradients" ; par R. Fletcher et C.M. Reeves ; dans Computer Journal, vol. 7, pp. 149–154, 1964.

[33] "Computing a trust-region step" ; par J.J. Moré et D.C. Sorensen ; dans SIAM J. Sci. Stat. Comput., vol. 4, no. 3, pp. 553–572, Sept. 1983.

[34] "A new algorithm for unconstrained optimization" ; par M.J.D. Powell ; dans Nonlinear Programming, 1970.

[35] "On iterated-subspace minimization methods for non-linear optimization" ; par A.R. Conn, N. Gould, A. Sartenaer et Ph. Toint ; dans un rapport Technique de IBM TJ Watson Research Center, 1994.

[36] "A subspace minimization method for the trust-region step" ; par J.B. Erway et P.E. Gill ; dans SIAM Journal on Optimization, vol. 20, pp. 1439-1461, 2009.

[37] "Maximum entropy image reconstruction : general algorithm" ; par J. Skilling et R. Bryan ; dans Mon. Not. R. Astron. Soc., vol. 211, pp. 111–124, 1984.

[38] "Non parametric reconstruction of distribution functions from observed galactic disks" ; par C. Pichon et E. Thiébaut ; dans Mon. Not. R. Astron. Soc., 1997.

[39] "The conjugate gradient method and trust regions in large scale optimization" ; par T. Steihaug ; dans SIAM J. Numer. Anal., vol. 20, pp. 626–637, 1983.

[40] "Towards an efficient sparsity exploiting Newton method for minimization" ; par Ph. Toint ; dans Sparse Matrices and Their Uses, I. S. Duff, ed., London and New York, Academic Press, pp. 57–88, 1981.

[41] "On the truncated conjugate-gradient method" ; par Y. Yuan ; dans Math. Program., Ser. A, vol. 87, no. 3, pp. 561–573, 2000.

[42] "Imagerie à travers la turbulence : mesure inverse du front d'onde et centrage optimal" ; par X. Rondeau dans ma thèse de doctorat en physique et Astrophysique à l'UCBL/ENSL, 2007.

[43] "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision" ; par Y. Boykov et V. Kolmogorov ; dans IEEE Trans. on PAMI, vol. 26, no. 9, Nov. 2004.

[44] "Maximal flow through a network" ; par L.R. Ford et D.R. Fulkerson ; dans Canadian Journal of Mathematics, vol. 8, pp. 399–404, 1956.

[45] "Exact optimization for Markov random fields with convex priors" ; par H. Ishikawa ; dans IEEE Trans. on PAMI, vol. 25, no. 10, pp. 1333–1336, Oct. 2003.

[46] "Exact maximum a posteriori estimation for binary images" ; par D.M. Greig, B.T. Porteous et A.H. Seheult ; dans Journal of the Royal Stat. Soc. series B, vol. 51, pp. 271–279, 1989.

[47] "Introduction to Monte-Carlo methods" ; par D. Mackay Departement of physics at Cambridge University.

[48] "The Monte-Carlo method" ; par N. Metropolis and S.M. Ulam ; dans J. Am. Stat. Assoc., vol. 44, pp. 335–341, 1949.

[49] "Monte-Carlo sampling methods using markov chains and their applications" ; par W.K. Hastings ; dans Biometrika, vol. 57, pp. 97–109, 1970.

[50] "Inverse problem theory and methods for model parameter estimation" ; par A. Tarantola ; publié par SIAM en 2005.

[51] "Probabilistic inference using Markov Chain Monte-Carlo methods" ; par R.M. Neal ; dans un rapport technique du Dept. of Comput. Sci. de l'université de Toronto en 1993.

[52] "Hybrid Monte-Carlo" ; par S. Duane, A. Kennedy, B. Pendleton et D. Roweth ; dans Phys. Lett. B, vol. 195, pp. 216–222, 1987.

[53] "Exploring Hybrid Monte-Carlo in Bayesian Computation" ; par L. Chen, Z. Qin et J. Liu ; dans ISBA proceedings, 2001.

[54] "The multiple-try method and local optimization in Metropolis sampling" ; par J. Liu, F. Liang et W. Wong ; dans J. Amer. Stat. Assoc., vol. 95, pp. 121–134, 2000.

[55] "Shadow Hybrid Monte Carlo : An efficient propagator in phase space of macromolecules" ; par J.A. Izaguirre et S.S. Hampton ; dans J. Comput. Phys., vol. 200, pp. 581-–604, 2004.

[56] "The Targeted Shadowing Hybrid Monte Carlo (TSHMC) Method" ; par E. Akhmatskaya et S. Reich ; dans New Algorithms for Macromolecular Simulation, 2006.

[57] "Optimal tuning of Hybrid Monte-Carlo" ; par A. Beskos, N. Pillai, G. Roberts, J. Sanz-Serna et A. Stuart ; dans DSPDE proceedings, 2010.

[58] "Total variation denoising using posterior expectation" ; par C. Louchet et L. Moisan ; dans EUSIPCO Proceedings, 2008.

[59] "Modèles variationnels et bayésiens pour le débruitage d'images : de la variation totale vers les moyennes non-locales" ; par C. Louchet ; dans sa Thèse de doctorat, spécialité Mathématiques appliquées de l'université Paris Descartes, Dec. 2008.

[60] "On the geometric ergodicity of hybrid samplers" ; par G. Fort, E. Moulines, G. Roberts et J. Rosenthal ; dans S. J. Appl. Probab., vol. 40, no. 1, pp. 123–146, 2003.

[61] "From Metropolis to diffusions : Gibbs states and optimal scaling" ; par L. Breyer et G. Roberts ; dans Stochastic Process. Appl., vol. 90, no. 2, pp. 181–206, 2000.

[62] "Optimal scaling for various Metropolis-Hastings algorithms" ; par G. Roberts et J. Rosenthal ; dans Statist. Sci., vol. 16, no. 4, pp. 351–367, 2001.

[63] "Optimization by simulated annealing" ; par S.C. Kirkpatrick, D. Gelatt et M.P. Vecchi ; dans Science, vol. 220, pp. 2784–2796, 1983.

[64] "Simulated tempering - a new Monte-Carlo scheme" ; par E. Marineri et G. Parisi ; dans Europhysics Letters, vol. 19, no. 6, pp. 451–458, 1992.

[65] "Annealing Markov Chain Monte Carlo with applications to ancestral inference" ; par C. Geyer et E. Thompson ; dans le rapport technique no. 589 de l'université du Minnesota, 1994.

[66] "Annealed importance sampling" ; par R. Neal ; dans Statistics and Computing, vol. 11, no. 2, pp. 125–139, 2001.

[67] "A Variational Approach to remove Multiplicative Noise" ; par G. Aubert et J.-F. Aujol ; dans SIAM Journal on App. Math., vol. 68, no. 4, pp 925–946, Janvier 2008.

[68] "Phase retrieval from speckle images" ; par X. Rondeau, E. Thiébaut, M. Tallon, et R. Foy ; dans Josa A., vol. 24, no. 10, pp. 3354-3365, 2007.

Dépôt légal : 2010 –4ème trimestre
Imprimé à Télécom ParisTech – Paris
ISSN 0751-1345 ENST D (Paris) (France 1983-9999)

Télécom ParisTech

Institut TELECOM - membre de ParisTech

46, rue Barrault - 75634 Paris Cedex 13 - Tél. + 33 (0)1 45 81 77 77 - www.telecom-paristech.frfr

Département TSI

© Institut TELECOM -Télécom ParisTech 2010