



HAL
open science

**Estimation régularisée d'attributs fractals par
minimisation convexe pour la segmentation de textures :
formulations variationnelles conjointes, algorithmes
proximaux rapides et sélection non supervisée des
paramètres de régularisation; Applications à l'étude du
frottement solide et de la microfluidique des écoulements
multiphasiques**

Barbara Pascal

► **To cite this version:**

Barbara Pascal. Estimation régularisée d'attributs fractals par minimisation convexe pour la segmentation de textures : formulations variationnelles conjointes, algorithmes proximaux rapides et sélection non supervisée des paramètres de régularisation; Applications à l'étude du frottement solide et de la microfluidique des écoulements multiphasiques. Analyse de données, Statistiques et Probabilités [physics.data-an]. Université de Lyon, 2020. Français. NNT : 2020LYSEN042 . tel-02971612

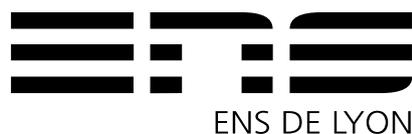
HAL Id: tel-02971612

<https://theses.hal.science/tel-02971612v1>

Submitted on 19 Oct 2020

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

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



Numéro National de Thèse : 2020LYSEN042

THÈSE de DOCTORAT DE L'UNIVERSITÉ DE LYON

opérée par

l'École Normale Supérieure de Lyon

École Doctorale N°52

École Doctorale de Physique et Astrophysique de Lyon (PHAST)

Spécialité de doctorat : Physique, Traitement du signal et des images

Discipline : Physique

Soutenue publiquement le 30 septembre 2020, par :

Barbara PASCAL

Estimation régularisée d'attributs fractals par minimisation convexe pour la segmentation de textures :

Formulations variationnelles conjointes, algorithmes proximaux rapides et sélection non supervisée des paramètres de régularisation ;

Applications à l'étude du frottement solide et de la microfluidique des écoulements multiphasiques.

Devant le jury composé de

Mme Laure Blanc-Féraud,	Directeur de recherche CNRS	Université Nice-Sophia-Antipolis	Examinatrice
M. Bruno Torrèsani,	Professeur des Universités	Aix-Marseille Université	Rapporteur
M. Gabriel Peyré,	Directeur de recherche CNRS	ENS-PSL	Rapporteur
M. Rémi Bardenet,	Chargé de recherche CNRS	Université de Lille	Examineur
M. Jean-François Giovannelli,	Professeur des Universités	Université de Bordeaux	Examineur
Mme Nelly Pustelnik,	Chargée de recherche CNRS	ENS de Lyon	Co-directrice
M. Patrice Abry,	Directeur de recherche CNRS	ENS de Lyon	Directeur

*À Jacques et Claudine,
Lucien et Josette.*

Résumé

La segmentation est un problème central en traitement d'images. Elle est nécessaire pour des applications aussi variées que le diagnostic médical assisté par ordinateur, la conduite autonome, ou encore pour le suivi d'expériences de physique. Afin de segmenter des photographies, des clichés biomédicaux, et plus largement des images complexes issues d'expériences *réelles*, il est nécessaire de prendre en compte leurs propriétés fines. L'aspect *texturé* des images à analyser fait partie des principales propriétés discriminantes pour la segmentation d'images réelles. Or, si les textures sont largement et facilement identifiées par l'œil humain, ce qui permet, par exemple, d'identifier sur la photographie d'un objet le matériau dans lequel il est réalisé, leur description mathématique et leur détection *via* des procédures automatiques de traitement d'image soulèvent de nombreuses difficultés, allant du choix de modélisation, à la gestion de gros volumes de données, en passant par la localisation précise des frontières entre textures distinctes.

Dans cette thèse, nous introduisons deux procédures de segmentation d'images texturées, s'appuyant conjointement sur deux attributs fractals : l'exposant de Hölder, quantifiant la régularité locale d'une texture, et la variance locale. Ces deux procédures reposent sur la construction de deux fonctionnelles convexes régularisantes dont la minimisation fournit des estimées constantes par morceaux des attributs fractals, dont sont ensuite dérivées des segmentations, grâce à une étape de seuillage *a posteriori*. En réalisant simultanément l'estimation et la régularisation des attributs de texture, l'erreur d'estimation est réduite par rapport aux méthodes de l'état-de-l'art dans lesquelles estimation et régularisation sont réalisées en deux phases successives disjointes, et où l'erreur d'estimation se cumule avec le biais de régularisation.

En outre, un modèle de textures fractales homogènes par morceaux, où chaque texture est caractérisée par sa régularité et sa variance locales, est construit et accompagné d'une procédure de synthèse. Cette modélisation fournit des images composées d'un assemblage de textures fractales d'attributs fixés et de segmentation connue, qui sont utilisées pour évaluer les performances des méthodes proposées.

Pour les deux procédures proposées, des schémas numériques rapides de minimisation sont mis en œuvre, grâce à l'étude détaillée des propriétés des fonctionnelles impliquées, et tout particulièrement leur forte-convexité. Puis, les performances de segmentation et les coûts de calcul de ces deux méthodes sont comparés. Afin de témoigner des performances des méthodes proposées, nous les comparons dans un premier temps aux méthodes non supervisées de l'état-de-l'art en segmentation de texture. Dans un second temps, nous nous intéressons aux approches récentes de segmentation par apprentissage statistique. Ainsi, nous élaborons à partir des travaux actuels sur les réseaux de neurones un formalisme de segmentation de textures fractales grâce à des réseaux de neurones convolutionnels, auquel sont comparées les méthodes par minimisation de fonctionnelles convexes proposées.

Le principal point limitant l'utilisation des méthodes variationnelles proposées pour des applications est leur grande sensibilité au réglage des paramètres de régularisation qu'elles font intervenir. En effet, en l'absence de toute connaissance préalable sur la vérité terrain, quantifier objectivement les performances de segmentation est impossible et il est alors nécessaire de s'ap-

puyer sur la décision d'un « expert », qui est prise à partir de critères entachés de biais de subjectivité. De plus, ce réglage est d'autant plus coûteux en temps de calcul, que l'on souhaite déterminer finement les paramètres optimaux. Nous construisons donc une procédure de réglage automatique d'hyperparamètres pilotée par les données, valable en particulier pour les problèmes de régularisation par minimisation de fonctionnelles convexes, sous l'hypothèse d'observations corrompues par un opérateur de déformation linéaire et un bruit gaussien de matrice de covariance très générale.

Les méthodes développées pour la segmentation de textures, auxquelles est adjoint le formalisme de réglage automatique des paramètres de régularisation, sont appliquées à des images provenant d'expérience étudiant les écoulements multiphasiques en milieu poreux. Une seconde application à la physique non linéaire du formalisme de réglage automatique des paramètres de régularisation est présentée : elle consiste en un débruitage linéaire par morceaux de signaux de force mesurés dans des expériences de frottement solide.

Abstract

Segmentation is a burning issue in image processing. It is needed for various applications, including computer-assisted medical diagnosis, autonomous driving and monitoring of physics experiments. In order to segment photographs, biomedical images, and more generally complex images from real experiments, taking into account their fine properties is crucial. The textured aspect of analyzed images is one of the most discriminating feature for real-world image segmentation. Yet, although human eye is very efficient at identifying precisely a lot of texture types, allowing to determine from a photography the material in which an object is made, these tasks appear to raise several challenges when performed by a computer, among which choosing an accurate texture model, dealing with huge amount of data or providing sharp contours between distinct textures.

In this thesis, two texture segmentation procedures, based on the Hölder exponent, measuring the regularity of textures, and on the local variance, are introduced. These two procedures rely on the design of two regularizing convex functionals, whose minimization provides piecewise constant estimates of the fractal features, from which a segmentation is then derived thanks to a posterior thresholding step. Performing simultaneously the estimation and the regularization of the texture features reduces the estimation error, compared to state-of-the-art methods in which the estimation and the regularization are performed in two disjoint successive steps, leading to a cumulation of the estimation error and the regularization bias.

Further, a piecewise homogeneous fractal texture model is proposed, in which each texture is characterized by its local regularity and its local variance, and a synthesis procedure is built. This model provides images composed of several patches of fractal textures, each characterized by given fractals features, arranged according to a prescribed segmentation, both chosen freely by the user. These synthetic piecewise textures are used to evaluate the performance of the proposed texture segmentation methods.

For the two proposed procedures, fast numerical schemes are implemented to solve the associated minimization problem, based on a close study of the properties of the involved functionals, particularly their strong convexity. Then, the performance and the computational cost of the two texture segmentation methods are compared. In order to assess the performance of the proposed procedures, they are first compared to state-of-the-art unsupervised texture segmentation methods. Second, we consider recent neural networks supervised learning for texture segmentation. A deep learning formalism for fractal texture segmentation is elaborated, inspired from recent work using convolutional neural networks for image segmentation. The performance of the derived supervised texture segmentation procedure is compared to the proposed convex functional minimization methods.

The major limitation when applying the proposed variational methods for real-worlds texture segmentation is their high sensibility to the choice of regularization parameters they involve. Indeed, when no prior knowledge on ground truth is available, quantifying objectively the segmentation performance is impossible and it is then necessary to recourse to an "expert" decision, which suffer from subjectivity bias. Further, this hyperparameter selection is all the more time

consuming as fine tuning is required. Thus, we build an automatic and data-driven procedure for hyperparameter selection, particularly suited for regularization performed through convex functional minimization. Theoretical assessment is provided, under the hypothesis of linearly deformed observation, corrupted by additive Gaussian noise, with any covariance matrix.

The texture segmentation methods developed, augmented by the automatic hyperparameter selection formalism, are applied to images from multiphasic flow in porous media experiments. A second application to nonlinear physics consists in using the automatic hyperparameter selection formalism to piecewise linear denoising of force signals, obtained from solid friction experiments.

Remerciements

« Nous recevons sans doute plus de "saluts" que nous n'en donnons et l'homme s'éduque en apprenant à dire merci. »

M. Steffens, *Rien de ce qui est inhumain ne m'est étranger*

Il me faut tout d'abord remercier Patrice Abry et Nelly Pustelnik pour avoir guidé mes pérégrinations scientifiques durant ces trois dernières années. Si je suis aujourd'hui heureuse et fière de présenter mon travail de thèse c'est grâce à leur exigence sans cesse renouvelée que chacun des choix scientifiques de ce travail soit le mien, sans jamais céder à la facilité, ni pour moi, ni surtout pour eux. Pendant trois ans j'ai été écoutée, soutenue tout autant qu'incitée à exercer ma liberté. C'est quelque chose de précieux.

Je remercie Bruno Torrèsani d'avoir participé à mon comité de suivi de thèse et d'être aujourd'hui, avec Gabriel Peyré que je remercie également, rapporteurs de ce travail. Merci à Laure Blanc-Féraud de présider mon jury, et à Rémi Bardenet et Jean-François Giovannelli d'avoir cordialement accepté d'examiner mon travail.

J'ai eu la chance de pouvoir ouvrir mon horizon scientifique par plusieurs collaborations, toutes fortement soutenues par mes encadrants, que je remercie pour cela. Merci à Jean-Christophe Pesquet pour son accompagnement bienveillant dans mes premiers pas avec les algorithmes proximaux par blocs. Au sein du Laboratoire de Physique, le projet SIROCCO a été, durant ces trois années, une source de motivation et de défis multiples, stimulés par l'enthousiasme et la bonne humeur contagieuse de Valérie Vidal, que je remercie chaleureusement pour ses encouragements. Merci également à Thomas Busser d'avoir été un collaborateur de si grande qualité et de m'avoir si bien coachée pour la présentation de la physique des écoulements multiphasiques en milieux poreux à la retraite du laboratoire. Pour la plongée dans le monde des réseaux de neurones profonds, je tiens à remercier Vincent Maudit pour son implication, sa capacité à questionner les outils pratiques, comme les soubassements théoriques, et pour toutes ces discussions où nous avons autant appris l'un que l'autre. Merci à Emmanuel Quemener pour son aide précieuse dans l'utilisation des serveurs de calcul du Centre Blaise Pascal. La collaboration avec Samuel Vaiter m'a beaucoup appris. Du point de vue scientifique tout d'abord, j'ai eu l'occasion de me confronter à d'autres points de vue sur mes problèmes de recherche, ce qui a été une expérience très formatrice. Du point de vue humain également, découvrir d'autres horizons et parcours scientifiques a été très inspirant.

La vie au laboratoire aurait eu beaucoup moins de saveur sans toutes les amitiés du quotidien, que je ne pourrai énumérer de manière exhaustive. Merci à Marion et Roberto d'avoir partagé si généreusement leurs expériences de la recherche avec moi. Merci à l'équipe d'astrophysiciens du M7 (Étienne, Maxime, Hugo, Quentin, Florian) d'avoir été les cobayes de mes expériences culinaires à haute densité calorique. Un immense merci à Hadi, pour m'avoir fait découvrir la si généreuse culture libanaise et pour sa présence bienveillante et chaleureuse durant plus de trois ans.

L'amoureuse du thé que je suis ne saurais oublier les provisions de thé aux algues ramenées régulièrement de Bretagne par Raphaël, et qui ont soutenu nombre de mes réflexions.

J'ai été admirablement entourée durant toutes ces années par mes proches, aucune distance n'a jamais eu raison de leur affection et je ne les remercierai jamais assez pour cela. Merci à Vivian et Mylène pour leur indéfectible amitié qui traverse les années sans jamais s'émousser. Pour leur soutien, leur amour et leur fierté, merci à mes grands-parents. Merci à mes parents, pour leur présence aimante et confiante. Merci de n'avoir eu de cesse de m'inciter à m'engager librement dans les aventures qui m'enthousiasmaient. Pour m'avoir accueillie si chaleureusement, merci aux Gozard et aux Fouretier.

Merci, enfin, à Pierre, spectateur privilégié des dix années d'études qui m'ont amenées jusqu'ici, pour son infinie patience, pour les abbayes et monastères, pour le rire et la tendresse.

Table des matières

Table des matières	ix
1 Introduction	1
2 Segmentation de textures : État de l'art	9
2.1 Segmentation d'images	10
2.2 Textures	27
2.3 Segmentation de texture non supervisée	37
2.4 Apprentissage profond pour la segmentation supervisée	46
3 Synthèse de textures fractales	67
3.1 Champs gaussiens auto similaires	68
3.2 Synthèse de champs gaussiens	78
3.3 Textures fractales homogènes par morceaux	80
3.4 Bilan	82
4 Segmentation de textures fractales par minimisation convexe	85
4.1 Motivations pour l'estimation et la segmentation conjointe	86
4.2 Construction de fonctionnelles pour la segmentation de texture	87
4.3 Optimisation convexe non lisse	89
4.4 Minimisation de fonctionnelles pour la segmentation de texture	94
4.5 Performance d'estimation et de segmentation	106
4.6 Apprentissage profond pour la segmentation de textures fractales	113
5 Stratégies de type Stein pour le réglage des hyperparamètres	125
5.1 Introduction	127
5.2 SURE (<i>Stein Unbiased Risk Estimate</i>) généralisé	131
5.3 SUGAR (<i>Stein's Unbiased Gradient estimator of the Risk</i>) généralisé	135
5.4 Sélection d'hyperparamètres pour la segmentation de textures	143
5.5 Performances de la segmentation automatique	147
5.6 Conclusion	155
6 Applications à la physique	161
6.1 Motivations	162
6.2 Frottement solide en environnement faiblement contraint	163
6.3 Écoulements multiphasiques en milieu poreux	169
6.4 Bilan	183
7 Conclusion	185
A Propriétés du champ brownien fractionnaire	I
B Synthèse de textures fractales par convolution	V

C	Estimateurs de Stein de l'erreur quadratique et de son gradient	IX
C.1	Démonstration du Theorème 5.1	IX
C.2	Démonstration du Théorème 5.2	XI
C.3	Démonstration du Théorème 5.3	XII
C.4	Démonstration de la Proposition 5.4	XVI

Chapitre 1

Introduction

« *Vanitas vanitatum, et omnia vanitas.* »

Qohélet

Traitement d'images. Autrefois concentré sur quelques domaines scientifiques, tels que l'imagerie médicale, le suivi d'expériences en physique ou l'astronomie, le traitement d'image tend aujourd'hui à s'imposer également dans nos usages numériques quotidiens, que ce soit *via* les appareils photos intégrés à nos *smartphones* ou la conduite automobile assistée. Cette situation a amené à un profond changement de paradigme. Tandis que, en physique par exemple, les dispositifs expérimentaux étaient construits de manière à pouvoir être imagés facilement, i.e., en proposant un environnement contrôlé adapté aux possibilités offertes par le traitement d'images, aujourd'hui on souhaite au contraire traiter des données brutes, extraites de situation réelles. Pour cela, il est nécessaire d'extraire les propriétés fines des images, c'est-à-dire non seulement de détecter des formes complexes mais également de savoir discriminer des aspects visuels différents, correspondant à des *textures* distinctes. En outre, les capacités d'acquisition et de traitement de données ayant augmenté substantiellement grâce aux nouvelles technologies, les tâches de traitement d'image se sont complexifiées et requièrent la gestion de volumes de données de plus en plus grands, tout en maîtrisant les efforts de calculs nécessaires.

Une tâche centrale en traitement d'image a cristallisé l'attention des chercheurs ces dernières années : la *segmentation*. Elle consiste à découper une image en un nombre k de régions homogènes, qui forment une partition sans recouvrement de cette image. Parmi les domaines où apparaît naturellement le problème de la segmentation d'images, citons par exemple le diagnostic médical, où l'on cherche à discriminer les tissus sains des tissus malades. La segmentation d'une image donnée est dépendante du critère d'homogénéité retenu. La définition la plus directe de l'homogénéité consiste à rechercher des régions d'intensité lumineuse, ou de couleur, uniforme, ou plus largement à l'intérieur desquelles les variations d'intensité lumineuse ou de couleur sont faibles et suffisamment lentes. Néanmoins, pour des images présentant, par exemple, des motifs périodiques, l'intensité lumineuse et la couleur ne sont pas des indicateurs pertinents sur lesquels s'appuyer pour une segmentation.

Plus généralement, l'aspect *texturé* des images « réelles », comme par exemple les photographies, doit être pris en compte si l'on souhaite obtenir une segmentation correcte. On rencontre en traitement d'image une grande diversité de *textures*, qui apparaissent par exemple naturellement dans les réalisations humaines, telles que les tissus, les grillages, composés de motifs périodiques, ou dans les structures biologiques, telles que les feuilles ou l'écorce des arbres, pouvant également présenter un caractère aléatoire. Ainsi, on voit que le terme de « texture » peut renvoyer à des notions très diverses, de la répétition périodique d'un motif déterministe, à un processus complètement aléatoire caractérisé par sa structure de covariance. Par conséquent, la capacité à distinguer les différentes textures qui composent une image repose sur un choix pertinent d'*attributs*

de texture locaux. Une grande diversité d'attributs de texture ont ainsi été proposés, allant d'histogrammes locaux de l'intensité lumineuse ou de la couleur, à des indicateurs quantifiant localement la périodicité d'un motif, ou mesurant la granularité d'une texture.

Segmentation de textures à l'aide d'attributs. Afin d'effectuer la segmentation précise d'une image texturée à partir d'attributs locaux, ces derniers doivent être estimés, d'une part, le plus fidèlement possible, afin de distinguer correctement les différentes textures, et d'autre part, suffisamment localement, pour obtenir des contours précis. Or, les procédures directes d'extraction des attributs de texture en chaque pixel d'une image souffrent d'une grande variance d'estimation, qui empêche d'en tirer une segmentation pertinente. En traitement d'image, il est d'usage courant, pour réduire le bruit, d'effectuer un filtrage linéaire lissant les estimées. Néanmoins, un tel lissage entraînerait une perte de localisation importante, qui rendrait impossible l'extraction de démarcations franches entre les différentes textures. Il est donc nécessaire, pour diminuer la variance d'estimation, tout en conservant la localisation de l'information portée par les attributs de texture, de se tourner vers des approches plus sophistiquées. Deux voies sont alors envisageables. D'une part, une modélisation adéquate à la fois de la statistique des attributs de texture et de la statistique du bruit, permet de réaliser une estimation *bayésienne* des cartes d'attributs. Pour cela, la distribution *a posteriori* conditionnellement à une observation, i.e., une estimation bruitée des attributs, est maximisée grâce à un formalisme de *Maximum A Posteriori*. Une seconde approche s'appuie sur les propriétés des cartes d'attributs attendues *a priori* pour réaliser une *régularisation* de l'observation bruitée. Les méthodes de régularisation reposent sur la construction et la minimisation d'une *vraisemblance pénalisée*, c'est-à-dire d'une fonctionnelle permettant de gérer un compromis entre un terme de fidélité aux données et un terme de pénalisation favorisant une certaine régularité des estimées. La procédure de régularisation la plus usuelle est de réaliser un débruitage *a posteriori*, via la minimisation d'une fonctionnelle dont l'attache aux données favorise l'adéquation avec l'estimation directe bruitée, et dont la pénalisation favorise la constance par morceaux des cartes d'attributs. Néanmoins, la combinaison de l'erreur d'estimation et du biais dû à la régularisation détériore la carte finale régularisée, et *a fortiori* la segmentation que l'on peut en extraire. On gagne alors à chercher à élaborer un terme d'attache aux données plus complexe, qui permette de combiner ces deux étapes.

Si les outils utilisés pour les approches bayésiennes et pour la minimisation de vraisemblance pénalisée diffèrent, ces deux approches soulèvent les trois mêmes questions pour leur mise en œuvre. Premièrement, le choix des densités de probabilité *a priori* des attributs et du bruit d'estimation dans le cas bayésien, ou le choix de l'attache aux données et de la pénalisation intervenant dans la vraisemblance pénalisée, en fonction des connaissances dont on dispose à la fois sur les images à étudier et sur la tâche à réaliser. Deuxièmement, la construction de schémas numériques bénéficiant de garanties de convergence théoriques solides et suffisamment rapides pour pouvoir être utilisés, ou bien pour l'analyse de volumes de données conséquents, qui sont désormais la norme en traitement d'images, ou bien pour un suivi « en temps réel », par exemple pour le pilotage d'expérience. Troisièmement, les performances de ces méthodes sont fortement dépendantes du choix de leurs hyperparamètres, que sont les paramètres des lois de probabilités utilisées pour décrire les statistiques des attributs et du bruit dans le cas bayésien, ou les paramètres de régularisation pilotant le compromis entre la fidélité aux observations et la satisfaction des contraintes de régularité dans le cas de la vraisemblance pénalisée. Ainsi, pour l'application des deux types de procédures à des images « réelles », ces méthodes doivent s'accompagner de stratégies de réglage automatique des hyperparamètres requérant un minimum d'information préalable.

Apprentissage statistique pour la segmentation. L'augmentation spectaculaire des capacités de calcul ces dernières années a permis l'essor d'une nouvelle approche des problèmes de traitement d'image : les méthodes *supervisées*, qui consistent à « apprendre » une représentation adéquate des

données pour une tâche fixée, à partir d'une base d'entraînement composée de données étiquetées. Les représentations par réseaux de neurones, inspirées du fonctionnement physiologique du cerveau, sont aujourd'hui plébiscitées, notamment pour l'immense richesse d'architectures permise par les réseaux profonds. L'ajustement du réseau aux données d'entraînement, réalisé pendant la phase d'apprentissage, sélectionne automatiquement, et implicitement, les attributs pertinents pour la tâche considérée, qui sont alors encodés dans le réseau appris. Ces méthodes soulèvent des questions nouvelles par rapport aux approches classiques, qui se concentrent autour de deux points. D'une part, il s'agit de proposer, pour un problème donné, l'architecture la plus adaptée pour la représentation des données sous forme d'un réseau de neurones, tout en étant la plus économe possible pour que la phase d'apprentissage soit efficace. D'autre part, réunir une base de données d'entraînement pertinentes et de bonne qualité constitue un réel défi. Dans le cas de la segmentation, récolter des segmentations effectuées manuellement peut s'avérer très coûteux, une voie alternative consiste à générer des données d'entraînement synthétiques, avec le risque d'obtenir un jeu de données insuffisamment varié et que la représentation finale ne soit pas adaptée aux données réelles.

Contributions

L'objet principal de cette thèse est la construction de deux variantes d'une procédure de segmentation de textures non supervisée, s'appuyant sur des attributs fractals, *via* la minimisation d'une *vraisemblance pénalisée*, et leur application à des données issues d'expériences de physique. Nous commençons par proposer un modèle de texture fractale homogène par morceaux à partir de champs gaussiens, permettant de synthétiser des images texturées de géométrie et de caractéristiques données. Puis, deux vraisemblances pénalisées sont proposées. Les propriétés des fonctionnelles associées sont étudiées afin de fournir une méthode de segmentation rapide, possédant à la fois une assise théorique solide, et des modalités de mise en œuvre des algorithmes rapides précises. Ces deux contributions ont fait l'objet d'article [6, ACHA19] soumis au journal *Applied and Computational Harmonic Analysis*, ainsi que d'une communication acceptée en conférence internationale [4, ICIP18]. Un formalisme de réglage automatique et piloté par les données des paramètres de régularisation est construit, et plusieurs résultats théoriques concernant le caractère non biaisé des estimateurs impliqués sont démontrés. Une publication [8, JMIV20], soumise au *Journal of Mathematical Imaging and Vision*, détaille ce formalisme dans un cadre général et sa particularisation au cas de la segmentation de texture. Une boîte à outils MATLAB regroupe les méthodes de segmentation ROF_{*h*}-S, à contours « libres » et à contours « co-localisés » décrites dans cette thèse, avec réglage automatique des paramètres de régularisation, ainsi que leurs équivalents pour l'étude de signaux monofractals homogènes par morceaux [GSUGAR20]¹. Enfin, deux applications à des données issues de la physique non linéaire permettent d'illustrer les performances des procédures proposées dans des conditions réelles. Ces applications ont conduit à un article [7, TELECOM20], accepté au journal *Annals of Telecommunications*, faisant suite à une communication au colloque GRETSI de traitement du signal [5, GRETSI19], ainsi qu'à une boîte à outils MATLAB, disponible en ligne [STEIN20]². En outre, nous avons également étudié la possibilité d'accélérer les algorithmes primaires-dux au moyen de stratégies de mise à jour « par blocs » et discuté de l'influence du choix de ces blocs dans l'accélération de l'algorithme forward-backward sur le dual pour le débruitage par Variation Totale [3, ICASSP]. Si cette stratégie ne s'est pas avérée aussi efficace que les méthodes d'accélération considérées dans [6, ACHA19], et ne sera donc pas décrite dans ce manuscrit, nous travaillons actuellement à généraliser ces résultats afin de déterminer quels choix de blocs aboutissent au taux de convergence optimal.

Le **Chapitre 2** propose un panorama de l'état-de-l'art en traitement d'image, centré sur la segmentation de textures. Tout d'abord, la Section 2.1 permet d'introduire les outils et méthodes

1. <https://github.com/bpascal-fr/gsugar>

2. <https://github.com/bpascal-fr/stein-piecewise-filtering>

propres au traitement d'image, et particulièrement à la segmentation, en présentant les principales méthodes de segmentation d'images non supervisées, divisées en deux grandes familles. Une première classe de méthodes consiste en la construction et la minimisation d'une vraisemblance pénalisée, réalisant un compromis entre la fidélité aux observations et des contraintes de régularité spatiales permettant d'extraire une segmentation pertinente à partir de l'estimée régularisée. Les procédures de segmentation construites au Chapitre 4 seront formulées comme des vraisemblances pénalisées. Une seconde classe de procédures de segmentation regroupe les approches de partitionnement de données, telles que la méthode des k -moyennes ou les méthodes reposant sur une représentation des données par un graphe. Dans un second temps, la Section 2.2 propose de se familiariser avec la notion de *texture* en traitement d'image en explorant cette notion sous deux angles différents : la modélisation, permettant de représenter mathématiquement ce qu'est une texture, et la description au moyen d'attributs, qui s'intéresse aux éléments caractéristiques mesurables des textures. Nous définissons à la Section 2.2 le modèle du champ gaussien, utilisé au Chapitre 3 pour la génération de textures synthétiques, ainsi qu'un premier attribut fractal : l'exposant de Hölder, qui sera utilisé dans cette thèse pour décrire la régularité locale d'une texture. La combinaison des méthodes de segmentation, décrites à la Section 2.1, et des attributs de textures, introduits à la Section 2.2, permet de percevoir la genèse des méthodes de segmentation de textures non supervisées exposées à la Section 2.3. Nous détaillons la méthode de segmentation de textures par débruitage par Variation Totale de la régularité locale [10], à laquelle est adjointe une étape de seuillage itératif [2], appelée ROF $_h$ -S, que nous considérerons comme l'état-de-l'art de la segmentation de texture s'appuyant sur la régularité locale. Enfin, la Section 2.4 se concentre sur les méthodes de segmentation d'image, éventuellement texturées, par apprentissage statistique au moyen des réseaux de neurones convolutionnels profonds, largement représentées dans les contributions récentes de traitement d'image. Introduire l'ensemble des outils nécessaires à la mise en œuvre d'une stratégie d'apprentissage profond permet de souligner les points communs et les points de divergence avec le traitement d'image non supervisé et de comprendre leurs positionnements respectifs dans le domaine du traitement d'image.

Le **Chapitre 3** étudie la modélisation et la synthèse de textures *fractales* homogènes par morceaux dont la géométrie et les caractéristiques sont prescrites, par le biais de champs gaussiens. Cette étape est en effet nécessaire pour quantifier les performances des procédures de segmentation de textures que nous allons proposer. Une première étape est la synthèse de textures fractales homogènes, pour laquelle plusieurs modèles de champs *gaussiens* de régularité locale uniforme sont envisagés. À partir du champ brownien fractionnaire, non stationnaire auto similaire, nous détaillons la construction du bruit gaussien fractionnaire, stationnaire asymptotiquement auto similaire, dont nous pointons les limites pour la synthèse de textures isotropes. Nous proposons donc un champ gaussien stationnaire asymptotiquement auto similaire, que nous définissons à partir d'accroissements horizontaux et verticaux du champ brownien fractionnaire [6, Section 4, ACHA19]. Une analyse détaillée de sa structure de covariance montre que le champ gaussien ainsi construit est caractérisée par sa régularité et sa variance locale. Nous appelons *texture fractale homogène* un tel champ stationnaire. Enfin, une procédure de synthèse de textures fractales *homogènes par morceaux*, de segmentation connue *a priori* et de régularité et variance locales prescrites sur chaque région est détaillée. Nous nous assurons également de l'absence de discontinuités aux frontières entre textures.

Au **Chapitre 4**, nous introduisons et étudions deux procédures de segmentation de textures, utilisant deux attributs fractals, la régularité et la puissance locales, pour discriminer les différentes textures présentes dans une image. Ces deux procédures réalisent conjointement l'estimation des attributs fractals et leur régularisation *via* la minimisation de deux fonctionnelles convexes dont la construction est détaillée à la Section 4.2. Contrairement à la méthode ROF $_h$ -S, les étapes d'estimation et de régularisation ne sont pas réalisées successivement mais simultanément par la minimisation d'une vraisemblance pénalisée, ce qui évite de cumuler erreur d'estima-

tion et biais dû à la régularisation. Les deux fonctionnelles considérées possèdent la même attache aux données, permettant l'extraction des attributs fractals à partir de quantités mutli-échelles, mais diffèrent dans la forme de leur terme de pénalisation, imposant la constance par morceaux des attributs. Une première pénalisation, dite à contours « libres », impose indépendamment le caractère constant par morceaux de la régularité et de la puissance locales, tandis qu'une seconde pénalisation, dite à contours « co-localisés » favorise des sauts de régularité et de variance locales simultanés. Une stratégie de seuillage *a posteriori* fournit ensuite les segmentations associées [6, ACHA19]. Les fonctionnelles à contours « libres » et à contours « co-localisés » sont convexes mais non différentiables et leur minimisation requiert donc des outils spécifiques, présentés à la Section 4.3, qui conduisent aux schémas de minimisation *proximaux*. L'inconvénient majeur de ces schémas numériques est leur vitesse de convergence très faible, qui entraîne des coûts de calculs importants. Pour pallier ce problème, nous étudions plus en détails les propriétés de convexité des fonctionnelles considérées, nous montrons que leur attache aux données commune est fortement convexe, et nous calculons son module de forte-convexité explicitement. Grâce à ce résultat, deux schémas proximaux accélérés sont dérivés et décrits en détails à la Section 4.4. La Section 4.5 compare les trois procédures de segmentation de texture s'appuyant sur les attributs fractals : ROF_h-S, segmentation à contours « libres » et à contours « co-localisés » en terme de pourcentage de pixels bien classés et en terme d'estimation de la régularité locale des différentes textures en présence. Les deux schémas d'optimisation sont également comparés en nombre d'itérations nécessaires et en temps de calcul, ce qui permet de quantifier l'impact majeur de l'accélération permise par la forte convexité de l'attache aux données. Puis, ces méthodes sont comparées aux méthodes de l'état-de-l'art pour la segmentation d'assemblages de textures réelles. Enfin, la Section 4.6 présente plusieurs réseaux de neurones dérivés du réseaux convolutionnel profond proposé par [1] et décrit à la Section 2.4 pour la segmentation de texture supervisée [9, EUSIPCO20]. Ces réseaux sont entraînés sur des bases de données formées de textures fractales homogènes par morceaux suivant le modèle décrit au Chapitre 3. Les performances de segmentation des différents réseaux de neurones construits sont comparées avec la méthode à contours « libres ». En outre, la robustesse des différentes approches à une légère inadéquation entre jeux d'entraînement et données de test est évaluée.

Le **Chapitre 5** s'intéresse au problème général du choix des paramètres de régularisation apparaissant dans les méthodes d'estimation variationnelles par minimisation de vraisemblance pénalisée. Le but est ensuite d'appliquer le formalisme élaboré aux problèmes de segmentation par estimation régularisée de la régularité locale ROF_h-S, à contours « libres » et à contours « co-localisés » introduits au Chapitre 4. En effet, le principal point d'achoppement des méthodes d'estimation *via* une fonctionnelle régularisante est le nécessaire réglage fin des paramètres pour obtenir les performances optimales. Pour la réalisation d'un réglage idéal, il faut, d'une part, connaître *a priori* la vérité terrain, afin de calculer l'erreur d'estimation exacte et, d'autre part, explorer efficacement l'espace des hyperparamètres afin de déterminer le choix d'hyperparamètres minimisant l'erreur. Or, en pratique, on dispose uniquement d'une observation bruitée de la quantité d'intérêt et l'erreur exacte est donc inaccessible. Afin de pallier ce premier obstacle, nous nous sommes tournés vers les approches développées à la suite des travaux originels de Stein [11], qui permettent d'estimer l'erreur commise sans connaître *a priori* la vraie quantité non bruitée, à partir de connaissances sur la statistique du bruit. Initialement formulé dans le cas d'un bruit gaussien i.i.d., l'estimateur de Stein de l'erreur quadratique est étendu au cas d'un bruit gaussien de matrice de covariance générale à la Section 5.2. L'exploration de l'espace des paramètres est réalisée de manière automatique, grâce à l'introduction d'un estimateur du gradient de l'erreur à la Section 5.3, qui permet de minimiser l'estimée de l'erreur quadratique par un algorithme de quasi-Newton [8, JMIV20]. Après la particularisation de ce formalisme de sélection automatique des hyperparamètres au problème de la segmentation de textures, ses performances sont évaluées sur des textures synthétiques fractales homogènes par morceaux.

Nous présentons au **Chapitre 6** deux applications des outils présentés dans cette thèse à la physique non linéaire [7, TELECOM20]. Dans une première partie, nous considérons des signaux de force mesurés dans des expériences étudiant le frottement solide, auxquels nous appliquons un débruitage favorisant un comportement linéaire par morceaux avec la mise en œuvre du formalisme de réglage automatique des paramètres de régularisation construit au Chapitre 5. Une seconde partie s'intéresse à la dynamique des écoulements multiphasiques en milieu poreux. L'analyse des images issues de telles expériences consiste alors en un problème de segmentation de textures qui est traité grâce aux méthodes à contours « libres » et à contours « co-localisés », avec réglage automatique des paramètres de régularisation.

Publications

Articles de journaux

[TELECOM20] B. Pascal, N. Pustelnik, P. Abry, J.-C. Géminard, V. Vidal. Parameter-free and fast nonlinear piecewise filtering. Application to experimental physics. Accepté pour publication dans *Annals of Telecommunications*, 2020 (PDF).

[JMIV20] B. Pascal, S. Vaiter, N. Pustelnik, P. Abry. Automated data-driven selection of the hyperparameters for Total-Variation based texture segmentation. Soumis au *Journal of Mathematical Imaging and Vision*, 2020 (PDF).

[ACHA19] B. Pascal, N. Pustelnik, P. Abry. How Joint Fractal Features Estimation and Texture Segmentation can be cast into a Strongly Convex Optimization Problem?. Soumis au journal *Applied and Computational Harmonic Analysis*, 2019 (PDF).

Communications en conférences avec actes

[EUSIPCO20] B. Pascal, V. Mauduit, P. Abry, N. Pustelnik. Scale-free texture segmentation : Expert feature-based versus Deep Learning strategies. Accepté pour la conférence *European Signal Processing Conference*, 2020 (PDF).

[GRETSI19] B. Pascal, T. Bussier, N. Pustelnik, P. Abry, V. Vidal. Segmentation d'images texturées en grande dimension. Application à l'analyse d'écoulements multiphasiques. Présenté et publié dans les rapports du *XXVII^{ème} colloque GRETSI*, ayant eu lieu à Lille, France, du 26 au 29 août 2019 (PDF).

[ICIP18] B. Pascal, N. Pustelnik, P. Abry, M. Serres, V. Vidal. Joint estimation of local variance and local regularity for texture segmentation. Application to multiphase flow characterization. Présenté et publié dans les rapports de la conférence *IEEE International Conference on Image Processing*, ayant eu lieu à Athènes, Grèce, du 7 au 10 octobre 2018 (PDF).

[ICASSP18] B. Pascal, N. Pustelnik, P. Abry, J.-C. Pesquet. Block-coordinate proximal algorithms for scale-free texture segmentation. Présenté et publié dans les rapports de la conférence *IEEE International Conference on Acoustics, Speech, and Signal Processing*, ayant eu lieu à Calgary, Alberta, Canada, du 15 au 20 avril 2018 (PDF).

Communications en conférences sans actes

[SIAM18] B. Pascal, N. Pustelnik, P. Abry. Combining Local Regularity Estimation and Total Variation Optimization for Scale-Free Texture Segmentation. *SIAM Conference on Imaging Science*, ayant eu lieu à Bologne, Italie, du 5 au 8 juin 2018.

[CURVES18] B. Pascal, N. Pustelnik, P. Abry. Joint estimation of local variance and local regularity for texture segmentation. *Curves and Surfaces*, ayant eu lieu à Arcachon, France, du 28 juin au 4 juillet 2018.

Boîtes à outils - Codes MATLAB

[STEIN20] B. Pascal, N. Pustelnik, P. Abry. Parameter-free and fast nonlinear piecewise filtering for signals, multivariate signals and images. ([GitHub](#))

[GSUGAR20] B. Pascal, S. Vaiteer, N. Pustelnik, P. Abry. Automated data-driven selection of the hyperparameters for Total-Variation based texture segmentation. ([GitHub](#))

Références

- [1] V. Andrearczyk and P. Whelan. Texture segmentation with Fully Convolutional Networks. *arXiv :1703.05230*, 2017. 5
- [2] X. Cai and G. Steidl. Multiclass segmentation by iterated ROF thresholding. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 237–250. Springer, 2013. 4
- [3] B. Pascal, N. Pustelnik, P. Abry, and J.-C. Pesquet. Block-coordinate proximal algorithms for scale-free texture segmentation. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1253–1257, Calgary, Alberta, Canada, Apr. 15-20 2018. IEEE. 3
- [4] B. Pascal, N. Pustelnik, P. Abry, M. Serres, and V. Vidal. Joint estimation of local variance and local regularity for texture segmentation. Application to multiphase flow characterization. In *Proc. Int. Conf. Image Process.*, pages 2092–2096, Athens, Greece, 2018. IEEE. 3
- [5] B. Pascal, T. Busser, N. Pustelnik, P. Abry, and V. Vidal. Segmentation d’images texturées en grande dimension. application à l’analyse d’écoulements multiphasiques. In *GRETSI 2019 XXVIIème Colloque francophone de traitement du signal et des images*, 2019. 3
- [6] B. Pascal, N. Pustelnik, and P. Abry. How joint fractal features estimation and texture segmentation can be cast into a strongly convex optimization problem? *Preprint arXiv :1910.05246 [math.OC]*, 2019. 3, 4, 5
- [7] B. Pascal, N. Pustelnik, P. Abry, J.-C. Géminard, and V. Vidal. Parameter-free and fast nonlinear piecewise filtering. Application to experimental physics. *Preprint arxiv :2006.03297 [physics.data-an]*, 2020. 3, 6
- [8] B. Pascal, S. Vaiteer, N. Pustelnik, and P. Abry. Automated data-driven selection of the hyperparameters for total-variation based texture segmentation. *Preprint arxiv :2004.09434 [stat.ML]*, 2020. 3, 5
- [9] B. Pascal, V. Mauduit, P. Abry, and N. Pustelnik. Scale-free Texture Segmentation : Expert Feature-based versus Deep Learning strategies. 2021. 5
- [10] N. Pustelnik, H. Wendt, P. Abry, and N. Dobigeon. Combining local regularity estimation and total variation optimization for scale-free texture segmentation. *IEEE Trans. Comput. Imaging*, 2(4) :468–479, 2016. 4
- [11] C. M. Stein. Estimation of the mean of a multivariate normal distribution. *The annals of Statistics*, pages 1135–1151, 1981. 5

Chapitre 2

Segmentation de textures : État de l'art

« On ne peut bien voir qu'à condition de ne pas chercher son intérêt dans ce qu'on voit. »

C. Bobin, *Ressusciter*

Sommaire

2.1 Segmentation d'images	10
2.1.1 Minimisation de vraisemblance pénalisée	11
2.1.1.1 Motivation et liens avec les méthodes bayésiennes	11
2.1.1.2 Champs de Markov et densité de Gibbs	12
2.1.1.3 Modèle de Mumford-Shah	15
2.1.1.4 Contours actifs de Chan et Vese	16
2.1.1.5 « Convexification » du problème de partition minimale	18
2.1.1.6 Variation Totale et seuillage itéré <i>a posteriori</i> (ROF _X -S)	20
2.1.1.7 Réglage des hyperparamètres des fonctionnelles régularisantes	22
2.1.2 Méthodes de partitionnement de données (<i>clustering</i>)	23
2.1.2.1 <i>k</i> -moyennes	23
2.1.2.2 <i>c</i> -moyennes relaxées (<i>Fuzzy c-means</i>)	24
2.1.2.3 Ligne de partage des eaux	25
2.1.2.4 Optimisation sur graphe	26
2.2 Textures	27
2.2.1 Intuition et brève catégorisation	28
2.2.2 Modèle de texture	29
2.2.2.1 Modulation en amplitude et en fréquence	29
2.2.2.2 Champ aléatoire markovien	30
2.2.2.3 Mélange gaussien	31
2.2.3 Attributs de texture	32
2.2.3.1 Réponse à des filtres de Gabor	32
2.2.3.2 Attributs fractals et transformée multi-échelle	33
2.3 Segmentation de texture non supervisée	37
2.3.1 Comparaison de patches pour la segmentation	37
2.3.1.1 Détection d'objets texturés par contours actifs	37
2.3.1.2 Segmentation de texture par patches grâce à la norme L^1	39
2.3.2 Modèle bayésien pour la segmentation de texture	41
2.3.3 Segmentation à partir d'attributs de texture locaux	42
2.3.3.1 Factorisation matricielle	42
2.3.3.2 Détection de contours et segmentation hiérarchisée	43
2.3.4 Variation Totale seuillée appliquée à la régularité locale (ROF _h -S)	45

2.3.4.1	Principe	45
2.3.4.2	Texture fractale par morceaux	45
2.3.4.3	Variation Totale seuillée sur la régularité locale	45
2.4	Apprentissage profond pour la segmentation supervisée	46
2.4.1	Apprentissage profond : principes généraux, méthodes et outils	47
2.4.1.1	Objectif	47
2.4.1.2	Réseau de neurones	47
2.4.1.3	Entraînement	48
2.4.2	Réseaux de neurones convolutionnels	49
2.4.3	De la classification à la segmentation d'images par réseaux de neurones	52
2.4.3.1	Classification	52
2.4.3.2	Segmentation	55
2.4.3.3	Apprentissage profond pour les textures	58

2.1 Segmentation d'images

Une tâche majeure en traitement d'image consiste à diviser une image en régions homogènes afin, ensuite, d'identifier le contenu de chaque région et reconnaître, par exemple, un objet, un matériau ou un tissu biologique. On parle alors de segmentation, ou segmentation *sémantique*, pour insister sur le *sens* qu'elle permet de donner à une image, en organisant et en hiérarchisant l'information qu'elle contient. Deux exemples classiques sont l'interprétation de scène, dans laquelle on souhaite identifier le contexte, l'agencement des objets et des individus sur une photographie, et le diagnostic médical, qui vise à discriminer tissus sains et tissus malades dans un contexte anatomique donné.

Mathématiquement, on formule le problème comme suit. Soit $X : \Omega \rightarrow \mathbb{R}^d$ une image de taille $N_1 \times N_2$ pixels (avec $d = 1$ pour une image en niveau de gris et $d = 3$ pour une image en couleur) vue comme un champ discret défini sur le domaine

$$\Omega = \{1, \dots, N_1\} \times \{1, \dots, N_2\}, \quad (2.1)$$

la *segmentation* de l'image X consiste à trouver une partition de Ω en un certain nombre k de régions *homogènes*

$$\Omega = \cup_{i=1}^k \Omega_i, \quad \text{avec } \Omega_i \cap \Omega_{i'} = \emptyset \text{ si } i \neq i'. \quad (2.2)$$

Le sens du mot *homogène* varie d'une application à l'autre. L'approche la plus directe consiste à considérer qu'un ensemble de pixels adjacents partageant la même intensité lumineuse, ou la même couleur, forment une région homogène. Une version plus souple consiste à considérer que des variations suffisamment lentes de l'intensité lumineuse, ou de la couleur, peuvent exister au sein d'une région homogène. Néanmoins, il existe de nombreuses autres définitions, qui permettent de s'adapter au problème considéré. En effet, nous verrons dans la suite que l'homogénéité peut aussi être comprise au sens d'un motif répété, dont un exemple très simple serait l'identification d'un damier sur une photographie, caractérisé par l'alternance de cases blanches et noires. Ainsi, deux acceptions différentes de l'homogénéité peuvent conduire à deux segmentations d'une même image complètement dissemblables. Le choix d'un critère d'homogénéité est, à part entière, un problème de traitement d'image.

La qualité d'une segmentation, du point de vue d'un observateur humain, est fortement dépendante de la cohérence spatiale des régions proposées : l'apparition d'artefacts, de « trous » très petits à l'intérieur d'une région homogène, ou encore de contours extrêmement irrégulier sont souvent de bon indicateurs d'une piètre segmentation. Il est donc pertinent de mettre à profit la

topologie des données pour favoriser certains attendus, comme par exemple la similarité *a priori* entre pixels voisins, sauf en quelques localisations correspondant à des frontières de l'image. Le moyen le plus direct est de prendre en compte des contraintes spatiales, telles que l'aspect visuel régulier de l'image ou le caractère « par morceaux » d'une photographie contenant différents objets, par exemple. Il est également possible de chercher à exclure les segmentations qui possèdent des contours trop irréguliers.

D'autres approches envisagent le problème de segmentation d'image (2.2) comme un problème de partitionnement de données, où on cherche à classer en k ensembles les $N_1 \times N_2$ pixels de l'image X . Cette interprétation permet alors de faire appel aux outils développés dans un cadre plus général pour la séparations de données en plusieurs groupes représentatifs. Pour cela, on peut utiliser les algorithmes de k -moyennes, introduits il y a quelques décennies mais toujours utilisés pour le partitionnement de gros volumes de données en apprentissage statistique [62, 46], mais également les outils d'optimisation sur graphes [27].

2.1.1 Minimisation de vraisemblance pénalisée

2.1.1.1 Motivation et liens avec les méthodes bayésiennes

Les travaux pionniers de Geman et Geman [45, 43, 44] ont posé les bases théoriques et pratiques du traitement d'image par méthode variationnelle et initié le traitement d'image bayésien. Le principe général est le suivant : on suppose disposer d'une observation $Y \in \mathbb{R}^{N_1 \times N_2}$ qui consiste en l'altération d'une image « pure » X que l'on souhaite retrouver, telle que

$$Y = \Phi(X) + B \quad (2.3)$$

où Φ modélise une déformation de l'image (agissant souvent linéairement sur X , comme par exemple un effet de flou) et B un bruit additif de moyenne nulle. Dans la suite, les lettres en gras désigneront les variables aléatoires, tandis que les lettres usuelles désigneront leurs réalisations respectives.

Connaissant Y , la méthode du Maximum *A Posteriori* (MAP) consiste à trouver \hat{X} maximisant la probabilité conditionnelle *a posteriori* $\mathbb{P}(X = X | Y = Y)$. Or, l'application de la loi des probabilités de Bayes induit

$$\mathbb{P}(X = X | Y = Y) \mathbb{P}(Y = Y) = \mathbb{P}(Y = Y | X = X) \mathbb{P}(X = X). \quad (2.4)$$

En postulant une distribution de probabilité gaussienne de variance σ^2 pour le bruit B , et en supposant que les variables aléatoires X et B sont indépendantes, on obtient l'expression *a priori* de la loi de probabilité de Y

$$\mathbb{P}(Y = Y | X = X) = \mathbb{P}(B = Y - \Phi(X)) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\|Y - \Phi(X)\|^2}{2\sigma^2}\right). \quad (2.5)$$

Si la variable aléatoire X admet une densité de probabilité proportionnelle à $\exp(-N(X))$, alors en injectant l'expression de la densité *a priori* (2.5) dans l'Équation (2.4), on obtient l'expression de la densité *a posteriori*

$$\mathbb{P}(X = X | Y = Y) = \frac{\exp\left(-\frac{\|Y - \Phi(X)\|^2}{2\sigma^2}\right) \exp(-N(X))}{\int \exp\left(-\frac{\|Y - \Phi(X)\|^2}{2\sigma^2}\right) \exp(-N(X)) dX} \propto \exp\left(-\frac{\|Y - \Phi(X)\|^2}{2\sigma^2} - N(X)\right). \quad (2.6)$$

La résolution du problème MAP consiste en la maximisation de (2.6) par rapport à X . En passant au logarithme, qui est une fonction croissante, on remarque que le problème MAP s'écrit

$$\hat{X} = \arg \max_{X \in \mathbb{R}^{N_1 \times N_2}} \mathbb{P}(X = X | Y = Y) = \arg \max_{X \in \mathbb{R}^{N_1 \times N_2}} \ln(\mathbb{P}(X = X | Y = Y)) = \arg \max_{X \in \mathbb{R}^{N_1 \times N_2}} -\frac{\|Y - \Phi(X)\|^2}{2\sigma^2} - N(X), \quad (2.7)$$

qui est équivalent au problème de *minimisation* suivant

$$\hat{X} = \operatorname{argmin}_{X \in \mathbb{R}^{N_1 \times N_2}} \frac{\|Y - \Phi(X)\|^2}{2\sigma^2} + N(X). \quad (2.8)$$

Ces dernières années, une grande variété de fonctionnelles ont été proposées pour la résolution du problème inverse (2.3) *via* une minimisation de fonctionnelle de la forme (2.8). La construction de différentes pénalisations $N(X)$ permet de refléter les connaissances dont on dispose sur l'image recherchée [95, 15]. Les liens entre minimisation de fonctionnelle et estimateurs bayésiens ont récemment fait l'objet de questionnements théoriques [52] et de théorèmes s'appuyant sur les outils développés en optimisation convexe [53].

Pour la segmentation d'image, on s'intéresse principalement à des contraintes permettant de modéliser un comportement "par morceaux", lisse par morceaux dans le cas des modèles de Blake-Zisserman [126] et Mumford-Shah [82], constant par morceaux dans le cas du modèle de Potts [93, 113] et de la Variation Totale [105], que nous présentons dans cette section.

2.1.1.2 Champs de Markov et densité de Gibbs

Afin de construire le terme $N(X)$, Geman et Geman [45] ont proposé de modéliser l'image X par un champ de Markov aléatoire et d'autre part d'interpréter (2.6) comme une densité de Gibbs, omniprésente en physique statistique. L'originalité de leur travail consiste à s'inspirer des énergies classiques de la physique statistique pour en tirer un panel de fonctionnelles aux propriétés connues, qui peuvent être généralisées et utilisées pour le traitement d'images. De plus, les méthodes de Metropolis-Hastings, développées initialement pour la physique, peuvent être mises à profit pour résoudre le problème inverse (2.3). Grâce à la dualité entre les champs de Markov et les distributions de Gibbs, Geman et Geman [45] obtiennent des garanties théoriques sur les résultats obtenus en appliquant les outils de physique statistique au traitement d'images.

Définition 2.1 (Champ aléatoire de Markov). [45, Section IV] Soit $\mathbf{Y} : \Omega \rightarrow \mathbb{V}$ un champ aléatoire défini sur un espace Ω , par exemple une grille de pixels $\{1, \dots, N_1\} \times \{1, \dots, N_2\}$ pour une image numérique, vers un espace des états \mathbb{V} , par exemple \mathbb{R} pour l'intensité lumineuse. (On suppose ici que cet espace est le même en tout point de Ω .) Le champ \mathbf{Y} est un champ de Markov aléatoire (*Markovian Random Field*, MRF) si et seulement s'il vérifie les conditions suivantes

- (i) Positivité : pour toute réalisation $Y : \Omega \rightarrow \mathbb{V}$, $\mathbb{P}(\mathbf{Y} = Y) > 0$.
- (ii) Markovianité : il existe un système de voisinages tel que la distribution de la variable aléatoire $\mathbf{Y}(\underline{n})$ ne dépende que des états du champ \mathbf{Y} aux points voisins de \underline{n} , i.e.,

$$\mathbb{P}(\mathbf{Y}(\underline{n}) = Y(\underline{n}) \mid \mathbf{Y}(\underline{n}') = Y(\underline{n}'), \underline{n}' \neq \underline{n}) = \mathbb{P}(\mathbf{Y}(\underline{n}) = Y(\underline{n}) \mid \mathbf{Y}(\underline{n}') = Y(\underline{n}'), \underline{n} \sim \underline{n}'), \quad (2.9)$$

où $\underline{n} \sim \underline{n}'$ signifie que \underline{n}' appartient au voisinage de \underline{n} .

- (iii) Homogénéité : la probabilité $\mathbb{P}(\mathbf{Y}(\underline{n}) = Y(\underline{n}) \mid \mathbf{Y}(\underline{n}') = Y(\underline{n}'), \underline{n} \sim \underline{n}')$ ne dépend que de la configuration du voisinage, i.e., le champ est invariant sous les translations spatiales conservant les voisinages.

Remarque. L'hypothèse (iii) de markovianité formulée à la Définition 2.1 traduit mathématiquement l'observation que des pixels proches au sein d'une image ont très souvent une intensité lumineuse similaire, excepté en un nombre très réduit de pixels situés à la frontière entre deux objets.

À la suite de [45], Geman et Geman se sont intéressés à de nombreux problèmes de traitement d'images en mettant à profit les propriétés des champs de Markov et leur équivalence avec les distributions de Gibbs. Pour la détection de contours [43], le modèle de champ de Markov pour

l'intensité lumineuse est alors augmenté grâce à l'introduction un champ représentant les *liaisons* entre les pixels, et permettant la modélisation des contours.

Dans le modèle de Geman et Geman [45], les pixels, notés \underline{n} forment les sommets d'un graphe régulier, la grille de points en gris foncé sur la Figure 2.1, tandis que les liaisons, notées par leurs extrémités $\langle \underline{n}, \underline{n}' \rangle$, sont les arêtes de ce graphe, en traits gris clairs sur la Figure 2.1. L'ensemble des intensités lumineuses en chaque pixel de l'image est représentée par un champ réel discret, $X(\underline{n}) \in \mathbb{R}$, tandis que les liaisons entre pixels sont codées sous la forme d'un champ binaire discret, $L(\langle \underline{n}, \underline{n}' \rangle) \in \{0, 1\}$, défini pour tout couple $\langle \underline{n}, \underline{n}' \rangle$ de pixels adjacents. Si \underline{n} et \underline{n}' appartiennent à une même région, alors $L(\langle \underline{n}, \underline{n}' \rangle) = 0$, tandis que $L(\langle \underline{n}, \underline{n}' \rangle) = 1$ signale qu'un contour passe entre les pixels \underline{n} et \underline{n}' . Les champs stochastiques d'intensité lumineuse, \mathbf{X} , et de liaisons, \mathbf{L} , sont considérés comme aléatoires et une *image* est donc une réalisation ω du couple de champs $\mathbf{S} = (\mathbf{X}, \mathbf{L})$.

La variable aléatoire S est supposée suivre une loi de probabilité de Gibbs, où la probabilité d'observer une réalisation ω , i.e., un couple (X_ω, L_ω) composé d'une carte de l'intensité lumineuse en chaque pixel et d'une carte des liaisons actives ou non, s'écrit

$$\pi_T(\omega) = \frac{1}{Z(T)} e^{-U(\omega)/T}, \quad \text{avec } Z(T) = \sum_{\omega} e^{-U(\omega)/T}. \quad (2.10)$$

Par analogie avec la physique statistique, $U(\omega)$ est une *énergie* caractérisant le modèle et T un paramètre de température, mesurant le caractère plus ou moins étalé de la distribution.

La forme de l'énergie $U(\omega)$ est fortement contrainte par l'hypothèse supplémentaire de « markovianité » donnée à la Définition 2.1, qui stipule que l'état d'un pixel \underline{n} , ou d'une liaison $\langle \underline{n}, \underline{n}' \rangle$, n'est influencé que par l'état de ses liaisons et pixels *voisins*. Le lien entre les distributions admissibles $\pi_T(\omega)$ et le caractère markovien a été explicité grâce au théorème de Hammersley-Clifford [6], formulé en terme de conditions nécessaires sur les probabilités conditionnelles. Pour traduire ces conditions en terme de forme d'énergie $U(\omega)$ admissible, Geman et Geman introduisent les « cliques » [45], définies comme les sous-ensembles du domaine Ω tels que deux points distincts d'une clique sont nécessairement voisins. L'hypothèse de Markovianité de la Définition 2.1 impose alors que les interactions intervenant dans l'énergie $U(\omega)$ couplent uniquement des points appartenant à une même clique. Par conséquent la forme de l'énergie est complètement dépendante du système de voisinages.

Le choix d'un tel système dépend de l'application considérée, la proposition formulée par Geman et Geman [43] est la celle de la Définition 2.2. En pratique, des voisinages de petites tailles permettent de limiter les interactions « à longue portée », et rendent ainsi possible l'échantillonnage de la distribution (2.10). La résolution du problème de détection de contours proposée par [43] reposant sur des échantillonneurs de Gibbs, cette hypothèse est cruciale. Notons que cette hypothèse est naturelle, au sens où deux points très éloignés au sein d'une image ont une grande probabilité d'appartenir à des objets ou des domaines distincts.

Définition 2.2 (Voisinages de pixels et de liaisons). [43, Figure 1] Un pixel de l'image, en bleu foncé sur la Figure 2.1a, contient dans son voisinage ses 8 pixels plus proches voisins, en bleu clair sur la Figure 2.1a, et les quatre liaisons dont ce pixel est une extrémité, en jaune sur la Figure 2.1a. Une liaison, en jaune foncé sur la Figure 2.1b, possède deux pixels voisins : ses extrémités, en bleu foncé sur la Figure 2.1b, et six liaisons voisines : celles issues de ses extrémités, en jaune pâle sur la Figure 2.1b.

L'énergie construite par Geman et Geman pour la détection de contours s'écrit

$$-U(\omega) = \lambda \sum_{\underline{n} \sim \underline{n}'} \phi(X(\underline{n}) - X(\underline{n}')) (1 - L(\langle \underline{n}, \underline{n}' \rangle)) + \mu \sum_{\langle \underline{n}, \underline{n}' \rangle} L(\langle \underline{n}, \underline{n}' \rangle) + \gamma \sum_D V_D(L) \quad (2.11)$$

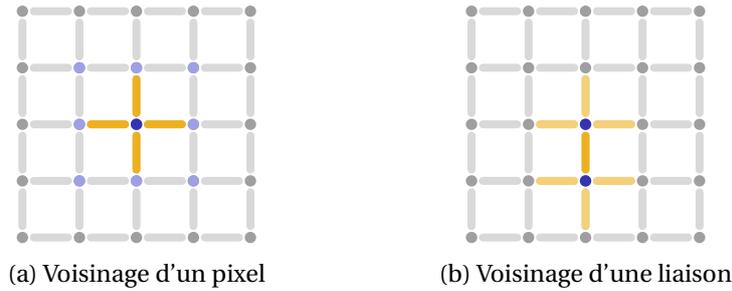


FIGURE 2.1 – Voisinages dans le modèle de Geman et Geman pour la détection de contours.

où $\underline{n} \sim \underline{n}'$ signifie que $\underline{n}, \underline{n}'$ sont deux pixels voisins au sens de la Définition 2.2, et Φ est une fonction paire décroissante. Le premier terme traduit la tendance de pixels voisins $\underline{n}, \underline{n}'$ appartenant à la même région, i.e., tels que $L(\langle \underline{n}, \underline{n}' \rangle) = 0$, à avoir une intensité lumineuse similaire. Il représente les régions lisses d'une image. À l'extrême, si $\lambda \rightarrow +\infty$, on modélise une région constante. Au contraire, lorsque l'intensité lumineuse entre deux pixels adjacents est trop grande, le premier terme force l'ajout d'un contour entre des ceux pixels, i.e., $L(\langle \underline{n}, \underline{n}' \rangle) = 1$.

La somme apparaissant dans le deuxième terme de l'énergie, courant sur l'ensemble des liaisons, permet de pénaliser le périmètre total des contours dans l'image.

Enfin, le troisième terme permet de sélectionner les configurations de contours les plus pertinentes. La somme court sur l'ensemble des quadruplets de liaisons adjacentes, illustré en jaune à la Figure 2.1b. Partant du principe que, dans une image réelle, la plupart des pixels sont à l'intérieur d'une région, loin des contours, et que les frontières forment des lignes continues, les potentiels V_D sont construits pour favoriser l'absence de contours et les contours en ligne droite (voir la remarque ci-dessous).

Remarque. Le modèle de champ de Markov pour les pixels et les liaisons de Geman et Geman, décrit ci-dessus, ne donne *a priori* pas forcément des contours fermés. Néanmoins, un choix judicieux de potentiels V_D permet de favoriser une certaine « cohérence » des contours en favorisant les géométries les plus pertinentes pour l'application considérée. Au rotations du plan près, il existe six configurations pour un quadruplet de liaisons adjacentes, composant une « clique », telle que définie dans [45]. Les liaisons actives, signalant la présence d'un contour sont représentées en rouge sur la Figure 2.2, tandis que les liaisons en gris sur la Figure 2.2 sont inactives, i.e., localisées à l'intérieur d'une région homogène. Ces configurations sont classées de la moins probable, une accumulation de contours formant une cellule fermée, à la plus probable, l'absence de contours. Afin d'intégrer cet *a priori* dans l'énergie introduite à l'Équation (2.11), un potentiel V_D *décroissant* est assigné aux configurations de la Figure 2.2, i.e., $\xi_1 \leq \xi_2 \leq \xi_3 \leq \xi_4 \leq \xi_5 \leq \xi_6$. Le paramètre γ à l'Équation (2.11) étant positif, la situation la moins probable est alors la plus coûteuse en énergie.

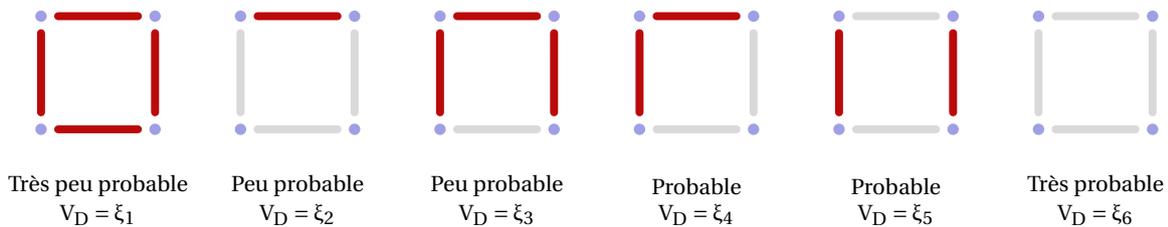


FIGURE 2.2 – Configurations D de quatre liaisons adjacentes, i.e., états possibles des « cliques » associées aux voisinages de la Définition 2.2. En rouge les liaisons actives forment un « contour », en gris les liaisons inactives sont à l'intérieur d'une région.

La distribution *a priori* du couple (X, L) , des intensités lumineuses et des contours, écrite à l'Équation (2.10) étant une distribution de Gibbs, dont l'énergie associée est donnée à l'Équation (2.11), sous l'hypothèse d'une observation Y corrompue par un bruit additif gaussien de va-

riance σ^2 , la distribution *a posteriori* est également une distribution de Gibbs

$$\pi_T^P(\omega) = \frac{1}{Z^P(T)} e^{-U^P(\omega)/T}, \quad \text{avec } Z^P(T) = \sum_{\omega} e^{-U^P(\omega)/T}. \quad (2.12)$$

dont l'énergie s'écrit

$$-U^P(\omega) = -\frac{1}{2\sigma^2} \|Y - \Phi(X)\|_2^2 + \lambda \sum_{\underline{n} \sim \underline{n}'} \phi(X(\underline{n}) - X(\underline{n}')) (1 - L(\langle \underline{n}, \underline{n}' \rangle)) + \mu \sum_{\langle \underline{n}, \underline{n}' \rangle} L(\langle \underline{n}, \underline{n}' \rangle) + \gamma \sum_D V_D(L) \quad (2.13)$$

L'obtention des contours de l'image à partir du modèle développé par Geman et Geman [43], consiste à minimiser l'énergie *a posteriori* U^P introduite à l'Équation (2.13). Pour cela, l'approche développée dans [45] s'appuie sur une analogie avec la physique et interprète la densité de Gibbs π_T de l'Équation (2.10) comme une distribution de Boltzmann à la température T . Dans cette interprétation, le minimum de l'énergie U est alors l'état fondamental, atteint à température nulle. En effet, au fur et à mesure que la température T décroît, l'expression de π_T à l'Équation (2.10) montre que la distribution se resserre autour de l'état d'énergie minimale.

Ce parallèle avec la physique permet de construire une méthode de résolution inspirée de la méthode de *recuit*. Afin de placer un système dans son état fondamental, la technique du recuit consiste à diminuer progressivement sa température jusqu'à atteindre une température suffisamment faible pour pouvoir considérer que la distribution de Boltzmann est très localisée (autour de l'état d'énergie minimale). Geman et Geman [45] proposent donc de construire une chaîne de Markov d'états $(\omega^{[m]})_{m \in \mathbb{N}}$, où $\omega^{[m]}$ est *asymptotiquement* un échantillon de la distribution $\pi_{T^{[m]}}$ à la température $T^{[m]}$. À condition que la température ne diminue pas trop violemment, la suite de variables aléatoires $(\omega^{[m]})_{m \in \mathbb{N}}$ converge vers l'état d'énergie minimale. La dynamique de recuit proposée est

$$T^{[m]} \geq \frac{C}{\log(1+m)} \quad (2.14)$$

avec C une constante dépendant de la forme de l'énergie U . De plus amples détails et des résultats théoriques sont donnés dans [45].

2.1.1.3 Modèle de Mumford-Shah

Le pendant continu, i.e., défini sur \mathbb{R}^2 , du formalisme discret, i.e., défini sur un graphe, de détection de contours de Geman et Geman a été proposé par Mumford et Shah en 1989, introduisant un modèle descriptif des images réelles [82], comme une collection de régions sur lesquelles l'image a des variations lentes, et aux frontières desquelles l'intensité lumineuse est fortement discontinue.

On considère une image $X : \Omega \rightarrow \mathbb{R}$ comme la discrétisation d'un champ bidimensionnel continu sous-jacent $f : Y \subset \mathbb{R}^2 \rightarrow \mathbb{R}$. Dans le modèle de Mumford-Shah, le champ f est approximé par un champ $u : Y \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ qui est lisse sur chaque domaine Y_1, \dots, Y_k et qui présente un ensemble de singularités \mathcal{S}_u coïncidant exactement avec les frontières des domaines $(Y_i)_{i=1}^k$.

Pour déterminer la meilleure approximation de l'image par un couple (u, \mathcal{S}_u) , Mumford et Shah ont proposé de minimiser la fonctionnelle qui porte aujourd'hui leur nom

$$\text{MS}(u, \mathcal{S}_u) = \lambda \int_Y (f - u)^2 d\underline{x} + \int_{Y \setminus \mathcal{S}_u} \|\nabla u\|^2 d\underline{x} + \nu \mathcal{H}^1(\mathcal{S}_u) \quad (2.15)$$

où ∇u désigne le gradient de u , \mathcal{H}^1 la mesure de Hausdorff unidimensionnelle et $\lambda, \nu > 0$ sont des paramètres de régularisation.

Le premier terme de la fonctionnelle de Mumford-Shah est un terme de fidélité aux données, qui mesure l'adéquation entre l'approximation u et le champ f . Le second terme impose des variations lentes à l'intérieur des régions Y_i . Enfin, le troisième terme pénalise la longueur des contours et favorise ainsi leur régularité. La minimisation de la fonctionnelle $MS(u, \mathcal{S}_u)$ réalise un compromis entre ces trois contraintes et fournit à la fois une régularisation lisse par morceaux de l'image et une estimation de l'ensemble de ces contours.

La fonctionnelle de Mumford-Shah faisant intervenir la mesure de Hausdorff, le problème d'optimisation associé s'avère délicat à résoudre. Afin de contourner cette difficulté, plusieurs variantes de la fonctionnelle $MS(u, \mathcal{S}_u)$ ont été proposées.

Parmi elles, on notera la fonctionnelle d'Ambrosio-Tortorelli [2], qui permet de formuler la minimisation de (2.15) comme la succession de problèmes elliptiques plus simples. Pour cela on construit une approximation régulière du terme de la fonctionnelle $MS(u, \mathcal{S}_u)$, dite « par champ de phase », qui décrit les contours par un champ \mathcal{C}^1 bidimensionnel z

$$AT_\varepsilon(u, z) = \lambda \int_Y (f - u)^2 d\underline{x} + \int_Y z^2 \|\nabla u\|^2 d\underline{x} + \nu \left(\int_Y \varepsilon \|\nabla z\|^2 d\underline{x} + \int_Y \frac{(1 - z)^2}{4\varepsilon} d\underline{x} \right), \quad (2.16)$$

où la fonction z est à valeurs dans $[0, 1]$. Le dernier terme de la fonctionnelle d'Ambrosio-Tortorelli (2.16) pousse z à valoir 1 tandis que le deuxième terme favorise l'annulation de z aux points de discontinuité. Les « contours » sont donc localisés aux endroits où la fonction z passe rapidement de 1 à 0. Enfin, combiné ensemble, le troisième et le quatrième termes pénalisent la « quantité de contours » en forçant le gradient de la fonction z à être d'autant plus localisé que ε est petit. En effet dans la limite où ε tend vers 0, on retrouve la mesure de Hausdorff des contours :

$$L_{z,\varepsilon} \underset{\varepsilon \rightarrow 0}{\simeq} \mathcal{H}^1(\mathcal{S}_u). \quad (2.17)$$

La convergence de $L_{z,\varepsilon}$ vers $\mathcal{H}^1(\mathcal{S}_u)$ a lieu au sens de la Γ -convergence introduite par De Giorgi [30] que nous ne détaillerons pas ici.

Un inconvénient majeur de ces méthodes pour une visée de segmentation, est la possibilité d'obtenir des contours non fermés et ainsi d'avoir une ambiguïté dans la définition des régions Ω_i . Une seconde difficulté est la complexité des algorithmes permettant de minimiser la fonctionnelle de Mumford-Shah (resp. son approximation par la fonctionnelle d'Ambrosio-Tortorelli), notamment à cause de son caractère non convexe. Ce problème fait toujours l'objet de développements récents [21, 40].

Des relaxations convexes de la fonctionnelle de Mumford-Shah ont également été proposées à la suite des travaux de Bouchitté [13, 1].

2.1.1.4 Contours actifs de Chan et Vese

Pour la segmentation à deux classes Chan et Vese ont construit un modèle de contours actifs purement variationnel, c'est-à-dire ne faisant pas usage d'un détecteur de contour [20]. Pour cela, les contours sont modélisés par la ligne de niveau 0 d'une fonction, ce qui permet une plus grande souplesse dans la gestion des contours que les méthodes par paramétrisation explicite.

Le modèle de Chan-Vese peut être dérivé à partir du modèle de Mumford-Shah constant par morceaux, qui correspond à (2.15) où on annule le terme $\|\nabla u\|^2$

$$\widetilde{MS}(u, \mathcal{S}_u) = \lambda \int_Y (f - u)^2 d\underline{x} + \nu \mathcal{H}^1(\mathcal{S}_u) \quad (2.18)$$

et où on impose que le champ $u : Y \rightarrow \mathbb{R}$ est constant par morceaux.

L'existence d'un minimum (u, \mathcal{S}_u) de la fonctionnelle $\widetilde{\text{MS}}$ a été démontrée dans [82]. De plus, dans ce cas, on peut montrer que \mathcal{S}_u est nécessairement l'union de courbes *fermées*, et on n'a donc aucune ambiguïté sur la définition des régions Y_i .

Le modèle de Chan-Vese s'intéresse plus particulièrement au cas de la segmentation en $k = 2$ régions. Une fois les contours fermés constituant \mathcal{S}_u orientés, le domaine Y se partage alors entre Y_1 , la région située à l'intérieur de \mathcal{S}_u et Y_2 la région située à l'extérieur de \mathcal{S}_u . Ainsi, en définissant les constantes u_1, u_2 telles que

$$u_{\underline{x}} = \begin{cases} u_1 & \text{si } \underline{x} \in Y_1 \\ u_2 & \text{si } \underline{x} \in Y_2. \end{cases} \quad (2.19)$$

on aboutit à la fonctionnelle de Chan-Vese

$$\text{CV}(u, \mathcal{S}_u) = \lambda_1 \int_{Y_1} (f - u_1)^2 d\underline{x} + \lambda_2 \int_{Y_2} (f - u_2)^2 d\underline{x} + \nu \mathcal{H}^1(\mathcal{S}_u) + \mathcal{H}^2(Y_1). \quad (2.20)$$

L'ajout du terme $\mathcal{H}^2(Y_1)$ pénalisant l'aire de la région Y_1 permet d'exclure le cas trivial où la région Y_1 remplit tout le domaine Y .

Suivant une idée proposée par Osher et Sethian en 1988 [88], les contours sont définis comme une ligne de niveau

$$\mathcal{S}_u = \{\underline{x} \in Y, \text{ tels que } \varphi(\underline{x}) = 0\}, \quad (2.21)$$

l'« intérieur » et l'« extérieur » de \mathcal{S}_u sont définis comme

$$Y_1 = \{\underline{x} \in Y, \text{ tels que } \varphi(\underline{x}) > 0\}, \quad Y_2 = \{\underline{x} \in Y, \text{ tels que } \varphi(\underline{x}) < 0\}. \quad (2.22)$$

La fonctionnelle (2.20) peut se réécrire comme une fonctionnelle des variables u_1, u_2 et φ . En effet, en définissant la fonction de Heaviside et sa dérivée au sens des distributions, la distribution de Dirac δ ,

$$H(t) \triangleq \begin{cases} 1 & \text{si } t \geq 0 \\ 0 & \text{si } t < 0 \end{cases} \quad \delta(t) \triangleq \frac{dH}{dt} \quad (2.23)$$

$H(\varphi(\underline{x}))$ est l'indicatrice de la région Y_1 .

L'aire de la région Y_1 qui intervient dans (2.20), se réécrit

$$\mathcal{H}^2(Y_1) = \int_Y H(\varphi(\underline{x})) d\underline{x}. \quad (2.24)$$

Afin de reformuler le terme de longueur $\mathcal{H}^1(\mathcal{S}_u)$ on remarque que la longueur du contour de la région Y_1 correspond à la variation totale de $H(\varphi)$, c'est à dire

$$\mathcal{H}^1(\mathcal{S}_u) = \int_Y \|\nabla H(\varphi(\underline{x}))\| d\underline{x}. \quad (2.25)$$

On peut alors réécrire le problème de minimisation de la fonctionnelle de Chan-Vese comme

$$\begin{aligned} \underset{u_1, u_2, \varphi}{\text{minimiser}} & \lambda_1 \int_Y (f(\underline{x}) - u_1)^2 H(\varphi(\underline{x})) d\underline{x} + \lambda_2 \int_Y (f(\underline{x}) - u_2)^2 (1 - H(\varphi(\underline{x}))) d\underline{x} \\ & + \int_Y H(\varphi(\underline{x})) d\underline{x} + \nu \int_Y \|\nabla H(\varphi(\underline{x}))\| d\underline{x}. \end{aligned} \quad (2.26)$$

Afin de résoudre le Problème (2.26) on peut réaliser une minimisation alternée sur u_1 , u_2 et φ . Pour cela, on choisit une régularisation de la fonction de Heaviside par une fonction différentiable, par exemple

$$H_\varepsilon(t) = \frac{1}{2} \left(1 + \frac{2}{\pi} \arctan \left(\frac{t}{\varepsilon} \right) \right) \quad (2.27)$$

dont la dérivée est une fonction de Dirac régularisée, dont l'expression pour notre exemple est

$$\delta_\varepsilon(t) = \frac{dH_\varepsilon}{dt} = \frac{\varepsilon}{\pi(\varepsilon^2 + t^2)}. \quad (2.28)$$

On procède ensuite en deux étapes :

1. En fixant φ la mise à jour de u_1 (resp. u_2) s'effectue par le calcul de la moyenne de f sur Y_1 (resp. Y_2) :

$$u_1 = \frac{\int_Y f(\underline{x}) H(\varphi(\underline{x})) d\underline{x}}{\int_Y H(\varphi(\underline{x})) d\underline{x}} \quad (2.29)$$

$$u_2 = \frac{\int_Y f(\underline{x}) (1 - H(\varphi(\underline{x}))) d\underline{x}}{\int_Y (1 - H(\varphi(\underline{x}))) d\underline{x}} \quad (2.30)$$

2. Les variables u_1 et u_2 étant fixées, la descente de gradient sur φ consiste à résoudre les Équations d'Euler-Lagrange associées à (2.26) :

$$\begin{cases} \frac{\partial \varphi}{\partial t} = \delta_\varepsilon(\varphi) \left[\text{vdiv} \left(\frac{\nabla \varphi}{|\nabla \varphi|} \right) - 1 - \lambda_1 (f - u_1) + \lambda_2 (f - u_2) \right] \\ \frac{\delta_\varepsilon(\varphi)}{|\nabla \varphi|} \frac{\partial \varphi}{\partial \underline{n}} = 0 \quad \text{sur} \quad \partial Y, \end{cases} \quad (2.31)$$

où ∂Y désigne le bord du domaine Y et $\partial \varphi / \partial \underline{n}$ le gradient de φ par rapport à la normale \underline{n} au bord ∂Y .

Un exemple de segmentation par la méthode de Chan-Vese est proposé à la Figure 2.3c.

2.1.1.5 « Convexification » du problème de partition minimale

Si elle permet de réaliser efficacement une segmentation à $k = 2$ classes, la méthode de Chan et Vese possède plusieurs inconvénients. D'une part, dans sa formulation, elle n'est pas directement généralisable à un nombre quelconque de classes. En effet, Chambolle *et coll.* [19] ont montré que le cas de la partition à deux classes bénéficiait de propriétés théoriques très fortes, qui en font un cas particulier qui peut être traité efficacement grâce à une approche de contours actifs. Ce n'est pas le cas de la partition à k classes. De plus, cette approche ne produit pas nécessairement un minimiseur global, mais uniquement un point stationnaire [19]. Enfin, elle ne permet pas de prendre en compte d'éventuelles énergies d'interface.

Ces limitations ont conduit à plusieurs stratégies visant à obtenir la meilleure relaxation possible du problème de Mumford-Shah constant par morceaux (2.18), i.e., présentant des garanties de bonne approximation du problème initial et suffisamment maniables en pratique. Pour cela, le Problème (2.18) est réécrit en faisant apparaître explicitement les régions inconnues Y_1, \dots, Y_k et les valeurs de l'image sur ces régions u_1, \dots, u_k

$$\underset{Y_1, \dots, Y_k, u_1, \dots, u_k}{\text{minimiser}} \quad \frac{1}{2} \sum_{i=1}^k \text{Per}(Y_i, Y) + \lambda \sum_{i=1}^k \int_{Y_i} (f - u_i)^2 d\underline{x}. \quad (2.32)$$

La quantité $(f(\underline{x}) - u_i(\underline{x}))^2$ correspond alors au coût d'assignation du pixel \underline{x} à la région Y_i , tandis que $\text{Per}(Y_i, Y)$ désigne le périmètre de la région Y_i au sein du domaine Y .

Définition 2.3 (Fonctions à variations bornées et variation totale). [3, Définition 3.4] Soit $\mathbf{v} = (v_1, \dots, v_k) \in L^1(\Upsilon, \mathbb{R}^k)$ une fonction intégrable définie sur $\Upsilon \subset \mathbb{R}^2$ et à valeurs dans \mathbb{R}^k , la *variation totale* de la fonction \mathbf{v} est définie comme le supremum suivant :

$$\int_{\Upsilon} |D\mathbf{v}| \triangleq \sup \left\{ - \int_{\Upsilon} \sum_{i=1}^k v_i \operatorname{div} \zeta_i : \zeta \in C_c^1(\Upsilon, \mathbb{R})^k, \|\zeta(\underline{x})\|^2 = \sum_{i=1}^k |\zeta_i(\underline{x})|^2 \leq 1, \forall \underline{x} \in \Upsilon \right\}, \quad (2.33)$$

où $C_c^1(\Upsilon, \mathbb{R})$ désigne l'ensemble des fonctions continument dérivables à support compact définies sur Υ , à valeurs réelles. Lorsque le supremum (2.33) est fini, on dit que la fonction \mathbf{v} est à *variations bornées*.

Proposition 2.1 (Variation totale d'une fonction indicatrice). [3, Théorème 3.36] En notant $v_i : \Upsilon \rightarrow \{0, 1\}$ l'indicatrice de l'ensemble Υ_i , le terme de périmètre peut être exprimé à partir de la variation totale de la fonction v_i via :

$$\operatorname{Per}(\Upsilon_i, \Upsilon) = \int_{\Upsilon} |Dv_i|. \quad (2.34)$$

Le fait que les ensembles $\Upsilon_1, \dots, \Upsilon_k$ forment une *partition* de Υ , se traduit par les conditions suivantes satisfaites par les fonctions v_1, \dots, v_k

$$\sum_{i=1}^k v_i(\underline{x}) = 1 \quad \text{p.p. } \underline{x} \in \Upsilon \quad (2.35)$$

$$\forall i \in \{1, \dots, k\}, \quad v_i(\underline{x}) \in \{0, 1\} \quad \text{p.p. } \underline{x} \in \Upsilon \quad (2.36)$$

où p.p. signifie « presque partout » au sens de la mesure de Lebesgue. Or les deux conditions ci-dessus ne définissent pas un ensemble *convexe*, ce qui peut conduire à des difficultés techniques pour montrer l'existence d'un minimum, mais également à des obstacles techniques pour calculer cet éventuel minimum. Une relaxation possible est alors d'assouplir de la seconde condition, et de chercher des solutions dans l'ensemble *convexe* des fonctions à *variations bornées* (voir Définition 2.3) vérifiant les conditions relaxées suivantes :

$$\sum_{i=1}^k v_i(\underline{x}) = 1 \quad \text{p.p. } \underline{x} \in \Upsilon \quad (2.37)$$

$$\forall i \in \{1, \dots, k\}, \quad v_i(\underline{x}) \geq 0 \quad \text{p.p. } \underline{x} \in \Upsilon. \quad (2.38)$$

L'ensemble de ces fonctions forme le *simplexe*

$$\Sigma \triangleq \left\{ \mathbf{v} : \Upsilon \rightarrow [0, 1]^k, \quad \forall i \in \{1, \dots, k\}, v_i(\underline{x}) \geq 0, \sum_{i=1}^k v_i(\underline{x}) = 1, \quad \text{p.p. } \underline{x} \in \Upsilon \right\}. \quad (2.39)$$

Le problème relaxé s'écrit alors

$$\underset{\mathbf{v} \in \Sigma, (u_1, \dots, u_k) \in \mathbb{R}^k}{\text{minimiser}} \quad \frac{1}{2} \sum_{i=1}^k \int_{\Upsilon} |Dv_i| + \lambda \sum_{i=1}^k \int_{\Upsilon} v_i(\underline{x}) (f(\underline{x}) - u_i)^2 d\underline{x}, \quad (2.40)$$

sous la condition que u soit la fonction constante par morceaux valant u_i sur Υ_i pour tout $i \in \{1, \dots, k\}$.

L'intérêt de la formulation continue ci-dessus est qu'elle permet de démontrer des propriétés d'existence, et le cas échéant d'unicité, de la solution grâce à des théorèmes classiques d'analyse fonctionnelle. Néanmoins, en pratique, on observe uniquement une image, i.e., une discrétisation du champ f et le Problème (2.40) doit être discrétisé. Le lecteur intéressé se reportera à [19, 94] par exemple pour la résolution du problème de Chan-Vese à k niveaux. La section suivante présentera la discrétisation de la variation totale continue (2.33) que nous utiliserons dans la suite de cette thèse.

2.1.1.6 Variation Totale et seuillage itéré *a posteriori* (ROF_X-S)

Dans le cas discret, une image segmentée U peut être représentée comme une carte où une intensité appartenant à un ensemble *fini* de valeurs $\{F_1, \dots, F_k\}$ est attribuée à chaque pixel. Sur chaque région Ω_i , l'image segmentée est constante et vaut F_i . Par conséquent, U est une image *constante par morceaux*. En utilisant l'opérateur de gradient discret \mathbf{D} de la Définition 2.4, cela se traduit par le fait que le gradient $\mathbf{D}U$ d'intensité lumineuse de U est non-nul en très peu de pixels. Le gradient $\mathbf{D}U$ est alors qualifié de *parcimonieux*.

Définition 2.4 (Gradient discret). [19, Section 4.1] Pour une image $U \in \mathbb{R}^{N_1 \times N_2}$, l'opérateur de gradient d'intensité lumineuse discret $\mathbf{D} : \mathbb{R}^{N_1 \times N_2} \rightarrow \mathbb{R}^{2 \times N_1 \times N_2}$ est défini à partir des accroissements horizontaux et verticaux suivants :

$$(\mathbf{D}U)(\underline{n}) = \begin{pmatrix} U(\underline{n} + \underline{e}_1) - U(\underline{n}) \\ U(\underline{n} + \underline{e}_2) - U(\underline{n}) \end{pmatrix} = \begin{pmatrix} (\mathbf{V}U)(\underline{n}) \\ (\mathbf{H}U)(\underline{n}) \end{pmatrix} \quad (2.41)$$

où $(\underline{e}_1, \underline{e}_2)$ désigne la base canonique de \mathcal{R}^2 . Les différences finies verticales, $\mathbf{V}U$, et horizontales, $\mathbf{H}U$, sont calculées en chaque pixel \underline{n} . Lorsque $\underline{n} = (N_1, \cdot)$ (resp. $\underline{n} = (\cdot, N_2)$) se trouve sur un bord de l'image, $(\mathbf{V}U)(\underline{n})$ (resp. $(\mathbf{H}U)(\underline{n})$) vaut zéro. De la sorte $\mathbf{D}U \in \mathbb{R}^{2 \times N_1 \times N_2}$ consiste en la concaténation de la carte des différences verticales, $\mathbf{V}U \in \mathbb{R}^{N_1 \times N_2}$, et de la carte des différences horizontales $\mathbf{H}U \in \mathbb{R}^{N_1 \times N_2}$, de même résolution que l'image U .

Ainsi, il est possible de mettre à profit les outils favorisant la *parcimonie*, introduit initialement pour les problèmes d'acquisition comprimée (*compressed sensing*), en segmentation d'images. On mesure la parcimonie d'un champ discret $Z \in \mathbb{R}^{N_1 \times N_2}$ en comptant le nombre de pixels où il est non nul, ce qui définit la pseudo-norme ℓ_0

$$\|Z\|_0 = \text{card}(\{\underline{n} \in \Omega \text{ tel que } Z(\underline{n}) \neq 0\}). \quad (2.42)$$

Cette pseudo-norme, appliquée au gradient de l'image, $\|\mathbf{D}U\|_0$ mesure son degré de constance par morceaux, plus $\|\mathbf{D}U\|_0$ est faible plus l'image U est « simple », avec peu de régions différentes. L'utilisation de la pseudo-norme ℓ_0 reste néanmoins limitée. En effet, elle n'est pas convexe et ne possède aucune propriété de régularité, ce qui exclut de nombreuses méthodes de minimisation de fonctionnelle classique.

Afin de proposer une relaxation convexe de la pseudo-norme ℓ_0 , Rudin, Osher et Fatemi (ROF) se sont appuyés sur la formulation continue du problème de Mumford-Shah constant par morceaux (2.18) et ont proposé d'utiliser la norme L_1 du gradient spatial comme pénalisation [105], favorisant ainsi sa parcimonie, et donc le caractère constant par morceaux du champ régularisé u . Cette démarche aboutit à la formulation *continue* suivante :

$$\underset{u: Y \rightarrow \mathbb{R}}{\text{minimiser}} \lambda \int_Y (f - u)^2 + \int_Y \|\nabla u\| \, d\underline{x}. \quad (2.43)$$

Remarque. La fonctionnelle (2.43) peut également être interprétée comme une relaxation convexe de la fonctionnelle de Chan-Vese (2.20).

La solution du Problème (2.43) est un champ d'autant plus régularisée que λ^{-1} est grand, mais pas, à proprement parler, une segmentation. Cai et Steidl [16] ont construit une méthode de seuillage itératif, présentée à l'Algorithme 1 qui permet, à partir de la solution du Problème (2.43) de retrouver la segmentation de Chan-Vese. La convergence du seuillage itératif de l'Algorithme 1 vers la solution du problème de Chan-Vese est démontrée [16, Théorème 1] en s'appuyant sur les travaux de Chambolle, Caselles, Novaga *et coll.* [17, 18], donnant ainsi une assise théorique solide à l'utilisation de la fonctionnelle (2.43) pour la segmentation d'images [16, Proposition 3].

Afin de traiter des images, la fonctionnelle (2.43) est discrétisée. Pour cela, la pénalisation de la norme du gradient est réécrite à partir du gradient discret de l'image $\mathbf{D}U$ et de la norme mixte $\ell_{2,1}$.

Algorithme 1 Algorithme de seuillage itératif

Entrée : $u : \Upsilon \rightarrow \mathbb{R}$ solution du Problème (2.43)

Paramètres : k : nombre de régions recherchées

 M_{\max} : nombre d'itérations

Initialisation : Choisir les seuils $T_1^{[0]} < \dots < T_{k-1}^{[0]}$
 $T_0^{[0]} = -\infty$ et $T_k^{[0]} = +\infty$ (par convention)

for $m = 0$ **to** $M_{\max} - 1$ **do**
for $i = 1$ **to** k **do**
Mise à jour des régions

$$\Upsilon_i^{[m]} = \left\{ \underline{x} \in \Upsilon, \text{ tels que } T_{i-1}^{[m]} \leq u(\underline{x}) < T_i^{[m]} \right\}$$

Mise à jour des moyennes

$$F_i^{[m]} = \langle f \rangle_{\Upsilon_i^{[m]}}$$

Mise à jour des seuils

$$T_i^{[m+1]} = \frac{1}{2}(F_i^{[m]} + F_{i+1}^{[m]}) \quad \text{si } i \neq k$$

end for

$$T_0^{[m+1]} = -\infty \text{ et } T_k^{[m+1]} = +\infty \text{ (par convention)}$$

end for

{Régions et moyennes finales}

for $i = 1$ **to** k **do**

$$\Upsilon_i \leftarrow \Upsilon_i^{[T_{\max}]}$$

$$F_i \leftarrow F_i^{[T_{\max}]}$$

end for
Sortie : Segmentation $\Upsilon = \cup_{i=1}^k \Upsilon_i$,

 Moyennes F_1, \dots, F_k

Définition 2.5 (Norme mixte $\ell_{2,1}$). La norme mixte $\|\cdot\|_{2,1}$, est définie pour une image multi-composantes $Z = [Z_1; \dots; Z_I]^T \in \mathbb{R}^{I \times N_1 \times N_2}$ comme

$$\|Z\|_{2,1} = \sum_{\underline{n} \in \Omega} \sqrt{\sum_{i=1}^I (Z_i(\underline{n}))^2}. \quad (2.44)$$

La combinaison de l'opérateur de gradient \mathbf{D} de la Définition 2.4, et de la norme mixte $\ell_{2,1}$ de la Définition 2.5, définit la *Variation Totale isotrope*, qui permet de favoriser la constance par morceaux de l'image régularisée.

Définition 2.6 (Variation Totale isotrope). [19, Section 4.1] La *Variation Totale isotrope* d'une image $U \in \mathbb{R}^{N_1 \times N_2}$ suivante :

$$\text{TV}(U) \triangleq \|\mathbf{D}U\|_{2,1} = \sum_{\underline{n} \in \Omega} \sqrt{(\mathbf{V}U)(\underline{n})^2 + (\mathbf{H}U)(\underline{n})^2}. \quad (2.45)$$

Finalement, la solution du Problème (2.43) *discrétisé*, notée \hat{X}^{ROF} , est définie par :

$$\hat{X}^{\text{ROF}} = \underset{U \in \mathbb{R}^{N_1 \times N_2}}{\text{argmin}} \frac{1}{2} \|X - U\|_{\text{Fro}}^2 + \lambda \text{TV}(U). \quad (2.46)$$

Elle est le résultat d'un compromis entre la fidélité à l'image X , et le caractère constant par morceaux, requis pour la segmentation. On peut ensuite appliquer directement l'Algorithme 1 à \hat{X}^{ROF} , en remplaçant la variable continue \underline{x} par la variable discrète \underline{n} . La succession du débruitage par Variation Totale (2.45) et du seuillage itéré de l'Algorithme 1 sera notée dans la suite ROF $_X$ -S. Un exemple de segmentation par la méthode de Variation Totale seuillée, ROF $_X$ -S, est présenté en Figure 2.3d.

La fonctionnelle de l'Équation (2.46) étant convexe, mais non différentiable, sa minimisation pratique requiert l'utilisation d'outils d'analyse convexe non lisse qui seront décrits en détails à la Section 4.3, menant à l'implémentation d'algorithmes *proximaux*.

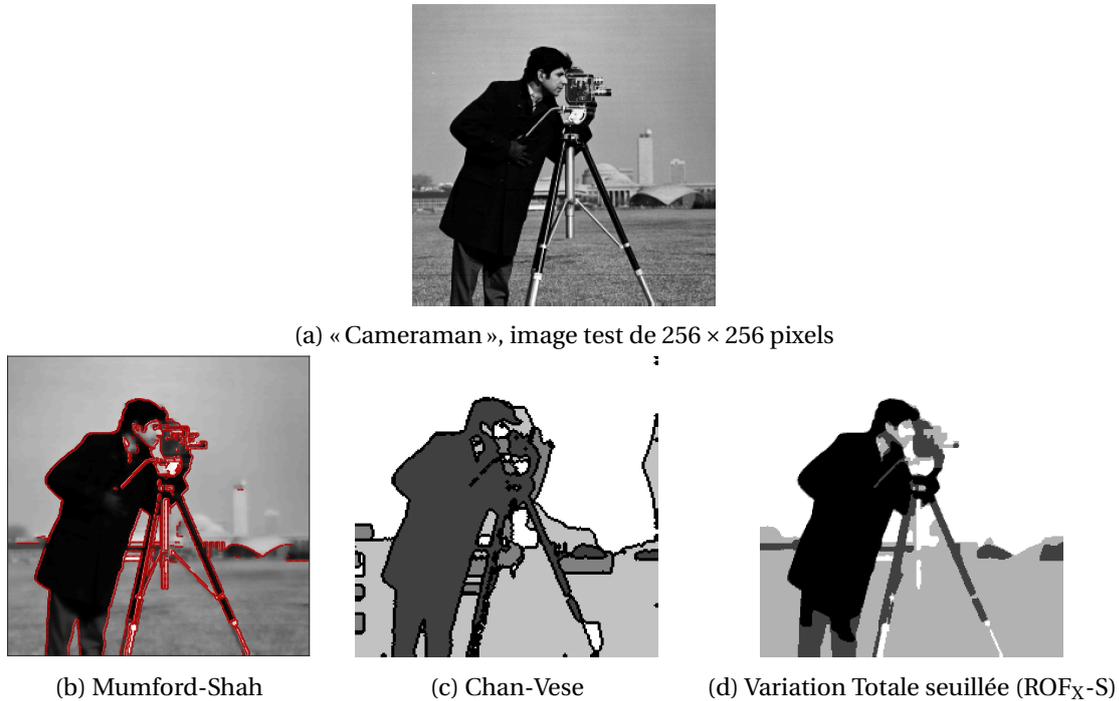


FIGURE 2.3 – Comparaison des méthodes variationnelles pour la segmentation d'images. (a) Image de test : « cameraman » de MATLAB. (b) Méthode de Mumford-Shah discrète, les contours obtenus sont superposés à l'image régularisée. (c) Méthode Chan-Vese étendue à un nombre de région $k > 2$, les contours sont signalés en noir, tandis que les régions sont représentées par un nombre $k = 4$ de niveaux de gris. (d) ROF_X-S : Débruitage par Variation Totale, suivi d'une étape de post-traitement par seuillage itératif à $k = 4$ régions.

2.1.1.7 Réglage des hyperparamètres des fonctionnelles régularisantes

Les différentes méthodes par minimisation de vraisemblance pénalisée sont pilotées par un ou plusieurs hyperparamètres, qui permettent de contrôler le compromis entre la fidélité aux données, l'image réelle, éventuellement bruitée, et les contraintes régularisantes qui permettent de réaliser une segmentation, accompagnée d'un possible débruitage. Le choix de ces paramètres est déterminant pour la qualité des estimées régularisées, et donc des segmentations. La difficulté à sélectionner de manière optimale leurs hyperparamètres constitue une limitation majeure des approches par vraisemblance pénalisée pour le traitement d'images « réelles » [115].

Un premier obstacle est l'absence de critère pour évaluer précisément la qualité des estimées. En effet, en pratique, on ne dispose pas de la vérité terrain et un calcul *exact* de l'erreur d'estimation est impossible. Pour palier ce problème, plusieurs approches ont été proposées, telle que les méthodes de validation croisée [48], s'appuyant sur une formule explicite liant l'image observée à son estimée régularisée, les méthodes de « courbe-L », comparant la norme de l'image régularisée avec la norme du résidu, ou encore les approches issues des travaux de Stein [111]. Sous l'hypothèse que l'image observée est corrompue par un bruit gaussien i.i.d., a été construit un estimateur sans biais de l'erreur quadratique, appelé SURE (*Stein Unbiased Risk Estimate*) [111]. L'hypothèse d'un bruit décorréolé est néanmoins très restrictive et s'applique rarement au bruit d'estimation d'attributs de textures, souvent calculés en prenant en compte plusieurs pixels de l'image analysée et des extensions ont été proposées, comme par exemple pour les lois exponentielles [37]. La difficulté est alors déplacée vers le calcul effectif de ces estimateurs de l'erreur, et notamment du

terme de *degrés de liberté* qu'ils font intervenir [66, 116, 33, 118].

Une seconde difficulté consiste à produire une stratégie efficace d'optimisation de l'indicateur de qualité retenu. Lorsque le nombre d'hyperparamètres est faible, i.e., inférieur ou égal à deux, une recherche exhaustive sur une grille est envisageable [32]. Néanmoins, son coût de calcul, augmentant algébriquement avec le nombre d'hyperparamètres, cette méthode s'avère peu adaptée en pratique pour traiter de gros volumes de données. Des procédures automatiques explorant efficacement l'espace des hyperparamètres [98, 23] sont alors nécessaires, particulièrement pour l'analyse de gros volumes de données.

Récemment, sous l'hypothèse d'un bruit gaussien i.i.d., un estimateur du gradient de l'erreur, SUGAR (*Stein Unbiased GrAdient Risk estimate*) a été proposé [31], permettant de réaliser un réglage automatique des hyperparamètres grâce à un algorithme de quasi-Newton. Lorsque le bruit d'observation, gaussien, présente des corrélations, il est nécessaire de construire des estimateurs SURE et SUGAR généralisés, prenant en compte une matrice de covariance *non triviale*. Plus encore que dans le cas gaussien i.i.d., où des estimateurs robuste de la variance ont été proposés à partir des coefficients d'ondelettes [78, Formule (11.85)], la question de l'estimation correcte de la structure de covariance est centrale. En effet, la qualité de ces estimateurs, et donc la précision du réglage automatique des hyperparamètres, dépend de la précision avec laquelle est connue la matrice de covariance des observations. Ainsi, pour le traitement de données « réelles », le défi consiste alors également à proposer une sélection des hyperparamètres pilotée par les données, i.e., où toutes les quantités sont calculées à partir des observations *uniquement*.

2.1.2 Méthodes de partitionnement de données (*clustering*)

2.1.2.1 *k*-moyennes

L'algorithme des *k*-moyennes consiste à chercher *k* centroïdes représentatifs des *k* régions Ω_i recherchées [112, 77]. Les pixels sont alors réunis dans la classe correspondant au centroïde dont ils sont le plus proche, ce qui se traduit par

$$\Omega_i = \{\underline{n} \in \Omega, \text{ tels que } \forall i' \neq i, \quad \|X(\underline{n}) - C_{i'}\| \geq \|X(\underline{n}) - C_i\|\}, \quad (2.47)$$

où $\underline{n} = (n_1, n_2)$ désigne un pixel de l'image X et $X(\underline{n}) \in \mathbb{R}^d$ est la valeur du niveau de gris ($d = 1$) ou la couleur ($d = 3$) de l'image au pixel \underline{n} .

Dans le cas d'images en niveaux de gris, un centroïde correspond à la valeur médiane de niveau de gris sur tous les pixels de la région associée, dans le cas d'une image couleur, représentée par ses trois canaux RVB, le centroïde est un vecteur de trois réels représentant la couleur moyenne de tous les pixels de la région.

Le problème des *k*-moyennes se formule mathématiquement comme la minimisation de l'étalement des points d'une région autour du centroïde et s'écrit

$$\underset{C_1, \dots, C_k \in \mathbb{R}^d}{\text{minimiser}} \sum_{i=1}^k \sum_{\underline{n} \in \Omega_i} \|X(\underline{n}) - C_i\|^2. \quad (2.48)$$

Remarquons que la $i^{\text{ème}}$ somme dans l'Équation (2.48), portant sur l'ensemble des pixels de la région Ω_i , correspond, à un facteur de normalisation près, à la variance de niveau de gris ($d = 1$) ou de couleur ($d = 3$) au sein de la $i^{\text{ème}}$ région,

$$\sum_{\underline{n} \in \Omega_i} \|X(\underline{n}) - C_i\|^2 = \text{card}(\Omega_i) \mathbb{V}(\Omega_i), \quad (2.49)$$

où $\text{card}(\Omega_i)$ désigne le nombre de pixels contenus dans la région Ω_i et \mathbb{V} la variance. Ainsi l'algorithme des k -moyennes minimise effectivement la dispersion au sein de chaque région Ω_i .

Le Problème (2.48) est NP-difficile. Seuls des algorithmes heuristiques ont donc été proposés pour le résoudre de manière itérative. L'algorithme de Lloyd-Forgy [74, 41], retranscrit à l'Algorithme 2, effectue des itérations en deux étapes : une étape d'assignation construisant les régions $\Omega_i^{[t]}$ à partir des centroïdes $C_i^{[t]}$, puis une étape de mise à jour des centroïdes. L'algorithme de Lloyd-Forgy s'arrête au bout de T_{\max} itérations, lorsque

$$\forall i \in \{1, \dots, k\}, \quad \Omega_i^{[T_{\max}-1]} = \Omega_i^{[T_{\max}]}. \quad (2.50)$$

Algorithme 2 Algorithme de Lloyd pour la résolution du problème des k -moyennes (2.48).

Entrée : X Image à segmenter

Paramètres : k nombre de régions

Initialisation : Choisir au hasard k centroïdes *distincts* $C_1^{[0]}, \dots, C_k^{[0]}$

for $t = 0$ **to** $T_{\max} - 1$ **do**

{Assignation}

for $i = 1$ **to** k **do**

$$\Omega_i^{[t]} \leftarrow \left\{ \underline{n} \in \Omega, \text{ tels que } \forall i' \neq i, \quad \|X(\underline{n}) - C_i^{[t]}\| \geq \|X(\underline{n}) - C_{i'}^{[t]}\| \right\}$$

end for

{Mise à jour des centroïdes}

for $i = 1$ **to** k **do**

$$C_i^{[t+1]} = \frac{1}{\text{card}(\Omega_i^{[t]})} \sum_{\underline{n} \in \Omega_i^{[t]}} X(\underline{n})$$

end for

end for

{Assignation et centroïdes finaux}

for $i = 1$ **to** k **do**

$$\Omega_i \leftarrow \Omega_i^{[T_{\max}]}$$

$$C_i \leftarrow C_i^{[T_{\max}]}$$

end for

Sortie : Segmentation $\Omega = \cup_{i=1}^k \Omega_i$

Centroïdes C_1, \dots, C_k

Une illustration de segmentation obtenue par l'algorithme des k -moyennes est proposée en Figure 2.4 pour $k = 3$ régions. On remarque que les régions de niveau de gris constant sont correctement segmentées, par exemple le costume du cameraman, tandis que les régions inhomogènes, comme la pelouse, sont mal segmentées.

Remarque (Seuillage adaptatif). Dans le cas des images en niveau de gris $d = 1$, l'algorithme des k -moyennes peut s'interpréter comme un algorithme de seuillage adaptatif. Les k centroïdes $(C_i)_{i=1}^k$ sont alors remplacés par $k - 1$ seuils

$$\begin{cases} T_0 = -\infty, \\ T_i = C_{i+1} - C_i, \quad i = 1, \dots, k-1, \\ T_k = +\infty \end{cases} \quad (2.51)$$

et les régions sont définies à partir de ces seuils par

$$\Omega_i = \{ \underline{n} \in \Omega, \text{ tels que } T_{i-1} \leq X(\underline{n}) < T_i \}. \quad (2.52)$$

2.1.2.2 c -moyennes relaxées (*Fuzzy c-means*)

Dans l'algorithme des k -moyennes décrit ci-dessus, chaque pixel \underline{n} est associé à une unique région Ω_i . Cette condition, très forte, peut-être assouplie. L'algorithme des c -moyennes *relaxé* [36,

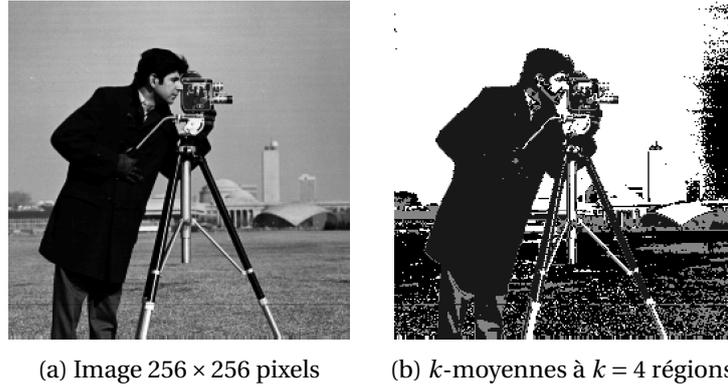


FIGURE 2.4 – Algorithmes de segmentation sur l'image « cameraman » de MATLAB. (a) Image originale. (b) k -moyennes avec $k = 4$.

10] associe à chaque pixel \underline{n} un vecteur $p(\underline{n}) \in \mathbb{R}^c$ mesurant le degré d'appartenance de ce pixels \underline{n} à chacune des régions Ω_i qui vérifie

$$\forall i \in \{1, \dots, c\}, \forall \underline{n} \in \Omega, \quad 0 \leq p_i(\underline{n}) \leq 1, \quad \text{et} \quad \sum_{i=1}^c p_i(\underline{n}) = 1. \quad (2.53)$$

La fonction de coût à minimiser s'écrit alors comme une somme pondérée par les probabilités $p_i(\underline{n})$ que le pixel \underline{n} appartienne à la $i^{\text{ème}}$ région

$$\text{minimiser}_{C_1, \dots, C_c \in \mathbb{R}^d} \sum_{i=1}^c \sum_{\underline{n} \in \Omega} p_i^m(\underline{n}) \|X(\underline{n}) - C_i\|^2, \quad (2.54)$$

où $1 \leq m < +\infty$ est un réel fixé arbitrairement. Le centroïde C_i est alors une moyenne pondérée

$$C_i = \frac{\sum_{\underline{n} \in \Omega} p_i^m(\underline{n}) X(\underline{n})}{\sum_{\underline{n} \in \Omega} p_i^m(\underline{n})}. \quad (2.55)$$

Le principe de l'Algorithme 2 s'adapte facilement au problème relaxé en remplaçant l'étape d'assignation des régions Ω_i par une étape de mise à jour des degrés d'appartenance *via* la relation

$$(p_i(\underline{n}))^{-1} = \sum_{j=1}^c \left(\frac{\|X(\underline{n}) - C_i\|}{\|X(\underline{n}) - C_j\|} \right)^{\frac{2}{m-1}}. \quad (2.56)$$

Une segmentation est déduite en associant à chaque pixel \underline{n} la région i maximisant $p(\underline{n})$:

$$\underline{n} \in \Omega_i \iff i = \arg \max_{1 \leq i' \leq c} p_{i'}(\underline{n}). \quad (2.57)$$

A posteriori l'utilisateur peut raffiner cette segmentation, par exemple pour satisfaire des contraintes spatiales, en choisissant de classer le pixel \underline{n} dans la région de probabilité d'appartenance suivante dans l'ordre des probabilités $p_i(\underline{n})$.

2.1.2.3 Ligne de partage des eaux

Une seconde classe de méthodes de segmentation d'image consiste à résoudre dans un premier temps un problème de détection de contours, pour ensuite chercher une partition de l'image coïncidant avec ces contours. Étant entendu que la plupart des méthodes de détection de contours ne fournissent pas *a priori* des contours fermés, on n'a pas équivalence stricte entre les problèmes de détection de contours et les méthodes de segmentation d'image. Fournir, à partir d'un ensemble de contours de l'image X , une partition de l'image compatible peut donc s'avérer être un problème mathématique à part entière. Nous allons présenter ici la méthode dite de « ligne de

partage des eaux», dans le cas d'une détection de contours à partir du gradient de l'intensité lumineuse.

Issue de la morphologie mathématique [7], la méthode de ligne de partage des eaux s'inspire de la géologie pour fournir une segmentation de l'image X [8]. À partir d'une image $X \in \mathbb{R}^{N_1 \times N_2}$ (par exemple Figure 2.5a), en niveau de gris, on construit la carte d'intensité de gradient de X définie par

$$\nabla X(\underline{n}) = \sqrt{(X(\underline{n} + \underline{e}_1) - X(\underline{n}))^2 + (X(\underline{n} + \underline{e}_2) - X(\underline{n}))^2}, \quad (2.58)$$

où $\underline{e}_1 = (1, 0)$ et $\underline{e}_2 = (0, 1)$. Un exemple de carte d'intensité de gradient est donné en Figure 2.5b. La Figure 2.5c présente une binarisation de la carte de gradient ∇X de la Figure 2.5b : en blanc sont représentés les pixels où le gradient est supérieur ou égal à 75% de la valeur maximale de la carte de gradient.

Cette carte d'intensité de gradient ∇X est interprétée comme un relevé topographique, où l'intensité $\nabla X(\underline{n})$ est assimilée à l'altitude au pixel \underline{n} [9]. Ainsi, les lignes de maxima locaux sur la carte de gradient sont vues comme des « lignes de crête », séparant différents « bassins versants », qui sont les régions Ω_i recherchées.

Plusieurs algorithmes ont été proposés pour la résolution de ce problème [108, 120, 119]. Ces algorithmes sont construits sur le principe d'une « inondation » progressive du relief topographique, à partir de laquelle les lignes de crêtes sont détectées automatiquement et identifiées comme des contours en même temps que les bassins versant sont localisés.

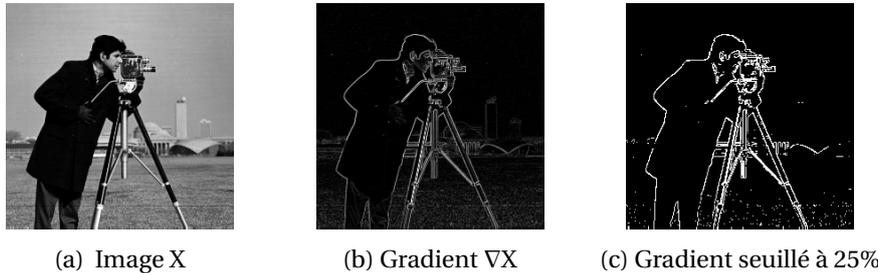


FIGURE 2.5 – Détection des gradients pour la méthode de ligne de partage des eaux.

2.1.2.4 Optimisation sur graphe

Plusieurs problèmes de traitement d'image, tels que le débruitage ou la segmentation, peuvent être exprimés comme la minimisation d'une énergie définie à partir d'un *graphe régulier*, ce qui permet ensuite de tirer profit des méthodes d'optimisation développées dans le cadre de la théorie des graphes.

Définition 2.7 (Graphe). [27, Section 2.1] Un graphe G est défini à partir d'un ensemble de sommets $V = \{v_i, i = 1, \dots, n\}$, reliés entre eux par des arêtes formant un ensemble $E \subseteq V \times V$ à m éléments, notés $\{e_{i,j} \triangleq (v_i, v_j), 1 \leq i, j \leq n\}$, où $e_{i,j}$ désigne une arête reliant les sommets v_i et v_j . Les arêtes d'un graphe peuvent être étiquetées par une fonction de poids $w : E \rightarrow \mathbb{R}$, où on note $w_{i,j} \triangleq w(e_{i,j})$, qui permet de traduire l'affinité entre les sommets v_i et v_j .

Exemple 2.1 (Fonction de poids). Dans le cas de la segmentation d'image, v_i correspond à un pixel et, en désignant par X_i l'intensité lumineuse au pixel v_i , un premier exemple de fonction de poids consiste à mesurer l'écart d'intensité lumineuse entre deux pixels via

$$w_{i,j} = \exp(-\beta(\nabla X)^2), \quad \text{avec} \quad \nabla X = X_i - X_j, \quad (2.59)$$

où $\beta > 0$ est un paramètre contrôlant la décroissance de l'affinité entre les pixels v_i et v_j lorsque $\forall X$ augmente.

Les auteurs de [27] proposent de formuler le problème de segmentation à deux classes sous la forme d'une minimisation sur graphe :

$$\underset{X}{\text{minimiser}} \quad \lambda \sum_{e_{i,j} \in E} w_{i,j}^p |X_i - X_j|^q + \sum_{v_i \in V} w_{F_i}^p |X_i|^q + \sum_{v_i \in V} w_{B_i}^p |X_i - 1|^q, \quad \text{avec } X_F = 0 \text{ et } X_B = 1, \quad (2.60)$$

où F et B sont deux sommets auxiliaires symbolisant chacun une « source », à laquelle sont reliés préférentiellement les pixels de la première (resp. deuxième) région, associées aux valeurs de X proches de 0 (resp. proches de 1). Une segmentation, i.e., une image binaire, S est alors obtenue par un simple seuillage

$$S_i = \begin{cases} 1 & \text{si } x_i \geq \frac{1}{2} \\ 0 & \text{si } x_i < \frac{1}{2}. \end{cases} \quad (2.61)$$

Lorsque $q = 1$ et $p \rightarrow \infty$, le Problème (2.60) se ramène à la méthode de segmentation par ligne de partage des eaux, décrite à la Section 2.1.2.3, formulée sur un graphe. La Formulation (2.60) permet une extension naturelle au cas $q \geq 1$, avec toujours $p \rightarrow \infty$, et est alors appelée ligne de partage des eaux à la puissance (*power watershed*) [27].

Remarque. La segmentation à deux classes grâce à (2.60) et (2.61), peut être vue comme l'estimation d'une image binaire. On peut alors interpréter le Problème (2.60) comme un cas particulier de la minimisation d'une énergie définie sur un graphe proposée par [51], i.e.,

$$\underset{X}{\text{minimiser}} \quad \underbrace{\lambda \sum_{e_{i,j} \in E} w_{i,j}^p |X_i - X_j|^q}_{\text{régularisation}} + \underbrace{\sum_{v_i \in V} w_i^p |X_i - Y_i|^q}_{\text{fidélité aux données}}, \quad (2.62)$$

où Y joue le rôle d'une « observation », les poids binaires $w_{i,j}$ imposent plus ou moins fortement un accord entre les valeurs de X aux sommets v_i et v_j , et les poids unaires $(w_i)_{i=1}^n$ sont des poids favorisent la fidélité à « l'observation » Y.

La Formulation (2.62), où terme d'attache aux données et terme de régularisation sont clairement identifiés, permet de souligner la ressemblance entre les Étapes (2.60) et (2.61), et la méthode ROF_X-S, décrite en Section 2.1.1.6, chacune étant composée d'une régularisation suivie d'un seuillage.

2.2 Textures

Un premier niveau de description d'une image réelle consiste à la modéliser par un assemblage de régions lisses de couleur ou de niveau de gris distincts, ce qui justifie les approches de segmentation présentées à la Section 2.1. La transition entre deux objets est alors décrite par une forte discontinuité de l'intensité lumineuse ou de la couleur, comme dans le modèle proposé par Mumford et Shah [82]. Or, la grande majorité des images réelles ne peuvent pas être décrites comme un ensemble de régions homogènes mais au contraire présentent un aspect *texturé*. En effet, les *textures* apparaissent naturellement dans les structures organiques et biologiques. Elles sont également la signature visuelle d'un matériau : on reconnaît aisément du bois, de la fourrure, de la pierre malgré les variétés de couleurs possibles. Ainsi, l'information contenue dans la texture des différentes régions d'une image est cruciale dans son interprétation humaine [59] et doit donc être prise en compte pour effectuer des tâches de reconnaissance d'image par ordinateur.

Nous proposons, dans un premier temps, de préciser ce que recouvre le concept de « texture » en traitement d'images en nous appuyant sur quelques exemples illustratifs, qui permettront de

souligner l'importance des textures pour le traitement numérique d'images réelles. La nécessité de fournir une formulation et un sens mathématique aux images « texturées » pour réaliser les tâches usuelles de classification, segmentation, . . . , a conduit à deux stratégies, différentes dans leur approche, mais se nourrissant néanmoins l'une de l'autre. Une première voie consiste à proposer un modèle mathématique, déterministe ou stochastique, permettant, au moyen de paramètres caractéristiques, de synthétiser des textures numériques. Le modèle construit peut alors être utilisé comme grille de lecture et d'analyse des textures réelles [28]. Cette stratégie est plébiscitée lorsque suffisamment d'information est disponible *a priori* sur les textures à étudier. Quelques modèles de textures courants sont présentés à la Section 2.2.2. Au contraire, pour l'analyse de textures *a priori* inconnues, ou dont la variabilité est très grande, des attributs sont proposés et combinés entre eux afin d'extraire les propriétés pertinentes. Cette démarche est illustrée à la Section 2.2.3.

2.2.1 Intuition et brève catégorisation

Le Larousse¹ propose la définition suivante d'une texture :

« Représentation graphique d'une matière, d'une surface, dont le rendu en volume est effectué par placage sur un modèle en trois dimensions. (L'effet de matière [bois, pierre, etc.] est permis grâce à des procédés de numérisation d'images photographiées ou dessinées.) »

L'accent de cette définition porte sur le lien entre la structure du matériau représenté et l'aspect texturé, ce qui est en accord avec le fait que, si la reconnaissance d'objets se fait surtout à partir de leur forme géométrique, les textures contiennent un second niveau d'information indiquant de quelle matière ces objets sont constitués. La grande diversité des matériaux donne donc lieu à une grande variété de textures au sein de laquelle nous pouvons schématiquement distinguer plusieurs catégories.

D'un côté, les textures périodiques, caractérisées par la répétition régulière d'un motif donné, se rencontrent à la fois dans la nature, voir par exemple les nervures du bois en Figure 2.6a, mais également dans les objets manufacturés, comme par exemple les mailles d'un tricot en Figure 2.6b. Pour ces textures, la connaissance du(des) motif(s), ainsi que de leur périodicité, permet de les décrire et de les reproduire entièrement, *via* par exemple le spectrogramme de l'image [70] ou en étudiant la distorsion locale [100]. L'étude de l'organisation des villes *via* des images satellites fournit également un riche champ d'application, utilisant par exemple les matrices de co-occurrence [127] pour identifier le motif répété. Un modèle de texture périodique est présenté à la Section 2.2.2.1.

D'un autre côté, les textures fractales, sont composées de la répétition d'un motif à toutes les échelles, comme c'est le cas pour les fractales, illustré en Figure 2.6c, représentant un chou romanesco, caractérisées par leur dimension fractale [22, 49]. La propriété principale de ces textures réside dans leur invariance par changement d'échelle, aussi appelée auto-similarité. Nous étudierons au Chapitre 3 une grande classe de textures fractales : les champs gaussiens auto-similaires.

Parmi les textures périodiques ou fractales, une autre distinction importante réside dans le caractère déterministe ou stochastique des motifs et de leur agencement. Les constructions humaines, telles que la pièce de tricot de la Figure 2.6b, sont souvent bien représentées par des motifs donnés, fixes, organisés de manière régulière dans l'espace, i.e., des textures déterministes. À l'inverse, les textures stochastiques, caractérisées par leur caractère aléatoire et définies par leurs propriétés statistiques [81], permettent de reproduire de nombreux motifs naturels. Plusieurs exemples provenant de l'histologie, un tissu musculaire en Figure 2.6d, ou de la nature, une fourrure en Figure 2.6e, illustrent non seulement la grande variété des textures aléatoires mais aussi leur représentativité. Associant l'auto-similarité des fractales et le caractère aléatoire, les matériaux naturels, comme les roches illustrées en Figure 2.6f, ont été décrits comme des textures browniennes fractionnaires en premier par Mandelbrot [79]. Ce modèle a été largement repris

1. <https://www.larousse.fr>

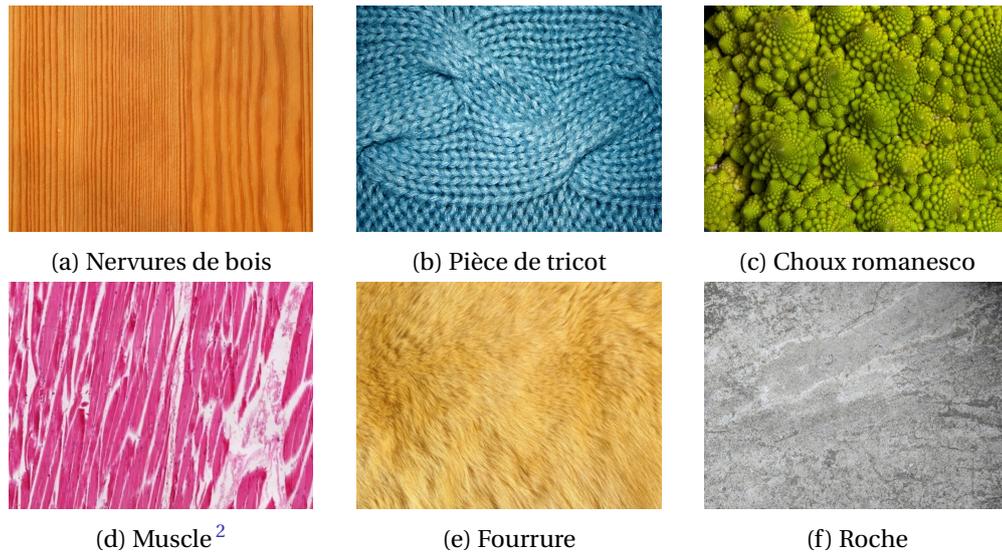


FIGURE 2.6 – Exemples de textures.

dans l'étude des fractures [86].

Afin d'effectuer de la classification d'images texturées, puis à terme de la segmentation, plusieurs attributs ont été proposés. La qualité d'un attribut est liée, d'une part, à sa capacité à discriminer efficacement des textures différentes et d'autre part à la possibilité de l'estimer de manière fiable, et notamment d'obtenir une estimation locale à partir de laquelle construire une segmentation aux contours précis.

2.2.2 Modèle de texture

2.2.2.1 Modulation en amplitude et en fréquence

Tout comme un signal périodique se décompose en sa série de Fourier, il est naturel de considérer les textures périodiques comme la somme la superposition de fonctions périodiques élémentaires. Par extension, la modulation en amplitude et en fréquence donne le modèle de texture multi-composantes, dit « AM-FM » [57], où l'image $X \in \mathbb{R}^{N_1 \times N_2}$ est décrite comme la somme de termes modulés en amplitude et en fréquence, s'écrivant au pixel \underline{n} :

$$X(\underline{n}) = \sum_{r=1}^R a_r(\underline{n}) \cos(2\pi\Phi_r(\underline{n})), \quad (2.63)$$

avec R composantes, dont $a_r : \mathbb{R}^2 \rightarrow \mathbb{R}$ et $\nabla\Phi_r : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ sont respectivement l'amplitude et la fréquence locale au pixel \underline{n} et sont supposées varier lentement, c'est-à-dire de manière continue et lisse. Un exemple de texture AM-FM à trois composantes est proposé à la Figure 2.7.

La démodulation de l'image X consiste alors à inverser la relation (2.63), et retrouver ainsi les amplitudes locales a_r et les fréquences locales $\nabla\Phi_r$. L'accès aux amplitudes a_r et aux fréquences $\nabla\Phi_r$ locales est possible si l'on suppose que chaque composante de l'image est à bande de fréquence étroite, ces bandes ne se recouvrant pas. Dans ce cas, l'utilisation de démodulateurs régularisés [68] permet alors de proposer une procédure de segmentation de texture utilisant sur une décomposition multi-composantes « AM-FM ».

2. <http://www.histologyguide.com/>

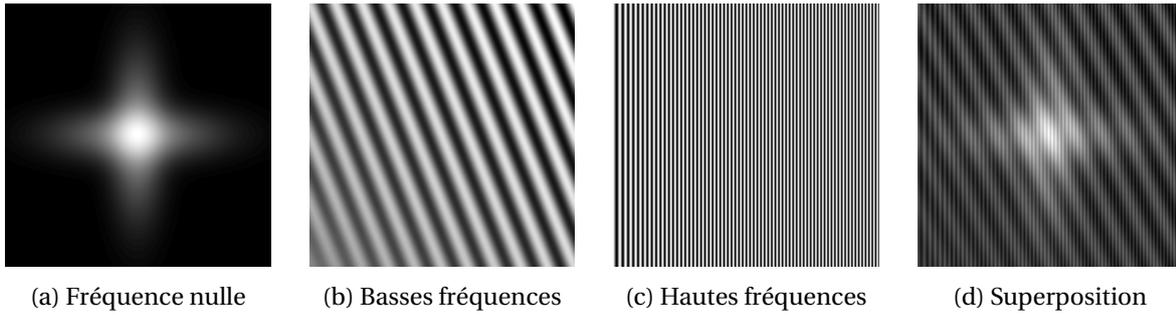


FIGURE 2.7 – Texture AM-FM à trois composantes ($R = 3$). (a) Amplitude $a_1 = \sqrt{2\pi\sigma^2}^{-1} \exp\left(-10\sigma^{-2}\left(n_1 - \frac{N_1}{2}\right)^2 - \sigma^{-2}\left(n_2 - \frac{N_2}{2}\right)^2\right) + \sqrt{2\pi\sigma^2}^{-1} \exp\left(-\sigma^{-2}\left(n_1 - \frac{N_1}{2}\right)^2 - 10\sigma^{-2}\left(n_2 - \frac{N_2}{2}\right)^2\right)$ et phase $\Phi_1 \equiv 0$. (b) Amplitude $a_2 = \sqrt{|n_1| + |n_2|}$ et phase $\Phi_2 = 10\frac{n_1}{N_1} + 4\frac{n_2}{N_2}$. (c) Amplitude $a_3 = (N_1N_2)^{1/4}$ et phase $\Phi_3 = 10\ln(n_1)\frac{n_1}{N_1}$.

2.2.2.2 Champ aléatoire markovien

Le modèle de champ de Markov aléatoire, introduit à la Définition 2.1, a également été utilisé pour la modélisation et la génération de textures [28]. Les auteurs de [28] proposent un modèle de champ stochastique dans lequel l'intensité lumineuse en chaque pixel suit une variable binomiale dont le paramètre dépend des paramètres des lois binomiales des pixels voisins.

Le champ d'intensité lumineuse X est défini sur une grille de pixels $\Omega = \{1, \dots, N_1\} \times \{1, \dots, N_2\}$ de sorte que, en chaque pixel \underline{n} , la variable aléatoire $X(\underline{n})$ suit une loi binomiale à $G - 1$ tirages de paramètre $\theta(\underline{n})$

$$X(\underline{n}) \sim B(G - 1, \theta(\underline{n})) \quad (2.64)$$

où G désigne le nombre de niveau de gris avec lequel est représentée l'image, et dont le paramètre $\theta(\underline{n})$ dépend des intensités lumineuses $X(\underline{n}')$ des pixels $\underline{n}' \sim \underline{n}$, voisins de \underline{n} .

Pour modéliser les liens entre les paramètres des lois binomiales de pixels voisins, le paramètre $\theta(\underline{n})$ est mis sous la forme

$$\theta(\underline{n}) = \frac{\exp(T(\underline{n}))}{1 + \exp(T(\underline{n}))}. \quad (2.65)$$

Les lois binomiales en différents pixels sont alors liées grâce à la structure du champ $T(\underline{n})$

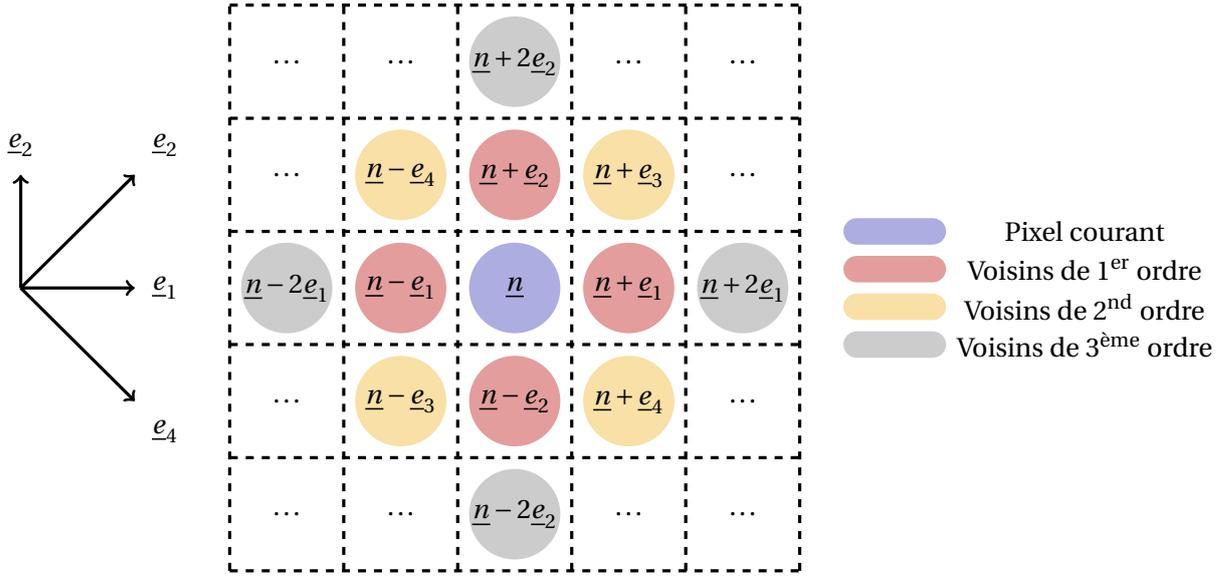
$$T(\underline{n}) = a + \underbrace{b(1, 1) (X(\underline{n} + \underline{e}_1) + X(\underline{n} - \underline{e}_1)) + b(1, 2) (X(\underline{n} + \underline{e}_2) + X(\underline{n} - \underline{e}_2))}_{1^{\text{er}} \text{ ordre}} \quad (2.66)$$

$$+ \underbrace{b(2, 1) (X(\underline{n} + \underline{e}_3) + X(\underline{n} - \underline{e}_3)) + b(2, 2) (X(\underline{n} + \underline{e}_4) + X(\underline{n} - \underline{e}_4))}_{2^{\text{nd}} \text{ ordre}} \quad (2.67)$$

$$+ \underbrace{b(2, 1) (X(\underline{n} + 2\underline{e}_1) + X(\underline{n} - 2\underline{e}_1)) + b(2, 2) (X(\underline{n} + 2\underline{e}_2) + X(\underline{n} - 2\underline{e}_2))}_{3^{\text{ème}} \text{ ordre}} \quad (2.68)$$

$$+ \dots, \quad (2.69)$$

encodée sous la forme de coefficients $b(i, p)$ où i désigne l'ordre et p encode la « direction ». Lorsque, $b(i, 1) = b(i, 2)$ à tout ordre i , le champ résultant est isotrope. Tandis que des coefficients $b(i, p)$ positifs favorisent des intensités lumineuses similaires entre pixels voisins, les coefficients $b(i, p)$ négatifs entraînent un caractère répulsif et génèrent donc des variations brutales d'intensité lumineuse. Le coefficient a consiste en un terme de base, qui influence de manière homogène l'ensemble des pixels. Une texture X est alors complètement caractérisée par la valeur des coefficients a et $(b(i, p))_{i,p}$.


 FIGURE 2.8 – Voisinage du pixel \underline{n} pour la génération de textures par champ aléatoire markovien.

Pour générer un échantillon d'une texture aléatoire de coefficients donnés, les méthodes de type Metropolis-Hastings, par acceptation/rejet sont plébiscitées. Elles permettent de générer une suite de champs aléatoires $(X^{[t]})_{t \in \mathbb{N}}$ qui converge en loi vers un champ de Markov dont l'intensité lumineuse en chaque pixel est une loi binomiale de paramètre $\vartheta(\underline{n})$ de la forme (2.65), et dont le caractère markovien est contrôlé par la Relation (2.66).

2.2.2.3 Mélange gaussien

Un modèle *stochastique* très répandu pour la modélisation d'une texture est le modèle du champ aléatoire gaussien, complètement caractérisé par sa moyenne et sa matrice de covariance. Dans ce cadre, une texture stochastique $X \in \mathbb{R}^{N_1 \times N_2}$ est décrite comme un champ gaussien de moyenne nulle, de matrice de covariance \mathbf{R}

$$X \sim \mathcal{N}(\mathbf{0}, \mathbf{R}). \quad (2.70)$$

Sa densité de probabilité *a priori* s'écrit donc

$$f(X|\mathbf{R}) = \frac{1}{\sqrt{(2\pi)^N \det(\mathbf{R})}} \exp\left(-\frac{X^T \mathbf{R}^{-1} X}{2}\right), \quad (2.71)$$

en utilisant l'isomorphisme canonique $\mathbb{R}^{N_1 \times N_2} \cong \mathbb{R}^N$, avec $N = N_1 N_2$ pour effectuer le calcul de la norme pondérée.

Lorsque le champ est *stationnaire*, la matrice \mathbf{R} est de Toeplitz. Grâce à une approximation de Whittle, la matrice de covariance \mathbf{R} peut donc être approchée par une matrice circulante $\tilde{\mathbf{R}}$, qui est diagonalisable dans le domaine de Fourier. En notant \mathbf{F} la matrice orthogonale de la transformée de Fourier discrète 2D vectorisée, il existe donc une matrice diagonale $\mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_N)$ telle que

$$\tilde{\mathbf{R}}^{-1} = \mathbf{F}^* \mathbf{\Lambda} \mathbf{F}. \quad (2.72)$$

La densité de probabilité $f(X|\tilde{\mathbf{R}})$ est alors séparable dans le domaine de Fourier

$$f(X|\tilde{\mathbf{R}}) = \prod_{i=1}^N \frac{1}{\sqrt{(2\pi)\lambda_i}} \exp\left(-\frac{\lambda_i |\tilde{x}_i|^2}{2}\right), \quad \tilde{X} \triangleq \mathbf{F}X \quad (2.73)$$

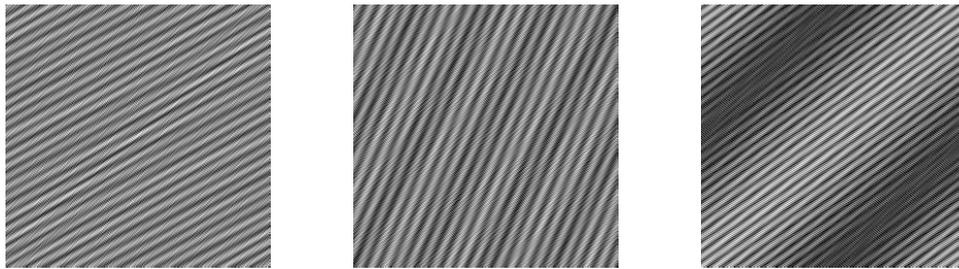
où \tilde{X} désigne la transformée de Fourier discrète de l'image X . En revenant à des cartes de $\mathbb{R}^{N_1 \times N_2}$, (2.73) signifie que la transformée de Fourier discrète de X à la fréquence spatiale $\underline{v} = (v_1, v_2)$, notée $\tilde{X}(v_1, v_2)$, est une variable aléatoire gaussienne de variance $\lambda^{-1}(v_1, v_2)$. La Figure 2.9 présente différentes textures obtenues avec des variances spectrales de la forme [117]

$$\lambda^{-1}(v_1, v_2) = \rho_1 \rho_2 \left(1 + \frac{(v_1 \cos(\vartheta) - v_2 \sin(\vartheta) - \bar{v}_1)^2}{\rho_1^2} + \frac{(v_1 \sin(\vartheta) + v_2 \cos(\vartheta) - \bar{v}_2)^2}{\rho_2^2} \right), \quad (2.74)$$

où (\bar{v}_1, \bar{v}_2) est une fréquence centrale, fixée à $(0,05; 0,01)$. Le rendu visuel des textures obtenues est alors contrôlé par la forme des variances spectrales $\lambda^{-1}(v_1, v_2)$, qui est paramétrée ici par l'angle ϑ , les amplitudes ρ_1, ρ_2 et la fréquence centrale (\bar{v}_1, \bar{v}_2) .

Remarque. En traitement d'images par méthodes bayésiennes, ce type de modélisation est d'usage courant. Les textures à étudier sont alors pilotées par leur matrice de covariance, pour laquelle un modèle paramétrique est fixé en amont (voir par exemple [117]).

Parmi les champs gaussiens, la classe des champs gaussiens auto similaires, *non stationnaires*, est également très riche et a été largement étudiée [106, 26, 92]. Nous introduirons notamment au Chapitre 3 le modèle du champ brownien fractionnaire. Néanmoins, la non-stationnarité rend caduque l'approximation de Whittle et la synthèse de tels champs requiert des outils supplémentaires, que nous présenterons à la Section 3.2.



(a) $\vartheta = \pi/4, \rho_1 = 0,02, \rho_2 = 0,01$. (b) $\vartheta = \pi/4, \rho_1 = 0,05, \rho_2 = 0,01$. (c) $\vartheta = \pi/6, \rho_1 = 0,05, \rho_2 = 0,01$.

FIGURE 2.9 – Textures gaussiennes stationnaires de densité de probabilité (2.71) et de variances spectrales (2.74).

2.2.3 Attributs de texture

2.2.3.1 Réponse à des filtres de Gabor

Suivant l'idée de représenter une texture comme la superposition de motifs élémentaires, les banques de filtres ont été largement utilisés dans la caractérisation des textures [35]. Dans ce modèle, les attributs discriminants d'une texture correspondent aux réponses locales à chaque filtre de la banque. Les filtres de Gabor ont notamment montré une très bonne adéquation avec les modalités de la vision humaine [80], ce qui explique leur popularité pour la classification et la segmentation de textures [121, 25, 72, 125].

Dans le domaine spatial, un filtre de Gabor est le produit d'une sinusoïde complexe et d'une fenêtre gaussienne

$$g(\underline{x}; \underline{v}, \underline{\mu}) = \exp(2i\pi \underline{v} \underline{x} + \phi_0) \exp\left(-\frac{(\underline{x} - \underline{\mu})^2}{2\underline{\sigma}^2}\right), \quad \text{avec} \quad \frac{(\underline{x} - \underline{\mu})^2}{2\underline{\sigma}^2} \triangleq \frac{(x_1 - \mu_1)^2}{2\sigma_1^2} + \frac{(x_2 - \mu_2)^2}{2\sigma_2^2} \quad (2.75)$$

où \underline{v} désigne la fréquence bidimensionnelle de la sinusoïde, ϕ_0 une phase, $\underline{\mu}$ le centre de la gaussienne et $\underline{\sigma}^2 \triangleq (\sigma_1^2, \sigma_2^2)$ la variance de la gaussienne suivant les axes des abscisses et des ordonnées.

Quelques exemples de filtres de Gabor ayant différentes fréquences $|\underline{\nu}|$, orientations ϑ et variances $\underline{\sigma}^2$ sont proposés en Figure 2.10.

Ce type de filtre permet d'étudier la présence de la fréquence bidimensionnelle $\underline{\nu} \triangleq |\underline{\nu}|e^{i\vartheta}$, orientée selon l'angle ϑ , dans un voisinage de $\underline{\mu}$ et ainsi de décrire une texture par la répartition spatiale de son contenu fréquentiel. Pour cela la transformée de Gabor de l'image est obtenue en effectuant le produit scalaire de l'image $f : \Upsilon \rightarrow \mathbb{R}$ avec une famille de filtres de Gabor,

$$G(\underline{\mu}, \underline{\nu}) \triangleq \int_{\Upsilon} f(\underline{x}) \overline{g(\underline{x}; \underline{\mu}, \underline{\nu})} d\underline{x} \quad (2.76)$$

et constitue un cas particulier de transformée de Fourier à court-terme [107].

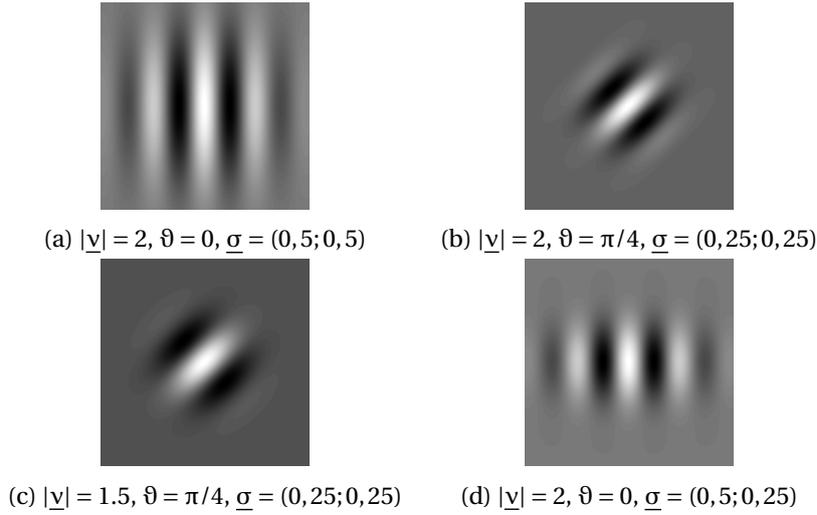


FIGURE 2.10 – Exemple de filtres de Gabor définis par l'Équation (2.75) pour différentes fréquences $|\underline{\nu}|$, orientation ϑ et variances $\underline{\sigma}^2$.

2.2.3.2 Attributs fractals et transformée multi-échelle

Nous allons principalement nous intéresser dans cette thèse à des attributs *fractals*. Le premier, que nous considérerons comme le principal attribut discriminant, est l'*exposant de Hölder local*, qui permet de mesurer la *régularité locale* [61].

Définition 2.8 (Exposant de Hölder). [61, Définition 1] Soit un champ scalaire $f : \mathbb{R}^2 \rightarrow \mathbb{R}$. L'*exposant de Hölder* du champ f au point \underline{x} , notée $h(\underline{x})$, correspond au plus grand réel $\alpha > 0$ tel qu'il existe un polynôme $\mathcal{P}_{\underline{x}}$, de degré strictement inférieur à α , et $\kappa > 0$ vérifiant

$$\forall \underline{x}' \in \mathbb{R}^2, \quad |f(\underline{x}') - \mathcal{P}_{\underline{x}}(\underline{x}')| \leq \kappa \|\underline{x}' - \underline{x}\|^\alpha. \quad (2.77)$$

Remarque. Dans cette thèse on s'intéressera à des champs dont l'exposant de Hölder local est compris entre 0 et 1, par conséquent le polynôme $\mathcal{P}_{\underline{x}}$ se réduit à une constante $f(\underline{x})$ et (2.77) se simplifie [61] en

$$\forall \underline{x}' \in \mathbb{R}^2, \quad |f(\underline{x}') - f(\underline{x})| \leq \kappa \|\underline{x}' - \underline{x}\|^\alpha. \quad (2.78)$$

Lorsque l'exposant de Hölder local $h(\underline{x})$ du champ f au point \underline{x} est légèrement inférieur à 1, le champ f se rapproche d'un comportement localement Lipschitzien au voisinage de \underline{x} . À l'inverse, lorsque $h(\underline{x})$ est faible, le champ f présente un comportement très irrégulier au point \underline{x} . Pour illustrer ces tendances, la Figure 2.11 présente deux exemples de champs de régularité locale uniforme $h(\underline{x}) \equiv H$. Le comportement du champ de la Figure 2.11a est caractérisé en tout point par un exposant de Hölder $H = 0,9$ « grand », tandis que le champ de la Figure 2.11b possède en

tout point un exposant de $H = 0,3$, « petit ». La courbe bleue tracée en Figure 2.11c (resp. 2.11d) est une coupe transverse du champ de la Figure 2.11a (resp. 2.11b). Pour un point \underline{x} , représenté par le point rouge, la fonction $\underline{x}' \mapsto \|\underline{x}' - \underline{x}\|^H$, qui représente la meilleure approximation locale du champ f au point \underline{x} par une fonction puissance, est tracée en rouge.

De manière générale, lorsque $0 < h(\underline{x}) < 1$, le champ f est continu mais non dérivable au point \underline{x} et la non-différentiabilité de f en \underline{x} est localement « équivalente » à la non-différentiabilité en 0 d'une fonction puissance d'exposant $h(\underline{x})$.

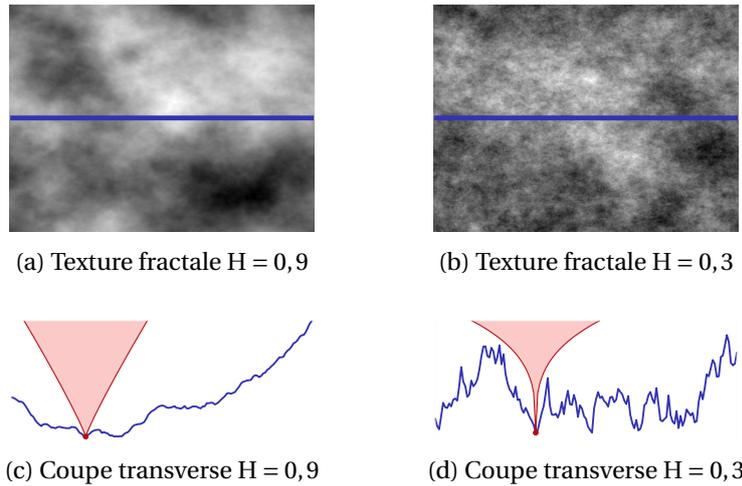


FIGURE 2.11 – Exposant de Hölder local d'une texture fractale. Deux textures fractales de régularité locale uniforme $h(\underline{x} \equiv H$ avec $H = 0,9$ (grand) et $H = 0,3$ (petit), accompagnées d'une coupe transverse de l'intensité lumineuse.

La Définition 2.8, théorique, étant formulée localement, elle ne peut être utilisée directement pour estimer la régularité locale en pratique. Néanmoins, en observant la Formule (2.78), on remarque que toute dilatation $g(\underline{x}) \triangleq f(a\underline{x})$, d'un facteur $a > 0$ de la fonction f possède la même régularité locale $h(\underline{x})$ que f en tout point \underline{x} . Cette remarque amène à se tourner vers l'analyse multi-échelle pour l'estimation de la régularité locale.

Pour cela, on définit dans un premier temps les coefficients d'ondelettes *non décimés* du champ f [78]. En effet, cette transformée *non décimée* permettra d'extraire les attributs fractals *en chaque point*.

Définition 2.9 (Transformée en ondelettes *non décimée*). [78, Chapitre 7] On se donne φ la fonction d'échelle et ψ l'ondelette mère, caractérisée par son nombre de moments nuls N_ψ telle que

$$\forall q = 0, \dots, N_\psi - 1, \quad \int |t|^q \psi(t) dt = 0, \quad \text{et} \quad \int |t|^{N_\psi} \psi(t) dt \neq 0 \quad (2.79)$$

définissant une analyse multi-échelle unidimensionnelle [78, Section 7.1]. Dans cette thèse nous utiliserons des ondelettes de Daubechies [29] non symétrisées, construites pour avoir le maximum de moments nuls N_ψ et une taille de support $2N_\psi - 1$ donnée. Les ondelettes mères bidimensionnelles sont définies pour une variable $\underline{x} = (x_1, x_2) \in \mathbb{R}^2$ comme

$$\begin{cases} \psi^{(0)}(\underline{x}) = \phi(x_1)\phi(x_2) & \psi^{(1)}(\underline{x}) = \psi(x_1)\phi(x_2) \\ \psi^{(2)}(\underline{x}) = \phi(x_1)\psi(x_2) & \psi^{(3)}(\underline{x}) = \psi(x_1)\psi(x_2), \end{cases} \quad (2.80)$$

et utilisées pour définir les coefficients d'ondelettes *non décimés* d'un champ scalaire $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, à l'échelle $a = 2^j$, au point $\underline{n} \in \mathbb{Z}^2$ et selon la direction m :

$$\chi_j^{(m)}[f](\underline{n}) \triangleq 2^{-j} \int_{\mathbb{R}^2} 2^{-j} \overline{f(\underline{x})} \psi^{(m)}(2^{-j}(\underline{x} - \underline{n})) d\underline{x}. \quad (2.81)$$

Le préfacteur 2^{-j} est un facteur de normalisation au sens de la norme ℓ_1 qui permet de rendre compte correctement des propriétés d'invariance d'échelle du champ f tandis que la famille de fonctions normalisées au sens de la norme ℓ_2 :

$$\left\{ 2^{-j} \psi^{(m)}(2^{-j}(\underline{x} - \underline{n})), \quad m = 0, 1, 2, 3, \quad j \in \mathbb{Z}, \quad \underline{n} \in \mathbb{Z}^2 \right\} \quad (2.82)$$

est composée des dilatations et translations des ondelettes mères $\psi^{(m)}$.

À partir des coefficients d'ondelettes *non décimés* $\chi_{j,\underline{n}}^{(m)}[f]$, obtenus grâce à une opération *linéaire* sur le champ f , on peut construire les coefficients d'ondelettes *dominants non décimés*. Les coefficients d'ondelettes dominants ont été introduits initialement sous une forme *décimée* dans [61, 122, 123], qui est rappelée à l'Équation (2.86), adaptée à l'étude globale du champ f (mesure globale d'une régularité locale *uniforme*, étude du spectre multifractal [123], ...). L'extraction de la régularité locale $h(\underline{x})$ en chaque point \underline{x} nécessite l'utilisation des coefficients d'ondelettes dominants *non décimés*, introduits à la Définition 2.10.

Définition 2.10 (Coefficients d'ondelettes *dominants non décimés*). [61, Définition 10] Soient $(\chi_{j,\underline{n}}^{(m)}[f])_{j,\underline{n}}$ les coefficients d'ondelettes d'un champ f . On définit les coefficients d'ondelettes *dominants* à l'échelle 2^j et au point $\underline{n} \in \mathbb{Z}^2$ par :

$$\mathcal{L}_j[f](\underline{n}) = \sup_{\substack{m = \{1, 2, 3\} \\ \lambda_{j',\underline{n}'} \subset 3\lambda_{j,\underline{n}}}} \left| 2^{j\gamma} \chi_{j',\underline{n}'}^{(m)}[f](\underline{n}') \right|, \quad \text{avec} \quad \begin{cases} \lambda_{j,\underline{n}} = [\underline{n}, \underline{n} + 2^j[\\ 3\lambda_{j,\underline{n}} = \bigcup_{\underline{p} \in \{-2^j, 0, 2^j\}^2} \lambda_{j,\underline{n}+\underline{p}}, \end{cases} \quad (2.83)$$

où le supremum porte à la fois sur un voisinage spatial, sur les trois directions m (horizontale, verticale et diagonale) et sur toutes les échelles plus fines disponibles. La Figure 2.12 illustre le voisinage spatial et inter-échelles utilisé pour le calcul d'un coefficient d'ondelette dominant à l'échelle $j = 3$, en \underline{x} représenté par un point rouge. Le paramètre additionnel γ , réel positif, introduit dans [122, 123], permet d'assurer, *via* une intégration fractionnaire, des conditions minimales de régularité, dans le cas où le champ f n'est pas borné. Pour plus de précisions sur le rôle du paramètre d'intégration fractionnaire le lecteur pourra se reporter à [122, 123].

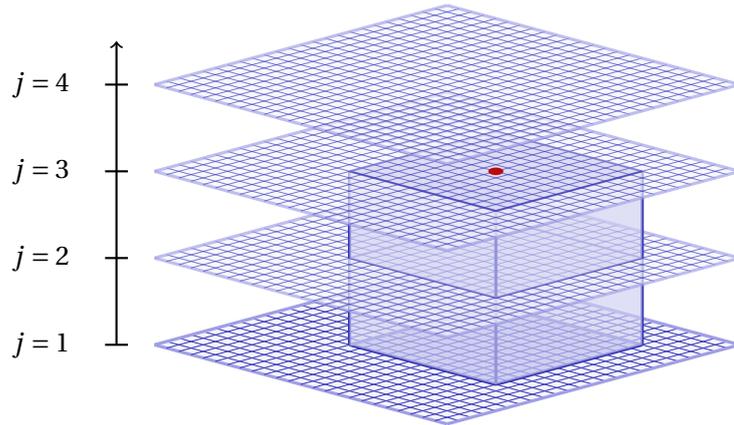


FIGURE 2.12 – Coefficients d'ondelettes dominants non décimés définis à l'Équation (2.83).

Remarque (Lien avec la transformée en ondelettes décimée). La décomposition en ondelettes proposée initialement par [78] est construite à partir d'une famille d'ondelettes *décimées* :

$$\left\{ \Psi_{j,\underline{k}}^{(m)}(\underline{x}) \triangleq 2^{-j} \psi^{(m)}(2^{-j} \underline{x} - \underline{k}), \quad m = 0, 1, 2, 3, \quad j \in \mathbb{Z}, \quad \underline{k} \in \mathbb{Z}^2 \right\} \quad (2.84)$$

formée des dilatations et translations des ondelettes $\psi^{(m)}$. L'orthonormalité de cette famille de fonctions permet à la transformée multiéchelle d'être interprétée comme une décomposition. Les

coefficients d'ondelettes du champ scalaire $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, à l'échelle $a = 2^j$, au point $\underline{n} = 2^j \underline{k}$ et selon la direction m , s'écrivent alors comme un produit scalaire avec l'ondelette $\psi_{j,\underline{k}}^{(m)}$:

$$\tilde{\chi}_{j,\underline{k}}^{(m)}[f] \triangleq 2^{-j} \langle f, \psi_{j,\underline{k}}^{(m)} \rangle = \int_{\mathbb{R}^2} 2^{-2j} \overline{f(\underline{x})} \psi^{(m)}(2^{-j} \underline{x} - \underline{k}) d\underline{x}. \quad (2.85)$$

Cette définition conduit naturellement à la définition des coefficients d'ondelettes dominants *décimés* suivante :

$$\tilde{\mathcal{L}}_j[f](\underline{k}) = \sup_{\substack{m = \{1, 2, 3\} \\ \lambda_{j',\underline{n}'} \subset 3\lambda_{j,\underline{n}}}} \left| 2^{j\gamma} \chi_{j'}^{(m)}[f](\underline{k}') \right|, \quad \text{avec} \quad \begin{cases} \lambda_{j,\underline{n}} = [k2^j, (k+1)2^j[\\ 3\lambda_{j,\underline{n}} = \bigcup_{\underline{p} \in \{-1, 0, 1\}^2} \lambda_{j, \underline{k} + \underline{p}}, \end{cases} \quad (2.86)$$

dont la construction est illustrée à la Figure 2.13.

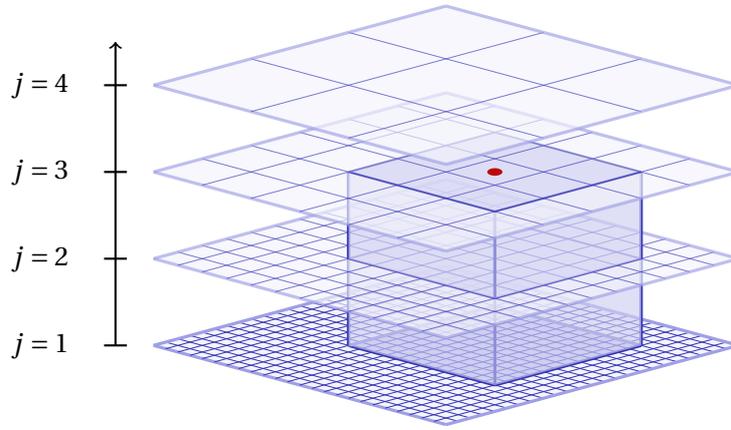


FIGURE 2.13 – Coefficients d'ondelettes dominants décimés.

On peut alors accéder à la régularité locale du champ f au point $\underline{x} \in Y$, notée $h(\underline{x})$, via le comportement au travers des échelles des coefficients d'ondelettes *dominants* [61, 122, 123] grâce à la proposition suivante.

Proposition 2.2. [61, Théorème 1] Soient $(\mathcal{L}_j[f](\underline{n}))_{j,\underline{n}}$ les coefficients d'ondelettes dominants (2.83) d'un champ f dont la régularité locale au point \underline{x} est notée $h(\underline{x})$. Alors, aux échelles les plus fines, les coefficients d'ondelettes dominants ont un comportement en loi de puissance :

$$\mathbb{E} \mathcal{L}_j[f](\underline{n}) \underset{2^j \rightarrow 0}{\lesssim} \eta(\underline{x}) 2^{j(h(\underline{x}) + \gamma)}, \quad (2.87)$$

si et seulement si $h(\underline{x}) + \gamma > 1$. En appliquant le logarithme en base 2, on obtient le comportement linéaire asymptotique au travers des échelles suivant :

$$\mathbb{E} \log_2(\mathcal{L}_j[f](\underline{n})) \underset{2^j \rightarrow 0}{\simeq} j(h(\underline{x}) + \gamma) + \log_2 \eta(\underline{x}). \quad (2.88)$$

À partir de l'Équation (2.88), on remarque qu'une régression linéaire au travers des échelles 2^{j_1} à 2^{j_2} sur le logarithme des coefficients d'ondelettes dominants $\log_2(\mathcal{L}_j[f](\underline{n}))$ permet de construire un estimateur de la régularité locale au point \underline{n} de la forme

$$\hat{h}^{\text{RL}}(\underline{n}) = \sum_{j=j_1}^{j_2} \omega_j \log_2(\mathcal{L}_j(\underline{n})), \quad (2.89)$$

pour des poids ω_j vérifiant les conditions $\sum_{j=j_1}^{j_2} \omega_j = 0$, et $\sum_{j=j_1}^{j_2} j \omega_j$ qui assurent que l'estimateur $\hat{h}^{\text{RL}}(\underline{n})$ est non biaisé [122, 123].

Remarque. Des extensions bayésiennes de la régression linéaire ont également été étudiées pour l'estimation de la régularité locale. Elles ne seront pas discutées dans cette thèse, le lecteur intéressé pourra se référer à [124].

Définition 2.11 (Régularité locale d'une image). [123, Section 2.1] La Définition 2.8 s'étend au cas d'une image $X \in \mathbb{R}^{N_1 \times N_2}$ considérée comme la discrétisation d'un champ scalaire f . La régularité locale de l'image X , au pixel $\underline{n} = (n_1, n_2) \in \{1, \dots, N_1\} \times \{1, \dots, N_2\}$, correspond à la régularité locale du champ f au point \underline{n} et sera donc notée $h(\underline{n})$.

La Proposition 2.3 ci-dessous explicite l'estimation de la régularité locale en chaque pixel d'une image *via* une régression linéaire *pixel par pixel* sur le logarithme de ses coefficients d'ondelettes dominants.

Proposition 2.3 (Régression linéaire au travers des échelles). *La régression linéaire au travers des échelles fournit les estimateurs non-biaisés de la régularité locale, \hat{h}^{RL} et de la puissance locale, \hat{v}^{RL} suivants :*

$$\forall \underline{n} \in \Omega, \quad \begin{pmatrix} \hat{h}^{\text{RL}}(\underline{n}) \\ \hat{v}^{\text{RL}}(\underline{n}) \end{pmatrix} = \mathcal{J}^{-1} \begin{pmatrix} \mathcal{T}(\underline{n}) \\ \mathcal{S}(\underline{n}) \end{pmatrix} \quad (2.90)$$

$$\text{avec } \mathcal{J} = \begin{pmatrix} S_2 & S_1 \\ S_1 & S_0 \end{pmatrix}, \quad S_m = \sum_{j=j_1}^{j_2} j^m, \quad \mathcal{S}(\underline{n}) = \sum_{j=j_1}^{j_2} \log_2 \mathcal{L}_j(\underline{n}), \quad \mathcal{T}(\underline{n}) = \sum_{j=j_1}^{j_2} j \log_2 \mathcal{L}_j(\underline{n}). \quad (2.91)$$

2.3 Segmentation de texture non supervisée

2.3.1 Comparaison de patches pour la segmentation

Comme nous l'avons vu à la Section 2.2, les textures sont caractérisées par des propriétés qui, bien qu'elles aient une définition locale, nécessitent une certaine extension spatiale pour être correctement observées. Par exemple, dans le cas du modèle de texture AM-FM (2.63), une fréquence locale $\nabla \Phi_r(\underline{n})$ ne peut être observée que sur un voisinage spatial du pixel \underline{n} de taille caractéristique minimale $\gtrsim 1/|\nabla \Phi_r(\underline{n})|$.

Afin d'identifier les différentes textures présentes dans une image sans spécifier au préalable les attributs d'intérêt, une méthode consiste à extraire des *patches* élémentaires de cette image. Ensuite, ces patches sont comparés entre eux afin de former des groupes de patches similaires qui pourront être considérés comme représentatifs d'une texture donnée [64, 87, 47]. En choisissant la mesure de similarité avec laquelle sont comparés les patches et le formalisme de segmentation utilisé, il est possible de couvrir une large gamme de problèmes de segmentation de texture.

2.3.1.1 Détection d'objets texturés par contours actifs

Jung et coll. [63, 64, 65] ont proposé une méthode de détection d'objets texturés grâce à une stratégie de contours actifs, dans laquelle les patches de texture sont comparés *via* une métrique de Wasserstein. Pour cela, l'objet texturé est considéré comme un échantillon d'un processus aléatoire dont la distribution, non stationnaire, varie continûment d'un pixel à l'autre. Puis, une stratégie de contours actifs, similaire à celle présentée à la Section 2.1.1 est mise en œuvre, en se concentrant sur la comparaison de patches au sein d'une même région, afin d'affiner progressivement le contour, décrit comme la ligne de niveau d'une fonction indicatrice φ .

Pour une image à segmenter $f : \Upsilon = [0, 1]^2 \rightarrow \mathbb{R}^d$, en chaque pixel \underline{x} de l'image, une collection de *patches* est définie par

$$\forall \underline{x} \in \Upsilon, \quad p_{\underline{x}} \triangleq \left\{ p_{\underline{x}}(\underline{t}) := f(\underline{x} + \underline{t}), \quad \underline{t} \in \left[-\frac{\tau}{2}, \frac{\tau}{2} \right]^2 \right\}. \quad (2.92)$$

La manière la plus directe pour comparer deux patches, centrés respectivement en \underline{x} et \underline{y} , est d'utiliser par exemple une distance construite à partir de la norme L^2 , c'est-à-dire de la forme :

$$d(p_{\underline{x}}, p_{\underline{y}}) \triangleq \int_{\underline{t} \in Y} \exp\left(-\frac{\|\underline{t}\|^2}{2a^2}\right) \|p_{\underline{x}}(\underline{t}) - p_{\underline{y}}(\underline{t})\|^2 d\underline{t}, \quad (2.93)$$

où la gaussienne de largeur a permet de donner plus de poids au centre des patches. Cela amène à une énergie de segmentation *non-locale*, de la forme

$$\mathcal{E}_0(\varphi) \triangleq \int \int \rho\left(H(\varphi(\underline{x})), H(\varphi(\underline{y}))\right) \exp\left(-\frac{\|\underline{x} - \underline{y}\|^2}{2\sigma^2}\right) d(p_{\underline{x}}, p_{\underline{y}}) d\underline{x} d\underline{y} \quad (2.94)$$

avec H la fonction de Heaviside (voir Section 2.1.1) et une fonction ρ qui permet de ne comparer que des patches dans la même région (intérieur ou extérieur de la ligne de contour défini par $\varphi \equiv 0$). Les auteurs de [64] proposent de choisir $\rho(u, v) = 1 - |u - v|$. La variance σ^2 de la fenêtre gaussienne contrôle l'échelle à laquelle l'homogénéité de la texture est imposée.

Une régularisation

$$L(\varphi) = \int \|\nabla H(\varphi(\underline{x}))\| d\underline{x} = \int H'(\varphi(\underline{x})) \|\nabla \varphi(\underline{x})\| d\underline{x} \quad (2.95)$$

est ajoutée afin de pénaliser le périmètre de l'objet et ainsi obtenir des contours suffisamment réguliers, ce qui amène au problème de minimisation de fonctionnelle :

$$\underset{\varphi}{\text{minimize}} \mathcal{E}_0(\varphi) + \gamma L(\varphi) \quad (2.96)$$

avec $\gamma > 0$ un paramètre de régularisation.

Si les méthodes de ce type obtiennent de bons résultats pour des patches de petite taille (τ faible à l'Équation (2.92)), elles achoppent lorsque des patches de taille plus grande sont nécessaires, ce qui est le cas notamment lorsque du bruit est présent sur l'image.

Afin de pallier ce problème, les auteurs de [64] proposent deux améliorations dans la procédure. D'une part, une normalisation locale $C(\varphi, \underline{x})$ est introduite et amène à la définition d'une énergie renormalisée en chaque pixel :

$$\mathcal{E}(\varphi) = \int \frac{1}{C(\varphi, \underline{x})} \left(\int \rho\left(H(\varphi(\underline{x})), H(\varphi(\underline{y}))\right) \exp\left(-\frac{\|\underline{x} - \underline{y}\|^2}{2\sigma^2}\right) d(p_{\underline{x}}, p_{\underline{y}}) d\underline{y} \right) d\underline{x}. \quad (2.97)$$

Les disparités entre pixels sont ainsi réduites, particulièrement pour les pixels proches de la frontière $\varphi \equiv 0$.

D'autre part, la distance L^2 de l'Équation (2.93) est remplacée par une distance alternative, construite à partir de la distance de Wasserstein entre les patches $p_{\underline{x}}$ et $p_{\underline{y}}$, interprétés comme des distributions à valeurs dans \mathbb{R}^d .

Définition 2.12 (Distance de Wasserstein). [90, Proposition 2.2] Soient $X = \{X_i\}_{i=1}^N$ et $Y = \{Y_i\}_{i=1}^N$ deux nuages de points dans \mathbb{R}^d , leur distance de Wasserstein au sens L^p est définie par

$$W^p(X, Y) \triangleq \min_{\sigma \in \Sigma_N} \sum_{i=1}^N \|X_i - Y_{\sigma(i)}\|^p \quad (2.98)$$

où Σ_N est l'ensemble des permutations de $\{1, \dots, N\}$.

Contrairement à la distance L^2 , opérant sur des vecteurs, la distance de Wasserstein est intrinsèquement une distance entre *distributions*. Ainsi, la distance entre deux patchs utilisée pour construire l'énergie \mathcal{E} s'écrit

$$d(p_{\underline{x}}, p_{\underline{y}}) = W^p \left([p_{\underline{x}}], [p_{\underline{y}}] \right) \quad (2.99)$$

où $[p_{\underline{x}}]$ désigne le nuage de points de \mathbb{R}^d constitué des éléments du patch $p_{\underline{x}}$.

Si une distance L^2 , telle que celle présentée à l'Équation (2.93), permet de segmenter des images lisses par morceaux ou comportant des motifs simples [65], la segmentation de textures complexes, nécessite d'être en mesure de discriminer finement des distributions, ce que permet la distance de Wasserstein [84]. En effet, outre qu'elle gère tous les types de distributions, notamment les distributions localisées, la distance de Wasserstein considère tous les modes des distributions considérées et prend en compte leurs positions respectives ce qui lui permet de quantifier très précisément la distance entre des distributions riches, comme celles caractérisant les textures. À la suite de [84], recommandant l'utilisation de la distance de Wasserstein en traitement d'images, elle a été utilisée avec succès pour traiter de nombreux problèmes, tels que la recherche d'images à partir de leur contenu [104], la correction de distorsion [55] ou la synthèse de textures [97].

En pratique, la distance de Wasserstein est extrêmement coûteuse à calculer : une résolution combinatoire peut être obtenue en $\mathcal{O}(N^{5/2} \log(N))$ opérations [14] et les auteurs de [64] préconisent l'utilisation de projections uni-dimensionnelles de la distance de Wasserstein [97], calculable chacune en $\mathcal{O}(N \log(N))$ opérations. La distance de Wasserstein « par tranches » construite à partir de l'agrégation de distances de Wasserstein uni-dimensionnelles s'avère une bonne approximation de la distance W^p [97].

2.3.1.2 Segmentation de texture par patchs grâce à la norme L^1

Une alternative à l'utilisation d'une distance complexe, et donc lourde à calculer, telle que celle de Wasserstein décrite à la section précédente, est de définir des patchs incluant d'éventuelles transformations géométriques (rotations, déformations, par exemple). Ce type d'approches est motivé par le fait que plusieurs occurrences d'une même texture peuvent apparaître dans une image sous des formes variées, avec des effets de perspectives différents par exemples, ou orientées selon plusieurs directions.

La méthode développée par Oliver *et coll.* [87] permet, grâce à une adaptation automatique de la taille, de la forme et de l'orientation des patchs, d'identifier précisément les motifs de textures apparaissant dans une image, malgré de possibles déformations. Puis, la mesure de similarité construite est insérée dans une fonctionnelle de type Mumford-Shah, telle qu'introduite à la Section 2.1.1, avec une attache aux données L^1 afin de fournir une segmentation de l'image en différentes textures. De plus, un échantillon représentatif de la texture de chaque région est fourni par la procédure.

Pour une image à segmenter $f : Y \subset \mathbb{R}^2 \rightarrow \mathbb{R}^d$, une collection $\mathcal{P}_f = \{p_{\underline{x}}, \underline{x} \in Y\}$ de *patches covariants* est définie par

$$\forall \underline{x} \in Y, p_{\underline{x}} \triangleq \left\{ p_{\underline{x}}(\underline{t}) := f(\underline{x} + T_{\underline{x}}^{-1/2} \underline{t}), \quad \underline{t} \in \Delta \right\}, \quad (2.100)$$

où Δ est un disque centré en l'origine, de rayon fixé, et $T_{\underline{x}}$ est la métrique donnée, sous forme matricielle, par le tenseur covariant sous les transformations affines associé à l'image f , décrite par [39].

Grâce à la métrique covariante $T_{\underline{x}}$, une mesure de similarité multi-échelle, invariante sous les transformations affines, est fournie par :

$$\mathcal{D}^q(p_{\underline{x}}, p_{\underline{y}}) \triangleq \int_{\Delta} g(\underline{t}) \left\| f(\underline{x} + T_{\underline{x}}^{-1/2} \underline{t}) - f(\underline{y} + T_{\underline{y}}^{-1/2} \underline{t}) \right\|_q^q d\underline{t} \quad (2.101)$$

où g est une fonction géodésique de pondération et $p_{\underline{x}}$ et $p_{\underline{y}}$ les deux patches à comparer.

La procédure de segmentation de l'image f en k régions développée par [87] permet d'obtenir deux choses : d'une part, une partition du domaine de l'image

$$Y = \cup_{i=1}^k \overline{Y}_i \quad (2.102)$$

avec $(Y_i)_{i=1}^k$ des ouverts disjoints dont \overline{Y}_i désigne l'adhérence, et d'autre part, un petit nombre de patches représentatifs $\mathbf{P} \triangleq \{p_{Y_1}, \dots, p_{Y_k}\}$, associés respectivement aux régions Y_1, \dots, Y_k . Ainsi, un pixel de la région Y_i correspond à un patch $p_{\underline{x}}$ proche de p_{Y_i} au sens de la mesure de similarité \mathcal{D}^q , définie par (2.101). Tandis qu'en segmentation d'image on cherche à représenter une image complexe à partir d'un nombre limité de couleurs (ou de niveaux de gris), pour la segmentation de texture *via* des patches il s'agit de simplifier la collection de patches \mathcal{P}_f pour se ramener à un petit nombre de patches élémentaires.

En notant les contours $B \triangleq \cup_{i=1}^k \partial Y_i$, où ∂Y_i désigne la frontière de Y_i , et χ_{Y_i} l'indicatrice de Y_i , à valeurs dans $\{0, 1\}$, l'énergie de type L^1 -Mumford-Shah se formule

$$\mathcal{E}(\mathbf{P}, B) = \mathcal{H}^1(B) + \lambda \int_Y \mathcal{D}^1(p_{\underline{x}}, p_{\underline{x}}) d\underline{x} \quad (2.103)$$

où \mathbf{p} désigne une « fonction de patch constante par morceaux », définie comme $\mathbf{p}_{\underline{x}} = \sum_{i=1}^k p_{Y_i} \chi_{Y_i}(\underline{x})$. Or, en utilisant le fait que les indicatrices $(\chi_i)_{i=1}^k$ sont des fonctions à variations bornées, le terme de longueur de contours $\mathcal{H}^1(B)$ peut se réécrire sous forme d'une variation totale et la fonctionnelle $\mathcal{E}(\mathbf{P}, B)$ se reformule comme

$$\mathcal{E}(\mathbf{P}, \boldsymbol{\chi}) = \sum_{i=1}^k \int_Y \|\nabla \chi_{Y_i}(\underline{x})\| d\underline{x} + \lambda \sum_{i=1}^k \int_Y \int_{\Delta} g(\underline{t}) \left\| p_{Y_i}(\underline{t}) - f(\underline{x} + T_{\underline{x}}^{-1/2} \underline{t}) \right\|_1 d\underline{x} d\underline{t}, \quad (2.104)$$

où $\boldsymbol{\chi} = (\chi_{Y_1}, \dots, \chi_{Y_k})$ et q est fixé à 1.

Le problème de minimisation

$$\underset{(\mathbf{P}, \boldsymbol{\chi})}{\text{minimize}} \mathcal{E}(\mathbf{P}, \boldsymbol{\chi}) \quad (2.105)$$

réalise un compromis entre la régularité des contours et la fidélité de l'approximation des patches locaux par la fonction de patches constante par morceaux. En outre, grâce à l'utilisation de la norme L^1 les patches élémentaires $(p_{Y_i})_{i=1}^k$ correspondent aux patches médians respectifs des régions $(Y_i)_{i=1}^k$. Enfin, la norme L^1 assure la robustesse par rapport au bruit impulsionnel et aux valeurs extrêmes, qui peuvent apparaître lorsqu'un patch est centré sur un contour par exemple.

La minimisation (2.105) est un problème d'optimisation non convexe, non continu et non différentiable soulevant de nombreux problèmes théoriques et pratiques. Les auteurs de [87] proposent donc de relaxer les indicatrices χ_{Y_i} et de travailler avec des probabilités d'appartenance $0 \leq \omega_{Y_i} \leq 1$, satisfaisant pour tout $\underline{x} \in Y$, $\sum_{i=1}^k \omega_{Y_i}(\underline{x}) = 1$. La fonctionnelle relaxée $\tilde{\mathcal{E}}(\mathbf{P}, \boldsymbol{\omega})$ est alors convexe *séparément* en $\boldsymbol{\omega} = (\omega_{Y_1}, \dots, \omega_{Y_k})$, ce qui permet de montrer l'existence d'au moins une solution [71]. Enfin, Oliver *et coll.* résolvent le problème relaxé en introduisant des variables auxiliaires et en effectuant une minimisation alternée.

2.3.2 Modèle bayésien pour la segmentation de texture

À partir du modèle de textures gaussiennes présenté à la Section 2.2.2.3, Vacar et Giovanelli [117] construisent un modèle hiérarchique permettant d'effectuer de manière jointe la déconvolution et la segmentation d'images texturées.

L'image observée $Y \in \mathbb{R}^{N_1 \times N_2}$ consiste en une version floutée et bruitée d'une image originale Z , composée de patches extrait de k textures gaussiennes $(X_i)_{i=1}^k$, dont les matrices de covariances \mathbf{R}_i sont inconnues et paramétrées par $(\gamma_i, \vartheta_i)_{i=1}^k$, où γ_i désigne la puissance spectrale et ϑ_i encode la structure de \mathbf{R}_i (voir [117]).

L'agencement des patches de textures, i.e., la segmentation sous-jacente, est modélisée par un champ aléatoire de Potts L . En chaque pixel de l'image, $L(\underline{n}) = i$ consiste en une étiquette $1 \leq i \leq k$ qui indique quelle texture X_i est présente à ce pixel de l'image Z , ce qui se traduit par

$$Z = \sum_{i=1}^k S_i(L)X_i \quad (2.106)$$

où $S_i(L)$ est une matrice binaire diagonale de taille $N_1 N_2 \times N_1 N_2$ qui sélectionne au pixel \underline{n} la texture telle que $i = L(\underline{n})$, i.e., $Z(\underline{n}) = X_i(\underline{n})$.

Le champ des étiquettes est paramétré par sa *granularité* β , inconnue, qui détermine la taille caractéristique des régions de la segmentation. La distribution de probabilité *a priori* du champ L s'écrit

$$\Pr(L|\beta) = \frac{1}{\mathcal{Z}(\beta)} \exp\left(\beta \sum_{\underline{n} \sim \underline{n}'} \delta(L(\underline{n}), L(\underline{n}'))\right) \quad (2.107)$$

où $\mathcal{Z}(\beta)$ est une constante de normalisation, $\underline{n} \sim \underline{n}'$ signifie que \underline{n} et \underline{n}' sont des pixels adjacents et δ est le delta de Kronecker.

Remarque (Lien avec la Variation Totale ℓ_0). La quantité définie par

$$\|\nabla L\|_0 \triangleq 1 - \sum_{\underline{n} \sim \underline{n}'} \delta(L(\underline{n}), L(\underline{n}')) \quad (2.108)$$

mesure le nombre de pixels voisins ne partageant pas la même étiquette. Ainsi, $\|\nabla L\|_0$ peut s'interpréter comme le nombre de pixels où le gradient de la carte des étiquettes est non nul. Nous verrons à la section suivante que les pénalisations de la forme $\|\nabla \cdot\|$ sont bien adaptées pour la segmentation d'image.

Enfin, la dégradation de l'image est modélisée par une déformation Φ et un bruit additif gaussien i.i.d. ε de variance σ^2 inconnu, de sorte que

$$Y = \Phi Z + \varepsilon \implies \Pr(Y|Z, \sigma^2) = (2\pi\sigma^2)^{-N_1 N_2/2} \exp\left(-\frac{\|Y - \Phi Z\|^2}{2\sigma^2}\right) \quad (2.109)$$

où Φ est une matrice Toeplitz, représentant une convolution, par exemple par un opérateur de flou.

Tous ces éléments sont inclus dans un modèle bayésien fortement hiérarchisé, dont la distribution jointe s'exprime en fonction des distributions des hyperparamètres :

$$\begin{aligned} \Pr(Y, X_{1,\dots,k}, L, \gamma_{1,\dots,k}, \vartheta_{1,\dots,k}, \sigma^2) &= \Pr(Y|X_{1,\dots,k}, L, \sigma^2) \prod_{i=1}^k \Pr(X_i|\gamma_{1,\dots,k}, \vartheta_{1,\dots,k}) \Pr(L|\beta) \times \\ &\Pr(\beta) \Pr(\sigma^2) \prod_{i=1}^k \Pr(\gamma_{1,\dots,k}) \prod_{i=1}^k \Pr(\vartheta_{1,\dots,k}), \end{aligned} \quad (2.110)$$

À partir de cette distribution jointe on peut écrire la distribution *a posteriori*

$$\Pr(X_{1,\dots,k}, L, \gamma_{1,\dots,k}, \vartheta_{1,\dots,k}, \sigma^2 | Y) \quad (2.111)$$

$$\propto \exp\left(-\frac{1}{2\sigma^2} \left\| Y - \Phi \sum_{i=1}^k S_i(L) X_i \right\|^2\right) \exp\left(\beta \sum_{\underline{n} \sim \underline{n}'} \delta_{L(\underline{n}), L(\underline{n}')}\right) \prod_{i=1}^k \exp\left(-\frac{1}{2} \|X_i\|_{\mathbf{R}_i^{-1}}^2\right), \quad (2.112)$$

où les distributions des hyperparamètres ne sont pas explicitées (le lecteur pourra se référer à [117] pour plus de précisions sur leur choix).

La reconstruction bayésienne consiste à maximiser la distribution *a posteriori* pour une donnée Y observée fixée, ce qui se traduit différemment selon le type d'erreur considéré. La qualité de reconstruction de la carte des étiquettes est mesurée grâce à une fonction de coût binaire, ainsi \hat{L} maximise la probabilité marginale *a posteriori*. Les variables de texture $(X_i, \gamma_i, \vartheta_i)_{i=1}^k$ et la granularité β sont comparés en terme d'erreur quadratique, par conséquent les estimées associées, $(\hat{X}_i, \hat{\gamma}_i, \hat{\vartheta}_i)_{i=1}^k$, sont des moyennes *a posteriori*.

La complexité de la distribution *a posteriori* (2.111) exclut le calcul direct des estimées, qui sont obtenus grâce à un algorithme d'échantillonnage de Gibbs séquentiel [117]. Les propriétés de séparabilité de la distribution *a posteriori* sont mise à profit pour effectuer successivement l'échantillonnage de chaque variable et chaque paramètre, ce qui permet de se ramener à des problèmes de taille réduite.

2.3.3 Segmentation à partir d'attributs de texture locaux

2.3.3.1 Factorisation matricielle

Yuan *et coll.* [125] ont proposé une méthode de segmentation de textures grâce à une factorisation matricielle. Sur la base de précédents travaux [73], une texture est caractérisée par les statistiques de sa réponse à une base de \mathcal{F} filtres de Gabor $(g_1, \dots, g_{\mathcal{F}})$.

En chaque pixel \underline{n} de l'image, une fenêtre carré $\mathbf{W}_{\underline{n}}$, centrée en \underline{n} , est extraite et convoluée avec chacun des filtres $g_1, \dots, g_{\mathcal{F}}$ pour former \mathcal{F} sous-bandes $\mathbf{W}_{\underline{n}}^{(1)}, \dots, \mathbf{W}_{\underline{n}}^{(\mathcal{F})}$. Puis, les \mathcal{F} histogrammes $H_{\mathbf{W}_{\underline{n}}}^1, \dots, H_{\mathbf{W}_{\underline{n}}}^{\mathcal{F}}$ de ces sous-bandes sont calculés et réunis pour former un histogramme spectral local au pixel \underline{n}

$$H_{\mathbf{W}_{\underline{n}}} \triangleq \frac{1}{\text{card}(\mathbf{W}_{\underline{n}})} \left(H_{\mathbf{W}_{\underline{n}}}^1, \dots, H_{\mathbf{W}_{\underline{n}}}^{\mathcal{F}} \right), \quad (2.113)$$

où $\text{card}(\mathbf{W}_{\underline{n}})$ désigne le nombre de pixels de la fenêtre $\mathbf{W}_{\underline{n}}$. Ainsi, en notant \mathcal{B} le nombre de classes (*bins*) des histogrammes (fixe et commun à tous), pour chaque pixel \underline{n} , l'histogramme spectral local est un vecteur $H_{\mathbf{W}_{\underline{n}}} \in \mathcal{F} \times \mathcal{B} \triangleq M$ composantes.

L'histogramme local $H_{\mathbf{W}_{\underline{n}}}$ est supposé *uniforme* au sein d'une région correspondant à une texture homogène. Chaque texture homogène est donc caractérisée par un attribut de taille M . Pour une image composée de N pixels, est associé à chaque pixel son histogramme spectral local $H_{\mathbf{W}_{\underline{n}}}$, de dimension M . L'ensemble des attributs est donc représenté par une matrice $\mathbf{Y} \in \mathbb{R}^{M \times N}$.

A priori, le nombre d'attributs extraits de cette manière est élevé. L'hypothèse retenue par les auteurs de [125] est que seul un petit nombre L de combinaisons linéaires de ces attributs sont discriminantes pour distinguer les textures rencontrées. La matrice \mathbf{Y} peut alors se décomposer en

$$\mathbf{Y} = \mathbf{Z}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

où $\mathbf{Z} \in \mathbb{R}^{M \times L}$ contient les attributs discriminants, $\boldsymbol{\beta} \in \mathbb{R}^{L \times N}$ est un vecteur de poids et $\boldsymbol{\epsilon}$ correspond à une erreur de modèle.

Lorsque la matrice \mathbf{Z} est connue, une régression linéaire fournit un estimateur des poids

$$\hat{\boldsymbol{\beta}}(\mathbf{Y}) \triangleq (\mathbf{Z}^* \mathbf{Z})^{-1} \mathbf{Z}^* \mathbf{Y}, \quad (2.114)$$

où \mathbf{Z}^* désigne la matrice transposée de \mathbf{Z} , à partir duquel on peut classer un pixel \underline{n} dans la classe ℓ de poids $\beta_{\ell, \underline{n}}$ maximal.

Néanmoins, en général, on souhaite également estimer \mathbf{Z} afin d'obtenir une procédure automatisée et suffisamment souple pour décrire une grande variété de textures. Pour cela, Yuan *et coll.* proposent de résoudre le problème de moindres carrés positifs suivant

$$(\hat{\mathbf{Z}}, \hat{\boldsymbol{\beta}}) \in \underset{\mathbf{Z}, \boldsymbol{\beta} \geq 0}{\text{Argmin}} \|\mathbf{Y} - \mathbf{Z}\boldsymbol{\beta}\|^2 + \lambda_1 \|\mathbf{Z}\|^2 + \lambda_2 \|\boldsymbol{\beta}\|^2 \quad (2.115)$$

où $\mathbf{Z}, \boldsymbol{\beta} \geq 0$ signifie que les coefficients de \mathbf{Z} et de $\boldsymbol{\beta}$ doivent tous être positifs. Les termes de pénalisation quadratique permettent d'éviter que les matrices \mathbf{Z} et $\boldsymbol{\beta}$ n'aient des normes élevées ce qui causerait des instabilités numériques. La résolution du Problème (2.115) est effectuée grâce à un algorithme de descente alternant entre la minimisation des moindres carrés et la satisfaction des contraintes de positivité.

Afin d'obtenir un algorithme efficace, un soin particulier est apportée à l'initialisation, cruciale pour la convergence des schémas alternés. Les auteurs proposent dans ce but d'utiliser la décomposition en valeurs singulières de la matrice \mathbf{Y} pour effectuer une approximation de faible rang de \mathbf{Y} et se ramener à un nombre d'attributs discriminants $L' = \mathcal{O}(1) \ll L$.

2.3.3.2 Détection de contours et segmentation hiérarchisée

Arbelaez *et coll.* ont combiné des attributs d'intensité lumineuse, de couleur et de texture pour construire une méthode très robuste de détection de contours [5]. Ensuite, grâce à l'extraction d'une information quantitative modélisant la *probabilité de présence* d'un contour en chaque pixel, une méthode de segmentation hiérarchisée est mise en œuvre. Ainsi, une segmentation initiale, obtenue par la méthode de ligne de partage des eaux décrite à la Section 2.1.2.3, est raffinée en fusionnant adéquatement certaines régions séparées par un contour « de faible probabilité » au sens de la *carte de contours ultramétrique*.

Un premier ingrédient consiste en la construction d'un gradient orienté $\mathcal{G}(\underline{n}, \vartheta)$ au pixel \underline{n} et d'angle ϑ . Pour cela, un disque de rayon r , centré en \underline{n} est coupé en deux par un diamètre d'orientation $\vartheta \in [0, \pi[$, les histogrammes h_+ , h_- de la quantité considérée (intensité, couleur ou attribut de texture), calculés sur chaque demi-disque, sont comparés grâce à une distance

$$\chi^2(h_+, h_-) = \frac{1}{2} \sum_i \frac{(h_+(i) - h_-(i))^2}{h_+(i) + h_-(i)} \quad (2.116)$$

qui est ensuite filtrée pour faire ressortir les maxima locaux et lisser les pics de détection multiples dans la direction orthogonale à ϑ . Les gradients orientés sont calculés sur quatre canaux de l'image à segmenter : les trois canaux de la représentation CIE Lab (intensité lumineuse, couleur a et couleur b) et un canal de texture contenant un *indice de texton*.

Les indices de texton sont définis en amont du calcul des gradients orientés. Pour cela, l'image est filtrée par un ensemble de 17 filtres gaussiens qui permettent d'obtenir en chaque pixel un vecteur d'attributs de texture de taille 17, puis un algorithme de k -moyennes (tel que décrit à la

Section 2.1.2.1) permet de regrouper ces attributs en $k = 32$ groupes dont les k centroïdes définissent ainsi k textures élémentaires de références, appelées *textons*. Chaque pixel se voit associer une étiquette dans $\{1, \dots, k\}$ en fonction du texton auquel il est rattaché dans la partition par k -moyennes.

L'originalité de la méthode proposée par les auteurs de [5] consiste à combiner des gradients calculés à différentes échelles, c'est-à-dire à calculer les histogrammes locaux h_+ et h_- pour différentes tailles de disque. Trois échelles sont considérées $[r/2, r, 2r]$, ce qui amène à un premier détecteur de contours local, multi-attributs et multi-échelles

$$\text{mPb}(\underline{n}, \vartheta) = \sum_{r' \in [r/2, r, 2r]} \sum_c \alpha_{r', c} \mathcal{G}_{r', c}(\underline{n}, \vartheta) \quad (2.117)$$

où l'indice c désigne le canal considéré (intensité lumineuse, couleur a, couleur b ou indice de texton) et $\alpha_{r', c}$ sont des poids appris sur une base de données d'entraînement (par exemple BSDS³).

Un deuxième détecteur de contour, global, est construit grâce à une matrice d'affinité W , qui connecte les pixels \underline{n} et \underline{n}' , au sein d'un voisinage de taille fixée, *via* une affinité W et une information globale D

$$W(\underline{n}, \underline{n}') \triangleq \exp\left(-\max_{\underline{p} \in [\underline{n}, \underline{n}']} \text{mPb}(\underline{p})/\rho\right) \quad \text{et} \quad D(\underline{n}, \underline{n}) = \sum_{\underline{n}'} W(\underline{n}, \underline{n}') \quad (2.118)$$

où $[\underline{n}, \underline{n}']$ désigne le segment reliant \underline{n} et \underline{n}' . La résolution du problème de recherche de vecteurs propres généralisés $\{\mathbf{v}_0, \dots, \mathbf{v}_M\}$

$$(D - W) \mathbf{v} = \lambda D \mathbf{v}, \quad (2.119)$$

associés aux valeurs propres $\lambda_0 = 0 \leq \lambda_1 \leq \dots \leq \lambda_M$, permet de définir le second détecteur de contours

$$\text{sPb}(\underline{n}, \vartheta) = \sum_{m=1}^M \frac{1}{\sqrt{\lambda_m}} \nabla_{\vartheta} v_m(\underline{n}) \quad (2.120)$$

où $\nabla_{\vartheta} v_m(\underline{n})$ désigne la dérivée directionnelle selon ϑ du vecteur propre \mathbf{v}_m au pixel \underline{n} .

Les détecteurs de contours mPb et sPb s'avèrent complémentaires, au sens où mPb détecte tous les bords de l'image, tandis que sPb est sensible aux courbes les plus saillantes de l'image. Leur combinaison linéaire donne alors un détecteur de contours multi-attributs, multi-échelles combinant des informations locales et spectrales

$$\text{gPb}(\underline{n}, \vartheta) \triangleq \text{mPb}(\underline{n}, \vartheta) + \gamma \text{sPb}(\underline{n}, \vartheta), \quad (2.121)$$

où le paramètre γ est appris sur une base d'entraînement (par exemple BSDS³), conjointement aux poids $\alpha_{r', c}$ intervenant dans la définition de mPb (2.117).

À partir des contours extraits par gPb, contenant des informations multi-échelles sur les textures, une segmentation est effectuée grâce à une méthode de ligne de partage des eaux (voir Section 2.1.2.3). De plus, le résultat peut être raffiné en prenant en compte l'intensité des contours pour fusionner les régions séparées par des contours de très faible intensité [5], ce qui amène à une amélioration sensible de la segmentation finale.

3. <https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/>

2.3.4 Variation Totale seuillée appliquée à la régularité locale (ROF_h-S)

2.3.4.1 Principe

Une classe de méthodes de segmentation de textures consiste à appliquer une procédure de segmentation d'image, par exemple une de celles décrites à la Section 2.1, non pas à l'image elle-même, mais à une quantité estimée localement, caractéristique des textures considérées. Nous reprenons ici l'exemple, étudié dans des travaux récents [96, 83], de la segmentation de texture à partir de la régularité locale h , attribut fractal décrit à la Définition 2.8 grâce à au débruitage par Variation Totale, décrit à la Section 2.1.1.6.

2.3.4.2 Texture fractale par morceaux

Par définition, une image $\{X(\underline{n}), \underline{n} \in \Omega\}$, ou une région $\Omega_i \subset \Omega$ d'une image, correspond à une texture fractale *homogène* si elle possède la même régularité locale $h(\underline{n}) \equiv H$ en tout pixel $\underline{n} \in \Omega$ (ou $\underline{n} \in \Omega_i$ pour une région). Dans ce modèle, une image composée de plusieurs textures *homogènes* est une image dont la régularité locale $h(\underline{n})$ est *constante par morceaux*. Chaque région Ω_i , correspondant à une texture différente, est associée une valeur donnée de la régularité locale

$$\Omega_i \triangleq \{\underline{n} \in \Omega, h(\underline{n}) \equiv H_i\}. \quad (2.122)$$

Un exemple de texture fractale par morceaux est proposé à la Figure 2.14, qui permettra d'illustrer les performances de la segmentation de textures par Variation Totale seuillée appliquée à la régularité locale. À partir d'une segmentation sous-jacente, ici le masque elliptique présenté en Figure 2.14a, une texture synthétique est générée, en Figure 2.14b, composée de deux textures homogènes, respectivement caractérisées par une régularité locale $h(\underline{n}) \equiv H_1$ pour la texture de fond et $h(\underline{n}) \equiv H_2$ pour la texture occupant l'ellipse centrale. La définition et la synthèse d'une telle texture fractale homogène par morceaux sera discutée en détail au Chapitre 3.

2.3.4.3 Variation Totale seuillée sur la régularité locale

Une première étape consiste à estimer en chaque pixel la régularité locale $\hat{h}^{\text{RL}}(\underline{n})$ au moyen de la régression linéaire sur les logarithmes des coefficients dominants de l'Équation (2.89). La carte de régularité locale $\hat{\mathbf{h}}^{\text{RL}} \in \mathbb{R}^{N_1 \times N_2}$ obtenue est présentée en Figure 2.14c. Dans le cas idéal d'une estimation « parfaite », on s'attend à une carte constante par morceaux, ayant la géométrie elliptique de la Figure 2.14a. Or, on remarque qu'elle est extrêmement bruitée, ce qui s'explique par la forte variance de l'estimateur des moindres carrés de l'Équation (2.89). Tel quel, cet estimateur $\hat{\mathbf{h}}^{\text{RL}}$ n'est pas adapté pour fournir une segmentation de texture. Il peut néanmoins être utilisé comme un point de départ, puis être régularisé pour prendre en compte la contrainte de constance par morceaux, nécessaire pour une tâche de segmentation [96, 83, 89].

La méthode de segmentation de textures utilisant la Variation Totale consiste à calculer une estimée régularisée $\hat{\mathbf{h}}^{\text{ROF}}$, définie par :

$$\hat{\mathbf{h}}^{\text{ROF}} = \underset{\mathbf{h} \in \mathbb{R}^{N_1 \times N_2}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{h} - \hat{\mathbf{h}}^{\text{RL}}\|_{\text{Fro}}^2 + \lambda \text{TV}(\mathbf{h}), \quad (2.123)$$

qui est le résultat d'un compromis entre la fidélité à l'estimée de la régularité locale par régression linéaire, c'est-à-dire l'adéquation avec le modèle de texture fractale, et le caractère constant par morceaux, requis pour la segmentation. Le Problème (2.123) est résolu numériquement grâce aux algorithmes proximaux, qui seront détaillés aux Sections 4.4.4 et 4.4.5. Sur l'exemple de la Figure 2.14, on constate que l'estimée par régression linéaire, en Figure 2.14c, très bruitée, est fortement améliorée par la régularisation, qui amène à l'estimée $\hat{\mathbf{h}}^{\text{ROF}}$ de la Figure 2.14d.

Enfin, une segmentation $\Omega = \cup_{i=1}^k \hat{\Omega}_i^{\text{ROF}}$ en k régions est obtenue par application de l'Algorithme 1 de seuillage itéré proposé par [16, Algorithme T-ROF] à l'estimée régularisée \hat{h}^{ROF} . Le débruitage par Variation Totale (2.123) appliqué à \hat{h}^{RL} suivi par une étape de seuillage itéré constitue la méthode de segmentation de texture appelée ROF $_h$ -S dans la suite. De plus, l'Algorithme 1 fournit les valeurs moyennes $\hat{H}_1^{\text{ROF}}, \dots, \hat{H}_2^{\text{ROF}}$. Ces résultats peuvent être présentés sous la forme d'une carte de régularité locale composée de k régions, notée $S\hat{h}^{\text{ROF}}$, telle que $S\hat{h}^{\text{ROF}}$ est constante égale à \hat{H}_i^{ROF} sur la région $\hat{\Omega}_i^{\text{ROF}}$. Dans notre exemple à $k = 2$ régions, le nombre d'itérations de l'Algorithme 1 est fixé à $M_{\max} = 12$, et le seuillage décrit est appliqué à l'estimée \hat{h}^{ROF} de la Figure 2.14d. La segmentation résultante, présentée en Figure 2.14e, reproduit correctement la géométrie du masque de la Figure 2.14a et atteint un score de 93% de pixels bien classifiés. On note néanmoins que le contour est irrégulier, comparé au contour extrêmement lisse de l'ellipse sous-jacente. Cela nous amènera au Chapitre 4 à proposer des méthodes plus élaborées.

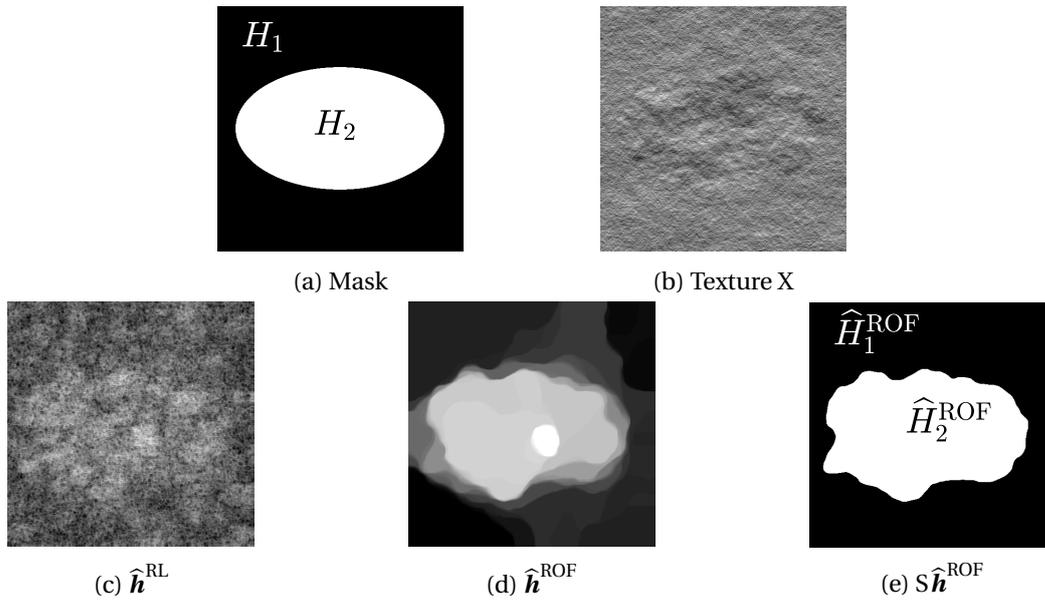


FIGURE 2.14 – **Analyse d'une texture fractale par morceaux.** (a) Masque $H_1 = 0.5$ et $H_2 = 0.8$. (b) Échantillon de texture. (c) Estimée par régression linéaire de la régularité locale, \hat{h}^{RL} . (d) Estimée, \hat{h}^{ROF} , régularisée par Variation Totale de la régularité locale. (e) Segmentation associée, $S\hat{h}^{\text{ROF}}$, obtenue par le seuillage itératif à $k = 2$ régions présenté à l'Algorithme 1.

2.4 Apprentissage profond pour la segmentation supervisée

Ces dernières années, l'augmentation spectaculaire des capacités de calcul a permis de traiter des volumes de données de plus en plus importants, ce qui a entraîné une explosion de l'utilisation des méthodes d'apprentissage statistique, notamment pour les tâches de segmentation d'images. Cette section propose de présenter les grandes idées sur lesquelles repose l'apprentissage statistique par réseaux de neurones profonds. Néanmoins, nous ne visons pas à l'exhaustivité dans ce domaine très actif et déjà largement documenté [110, 58, 50]. La Section 2.4.1 présente les principes généraux de l'apprentissage statistique et les outils qu'il nécessite. Les réseaux de neurones convolutionnels, au cœur de l'apprentissage statistique sur des bases d'images sont introduits à la Section 2.4.2. La Section 2.4.3.3 détaille comment le problème de classification d'images par réseau de neurones s'étend au cas de la segmentation d'images, vue comme un problème de classification pixel par pixel. De plus, quelques éléments techniques nécessaires à la construction de réseaux de neurones pour la segmentation y sont décrits. Enfin, nous présentons une sélection de contributions récentes au problème de segmentation d'images texturées par apprentissage profond.

2.4.1 Apprentissage profond : principes généraux, méthodes et outils

2.4.1.1 Objectif

L'apprentissage statistique consiste à construire un estimateur $\hat{\mathbf{x}}(\mathbf{y}; \Theta)$, d'une quantité \mathbf{x} , à partir d'une observation \mathbf{y} . Cet estimateur est paramétré par un ensemble de poids, stockés dans l'hyperparamètre Θ . On parle d'apprentissage *supervisé* lorsque les poids Θ sont inférés à partir d'une base de données, composée de couples $(\mathbf{x}_i, \mathbf{y}_i)$, constitués d'observées \mathbf{y}_i pour lesquels on connaît la « vraie » quantité sous-jacente \mathbf{x}_i .

La quantité \mathbf{x} que l'on souhaite estimer peut être, par exemple, une étiquette $x_i \in \{1, \dots, K\}$, lorsqu'on effectue une tâche de classification d'images \mathbf{y}_i en K groupes. Lorsqu'on cherche à effectuer une segmentation, telle que décrite à la Section 2.1, on cherche à assigner une étiquette à *chaque* pixel. Ce problème peut alors s'interpréter comme une classification effectuée pixel par pixel, de sorte que si $\mathbf{y}_i \in \mathbb{R}^{N_1 \times N_2}$ alors $\mathbf{x}_i \in \{1, \dots, K\}^{N_1 \times N_2}$. De nombreuses autres tâches donnent lieu à une grande variété dans le type de données traitées. La puissance des réseaux de neurones réside dans leur grande adaptabilité à des données non seulement riches, mais également de structure complexe. Nous nous concentrons dans la suite sur une base de données constituées d'images en niveaux de gris.

2.4.1.2 Réseau de neurones

Les réseaux de neurones forment une classe particulière, très riche, d'estimateurs *paramétriques* $\hat{\mathbf{x}}(\mathbf{y}; \Theta)$, qui sont largement plébiscités pour leur grande adaptabilité.

Un neurone élémentaire, ou *neurone formel*, est une fonction paramétrée par un vecteur de poids $\mathbf{w} \in \mathbb{R}^N$, un biais $b \in \mathbb{R}$ et une fonction non linéaire $\sigma : \mathbb{R} \rightarrow \mathbb{R}$. Le neurone prend en entrée un vecteur $\mathbf{y}_e \in \mathbb{R}^N$ auquel il associe une sortie

$$y_s = \sigma(\langle \mathbf{w}, \mathbf{y}_e \rangle + b). \quad (2.124)$$

Le produit scalaire $\langle \mathbf{w}, \mathbf{y}_e \rangle$ extrait de l'entrée \mathbf{y}_e une information, qui dépend du vecteur de poids \mathbf{w} . Puis, la fonction non linéaire σ modélise l'*activation* du neurone, et permet, avec l'aide du biais b , de mesurer la pertinence de l'information contenue dans l'entrée \mathbf{y}_e .

L'utilisation d'un seul neurone, connu sous le nom de *perceptron* [102], permet de réaliser des tâches simples de classification binaire, néanmoins la puissance des méthodes d'apprentissage statistique construites à partir de neurones formels réside dans leur mise en réseau.

Une *couche* de neurones, numérotée par ℓ , possède J entrées et J' sorties, qui sont reliées grâce à un ensemble de $J' \times J$ vecteurs de poids $\mathbf{w}^{(\ell)} = \left(w_{j',j}^{(\ell)} \right)_{j',j}$ et de J' biais $\mathbf{b}^{(\ell)} = \left(b_{j'}^{(\ell)} \right)_{j'=1}^{J'}$ via

$$y_{j'}^{(\ell+1)} = \sigma_{j'} \left(\sum_{j=1}^J \langle w_{j',j}^{(\ell)}, y_j^{(\ell)} \rangle + b_{j'}^{(\ell)} \right) \quad (2.125)$$

reliant la sortie de la couche ℓ à l'entrée de la couche $\ell + 1$. Une illustration du principe de *couche de neurones* est présentée à la Figure 2.15.

Remarque. Le plus souvent la fonction d'activation σ est identique pour tous les neurones d'une même couche et nous nous restreindrons à ce cas dans cette thèse.

En composant un grand nombre de couches de neurones, on obtient un *réseau de neurones* $\hat{\mathbf{x}}(\mathbf{y}; \Theta)$, dont les paramètres sont l'ensemble des poids $\mathbf{w}^{(\ell)}$ et des biais $\mathbf{b}^{(\ell)}$ de tous les neurones et de toutes les couches. Ces paramètres sont stockés dans le vecteur $\Theta = \left(\mathbf{w}^{(\ell)}, \mathbf{b}^{(\ell)} \right)_{\ell=1}^L$.

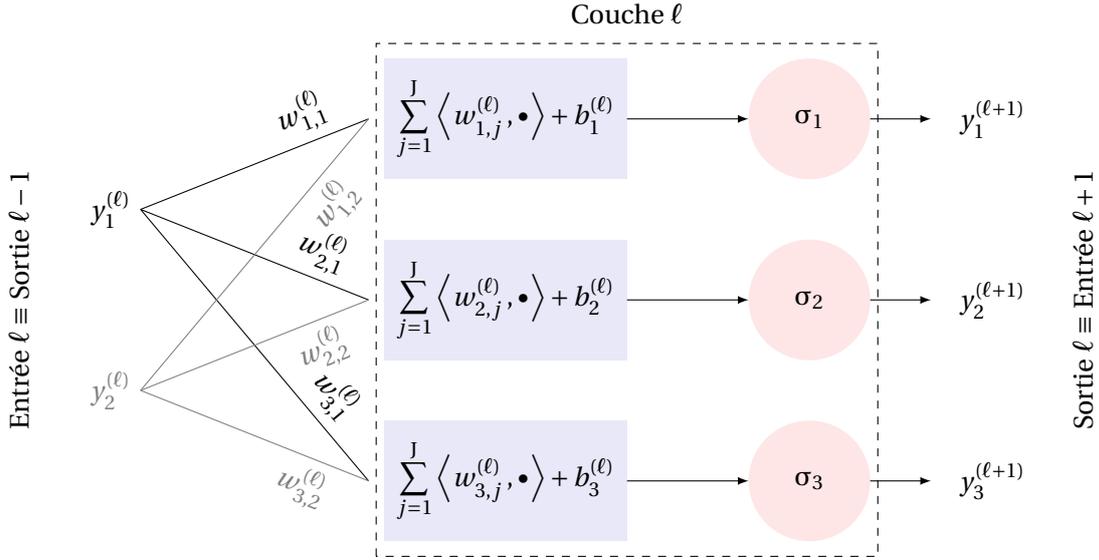


FIGURE 2.15 – Couche de neurones à 2 entrées et 3 sorties, selon la définition de l'Équation (2.125).

2.4.1.3 Entraînement

À partir de la base d'entraînement $(\mathbf{x}_i, \mathbf{y}_i)_{i=1}^I$, on construit une *fonction de perte* \mathcal{C} , qui mesure la pertinence moyenne de l'estimateur $\hat{\mathbf{x}}(\mathbf{y}; \Theta)$, pour une valeur des paramètres Θ donnée, sur les échantillons disponibles. La phase d'entraînement consiste à minimiser cette fonction par rapport à Θ , l'ensemble des poids du réseau, pour déterminer le meilleur estimateur possible $\hat{\mathbf{x}}(\mathbf{y}; \Theta^\dagger)$. On recherche alors le meilleur paramétrage :

$$\Theta^\dagger = \underset{\Theta}{\operatorname{argmin}} \mathcal{C} \left(\Theta \mid (\mathbf{x}_i, \mathbf{y}_i)_{i=1}^I \right). \quad (2.126)$$

La fonction de perte \mathcal{C} est la somme des erreurs d'estimation de \mathbf{x}_i par $\hat{\mathbf{x}}(\mathbf{y}_i; \Theta)$ et peut s'écrire sous la forme

$$\mathcal{C} \left(\Theta \mid (\mathbf{x}_i, \mathbf{y}_i)_{i=1}^I \right) = \sum_{i=1}^I c(\mathbf{x}_i, \hat{\mathbf{x}}(\mathbf{y}_i; \Theta)), \quad (2.127)$$

pour une fonction de perte *élémentaire* c . Une grande variété de fonctions c sont utilisées en apprentissage statistique, afin de s'adapter au problème considéré. À titre d'exemple, dans le cas d'un problème de classification binaire, on cherche à prédire une étiquette $\mathbf{x} \in \{0, 1\}$, et le réseau de neurones fournit en sortie une estimée $\hat{\mathbf{x}}(\mathbf{y}; \Theta) \in [0, 1]$. C'est alors l'*entropie croisée* qui mesure de façon idoine l'erreur *via*

$$c(\mathbf{x}, \hat{\mathbf{x}}) = -(\mathbf{x} \log(\hat{\mathbf{x}}) + (1 - \mathbf{x}) \log(1 - \hat{\mathbf{x}})). \quad (2.128)$$

Nous verrons d'autres fonctions de perte élémentaires dans la suite.

La manière la plus simple pour réaliser la phase d'entraînement (2.126) est d'utiliser un algorithme de descente de gradient. La mise à jour d'un paramètre θ à l'itération t s'effectue alors grâce à une étape de descente de la forme

$$\theta^{[t+1]} = \theta^{[t]} - \gamma \frac{\partial \mathcal{C}}{\partial \theta} \left(\Theta^{[t]} \mid (\mathbf{x}_i, \mathbf{y}_i)_{i=1}^I \right), \quad (2.129)$$

où $\partial \mathcal{C} / \partial \Theta$ désigne le gradient de la fonction de perte par rapports aux poids du réseau et $\gamma > 0$ est le pas de descente de l'algorithme, appelé « vitesse d'apprentissage ». Néanmoins, en pratique, (2.126) est un problème à la fois mathématiquement difficile et numériquement coûteux pour

plusieurs raisons et (2.129) est insuffisant. En effet, d'une part, la variable Θ , contenant les poids et biais du réseau, sur laquelle est réalisée la minimisation est de très grande dimension, ce qui implique des coûts en temps de calcul et en mémoire disponibles importants. D'autre part, afin d'obtenir le meilleur estimateur possible $\hat{\mathbf{x}}(\mathbf{y}; \Theta)$, la taille de la base de données $(\mathbf{x}_i, \mathbf{y}_i)_{i=1}^I$ doit être conséquente, ce qui rend extrêmement coûteuse l'évaluation de la fonctionnelle \mathcal{L} (2.127) complète, ainsi que le calcul de son gradient par rapport à la variable Θ . Pour dépasser ces difficultés, la minimisation (2.126) nécessite l'usage d'algorithmes de gradient stochastiques [12, 34, 103], faisant l'objet d'une recherche très active [67, 99], couplés à des méthodes de rétro-propagation des gradients au travers des (nombreuses) couches du réseau de neurones [60].

Pour les réseaux profonds, un dernier obstacle provient directement de la structure même de l'estimateur $\hat{\mathbf{x}}(\mathbf{y}; \Theta)$ qui est construit comme la composition d'un très grand nombre de fonctions. Le calcul du gradient nécessite une rétro-propagation des gradients, remontant depuis la sortie du réseau jusqu'à l'entrée. Ainsi, le passage de la couche ℓ à la couche $\ell + 1$ se fait *via*

$$\mathbf{y}^{(\ell+1)} = \sigma(\langle \mathbf{w}^{(\ell)}, \mathbf{y}^{(\ell)} \rangle + \mathbf{b}^\ell), \quad (2.130)$$

en notant de manière compacte la couche ℓ définie à l'Équation (2.125). Le gradient par rapport à un paramètre θ d'une couche *postérieure* à $\ell + 1$ se rétro-propage de la couche $\ell + 1$ à la couche ℓ par

$$\frac{\partial y_{j'}^{(\ell+1)}}{\partial \theta} = \left\langle \mathbf{w}_{j'}^{(\ell)}, \frac{\partial \mathbf{y}^{(\ell)}}{\partial \theta} \right\rangle \sigma'(\langle \mathbf{w}_{j'}^{(\ell)}, \mathbf{y}^{(\ell)} \rangle + b_{j'}^\ell). \quad (2.131)$$

Or, si le neurone considéré est *saturé*, c'est-à-dire si $\langle \mathbf{w}_{j'}^{(\ell)}, \mathbf{y}^{(\ell)} \rangle + b_{j'}^\ell$ se trouve dans une zone où la fonction d'activation σ est plate, alors $\sigma'(\langle \mathbf{w}_{j'}^{(\ell)}, \mathbf{y}^{(\ell)} \rangle + b_{j'}^\ell) \ll 1$. Une illustration pour quelques fonctions d'activation classiques est proposée en Figure 2.16. Par conséquent, si la rétro-propagation rencontre de nombreux neurones saturés, un grand nombre de dérivées inférieures à 1 sont multipliées et le gradient devient de plus en plus faible. Ce phénomène s'appelle le *gradient évanescent* et constitue un obstacle majeur et intrinsèque dans l'entraînement des réseaux profonds.

Nous allons voir que, dans le cas des images, on peut s'inspirer des méthodes de filtrage pour construire des couches de neurones adaptées à la topologie des données et réduire ainsi le nombre de neurones par couche tout en augmentant leur représentativité. De la sorte, le problème de gradient évanescent est atténué.

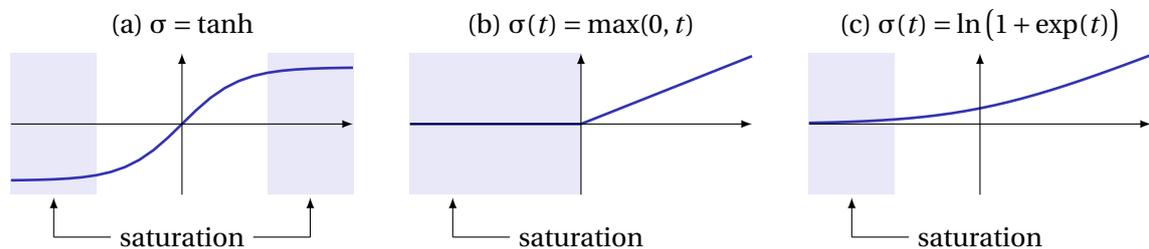


FIGURE 2.16 – Saturation de la fonction d'activation d'un neurone $\sigma(\langle \mathbf{w}, \mathbf{y} \rangle + b)$ pour une fonction activation (a) tangente hyperbolique, (b) ReLU (*Rectified Linear Unit*) et (c) ReLU lissée, ou *Softplus*.

2.4.2 Réseaux de neurones convolutionnels

On parle de couche « complètement connectée » lorsque chaque unité de l'entrée \mathbf{y}_e est reliée à toutes les unités de la sortie. Ce type d'architecture naïve est difficile à gérer. En effet, d'une part le nombre de poids à apprendre est très élevé, et, d'autre part, elle souffre du problème du gradient

évanescent (voir la fin de la Section 2.4.1 et la Figure 2.16), qui diminue drastiquement l'efficacité de la descente de gradient (2.129).

Pour les tâches d'apprentissage sur des images, on tire parti de la topologie des données d'entrée pour construire des architectures plus adaptées grâce à des *couches convolutionnelles*. Ce type d'architectures s'inspire du traitement d'image classique qui fait massivement usage de la convolution par des *filtres locaux*, construits par les utilisateurs en fonction de la tâche à réaliser, dont un exemple très populaire est la famille des filtres de Gabor, définis par la Formule (2.75). À la différence du traitement d'image classique où les filtres sont fixés en amont de l'analyse, dans le cas des réseaux de neurones, ces filtres sont appris lors de la phase d'entraînement.

Dans la pratique, pour une couche de neurones

$$\mathbf{y}_s = \sigma(\langle \mathbf{w}, \mathbf{y}_e \rangle + \mathbf{b}). \quad (2.132)$$

dont l'entrée $\mathbf{y}_e \in \mathbb{R}^{N_1 \times N_2}$ est une image (ou une carte d'attributs), au lieu d'utiliser un vecteur de poids \mathbf{w} quelconque à $N_1 N_2$ coefficients, on considère un filtre \mathbf{h} de taille fixée, faible devant la résolution de \mathbf{y}_e . Ce filtre est déplacé sur toute l'image, comme illustré par la Figure 2.18(b) pour deux filtres, l'un de taille 3×3 (en rouge), l'autre de taille 2×2 (en bleu). On obtient alors des cartes d'attributs locaux, telles que celles des Figures 2.18(c) et (d).

Définition 2.13 (Décalage (*stride*)). Le filtre \mathbf{h} parcourt l'image d'entrée \mathbf{y}_e en se déplaçant verticalement et horizontalement en faisant des pas de taille s , qu'on appelle le décalage (*stride* en anglais). Lorsque $s = 1$, les cartes d'attributs $\mathbf{h} * \mathbf{y}_e$ sont de même résolution que l'image d'entrée. C'est le cas dans l'illustration de la Figure 2.18. Néanmoins, il est courant, afin de limiter la redondance des cartes d'attributs, de choisir des décalages $s = 2, 3$. Dans ce cas la résolution de la carte en sortie $\mathbf{h} * \mathbf{y}_e$ est diminuée d'un facteur s selon chaque direction.

Ainsi, si on note \mathbf{z} la convolution de \mathbf{y}_e par le filtre \mathbf{h} avec un décalage s

$$z_{i,j} = \sum_{i',j'} \mathbf{h}_{i',j'}(\mathbf{y}_e)_{si-i',sj-j'}, \quad (2.133)$$

où la somme porte sur tous les indices (i', j') des éléments du filtre \mathbf{h} . En sortie d'une couche convolutionnelle à un filtre, on obtient donc une carte \mathbf{y}_s contenant les réponses locales de l'image au filtre \mathbf{h} auxquelles on applique pixel par pixel un biais et une fonction d'activation :

$$\mathbf{y}_s = \sigma(\mathbf{h} * \mathbf{y}_e + \mathbf{b}). \quad (2.134)$$

Afin d'illustrer la notion de filtrage d'une image avec décalage, on considère à la Figure 2.17 une image $\mathbf{y}_e \in \mathbb{R}^{4 \times 4}$ de résolution 4×4 et $\mathbf{h} \in \mathbb{R}^{2 \times 2}$ un filtre de taille 2×2 et l'opérateur linéaire qui associe à la carte \mathbf{y}_e sa convolution \mathbf{z} par le filtre \mathbf{h} avec un décalage (*stride*, voir Définition 2.13) de deux pixels qui vérifie

$$\begin{aligned} z_{1,1} &= \mathbf{h}_{1,1}(\mathbf{y}_e)_{1,1} + \mathbf{h}_{1,2}(\mathbf{y}_e)_{1,2} + \mathbf{h}_{2,1}(\mathbf{y}_e)_{2,1} + \mathbf{h}_{2,2}(\mathbf{y}_e)_{2,2} \\ z_{1,2} &= \mathbf{h}_{1,1}(\mathbf{y}_e)_{1,3} + \mathbf{h}_{1,2}(\mathbf{y}_e)_{1,4} + \mathbf{h}_{2,1}(\mathbf{y}_e)_{2,3} + \mathbf{h}_{2,2}(\mathbf{y}_e)_{2,4} \\ z_{2,1} &= \mathbf{h}_{1,1}(\mathbf{y}_e)_{3,1} + \mathbf{h}_{1,2}(\mathbf{y}_e)_{3,2} + \mathbf{h}_{2,1}(\mathbf{y}_e)_{4,1} + \mathbf{h}_{2,2}(\mathbf{y}_e)_{4,2} \\ z_{2,2} &= \mathbf{h}_{1,1}(\mathbf{y}_e)_{3,3} + \mathbf{h}_{1,2}(\mathbf{y}_e)_{3,4} + \mathbf{h}_{2,1}(\mathbf{y}_e)_{4,3} + \mathbf{h}_{2,2}(\mathbf{y}_e)_{4,4}. \end{aligned}$$

Ainsi, le nombre de poids à apprendre, i.e., le nombre de coefficients de \mathbf{h} , est grandement réduit par rapport à une couche complètement connectée qui possède $N_1 N_2$ poids à apprendre, ce qui permet de diminuer l'évanescence du gradient.

Remarque (Normalisation par paquets (*batch normalization*)). Afin de réduire encore le phénomène du gradient évanescent, il est possible de renormaliser à chaque couche les données par

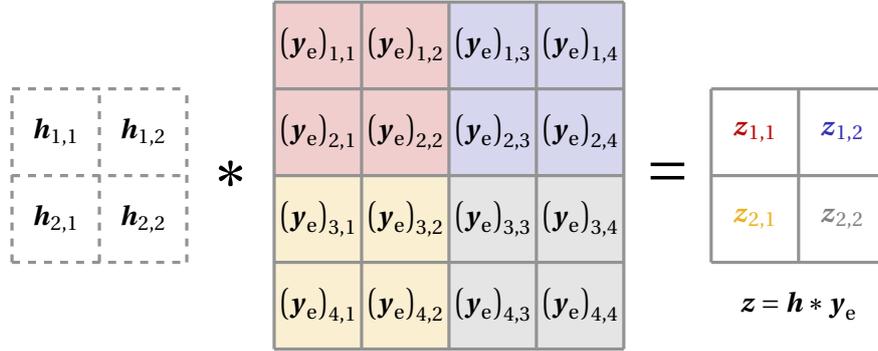


FIGURE 2.17 – Filtrage d’une image de résolution 4×4 par un filtre de taille 2×2 avec un décalage de deux pixels, donc sans recouvrement.

« paquet », un paquet (*batch*) désignant un groupe de pixels ou d’attributs d’effectif fixé. En notant \mathcal{B} un paquet, et x une donnée de \mathcal{B} , la renormalisation consiste à effectuer l’opération (non-linéaire)

$$x \leftarrow \frac{x - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2}} \quad (2.135)$$

où $\mu_{\mathcal{B}}$ désigne la moyenne des éléments du paquet et $\sigma_{\mathcal{B}}^2$ leur variance.

En pratique, afin d’extraire simultanément plusieurs informations, on considère des couches convolutionnelles à plusieurs filtres $\mathbf{h}^{[t]}$, qui sont appliqués en parallèle à l’entrée et on obtient alors des cartes d’attributs en sortie $\mathbf{y}_s \in \mathbb{R}^{N'_1 \times N'_2 \times T}$ à plusieurs canaux. Les couches convolutionnelles sont ensuite empilées pour former un réseau. Ainsi, si nous avons présenté le principe d’une couche convolutionnelle pour le cas d’une entrée $\mathbf{y}_e \in \mathbb{R}^{N_1 \times N_2}$ à un seul canal, la définition ci-dessous est plus générale et englobe le cas de cartes à plusieurs canaux $\mathbf{y}_e \in \mathbb{R}^{N_1 \times N_2 \times T}$.

Définition 2.14 (Couche convolutionnelle). Une couche de convolution relie une entrée \mathbf{y}_e à T sorties au moyen de T filtres $(\mathbf{h}^{[t]})_{t=1}^T$ (aussi appelés noyaux de convolution), de T biais $(\mathbf{b}_t)_{t=1}^T$ et d’une fonction d’activation :

$$\mathbf{y}_s^{[t]} = \sigma(\mathbf{h}^{[t]} * \mathbf{y}_e + \mathbf{b}_t). \quad (2.136)$$

Chaque carte $\mathbf{y}_s^{[t]}$ en sortie correspond à la carte d’attributs mesurés par le filtre $\mathbf{h}^{[t]}$. Dans le cas d’une entrée $\mathbf{y}_e \in \mathbb{R}^{N_1 \times N_2 \times d}$ à d canaux (par exemple une image couleur à $d = 3$ canaux et de manière plus générale une carte à d attributs), le filtre $\mathbf{h}^{[t]}$ à apprendre possède non seulement une extension spatiale donnée mais également une profondeur, égale à d . La convolution $\mathbf{h}^{[t]} * \mathbf{y}_e$ est effectuée spatialement et au travers des canaux. La carte $\mathbf{y}_s^{[t]}$ en sortie est obtenue en se déplaçant *uniquement spatialement*. Il s’agit donc d’une convolution *mixte*, au sens où les filtres sont tridimensionnels mais leur déplacement s’effectue en deux dimensions.

Pour éviter que le nombre d’attributs transportés de couche en couche n’augmente trop fortement, la résolution des cartes $\mathbf{y}_s^{[t]}$ est diminuée grâce à une étape de « mise en commun » ou *pooling*.

Définition 2.15 (Mise en commun (*pooling*)). Une couche de mise en commun consiste, pour chaque carte d’attributs $\mathbf{y}_s^{[t]}$, à ne conserver qu’un nombre limité d’unité. Pour cela, l’image, ou la carte d’attributs, considérée est découpée en cellules régulières. Puis, au sein de chaque cellule, un attribut représentatif est calculé et conservé. La mise en commun peut s’effectuer de différentes manières : en conservant l’attribut maximum de la cellule (noté MaxPool et illustré à la Figure 2.18) ou en effectuant la moyenne des attributs de la cellule.

Dans l'exemple de la Figure 2.18 une mise en commun par maximum est effectuée sur des cellules de 4×4 pixels.

L'ensemble de ces étapes, présentées à la Figure 2.18, constitue un *bloc de convolution*, dont les poids à apprendre sont les coefficients des filtres $(\mathbf{h}^{[l]})_{l=1}^T$ et les biais $(\mathbf{b}_1, \dots, \mathbf{b}_T)$. Pour la description des réseaux profonds, nous adopterons une version simplifiée de la Figure 2.18 afin de représenter de manière compacte des réseaux de neurones convolutionnels à plusieurs couches. Dans cette optique, la Figure 2.19 représente un *bloc de convolution* schématisé, composé d'opérations élémentaires : convolution, activation et mise en commun (*max pooling*).

Définition 2.16 (Réseau de neurone convolutionnel (complètement connecté)). Un *réseau de neurone convolutionnel* est constitué de la succession de blocs de convolution. On dit que ce réseau est complètement connecté lorsqu'au moins une des couches, le plus souvent la ou les dernière(s) couche(s), du réseau est une couche « complètement connectée », telle que décrite en début de Section 2.4.2.

2.4.3 De la classification à la segmentation d'images par réseaux de neurones

2.4.3.1 Classification

La première tâche à avoir été réalisée à l'aide de réseaux de neurones convolutionnels est la classification d'images en k groupes. La base d'apprentissage est alors un ensemble d'images, chacune associée à une étiquette, comprise entre 1 et k indiquant à quelle classe cette image appartient. Un réseau de neurones pour la classification est une fonction

$$\hat{x}(\cdot; \Theta) = \begin{cases} \mathbb{R}^{N_1 \times N_2 \times d} & \longrightarrow [0, 1]^k \\ \mathbf{y} & \longrightarrow (\hat{p}_1, \dots, \hat{p}_k) \end{cases} \quad (2.137)$$

qui associe à une image (en niveau de gris, $d = 1$ ou en couleurs, $d = 3$) un vecteur à k composantes $(\hat{p}_1, \dots, \hat{p}_k)$ où \hat{p}_i est la probabilité que l'image \mathbf{y} appartienne à la classe i et qui vérifie donc

$$\sum_{i=1}^k \hat{p}_i = 1. \quad (2.138)$$

Parmi les réseaux convolutionnels profonds pour la classification d'image, quelques exemples très connus sont AlexNet [69], VGG net [109] et GoogLeNet [114]. Chacun de ces réseaux possède des propriétés différentes, néanmoins tous partagent une même architecture générale, qui peut être résumée par la Figure 2.20. L'entrée est d'abord analysée en passant au travers d'une succession de couches convolutionnelles (voir Définition 2.14). Puis, le réseau se termine par une ou plusieurs couches complètement connectées, éventuellement composées avec une (des) fonction(s) d'activation non-linéaire(s). Enfin, la dernière de ces couches complètement connectées encode la décision finale, c'est-à-dire le calcul du vecteur des probabilités \mathbf{p} introduit à l'Équation (2.137). Afin de présenter la méthodologie générale sous-jacente à ces approches, nous nous appuyerons sur le schéma de réseau de neurones convolutionnel pour la classification d'images proposé en Figure 2.20.

Tout réseau de neurone convolutionnel commence par une couche de convolution, suivie d'une activation, le plus souvent une correction ReLU. Cet enchaînement est éventuellement répété, puis complété par une étape de mise en commun (*pooling*) induisant une réduction de la résolution des cartes d'attributs, pour former un bloc de convolution, sur le modèle de la Figure 2.19. Plusieurs blocs de convolutions sont assemblés pour former une première partie du réseau qui extrait toute une hiérarchie d'attributs.

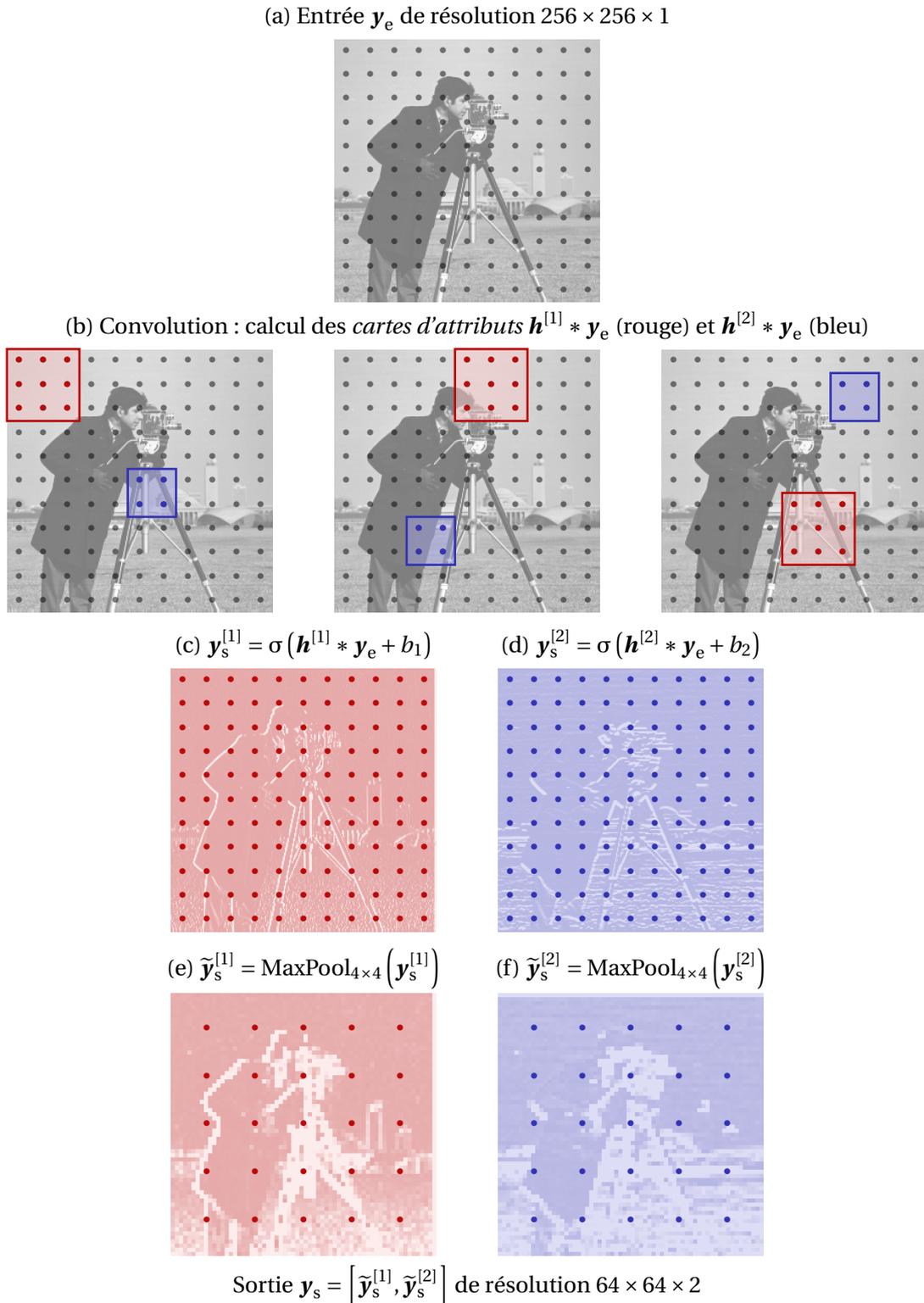


FIGURE 2.18 – Exemple détaillé d'une couche convolutionnelle, ou *bloc de convolution*, en apprentissage statistique pour le traitement d'images. Les • schématisent les pixels des images et des cartes d'attributs présentées. La figure se lit du haut vers le bas.

Comme on l'observe sur la Figure 2.20, la succession des blocs de convolution conduit à une diminution progressive de la résolution des cartes d'attributs. Ainsi, les couches superficielles permettent de mesurer des attributs locaux, i.e., correspondant aux petites échelles sur une image, et « simples », au sens où ils peuvent être extraits en peu d'opérations. Au contraire, les couches les plus profondes mesurent des quantités beaucoup plus riches et complexes, construites à partir de

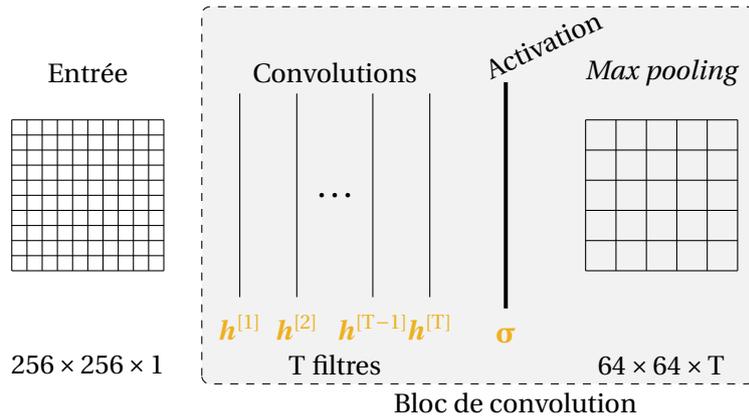


FIGURE 2.19 – Bloc de convolution schématique. Les filtres $h^{[l]}$ et la fonction d'activation σ , écrits en jaune, n'apparaîtront pas explicitement dans la représentation des réseaux profonds. La figure se lit de la gauche vers la droite.

combinaisons hiérarchisées des attributs des couches précédentes, qui sont très peu localisées. En effet, au fur et à mesure que l'on « descend » dans les couches du réseau, on accède à des attributs qui combinent l'information d'un plus grand nombre de pixels de l'image d'entrée : on dit que leur « champ de réceptivité » est plus grand.

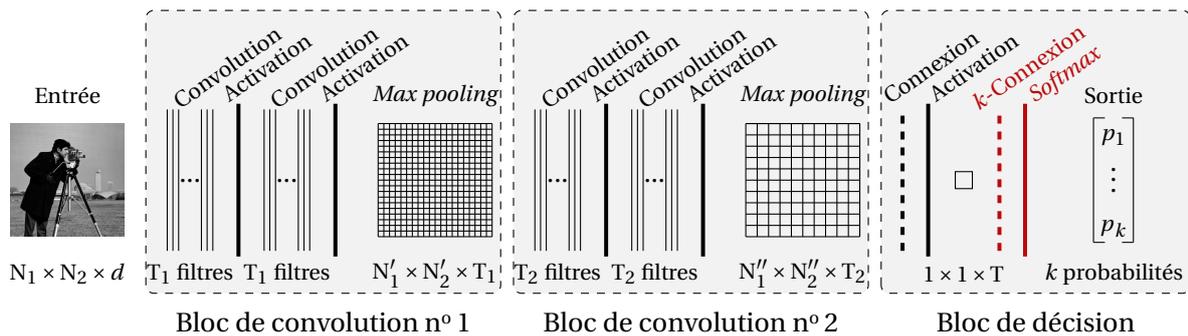


FIGURE 2.20 – Réseau de neurones convolutionnel pour la classification d'images en k classes, composé de deux blocs de convolution suivi de deux couches complètement connectées formant le bloc de décision. L'appellation « connexion » désigne une couche complètement connectée, telle que définie en début de Section 2.4.2. La notation « k -connexion » désigne une couche complètement connectée à k sorties.

Tous les attributs extraits par les blocs de convolution successifs sont ensuite réunis par des couches complètement connectées, appelées « connexions » sur la Figure 2.20, constituant la deuxième partie du réseau, que nous appelons « bloc de décision ». L'ultime couche du réseau, en rouge sur la Figure 2.20, est une couche complètement connectée qui contient autant de sorties qu'il y a de classes à distinguer. Ces sorties, notées $\mathbf{y} = (y_1, \dots, y_k)$, sont ensuite converties en un vecteur de probabilité $\mathbf{p} : (p_1, \dots, p_k)$ grâce à la fonction de « maximum doux » (*softmax*).

Définition 2.17 (Maximum doux (*softmax*)). Pour $k \in \mathbb{N}^*$, la fonction $\sigma : \mathbb{R}^k \rightarrow [0, 1]^k$ définie par

$$\forall i \in \{1, \dots, k\}, \quad \sigma_i(\mathbf{y}) \triangleq \frac{\exp(y_i)}{\sum_{j=1}^k \exp(y_j)} = p_i, \quad (2.139)$$

permet de transformer un vecteur de réels quelconques en un vecteur de probabilités. Elle est appelée « maximum doux » et représente une relaxation *différentiable* de la fonction sélectionnant le maximum d'un vecteur. Par conséquent, l'entraînement d'un réseau contenant une activation *softmax* peut être entraîné grâce un algorithme de rétro-propagation du gradient.

Remarque. Nous avons mentionné plus haut que les couches complètement connectées sont à la fois coûteuses et difficiles à apprendre dans la phase d'entraînement. C'est pour cela que ces couches sont placées en toute fin de réseau, en sortie d'une couche qui contient d'autant moins de poids que les blocs de convolution successifs ont entraîné une diminution drastique de la résolution des cartes.

Enfin, on obtient, en sortie du réseau, un vecteur contenant les probabilités d'appartenance à chacune des classes et la classification en elle-même consiste à associer à l'image l'étiquette de la classe correspondant à la probabilité maximale.

La phase d'entraînement d'un réseau de neurone pour la classification est effectuée en minimisant « l'entropie croisée catégorielle », qui permet d'étendre l'entropie croisée utilisée pour la classification binaire, introduite à l'Équation (2.128), au cas d'un nombre de classes $k \geq 3$. Pour cela il est tout d'abord nécessaire de se donner un encodage approprié.

Définition 2.18 (Encodage 1 parmi k (*one-hot encoding*)). Pour un problème de classification à k classes, la base de données d'entraînement est initialement composée de couples (x, \mathbf{y}) , où $x \in \{1, \dots, k\}$ est l'étiquette de la classe à laquelle appartient l'image \mathbf{y} . Si \mathbf{y} appartient à la classe i , alors $x = i$. Afin de comparer cette étiquette à une prédiction $\hat{\mathbf{x}}(\mathbf{y}; \Theta) = (\hat{p}_1, \dots, \hat{p}_k)$, « $x = i$ » est représenté par le $i^{\text{ème}}$ vecteur de la base canonique de \mathbb{R}^k . L'image \mathbf{y} est donc associé à un vecteur de probabilité contenant un unique 1, à la $i^{\text{ème}}$ position, c'est-à-dire que le couple (x, \mathbf{y}) est remplacé par (\mathbf{p}, \mathbf{y}) avec $\mathbf{p} = (0, \dots, 0, \underset{i^{\text{ème}}}{1}, 0, \dots, 0)$ représenté à la Figure 2.21a.

On peut donc maintenant définir la fonction de perte élémentaire, qui permet de comparer les deux vecteurs de probabilités \mathbf{p} (vrai) et $\hat{\mathbf{p}}$ (estimé) tels que représentés à la Figure 2.21.

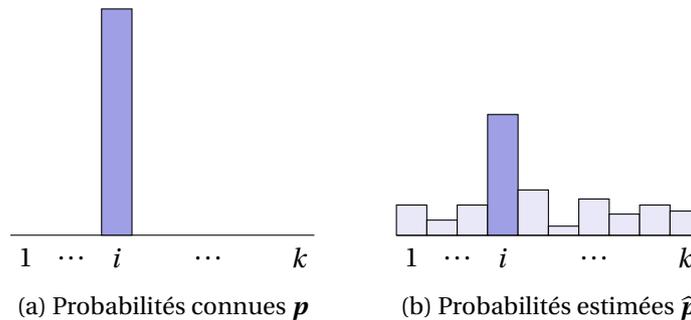


FIGURE 2.21 – Représentation des probabilités d'appartenance pour le calcul de l'erreur par entropie croisée catégorielle pour une image du jeu d'entraînement \mathbf{y} appartenant à la $i^{\text{ème}}$ classe, représentée en bleu plus foncé.

Définition 2.19 (Entropie croisée catégorielle (*categorical cross-entropy*)). Pour une donnée d'entraînement (\mathbf{p}, \mathbf{y}) , avec un encodage 1 parmi k (voir Définition 2.18), et une estimée $\hat{\mathbf{p}}$, la fonction de perte élémentaire pour le problème de classification à k catégories s'écrit

$$c(\mathbf{p}, \hat{\mathbf{p}}) = - \sum_{i=1}^k p_i \log(\hat{p}_i). \quad (2.140)$$

2.4.3.2 Segmentation

Pour une tâche de segmentation, on souhaite associer une étiquette à *chaque pixel*. Ainsi, un réseau de neurones pour la segmentation doit renvoyer en sortie une *carte* de vecteurs de probabilité $\hat{\mathbf{P}} = \{\hat{\mathbf{p}}_{\underline{n}} \in [0, 1]^k \mid \underline{n} \in \Omega\}$ où $\Omega = \{1, \dots, N_1\} \times \{1, \dots, N_2\}$ désigne l'ensemble des pixels de l'image. Plusieurs approches ont été explorées.

La méthode la plus directe est la classification *patch par patch via* un réseau convolutionnel [38, 91, 42]. Bien qu'elle fournisse des résultats pertinents, notamment pour les applications biomédicales [24, 85], elle entraîne une forte diminution de la résolution qui limite son extension à des problèmes plus généraux impliquant des images de plus faible résolution. Des stratégies plus élaborées font usage de pré- et/ou post-traitements, tels que l'augmentation des données *via* l'introduction de superpixels [38, 54], l'utilisation du réseau pour améliorer des suggestions grossières [54, 56] ou encore le raffinement des segmentations brutes grâce à des classifieurs locaux [56], des opérations de filtrage [24] ou des champs aléatoires [38].

Les limites de ces approches sont d'une part leur faible flexibilité, puis qu'elles sont pour la plupart élaborées pour répondre à un problème précis, et d'autre part leur complexité, qui les empêche de réaliser la tâche de segmentation de manière complètement supervisée. Long *et coll.* [75] ont proposé d'étendre l'architecture générale des réseaux convolutionnels pour la classification, dont un exemple est donné en Figure 2.20, au problème de segmentation pour fournir une prédiction pixel par pixel grâce à un réseau convolutionnel appris de part en part, et ne nécessitant aucun traitement supplémentaire. En effet, la succession de blocs de convolution, constituant la première partie des réseaux convolutionnels pour la classification, possède une grande capacité d'extraction d'attributs pertinents et variés, tant en terme de motifs détectés que de complexité. On tire donc avantage à s'appuyer sur les différents blocs de convolution développés pour la classification pour construire la première partie d'un réseau convolutionnel pour la segmentation.

Comme on l'a observé à la Figure 2.20, la résolution des cartes d'attributs diminue progressivement à chaque nouveau bloc. Dans un réseau de classification, les couches complètement connectées du bloc de décision renvoient uniquement des vecteurs, c'est-à-dire des cartes de taille $1 \times 1 \times T$, où T désigne le nombre de sorties de la couche complètement connectée. Or, pour la segmentation, on souhaite obtenir en sortie une carte de vecteur de probabilité, c'est-à-dire un élément de taille $N_1 \times N_2 \times k$. Par conséquent, une fois les attributs extraits grâce aux blocs de convolution successifs, une procédure de sur-échantillonnage est nécessaire. Dans ce but, Long *et coll.* [75] ont introduit de nouveaux outils, permettant, d'une part, d'obtenir en sortie du réseau une information ayant la même résolution que l'image en entrée, c'est-à-dire un vecteur de probabilité par pixel, et d'autre part de propager correctement la localisation des motifs détectés. Pour cela le bloc de décision est supprimé et remplacé par une succession de blocs qui effectuent un *sur-échantillonnage supervisé*. Pour réaliser un sur-échantillonnage, plusieurs possibilités existent, comme par exemple l'interpolation bi-linéaire ou cubique. La stratégie proposée par [75] consiste à permettre au réseau d'apprendre la meilleure manière d'effectuer le sur-échantillonnage. L'approche de [75] repose sur une interprétation élégante de l'interpolation comme une étape de convolution *transposée*, aussi appelée *déconvolution*. En effet, l'opération de convolution décrite à l'Équation (2.133) et illustrée en Figure 2.17 est linéaire, en dimension finie, elle possède donc une opération linéaire transposée. Dans l'exemple traité à la Figure 2.17 l'opération de convolution associée à une image $\mathbf{y}_e \in \mathbb{R}^{4 \times 4}$ une carte $\mathbf{z} \in \mathbb{R}^{2 \times 2}$, donc la convolution adjointe, notée $\mathbf{u} = \mathbf{h} \star \mathbf{z}$ associée à la carte $\mathbf{z} \in \mathbb{R}^{2 \times 2}$ une carte $\mathbf{u} \in \mathbb{R}^{4 \times 4}$ avec

$$\begin{aligned} \mathbf{u}_{1,1} &= \mathbf{h}_{1,1} \mathbf{z}_{1,1}, & \mathbf{u}_{1,2} &= \mathbf{h}_{1,2} \mathbf{z}_{1,1}, & \mathbf{u}_{2,1} &= \mathbf{h}_{2,1} \mathbf{z}_{1,1}, & \mathbf{u}_{2,2} &= \mathbf{h}_{2,2} \mathbf{z}_{1,1} \\ \mathbf{u}_{1,3} &= \mathbf{h}_{1,1} \mathbf{z}_{1,2}, & \mathbf{u}_{1,4} &= \mathbf{h}_{1,2} \mathbf{z}_{1,2}, & \mathbf{u}_{2,3} &= \mathbf{h}_{2,1} \mathbf{z}_{1,2}, & \mathbf{u}_{2,4} &= \mathbf{h}_{2,2} \mathbf{z}_{1,2} \\ \mathbf{u}_{3,1} &= \mathbf{h}_{1,1} \mathbf{z}_{2,1}, & \mathbf{u}_{3,2} &= \mathbf{h}_{1,2} \mathbf{z}_{2,1}, & \mathbf{u}_{4,1} &= \mathbf{h}_{2,1} \mathbf{z}_{2,1}, & \mathbf{u}_{4,2} &= \mathbf{h}_{2,2} \mathbf{z}_{2,1} \\ \mathbf{u}_{3,3} &= \mathbf{h}_{1,1} \mathbf{z}_{2,2}, & \mathbf{u}_{3,4} &= \mathbf{h}_{1,2} \mathbf{z}_{2,2}, & \mathbf{u}_{4,3} &= \mathbf{h}_{2,1} \mathbf{z}_{2,2}, & \mathbf{u}_{4,4} &= \mathbf{h}_{2,2} \mathbf{z}_{2,2}. \end{aligned}$$

Remarque. L'opération de convolution transposée peut également être vue comme une convolution dont le décalage (*stride*), serait de la forme $1/s$ avec s un entier. Cette interprétation permet, grâce à des décalages *fractionnaires*, de décrire un réseau de neurones pour la segmentation uniquement en terme d'opérations de convolution, d'activation et de mise en commun.

Si un tel réseau, composé de blocs de convolution et de blocs de déconvolution, permet d'effectuer une prédiction pixel par pixel, il n'échappe pas à la tension entre attributs globaux, de

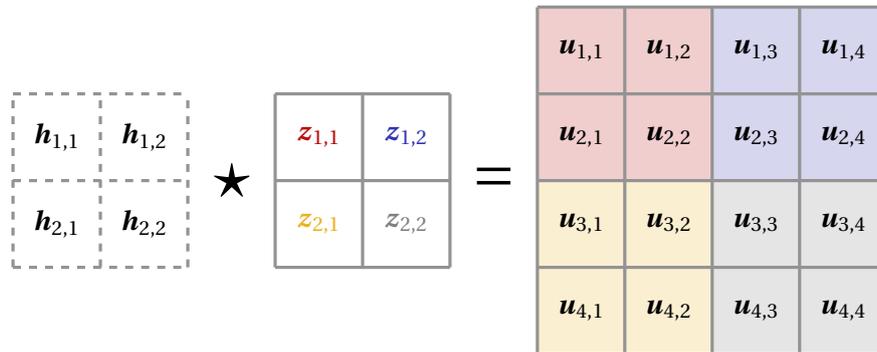


FIGURE 2.22 – Filtrage d’une image de résolution 4×4 par un filtre de taille 2×2 avec un décalage de deux pixels, donc sans recouvrement.

grand niveau de complexité, donc porteurs de sens, détectés par les couches profondes, et attributs locaux, traduisant l’apparence visuelle de l’image, encodés dans les couches superficielles. Une segmentation pertinente et précise nécessite un compromis très fin entre une représentation suffisamment riche du contenu (couches profondes) et une localisation précise (couches superficielles).

Obtenir des contours suffisamment résolus requiert de conserver, à chaque couche du réseau, de l’information localisée, provenant donc des couches superficielles. Des « connexions résiduelles » (*skip connections*) [11] sont construites pour transporter sans altérations les attributs locaux au travers d’un réseau profond. De la sorte, il est possible d’effectuer une segmentation à une résolution fine tout en respectant la structure globale de l’image d’entrée.

En pratique, le sur-échantillonnage s’effectue par étapes pour inclure progressivement des informations des couches superficielles *via* les connexions résiduelles. Illustrons cela sur l’exemple de la Figure 2.23. D’une part, le facteur de sur-échantillonnage (ou de manière équivalente le décalage fractionnaire) de la couche de convolution transposée est choisi de sorte qu’en sortie du bloc de déconvolution on obtienne une carte de la même résolution que la sortie du bloc de convolution n° 2, de profondeur k , le nombre de classes. D’autre part, on applique une convolution à k filtres à la sortie de ce bloc afin d’obtenir une carte de profondeur k également. Puis ces cartes sont fusionnées, par exemple en effectuant une somme pixel par pixel, canal par canal, notée Σ . Il en résulte une carte d’attributs fusionnés, qui est finalement déconvoluée une dernière fois pour obtenir une prédiction à la résolution initiale.

Remarque. Il est possible d’inclure plusieurs connexions résiduelles. Dans notre exemple de la Figure 2.23, il faudrait alors sur-échantillonner la sortie de la fusion Σ pour obtenir une carte de taille $N'_1 \times N'_2$, convoluer la sortie du bloc n° 1 par k filtres puis réaliser une seconde fusion Σ' . Enfin une convolution transposée à k filtres permettrait d’obtenir une segmentation à la résolution $N_1 \times N_2$, formulée en terme de probabilités d’appartenance.

Un tel réseau convolutionnel pour la segmentation a alors l’avantage de pouvoir être entraîné entièrement, *via* un algorithme de gradient avec rétro-propagation.

Grâce à sa structure profonde et son caractère complètement supervisé, le réseau proposé par Long *et coll.* [75] s’adapte très bien au jeu de données considéré. De plus, ces blocs de convolution étant similaires à ceux d’un réseau pour la classification, tel que celui de la Figure 2.20, il est facile de transférer les attributs appris par un réseau de classification antérieur vers un réseau de segmentation. L’entraînement porte alors uniquement sur la seconde partie du réseau et s’avère donc moins coûteux.

Suite aux travaux de Long *et coll.* [75], d’autres architectures ont été proposées, comme par

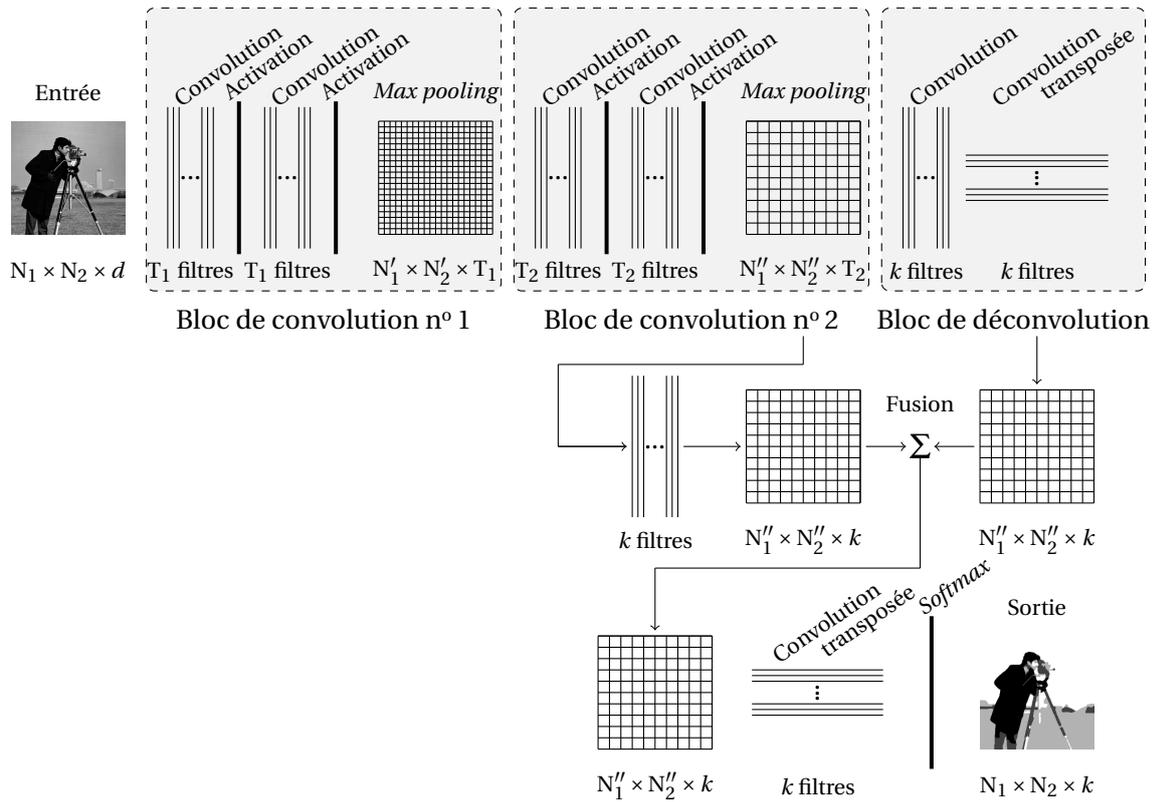


FIGURE 2.23 – Réseau de neurones convolutionnel pour la segmentation d’images à k classes, composé de deux blocs de convolution suivi d’un bloc de déconvolution.

exemple le réseau U-net pour la segmentation d’images médicales [101] qui permet de conserver beaucoup d’information localisée et contextualisée tout le long du réseau grâce à l’utilisation d’un grand nombre d’attributs dans la phase de sur-échantillonnage. Tirant également avantage de structures qui réduisent d’abord la résolution pour extraire les attributs discriminants, puis reviennent dans un second temps à la résolution initiale par sur-échantillonnage supervisé, les réseaux de neurones génératifs adverses (GAN, *Generative Adversarial Networks*) représentent une alternative pour la segmentation [76]. Si leur architecture évoque celle des réseaux précédemment décrit leur entraînement est drastiquement différent.

2.4.3.3 Apprentissage profond pour les textures

Grâce à leur capacité à saisir des attributs de complexités et d’échelles extrêmement variés les réseaux de neurones convolutifs profonds sont particulièrement adaptés à la segmentation de texture. En effet, les couches convolution peuvent être rapprochées des bases de filtres de Gabor, décrits à la Section 2.2.3. Le caractère multi-échelle induit par la succession des blocs et les étapes de sous-échantillonnage par mise en commun rappelle la décomposition en ondelettes, introduite à la Section 2.2.3.2 pour la mesure d’attributs fractals, et les activations non-linéaire évoquent la construction des coefficients d’ondelettes dominants à l’Équation (2.83).

Quelques précautions sont néanmoins indispensables pour obtenir des segmentations de texture de qualité. En effet, la décision de segmentation est prise en sortie des couches les plus profondes du réseau, qui correspondent donc à des attributs sophistiqués mais dont la localisation est mauvaise. Les premiers réseaux convolutionnels pour la segmentation se sont massivement concentrés sur la reconnaissance de formes complexes, par exemple un objet ou un visage, encodées dans les couches profondes. Ainsi, malgré l’amélioration de la localisation apportée par l’utilisation de connexions résiduelles, ces réseaux réalisent en grande partie leur segmentation sur des attributs profonds, et donc de grande extension. On le voit dans l’exemple de la Figure 2.23, la

connexion résiduelle est réalisée avec le bloc médian, contenant des attributs déjà complexes et moins localisés.

Si on considère l'exemple d'une texture AM-FM, ces attributs profonds peuvent par exemple être des propriétés de basses fréquences, qui nécessitent la prise en compte d'un nombre élevé de pixels et ne sont par conséquent ni bien localisées ni discriminantes.

Afin d'adapter le réseau convolutionnel de Long *et coll.* [75], les auteurs de [4] proposent de laisser de côté l'analyse de forme pour se concentrer sur les attributs d'apparence, encodés dans les couches médianes et superficielles. La profondeur du réseau est donc réduite, et les connexions résiduelles sont effectuées à partir de couches plus superficielles afin de capter les caractéristiques locales des textures, qui sont les plus discriminantes, surtout à la frontière entre deux textures. La Figure 2.24 illustre (en rouge) la modification du réseau de segmentation classique de la Figure 2.23, permettant son adaptation à un problème de segmentation de texture.

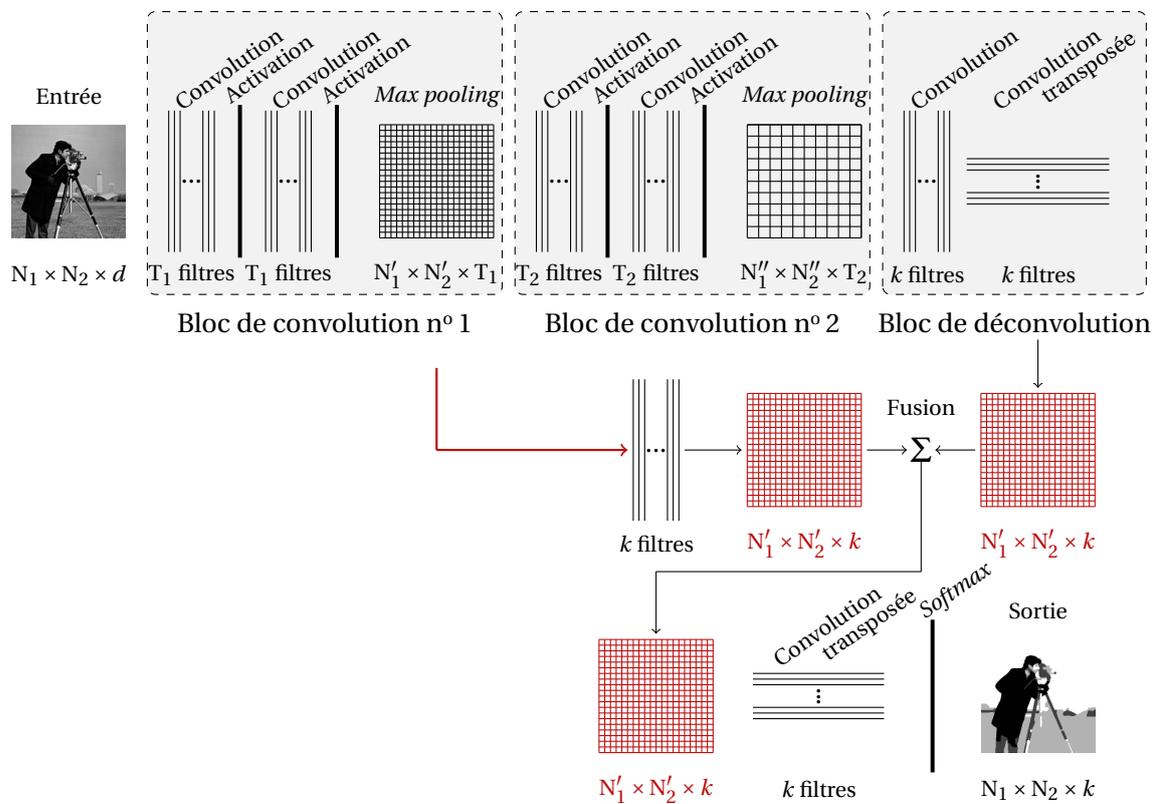


FIGURE 2.24 – Réseau de neurones convolutionnel pour la segmentation d'images texturées constituées de k textures, composé de deux blocs de convolution suivi d'un bloc de déconvolution sur le modèle de [75].

Références

- [1] G. Alberti, G. Bouchitté, and G. Dal Maso. The calibration method for the Mumford–Shah functional and free-discontinuity problems. *Calc. Var. Partial Differ. Equ.*, 16(3) :299–333, 2003. 16
- [2] L. Ambrosio and V. M. Tortorelli. Approximation of functional depending on jumps by elliptic functional via Γ -convergence. *Comm. Pure Appl. Math*, 43(8) :999–1036, 1990. 16
- [3] L. Ambrosio, N. Fusco, and D. Pallara. *Functions of bounded variation and free discontinuity problems*, volume 254. Clarendon Press Oxford, 2000. 19

- [4] V. Andrearczyk and P. Whelan. Texture segmentation with fully convolutional networks. *preprint arXiv :1703.05230*, 2017. 59
- [5] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Match. Int.*, 33(5) :898–916, 2011. 43, 44
- [6] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *J. Royal Stat. Soc. : Ser. B (Methodol.)*, 36(2) :192–225, 1974. 13
- [7] S. Beucher. *Segmentation d'images et morphologie mathématique*. PhD thesis, 1990. 26
- [8] S. Beucher. The watershed transformation applied to image segmentation. *Scanning microscopy supplement*, pages 299–299, 1992. 26
- [9] S. Beucher and F. Meyer. The morphological approach to segmentation : the watershed transformation. *Mathematical Morphology in image processing*, 34 :433–481, 1993. 26
- [10] J. C. Bezdek. *Pattern recognition with fuzzy objective function algorithms*. Springer Science & Business Media, 2013. 25
- [11] C. M. Bishop. *Pattern recognition and machine learning*. springer, 2006. 57
- [12] L. Bottou and O. Bousquet. The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems*, pages 161–168, 2008. 49
- [13] G. Bouchitté. Recent convexity arguments in the calculus of variations. *Lecture notes from the 3rd int. summer school on the calculus of variations*, 1998. 16
- [14] R. Burkard, M. Dell'Amico, and S. Martello. *Assignment problems, revised reprint*, volume 106. SIAM, 2012. 39
- [15] J.-F. Cai, B. Dong, S. Osher, and Z. Shen. Image restoration : total variation, wavelet frames, and beyond. *J. Am. Math. Soc.*, 25(4) :1033–1089, 2012. 12
- [16] X. Cai and G. Steidl. Multiclass segmentation by iterated ROF thresholding. In *Int. Workshop on Energy Minimization Methods in Comp. Vis. and Pat. Rec.*, pages 237–250. Springer, 2013. 20, 46
- [17] V. Caselles, A. Chambolle, and M. Novaga. The discontinuity set of solutions of the tv denoising problem and some extensions. *Multiscale Model. Simul.*, 6(3) :879–894, 2007. 20
- [18] A. Chambolle, V. Caselles, D. Cremers, M. Novaga, and T. Pock. An introduction to total variation for image analysis. *Theoretical foundations and numerical methods for sparse recovery*, 9(263-340) :227, 2010. 20
- [19] A. Chambolle, D. Cremers, and T. Pock. A convex approach to minimal partitions. *SIAM J. Imaging Sci.*, 5(4) :1113–1158, 2012. 18, 19, 20, 21
- [20] T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Trans. Image Process.*, 10(2) : 266–277, 2001. 16
- [21] T. F. Chan, S. Esedoglu, and M. Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM J. Appl. Math.*, 66(5) :1632–1648, 2006. 16
- [22] B. B. Chaudhuri and N. Sarkar. Texture segmentation using fractal dimension. *IEEE Trans. Pattern Anal. Match. Int.*, 17(1) :72–77, 1995. 28
- [23] C. Chaux, L. Duval, A. Benazza-Benyahia, and J.-C. Pesquet. A nonlinear Stein-based estimator for multichannel image denoising. *IEEE Trans. Signal Process.*, 56(8) :3855–3870, 2008. 23

- [24] D. Cirean, A. Giusti, L. M. Gambardella, and J. Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In *Advances in neural information processing systems*, pages 2843–2851, 2012. 56
- [25] D. A. Clausi and M. E. Jernigan. Designing Gabor filters for optimal texture separability. *Pattern Recognit.*, 33(11) :1835–1849, 2000. 32
- [26] S. Cohen and J. Istas. *Fractional fields and applications*. Springer, 2013. 32
- [27] C. Couprie, L. Grady, L. Najman, and H. Talbot. Power watershed : A unifying graph-based optimization framework. *IEEE Trans. Pattern Anal. Match. Int.*, 33(7) :1384–1399, 2010. 11, 26, 27
- [28] G. R. Cross and A. K. Jain. Markov random field texture models. *IEEE Trans. Pattern Anal. Match. Int.*, (1) :25–39, 1983. 28, 30
- [29] I. Daubechies. *Ten lectures on wavelets*, volume 61. SIAM, 1992. 34
- [30] E. De Giorgi and L. Ambrosio. New functionals in calculus of variations. In *Nonsmooth Optimization and Related Topics*, pages 49–59. Springer, 1989. 16
- [31] C.-A. Deledalle, S. Vaiteer, J. Fadili, and G. Peyré. Stein unbiased gradient estimator of the risk (SUGAR) for multiple parameter selection. *SIAM J. Imaging Sci.*, 7(4) :2448–2487, 2014. 23
- [32] D. L. Donoho and J. M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *biometrika*, 81(3) :425–455, 1994. 23
- [33] C. Dossal, M. Kachour, J. Fadili, G. Peyré, and C. Chesneau. The degrees of freedom of the Lasso for general design matrix. *Statistica Sinica*, pages 809–828, 2013. 23
- [34] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12(Jul) :2121–2159, 2011. 49
- [35] D. Dunn and W. E. Higgins. Optimal Gabor filters for texture segmentation. *IEEE Trans. Image Process.*, 4(7) :947–964, 1995. 32
- [36] J. C. Dunn. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *J. Cybern*, 1973. 24
- [37] Y. C. Eldar. Generalized SURE for exponential families : Applications to regularization. *IEEE Trans. Signal Process.*, 57(2) :471–481, 2008. 22
- [38] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *IEEE Trans. Pattern Anal. Match. Int.*, 35(8) :1915–1929, 2012. 56
- [39] V. Fedorov, P. Arias, R. Sadek, G. Facciolo, and C. Ballester. Linear multiscale analysis of similarities between images on Riemannian manifolds : Practical formula and affine covariant metrics. *SIAM J. Imaging Sci.*, 8(3) :2021–2069, 2015. 39
- [40] M. Foare, N. Pustelnik, and L. Condat. Semi-linearized proximal alternating minimization for a discrete Mumford–Shah model. *IEEE Trans. Image Process.*, 29(1) :2176–2189, 2019. 16
- [41] E. W. Forgy. Cluster analysis of multivariate data : efficiency versus interpretability of classifications. *Diometrics*, 21 :768–769, 1965. 24
- [42] Y. Ganin and V. Lempitsky. N^4 -Fields : Neural Network Nearest Neighbor Fields for Image Transforms. In *Asian Conference on Computer Vision*, pages 536–551. Springer, 2014. 56
- [43] D. Geman and S. Geman. Bayesian image analysis. In *Disordered systems and biological organization*, pages 301–319. Springer, 1986. 11, 12, 13, 15

- [44] D. Geman and G. Reynolds. Constrained image restoration and the recovery of discontinuities. *IEEE Trans. Pattern Anal. Match. Int.*, 14(3) :367–383, 1992. 11
- [45] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Anal. Match. Int.*, (6) :721–741, 1984. 11, 12, 13, 14, 15
- [46] L. Giffon, V. Emiya, L. Ralaivola, and H. Kadri. QuicK-means : Acceleration of K-means by learning a fast transform. *preprint arXiv :1908.08713*, 2019. 11
- [47] R. Giraud and Y. Berthoumieu. Texture Superpixel Clustering from Patch-based Nearest Neighbor Matching. In *European Signal Processing Conference (EUSIPCO 2019)*, La Corogne, Spain, Sep. 2019. URL <https://hal.archives-ouvertes.fr/hal-02059490>. 37
- [48] G. H. Golub, M. Heath, and G. Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2) :215–223, 1979. 22
- [49] U. Gonzales-Barron and F. Butler. Fractal texture analysis of bread crumb digital images. *European Food Research and Technology*, 226(4) :721–729, 2008. 28
- [50] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT press, 2016. 46
- [51] D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum a posteriori estimation for binary images. *J. Royal Stat. Soc. : Ser. B (Methodol.)*, 51(2) :271–279, 1989. 27
- [52] R. Gribonval. Should penalized least squares regression be interpreted as maximum a posteriori estimation? *IEEE Trans. Signal Process.*, 59(5) :2405–2410, 2011. 12
- [53] R. Gribonval and M. Nikolova. On bayesian estimation and proximity operators. *Appl. Comput. Harmon. Anal.*, 2019. 12
- [54] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning rich features from RGB-D images for object detection and segmentation. In *European conference on computer vision*, pages 345–360. Springer, 2014. 56
- [55] S. Haker, L. Zhu, A. Tannenbaum, and S. Angenent. Optimal mass transport for registration and warping. *Int. J. Comput. Vis.*, 60(3) :225–240, 2004. 39
- [56] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *European Conference on Computer Vision*, pages 297–312. Springer, 2014. 56
- [57] J. P. Havlicek, D. S. Harding, and A. C. Bovik. The multicomponent AM-FM image representation. *IEEE Trans. Image Process.*, 5(6) :1094–1100, 1996. 29
- [58] S. Haykin. *Neural networks : a comprehensive foundation*. Prentice Hall PTR, 1994. 46
- [59] C. Heaps and S. Handel. Similarity and features of natural textures. *J. Exp. Psychol. Hum. Percept. Perform.*, 25(2) :299, 1999. 27
- [60] R. Hecht-Nielsen. Theory of the backpropagation neural network. In *Neural networks for perception*, pages 65–93. Elsevier, 1992. 49
- [61] S. Jaffard. Wavelet techniques in multifractal analysis. *Fractal Geometry and Applications : A Jubilee of Benoit Mandelbrot, M. Lapidus et M. van Frankenhuysen Eds, Proceedings of Symposia in Pure Mathematics (AMS)*, 72(2) :91–152, 2004. 33, 35, 36
- [62] A. K. Jain. Data clustering : 50 years beyond K-means. *Pattern recognition letters*, 31(8) : 651–666, 2010. 11

- [63] M. Jung, G. Peyré, and L. D. Cohen. Non-local segmentation and inpainting. In *IEEE Proc. Int. Conf. on Image Process.*, pages 3373–3376. IEEE, 2011. [37](#)
- [64] M. Jung, G. Peyré, and L. D. Cohen. Texture segmentation via non-local non-parametric active contours. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 74–88. Springer, 2011. [37](#), [38](#), [39](#)
- [65] M. Jung, G. Peyré, and L. D. Cohen. Nonlocal active contours. *SIAM J. Imaging Sci.*, 5(3) : 1022–1054, 2012. [37](#), [39](#)
- [66] K. Kato. On the degrees of freedom in shrinkage estimation. *J. Multivariate Anal.*, 100(7) : 1338–1352, 2009. [23](#)
- [67] D. P. Kingma and J. Ba. Adam : A method for stochastic optimization. *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2014. [49](#)
- [68] I. Kokkinos, G. Evangelopoulos, and P. Maragos. Texture analysis and segmentation using modulation features, generative models, and weighted curve evolution. *IEEE Trans. Pattern Anal. Match. Int.*, 31(1) :142–157, 2008. [29](#)
- [69] I. Krizhevsky, A. Sutskever and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. [52](#)
- [70] J. Krumm and S. A. Shafer. Shape from periodic texture using the spectrogram. Technical report, Carnegie-Mellon Univ. Pittsburgh PA Robotics Inst., 1991. [28](#)
- [71] F. Li, S. Osher, J. Qin, and M. Yan. A multiphase image segmentation based on fuzzy membership functions and L1-norm fidelity. *J. Sci. Comput.*, 69(1) :82–106, 2016. [40](#)
- [72] M. Li and R. C. Staunton. Optimum Gabor filter design and local binary patterns for texture segmentation. *Pattern Recognit. Lett.*, 29(5) :664–672, 2008. [32](#)
- [73] X. Liu and D. Wang. A spectral histogram model for texton modeling and texture discrimination. *Vision Research*, 42(23) :2617–2634, 2002. [42](#)
- [74] S. Lloyd. Least squares quantization in PCM. *IEEE Trans. Inform. Theory*, 28(2) :129–137, 1982. [24](#)
- [75] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proc. IEEE Int. Conf. Comput. Vis. and Patt. Rec.*, pages 3431–3440, Boston, Massachusetts, 2015. [56](#), [57](#), [59](#)
- [76] P. Luc, C. Couprie, S. Chintala, and J. Verbeek. Semantic segmentation using adversarial networks. *Preprint arXiv :1611.08408*, 2016. [58](#)
- [77] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967. [23](#)
- [78] S. Mallat. *A wavelet tour of signal processing*. Elsevier, 1999. [23](#), [34](#), [35](#)
- [79] B. B. Mandelbrot, D. E. Passoja, and A. J. Paullay. Fractal character of fracture surfaces of metals. *Nature*, 308(5961) :721–722, 1984. [28](#)
- [80] M. Mermillod, P. Vuilleumier, C. Peyrin, D. Alleysson, and C. Marendaz. The importance of low spatial frequency information for recognising fearful facial expressions. *Connection Science*, 21(1) :75–83, 2009. [32](#)

- [81] J. W. Modestino, R. W. Fries, and A. L. Vickers. Texture discrimination based upon an assumed stochastic texture model. *IEEE Trans. Pattern Anal. Match. Int.*, (5) :557–580, 1981. [28](#)
- [82] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Commun. Pure Appl. Math*, 42(5) :577–685, 1989. [12](#), [15](#), [17](#), [27](#)
- [83] J. Nelson, C. Naornita, and A. Isar. Semi-local scaling exponent estimation with box-penalty constraints and total-variation regularization. *IEEE Trans. Image Process.*, 25(7) :3167–3181, 2016. [45](#)
- [84] K. Ni, X. Bresson, T. Chan, and S. Esedoglu. Local histogram based segmentation using the wasserstein distance. *International journal of computer vision*, 84(1) :97–111, 2009. [39](#)
- [85] F. Ning, D. Delhomme, Y. LeCun, F. Piano, L. Bottou, and P. E. Barbano. Toward automatic phenotyping of developing embryos from videos. *IEEE Trans. Image Process.*, 14(9) :1360–1371, 2005. [56](#)
- [86] N. E. Odling. Natural fracture profiles, fractal dimension and joint roughness coefficients. *Rock Mech. Rock Eng.*, 27(3) :135–153, 1994. [29](#)
- [87] M. Oliver, G. Haro, V. Fedorov, and C. Ballester. L1 patch-based image partitioning into homogeneous textured regions. In *Proc. Int. Conf. Acoust. Speech Signal Process.*, pages 1558–1562. IEEE, 2018. [37](#), [39](#), [40](#)
- [88] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed : algorithms based on hamilton-jacobi formulations. *J. Comput. Phys.*, 79(1) :12–49, 1988. [17](#)
- [89] B. Pascal, N. Pustelnik, P. Abry, and J.-C. Pesquet. Block-coordinate proximal algorithms for scale-free texture segmentation. In *Proc. Int. Conf. Acoust. Speech Signal Process.*, Calgary, Alberta, Canada, Apr. 15-20 2018. [45](#)
- [90] G. Peyré and M. Cuturi. *Computational Optimal Transport*. arXiv :1803.00567, 2018. [38](#)
- [91] P. H. O. Pinheiro and R. Collobert. Recurrent convolutional neural networks for scene labeling. In *31st International Conference on Machine Learning (ICML)*, 2014. [56](#)
- [92] K. Polisano. *Modélisation de textures anisotropes par la transformée en ondelettes monogènes et super-résolution de lignes 2-D*. PhD thesis, 2017. [32](#)
- [93] R. B. Potts. Some generalized order-disorder transformations. In *Mathematical proceedings of the Cambridge philosophical society*, volume 48, pages 106–109. Cambridge University Press, 1952. [12](#)
- [94] N. Pustelnik and L. Condat. Proximity operator of a sum of functions ; application to depth map estimation. *IEEE Signal Process. Lett.*, 24(12) :1827–1831, 2017. [19](#)
- [95] N. Pustelnik, A. Benazza-Benhayia, Y. Zheng, and J.-C. Pesquet. Wavelet-based image deconvolution and reconstruction. *Wiley Encyclopedia of Electrical and Electronics Engineering*, Feb. 2016. [12](#)
- [96] N. Pustelnik, H. Wendt, P. Abry, and N. Dobigeon. Combining local regularity estimation and total variation optimization for scale-free texture segmentation. *IEEE Trans. Comput. Imaging*, 2(4) :468–479, 2016. URL <http://ieeexplore.ieee.org/abstract/document/7523285/>. [45](#)

- [97] J. Rabin, G. Peyré, J. Delon, and M. Bernot. Wasserstein barycenter and its application to texture mixing. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 435–446. Springer, 2011. [39](#)
- [98] S. Ramani, T. Blu, and M. Unser. Monte-carlo SURE : A black-box optimization of regularization parameters for general denoising algorithms. *IEEE Trans. Image Process.*, 17(9) : 1540–1554, 2008. [23](#)
- [99] S. J. Reddi, S. Kale, and S. Kumar. On the convergence of ADAM and beyond. *preprint arXiv :1904.09237*, 2019. [49](#)
- [100] E. Ribeiro and E. R. Hancock. Shape from periodic texture using the eigenvectors of local affine distortion. *IEEE Trans. Pattern Anal. Match. Int.*, 23(12) :1459–1465, 2001. [28](#)
- [101] O. Ronneberger, P. Fischer, and T. Brox. U-net : Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. [58](#)
- [102] F. Rosenblatt. The perceptron : a probabilistic model for information storage and organization in the brain. *Psychol. Rev.*, 65(6) :386, 1958. [47](#)
- [103] N. L. Roux, M. Schmidt, and F. R. Bach. A stochastic gradient method with an exponential convergence _rate for finite training sets. In *Advances in neural information processing systems*, pages 2663–2671, 2012. [49](#)
- [104] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2) :99–121, 2000. [39](#)
- [105] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D : Nonlinear phenomena*, 60(1-4) :259–268, 1992. [12](#), [20](#)
- [106] G. Samorodnitsky, M. S. Taqqu, and R. W. Linde. *Stable non-gaussian random processes : stochastic models with infinite variance*. New York. Chapman & Hall, 1994. [32](#)
- [107] E. Sejdić, I. Djurović, and J. Jiang. Time–frequency feature representation using energy concentration : An overview of recent advances. *Digital signal processing*, 19(1) :153–183, 2009. [33](#)
- [108] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, Inc., 1983. [26](#)
- [109] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *Preprint arXiv :1409.1556*, 2014. [52](#)
- [110] P. K. Simpson. *Artificial neural systems : foundations, paradigms, applications, and implementations*. Elsevier Science Inc., 1989. [46](#)
- [111] C. M. Stein. Estimation of the mean of a multivariate normal distribution. *Ann. Stat.*, pages 1135–1151, 1981. [22](#)
- [112] H. Steinhaus. Sur la division des corp materiels en parties. *Bull. Acad. Polon. Sci*, 1(804) : 801, 1956. [23](#)
- [113] M. Storath, A. Weinmann, J. Friel, and M. Unser. Joint image reconstruction and segmentation using the Potts model. *Inverse Problems*, 31(2) :025003, 2015. [12](#)
- [114] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. [52](#)

- [115] A. M. Thompson, J. C. Brown, J. W. Kay, and D. M. Titterington. A study of methods of choosing the smoothing parameter in image restoration by regularization. *IEEE Trans. Pattern Anal. Match. Int.*, (4) :326–339, 1991. [22](#)
- [116] R. J. Tibshirani and J. Taylor. Degrees of freedom in lasso problems. *Ann. Stat.*, 40(2) :1198–1232, 2012. [23](#)
- [117] C. Vacar and J.-F. Giovannelli. Unsupervised joint deconvolution and segmentation method for textured images : a Bayesian approach and an advanced sampling algorithm. *EURASIP J. Adv. Signal Process.*, 2019(1) :17, 2019. [32](#), [41](#), [42](#)
- [118] S. Vaiter, C. Deledalle, J. Fadili, G. Peyré, and C. Dossal. The degrees of freedom of partly smooth regularizers. *Ann. Inst. Stat. Math*, 69(4) :791–832, 2017. [23](#)
- [119] L. Vincent. Morphological algorithms. *Mathematical Morphology in Image Processing*, pages 255–288, 1992. [26](#)
- [120] L. Vincent and P. Soille. Watersheds in digital spaces : an efficient algorithm based on immersion simulations. *IEEE Trans. Pattern Anal. Match. Int.*, (6) :583–598, 1991. [26](#)
- [121] T. P. Weldon, W. E. Higgins, and D. F. Dunn. Efficient Gabor filter design for texture segmentation. *Pattern Recognit.*, 29(12) :2005–2015, 1996. [32](#)
- [122] H. Wendt, P. Abry, and S. Jaffard. Bootstrap for empirical multifractal analysis. *IEEE Signal Process. Mag.*, 24(4) :38–48, 2007. [35](#), [36](#)
- [123] H. Wendt, S. G. Roux, S. Jaffard, and P. Abry. Wavelet leaders and bootstrap for multifractal analysis of images. *Signal Processing*, 89(6) :1100–1114, 2009. [35](#), [36](#), [37](#)
- [124] H. Wendt, S. Combrexelle, Y. Altmann, J.-Y. Tournet, S. McLaughlin, and P. Abry. Multifractal analysis of multivariate images using gamma Markov random field priors. *SIAM J. Imaging Sci.*, 11(2) :1294–1316, 2018. [37](#)
- [125] J. Yuan, D. Wang, and A. M. Cheriyyadat. Factorization-based texture segmentation. *IEEE Trans. Image Process.*, 24(11) :3488–3497, 2015. [32](#), [42](#)
- [126] M. Zanetti and A. Vitti. The Blake-Zisserman model for digital surface models segmentation. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 5 :W2, 2013. [12](#)
- [127] Y. Zhang. Optimisation of building detection in satellite images by combining multispectral classification and texture filtering. *ISPRS J. Photogramm. Remote Sens.*, 54(1) :50–60, 1999. [28](#)

Chapitre 3

Synthèse de textures fractales

« Tu ne craindras ni les terreurs de la nuit,
Ni la flèche qui vole au grand jour,
Ni la peste qui marche dans les ténèbres,
Ni la contagion qui frappe en plein midi. »

Psaume 91 : 5-6

Sommaire

3.1 Champs gaussiens auto similaires	68
3.1.1 Auto-similarité et accroissements stationnaires	68
3.1.2 Champs browniens fractionnaires	68
3.1.3 Bruits gaussiens fractionnaires	70
3.1.4 Proposition de champ gaussien stationnaire asymptotiquement auto-similaire	74
3.2 Synthèse de champs gaussiens	78
3.2.1 Méthode exacte par factorisation de Cholesky	78
3.2.2 Méthodes approchées	79
3.3 Textures fractales homogènes par morceaux	80
3.4 Bilan	82

Cette thèse s'intéresse à la segmentation de texture à partir des attributs fractals présentés à la Section 2.2.3.2 du Chapitre 2. Afin de valider les performances des méthodes proposées, il est nécessaire de générer des textures *synthétiques* dont la géométrie et les attributs fractals sont connus et prescrits *a priori*. Le modèle le plus courant de texture fractale de régularité locale uniforme est le champ brownien fractionnaire, que nous présentons à la Section 3.1.2. Néanmoins, pour la construction de textures homogènes par morceaux, ce type de champ *non stationnaire* présente des problèmes de recollement aux frontières entre textures. Pour pallier ce problème, nous construisons à la Section 3.1 à un modèle de champ aléatoire *stationnaire* qui fournit une classe de textures stochastiques dont la régularité et la variance locales sont uniformes : les champs gaussiens stationnaires asymptotiquement auto similaires. Les méthodes de synthèse de tels champs sur une grille de pixels, pour un paramètre d'auto-similarité H fixé, sont ensuite discutées en Section 3.2. Enfin, nous décrivons un modèle particulier de texture *fractale* homogène par morceaux et sa génération pratique en Section 3.3.

Ce modèle nous permettra, dans la suite de cette thèse, de disposer de textures synthétiques, composées de k régions $\Omega_1, \dots, \Omega_k$, où chaque région Ω_i correspond à une texture de régularité locale H_i et de variance locale Σ_i^2 uniformes. La procédure proposée permet de choisir librement la partition du domaine de l'image en k régions, ainsi que la régularité et la variance locales de chaque texture. Ainsi, les performances d'estimation de la régularité locale et de segmentation des méthodes proposées au Chapitre 4 pourront être quantifiées à partir de plusieurs réalisations d'une texture donnée.

3.1 Champs gaussiens auto similaires

Nous allons nous concentrer dans cette thèse sur des textures *auto similaires*, caractérisées par les attributs fractals décrits au Chapitre 2, Section 2.2.3.2. La notion d'auto-similarité a été introduite en tout premier lieu dans l'étude des signaux de vitesse issus d'expérience de turbulence des fluides [26, 16]. Par la suite, elle a montré sa grande efficacité pour l'étude de signaux provenant de domaines variés, dont l'étude du rythme cardiaque [13], du trafic Internet [15] ou encore de la finance [25]. Rapidement, la généralisation de l'auto-similarité aux images a permis de rendre compte de leur structure interne. Ainsi, dans de nombreuses disciplines, les propriétés d'auto-similarité des données, telles que des images satellites [31], des images médicales [20] ou encore des images issues du domaine de l'art et du patrimoine [2, 21], se sont avérées déterminantes pour de multiples tâches, allant de la classification à la segmentation.

3.1.1 Auto-similarité et accroissements stationnaires

Intuitivement, on dit qu'une image est *auto similaire* lorsqu'elle reste « identique » sous l'action d'opérations géométriques, telles que, par exemple, les dilatations par un facteur $a > 0$. Comme souligné dans [33, 9, 29], il existe plusieurs interprétations de l'auto-similarité : en fonction par exemple des opérations géométriques considérées, ou de la définition exacte du caractère « identique ». Par conséquent, une très grande variété de champs auto similaires ont été décrits dans la littérature mathématique [18, 14] et une classification exhaustive est exclue. Nous aborderons uniquement les champs *stochastiques* auto similaires gaussiens. Nous nous attacherons à présenter le formalisme utilisé dans cette thèse, ainsi que les principaux champs d'intérêt pour la synthèse de texture.

Définition 3.1 (Auto-similarité). [33, Définition 8.1.1] Soit $f : \mathbb{R}^D \rightarrow \mathbb{R}$ un champ stochastique D -dimensionnel et $0 < H < 1$, f est H -auto similaire si et seulement si

$$\forall a > 0, \quad \{f(a\underline{x})\}_{\underline{x} \in \mathbb{R}^D} \stackrel{(d)}{=} a^H \{f(\underline{x})\}_{\underline{x} \in \mathbb{R}^D}, \quad (3.1)$$

où $\stackrel{(d)}{=}$ désigne l'égalité au sens des distributions.

Définition 3.2 (Stationnarité des accroissements). [33, Définition 8.1.2] Un champ stochastique $f : \mathbb{R}^D \rightarrow \mathbb{R}$ est à *accroissements stationnaires* si et seulement si, pour tout vecteur $\underline{\delta} \in \mathbb{R}^D$,

$$\{f(\underline{x} + \underline{\delta}) - f(\underline{\delta})\}_{\underline{x} \in \mathbb{R}^D} \stackrel{(d)}{=} \{f(\underline{x}) - f(\underline{0})\}_{\underline{x} \in \mathbb{R}^D}. \quad (3.2)$$

L'Égalité (3.2) traduit l'invariance statistique des accroissements du champ f sous l'ensemble des translations de \mathbb{R}^D .

Proposition 3.1. *En dimension $D = 1$, à un facteur multiplicatif près, il existe un unique champ H -auto similaire à accroissements stationnaires, qui est appelé mouvement brownien fractionnaire [33, Lemme 7.2.1].*

3.1.2 Champs browniens fractionnaires

En dimension strictement supérieure à 1, la Proposition 3.1 n'est plus valable [29, Chapitre 9]. Néanmoins, grâce à la généralisation de sa représentation harmonisable (voir Équation (3.3) ci-dessous), le *mouvement brownien fractionnaire* possède une extension directe en dimension supérieure [33, Exemple 8.1.3]. Dans toute la suite de ce chapitre, nous noterons b_H l'extension harmonisable du *mouvement brownien fractionnaire* en dimension $D > 1$, appelé *champ brownien fractionnaire*, et défini à l'Équation (3.3).

Exemple 3.1 (Champ brownien fractionnaire). Pour $0 < H < 1$, et Σ réel positif, le champ D -dimensionnel b_H défini par la représentation harmonisable

$$b_H(\underline{x}) \stackrel{(d)}{=} \Sigma \int_{\mathbb{R}^D} \frac{e^{-i\langle \underline{x}, \underline{v} \rangle} - 1}{C_H^{1/2} \|\underline{v}\|^{H + \frac{D}{2}}} \tilde{w}(d\underline{v}), \quad (3.3)$$

où $\underline{v} \in \mathbb{R}^D$ désigne la fréquence spatiale, C_H est une constante de normalisation, Σ^2 est la variance du champ b_H au point $\underline{x} = (0, 0)$, et $\tilde{w}(d\underline{v})$ est une mesure de brownienne aléatoire sur \mathbb{R}^D [9, Définition 2.1.16], est appelé le *champ brownien fractionnaire*. La constante C_H , qui permet une normalisation de la variance sur le cercle unité $\mathcal{S} = \{\underline{x} : \|\underline{x}\| = 1\}$, peut être explicitée (voir par exemple [23, Proposition 3.3.1] pour le calcul détaillé)

$$C_H = \int_{\mathbb{R}^2} \frac{2(1 - \cos(v_1))}{\|\underline{v}\|^{2H+D}} \frac{d\underline{v}}{(2\pi)^{D/2}} = \frac{\pi^{1/2} \Gamma(H + 1/2)}{2^{D/2} H \Gamma(2H) \sin(\pi H) \Gamma(H + D/2)}, \quad (3.4)$$

où Γ désigne la fonction Gamma d'Euler.

Propriétés 3.1. [33, Proposition 8.1.4], [9, Définition 3.3.1] *Le champ b_H , défini à l'Équation (3.3) possède les propriétés suivantes :*

- (i) *Il est H -auto similaire.*
- (ii) *Il est à accroissements stationnaires (Définition 3.2).*
- (iii) *Il est isotrope, c'est-à-dire invariant, au sens des distributions, sous les rotations du plan \mathbb{R}^2 .*
- (iv) *Sa structure de covariance est donnée par*

$$\mathbb{E}[b_H(\underline{x}) b_H(\underline{x}')] = \frac{\Sigma^2}{2} (\|\underline{x}\|^{2H} + \|\underline{x}'\|^{2H} - \|\underline{x} - \underline{x}'\|^{2H}), \quad (3.5)$$

où nous appellerons variance typique la quantité Σ^2 , qui correspond à la variance sur le cercle unité \mathcal{S} .

Démonstration. La démonstration des Propriétés 3.1 est reproduite en Annexe A. □

On remarque à la structure de covariance (3.5) que le champ brownien fractionnaire est non stationnaire, puisque sa covariance dépend explicitement des positions \underline{x} et \underline{x}' et pas uniquement de leur différence $\underline{x} - \underline{x}'$. Une réalisation du champ brownien fractionnaire b_H , de paramètre d'auto-similarité $H = 0,7$, est présenté à la Figure 3.1a.

Proposition 3.2. *Le champ brownien fractionnaire b_H défini par (3.3) est caractérisé par son paramètre d'auto-similarité $0 < H < 1$ et par sa variance typique Σ^2 [33, Proposition 8.1.4]. De plus, sa régularité locale $h(\underline{x})$, introduite à la Définition 2.8, quantifiée grâce à l'exposant de Hölder local, est uniforme et égale à H en tout point du plan [6, Proposition 4]. Ainsi, si on note $\mathcal{L}_j[b_H](\underline{n})$ les coefficients d'ondelettes dominants du champ brownien fractionnaire de paramètre d'auto-similarité H , définis au Chapitre 2, Équation (2.83), aux échelles les plus fines on observe le comportement local suivant*

$$\mathbb{E} \mathcal{L}_j[b_H](\underline{n}) \underset{2^j \rightarrow 0}{\lesssim} \eta(\underline{n}) 2^{j(H+\gamma)}, \quad (3.6)$$

au point \underline{n} , où γ est le paramètre d'intégration fractionnaire inclus dans la définition des coefficients dominants à l'Équation (2.83).

Lorsque $D > 1$, des variantes des accroissements existent et permettent de proposer des alternatives de la Définition 3.2. En dimension $D = 2$, les *accroissements rectangulaires*, définis par (3.7), permettent d'explorer les deux dimensions pour définir une autre gamme de champs auto-similaires.

Définition 3.3 (Accroissements rectangulaires). [9, Section 3.3.2] Dans le plan réel, on peut définir des *accroissements rectangulaires* :

$$(\Delta_{\underline{\delta}}f)(\underline{x}) = f(x_1 + \delta_1, x_2 + \delta_2) - f(x_1 + \delta_1, \delta_2) - f(\delta_1, x_2 + \delta_2) + f(\delta_1, \delta_2). \quad (3.7)$$

Un champ f est à *accroissements rectangulaires stationnaires* si et seulement si, pour tout $\underline{\delta} \in \mathbb{R}^2$,

$$\{(\Delta_{\underline{\delta}}f)(\underline{x})\}_{\underline{x} \in \mathbb{R}^2} \stackrel{(d)}{=} \{(\Delta_{\underline{0}}f)(\underline{x})\}_{\underline{x} \in \mathbb{R}^2}. \quad (3.8)$$

Exemple 3.2 (Drap brownien fractionnaire). [9, Section 3.3.2] Pour deux exposants H_1, H_2 compris dans $]0, 1[$, on définit le champ

$$d_{H_1, H_2}(\underline{x}) \stackrel{(d)}{=} \Sigma \int_{\mathbb{R}^2} \frac{e^{-ix_1 v_1} - 1}{C_{H_1}^{1/2} |v_1|^{H_1+1/2}} \frac{e^{-ix_2 v_2} - 1}{C_{H_2}^{1/2} |v_2|^{H_2+1/2}} \tilde{w}(d\underline{v}), \quad (3.9)$$

que l'on l'appelle le *drap brownien fractionnaire*. Ce modèle de champ a été introduit par Léger [24], dans le but de modéliser des champs browniens fractionnaires anisotropes.

Propriétés 3.2. Le champ d_{H_1, H_2} , défini à l'Équation (3.9), possède les propriétés suivantes :

- (i) Il satisfait une propriété d'auto-similarité rectangulaire : pour tout $\underline{a} = (a_1, a_2)$ avec $a_1 > 0$ et $a_2 > 0$

$$d_{H_1, H_2}(a_1 x_1, a_2 x_2) \stackrel{(d)}{=} a_1^{H_1} a_2^{H_2} d_{H_1, H_2}(\underline{x}). \quad (3.10)$$

En particulier, en prenant $a_1 = a_2$, on remarque que le champ d_{H_1, H_2} est $H_1 + H_2$ -auto similaire.

- (ii) Il est à *accroissements rectangulaires stationnaires* (Définition 3.3).
(Notons que ce champ n'est pas à *accroissements stationnaires*, tels que définis à l'Équation (3.2).)
- (iii) Il est *anisotrope*, y compris dans le cas où les deux exposants H_1, H_2 sont égaux (voir la Figure 3.1b).
- (iv) Sa structure de covariance s'écrit comme un produit

$$\begin{aligned} & \mathbb{E} [d_{H_1, H_2}(\underline{x}) d_{H_1, H_2}(\underline{x}')] \\ &= \frac{\Sigma^2}{4} (\|x_1\|^{2H_1} + \|x'_1\|^{2H_1} - \|x_1 - x'_1\|^{2H_1}) (\|x_2\|^{2H_2} + \|x'_2\|^{2H_2} - \|x_2 - x'_2\|^{2H_2}). \end{aligned} \quad (3.11)$$

De même que pour le champ brownien fractionnaire, on remarque à sa structure de covariance (3.11) qu'un drap brownien fractionnaire d_{H_1, H_2} , défini par (3.9), n'est pas stationnaire.

3.1.3 Bruits gaussiens fractionnaires

Les deux champs b_H et d_{H_1, H_2} présentés à la Section 3.1.2 sont tous les deux gaussiens et auto similaires mais *non stationnaires*. Or, pour la modélisation de textures réelles, et en particulier pour la segmentation de textures, les champs *stationnaires*, c'est-à-dire statistiquement invariants sous les translations (voir Définition 3.4), sont plus représentatifs et plus pertinents, notamment par rapport aux applications que nous allons considérer au Chapitre 6. En effet, la segmentation de texture consiste en la partition de l'image X (associée au champ sous-jacent f) en différentes régions Ω_i , correspondant chacune à *une seule* texture. Or, il est commun [22, 27] de considérer qu'au sein de chaque région Ω_i les propriétés statistiques du champ f sont *uniformes*, i.e., stationnaires.

Comme illustré par les Propriétés 3.1 et 3.2, on peut construire des champs gaussiens *stationnaires* en considérant les accroissements du champ brownien fractionnaire (ou les accroissements rectangulaires du drap brownien fractionnaire), tels qu'illustrés en Figures 3.1c et 3.1d. Les textures réelles que nous étudierons au Chapitre 6, et que nous souhaitons modéliser, sont isotropes, nous allons donc privilégier les accroissements du champ brownien fractionnaire afin de reproduire le mieux possible ces textures réelles.

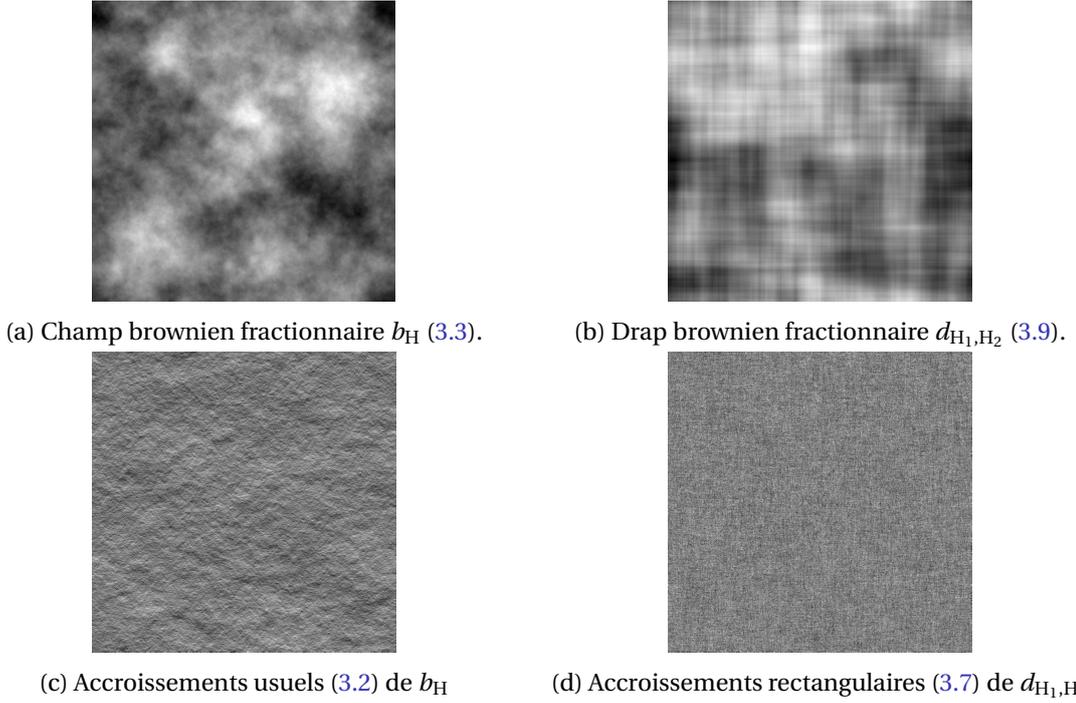


FIGURE 3.1 – Exemples de champs browniens fractionnaires avec $H = H_1 = H_2 = 0,7$ et leurs accroissements stationnaires.

Définition 3.4 (Champ stationnaire). Un champ stochastique $f : \mathbb{R}^D \rightarrow \mathbb{R}$ est *stationnaire* si et seulement si, pour tout vecteur $\underline{\delta} \in \mathbb{R}^D$,

$$\{f(\underline{x} + \underline{\delta})\}_{\underline{x} \in \mathbb{R}^D} \stackrel{(d)}{=} \{f(\underline{x})\}_{\underline{x} \in \mathbb{R}^D}. \quad (3.12)$$

L'égalité (3.12) traduit l'invariance statistique du champ f sous l'ensemble des translations de \mathbb{R}^D [33, 29].

Cette notion se simplifie grandement dans le cas gaussien grâce à la proposition suivante.

Proposition 3.3 (Stationnarité des champs gaussiens). *Un champ stochastique $f : \mathbb{R}^D \rightarrow \mathbb{R}$ gaussien est stationnaire si et seulement il est stationnaire au sens faible au second ordre [33, 29], c'est-à-dire si*

- (i) Sa moyenne $\mathbb{E}[f(\underline{x})]$ est indépendante du point $\underline{x} \in \mathbb{R}^D$.
- (ii) Sa covariance $\mathbb{E}[f(\underline{x})f(\underline{x}')]]$ ne dépend que de $\underline{x} - \underline{x}'$.

Nous détaillons ici l'exemple du champ des accroissements du champ brownien fractionnaire selon un vecteur fixé $\underline{\delta}$.

Exemple 3.3 (Bruit gaussien fractionnaire). Pour un choix de vecteur $\underline{\delta} \in \mathbb{R}^2$, on définit le champ f_H par :

$$f_H(\underline{x}) \triangleq b_H(\underline{x} + \underline{\delta}) - b_H(\underline{x}) = \Sigma \int_{\mathbb{R}^2} \frac{e^{-i\langle \underline{\delta}, \underline{v} \rangle} - 1}{C_H^{1/2} \|\underline{v}\|^{H+1}} e^{-i\langle \underline{x}, \underline{v} \rangle} \tilde{w}(d\underline{v}). \quad (3.13)$$

Ce champ est appelé le *bruit gaussien fractionnaire* de paramètre H et de variance Σ^2 .

Proposition 3.4. *Le bruit gaussien fractionnaire f_H a pour covariance*

$$\mathbb{E}[f_H(\underline{x})f_H(\underline{x}')] = \frac{\Sigma^2}{2} (\|\underline{x} - \underline{x}' + \underline{\delta}\|^{2H} + \|\underline{x} - \underline{x}' - \underline{\delta}\|^{2H} - 2\|\underline{x} - \underline{x}'\|^{2H}). \quad (3.14)$$

De plus, dans la limite des grandes échelles, i.e., $\|\underline{x} - \underline{x}'\| \gg \|\underline{\delta}\|$, la covariance du champ f_H se comporte comme

$$\mathbb{E}[f_H(\underline{x})f_H(\underline{x}')] \simeq \frac{\Sigma^2}{2} \|\underline{x} - \underline{x}'\|^{2H-2} \left(2H \|\underline{\delta}\|^2 + 4H(H-1) \left\langle \frac{\underline{x} - \underline{x}'}{\|\underline{x} - \underline{x}'\|}, \underline{\delta} \right\rangle^2 \right). \quad (3.15)$$

Démonstration. Le calcul de la covariance du champ f_H conduit à l'expression suivante :

$$\begin{aligned} \mathbb{E}[f_H(\underline{x})f_H(\underline{x}')] &= \mathbb{E}[b_H(\underline{x} + \underline{\delta})b_H(\underline{x}' + \underline{\delta})] - \mathbb{E}[b_H(\underline{x} + \underline{\delta})b_H(\underline{x}')] - \mathbb{E}[b_H(\underline{x})b_H(\underline{x}' + \underline{\delta})] + \mathbb{E}[b_H(\underline{x})b_H(\underline{x}')] \\ &= \frac{\Sigma^2}{2} (\|\underline{x} + \underline{\delta}\|^{2H} + \|\underline{x}' + \underline{\delta}\|^{2H} - \|\underline{x} - \underline{x}'\|^{2H}) \\ &\quad - \frac{\Sigma^2}{2} (\|\underline{x} + \underline{\delta}\|^{2H} + \|\underline{x}'\|^{2H} - \|\underline{x} - \underline{x}' + \underline{\delta}\|^{2H}) \\ &\quad - \frac{\Sigma^2}{2} (\|\underline{x}\|^{2H} + \|\underline{x}' + \underline{\delta}\|^{2H} - \|\underline{x} - \underline{x}' - \underline{\delta}\|^{2H}) \\ &\quad + \frac{\Sigma^2}{2} (\|\underline{x}\|^{2H} + \|\underline{x}'\|^{2H} - \|\underline{x} - \underline{x}'\|^{2H}) \\ &= \frac{\Sigma^2}{2} (\|\underline{x} - \underline{x}' + \underline{\delta}\|^{2H} + \|\underline{x} - \underline{x}' - \underline{\delta}\|^{2H} - 2\|\underline{x} - \underline{x}'\|^{2H}), \end{aligned} \quad (3.16)$$

qui ne dépend bien que de la différence $\underline{x} - \underline{x}'$. En fixant, $\underline{x} = \underline{x}'$ on obtient l'expression de la variance du champ f_H en tout point \underline{x} :

$$\mathbb{E}[f_H(\underline{x})^2] = \Sigma^2 \|\underline{\delta}\|^{2H}, \quad (3.17)$$

qui dépend de la norme du vecteur $\underline{\delta}$.

Pour effectuer le développement asymptotique de la covariance de l'Équation (3.14) dans la limite $\|\underline{x} - \underline{x}'\| \gg \|\underline{\delta}\|$ on pose $\underline{y} \triangleq (\underline{x} - \underline{x}')/\|\underline{x} - \underline{x}'\|$ et $\underline{\delta}' \triangleq \underline{\delta}/\|\underline{x} - \underline{x}'\|$ puis en développant la fonction $t \mapsto (1+t)^H$ au voisinage de $t = 0$ on obtient

$$\|\underline{x} - \underline{x}' \pm \underline{\delta}\|^{2H} = \|\underline{x} - \underline{x}'\|^{2H} \left(1 \pm 2 \langle \underline{y}, \underline{\delta}' \rangle + \|\underline{\delta}'\|^2 \right)^H \quad (3.18)$$

$$\simeq \|\underline{x} - \underline{x}'\|^{2H} \left(1 \pm 2H \langle \underline{y}, \underline{\delta}' \rangle + H \|\underline{\delta}'\|^2 + 2H(H-1) \langle \underline{y}, \underline{\delta}' \rangle^2 \right). \quad (3.19)$$

Réinjecter ce développement dans l'expression de la covariance de l'Équation (3.14) conduit à

$$\mathbb{E}[f_H(\underline{x})f_H(\underline{x}')] \simeq \frac{\Sigma^2}{2} \|\underline{x} - \underline{x}'\|^{2H} \left(2H \|\underline{\delta}'\|^2 + 4H(H-1) \langle \underline{y}, \underline{\delta}' \rangle^2 \right) \quad (3.20)$$

$$\simeq \frac{\Sigma^2}{2} \|\underline{x} - \underline{x}'\|^{2H-2} \left(2H \|\underline{\delta}\|^2 + 4H(H-1) \left\langle \frac{\underline{x} - \underline{x}'}{\|\underline{x} - \underline{x}'\|}, \underline{\delta} \right\rangle^2 \right). \quad (3.21)$$

L'expression ci-dessus est la signature d'un comportement $H - 1$ -auto similaire, asymptotique puisqu'obtenu dans la limite $\|\underline{x} - \underline{x}'\| \gg \|\underline{\delta}\|$. \square

Propriétés 3.3. Le bruit gaussien fractionnaire f_H , défini à l'Équation (3.13) possède les propriétés suivantes :

- (i) Il est asymptotiquement $H - 1$ -auto similaire dans la limite des grandes échelles.
- (ii) Il est stationnaire.
- (iii) Il est isotrope.

Démonstration. Ces trois propriétés découlent directement de l'expression de la covariance du bruit gaussien fractionnaire et de son comportement asymptotique, exposés à la Proposition 3.4. \square

Remarque. Contrairement au champ brownien fractionnaire b_H de l'Exemple 3.1, le bruit gaussien fractionnaire n'est pas *exactement* auto-similaire. En effet, pour un vecteur d'accroissement $\underline{\delta}$ et un facteur de dilatation $a > 0$, en notant f_H^δ le champ obtenu comme le $\underline{\delta}$ -accroissement de b_H :

$$\begin{aligned} f_H^\delta(\underline{ax}) &\stackrel{(\text{d\'ef.})}{=} b_H(\underline{ax} + \underline{\delta}) - b_H(\underline{ax}) \\ &= b_H(\underline{ax} + a(\underline{\delta}/a)) - b_H(\underline{ax}) \\ \text{auto-similarit\'e de } b_H &= a^H b_H(\underline{x} + \underline{\delta}/a) - a^H b_H(\underline{x}) \\ &= a^H f_H^{\underline{\delta}/a}(\underline{x}). \end{aligned} \quad (3.22)$$

Proposition 3.5 (R\'egularit\'e locale du bruit gaussien fractionnaire). *La fonction de structure d'ordre deux du bruit gaussien fractionnaire, f_H , d\'efini \u00e0 l'\'Equation (3.13), se comporte aux petites \u00e9chelles, i.e., $\|\underline{u}\| \ll \|\underline{\delta}\|$ comme*

$$\mathbb{E} \left[(f_H(\underline{x} + \underline{u}) - f_H(\underline{x}))^2 \right] = 2\Sigma^2 \|\underline{u}\|^{2H}. \quad (3.23)$$

Le champ f_H poss\ede donc une r\'egularit\'e locale uniforme $h(\underline{x}) \equiv H$.

D\'emonstration. Effectuons le d\'eveloppement asymptotique de la covariance de l'\'Equation (3.14) dans la limite $\|\underline{x} - \underline{x}'\| \ll \|\underline{\delta}\|$. Pour cela, on pose $\underline{y} \triangleq (\underline{x} - \underline{x}')/\|\underline{\delta}\|$ et $\underline{\delta}' \triangleq \underline{\delta}/\|\underline{\delta}\|$ puis en d\'eveloppant la fonction $t \mapsto (1+t)^H$ au voisinage de $t=0$ on obtient

$$\|\underline{x} - \underline{x}' \pm \underline{\delta}\|^{2H} = \|\underline{\delta}\|^{2H} \left(1 \pm 2 \langle \underline{y}, \underline{\delta}' \rangle + \|\underline{y}\|^2 \right)^H \quad (3.24)$$

$$\simeq \|\underline{\delta}\|^{2H} \left(1 \pm 2H \langle \underline{y}, \underline{\delta}' \rangle + H \|\underline{y}\|^2 + 2H(H-1) \langle \underline{y}, \underline{\delta}' \rangle^2 \right). \quad (3.25)$$

R\'einjecter ce d\'eveloppement dans l'expression de la covariance de l'\'Equation (3.14) conduit \u00e0

$$\mathbb{E} [f_H(\underline{x}) f_H(\underline{x}')] \simeq \frac{\Sigma^2}{2} \|\underline{\delta}\|^{2H} \left(2 + 2H \|\underline{\delta}'\|^2 + 4H(H-1) \langle \underline{y}, \underline{\delta}' \rangle^2 - 2\|\underline{y}\|^{2H} \right), \quad (3.26)$$

puis, gr\u00e2ce \u00e0 l'hypoth\ese $0 < H < 1$ on a $2H < 2$ et les termes quadratiques en $\|\underline{y}\|$ sont n\'egligeables devant le terme $\|\underline{y}\|^{2H}$ ce qui conduit au d\'eveloppement asymptotique de la covariance

$$\mathbb{E} [f_H(\underline{x}) f_H(\underline{x}')] \simeq \Sigma^2 \|\underline{\delta}\|^{2H} \left(1 - \|\underline{y}\|^{2H} \right), \quad (3.27)$$

$$\simeq \Sigma^2 \left(\|\underline{\delta}\|^{2H} - \|\underline{x} - \underline{x}'\|^{2H} \right). \quad (3.28)$$

Ainsi, dans la limite $\|\underline{x} - \underline{x}'\| \ll \|\underline{\delta}\|$, la fonction de structure d'ordre deux du bruit gaussien fractionnaire s'\'ecrit, pour tout $\underline{x} \in \mathbb{R}^D$ et pour tout $\underline{u} \in \mathbb{R}^D$,

$$\mathbb{E} \left[(f_H(\underline{x} + \underline{u}) - f_H(\underline{x}))^2 \right] = 2\Sigma^2 \|\underline{u}\|^{2H}, \quad (3.29)$$

et la r\'egularit\'e locale du bruit gaussien fractionnaire est donc uniforme, \u00e9gale \u00e0 H . \square

Le bruit gaussien fractionnaire f_H \u00e9tant asymptotiquement $H-1$ -auto-similaire, comme d\'emonstr\'e \u00e0 la Propri\'et\'e 3.4, il est qualifi\'e de texture *fractale*. En outre, la Proposition 3.5 assurant que la r\'egularit\'e locale du bruit gaussien fractionnaire est *uniforme*, une d\'enomination plus pr\'ecise est celle de texture *monofractale*.

L'expression de la covariance du bruit gaussien fractionnaire \u00e0 l'\'Equation (3.14) montre qu'il est *isotrope*, comme l'assure la Propri\'et\'e 3.3. N\'eanmoins, le choix arbitraire d'une direction pour le vecteur d'accroissement $\underline{\delta}$ de l'\'Equation (3.13) peut conduire \u00e0 des effets d'anisotropie, caus\'es par la manipulation d'images discr\etes. \u00c0 partir d'un m\eme champ brownien fractionnaire, pr\'esent\'e

en Figure 3.2a, deux exemples de bruits gaussiens sont construits pour $\underline{\delta} = (1; 0)$, à la Figure 3.2b, ou $\underline{\delta} = (0; 1)$, à la Figure 3.2c, et illustrent cette anisotropie « artificielle » due au caractère discret des images. Or les textures réelles que nous souhaitons modéliser sont isotropes. Afin de pallier ce problème, la section suivante s'inspire des champs étudiés par [5, 32, 11], pour proposer un modèle alternatif de champ gaussien, conservant les propriétés de stationnarité et de régularité décrites à la Section 3.1.3, et dont la discrétisation n'induit pas d'anisotropie.



(a) Champ brownien fractionnaire (3.3)

(b) Bruit gaussien fractionnaire (3.13) avec $\underline{\delta} = (1; 0)$

(c) Bruit gaussien fractionnaire (3.13) avec $\underline{\delta} = (0; 1)$

FIGURE 3.2 – Effet d'anisotropie « artificielle » induite sur le bruit gaussien fractionnaire par le caractère discret des images.

3.1.4 Proposition de champ gaussien stationnaire asymptotiquement auto-similaire

Nous proposons une variante du bruit gaussien fractionnaire de l'Équation (3.13) pour la définition d'un champ gaussien stationnaire asymptotiquement auto-similaire, isotrope, construite à partir d'accroissements horizontaux et verticaux. La structure de covariance du champ proposé est calculée de manière exacte et une représentation harmonisable de ce champ est dérivée.

Définition 3.5 (Champ gaussien stationnaire asymptotiquement auto-similaire par accroissements unitaires horizontaux et verticaux). Soit b_H un champ brownien fractionnaire, de paramètre d'auto-similarité H et de variance Σ^2 , tel que défini par la Formule (3.3). On note $\underline{e}_1 = (1; 0)$ et $\underline{e}_2 = (0; 1)$ les vecteurs de base canoniques de \mathbb{R}^2 . Pour $\delta \in \mathbb{R}^*$, le champ gaussien stationnaire asymptotiquement auto-similaire que nous proposons est défini comme la somme d'un accroissement horizontal et d'un accroissement vertical :

$$g_H(\underline{x}) \triangleq \frac{1}{2N_H^{1/2}} \underbrace{(b_H(\underline{x} + \delta \underline{e}_1) - b_H(\underline{x}))}_{\text{accroissement vertical}} + \frac{1}{2N_H^{1/2}} \underbrace{(b_H(\underline{x} + \delta \underline{e}_2) - b_H(\underline{x}))}_{\text{accroissement horizontal}}, \quad (3.30)$$

où N_H est une constante de normalisation de la variance valant $|\delta|^{2H}(1 - 2^{H-2})$.

Comme énoncé à la Propriété 3.1, tous les accroissements possibles du champ brownien b_H sont stationnaires. Le choix de sommer deux accroissements *orthogonaux* est motivé par la volonté d'éviter les effets d'anisotropie lors du passage aux champs discrets observés à la Figure 3.2, c'est-à-dire lors de la synthèse des textures $X \in \mathbb{R}^{N_1 \times N_2}$ à analyser, tout en conservant la stationnarité.

Proposition 3.6. La covariance du champ gaussien g_H , défini à l'Équation (3.30), s'écrit

$$\begin{aligned} & \mathbb{E}[g_H(\underline{x})g_H(\underline{x}')] \\ &= \frac{\Sigma^2}{4N_H} \|\underline{x} - \underline{x}' + \delta \underline{e}_1\|^{2H} + \frac{\Sigma^2}{4N_H} \|\underline{x} - \underline{x}' - \delta \underline{e}_1\|^{2H} + \frac{\Sigma^2}{4N_H} \|\underline{x} - \underline{x}' + \delta \underline{e}_2\|^{2H} + \frac{\Sigma^2}{4N_H} \|\underline{x} - \underline{x}' - \delta \underline{e}_2\|^{2H} \\ & - \frac{\Sigma^2}{8N_H} \|\underline{x} - \underline{x}' + \delta \underline{e}_1 - \delta \underline{e}_2\|^{2H} - \frac{\Sigma^2}{8N_H} \|\underline{x} - \underline{x}' - \delta \underline{e}_1 + \delta \underline{e}_2\|^{2H} - \frac{3\Sigma^2}{4N_H} \|\underline{x} - \underline{x}'\|^{2H}. \end{aligned} \quad (3.31)$$

Dont on déduit la variance du champ g_H :

$$\mathbb{E}[g_H(\underline{x})^2] = \Sigma^2. \quad (3.32)$$

En outre, dans la limite des grandes échelles, i.e., $\|\underline{x} - \underline{x}'\| \gg |\underline{\delta}|$, la covariance du champ g_H se comporte comme

$$\mathbb{E}[g_H(\underline{x})g_H(\underline{x}')] \simeq \frac{\Sigma^2}{4N_H} \|\underline{x} - \underline{x}'\|^{2H-2} \quad (3.33)$$

$$\left(2\delta^2 + 4H(H-1) \left\langle \frac{\underline{x} - \underline{x}'}{\|\underline{x} - \underline{x}'\|}, \delta \underline{e}_1 \right\rangle^2 + 4H(H-1) \left\langle \frac{\underline{x} - \underline{x}'}{\|\underline{x} - \underline{x}'\|}, \delta \underline{e}_2 \right\rangle^2 - 2H(H-1) \left\langle \frac{\underline{x} - \underline{x}'}{\|\underline{x} - \underline{x}'\|}, \delta \underline{e}_1 - \delta \underline{e}_2 \right\rangle^2 \right). \quad (3.34)$$

Démonstration. La covariance du champ gaussien g_H s'exprime en fonction de celle du champ brownien b_H

$$\begin{aligned} & \mathbb{E}[g_H(\underline{x})g_H(\underline{x}')] \\ &= \frac{1}{4N_H} \mathbb{E}[(b_H(\underline{x} + \delta \underline{e}_1) + b_H(\underline{x} + \delta \underline{e}_2) - 2b_H(\underline{x})) (b_H(\underline{x}' + \delta \underline{e}_1) + b_H(\underline{x}' + \delta \underline{e}_2) - 2b_H(\underline{x}'))] \\ &= \frac{1}{4N_H} \mathbb{E}[b_H(\underline{x} + \delta \underline{e}_1)b_H(\underline{x}' + \delta \underline{e}_1)] + \frac{1}{4N_H} \mathbb{E}[b_H(\underline{x} + \delta \underline{e}_1)b_H(\underline{x}' + \delta \underline{e}_2)] - \frac{1}{2N_H} \mathbb{E}[b_H(\underline{x} + \delta \underline{e}_1)b_H(\underline{x}')] \\ &+ \frac{1}{4N_H} \mathbb{E}[b_H(\underline{x} + \delta \underline{e}_2)b_H(\underline{x}' + \delta \underline{e}_1)] + \frac{1}{4N_H} \mathbb{E}[b_H(\underline{x} + \delta \underline{e}_2)b_H(\underline{x}' + \delta \underline{e}_2)] - \frac{1}{2N_H} \mathbb{E}[b_H(\underline{x} + \delta \underline{e}_2)b_H(\underline{x}')] \\ &- \frac{1}{2N_H} \mathbb{E}[b_H(\underline{x})b_H(\underline{x}' + \delta \underline{e}_1)] - \frac{1}{2N_H} \mathbb{E}[b_H(\underline{x})b_H(\underline{x}' + \delta \underline{e}_2)] + \mathbb{E}[b_H(\underline{x})b_H(\underline{x}')]. \end{aligned} \quad (3.35)$$

En utilisant la structure de covariance (3.5) du champ brownien fractionnaire, on dérive la structure de covariance du champ gaussien g_H :

$$\begin{aligned} & \mathbb{E}[g_H(\underline{x})g_H(\underline{x}')] \\ &= \frac{\Sigma^2}{8N_H} (\|\underline{x} + \delta \underline{e}_1\|^{2H} + \|\underline{x}' + \delta \underline{e}_1\|^{2H} - \|\underline{x} - \underline{x}'\|^{2H} + \|\underline{x} + \delta \underline{e}_1\|^{2H} + \|\underline{x}' + \delta \underline{e}_2\|^{2H} - \|\underline{x} - \underline{x}' + \delta \underline{e}_1 - \delta \underline{e}_2\|^{2H} \\ &- 2\|\underline{x} + \delta \underline{e}_1\|^{2H} - 2\|\underline{x}'\|^{2H} + 2\|\underline{x} - \underline{x}' + \delta \underline{e}_1\|^{2H} + \|\underline{x} + \delta \underline{e}_2\|^{2H} + \|\underline{x}' + \delta \underline{e}_1\|^{2H} - \|\underline{x} - \underline{x}' + \delta \underline{e}_2 - \delta \underline{e}_1\|^{2H} \\ &+ \|\underline{x} + \delta \underline{e}_2\|^{2H} + \|\underline{x}' + \delta \underline{e}_2\|^{2H} - \|\underline{x} - \underline{x}'\|^{2H} \\ &- 2\|\underline{x} + \delta \underline{e}_2\|^{2H} - 2\|\underline{x}'\|^{2H} + 2\|\underline{x} - \underline{x}' + \delta \underline{e}_2\|^{2H} - 2\|\underline{x}\|^{2H} - 2\|\underline{x}' + \delta \underline{e}_1\|^{2H} + 2\|\underline{x} - \underline{x}' - \delta \underline{e}_1\|^{2H} \\ &- 2\|\underline{x}\|^{2H} - 2\|\underline{x}' + \delta \underline{e}_2\|^{2H} + 2\|\underline{x} - \underline{x}' - \delta \underline{e}_2\|^{2H} + 4\|\underline{x}\|^{2H} + 4\|\underline{x}'\|^{2H} - 4\|\underline{x} - \underline{x}'\|^{2H}) \\ &= \frac{\Sigma^2}{4N_H} \|\underline{x} - \underline{x}' + \delta \underline{e}_1\|^{2H} + \frac{\Sigma^2}{4N_H} \|\underline{x} - \underline{x}' - \delta \underline{e}_1\|^{2H} + \frac{\Sigma^2}{4N_H} \|\underline{x} - \underline{x}' + \delta \underline{e}_2\|^{2H} + \frac{\Sigma^2}{4N_H} \|\underline{x} - \underline{x}' - \delta \underline{e}_2\|^{2H} \\ &- \frac{\Sigma^2}{8N_H} \|\underline{x} - \underline{x}' + \delta \underline{e}_1 - \delta \underline{e}_2\|^{2H} - \frac{\Sigma^2}{8N_H} \|\underline{x} - \underline{x}' - \delta \underline{e}_1 + \delta \underline{e}_2\|^{2H} - \frac{3\Sigma^2}{4N_H} \|\underline{x} - \underline{x}'\|^{2H}. \end{aligned} \quad (3.36)$$

Le développement asymptotique de la covariance du champ g_H de l'Équation (3.31) dans la limite $\|\underline{x} - \underline{x}'\| \gg |\underline{\delta}|$ est obtenu de manière similaire par des calculs très similaires à ceux utilisés pour le développement de la covariance du champ f_H à la Propriété 3.3. L'expression asymptotique de la covariance est plus riche, car faisant intervenir les directions horizontale \underline{e}_1 , verticale \underline{e}_2 et diagonale $\underline{e}_1 - \underline{e}_2$:

$$\mathbb{E}[g_H(\underline{x})g_H(\underline{x}')] \simeq \frac{\Sigma^2}{4N_H} \|\underline{x} - \underline{x}'\|^{2H-2} \quad (3.37)$$

$$\left(2\delta^2 + 4H(H-1) \left\langle \frac{\underline{x} - \underline{x}'}{\|\underline{x} - \underline{x}'\|}, \delta \underline{e}_1 \right\rangle^2 + 4H(H-1) \left\langle \frac{\underline{x} - \underline{x}'}{\|\underline{x} - \underline{x}'\|}, \delta \underline{e}_2 \right\rangle^2 - 2H(H-1) \left\langle \frac{\underline{x} - \underline{x}'}{\|\underline{x} - \underline{x}'\|}, \delta \underline{e}_1 - \delta \underline{e}_2 \right\rangle^2 \right), \quad (3.38)$$

néanmoins la signature du comportement asymptotiquement $H - 1$ -auto similaire, i.e., le préfacteur $\|\underline{x} - \underline{x}'\|^{2H-2}$, est toujours présente ce qui permet de conclure la démonstration. \square

Propriétés 3.4. *Le champ gaussien g_H , défini à l'Équation (3.30) possède les propriétés suivantes :*

- (i) *Il est asymptotiquement $H - 1$ -auto similaire dans la limite des grandes échelles.*
- (ii) *Il est stationnaire.*
- (iii) *Il est isotrope.*

Démonstration. Ces trois propriétés découlent directement de l'expression de la covariance du champ gaussien g_H et de son comportement asymptotique, exposés à la Proposition 3.6. \square

Proposition 3.7 (Représentation harmonique). *Le champ gaussien g_H de la Définition 3.5 admet la représentation harmonique suivante :*

$$g_H(\underline{x}) = \frac{\Sigma^2}{2\sqrt{C_H N_H}} \int_{\mathbb{R}^2} \left(e^{-i\delta v_1} + e^{-i\delta v_2} - 2 \right) \frac{e^{-i\underline{x} \cdot \underline{v}}}{\|\underline{k}\|^{H+d/2}} \tilde{w}d(\underline{v}). \quad (3.39)$$

Proposition 3.8 (Régularité locale du champ g_H). *Dans la limite des petites échelles, i.e., $\|\underline{u}\| \ll |\delta|$, la fonction de structure d'ordre deux du champ g_H , défini à l'Équation (3.30), se comporte comme*

$$\mathbb{E} \left[(g_H(\underline{x} + \underline{u}) - g_H(\underline{x}))^2 \right] = \frac{3\Sigma^2}{2N_H} \|\underline{u}\|^{2H}. \quad (3.40)$$

Le champ g_H possède donc une régularité locale uniforme, égale à H .

Démonstration. Effectuons le développement asymptotique de la covariance de l'Équation (3.31) dans la limite $\|\underline{x} - \underline{x}'\| \ll |\delta|$. Pour cela, nous supposons, sans perte de généralité, que $\delta > 0$. Soit $\underline{u} \in \{\pm \underline{e}_1, \pm \underline{e}_2\}$ et $\underline{y} \triangleq (\underline{x} - \underline{x}')/\delta$, le développement de la fonction $t \mapsto (1 + t)^H$ au voisinage de $t = 0$ conduit à

$$\|\underline{x} - \underline{x}' + \delta \underline{u}\|^{2H} = \delta^{2H} \left(1 + 2 \langle \underline{y}, \underline{u} \rangle + \|\underline{y}\|^2 \right)^H \quad (3.41)$$

$$\simeq \delta^{2H} \left(1 + 2H \langle \underline{y}, \underline{u} \rangle + H \|\underline{y}\|^2 + 2H(H-1) \langle \underline{y}, \underline{u} \rangle^2 \right). \quad (3.42)$$

Pour $\underline{v} = \pm(\underline{e}_1 - \underline{e}_2)$, en posant $\underline{z} \triangleq (\underline{x} - \underline{x}')/(\sqrt{2}\delta)$, le développement s'écrit

$$\|\underline{x} - \underline{x}' + \delta \underline{v}\|^{2H} = \left(\sqrt{2}\delta \right)^{2H} \left(1 + 2 \langle \underline{z}, \underline{v} \rangle + \|\underline{z}\|^2 \right)^H \quad (3.43)$$

$$\simeq \left(\sqrt{2}\delta \right)^{2H} \left(1 + 2H \langle \underline{z}, \underline{v} \rangle + H \|\underline{z}\|^2 + 2H(H-1) \langle \underline{z}, \underline{v} \rangle^2 \right). \quad (3.44)$$

Réinjecter ce développement dans l'expression de la covariance de l'Équation (3.14) conduit à l'expression suivante

$$\mathbb{E} [g_H(\underline{x}) g_H(\underline{x}')] \simeq \frac{\Sigma^2}{N_H} \delta^{2H} \left(1 + H \|\underline{y}\|^2 + H(H-1) \left(\langle \underline{y}, \underline{e}_1 \rangle^2 + \langle \underline{y}, \underline{e}_2 \rangle^2 \right) \right) \quad (3.45)$$

$$- \frac{\Sigma^2}{8N_H} \left(\sqrt{2}\delta \right)^{2H} \left(2 + 2H \|\underline{y}\|^2 + 4H(H-1) \langle \underline{y}, \underline{e}_1 - \underline{e}_2 \rangle^2 \right) \quad (3.46)$$

$$- \frac{3\Sigma^2}{4N_H} \|\underline{x} - \underline{x}'\|^{2H} \quad (3.47)$$

puis, grâce à l'hypothèse $0 < H < 1$ on a $2H < 2$ et les termes quadratiques en $\|\underline{y}\|$ et $\|\underline{z}\|$ sont négligeables devant les termes $\|\underline{y}\|^{2H}$ et $\|\underline{z}\|^{2H}$ ce qui conduit au développement asymptotique de la covariance

$$\mathbb{E} [g_H(\underline{x}) g_H(\underline{x}')] \simeq \frac{\Sigma^2}{N_H} \delta^{2H} - 2 \frac{\Sigma^2}{8N_H} \left(\sqrt{2}\delta \right)^{2H} - \frac{3\Sigma^2}{4N_H} \|\underline{x} - \underline{x}'\|^{2H} \quad (3.48)$$

$$\simeq \frac{\Sigma^2}{N_H} \delta^{2H} - \frac{\Sigma^2}{N_H} \delta^{2H} 2^{2H-2} - \frac{3\Sigma^2}{4N_H} \|\underline{x} - \underline{x}'\|^{2H} \quad (3.49)$$

$$\simeq \Sigma^2 - \frac{3\Sigma^2}{4N_H} \|\underline{x} - \underline{x}'\|^{2H} \quad (3.50)$$

car $N_H \triangleq \delta^{2H}(1 - 2^{H-2})$. Ainsi, dans la limite $\|x - x'\| \ll \|\underline{\delta}\|$, la fonction de structure d'ordre deux du champ gaussien g_H proposé à l'Équation (3.30) s'écrit, pour tout $\underline{x} \in \mathbb{R}^D$ et pour tout $\underline{u} \in \mathbb{R}^D$,

$$\mathbb{E} \left[(g_H(\underline{x} + \underline{u}) - g_H(\underline{x}))^2 \right] = \frac{3\Sigma^2}{2N_H} \|\underline{u}\|^{2H}, \quad (3.51)$$

et la régularité locale du champ gaussien g_H est donc uniforme, égale à H . \square

Remarque. Le choix le plus naturel pour le pas intervenant dans la définition du champ g_H est $\delta = 1$, qui coïncide alors avec l'échelle la plus fine des coefficients d'ondelettes. En pratique, il n'est donc pas possible d'observer le comportement décrit à l'Équation 3.40, qui a lieu aux échelles infiniment petites $\|\underline{u}\| \ll |\delta|$. Néanmoins, à condition que $H - 1 + \gamma > 0$, en extrapolant le résultat de [19, Théorème 1], les coefficients d'ondelettes *dominants* de g_H , notés $\mathcal{L}_j[g_H](\underline{n})$ se comportent aux échelles les plus fines comme

$$\mathbb{E} \mathcal{L}_j[g_H](\underline{n}) \underset{2^j \rightarrow 0}{\lesssim} \eta(\underline{n}) 2^{j(H-1+\gamma)}, \quad (3.52)$$

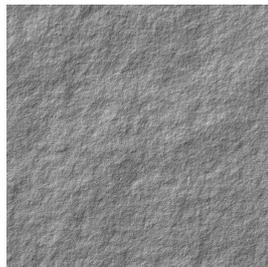
au point \underline{n} . En prenant un paramètre d'intégration fractionnaire $\gamma = 1$ dans la définition des coefficients d'ondelettes *dominants*, on remarque que l'Équation (3.52) s'écrit

$$\mathbb{E} \mathcal{L}_j[g_H](\underline{n}) \underset{2^j \rightarrow 0}{\lesssim} \eta(\underline{n}) 2^{jH}. \quad (3.53)$$

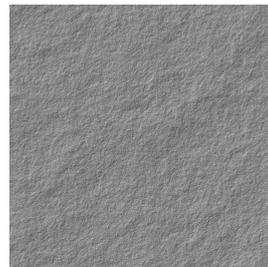
Le choix $\gamma \geq 1$ est ainsi nécessaire pour que la mesure de la régularité locale de g_H à l'aide des coefficients d'ondelettes dominants $\mathcal{L}_j[g_H](\underline{n})$, soit correcte [38, Section 3.3]. En fixant un paramètre d'intégration fractionnaire $\gamma \geq 1$, l'étude du comportement au travers les échelles de $\mathcal{L}_j[g_H](\underline{n})$ conduit à la mesure effective d'un exposant $H - 1 + \gamma$. Dans toute cette thèse le paramètre γ sera fixé à 1 et l'exposant mesuré *via* le comportement aux travers des échelles des coefficients d'ondelettes dominants sera donc H .

Définition 3.6 (Texture fractale homogène de paramètres (H, Σ^2)). Dans la suite de cette thèse, nous appellerons « texture fractale homogène de régularité locale H et de variance Σ^2 » tout échantillon statistique du champ gaussien stationnaire et asymptotiquement auto similaire g_H , construit en appliquant la Formule (3.13) à un champ brownien fractionnaire b_H (3.3), de paramètre d'auto-similarité H et de variance typique Σ^2 .

Deux exemples de textures fractales homogènes, i.e., de champ g_H , sont présentés à la Figure 3.3, pour $H = 0,7$ en Figure 3.3a et $H = 0,5$ en Figure 3.3b, sur lesquels on peut constater la disparition de l'effet d'anisotropie observé sur les réalisations de bruit gaussien fractionnaire des Figures 3.2c et 3.2b. La procédure de construction de ces textures homogènes sera détaillée à la Section 3.2.



(a) $H_1 = 0,7, \Sigma_1^2 = 1$



(b) $H_2 = 0,5, \Sigma_2^2 = 1$

FIGURE 3.3 – Exemple de champs gaussiens fractionnaires caractérisés par un exposant de Hölder H et une variance Σ^2 suivant le modèle de l'Équation (3.5).

3.2 Synthèse de champs gaussiens

Nous disposons, grâce à la Section 3.1, de plusieurs modèles de champs gaussiens *continus*, définis pour $\underline{x} \in \mathbb{R}^2$. D'une part, si ces définitions se prêtent aisément aux calculs théoriques, elles sont, dans la pratique, difficiles à utiliser. Et ce, notamment, à cause des intégrales sur des domaines infinis qu'elles requièrent. D'autre part, une image est intrinsèquement un objet *discret*, défini sur une grille de pixels. Il s'agit donc de passer de champs *continus* à des textures synthétiques *discrètes*, tout en préservant au mieux les propriétés d'auto-similarité et les structures de covariance décrites en Section 3.1. L'objectif final de cette section est de générer une discrétisation du champ gaussien stationnaire asymptotiquement auto similaire g_H introduit à l'Équation (3.13) qui permette d'obtenir les textures des Figures 3.3a et 3.3b.

Nous considérerons plus particulièrement le champ brownien fractionnaire $b_H(\underline{x})$, défini sur \mathbb{R}^2 par sa représentation harmonique (3.3), et dont la structure de covariance est donnée par la Formule (3.5) et nous intéresserons à la génération d'une discrétisation $B_{H,\Sigma}(\underline{n})$ de ce champs, caractérisée par le paramètre d'auto-similarité H et la variance typique Σ^2 , sur une grille de pixels $\underline{n} \in \Omega = \{1, \dots, N_1\} \times \{1, \dots, N_2\}$. Une application directe de la Formule (3.30) fournira alors une discrétisation du champ gaussien stationnaire asymptotiquement auto similaire proposé à la Section 3.1.4.

3.2.1 Méthode exacte par factorisation de Cholesky

Un champ gaussien discret $X \in \mathbb{R}^{N_1 \times N_2} \cong \mathbb{R}^N$ de moyenne nulle est complètement caractérisé par sa structure de covariance, stockée dans une matrice :

$$\mathbf{S} \triangleq \mathbb{E}[XX^T] \in \mathbb{R}^{N \times N} \quad (3.54)$$

Usuellement, une synthèse exacte de $B_{H,\Sigma}$ est possible en s'appuyant sur la décomposition de Cholesky de \mathbf{S} . En effet, \mathbf{S} étant symétrique définie positive, elle possède une factorisation

$$\mathbf{S} = \mathbf{L}\mathbf{L}^T \quad (3.55)$$

avec \mathbf{L} triangulaire supérieure [35, Théorème 11.2]. Ainsi, à partir d'un champ gaussien Z , indépendant, identiquement distribué, de moyenne nulle et normalisé en variance, en posant

$$X \triangleq LZ, \quad (3.56)$$

on obtient la structure de covariance souhaitée :

$$\mathbb{E}[XX^T] = \mathbb{E}[LZ(LZ)^T] = \mathbb{E}[LZZ^T L^T] = \underbrace{L\mathbb{E}[ZZ^T]L^T}_{=\mathbf{I}_N} = \mathbf{L}\mathbf{L}^T = \mathbf{S} \quad (3.57)$$

où \mathbf{I} désigne la matrice identité de taille N .

En l'absence de connaissances supplémentaires sur la matrice \mathbf{S} , la factorisation de Choleski s'obtient en $\mathcal{O}(N^3)$ opérations [8, Section 4.4]. Cette approche est donc trop coûteuse en temps de calcul pour être utilisée en pratique, notamment lorsqu'on s'intéresse à des champs bidimensionnels avec $N = N_1 \times N_2$ pixels.

Afin de pallier ce problème dans le cas où X est supposé *stationnaire*, une idée proposée dans [10], puis généralisée par [39, 12, 17] consiste à effectuer un plongement périodique de la matrice de covariance $\mathbf{S} \in \mathbb{R}^{N \times N}$ dans une matrice circulante de taille $2M$, $\mathbf{C} \in \mathbb{R}^{2M \times 2M}$. Ainsi, comme les vecteurs propres d'une matrice circulante sont les vecteurs de base de la transformée de Fourier discrète, \mathbf{C} est facilement diagonalisable et la décomposition de Cholesky de \mathbf{C} peut être obtenue avec un faible coût de calcul.

L'efficacité pratique de ces méthodes repose sur la détermination d'un plongement périodique $\mathbf{S} \rightarrow \mathbf{C}$ adéquat et n'engendrant pas d'augmentation trop importante de la dimension, c'est-à-dire avec $M \gtrsim N$. Une proposition de plongement pour le cas du bruit gaussien fractionnaire bidimensionnel a été élaborée par Stein *et coll.* [36], dans le cas où $0 < H < 0,75$. Afin de synthétiser des champs gaussiens stationnaires asymptotiquement auto-similaires caractérisés par un paramètre d'auto-similarité $0 < H < 1$ quelconque, il est nécessaire de se tourner vers des méthodes de synthèse approchées.

3.2.2 Méthodes approchées

De nombreuses procédures ont été proposées pour synthétiser de manière approchée un champ gaussien de structure de covariance donnée, par exemple (3.5) pour un champ brownien fractionnaire. On citera sans exhaustivité l'utilisation de filtres récursifs [7], de déplacements aléatoires du point médian [37], la méthode des bandes tournantes [40], la représentation sur une base d'ondelettes [1]. Ces méthodes sont approchées au sens où, puisqu'elles consistent en l'échantillonnage approximatif d'un champ continu, elle ne reproduisent pas *exactement* la structure de covariance de ce champ. Nous allons détailler dans cette section celle que nous avons privilégiée, utilisant la représentation harmonique du champ dans le domaine fréquentiel. Une seconde méthode [28], consistant à effectuer une convolution en intégrant par rapport à une mesure gaussienne, est présentée en Annexe B. La synthèse par convolution avec un noyau est notamment utilisée en physique, pour la modélisation du champ de vitesse dans les écoulements turbulents [30].

La méthode de synthèse de champs gaussiens à partir de leur densité spectrale est décrite dans [34], puis reprise par [3], qui propose également une comparaison avec d'autres méthodes. Pour nos travaux, nous avons pris comme base la description présentée par Biermé [4, 5, Annexe A.2], que nous reproduisons ici.

La définition du champ brownien fractionnaire (3.3), comme celle du drap brownien fractionnaire (3.9), sont construites *via* une représentation *harmonique*, i.e., dans le plan des fréquences, de ces champs. Il apparaît donc naturel de construire une discrétisation de ces champs à partir de leur transformée de Fourier.

Dans le cas du champ brownien fractionnaire b_H (3.3), la densité spectrale $\mathcal{T}_{H,\Sigma}(\underline{v}) \propto \|\underline{v}\|^{-H-1}$ est discrétisée, multipliée par un bruit blanc discret, et une transformée de Fourier discrète inverse permet de générer le champ discret $B_{H,\Sigma}$, comme décrit à l'Algorithme 3. Un exemple de synthèse d'un champ brownien de paramètre $H = 0,7$ est fourni en Figure 3.1a.

En utilisant la Définition 3.5 du champ gaussien stationnaire asymptotiquement auto-similaire, on déduit de la discrétisation $B_{H,\Sigma}(\underline{n})$ (3.62) du champ brownien fractionnaire $b_H(\underline{x})$, une discrétisation du champ g_H défini à l'Équation (3.30) :

$$G_{H,\Sigma}(\underline{n}) \triangleq \frac{1}{2N_H^{1/2}} (B_{H,\Sigma}(\underline{n} + \underline{e}_1) - B_{H,\Sigma}(\underline{n})) + \frac{1}{2N_H^{1/2}} (B_{H,\Sigma}(\underline{n} + \underline{e}_2) - B_{H,\Sigma}(\underline{n})). \quad (3.63)$$

Algorithme 3 Champ brownien fractionnaire par transformée de Fourier

Entrée : H régularité locale, Σ^2 variance typique

Paramètres : $N_1 \times N_2$ résolution

 {Bruit blanc de taille $N_1 \times N_2$ }

$$\tilde{W} \sim \mathcal{N}(0, \mathbf{I}) \in \mathbb{R}^{N_1 \times N_2} \quad (3.58)$$

 {Densité $\mathcal{T}_{H,\Sigma}$ échantillonnée sur une grille $\tilde{\Omega} = \{1, \dots, N_1\} \times \{1, \dots, N_2\}$ }

$$\forall \underline{q} \in \tilde{\Omega}, \quad T_{H,\Sigma}(\underline{q}) \triangleq \mathcal{T}_{H,\Sigma}(\underline{q}) = \frac{\Sigma}{C_H^{1/2} \|\underline{q}\|^{H+1}}, \quad (3.59)$$

$$\text{avec } C_H = \frac{\pi^{1/2} \Gamma(H+1/2)}{2H \Gamma(2H) \sin(\pi H) \Gamma(H+1)} \quad (3.60)$$

{Construction de la représentation du champ dans le domaine de Fourier}

$$\tilde{B}_{H,\Sigma} \triangleq \tilde{W} \cdot T_{H,\Sigma}, \quad (\tilde{W} \cdot T_{H,\Sigma})(\underline{q}) \triangleq \tilde{W}(\underline{q}) T_{H,\Sigma}(\underline{q}) \quad (3.61)$$

{Calcul du champ dans l'espace direct}

$$B_{H,\Sigma} \triangleq \text{ifft}(\tilde{B}_{H,\Sigma}) \quad (3.62)$$

Sortie : Discrétisation du champ brownien fractionnaire $B_{H,\Sigma}$

3.3 Textures fractales homogènes par morceaux

Afin de valider les méthodes de segmentation de texture que nous allons présenter dans cette thèse, il est nécessaire de produire des textures synthétiques, notées X , composées d'un assemblage de *textures fractales homogènes* (voir Définition 3.6), dont on connaît la segmentation sous-jacente

$$\Omega = \cup_{i=1}^k \Omega_i. \quad (3.64)$$

La texture X sera construite comme la discrétisation d'un champ gaussien $f : Y \subset \mathbb{R}^2 \rightarrow \mathbb{R}$, dont la segmentation sous-jacente sera notée

$$Y = \cup_{i=1}^k Y_i, \quad (3.65)$$

composé de *patches* de champs gaussiens *stationnaires* asymptotiquement auto similaires (voir Définition 3.5). C'est-à-dire tel qu'il existe g_1, \dots, g_k des champs gaussiens stationnaires asymptotiquement auto similaires, de sorte que le champ f restreint au domaine Y_i coïncide avec le champ g_i :

$$f|_{Y_i} \equiv g_i. \quad (3.66)$$

De plus, on souhaite que le champ g_i soit une texture fractale homogène de paramètres (H_i, Σ_i^2) (voir Définition 3.6).

Remarque. La stationnarité des champs g_1, \dots, g_k permet d'éviter les sauts de variance à la frontière entre deux textures homogènes qui rendraient le problème de segmentation trivial et peu réaliste, sans avoir besoin de procédure sophistiquée de raccordement.

Le réglage de la régularité locale H_i et de la variance locale Σ_i^2 des champs g_i permet non seulement d'obtenir des textures d'apparences variées mais aussi de produire des textures différentes

dans leur structure de covariance mais dont le rendu visuel est très similaire, comme illustré dans les exemples de la Figure 3.4. Ces textures correspondent en effet à des problèmes de segmentation difficile pour l'œil humain, pour lesquels les méthodes développées dans cette thèse montreraient tout leur intérêt. Le but de cette section est de détailler la synthèse pratique d'un tel champ gaussien f , amenant à la génération de *textures fractales homogènes par morceaux*, notées X .

On se donne k couples d'attributs $(H_i, \Sigma_i^2)_{i=1}^k$, qui vont caractériser les k textures différentes qui constitueront l'image X . On se donne également un *masque*, c'est-à-dire une segmentation sous-jacente

$$\Omega = \cup_{i=1}^k \Omega_i. \quad (3.67)$$

La synthèse de texture fractale homogène par morceaux se fait en trois temps. Tout d'abord k champs browniens fractionnaires discrets B_{H_i} , de régularité locale H_i et de *variance typique* Σ_i^2 sont générés grâce à la procédure décrite à l'Algorithme 3. Le champ discret gaussien stationnaire G_i , correspondant à un échantillon de la texture i , est construit en appliquant au champ B_{H_i} la formule des accroissements horizontaux et verticaux, proposée à l'Équation (3.30). Enfin, l'image X est construite en assemblant les textures G_i selon le découpage (3.67) :

$$\underline{n} \in \Omega_i \implies X(\underline{n}) \equiv G_i(\underline{n}). \quad (3.68)$$

La procédure complète est résumée à l'Algorithme 4. Trois exemples de textures homogènes par morceaux, suivant la géométrie elliptique de la Figure 3.4a sont proposés en Figure 3.4, illustrant en Figure 3.4b un changement de variance Σ^2 , en Figure 3.4c un changement de régularité H et en Figure 3.4d un changement simultané de variance et de régularité.

Algorithme 4 Synthèse de texture fractale homogène par morceaux

Entrée : k nombre de régions

$\Omega = \cup_{i=1}^k \Omega_i$ segmentation sous-jacente

$(H_i)_{i=1}^k$ régularités locales

$(\Sigma_i^2)_{i=1}^k$ variances typiques

Paramètres : $N_1 \times N_2$ résolution

for $i = 1$ **to** k **do**

1. Générer le champ brownien discret B_{H_i, Σ_i} à partir de l'Algorithme 3

2. Calculer les accroissements horizontaux et verticaux (3.30) avec $\delta = 1$

$$\forall \underline{n} \in \Omega, \quad G_i(\underline{n}) = \frac{1}{2N_H^{1/2}} (B_{H_i, \Sigma_i}(\underline{n} + \underline{e}_1) - B_{H_i, \Sigma_i}(\underline{n})) + \frac{1}{2N_H^{1/2}} (B_{H_i, \Sigma_i}(\underline{n} + \underline{e}_2) - B_{H_i, \Sigma_i}(\underline{n}))$$

3. Placer un *patch* de la texture i sur la région Ω_i

$$\forall \underline{n} \in \Omega_i, \quad X(\underline{n}) = G_i(\underline{n})$$

end for

Sortie : Texture fractale homogène par morceaux X

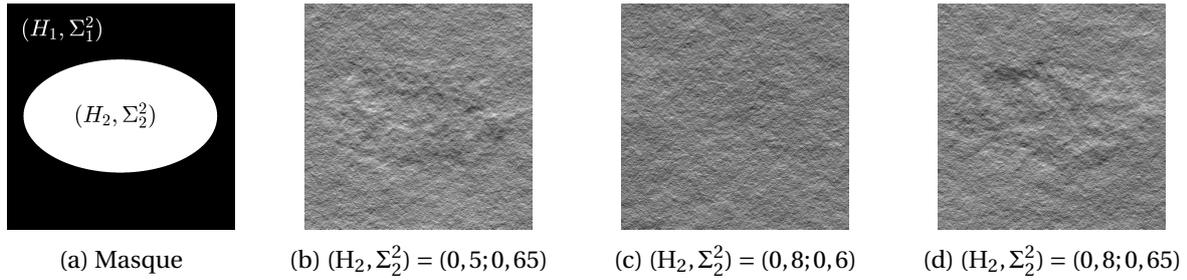


FIGURE 3.4 – **Textures fractales par morceaux à $k = 2$ régions.** Texture de fond caractérisée par $(H_1, \Sigma_1^2) = (0,5;0,6)$ fixés. Texture centrale caractérisée par différentes régularités et variances (H_2, Σ_2^2) : (b) changement de variance, (c) changement de régularité, (d) changement de régularité et de variance.

3.4 Bilan

Nous avons proposé un modèle théorique (3.30) de champ gaussien stationnaire asymptotiquement auto similaire, défini à partir des accroissements horizontaux et verticaux d'un champ brownien fractionnaire. La calcul de la covariance (3.31) de ce champ gaussien g_H a montré que pour tout paramètre d'auto-similarité $0 < H < 1$, le champ g_H est stationnaire à l'ordre deux. De plus, nous avons montré que le champ g_H ainsi construit possède une régularité locale uniforme, $h(x) \equiv H$.

Puis, une méthode de synthèse approchée d'un tel champ g_H , à partir de la discrétisation du champ brownien fractionnaire b_H , exposée et résumée par l'Algorithme 3, est explicitée à l'Équation (3.63).

Enfin, l'assemblage de *patches* de textures homogènes caractérisées par leur régularité locale et leur variance, décrit à l'Algorithme 4, a conduit à la construction de textures synthétiques fractales homogènes par morceaux, dont la segmentation sous-jacente $\Omega = \cup_{i=1}^k \Omega_i$ et les attributs caractéristiques $(H_i, \Sigma_i^2)_{i=1}^k$ sont fixés librement par l'utilisateur.

Références

- [1] P. Abry and F. Sellan. The wavelet-based synthesis for fractional Brownian motion proposed by F. Sellan and Y. Meyer : Remarks and fast implementation, 1996.
- [2] P. Abry, H. Wendt, and S. Jaffard. When Van Gogh meets Mandelbrot : Multifractal classification of painting's texture. *Signal Processing*, 93(3) :554–572, 2013. 68
- [3] J.-M. Bardet, G. Lang, G. Oppenheim, A. Philippe, and M. S. Taqqu. Generators of long-range dependent processes : a survey. *Theory and applications of long-range dependence*, pages 579–623, 2003. 79
- [4] H. Biermé. *Champs aléatoires : autosimilarité, anisotropie et étude directionnelle*. PhD thesis, Université d'Orléans, 2005. 79
- [5] H. Biermé, M. Meerschaert, and H. Scheffle. Operator scaling stable random fields. *Stoch. Process. Their Appl*, 117(3) :312–332, 2007. 74, 79
- [6] A. Bonami and A. Estrade. Anisotropic analysis of some Gaussian models. *J. Fourier Anal. Appl.*, 9(3) :215–236, 2003. 69
- [7] L. M. Braton and N. R. Bartley. Simulation of fractal multidimensional images using multidimensional recursive filter. *IEEE Trans. Circuits Syst.*, 41 :181–188, 1994. 79

- [8] P. G. Ciarlet. *Introduction à l'analyse numérique matricielle et à l'optimisation*. Dunod, 2007. 78
- [9] S. Cohen and J. Istas. *Fractional fields and applications*. Springer, 2013. 68, 69, 70
- [10] R. B. Davies and D. S. Harte. Tests for hurst effect. *Biometrika*, 74(1) :95–101, 1987. 78
- [11] G. Didier, M. M. Meerschaert, and V. Pipiras. Domain and range symmetries of operator fractional Brownian fields. *Stoch. Process. Their Appl*, 128(1) :39–78, 2018. 74
- [12] C. R. Dietrich and G. N. Newsam. Fast and exact simulation of stationary gaussian processes through circulant embedding of the covariance matrix. *SIAM J. Sci. Comput.*, 18(4) :1088–1107, 1997. 78
- [13] M. Doret, J. Spilka, V. Chudáček, P. Gonçalves, and P. Abry. Fractal analysis and hurst parameter for intrapartum fetal heart rate variability analysis : a versatile alternative to frequency bands and LF/HF ratio. *PLOS One*, 10(8), 2015. 68
- [14] K. Falconer. *Fractal geometry : mathematical foundations and applications*. John Wiley & Sons, 2004. 68
- [15] R. Fontugne, P. Abry, K. Fukuda, D. Veitch, K. Cho, P. Borgnat, and H. Wendt. Scaling in internet traffic : a 14 year and 3 day longitudinal study, with multiscale analyses and random projections. *IEEE ACM Trans. Netw.*, 25(4) :2152–2165, 2017. 68
- [16] Y. Gagne. *Etude expérimentale de l'intermittence et des singularités dans le plan complexe en turbulence développée*. PhD thesis, Université Joseph Fourier (Grenoble), 1987. 68
- [17] H. Helgason, V. Pipiras, and P. Abry. Fast and exact synthesis of stationary multivariate Gaussian time series using circulant embedding. *Signal Process.*, 91(5) :1123–1133, 2011. 78
- [18] J. E. Hutchinson. Fractals and self similarity. *Indiana Univ. Math. J.*, 30(5) :713–747, 1981. 68
- [19] S. Jaffard. Wavelet techniques in multifractal analysis. *Fractal Geometry and Applications : A Jubilee of Benoit Mandelbrot, M. Lapidus et M. van Frankenhuysen Eds, Proceedings of Symposia in Pure Mathematics (AMS)*, 72(2) :91–152, 2004. 77
- [20] R. Jennane, W. J. Ohley, S. Majumdar, and G. Lemineur. Fractal analysis of bone x-ray tomographic microscopy projections. *IEEE Trans. Med. Imag.*, 20(5) :443–449, 2001. 68
- [21] C. R. Johnson, P. Messier, W. A. Sethares, A. G. Klein, C. Brown, A. H. Do, P. Klausmeyer, P. Abry, S. Jaffard, and H. Wendt. Pursuing automated classification of historic photographic papers from raking light images. *Journal of the American Institute for Conservation*, 53(3) :159–170, 2014. 68
- [22] M. Jung, G. Peyré, and L. D. Cohen. Texture segmentation via non-local non-parametric active contours. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 74–88. Springer, 2011. 70
- [23] C. Lacaux. *Champs de Lévy multifractionnaires*. PhD thesis, Toulouse 3, 2004. 69
- [24] S. Léger. *Analyse stochastique de signaux multi-fractaux et estimations de paramètres*. PhD thesis, Orléans, 2000. 70
- [25] B. Mandelbrot. Survey of multifractality in finance. Technical report, Cowles Foundation for Research in Economics, Yale University, 1999. 68
- [26] B. B. Mandelbrot. Intermittent turbulence in self-similar cascades : divergence of high moments and dimension of the carrier. *Journal of fluid Mechanics*, 62(2) :331–358, 1974. 68

- [27] M. Oliver, G. Haro, V. Fedorov, and C. Ballester. L1 patch-based image partitioning into homogeneous textured regions. In *Proc. Int. Conf. Acoust. Speech Signal Process.*, pages 1558–1562. IEEE, 2018. 70
- [28] R. M. Pereira, C. Garban, and L. Chevillard. A dissipative random velocity field for fully developed fluid turbulence. *J. Fluid Mech.*, 794 :369–408, 2016. 79
- [29] V. Pipiras and M. S. Taqqu. *Long-range dependence and self-similarity*, volume 45. Cambridge university press, 2017. 68, 71
- [30] J. Reneuve and L. Chevillard. Flow of spatiotemporal turbulentlike random fields. *Preprint arXiv :2004.02864*, 2020. 79
- [31] S. Roux, A. Arneodo, and N. Decoster. A wavelet-based method for multifractal image analysis. III. Applications to high-resolution satellite images of cloud structure. *The European Physical Journal B-Condensed Matter and Complex Systems*, 15(4) :765–786, 2000. 68
- [32] S. Roux, M. Clausel, B. Vedel, S. Jaffard, and P. Abry. Self-Similar Anisotropic Texture Analysis : the Hyperbolic Wavelet transform contribution. *IEEE Trans. Image Process.*, 22(11) :4353 – 4363, 2013. 74
- [33] G. Samorodnitsky, M. S. Taqqu, and R. W. Linde. *Stable non-gaussian random processes : stochastic models with infinite variance*. New York. Chapman & Hall, 1994. 68, 69, 71
- [34] D. Saupe. Algorithms for random fractals. In *The Science of Fractal Images*, pages 71–136. Springer, 1988. 79
- [35] D. Serre. Matrices : Theory and applications. *Graduate texts in mathematics*, 2000. 78
- [36] M. L. Stein. Fast and exact simulation of fractional brownian surfaces. *J. Comput. Graph. Stat.*, 11(3) :587–599, 2002. 79
- [37] M. A. Stoksik, R. G. Lane, and D. T. Nguyen. Practical synthesis of accurate fractal images. *Graph. Models and Image Process.*, 57(3) :206–219, 1995. 79
- [38] H. Wendt, S. G. Roux, S. Jaffard, and P. Abry. Wavelet leaders and bootstrap for multifractal analysis of images. *Signal Processing*, 89(6) :1100–1114, 2009. 77
- [39] A. T. A. Wood and G. Chan. Simulation of stationary gaussian processes in $[0, 1]$ d. *J. Comput. Graph. Stat.*, 3(4) :409–432, 1994. 78
- [40] Z.-M. Yin. New methods for simulation of fractional Brownian motion. *J.of Comput. Phys.*, 127(1) :66–72, 1996. 79

Chapitre 4

Segmentation de textures fractales par minimisation convexe

« Il est temps que la pierre se résolve enfin à fleurir. »

P. Celan

Sommaire

4.1 Motivations pour l'estimation et la segmentation conjointe	86
4.2 Construction de fonctionnelles pour la segmentation de texture	87
4.2.1 Ajustement au modèle	87
4.2.2 Propositions de pénalisations construites à partir de la Variation Totale	88
4.3 Optimisation convexe non lisse	89
4.3.1 Sous-différentiabilité	90
4.3.2 Dualité de Fenchel-Rockafellar	91
4.3.3 Algorithmes proximaux	92
4.3.4 Forte-convexité	93
4.4 Minimisation de fonctionnelles pour la segmentation de texture	94
4.4.1 Propriétés de l'attache aux données	95
4.4.2 Pénalisations à contours « libres » et à contours « co-localisés »	98
4.4.3 Formulation des problèmes duaux	99
4.4.4 <i>Fast Iterative Soft Thresholding Algorithm (FISTA)</i> , version duale	100
4.4.5 Algorithme primal-dual accéléré grâce à la forte-convexité	101
4.4.6 Critères d'arrêt	104
4.5 Performance d'estimation et de segmentation	106
4.5.1 Stratégie de seuillage	106
4.5.2 Protocole de comparaison	106
4.5.2.1 Échantillons de textures synthétiques	106
4.5.2.2 Transformée en ondelettes	107
4.5.2.3 Paramètres des algorithmes de minimisation	108
4.5.2.4 Mesures de performance	108
4.5.2.5 Choix des paramètres de régularisation	109
4.5.3 ROF_h -S <i>versus</i> segmentation à contours « libres » ou à contours « co-localisés »	109
4.5.3.1 Performances	109
4.5.3.2 Coûts de calcul	111
4.5.3.3 Bilan	111
4.5.4 Segmentation d'assemblages de textures réelles	112
4.6 Apprentissage profond pour la segmentation de textures fractales	113
4.6.1 Mise en œuvre	114

4.6.1.1	Architectures proposées	114
4.6.1.2	Problèmes considérés et bases de données d'entraînement	115
4.6.1.3	Entraînement	116
4.6.2	Segmentation de textures synthétiques à contours aléatoires	117
4.6.2.1	Jeux de données tests et procédures de comparaison	117
4.6.2.2	Performances	118
4.6.2.3	Robustesse	119
4.6.2.4	Estimation d'interfaces	119
4.6.2.5	Complexité et effort de calcul	121
4.6.3	Bilan de l'apprentissage supervisé pour la segmentation de textures fractales	121

Les travaux présentés dans cette section ont fait l'objet de plusieurs publications, sous la forme d'un article long [27] soumis au *Journal of Mathematical Imaging and Vision* et dans le cadre de conférences internationales [26, 28].

L'entraînement des réseaux de neurones pour les méthodes d'apprentissage statistique présentées à la Section 4.6 a été réalisée avec le soutien du Centre Blaise Pascal¹ grâce au système SIDUS (Single Instance Distributing Universal System) élaboré par Emmanuel Quemener.

4.1 Motivations pour l'estimation et la segmentation conjointe

La méthode $\text{ROF}_h\text{-S}$, présentée au Chapitre 2, à la Section 2.3.4, réalisant un débruitage par Variation Totale de l'estimée de la régularité locale par régression linéaire \hat{h}^{RL} présente plusieurs limitations. D'une part, l'erreur d'estimation, provenant de l'estimateur par minimisation des moindres carrés, se cumule avec le biais dû à la régularisation par Variation Totale [16]. De plus, l'ordonnée à l'origine \hat{v}^{RL} , obtenue lors de la régression linéaire, n'est pas utilisée. Enfin, la segmentation obtenue, présentée en Figure 2.14e, présente des contours irréguliers, alors même que le contour elliptique imposé par le masque 2.14a était très lisse.

Afin de pallier ces insuffisances, nous proposons deux méthodes *conjointes*, qui réalisent simultanément l'estimation et la régularisation. Dans la mise œuvre de ces deux méthodes, la *puissance locale* $\nu(\underline{n})$, définie comme l'ordonnée à l'origine des logarithmes des coefficients dominants, apparaît naturellement comme un attribut auxiliaire à la régularité locale, mesurée par la pente des logarithmes des coefficients dominants. Ainsi, il sera possible de détecter un changement de variance entre deux textures de même régularité locale, grâce à sa répercussion sur la puissance locale $\nu(\underline{n})$. Deux fonctionnelles globales, dites à contours « libres » et à contours « colocalisés », sont construites. Elles se composent toutes les deux d'une attache aux données modélisant le comportement linéaire des logarithmes des coefficients dominants au travers des échelles et d'une pénalisation favorisant la constance par morceaux. Deux pénalisations distinctes sont construites et comparées. La pénalisation dite à contours « libres » favorise indépendamment la constance par morceaux de la régularité et de la puissance locales, tandis que la pénalisation *colocalisée* est construite de manière à favoriser des contours plus lisses à la frontière entre deux textures, en imposant une contrainte de constance par morceaux plus forte.

La Section 4.2 détaille la construction des deux fonctionnelles proposées. La Section 4.3 introduit les principes et outils d'analyse convexe qui seront utilisés pour l'implémentation de schémas numériques efficaces pour la minimisation des fonctionnelles construites. Puis, les Sections 4.4.1 et 4.4.2 s'attachent à étudier en détail les propriétés des fonctionnelles proposées, et nous démontrons notamment la forte-convexité de l'attache aux données. Grâce à ce résultat, deux types d'algorithmes proximaux accélérés sont dérivés aux Sections 4.4.4 et 4.4.5 pour la résolution des Problèmes (4.5) et (4.7). En outre, nous proposons à la Section 4.4.6 un critère d'arrêt inspiré de l'écart

1. <http://www.cbp.ens-lyon.fr/>

de dualité (*duality gap*), permettant d'arrêter les algorithmes à un niveau similaire d'approximation du minimum. Afin d'obtenir non seulement des estimées régularisées, mais également des segmentations à k régions, un seuillage itératif adapté est décrit à l'Algorithme 11 puis appliqué aux méthodes de segmentation à contours « libres » et à contours « co-localisés » proposées. Les performances des méthodes conjointes proposées sont évaluées sur des textures synthétiques à la Section 4.5, par le biais de simulations de type Monte Carlo, et comparées à la méthode ROF $_{h-S}$ de la Section 2.3.4, puis comparées à une méthode de l'état-de-l'art sur des assemblages de textures réelles en Section 4.5.4. L'effort de calcul requis par chaque type d'algorithme proximal est mesuré et comparé en Section 4.5.3.2, afin de sélectionner, pour chaque problème, la méthode de résolution la plus efficace. Ces travaux ont fait l'objet de plusieurs publications [26, 27]. Enfin, l'apprentissage statistique par réseaux de neurones profond faisant état de succès très nombreux ces dernières années, nous proposons à la Section 4.6 une procédure adaptée au problème de segmentation de texture, inspirée par [1], et discutons ses avantages et inconvénients en comparaison des méthodes non-supervisées reposant sur la régularité locale.

4.2 Construction de fonctionnelles pour la segmentation de texture

En traitement d'images, les approches variationnelles consistent à définir une fonctionnelle qui réalise un ajustement prenant en compte plusieurs contraintes. Typiquement, elles réalisent un compromis entre la fidélité aux observations/données disponibles et des contraintes de régularité, attendues *a priori*. Les résultats d'estimation et de segmentation dépendent alors de la construction de la fonctionnelle. Pour réaliser en *une seule étape* l'estimation des attributs de régularité et de puissance locales deux fonctionnelles sont proposées, appelées :

- À contours « libres » (*joint* en anglais dans les articles).
- À contours « co-localisés » (*coupled* en anglais dans les articles).

Remarque. Dans les publications scientifiques anglophones liées au travail effectué dans cette thèse, *joint* fait référence à la fonctionnelle à contours « libres » et *conjoint* à la fonctionnelle à contours « co-localisés », décrites dans la suite. En français dans cette thèse les adjectifs *joint* ou *conjoint* seront utilisés indistinctement pour souligner le caractère *simultané* de l'estimation des attributs et de leur régularisation favorisant la segmentation.

4.2.1 Ajustement au modèle

Rappelons que le comportement local du logarithme des coefficients d'ondelettes *dominants* au travers des échelles, formulé à l'Équation (2.88), s'écrit

$$\log_2(\mathcal{L}_j(\underline{n})) \underset{2^j \rightarrow 0}{\simeq} jh(\underline{n}) + v(\underline{n}), \quad (4.1)$$

où $h(\underline{n})$ désigne la régularité locale de l'image étudiée au pixel \underline{n} et $v(\underline{n})$ est la *puissance locale* de l'image au pixel \underline{n} .

Contrairement à la méthode ROF $_{h-S}$, présentée à la Section 2.3.4, où on réalise tout d'abord la régression linéaire au travers des échelles de la Proposition 2.3, suivie d'une régularisation *a posteriori* de l'estimée de la régularité locale \hat{h}^{RL} , nous proposons d'inclure dans une fonctionnelle régularisante le comportement des coefficients dominants (4.1) grâce au terme de moindres carrés défini ci-dessous.

Définition 4.1 (Moindres carrés pour $\log_2 \mathcal{L}$). Pour une gamme d'échelles $j \in \{j_1, \dots, j_2\}$, la fonctionnelle des moindres carrés est définie par

$$F(\mathbf{h}, \mathbf{v}; \log_2 \mathcal{L}) = \frac{1}{2} \sum_{j=j_1}^{j_2} \|j\mathbf{h} + \mathbf{v} - \log_2 \mathcal{L}_j\|_{\text{Fro}}^2, \quad (4.2)$$

où toutes les quantités locales sont stockées sous forme de cartes à $N_1 \times N_2$ pixels :

- $\mathbf{h} \in \mathbb{R}^{N_1 \times N_2}$ désigne la *régularité locale*, définie en tout pixel $\underline{n} \in \Omega$ comme la pente de $\log_2 \mathcal{L}_j$,
- $\mathbf{v} \triangleq \log_2 \boldsymbol{\eta} \in \mathbb{R}^{N_1 \times N_2}$ désigne la *puissance locale*, définie en tout pixel $\underline{n} \in \Omega$ comme l'ordonnée à l'origine de $\log_2 \mathcal{L}_j$,
- $\|\cdot\|_{\text{Fro}}^2$ désigne la somme des carrés des coefficients d'un élément de $\mathbb{R}^{N_1 \times N_2}$, appelée norme de Frobenius.
- $\mathcal{L}_j \in \mathbb{R}^{N_1 \times N_2}$ désigne la carte des coefficients dominants à l'échelle 2^j .

Il sera utile dans la suite d'écrire ce terme de moindres carrés de manière plus compacte grâce à l'introduction d'un opérateur linéaire

$$\Phi \triangleq \begin{cases} \mathbb{R}^{N_1 \times N_2} \times \mathbb{R}^{N_1 \times N_2} & \longrightarrow \mathbb{R}^{N_j \times N_1 \times N_2} \\ (\mathbf{h}, \mathbf{v}) & \longmapsto \{\mathbf{j}\mathbf{h} + \mathbf{v}\}_{j_1=1}^{j_2} \end{cases} \quad (4.3)$$

avec $N_j = j_2 - j_1 + 1$ le nombre d'échelles considérées, ce qui amène à la réécriture

$$F(\mathbf{h}, \mathbf{v}; \log_2 \mathcal{L}) = \frac{1}{2} \|\Phi(\mathbf{h}, \mathbf{v}) - \log_2 \mathcal{L}\|_{\text{Fro}}^2, \quad (4.4)$$

où $\log_2 \mathcal{L}$ consiste en la concaténation des logarithmes des coefficients *dominants* à toutes les échelles 2^{j_1} à 2^{j_2} .

Comme souligné à la Section 2.3.4, l'estimateur de régularité locale par régression linéaire, qui consiste à minimiser F , possède une variance trop élevée et les cartes de régularité locale obtenues sont extrêmement bruitées. Il est donc nécessaire d'ajouter au terme de moindres carrés de l'Équation (4.2) un terme de pénalisation afin obtenir des estimées régularisées.

4.2.2 Propositions de pénalisations construites à partir de la Variation Totale

D'après le modèle mathématique de texture fractale homogène par morceaux la régularité locale $\bar{\mathbf{h}}$ et la puissance locale $\bar{\mathbf{v}}$ théoriques sont *constantes par morceaux*. Cette connaissance *a priori* est incluse dans la procédure d'estimation et de régularisation simultanées par l'introduction d'une pénalisation, dont la construction s'appuie sur la Variation Totale, introduite à l'Équation (2.45), en Section 2.3.4.

Deux stratégies différentes par la construction de leur pénalisation sont proposées :

(i) *Fonctionnelle à pénalisation « libre »* :

$$(\hat{\mathbf{h}}^L, \hat{\mathbf{v}}^L) = \underset{\mathbf{h}, \mathbf{v} \in \mathbb{R}^{N_1 \times N_2}}{\operatorname{argmin}} F(\mathbf{h}, \mathbf{v}; \log_2 \mathcal{L}) + G^L(\mathbf{h}, \mathbf{v}), \quad (4.5)$$

$$\text{avec } G^L(\mathbf{h}, \mathbf{v}) = \lambda (\alpha \text{TV}(\mathbf{h}) + \text{TV}(\mathbf{v})), \quad (4.6)$$

qui favorise indépendamment le caractère constant par morceaux de \mathbf{h} et \mathbf{v} ;

(ii) *Fonctionnelle à pénalisation « co-localisée »* :

$$(\hat{\mathbf{h}}^C, \hat{\mathbf{v}}^C) = \underset{\mathbf{h}, \mathbf{v} \in \mathbb{R}^{N_1 \times N_2}}{\operatorname{argmin}} F(\mathbf{h}, \mathbf{v}; \log_2 \mathcal{L}) + G^C(\mathbf{h}, \mathbf{v}), \quad (4.7)$$

$$\text{avec } G^C(\mathbf{v}, \mathbf{h}) = \lambda \text{TV}_\alpha(\mathbf{h}, \mathbf{v}) \quad (4.8)$$

où TV_α couple les variations spatiales de \mathbf{h} et de \mathbf{v} *via*

$$\text{TV}_\alpha(\mathbf{h}, \mathbf{v}) \triangleq \sum_{\underline{n} \in \Omega} \sqrt{\alpha^2 (\mathbf{V}\mathbf{h})(\underline{n})^2 + \alpha^2 (\mathbf{H}\mathbf{h})(\underline{n})^2 + (\mathbf{V}\mathbf{v})(\underline{n})^2 + (\mathbf{H}\mathbf{v})(\underline{n})^2} \quad (4.9)$$

et, par conséquent, favorise des changements de régularité et de puissance locales localisés aux mêmes pixels.

Les hyperparamètres $\lambda > 0$ et $\alpha > 0$ sont des *paramètres de régularisation*, dont le réglage pilote l'allure générale des estimées obtenues. Le paramètre λ permet de régler le compromis entre la fidélité au modèle mathématique, *via* le terme de moindres carrés $F(\mathbf{h}, \mathbf{v}; \log_2 \mathcal{L})$, et la contrainte de constance par morceaux, *via* les pénalisations basées sur la Variation Totale. Quant au paramètre α , il permet de gérer l'importance relative des sauts de régularité et de puissance locales.

Exemple 4.1. Prenons un exemple élémentaire pour illustrer la différence entre ces deux pénalisations. La Figure 4.1 présente deux cartes binaires \mathbf{h} et \mathbf{v} , composées de 2×2 pixels. Dans le cas de la Figure 4.1a, les cartes \mathbf{h} et \mathbf{v} ont chacune un contour (en rouge) de longueur 2, qui correspond à des sauts de même amplitude en valeur absolue, mais ces contours ne sont pas localisés aux mêmes endroits. Au contraire, à la Figure 4.1b, les contours sont localisés sur les mêmes arrêtes de la grille de pixels, sans néanmoins que les images soient identiques.

Fixons $\lambda = \alpha = 1$ et calculons, pour chacun des exemples (a) et (b) de la Figure 4.1, les pénalisations G^L et G^C :

(i) *Pénalisation « libre » :*

$$\text{Contours disjoints : } G^L(\mathbf{h}, \mathbf{v}) = \sqrt{(-1)^2} + \sqrt{(-1)^2} + \sqrt{1^2} + \sqrt{1^2} = 4.$$

$$\text{Contours communs : } G^L(\mathbf{h}, \mathbf{v}) = \sqrt{1^2} + \sqrt{1^2} + \sqrt{(-1)^2} + \sqrt{(-1)^2} = 4.$$

(ii) *Pénalisation « co-localisée » :*

$$\text{Contours disjoints : } G^C(\mathbf{h}, \mathbf{v}) = \sqrt{(-1)^2 + 1^2} + \sqrt{1^2} + \sqrt{(-1)^2} = \sqrt{2} + 2$$

$$\text{Contours communs : } G^C(\mathbf{h}, \mathbf{v}) = \sqrt{1^2 + (-1)^2} + \sqrt{1^2 + (-1)^2} = 2\sqrt{2}.$$

Tandis que la pénalisation « libre » ne fait aucune discrimination entre contours disjoints et contours communs, la pénalisation « co-localisée » prend une valeur plus faible lorsque les contours sont partagés par les deux cartes \mathbf{h} et \mathbf{v} . Ainsi, la minimisation de la fonctionnelle à pénalisation « co-localisée » favorise des estimées de régularité et de puissance locales non seulement constantes par morceaux, mais également partageant la même segmentation sous-jacente.

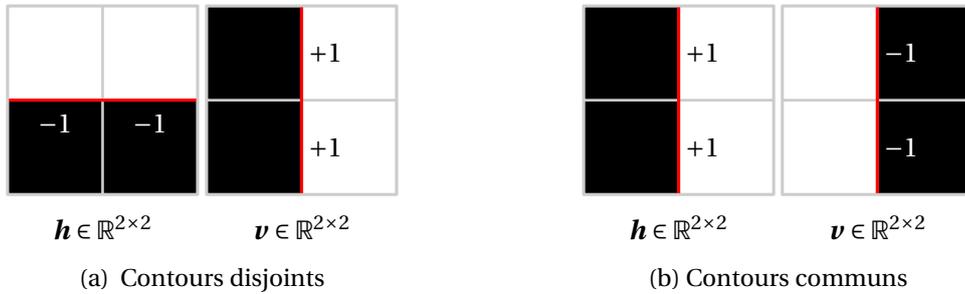


FIGURE 4.1 – Régularité et puissance locales avec des contours disjoints ou communs.

4.3 Optimisation convexe non lisse

Disposant de deux fonctionnelles, à contours « libres » et à contours « co-localisés », il s'agit dans cette section de présenter les outils d'optimisation nécessaires à leur minimisation, i.e., à la résolution algorithmique des Problèmes (4.5) et (4.7).

Remarquons tout d'abord que le terme de moindres carrés, formulé de manière compacte à l'Équation (4.4), est convexe car il s'écrit comme la composition d'une fonctionnelle affine et du carré de la norme ℓ_2 . De même, les pénalisations à contours « libres » et à contours « co-localisés » sont convexes, par convexité de la Variation Totale. Par conséquent, les fonctions objectifs apparaissant aux Équations (4.5) et (4.7) étant convexes et positives donc bornées inférieurement, elles possèdent chacune un minimum.

Néanmoins, les deux problèmes de minimisation (4.6) et (4.8) faisant chacun intervenir une pénalisation non-différentiable, à cause de l'utilisation de la norme mixte $\ell_{2,1}$ dans leur pénalisation, leur résolution ne peut être effectuée à l'aide des algorithmes de descente de gradient usuels. Il est donc nécessaire de se tourner vers les schémas d'optimisation convexe non-lisse, dans lesquels le gradient est remplacé par la sous-différentielle, et où les étapes de descente sont réalisées à l'aide d'opérateurs proximaux [12, 13, 3, 25].

4.3.1 Sous-différentiabilité

Pour les fonctions non différentiables mais convexes, le concept de *sous-différentielle*, introduit à la Définition 4.2, permet de généraliser la notion de gradient en relaxant la condition de tangence au point considéré.

Définition 4.2 (Sous-différentielle et opérateur proximal). [24, Sections 2.c et 3.a] Soit $\varphi : Y \subset \mathcal{H} \rightarrow \mathbb{R} \cup \{+\infty\}$ une fonction dont le domaine est inclus dans un espace de Hilbert \mathcal{H} muni de la norme $\|\cdot\|$, issue du produit scalaire $\langle \cdot, \cdot \rangle$. On suppose que φ est convexe, propre (i.e., prend au moins une valeur finie) et semi continue inférieurement (voir [3] pour la définition de la semi-continuité). On définit alors la *sous-différentielle* de φ , à valeurs dans l'ensemble des parties de \mathcal{H} ,

$$\partial\varphi \triangleq \begin{cases} Y & \longrightarrow \mathcal{P}(\mathcal{H}) \\ \mathbf{x} & \longmapsto \{\mathbf{u} \in \mathcal{H}, \text{ tel que, } \forall \mathbf{y} \in \mathcal{H}, \varphi(\mathbf{y}) \geq \varphi(\mathbf{x}) + \langle \mathbf{u}, \mathbf{y} - \mathbf{x} \rangle\}. \end{cases} \quad (4.10)$$

qui généralise la notion de gradient, puis l'*opérateur proximal* de φ

$$\text{prox}_\varphi \triangleq \begin{cases} Y & \longrightarrow Y \\ \mathbf{x} & \longmapsto \arg \min_{\mathbf{y} \in Y} \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|^2 + \varphi(\mathbf{y}). \end{cases} \quad (4.11)$$

Exemple 4.2 (Opérateur proximal du carré de la norme ℓ_2). Soit $\hat{\mathbf{x}} \in \mathcal{H}$ un vecteur et

$$\varphi : \begin{cases} \mathcal{H} & \longrightarrow \mathcal{H} \\ \mathbf{x} & \longmapsto \frac{1}{2} \|\mathbf{x} - \hat{\mathbf{x}}\|^2 \end{cases} \quad (4.12)$$

alors l'opérateur proximal de φ s'écrit de manière explicite :

$$\text{prox}_{\delta\varphi}(\mathbf{x}) = \frac{1}{1 + \delta} (\mathbf{x} + \delta\hat{\mathbf{x}}), \quad (4.13)$$

pour toute constante multiplicative $\delta > 0$.

Lorsqu'une fonction est différentiable, alors sa sous-différentielle au point \mathbf{x} se réduit à sa dérivée en ce point, i.e., $\partial\varphi(\mathbf{x}) = \{\nabla\varphi(\mathbf{x})\}$. Néanmoins, en général, plusieurs sous-gradients \mathbf{u} sont contenus dans la sous-différentielle, ce qui rend ambiguë la définition d'une étape de descente de sous-gradient explicite. Au contraire, l'opérateur proximal est monovalué et la proposition suivante montre qu'il est alors possible de définir de manière univoque une étape de descente de sous-gradient implicite.

Proposition 4.1. [3, Proposition 16.34] Soit $F : Y \subset \mathcal{H} \rightarrow \mathbb{R} \cup \{+\infty\}$ une fonction convexe, semi-continue inférieurement. Soit $\mathbf{p} \in \mathcal{H}$, alors

$$\mathbf{p} = \text{prox}_\varphi(\mathbf{x}) \iff \mathbf{x} - \mathbf{p} \in \partial\varphi(\mathbf{p}). \quad (4.14)$$

Une formulation équivalente de la relation (4.14),

$$\text{prox}_\varphi = (\mathbf{I} + \partial\varphi)^{-1} \quad (4.15)$$

où \mathbf{I} désigne l'identité de \mathcal{H} , permet d'interpréter l'action de l'opérateur proximal comme une étape de descente de gradient implicite sur la fonction φ .

4.3.2 Dualité de Fenchel-Rockafellar

L'opérateur proximal d'une fonction, étant défini à l'Équation (4.11) *via* un problème de minimisation, il peut s'avérer difficile à calculer. C'est le cas par exemple pour une fonction $G(\mathbf{x}) = \|\mathbf{U}\mathbf{x}\|$ écrite comme la composition d'un opérateur linéaire \mathbf{U} et d'une norme $\|\cdot\|$, dont l'opérateur proximal est très souvent difficile à calculer de manière exacte [9, 29]. Ce problème peut être contourné en s'intéressant à la résolution du problème d'optimisation *dual* [3], qui s'exprime grâce aux conjuguées convexes (voir Définition 4.3) des différents termes de la fonction objectif.

Définition 4.3 (Fonction conjuguée convexe). [3, Définition 13.1] Soit $\varphi : Y \subset \mathcal{H} \rightarrow \mathbb{R} \cup \{+\infty\}$, sa conjuguée convexe, notée φ^* est définie par

$$\varphi^* \triangleq \begin{cases} \mathcal{H} & \rightarrow \mathcal{H} \\ \mathbf{x} & \mapsto \sup_{\mathbf{y} \in Y} \langle \mathbf{y}, \mathbf{x} \rangle - \varphi(\mathbf{y}). \end{cases} \quad (4.16)$$

Exemple 4.3 (Convexe conjuguée du carré de la norme ℓ_2). Soit $\hat{\mathbf{x}} \in \mathcal{H}$ un vecteur et

$$\varphi : \begin{cases} \mathcal{H} & \rightarrow \mathcal{H} \\ \mathbf{x} & \mapsto \frac{1}{2} \|\mathbf{x} - \hat{\mathbf{x}}\|^2 \end{cases} \quad (4.17)$$

alors la fonction convexe conjuguée de φ s'écrit de manière explicite :

$$\varphi^*(\mathbf{x}) = \frac{1}{2} \|\mathbf{x} + \hat{\mathbf{x}}\|^2 - \frac{1}{2} \|\hat{\mathbf{x}}\|^2. \quad (4.18)$$

Pour une fonction φ quelconque, la conjuguée convexe φ^* est convexe, propre et semi-continue inférieurement. Ainsi, l'opérateur proximal de φ^* est bien défini (voir Définition 4.2). De plus, lorsque la fonction φ est convexe, propre et semi-continue inférieurement les opérateurs proximaux de φ et de φ^* sont reliés par l'identité de Moreau.

Proposition 4.2 (Identité de Moreau). [3, Théorème 14.3(ii)] Soit $\varphi : Y \subset \mathcal{H} \rightarrow \mathbb{R} \cup \{+\infty\}$ convexe, propre et semi-continue inférieurement et φ^* sa conjuguée convexe, alors pour tout réel $\tau > 0$, pour tout élément $\mathbf{x} \in \mathcal{H}$,

$$\mathbf{x} = \tau \text{prox}_{\varphi/\tau}(\mathbf{x}/\tau) + \text{prox}_{\tau\varphi^*}(\mathbf{x}). \quad (4.19)$$

Cette identité est très utile pour l'implémentation numérique des algorithmes proximaux, comme nous le verrons aux Sections 4.4.4 et 4.4.5.

Définition 4.4 (Problème dual). [15, Lemme 3.1] En considérant un terme d'attache aux données φ différentiable et une pénalisation de la forme $G(\mathbf{x}) = \|\mathbf{U}\mathbf{x}\|$, où $\mathbf{U} : \mathcal{H} \rightarrow \mathcal{G}$ est un opérateur linéaire borné et \mathcal{H} et \mathcal{G} sont deux espaces de Hilbert, dans le problème général de minimisation (4.24), le problème *primal*

$$\hat{\mathbf{x}} = \underset{\mathbf{x} \in \mathcal{H}}{\text{argmin}} \varphi(\mathbf{x}) + \|\mathbf{U}\mathbf{x}\| \quad (4.20)$$

est associé au problème *dual*

$$\hat{\mathbf{y}} = \underset{\mathbf{y} \in \mathcal{G}}{\text{argmax}} -\varphi^*(-\mathbf{U}^*\mathbf{y}) - \|\mathbf{y}\|^*, \quad (4.21)$$

où φ^* (resp. $\|\cdot\|^*$) désigne la conjuguée convexe au sens de la Définition 4.3 de φ (resp. $\|\cdot\|$) et \mathbf{U}^* est l'opérateur adjoint de \mathbf{U} .

Les problèmes (4.20) et (4.21) sont équivalents au sens de la dualité de Fenchel-Rockafellar et le théorème suivant permet de lier les solutions du problème primal et du problème dual.

Théorème 4.1 (Dualité de Fenchel-Rockafellar). [3, Théorème 19.1] S'il existe $\hat{\mathbf{x}} \in \mathcal{H}$ solution du problème primal (4.20), alors il existe également une solution du problème dual (4.21), notée $\hat{\mathbf{y}} \in \mathcal{G}$. De plus, $\hat{\mathbf{x}}$ et $\hat{\mathbf{y}}$ vérifient alors les relations suivantes

$$-\mathbf{U}^* \hat{\mathbf{y}} = \nabla \varphi(\hat{\mathbf{x}}) \quad \text{et} \quad \mathbf{U} \hat{\mathbf{x}} \in \partial \|\cdot\|^*(\hat{\mathbf{y}}) \quad (4.22)$$

où $\partial \|\cdot\|^*$ désigne la sous-différentielle de la conjuguée convexe de la norme $\|\cdot\|^*$ (voir les Définitions 4.2 et 4.3 et la Proposition 4.3). Un couple $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ satisfaisant les relations (4.22) est appelé point de Kuhn-Tucker.

Nous utiliserons à plusieurs reprises dans la suite la norme ℓ_p , $p \geq 1$, dont la conjuguée convexe admet la forme explicite de la Proposition 4.3.

Proposition 4.3 (Conjuguée de Fenchel d'une norme). [5, Exemple 3.26] Soit $\|\cdot\|_p$ la norme ℓ_p , avec $p \geq 1$, sur un espace de Hilbert \mathcal{G} et $\gamma > 0$ un réel. La conjuguée convexe de $\gamma \|\cdot\|_p$ est l'indicatrice de la boule centrée en $\mathbf{0}_{\mathcal{G}}$, de rayon γ pour la norme $\|\cdot\|_q$ avec $q \geq 1$ vérifiant $1/p + 1/q = 1$ [3], i.e.,

$$(\gamma \|\cdot\|_p)^* = \iota_{\{\|\cdot\|_q \leq \gamma\}}. \quad (4.23)$$

Remarque. De manière plus générale [5, Exemple 3.26], stipule que la conjuguée convexe d'une norme est l'indicatrice de la boule unité pour sa norme duale, ce qui justifiera l'extension naturelle du résultat de la Proposition 4.3 au cas d'une norme mixte.

4.3.3 Algorithmes proximaux

À partir des outils présentés aux Sections 4.3.1 et 4.3.2, plusieurs schémas de minimisation ont été élaborés pour résoudre les problèmes de minimisation du type

$$\underset{\mathbf{x} \in \mathcal{H}}{\text{minimize}} \quad \varphi(\mathbf{x}) + G(\mathbf{x}), \quad (4.24)$$

où φ et G sont deux fonctions propres, convexes et semi-continues inférieurement définies sur un espace de Hilbert \mathcal{H} . Un inconvénient majeur de ces algorithmes, dans le cas général de l'Équation (4.24), est leur convergence lente. En effet, ils présentent un taux de convergence typique qui se comporte comme $\mathcal{O}(1/T)$ où T est le nombre d'itérations [4]. Néanmoins des stratégies d'accélération ont été élaborées pour tirer le meilleur parti possible des propriétés de la fonction objectif. Nous allons nous concentrer dans la suite sur deux types de schémas proximaux : les algorithmes de type *forward-backward* et les algorithmes proximaux primaux-duaux.

Les algorithmes de type *forward-backward*, introduits par [14], reposent sur la différentiabilité de la fonction φ de l'Équation (4.24). Le réglage du pas de descente dépend alors de la constante de Lipschitz du gradient de φ . Une sur-relaxation des itérations permet alors d'obtenir un schéma accéléré, appelé *Fast Iterative Soft Thresholding Algorithm* (FISTA) [4, 2], qui a été initialement construit pour le débruitage par Variation Totale de l'Équation (2.123), mais qui s'étend à des fonctionnelles plus générales. Le taux de convergence est alors amélioré et on obtient une convergence en $\mathcal{O}(1/T^2)$.

Quant aux algorithmes proximaux primaux-duaux [11], ils permettent de minimiser des fonctionnelles pour lesquelles on sait calculer les opérateurs proximaux de l'attache aux données et de la pénalisation, sans faire appel à un gradient, ni requérir une constante de Lipschitz. Chambolle et Pock ont construit une stratégie d'accélération de convergence pour l'algorithme primal-dual basée sur la *forte-convexité*, présentée à la Section 4.3.4, de la fonction objectif [8, Algorithme 2], qui permet d'atteindre une convergence asymptotique en $\mathcal{O}(1/T^2)$.

4.3.4 Forte-convexité

Définition 4.5 (Forte-convexité). [3, Définition 10.5] Une fonction $\varphi : \mathcal{H} \rightarrow \mathbb{R}$, définie sur un espace de Hilbert \mathcal{H} est fortement convexe de module $\mu > 0$ si et seulement si la fonction :

$$\mathbf{y} \mapsto \varphi(\mathbf{y}) - \frac{\mu}{2} \|\mathbf{y}\|^2 \quad (4.25)$$

est convexe, où $\|\cdot\|$ la norme est la norme associée au produit scalaire $\langle \cdot, \cdot \rangle$ de l'espace Hilbert \mathcal{H} . Le *module de forte-convexité*, de la fonction convexe φ , noté μ_φ , est la borne supérieure des réels positifs μ satisfaisant la condition (4.25).

Pour montrer la forte convexité d'une fonctionnelle telle que celle de l'Équation (4.24), la plupart du temps, on montre que l'un des termes, par exemple une pénalisation G est convexe, tandis que l'autre terme, par exemple une attache aux données φ , est fortement convexe. En effet, la forte-convexité de l'attache aux données entraîne alors celle de la fonction objectif, comme somme d'une fonction fortement convexe et d'une fonction convexe. La proposition suivante fournit un outil efficace non seulement pour montrer la forte convexité d'une fonction φ , mais également pour calculer son module de forte convexité μ , qui est nécessaire pour l'implémentation d'un algorithme primal-dual accéléré résolvant le Problème (4.24).

Proposition 4.4 (Hessienne et convexité). [5, Sections 3.1.3 et 3.1.4] Soit $\varphi : \mathcal{H} \rightarrow \mathbb{R}$, définie sur un espace de Hilbert \mathcal{H} , deux fois dérivable et $\nabla^2\varphi(\mathbf{y})$ sa dérivée seconde en \mathbf{y} .

- (i) Convexité : La fonction φ est convexe si et seulement si sa matrice hessienne $\nabla^2\varphi(\mathbf{y})$ est positive en tout point \mathbf{y} .
- (ii) Stricte-convexité : La fonction φ est strictement convexe si et seulement si sa matrice hessienne $\nabla^2\varphi(\mathbf{y})$ est définie positive en tout point \mathbf{y} .
- (iii) Forte-convexité : La fonction φ est μ fortement convexe si et seulement si

$$\forall \mathbf{y} \in \mathcal{H}, \quad \nabla^2\varphi(\mathbf{y}) \succeq \mu \mathbf{I} \quad (4.26)$$

où \mathbf{I} est l'opérateur identité sur \mathcal{H} . Dans le cas où \mathcal{H} est de dimension finie, $\nabla^2\varphi(\mathbf{y})$ est la matrice hessienne de φ au point \mathbf{y} et la condition (4.26) est équivalente au fait que toutes les valeurs propres de la hessienne soient supérieures ou égales à $\mu > 0$. De plus, le module de forte convexité de φ correspond à la borne inférieure de l'ensemble des valeurs propres des matrices hessiennes $(\nabla^2\varphi(\mathbf{y}))_{\mathbf{y} \in \mathcal{H}}$:

$$\mu_\varphi = \inf \{ \mu \geq 0 \mid \exists \mathbf{y} \in \mathcal{H}, \mu \in \text{Sp}(\nabla^2\varphi(\mathbf{y})) \}, \quad (4.27)$$

où $\text{Sp}(\nabla^2\varphi(\mathbf{y}))$ désigne le spectre de la matrice $\nabla^2\varphi(\mathbf{y})$.

Exemple 4.4 (Convexité, stricte-convexité et forte-convexité). Pour une variable $\mathbf{y} = \{y(\underline{n})\}_{\underline{n} \in \Omega} \in \mathbb{R}^{\mathbb{N}_1 \times \mathbb{N}_2}$ et une donnée $\mathbf{z} = \{z(\underline{n})\}_{\underline{n} \in \Omega} \in \mathbb{R}^{\mathbb{N}_1 \times \mathbb{N}_2}$, les fonctions suivantes sont d'usage courant pour mesurer la fidélité aux données dans les problèmes de traitement d'image par minimisation de fonctionnelle.

- (i) La norme ℓ_1 est définie comme

$$\|\mathbf{y} - \mathbf{z}\|_1 \triangleq \sum_{\underline{n} \in \Omega} |y(\underline{n}) - z(\underline{n})|. \quad (4.28)$$

En tant que norme, la fonction $\mathbf{x} \mapsto \|\mathbf{y} - \mathbf{z}\|_1$ est convexe. Néanmoins, comme illustré dans le cas unidimensionnel à la Figure 4.2a, la norme ℓ_1 n'est pas strictement convexe, puisqu'elle possède une courbure nulle en tout point où $\mathbf{y} \mapsto \|\mathbf{y} - \mathbf{z}\|_1$ est deux fois différentiable.

- (ii) La divergence de Kullback-Leibler, définie pour des quantités \mathbf{y} et \mathbf{z} strictement positives, s'écrit [20]

$$\text{KL}(\mathbf{y}, \mathbf{z}) \triangleq \sum_{\underline{n} \in \Omega} z(\underline{n}) \ln \left(\frac{z(\underline{n})}{y(\underline{n})} \right) + y(\underline{n}) - z(\underline{n}). \quad (4.29)$$

La fonction $\mathbf{y} \mapsto \text{KL}(\mathbf{y}, \mathbf{z})$ est deux fois dérivable et

$$\frac{\partial^2 \text{KL}}{\partial y(\underline{n}) \partial y(\underline{n}')} (\mathbf{y}, \mathbf{z}) = \delta(\underline{n}, \underline{n}') \frac{1}{y(\underline{n})} > 0. \quad (4.30)$$

Par conséquent, le point (ii) de la Proposition 4.4 nous assure que la divergence de Kullback-Leibler est strictement convexe. On constate, en prenant la limite $y(\underline{n}) \rightarrow +\infty$, que la courbure de $\mathbf{y} \mapsto \text{KL}(\mathbf{y}, \mathbf{z})$ tend vers 0 à l'infini (voir Figure 4.2b, où la partie droite de la courbe présente une asymptote), par conséquent, la divergence de Kullback-Leibler n'est pas fortement convexe.

- (iii) L'erreur quadratique est définie à partir du carré de la norme ℓ_2 :

$$\text{EQ}(\mathbf{y}, \mathbf{z}) \triangleq \frac{1}{2} \|\mathbf{y} - \mathbf{z}\|_2^2 = \frac{1}{2} \sum_{\underline{n} \in \Omega} (y(\underline{n}) - z(\underline{n}))^2. \quad (4.31)$$

La fonction $\mathbf{y} \mapsto \text{EQ}(\mathbf{y}, \mathbf{z})$ est deux fois différentiable par rapport à la variable \mathbf{y} et sa matrice hessienne, $\nabla^2 \text{EQ}$, est égale à l'identité. Par conséquent, le point (iii) de la Proposition 4.4 montre que $\mathbf{y} \mapsto \text{EQ}(\mathbf{y}, \mathbf{z})$ est 1-fortement convexe (voir Figure 4.2c).

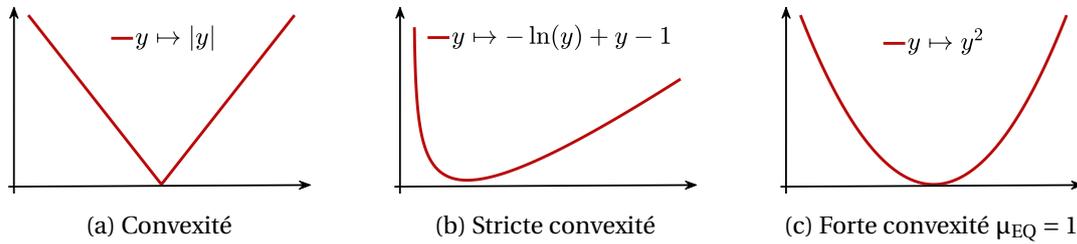


FIGURE 4.2 – Illustration des différentes propriétés de convexité.

4.4 Minimisation de fonctionnelles pour la segmentation de texture

Les deux schémas algorithmiques, *forward-backward* et primal-dual, sont implémentés et comparés pour la résolution des problèmes de minimisation (2.123), (4.5) et (4.7) pour la segmentation de textures. Dans un premier temps, nous nous attachons au calcul de la conjuguée convexe de F , nécessaire pour dériver les formulations duales des Problèmes (4.5) et (4.7). En effet, à cause de la présence d'une pénalisation de type $\|\mathbf{U}\mathbf{x}\|$ dans les Problèmes (2.123), (4.5) et (4.7), l'algorithme *forward-backward* sera implémenté pour la résolution de leurs problèmes *duaux* (voir Section 4.3.2). Puis, dans un second temps, nous démontrons la forte-convexité de l'attache aux données $F(\mathbf{h}, \mathbf{v}; \log_2 \mathcal{L})$, nécessaire pour la mise en œuvre de la stratégie d'accélération de l'algorithme primal-dual de [8, Algorithme 2] et calculons son module de forte-convexité μ_F . Nous calculons ensuite la forme explicite de l'opérateur proximal de F , également nécessaire pour l'implémentation de l'algorithme primal-dual de Chambolle-Pock [8, Algorithme 2]. Enfin, nous détaillons la mise en œuvre de ces algorithmes pour la segmentation de textures *via* la méthode $\text{ROF}_{\mathbf{h}}$, i.e. par la résolution du Problème (2.123), *via* la méthode de segmentation à contours « libres », i.e., par résolution du Problème (4.5), et *via* la méthode de segmentation à contours « co-localisés », i.e., par résolution du Problème (4.7).

4.4.1 Propriétés de l'attache aux données

Afin d'écrire les problèmes duaux associées aux problèmes de minimisation (4.5) (pénalisation à contours « libres ») et (4.7) (pénalisation à contours « co-localisés ») il est tout d'abord nécessaire de calculer F^* , la fonction convexe conjuguée du terme d'attache aux données.

Commençons par calculer l'adjoint de l'opérateur linéaire (borné) Φ défini à l'Équation (4.3).

Proposition 4.5 (Adjoint de Φ). *L'opérateur linéaire Φ , défini à l'Équation (4.3), est borné et son adjoint, agissant sur une variable multi-échelle $\ell = \{\ell_j\}_{j=j_1}^{j_2}$, s'écrit*

$$\Phi^* : \begin{cases} \mathbb{R}^{N_j \times N_1 \times N_2} \longrightarrow \mathbb{R}^{N_1 \times N_2} \times \mathbb{R}^{N_1 \times N_2} \\ \{\ell_j\}_{j=j_1}^{j_2} \longmapsto \left(\sum_{j=j_1}^{j_2} j \ell_j, \sum_{j=j_1}^{j_2} \ell_j \right) \end{cases} \quad (4.32)$$

Remarque. Les quantités $\mathcal{S}(n)$ et $\mathcal{T}(n)$ introduites à l'Équation (2.91) peuvent se réécrire comme l'action de Φ^* sur les logarithmes des coefficients dominants, c'est-à-dire que les cartes \mathcal{T} et \mathcal{S} sont obtenues comme :

$$(\mathcal{T}, \mathcal{S}) = \Phi^* (\log_2 \mathcal{L}). \quad (4.33)$$

L'inversion de l'opérateur $\Phi^* \Phi$ réalisée à la proposition suivante sera utile dans le calcul de la fonction convexe conjuguée de F .

Proposition 4.6 (Inversion de $\Phi^* \Phi$). *L'opérateur linéaire $\Phi^* \Phi$ est inversible et son inverse s'écrit :*

$$(\Phi^* \Phi)^{-1} : \begin{cases} \mathbb{R}^{N_1 \times N_2} \times \mathbb{R}^{N_1 \times N_2} \longrightarrow \mathbb{R}^{N_1 \times N_2} \times \mathbb{R}^{N_1 \times N_2} \\ (\mathbf{h}, \mathbf{v}) \longmapsto \left(\frac{1}{S_2 S_0 - S_1^2} (S_0 \mathbf{h} - S_1 \mathbf{v}), \frac{1}{S_2 S_0 - S_1^2} (-S_1 \mathbf{h} + S_2 \mathbf{v}) \right) \end{cases} \quad (4.34)$$

agissant sur un couple de cartes (\mathbf{h}, \mathbf{v}) chacune de taille $N_1 \times N_2$.

Démonstration. La combinaison des Relations (4.3) et (4.32) amène à l'expression de l'opérateur composé

$$\Phi^* \Phi : \begin{cases} \mathbb{R}^{N_1 \times N_2} \times \mathbb{R}^{N_1 \times N_2} \longrightarrow \mathbb{R}^{N_1 \times N_2} \times \mathbb{R}^{N_1 \times N_2} \\ (\mathbf{h}, \mathbf{v}) \longmapsto (S_2 \mathbf{h} + S_1 \mathbf{v}, S_1 \mathbf{h} + S_0 \mathbf{v}) \end{cases} \quad (4.35)$$

qui est inversé en utilisant la formule des cofacteurs en dimension 2. \square

Proposition 4.7 (Conjuguée de F). *Soit $(\mathcal{S}, \mathcal{T})$ les cartes définies à l'Équation (2.91) et Φ l'opérateur linéaire défini à l'Équation (4.3),*

$$F^*(\mathbf{h}, \mathbf{v}; \log_2 \mathcal{L}) = \frac{1}{2} \langle (\mathbf{h}, \mathbf{v}), (\Phi^* \Phi)^{-1} (\mathbf{h}, \mathbf{v}) \rangle + \langle (\mathcal{T}, \mathcal{S}), (\Phi^* \Phi)^{-1} (\mathbf{h}, \mathbf{v}) \rangle + \mathcal{C}, \quad (4.36)$$

avec $\mathcal{C} = \frac{1}{2} \langle (\mathcal{T}, \mathcal{S}), (\Phi^* \Phi)^{-1} (\mathcal{T}, \mathcal{S}) \rangle - \frac{1}{2} \sum_j \log_2 \mathcal{L}_j^2$ une carte constante,

où $\langle \cdot, \cdot \rangle$ désigne le produit scalaire canonique sur l'espace produit $\mathbb{R}^{N_1 \times N_2} \times \mathbb{R}^{N_1 \times N_2}$.

Démonstration. En utilisant la Définition 4.3 de la conjuguée convexe au sens de Fenchel, on obtient

$$F^*(\mathbf{h}, \mathbf{v}; \log_2 \mathcal{L}) = \sup_{\tilde{\mathbf{h}} \in \mathbb{R}^{N_1 \times N_2}, \tilde{\mathbf{v}} \in \mathbb{R}^{N_1 \times N_2}} \langle (\tilde{\mathbf{h}}, \tilde{\mathbf{v}}), (\mathbf{h}, \mathbf{v}) \rangle - F(\tilde{\mathbf{h}}, \tilde{\mathbf{v}}; \log_2 \mathcal{L}). \quad (4.37)$$

La fonction F étant différentiable, de même que le produit scalaire, l'annulation du gradient entraîne que le supremum est atteint en $(\bar{\mathbf{h}}, \bar{\mathbf{v}})$ vérifiant

$$(\mathbf{h}, \mathbf{v}) - \Phi^* \left(\Phi(\bar{\mathbf{h}}, \bar{\mathbf{v}}) - \log_2 \mathcal{L} \right) = (\mathbf{0}, \mathbf{0}), \quad (4.38)$$

où $\mathbf{0}$ désigne la carte nulle de taille $N_1 \times N_2$. Puis, en utilisant la Relation (4.33), on obtient :

$$\Phi^* \Phi(\bar{\mathbf{h}}, \bar{\mathbf{v}}) = (\mathbf{h}, \mathbf{v}) + (\mathcal{F}, \mathcal{S}) \quad (4.39)$$

ce qui conduit à

$$(\bar{\mathbf{h}}, \bar{\mathbf{v}}) = (\Phi^* \Phi)^{-1} (\mathbf{h} + \mathcal{F}, \mathbf{v} + \mathcal{S}). \quad (4.40)$$

Ce résultat peut alors être réinjecté dans l'expression explicite de $\langle \tilde{\mathbf{h}}, \mathbf{v} \rangle + \langle \tilde{\mathbf{h}}, \mathbf{h} \rangle - F(\tilde{\mathbf{h}}, \tilde{\mathbf{v}}; \log_2 \mathcal{L})$, ce qui fournit la Formule (4.36). \square

On constate que F^* est une fonctionnelle quadratique en (\mathbf{h}, \mathbf{v}) , par conséquent son gradient, utilisé pour l'implémentation de l'algorithme FISTA, est aisément calculable (voir Proposition 4.8).

Proposition 4.8 (Gradient de F^*). *La fonction convexe conjuguée de l'attache aux données $F(\mathbf{h}, \mathbf{v}; \log_2 \mathcal{L})$, calculée à l'Équation (4.36), est dérivable et son gradient s'écrit*

$$\nabla F^*(\mathbf{h}, \mathbf{v}; \log_2 \mathcal{L}) = (\Phi^* \Phi)^{-1} (\mathbf{h} + \mathcal{F}, \mathbf{v} + \mathcal{S}), \quad (4.41)$$

où \mathcal{S} et \mathcal{F} sont définis à l'Équation (2.91) et Φ est introduit à l'Équation (4.3). En outre, l'opérateur $(\Phi^* \Phi)^{-1}$ étant borné (grâce à la dimension finie), la fonction F^* est de gradient lipschitzien, avec une constante de Lipschitz égale à $\|(\Phi^* \Phi)^{-1}\|$. En outre, la norme d'opérateur $\|(\Phi^* \Phi)^{-1}\|$ est égale à la norme de la matrice \mathbf{J}^{-1} fournie à l'Équation (2.91) que nous rappelons ici :

$$\mathbf{J}^{-1} = \frac{1}{S_2 S_0 - S_1^2} \begin{pmatrix} S_0 & -S_1 \\ -S_1 & S_2 \end{pmatrix}, \quad \text{où } \forall m \in \{0, 1, 2\}, \quad S_m = \sum_{j=j_1}^{j_2} j^m. \quad (4.42)$$

Ainsi, l'évaluation numérique de $\|(\Phi^* \Phi)^{-1}\| = \|\mathbf{J}^{-1}\|$ est possible avec un faible coût de calcul.

Démonstration. La fonction $F^*(\mathbf{h}, \mathbf{v}; \log_2 \mathcal{L})$ est quadratique en (\mathbf{h}, \mathbf{v}) , ainsi sa différentielle est une fonction affine de (\mathbf{h}, \mathbf{v}) et un calcul direct donne

$$\nabla F^*(\mathbf{h}, \mathbf{v}; \log_2 \mathcal{L}) = (\Phi^* \Phi)^{-1} (\mathbf{h}, \mathbf{v}) + (\Phi^* \Phi)^{-1} (\mathcal{F}, \mathcal{S}) + (\mathbf{0}, \mathbf{0}) \quad (4.43)$$

$$= (\Phi^* \Phi)^{-1} (\mathbf{h} + \mathcal{F}, \mathbf{v} + \mathcal{S}). \quad (4.44)$$

L'Équation (4.34) montre qu'en chaque pixel \underline{n} l'action de $(\Phi^* \Phi)^{-1}$ est identique et correspond à l'action de la matrice \mathbf{J}^{-1} sur le vecteur $(h(\underline{n}), v(\underline{n}))^\top$. Par conséquent la norme de $(\Phi^* \Phi)^{-1}$ est égale à la norme de la matrice \mathbf{J}^{-1} . \square

Afin, d'une part, de démontrer l'existence et l'unicité d'une solution aux Problèmes (4.5) et (4.7), et d'autre part, pour implémenter l'algorithme primal-dual accéléré de Chambolle-Pock [8, Algorithme 2] pour leur résolution, il est nécessaire de montrer la forte-convexité du terme de moindres carrés $F(\mathbf{v}, \mathbf{h}; \log_2 \mathcal{L})$ et de calculer son module de forte-convexité (voir Proposition 4.9). Nous aurons également besoin de calculer l'opérateur proximal de la fonction F (voir Proposition 4.11).

Proposition 4.9 (Forte-convexité de F). *La fonction $F(\mathbf{v}, \mathbf{h}; \log_2 \mathcal{L})$ est μ_F -fortement convexe par rapport aux variables (\mathbf{v}, \mathbf{h}) , avec $\mu_F > 0$ la plus petite valeur propre de l'opérateur défini positif $\Phi^* \Phi$, défini à l'Équation (4.35). En outre, les valeurs propres de l'opérateur $\Phi^* \Phi$ sont exactement, aux multiplicités près, les valeurs propres de la matrice \mathbf{J} définie à l'Équation (2.91) (dont l'expression est rappelée à l'Équation (4.42)) et*

$$\mu_F = \min \text{Sp}(\mathbf{J}). \quad (4.45)$$

Démonstration. Remarquons tout d'abord que $F(\mathbf{h}, \mathbf{v}; \log_2 \mathcal{L})$ est quadratique en la variable $(\mathbf{h}, \mathbf{v}) \in (\mathbb{R}^{N_1 \times N_2})^2$ et peut se réécrire de manière compacte comme le carré de la norme d'une fonction affine :

$$F(\mathbf{h}, \mathbf{v}; \log_2 \mathcal{L}) = \frac{1}{2} \|\Phi(\mathbf{h}, \mathbf{v}) - \log_2 \mathcal{L}\|^2 \quad (4.46)$$

avec $\log_2 \mathcal{L}$ la concaténation des logarithmes des coefficients *dominants* à toutes les échelles 2^{j_1} à 2^{j_2} , $N_j = j_2 - j_1 + 1$ désigne le nombre d'échelles considérées et Φ est défini à l'Équation (4.3). Sous cette forme, la différentiation de la fonction F est directe et on obtient une Hessienne *constante* :

$$\forall (\mathbf{h}, \mathbf{v}) \in (\mathbb{R}^{N_1 \times N_2})^2, \quad \nabla^2 F(\mathbf{h}, \mathbf{v}; \log_2 \mathcal{L}) = \Phi^* \Phi, \quad (4.47)$$

où Φ^* désigne l'opérateur adjoint de l'opérateur Φ , bien défini car \mathcal{H} est de dimension finie. En utilisant la Proposition 4.4 on en déduit que F est fortement convexe, de module de forte convexité

$$\mu_F = \min \text{Sp}(\Phi^* \Phi). \quad (4.48)$$

De plus, l'Équation (4.35) montre qu'en chaque pixel \underline{n} l'action de $\Phi^* \Phi$ est identique et correspond à l'action de la matrice \mathbf{J} sur le vecteur $(h(\underline{n}), v(\underline{n}))^\top$. Par conséquent, $\Phi^* \Phi$ et \mathbf{J} partagent le même spectre, aux multiplicités près. On en conclut donc que

$$\mu_F = \min \text{Sp}(\mathbf{J}). \quad (4.49)$$

□

Le calcul pratique de μ_F est ainsi réalisé en remarquant que les valeurs propres de l'opérateur $\Phi^* \Phi$ sont, aux multiplicités près, celles de la matrice \mathbf{J} , de taille 2×2 , définie à l'Équation (2.91), aisément diagonalisable numérique.

Remarque. Le module de convexité μ_F ne dépend donc que de la gamme d'échelles considérées $\{2^{j_1}, \dots, 2^{j_2}\}$. La Figure 4.3 fournit les valeurs de μ_F en fonction de j_1 et j_2 . Pour comparaison, la fonction $y \mapsto y^2/2$ est 1-fortement convexe.

μ_F	$j_2 = 2$	$j_2 = 3$	$j_2 = 4$	$j_2 = 5$	$j_2 = 6$
$j_1 = 1$	0,29	0,72	1,20	1,69	2,20
$j_1 = 2$	0	0,13	0,38	0,70	1,06

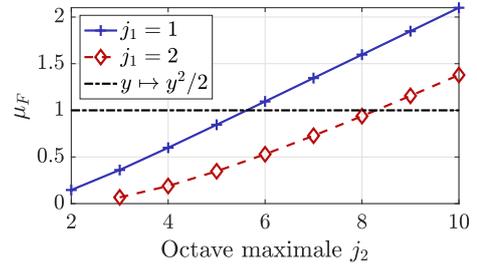


FIGURE 4.3 – Module de forte convexité μ_F du terme d'attache aux données $F(\mathbf{v}, \mathbf{h}; \log_2 \mathcal{L})$ (4.48) calculé comme la plus petite valeur propre (4.45) de la matrice \mathbf{J} définie en (2.91) pour j_1 fixé à $j_1 = 1$ (1^{ère} ligne du tableau, en bleu sur le graphique) ou $j_1 = 2$ (2^{ème} ligne du tableau, en rouge sur le graphique) et j_2 variable.

Grâce à la forte-convexité des fonctionnelles à pénalisations « libre » et « co-localisée » on peut montrer, pour chacune, l'existence et l'unicité d'un minimum et ainsi justifier l'utilisation d'une égalité dans les problèmes (4.5) et (4.7).

Proposition 4.10 (Existence et unicité d'un minimum). *La fonctionnelle à pénalisation « libre » (resp. « co-localisée ») admet un minimum global. De plus ce minimum est atteint en un unique couple $(\mathbf{h}^L, \mathbf{v}^L)$ (resp. $(\mathbf{h}^C, \mathbf{v}^C)$).*

Remarque. La forte-convexité du terme d'attache aux données du Problème (2.123) est directe puisqu'il s'agit d'un terme quadratique dont la matrice hessienne est l'identité. Son module de forte-convexité, utilisé à l'Algorithme 8, est donc $\mu = 1$. On en déduit également l'existence et l'unicité de la solution du Problème (2.123).

Proposition 4.11 (Opérateur proximal de F). *La fonction $F(\mathbf{v}, \mathbf{h}; \log_2 \mathcal{L})$ est propre, convexe et semi-continue inférieurement et son opérateur proximal possède l'expression explicite suivante :*

$$\begin{pmatrix} \tilde{\mathbf{h}} \\ \tilde{\mathbf{v}} \end{pmatrix} = \text{prox}_{\delta F} \begin{pmatrix} \mathbf{h} \\ \mathbf{v} \end{pmatrix} \iff \begin{cases} \tilde{\mathbf{h}} = \frac{1}{(1+\delta S_2)(1+\delta S_0) - \delta^2 S_1^2} ((1+\delta S_0)(\delta \mathcal{F} + \mathbf{h}) - \delta S_1(\delta \mathcal{S} + \mathbf{v})) \\ \tilde{\mathbf{v}} = \frac{1}{(1+\delta S_2)(1+\delta S_0) - \delta^2 S_1^2} ((1+\delta S_2)(\delta \mathcal{S} + \mathbf{v}) - \delta S_1(\delta \mathcal{F} + \mathbf{h})) \end{cases} \quad (4.50)$$

où $\delta > 0$ est une constante multiplicative, et les coefficients $\{S_m, m = 1, 2, 3\}$ sont ceux introduits à l'Équation (2.91), de même que les cartes \mathcal{S} et \mathcal{T} , exprimées à l'Équation (2.91) en fonction des coefficients d'ondelettes dominants.

Démonstration. La fonction $(\mathbf{h}, \mathbf{v}) \mapsto \frac{1}{2} \|\Phi(\mathbf{h}, \mathbf{v}) - \log_2 \mathcal{L}\|_2^2$ est quadratique. Par conséquent elle est propre, convexe et continue, donc *a fortiori* semi-continue inférieurement. En appliquant la Définition 4.2 de l'opérateur proximal à la fonction $F(\mathbf{v}, \mathbf{h}; \log_2 \mathcal{L})$ on obtient

$$\begin{pmatrix} \tilde{\mathbf{h}} \\ \tilde{\mathbf{v}} \end{pmatrix} = \text{prox}_{\delta F} \begin{pmatrix} \mathbf{h} \\ \mathbf{v} \end{pmatrix} \iff (\tilde{\mathbf{h}}, \tilde{\mathbf{v}}) = \arg \min_{\bar{\mathbf{h}}, \bar{\mathbf{v}}} \frac{1}{2} \|(\bar{\mathbf{h}}, \bar{\mathbf{v}}) - (\mathbf{h}, \mathbf{v})\|_2^2 + \frac{\delta}{2} \|\Phi(\bar{\mathbf{h}}, \bar{\mathbf{v}}) - \log_2 \mathcal{L}\|_2^2. \quad (4.51)$$

La fonction F étant différentiable, tout comme le terme quadratique, le minimum est atteint en $(\tilde{\mathbf{h}}, \tilde{\mathbf{v}})$ annulant le gradient, et vérifiant donc

$$(\bar{\mathbf{h}}, \bar{\mathbf{v}}) - (\mathbf{h}, \mathbf{v}) + \delta \Phi^* (\Phi(\bar{\mathbf{h}}, \bar{\mathbf{v}}) - \log_2 \mathcal{L}) = (\mathbf{0}, \mathbf{0}), \quad (4.52)$$

où $\mathbf{0}$ désigne la carte nulle de taille $N_1 \times N_2$. Puis, en utilisant la Relation (4.33), on obtient :

$$(\mathbf{I} + \delta \Phi^* \Phi) (\tilde{\mathbf{h}}, \tilde{\mathbf{v}}) = \delta (\mathcal{T}, \mathcal{S}) + (\mathbf{h}, \mathbf{v}), \quad (4.53)$$

où \mathbf{I} désigne l'opérateur identité s'appliquant à un couple de variables de taille $N_1 \times N_2$ chacune. En outre, on déduit de l'expression de $\Phi^* \Phi$ à l'Équation (4.35), celle de $(\mathbf{I} + \delta \Phi^* \Phi)$, qui peut alors être inversée :

$$(\mathbf{I} + \delta \Phi^* \Phi)^{-1} : \begin{cases} \mathbb{R}^{N_1 \times N_2} \times \mathbb{R}^{N_1 \times N_2} & \longrightarrow \mathbb{R}^{N_1 \times N_2} \times \mathbb{R}^{N_1 \times N_2} \\ (\mathbf{h}, \mathbf{v}) & \longmapsto \frac{1}{(1 + \delta S_2)(1 + \delta S_0) - \delta^2 S_1^2} ((1 + \delta S_0)\mathbf{h} - \delta S_1 \mathbf{v}, ((1 + \delta S_2)\mathbf{v} - \delta S_1 \mathbf{h})). \end{cases} \quad (4.54)$$

L'application de $(\mathbf{I} + \delta \Phi^* \Phi)^{-1}$ à l'Équation (4.53) fournit alors directement le résultat. \square

4.4.2 Pénalisations à contours « libres » et à contours « co-localisés »

Les pénalisations G^L (4.6) et G^C (4.8) consistent toutes les deux en la composition d'un opérateur linéaire de différences finies et de la norme mixte, $\|\cdot\|_{2,1}$, définie à l'Équation (2.44). Afin de pouvoir implémenter les algorithmes proximaux *forward-backward* et primal-dual, il est nécessaire d'explicitier cette décomposition. Pour cela, nous allons utiliser l'opérateur de différences finies \mathbf{D} , défini à l'Équation (2.41) à partir des opérateurs \mathbf{V} et \mathbf{H} de gradients vertical et horizontal discrets, et la norme $\|\cdot\|_{2,1}$ mixte de l'Équation (2.44) afin d'écrire :

$$G^L(\mathbf{h}, \mathbf{v}) = \left\| \mathbf{U}_{(\lambda, \alpha)}^L(\mathbf{h}, \mathbf{v}) \right\|_{2,1}, \quad \text{avec } \mathbf{U}_{(\lambda, \alpha)}^L(\mathbf{h}, \mathbf{v}) \triangleq \lambda (\alpha \mathbf{D}\mathbf{h} \quad \mathbf{D}\mathbf{v}) = \lambda \begin{pmatrix} \alpha \mathbf{V}\mathbf{h} & \mathbf{V}\mathbf{v} \\ \alpha \mathbf{H}\mathbf{h} & \mathbf{H}\mathbf{v} \end{pmatrix}, \quad (4.55)$$

où la norme mixte $\ell_{2,1}$ est effectuée sur $I = 2$ composantes (pour la partie ℓ_2), et parcourt deux fois $N_1 \times N_2$ pixels (pour la partie ℓ_1), et

$$G^C(\mathbf{h}, \mathbf{v}) = \left\| \mathbf{U}_{(\lambda, \alpha)}^C(\mathbf{h}, \mathbf{v}) \right\|_{2,1}, \quad \text{avec } \mathbf{U}_{(\lambda, \alpha)}^C(\mathbf{h}, \mathbf{v}) \triangleq \lambda \begin{pmatrix} \alpha \mathbf{D}\mathbf{h} \\ \mathbf{D}\mathbf{v} \end{pmatrix} = \lambda \begin{pmatrix} \alpha \mathbf{V}\mathbf{h} \\ \alpha \mathbf{H}\mathbf{h} \\ \mathbf{V}\mathbf{v} \\ \mathbf{H}\mathbf{v} \end{pmatrix} \quad (4.56)$$

où la norme mixte $\ell_{2,1}$ couple $I = 4$ composantes (pour la partie ℓ_2) et parcourt $N_1 \times N_2$ pixels (pour la partie ℓ_1).

En outre, nous aurons également besoin de l'expression de l'opérateur proximal de la norme mixte $\ell_{2,1}$, fournie à la définition suivante.

Définition 4.6 (Seuillage mixte $\ell_{2,1}$). Pour une variable multi-composante $\mathbf{z} = [z_1; \dots; z_I]^\top \in \mathbb{R}^{I \times N_1 \times N_2}$, en posant

$$\mathbf{z}(\underline{n}) \triangleq [z_1(\underline{n}), \dots, z_I(\underline{n})] \quad \text{et} \quad \|\mathbf{z}(\underline{n})\|_2 = \sqrt{\sum_{i=1}^I (z_i(\underline{n}))^2} \quad (4.57)$$

la norme ℓ_2 locale au pixel $\underline{n} = (n_1, n_2)$, alors l'opérateur proximal de la norme $\|\cdot\|_{2,1}$, qui est définie à l'Équation (2.44), s'écrit

$$\tilde{\mathbf{z}} \triangleq \text{prox}_{\lambda \|\cdot\|_{2,1}}(\mathbf{z}) \iff \tilde{z}_i(\underline{n}) = \begin{cases} \left(1 - \frac{\lambda}{\|\mathbf{z}(\underline{n})\|_2}\right) z_i(\underline{n}) & \text{si } \|\mathbf{z}(\underline{n})\|_2 \geq \lambda \\ 0 & \text{si } \|\mathbf{z}(\underline{n})\|_2 < \lambda. \end{cases} \quad (4.58)$$

4.4.3 Formulation des problèmes duaux

Grâce aux Sections 4.4.1 et 4.4.2, on dispose de tous les outils pour écrire les problèmes duaux, au sens de la Définition 4.4, associés aux Problèmes (2.123), (4.5) et (4.7) qui seront utilisés à la section suivante pour l'implémentation de schémas de minimisation sous la forme d'algorithmes *forward-backward*.

Proposition 4.12 (Fonctionnelle de Rudin-Osher-Fatemi duale). *Le Problème (2.123) possède la formulation duale suivante :*

$$\hat{\mathbf{g}}^{\text{ROF}} = \underset{\mathbf{g} \in \mathbb{R}^{2 \times N_1 \times N_2}}{\text{argmin}} \frac{1}{2} \|\mathbf{D}^* \mathbf{g} + \hat{\mathbf{h}}^{\text{RL}}\|^2 + \iota_{\|\cdot\|_{2,\infty} \leq \lambda}(\mathbf{g}). \quad (4.59)$$

De plus, le terme d'attache aux données dual est différentiable, de gradient lipschitzien avec une constante de Lipschitz $\|\mathbf{D}\|^2$.

Démonstration. La formulation du problème dual est obtenue en appliquant la Définition 4.4. L'Exemple 4.3 fournit l'expression de la fonction convexe conjuguée du carré de la norme ℓ_2 et permet d'écrire le terme d'attache aux données dual. Puis, la conjuguée de la norme mixte $\ell_{2,1}$ est donnée par la Proposition 4.3. On remarque ensuite que l'attache aux données du problème dual est quadratique, de gradient

$$\mathbf{g} \mapsto \mathbf{D}(\mathbf{D}^* \mathbf{g} - \hat{\mathbf{h}}^{\text{RL}}), \quad (4.60)$$

dont la constante de Lipschitz vaut $\|\mathbf{D}\|^2$ par définition de la norme d'opérateur $\|\cdot\|$. \square

Proposition 4.13 (Fonctionnelle à contours « libres » duale). *Le Problème (4.5) possède la formulation duale suivante :*

$$(\hat{\mathbf{g}}^{\text{L}}, \hat{\mathbf{u}}^{\text{L}}) = \underset{\mathbf{g}, \mathbf{u} \in \mathbb{R}^{2 \times N_1 \times N_2}}{\text{argmin}} F^*(-\lambda \alpha \mathbf{D}^* \mathbf{g}, -\lambda \mathbf{D}^* \mathbf{u}; \log_2 \mathcal{L}) + \iota_{\|\cdot\|_{2,\infty} \leq 1}(\mathbf{g}) + \iota_{\|\cdot\|_{2,\infty} \leq 1}(\mathbf{u}). \quad (4.61)$$

De plus, le terme d'attache aux données dual est différentiable, de gradient lipschitzien avec une constante de Lipschitz inférieure ou égale à $\|\mathbf{U}_{\lambda,\alpha}^{\text{L}}\|^2 \|\mathbf{J}^{-1}\|$.

Démonstration. La formulation du problème dual est obtenue en appliquant la Définition 4.4 et en s'appuyant sur la Proposition 4.7 qui fournit l'expression de la fonction convexe conjuguée de F et sur la Proposition 4.3 pour la conjuguée de la norme mixte $\ell_{2,1}$. Puis, on utilise la Proposition 4.8, fournissant l'expression du gradient de F^* ainsi que sa constante de Lipschitz. Enfin, on remarque que l'attache aux données duale consiste en la composition de l'opérateur $\mathbf{U}_{\lambda,\alpha}^{\text{L}}$ par F^* et la constante de Lipschitz du gradient de l'attache aux données duale est donc inférieure ou égale à $\|\mathbf{U}_{\lambda,\alpha}^{\text{L}}\|^2 \|\mathbf{J}^{-1}\|$. \square

Proposition 4.14 (Fonctionnelle à contours « co-localisés » duale). *Le Problème (4.7) possède la formulation duale suivante :*

$$(\hat{\mathbf{g}}^C, \hat{\mathbf{u}}^C) = \underset{\mathbf{g}, \mathbf{u} \in \mathbb{R}^{2 \times N_1 \times N_2}}{\operatorname{argmin}} F^*(-\lambda \alpha \mathbf{D}^* \mathbf{g}, -\lambda \mathbf{D}^* \mathbf{u}; \log_2 \mathcal{L}) + \iota_{\|\cdot\|_{2,\infty} \leq 1} \left(\begin{pmatrix} \mathbf{g} \\ \mathbf{u} \end{pmatrix} \right). \quad (4.62)$$

De plus, le terme d'attache aux données dual est différentiable, de gradient lipschitzien avec une constante de Lipschitz inférieure ou égale à $\|\mathbf{U}_{\lambda,\alpha}^C\|^2 \|\mathbf{J}^{-1}\|$.

Démonstration. La formulation du problème dual est obtenue en appliquant la Définition 4.4 et en s'appuyant sur la Proposition 4.7 qui fournit l'expression de la fonction convexe conjuguée de F et sur la Proposition 4.3 pour la conjuguée de la norme mixte $\ell_{2,1}$. Puis, on utilise la Proposition 4.8, fournissant l'expression du gradient de F^* ainsi que sa constante de Lipschitz. Enfin, on remarque que l'attache aux données duale consiste en la composition de l'opérateur $\mathbf{U}_{\lambda,\alpha}^C$ par F^* et la constante de Lipschitz du gradient de l'attache aux données duale est donc inférieure ou égale à $\|\mathbf{U}_{\lambda,\alpha}^C\|^2 \|\mathbf{J}^{-1}\|$. \square

4.4.4 Fast Iterative Soft Thresholding Algorithm (FISTA), version duale

Nous proposons à l'Algorithme 5, une implémentation FISTA pour la résolution du Problème (2.123), s'appuyant sur la fonctionnelle de Rudin-Osher-Fatemi duale, formulée à la Proposition 4.12. L'Algorithme 6 (resp. 7) présente l'implémentation FISTA pour la résolution du Problème (4.5) (resp. (4.7)) grâce à la formulation duale de la fonctionnelle à contours « libres » (resp. « co-localisés ») obtenue à la Proposition 4.13 (resp. 4.14). Les solutions des problèmes primaux sont retrouvées à partir des solutions des problèmes duaux grâce aux relations (4.22).

Le choix d'un paramètre d'inertie, noté b dans les Algorithmes 5, 6 et 7, est effectué en suivant [7], ce choix permettant de garantir la convergence des itérées, résumée par les théorèmes suivants.

Théorème 4.2 (Convergence de FISTA^{ROF}). *La suite $(\mathbf{h}^{[t]})_{t \in \mathbb{N}}$, générée par l'Algorithme 5, converge vers l'unique minimiseur \mathbf{h}^{ROF} de la fonctionnelle de Rudin-Osher-Fatemi avec un terme de fidélité à l'estimée $\hat{\mathbf{h}}^{\text{RL}}$, solution du Problème (2.123).*

Théorème 4.3 (Convergence de FISTA^L). *La suite $(\mathbf{h}^{[t]}, \mathbf{v}^{[t]})_{t \in \mathbb{N}}$, générée par l'Algorithme 6, converge vers l'unique minimiseur $(\mathbf{h}^L, \mathbf{v}^L)$ de la fonctionnelle « libre », solution du Problème (4.5).*

Théorème 4.4 (Convergence de FISTA^C). *La suite $(\mathbf{h}^{[t]}, \mathbf{v}^{[t]})_{t \in \mathbb{N}}$, générée par l'Algorithme 7, converge vers l'unique minimiseur $(\mathbf{h}^C, \mathbf{v}^C)$ de la fonctionnelle « co-localisée », solution du Problème (4.7).*

Démonstration. La formulation duale des Problèmes (2.123), (4.5) et (4.7) aux Propositions 4.12, 4.13 et 4.14, permet de particulariser le schéma algorithmique FISTA pour chacun de ces problèmes et fournit les Algorithmes 5, 6 et 7. En outre, grâce au calcul de la constante de Lipschitz du gradient de l'attache aux données duale aux Propositions 4.12, 4.13 et 4.14, le paramètre de descente γ des Algorithmes 5, 6 et 7 est fixé de manière à satisfaire les hypothèses de convergence requises par [7, Théorème 3] pour assurer la convergence de l'algorithme FISTA. \square

Remarque. L'implémentation de l'algorithme FISTA pour la minimisation de la fonctionnelle duale de (4.5) (resp. (4.7)), impliquant la pénalisation *libre* G^L (resp. *co-localisée* G^C), nécessite de connaître la constante de Lipschitz du gradient de l'attache aux données *duale* et donc d'avoir accès à la norme d'opérateur de $(\mathbf{U}_{(\lambda,\alpha)}^L)^* \Phi$ (resp. $(\mathbf{U}_{(\lambda,\alpha)}^C)^* \Phi$). Or cette norme ne possède pas d'expression analytique dépendant de α , $\|\mathbf{D}\|$ et $\|\Phi\|$. En pratique on utilise la méthode de la puissance pour estimer numériquement la norme de ces opérateurs linéaires [33, Section 13.5].

Algorithme 5 FISTA^{ROF} : Estimation ROF_h (Problème (2.123))

Entrée : Logarithme des coefficients d'ondelettes dominants $\{\log_2 \mathcal{L}_j, j = j_1, \dots, j_2\}$

Paramètre de régularisation $\lambda > 0$

Paramètres : $b > 2$, $\tau_0 = 1$ et $\gamma > 0$ tel que $\gamma \|\mathbf{D}\|^2 < 1$

Initialisation : Choisir $\mathbf{g}^{[0]} \in \mathbb{R}^{2 \times N_1 \times N_2}$ (variable duale)

$\bar{\mathbf{g}}^{[0]} \leftarrow \mathbf{g}^{[0]}$ (variable duale auxiliaire)

Régression linéaire (2.89) : $\forall \underline{n} \in \Omega, \quad h^{[0]}(\underline{n}) \leftarrow \sum_{j=j_1}^{j_2} \omega_j \log_2 \mathcal{L}_j(\underline{n})$ (variable primale)

for $t = 0$ **to** $T_{\max} - 1$ **do**

Mise à jour de la variable duale

$$\mathbf{g}^{[t+1]} = \text{prox}_{\gamma(\|\cdot\|_{2,1})} \left(\bar{\mathbf{g}}^{[t]} + \gamma \lambda \mathbf{D} \mathbf{h}^{[t]} \right)$$

Mise à jour du paramètre d'inertie de FISTA

$$\tau_{t+1} = \frac{t+b}{b}$$

Mise à jour de la variable auxiliaire

$$\bar{\mathbf{g}}^{[t+1]} = \mathbf{g}^{[t+1]} + \frac{\tau_t - 1}{\tau_{t+1}} (\mathbf{g}^{[t+1]} - \mathbf{g}^{[t]})$$

Mise à jour de la variable primale

$$\mathbf{h}^{[t+1]} = \mathbf{h}^{[t]} - \lambda \mathbf{D}^* (\mathbf{g}^{[t+1]} - \mathbf{g}^{[t]})$$

end for

Sortie : $\hat{\mathbf{h}}^{\text{ROF}} = \mathbf{h}^{[T_{\max}]}$ régularisation par Variation Totale

Algorithme 6 FISTA^L : Estimation à contours « libres » (Problème (4.5))

Entrée : Logarithme des coefficients d'ondelettes dominants $\{\log_2 \mathcal{L}_j, j = j_1, \dots, j_2\}$

Paramètres de régularisation $\lambda > 0$, $\alpha > 0$

Paramètres : $b > 2$, $\tau_0 = 1$ et $\gamma > 0$ tel que $\gamma \|\mathbf{J}^{-1}\| \|\mathbf{U}_{(\lambda, \alpha)}^L\|^2 < 1$, $\mathbf{U}_{(\lambda, \alpha)}^L$ défini à l'Équation (4.55)

Initialisation : Choisir $\mathbf{g}^{[0]} \in \mathbb{R}^{2 \times N_1 \times N_2}$, $\mathbf{u}^{[0]} \in \mathbb{R}^{2 \times N_1 \times N_2}$ (variables duales)

$\bar{\mathbf{g}}^{[0]} \leftarrow \mathbf{g}^{[0]}$, $\bar{\mathbf{u}}^{[0]} \leftarrow \mathbf{u}^{[0]}$ (variables duales auxiliaires)

$\forall \underline{n} \in \Omega, \quad (\mathbf{h}(\underline{n})^{[0]}, \mathbf{v}(\underline{n})^{[0]})^\top \leftarrow \mathbf{J}^{-1} (\mathcal{T}_{\underline{n}}, \mathcal{S}_{\underline{n}})^\top$ (variables primales), avec \mathbf{J} , $\mathcal{S}_{\underline{n}}$ et $\mathcal{T}_{\underline{n}}$ définis à l'Équation (2.91)

for $t = 0$ **to** $T_{\max} - 1$ **do**

Mise à jour des variables duales

$$\mathbf{g}^{[t+1]} = \text{prox}_{\gamma(\|\cdot\|_{2,1})} \left(\bar{\mathbf{g}}^{[t]} + \gamma \lambda \alpha \mathbf{D} \mathbf{h}^{[t]} \right)$$

$$\mathbf{u}^{[t+1]} = \text{prox}_{\gamma(\|\cdot\|_{2,1})} \left(\bar{\mathbf{u}}^{[t]} + \gamma \lambda \mathbf{D} \mathbf{v}^{[t]} \right)$$

Mise à jour du paramètre d'inertie de FISTA

$$\tau_{t+1} = \frac{t+b}{b}$$

Mise à jour des variables auxiliaires

$$\bar{\mathbf{g}}^{[t+1]} = \mathbf{g}^{[t+1]} + \frac{\tau_t - 1}{\tau_{t+1}} (\mathbf{g}^{[t+1]} - \mathbf{g}^{[t]})$$

$$\bar{\mathbf{u}}^{[t+1]} = \mathbf{u}^{[t+1]} + \frac{\tau_t - 1}{\tau_{t+1}} (\mathbf{u}^{[t+1]} - \mathbf{u}^{[t]})$$

Mise à jour des variables primales

$$(\mathbf{h}^{[t+1]}, \mathbf{v}^{[t+1]}) = (\mathbf{h}^{[t]}, \mathbf{v}^{[t]}) - (\Phi^* \Phi)^{-1} (\lambda \alpha \mathbf{D}^* (\mathbf{g}^{[t+1]} - \mathbf{g}^{[t]}), \lambda \mathbf{D}^* (\mathbf{u}^{[t+1]} - \mathbf{u}^{[t]}))$$

end for

Sortie : $(\hat{\mathbf{h}}^L, \hat{\mathbf{v}}^L) = (\mathbf{h}^{[T_{\max}]}, \mathbf{v}^{[T_{\max}]})$ régularisation par pénalisation à contours « libres »

4.4.5 Algorithme primal-dual accéléré grâce à la forte-convexité

L'algorithme primal-dual de Chambolle *et coll.* [8, Algorithme 2] est mis en œuvre pour la résolution des Problèmes (2.123), (4.5) et (4.7), en s'appuyant sur le résultat de forte-convexité du terme d'attache aux données, $F(\mathbf{h}, \mathbf{v}; \log_2 \mathcal{L})$, des fonctionnelles à contours « libres » et à contours

Algorithme 7 FISTA^C : Estimation à contours « co-localisés » (Problème (4.7))

Entrée : Logarithme des coefficients d'ondelettes dominants $\{\log_2 \mathcal{L}_j, j = j_1, \dots, j_2\}$

Paramètres de régularisation $\lambda > 0, \alpha > 0$

Paramètres : $b > 2, \tau_0 = 1$ et $\gamma > 0$ tel que $\gamma \|\mathbf{J}^{-1}\| \|\mathbf{U}_{(\lambda, \alpha)}^{\mathbf{C}}\|^2 < 1, \mathbf{U}_{(\lambda, \alpha)}^{\mathbf{C}}$ défini à l'Équation (4.56)

Initialisation : Choisir $\mathbf{g}^{[0]} \in \mathbb{R}^{2 \times N_1 \times N_2}, \mathbf{u}^{[0]} \in \mathbb{R}^{2 \times N_1 \times N_2}$ (variables duales)

$\bar{\mathbf{g}}^{[0]} \leftarrow \mathbf{g}^{[0]}, \bar{\mathbf{u}}^{[0]} \leftarrow \mathbf{u}^{[0]}$ (variables duales auxiliaires)

$\forall \underline{n} \in \Omega, (h(\underline{n})^{[0]}, v(\underline{n})^{[0]})^\top \leftarrow \mathbf{J}^{-1}(\mathcal{T}_{\underline{n}}, \mathcal{S}_{\underline{n}})^\top$ (variables primales), avec $\mathbf{J}, \mathcal{S}_{\underline{n}}$ et $\mathcal{T}_{\underline{n}}$ définis à l'Équation (2.91)

for $t = 0$ **to** $T_{\max} - 1$ **do**

Mise à jour des variables duales

$$\begin{pmatrix} \mathbf{g}^{[t+1]} \\ \mathbf{u}^{[t+1]} \end{pmatrix} = \text{prox}_{\gamma(\|\cdot\|_{2,1})^*} \left(\begin{pmatrix} \bar{\mathbf{g}}^{[t]} + \gamma \lambda \alpha \mathbf{D} \mathbf{h}^{[t]} \\ \bar{\mathbf{u}}^{[t]} + \gamma \lambda \mathbf{D} \mathbf{v}^{[t]} \end{pmatrix} \right)$$

Mise à jour du paramètre d'inertie

$$\tau_{t+1} = \frac{t+b}{b}$$

Mise à jour des variables auxiliaires

$$\bar{\mathbf{g}}^{[t+1]} = \mathbf{g}^{[t+1]} + \frac{\tau_t - 1}{\tau_{t+1}} (\mathbf{g}^{[t+1]} - \mathbf{g}^{[t]})$$

$$\bar{\mathbf{u}}^{[t+1]} = \mathbf{u}^{[t+1]} + \frac{\tau_t - 1}{\tau_{t+1}} (\mathbf{u}^{[t+1]} - \mathbf{u}^{[t]})$$

Mise à jour des variables primales

$$(h^{[t+1]}, v^{[t+1]}) = (h^{[t]}, v^{[t]}) - (\Phi^* \Phi)^{-1} (\lambda \alpha \mathbf{D}^* (\mathbf{g}^{[t+1]} - \mathbf{g}^{[t]}), \lambda \mathbf{D}^* (\mathbf{u}^{[t+1]} - \mathbf{u}^{[t]}))$$

end for

Sortie : $(\hat{\mathbf{h}}^{\mathbf{C}}, \hat{\mathbf{v}}^{\mathbf{C}}) = (\mathbf{h}^{[T_{\max}]}, \mathbf{v}^{[T_{\max}]})$ régularisation par pénalisation à contours « co-localisés »

« co-localisés », démontré à la Section 4.3.4. La stratégie d'accélération consiste, à chaque itération, à mettre à jour les pas de descente δ_t et v_t des Algorithmes 9 et 10 grâce à un facteur ϑ_t dépendant du module de forte-convexité μ_F , et elle est alors d'autant plus efficace que μ_F est grand. Ainsi, le calcul *exact* de μ_F à la Section 4.3.4 permet d'obtenir la plus forte accélération possible.

Le mécanisme d'accélération sous-jacent peut se comprendre en considérant que, schématiquement, le pas de descente δ_t correspond à un pas de minimisation de l'attache aux données, fortement convexe, donc « facile » à minimiser, tandis que le pas v_t est associé à la minimisation de la pénalisation, plus « difficile » car uniquement convexe. Au fur et à mesure des itérations le pas δ_t diminue, tandis que le pas v_t augmente en proportion, ce que l'on peut interpréter comme une concentration des efforts de minimisation sur le terme de pénalisation, le plus « plat » autour du minimum.

Les Théorèmes 4.6 et 4.7 assurent la convergence des Algorithmes 9 et 10 vers les solutions respectives des Problèmes (4.5) et (4.7).

Théorème 4.5 (Convergence de PD_{ROF}). *La suite $(\mathbf{h}^{[\ell]})_{\ell \in \mathbb{N}}$, générée par l'Algorithme 8 converge vers l'unique minimiseur \mathbf{h}^{ROF} de la fonctionnelle de Rudin-Osher-Fatemi avec un terme de fidélité à l'estimée $\hat{\mathbf{h}}^{\text{RL}}$, solution du Problème (2.123).*

Théorème 4.6 (Convergence de PD^L). *La suite $(\mathbf{h}^{[\ell]}, \mathbf{v}^{[\ell]})_{\ell \in \mathbb{N}}$, générée par l'Algorithme 9 converge vers l'unique minimiseur $(\mathbf{h}^{\text{L}}, \mathbf{v}^{\text{L}})$ de la fonctionnelle « libre », solution du Problème (4.5).*

Théorème 4.7 (Convergence de PD^C). *La suite $(\mathbf{h}^{[\ell]}, \mathbf{v}^{[\ell]})_{\ell \in \mathbb{N}}$, générée par l'Algorithme 10 converge vers l'unique minimiseur $(\mathbf{h}^{\text{C}}, \mathbf{v}^{\text{C}})$ de la fonctionnelle « co-localisée », solution du Problème (4.7).*

Démonstration. Les démonstrations des Théorèmes 4.5, 4.6 et 4.7 découlent directement du choix des paramètres de descente effectué dans les Algorithmes 8, et 10 et de la prise en compte du module de forte-convexité qui permettent ensuite d'appliquer les résultats de convergence démontrés par [8, Théorème 2]. \square

Remarque. De même que pour les algorithmes FISTA ci-dessus les normes des opérateurs $\mathbf{U}_{(\lambda,\alpha)}^L$ et $\mathbf{U}_{(\lambda,\alpha)}^C$ sont calculées numériquement *via* la méthode de la puissance [33, Section 13.5].

Algorithme 8 PD^{ROF} : Estimation ROF_h (Problème (2.123)).

Entrée : Logarithme des coefficients d'ondelettes dominants $\{\log_2 \mathcal{L}_j, j = j_1, \dots, j_2\}$

Paramètre de régularisation $\lambda > 0$

Paramètres : $\tau_0 > 0, \nu_0 > 0$ tels que $\tau_0 \nu_0 \|\mathbf{D}\|^2 < 1$,

$\mu = 1$ (attache aux données 1-fortement convexe)

Initialisation : Choisir $\mathbf{h} \in \mathbb{R}^{N_1 \times N_2}$ (variable primale)

$\mathbf{g} \leftarrow \mathbf{D}\mathbf{h} \in \mathbb{R}^{2 \times N_1 \times N_2}$ (variable duale)

$\bar{\mathbf{g}} \leftarrow \mathbf{g} \in \mathbb{R}^{2 \times N_1 \times N_2}$ (variable duale auxiliaire)

for $t = 0$ **to** $T_{\max} - 1$ **do**

Mise à jour de la variable primale

$$\mathbf{h}^{[t+1]} = \text{prox}_{\frac{\tau_t}{2} \|\cdot\|_{\text{Fro}} - \hat{\mathbf{h}}^{\text{RL}}}_{\|\cdot\|_{\text{Fro}}^2} \left(\mathbf{h}^{[t]} - \tau_t \mathbf{D}^* \bar{\mathbf{g}}^{[t]} \right)$$

Mise à jour de la variable duale

$$\mathbf{g}^{[t+1]} = \text{prox}_{\nu_t (\lambda \|\cdot\|_{2,1})^*} \left(\mathbf{g}^{[t]} + \nu_t \mathbf{D}\mathbf{h}^{[t]} \right)$$

Mise à jour des pas de descente

$$\vartheta_t = (1 + 2\mu\tau_t)^{-1/2}, \quad \tau_{t+1} = \vartheta_t \tau_t, \quad \nu_{t+1} = \nu_t \vartheta_t$$

Mise à jour de la variable duale auxiliaire

$$\bar{\mathbf{g}}^{[t+1]} = \mathbf{g}^{[t+1]} + \vartheta_t (\mathbf{g}^{[t+1]} - \mathbf{g}^{[t]})$$

end for

Sortie : $\hat{\mathbf{h}}^{\text{ROF}} = \mathbf{h}^{[T_{\max}]}$ régularisation par Variation Totale

Algorithme 9 PD^L : Estimation à contours « libres » (Problème (4.5))

Entrée : Logarithme des coefficients d'ondelettes dominants $\{\log_2 \mathcal{L}_j, j = j_1, \dots, j_2\}$

Paramètres de régularisation $\lambda > 0, \alpha > 0$

Paramètres : $\delta_0 > 0, \nu_0 > 0$ tels que $\delta_0 \nu_0 \|\mathbf{U}_{(\lambda,\alpha)}^L\|^2 < 1$, $\mathbf{U}_{(\lambda,\alpha)}^L$ défini à l'Équation (4.55)

Initialisation : Choisir $\mathbf{h}^{[0]} \in \mathbb{R}^{N_1 \times N_2}, \mathbf{v}^{[0]} \in \mathbb{R}^{N_1 \times N_2}$ (variables primales)

$\mathbf{g}^{[0]} \leftarrow \mathbf{D}\mathbf{h}^{[0]}, \bar{\mathbf{g}}^{[0]} \leftarrow \mathbf{g}^{[0]}$ (variables duales)

$\mathbf{u}^{[0]} \leftarrow \mathbf{D}\mathbf{v}^{[0]}, \bar{\mathbf{u}}^{[0]} \leftarrow \mathbf{u}^{[0]}$ (variables duales auxiliaires)

for $t \in \mathbb{N}^*$ **do**

Mise à jour des variables primales

$$\begin{pmatrix} \mathbf{h}^{[t+1]} \\ \mathbf{v}^{[t+1]} \end{pmatrix} = \text{prox}_{\delta_t \text{F}} \left(\begin{pmatrix} \mathbf{h}^{[t]} \\ \mathbf{v}^{[t]} \end{pmatrix} - \delta_t \begin{pmatrix} \lambda \alpha \mathbf{D}^* \bar{\mathbf{g}}^{[t]} \\ \lambda \mathbf{D}^* \bar{\mathbf{u}}^{[t]} \end{pmatrix} \right)$$

Mise à jour des variables duales

$$\mathbf{g}^{[t+1]} = \text{prox}_{\nu_t (\|\cdot\|_{2,1})^*} \left(\mathbf{g}^{[t]} + \nu_t \lambda \alpha \mathbf{D}\mathbf{h}^{[t]} \right)$$

$$\mathbf{u}^{[t+1]} = \text{prox}_{\nu_t (\|\cdot\|_{2,1})^*} \left(\mathbf{u}^{[t]} + \nu_t \lambda \mathbf{D}\mathbf{v}^{[t]} \right)$$

Mise à jour des paramètres de descente

$$\vartheta_t = (1 + 2\mu\delta_t)^{-1/2}, \quad \delta_{t+1} = \vartheta_t \delta_t, \quad \nu_{t+1} = \nu_t \vartheta_t$$

Mise à jour des variables auxiliaires

$$\begin{pmatrix} \bar{\mathbf{g}}^{[t+1]} \\ \bar{\mathbf{u}}^{[t+1]} \end{pmatrix} = \begin{pmatrix} \mathbf{g}^{[t+1]} \\ \mathbf{u}^{[t+1]} \end{pmatrix} + \vartheta_t \left(\begin{pmatrix} \mathbf{g}^{[t+1]} \\ \mathbf{u}^{[t+1]} \end{pmatrix} - \begin{pmatrix} \mathbf{g}^{[t]} \\ \mathbf{u}^{[t]} \end{pmatrix} \right)$$

end for

Sortie : $(\hat{\mathbf{h}}^L, \hat{\mathbf{v}}^L) = (\mathbf{h}^{[T_{\max}]}, \mathbf{v}^{[T_{\max}]})$ régularisation par pénalisation à contours « libres »

Algorithme 10 PDC^C : Estimation à contours « co-localisés » (Problème (4.7))

Entrée : Logarithme des coefficients d'ondelettes dominants $\{\log_2 \mathcal{L}_j, j = j_1, \dots, j_2\}$

Paramètres de régularisation $\lambda > 0, \alpha > 0$

Paramètres : $\delta_0 > 0, \nu_0 > 0$ tels que $\delta_0 \nu_0 \|\mathbf{U}_{(\lambda, \alpha)}^C\|^2 < 1$, $\mathbf{U}_{(\lambda, \alpha)}^C$ défini à l'Équation (4.56)

Initialisation : Choisir $\mathbf{h}^{[0]} \in \mathbb{R}^{N_1 \times N_2}$, $\mathbf{v}^{[0]} \in \mathbb{R}^{N_1 \times N_2}$ (variables primales)

$\mathbf{g}^{[0]} \leftarrow \mathbf{D}\mathbf{h}^{[0]}$, $\bar{\mathbf{g}}^{[0]} \leftarrow \mathbf{g}^{[0]}$ (variables duales)

$\mathbf{u}^{[0]} \leftarrow \mathbf{D}\mathbf{v}^{[0]}$, $\bar{\mathbf{u}}^{[0]} \leftarrow \mathbf{u}^{[0]}$ (variables duales auxiliaires)

for $t \in \mathbb{N}^*$ **do**

Mise à jour des variables duales

$$\begin{pmatrix} \mathbf{h}^{[t+1]} \\ \mathbf{v}^{[t+1]} \end{pmatrix} = \text{prox}_{\delta_t F} \left(\begin{pmatrix} \mathbf{h}^{[t]} \\ \mathbf{v}^{[t]} \end{pmatrix} - \delta_t \begin{pmatrix} \lambda \alpha \mathbf{D}^* \bar{\mathbf{g}}^{[t]} \\ \lambda \mathbf{D}^* \bar{\mathbf{u}}^{[t]} \end{pmatrix} \right)$$

Mise à jour des variables duales

$$\begin{pmatrix} \mathbf{g}^{[t+1]} \\ \mathbf{u}^{[t+1]} \end{pmatrix} = \text{prox}_{\nu_t (\|\cdot\|_{2,1})^*} \left(\begin{pmatrix} \mathbf{g}^{[t]} + \nu_t \lambda \alpha \mathbf{D}\mathbf{h}^{[t]} \\ \mathbf{u}^{[t]} + \nu_t \lambda \mathbf{D}\mathbf{v}^{[t]} \end{pmatrix} \right)$$

Mise à jour des pas de descente

$$\vartheta_t = (1 + 2\mu\delta_t)^{-1/2}, \delta_{t+1} = \vartheta_t \delta_t, \nu_{t+1} = \nu_t / \vartheta_t$$

Mise des variables auxiliaires

$$\begin{pmatrix} \bar{\mathbf{g}}^{[t+1]} \\ \bar{\mathbf{u}}^{[t+1]} \end{pmatrix} = \begin{pmatrix} \mathbf{g}^{[t+1]} \\ \mathbf{u}^{[t+1]} \end{pmatrix} + \vartheta_t \left(\begin{pmatrix} \mathbf{g}^{[t+1]} \\ \mathbf{u}^{[t+1]} \end{pmatrix} - \begin{pmatrix} \mathbf{g}^{[t]} \\ \mathbf{u}^{[t]} \end{pmatrix} \right)$$

end for

Sortie : $(\hat{\mathbf{h}}^C, \hat{\mathbf{v}}^C) = (\mathbf{h}^{[T_{\max}]}, \mathbf{v}^{[T_{\max}]})$ régularisation par pénalisation à contours « co-localisés »

4.4.6 Critères d'arrêt

Afin de pouvoir comparer de manière fiable les différents schémas d'optimisation, il est nécessaire de se donner un critère de convergence commun à tous, qui permette de stopper les algorithmes à un même niveau d'approximation de la solution. Pour cela, nous comparons le critère classique, construit à partir des accroissements de la fonctionnelle à minimiser et un critère s'appuyant sur le « saut de dualité » (*duality gap*), que nous explicitons pour les fonctionnelles à contours « libres » et à contours « co-localisés ».

Définition 4.7 (Accroissements normalisés). Pour un problème de minimisation de la forme de l'Équation (4.20), et une suite minimisante $(\mathbf{x}^{[t]})_{t \in \mathbb{N}}$, par exemple obtenue par un algorithme proximal du type FISTA ou primal-dual, les accroissements normalisés sont définis à chaque itération de l'algorithme par

$$\tilde{\omega}_t = \frac{|\varphi(\mathbf{x}^{[t]}) + \|\mathbf{U}\mathbf{x}^{[t]}\| - \varphi(\mathbf{x}^{[t-1]}) - \|\mathbf{U}\mathbf{x}^{[t-1]}\|}{|\varphi(\mathbf{x}^{[t]}) + \|\mathbf{U}\mathbf{x}^{[t]}\|}. \quad (4.63)$$

La Figure 4.4a illustre l'évolution de la suite des accroissements normalisés $(\tilde{\omega}_t)_{t \in \mathbb{N}}$ au fur et à mesure des itérations de l'Algorithme primal-dual 9, sur un exemple de résolution du Problème (4.5), avec différents couples d'hyperparamètres (λ, α) . La décroissance particulièrement irrégulière des accroissements normalisés rend leur utilisation pour la construction d'un critère d'arrêt difficile et peu fiable. Nous allons donc nous tourner vers des outils d'analyse convexe plus sophistiqués.

Définition 4.8 (Saut de dualité (*duality gap*)). [3, Définition 15.10] Le « saut de dualité », noté $\delta\Gamma$, associé aux problèmes primaux et duaux de la Définition 4.4 est la différence entre la fonctionnelle primale et la fonctionnelle duale aux points solutions. En définissant la fonctionnelle primale-

duale

$$\Gamma(\mathbf{x}; \mathbf{y}) := \varphi(\mathbf{x}) + \|\mathbf{U}\mathbf{x}\| - (-\varphi^*(-\mathbf{U}^*\mathbf{y}) - \|\mathbf{y}\|^*), \quad (4.64)$$

où φ^* (resp. $\|\cdot\|^*$) désigne la conjuguée de Fenchel de φ (resp. de $\|\cdot\|$), le saut de dualité est défini de manière équivalente comme la borne inférieure suivante :

$$\delta\Gamma := \inf_{(\mathbf{x}, \mathbf{y}) \in \mathcal{H} \times \mathcal{G}} \Gamma(\mathbf{x}; \mathbf{y}). \quad (4.65)$$

Proposition 4.15 (Forte dualité). [3, Proposition 15.13] *Sous les hypothèses de régularité sur φ , $\|\cdot\|$ et \mathbf{U} , constituant le cadre de la « forte dualité », le gap de dualité $\delta\Gamma$ est nul. De plus, les solutions respectives $\hat{\mathbf{x}}$ et $\hat{\mathbf{y}}$ du problème primal et du problème dual sont alors caractérisées par l'annulation du gap de dualité, i.e., $\Gamma(\hat{\mathbf{x}}; \hat{\mathbf{y}}) = \delta\Gamma = 0$ est équivalent à dire que $\hat{\mathbf{x}}$ est solution du Problème primal (4.20) et $\hat{\mathbf{y}}$ est solution du Problème dual (4.21).*

Or, par construction, les algorithmes FISTA et primaux-duaux, présentés ci-dessus, construisent chacun un couple de suites minimisantes $(\mathbf{x}^{[t]}, \mathbf{y}^{[t]})_{t \in \mathbb{N}}$ pour les fonctionnelles Γ^L (resp. Γ^C) associées respectivement aux Problèmes (4.5) (resp (4.7)) avec $\varphi = F$, $\mathbf{U} = \mathbf{U}_{(\lambda, \alpha)}^L$ (resp. $\mathbf{U} = \mathbf{U}_{(\lambda, \alpha)}^C$), $\|\cdot\| = \|\cdot\|_{2,1}$ et

$$\mathbf{x}^{[t]} = \begin{pmatrix} \mathbf{h}^{[t]} \\ \mathbf{v}^{[t]} \end{pmatrix} \text{ (variable primale), } \quad \mathbf{y}^{[t]} = \begin{pmatrix} \mathbf{g}^{[t]} \\ \mathbf{u}^{[t]} \end{pmatrix} \text{ (variable duale).} \quad (4.66)$$

Par conséquent la convergence des Algorithmes 6 et 9 (resp. 7 et 10) se traduit sur le comportement la fonctionnelle primale-duale Γ^L (resp. Γ^C) par

$$\Gamma^L(\mathbf{h}^{[t]}, \mathbf{v}^{[t]}; \mathbf{g}^{[t]}, \mathbf{u}^{[t]}) \xrightarrow{t \rightarrow +\infty} 0 \quad (\text{resp. } \Gamma^C(\mathbf{h}^{[t]}, \mathbf{v}^{[t]}; \mathbf{g}^{[t]}, \mathbf{u}^{[t]}) \xrightarrow{t \rightarrow +\infty} 0). \quad (4.67)$$

L'évaluation pratique des fonctionnelles Γ^L et Γ^C est effectuée en appliquant la Formule (4.64) respectivement aux Problèmes (4.5) et (4.7), nécessitant donc

- la conjuguée convexe de l'attache aux données, F^* , explicitée à la Proposition 4.7,
- le calcul des opérateurs adjoints $\mathbf{U}_{(\lambda, \alpha)}^L$ et $\mathbf{U}_{(\lambda, \alpha)}^C$, qui sont obtenus par un calcul direct à partir de l'opérateur adjoint \mathbf{D}^* du gradient discret,
- et la conjuguée convexe de la norme $\ell_{2,1}$, qui est obtenue grâce à la Proposition 4.3 et s'écrit $(\gamma \|\cdot\|_{2,1})^* = \mathbf{1}_{\|\cdot\|_{2,+\infty} \leq \gamma}$.

Afin de mesurer la convergence des différents algorithmes de manière la plus objective possible, la fonctionnelle primale-duale de l'Équation (4.64) est *normalisée* ce qui donne lieu à la définition suivante.

Définition 4.9 (Fonctionnelle primale-duale normalisée). La fonctionnelle primale-duale *normalisée* associée aux problèmes primaux et duaux de la Définition 4.4 s'écrit

$$\tilde{\Gamma}(\mathbf{x}; \mathbf{y}) := \frac{\Gamma(\mathbf{x}; \mathbf{y})}{|\varphi(\mathbf{x}) + \|\mathbf{U}\mathbf{x}\| + |(\varphi^*(-\mathbf{U}^*\mathbf{y}) + \|\mathbf{y}\|^*)|}. \quad (4.68)$$

Pour un couple de suites minimisantes, $(\mathbf{x}^{[t]}, \mathbf{y}^{[t]})_{t \in \mathbb{N}}$, de la fonctionnelle Γ définie à l'Équation (4.64), on définit la suite des « écarts de dualité normalisés » comme

$$\tilde{\Gamma}_t \triangleq \tilde{\Gamma}(\mathbf{x}^{[t]}; \mathbf{y}^{[t]}). \quad (4.69)$$

L'évolution de l'écart de dualité normalisé, $\tilde{\Gamma}_t$, est tracée à la Figure 4.4b, pour les mêmes appels à l'Algorithme 9 que ceux de la Figure 4.4a. L'évolution beaucoup plus régulière de l'écart de dualité normalisé en fait un critère bien plus efficace pour l'arrêt des algorithmes FISTA et primaux-duaux. De plus, grâce à la normalisation effectuée, un même critère d'arrêt sur l'écart de dualité peut être choisi, indépendamment des paramètres de régularisation (λ, α) .

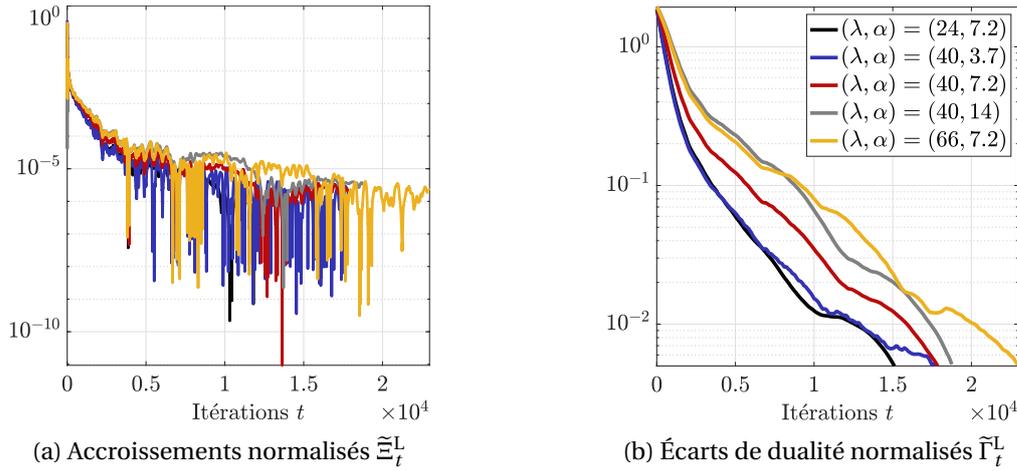


FIGURE 4.4 – Critères d'arrêt pour les algorithmes proximaux. Convergence de l'Algorithme 9 résolvant le problème à contours « libres » (4.5) pour cinq paires d'hyperparamètres (λ, α) évaluée via : (a) Les accroissements normalisés de la fonction objectif $\tilde{\Xi}_t^L$ (voir Définition 4.7), (b) Les écarts de dualité normalisés $\tilde{\Gamma}_t^L$ (voir Définition 4.9).

4.5 Performance d'estimation et de segmentation

Dans cette section, les méthodes de segmentation à contours « libres » et à contours « localisés », reposant sur la prise en compte conjointe de la régularité et de la puissance locales, sont comparées de manière exhaustive entre elles et avec la méthode ROF $_{\mathbf{h}}$ -S, décrite à la Section 2.3.4, s'appuyant uniquement sur la régularité locale, au moyen de simulations de Monte Carlo. Puis ces trois méthodes sont mises en œuvre pour la segmentation d'assemblages de textures réelles et comparées dans cette tâche avec la méthode de l'état-de-l'art proposée par Yuan *et coll.* [36], décrite à la Section 2.3.3.1.

4.5.1 Stratégie de seuillage

L'information de texture étant *principalement* contenue dans la régularité locale, nous nous concentrons sur les estimées régularisées $\hat{\mathbf{h}}^{\text{ROF}}$, $\hat{\mathbf{h}}^{\text{L}}$ et $\hat{\mathbf{h}}^{\text{C}}$. Afin d'obtenir à partir de chaque estimée une segmentation en k régions, l'algorithme de seuillage itératif proposé par [6], et décrit à la Section 2.1.1.6, est légèrement modifié pour s'adapter au problème de segmentation de textures.

Ce seuillage itératif adapté, détaillé à l'Algorithme 11, est appliqué à chaque estimée $\hat{\mathbf{h}}^\bullet$ et fournit des estimées seuillées, $S\hat{\mathbf{h}}^\bullet$, de la régularité locale prenant *exactement* k valeurs $\hat{H}_1^\bullet, \dots, \hat{H}_k^\bullet$, ainsi que les segmentations associées

$$\Omega = \cup_{i=1}^k \hat{\Omega}_i^\bullet, \quad \text{telle que} \quad \forall i \in \{1, \dots, k\}, \forall \underline{n} \in \hat{\Omega}_i^\bullet, \quad S\hat{\mathbf{h}}^\bullet(\underline{n}) = \hat{H}_i^\bullet, \quad (4.70)$$

où $\bullet \in \{\text{ROF}, \text{L}, \text{C}\}$ et avec \hat{H}_i^\bullet la régularité locale moyenne sur la région $\hat{\Omega}_i^\bullet$

4.5.2 Protocole de comparaison

4.5.2.1 Échantillons de textures synthétiques

L'Algorithme 4 est utilisé pour générer des textures fractales homogènes par morceaux, dont la géométrie est calquée sur le masque de la Figure 4.5a, de résolution 512×512 pixels. Ces images texturées sont composées de deux régions : une texture de fond, caractérisée par sa régularité et sa variance locales $(H_1, \Sigma_1^2) = (0, 5; 0, 6)$ qui sont fixées dans toutes les expériences, et une texture remplissant l'ellipse centrale, caractérisée par une régularité locale, H_2 , et une variance locale, Σ_2^2 , variables d'une configuration à une autre. Les sept configurations d'attributs fractals explorées sont présentées en Figure 4.5b, à partir des écarts de régularité locale, $\Delta H \triangleq H_2 - H_1$, et de

Algorithme 11 Seuillage itératif sur \hat{h}^\bullet pour k régions.

Entrée : \hat{h}^\bullet : carte de régularité à seuiller

Paramètres : k : nombre de régions recherchées

M_{\max} : nombre d'itérations

Initialisation : Choisir $H_1^{[0]} = \min_{\underline{n} \in \Omega} h(\underline{n}) < H_2^{[0]} < \dots < H_k^{[0]} = \max_{\underline{n} \in \Omega} h(\underline{n})$

for $m = 0$ **to** $M_{\max} - 1$ **do**

Calcul des seuils

$T_0^{[m]} = -\infty$ et $T_k^{[m]} = +\infty$ (convention)

for $i = 1$ **to** $k - 1$ **do**

$T_i^{[m]} = \frac{1}{2} (H_i^{[m]} + H_{i+1}^{[m]})$

end for

Seuillage de h

for $i = 1$ **to** k **do**

$\Omega_i^{[m]} = \{ \underline{n} \text{ tels que } T_{i-1}^{[m]} \leq h(\underline{n}) \leq T_i^{[m]} \}$

end for

Mise à jour des moyennes sur les régions

for $i = 1$ **to** k **do**

$H_i^{[m+1]} = \frac{1}{|\Omega_i^{[m]}|} \sum_{\underline{n} \in \Omega_i^{[m]}} h(\underline{n})$

end for

end for

Carte seuillée à k valeurs

for $i = 1$ **to** k **do**

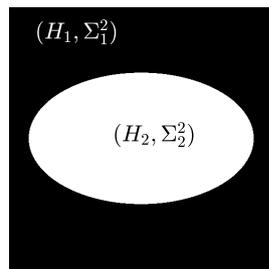
$Sh(\underline{n}) \leftarrow H_i^{[M_{\max}]}$ si $\underline{n} \in \Omega_i^{[M_{\max}]}$

end for

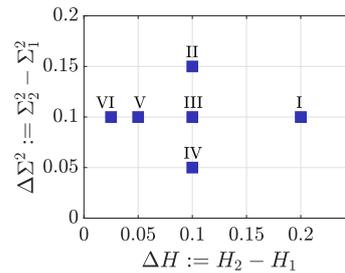
Sortie : $S\hat{h}^\bullet = Sh$: régularité locale seuillée,

$\Omega = \cup_{i=1}^k \hat{\Omega}_i^\bullet = \cup_{i=1}^k \Omega_i^{[M_{\max}]}$: partition associée

variance locale, $\Delta\Sigma^2 \triangleq \Sigma_2^2 - \Sigma_1^2$, entre la texture de fond et la texture centrale. Pour chaque configuration, cinq réalisations de la texture fractale homogène par morceaux sont réalisées, de manière à pouvoir fournir des performances moyennées.



(a) Masque



(b) Écarts ΔH et $\Delta\Sigma^2$ explorés.

FIGURE 4.5 – Configurations de textures synthétiques pour la comparaison des approches de segmentation ROF $_{h-S}$ (2.123), à contours « libres » (4.5) et à contours « co-localisés ».

4.5.2.2 Transformée en ondelettes

Une décomposition en ondelettes bidimensionnelle, non décimée, est calculée aux échelles $(2^j)_{j_1}^{j_2}$, avec une ondelette mère construite comme le produit tensoriel d'ondelettes de Daubechies asymétriques à $N_\psi = 3$ moments nuls [23], comme décrit en Définition 2.9, au Chapitre 2. Les coefficients d'ondelettes dominants sont calculés en appliquant la Définition 2.10, introduite

à la Section 2.2.3.2. L'estimation de la régularité locale fait intervenir la gamme d'octaves $(j_1, j_2) = (2, 5)$, la première octave, $j_1 = 1$ étant exclue à cause du biais des coefficients dominants à cette échelle [34, 35].

4.5.2.3 Paramètres des algorithmes de minimisation

Les pas de descente des algorithmes FISTA et primaux-duaux sont choisis les plus grands possible afin de garantir une convergence rapide, ce qui amène aux choix suivants

- pour l'Algorithme 6, FISTA^L : $\gamma = 0,99 / \left(\|\mathbf{J}^{-1}\| \|\mathbf{U}_{(\lambda,\alpha)}^L\|^2 \right)$,
- pour l'Algorithme 7, FISTA^C : $\gamma = 0,99 / \left(\|\mathbf{J}^{-1}\| \|\mathbf{U}_{(\lambda,\alpha)}^C\|^2 \right)$,
- pour l'Algorithme 9, PD^L : $\delta_0 = \nu_0 = 0,99 / \|\mathbf{U}_{(\lambda,\alpha)}^L\|$,
- pour l'Algorithme 10, PD^C : $\delta_0 = \nu_0 = 0,99 / \|\mathbf{U}_{(\lambda,\alpha)}^C\|$,

où $\mathbf{U}_{(\lambda,\alpha)}^L$ et $\mathbf{U}_{(\lambda,\alpha)}^C$ sont les opérateurs linéaires respectivement associés aux pénalisations « libre » et « co-localisée », définis aux Équations (4.55) et (4.56), et $\mathbf{J} = \Phi^* \Phi$ est l'opérateur linéaire explicite à l'Équation (2.91). La norme $\|\mathbf{J}^{-1}\|$ dépend uniquement des octaves prises en compte et pour la gamme d'octave considérée, $(j_1, j_2) = (2, 5)$, $\|\mathbf{J}^{-1}\| \simeq 2.88$. Le paramètre d'inertie des Algorithmes FISTA est fixé à $b = 4$ [7].

Pour tous les algorithmes proximaux étudiés, une limite sur le nombre maximal d'itérations est fixée à $T_{\max} = 2,5 \cdot 10^5$. De plus, un critère d'arrêt est imposé sur l'écart de dualité normalisé, introduit à la Définition 4.9. Une inspection minutieuse des cartes de régularité et de puissance locales obtenues pour différentes précisions sur l'écart de dualité, nous a en effet permis de conclure que la convergence des algorithmes FISTA et primaux-duaux était atteinte pour des valeurs d'écart de dualité

$$\tilde{\Gamma}_t^\bullet < 5 \cdot 10^{-3}, \quad \text{pour } \bullet \in \{\text{ROF}, \text{L}\}, \quad \text{et} \quad \tilde{\Gamma}_t^C < 10^{-4}. \quad (4.71)$$

4.5.2.4 Mesures de performance

Connaissant la véritable segmentation sous-jacente $\Omega = \Omega_1 \cup \Omega_2$, i.e., celle du masque de la Figure 4.5a, une première mesure de performance naturelle est le *score de segmentation*,

$$\text{Score} = \frac{1}{|\Omega|} (|\Omega_1 \cap \hat{\Omega}_1| + |\Omega_2 \cap \hat{\Omega}_2|) \quad (4.72)$$

comptabilisant la proportion de pixels correctement classés.

Un second indicateur de performance consiste à mesurer *a posteriori* l'écart de régularité locale entre les deux textures. Or, la régularisation utilisée pour obtenir les estimées $\hat{\mathbf{h}}^{\text{ROF}}$, $\hat{\mathbf{h}}^{\text{L}}$ et $\hat{\mathbf{h}}^{\text{C}}$ induit un biais d'estimation, qui se répercute sur les estimées seuillées $S\hat{\mathbf{h}}^{\text{ROF}}$, $S\hat{\mathbf{h}}^{\text{L}}$ et $S\hat{\mathbf{h}}^{\text{C}}$ et donc sur les valeurs moyennes de régularité locale obtenues au moyen de l'Algorithme 11. Ainsi, pour obtenir une estimation plus fiable de la régularité locale de chaque région, une technique d'estimation *globale* de la régularité locale [35, Équation (5)] est mise en œuvre, s'appuyant sur un moyennage préalable des coefficients d'ondelettes dominants au sein de la région considérée. Appliquée région par région, cette procédure fournit les estimées globales *a posteriori* \tilde{H}_1^\bullet et \tilde{H}_2^\bullet , pour chaque méthode $\bullet \in \{\text{ROF}, \text{L}, \text{C}\}$. L'écart de régularité locale *a posteriori*

$$\widehat{\Delta H}^\bullet := \tilde{H}_2^\bullet - \tilde{H}_1^\bullet, \quad \text{pour } \bullet \in \{\text{ROF}, \text{L}, \text{C}\}. \quad (4.73)$$

quantifie alors le degré de précision dans la discrimination des deux textures composant l'image à segmenter.

4.5.2.5 Choix des paramètres de régularisation

Le choix des paramètres de régularisation λ et α , apparaissant dans les Problèmes ROF $_h$ -S (2.123), à contours « libres » (4.5) et à contours « co-localisés » (4.7) détermine complètement les performances de segmentation obtenues. En effet, λ pilote le compromis entre la fidélité au modèle fractal (4.1) et la constance par morceaux requise pour effectuer une segmentation, tandis que α permet de régler le poids relatif de la régularité locale h par rapport à la puissance locale ν dans les pénalisations à contours « libres » et à contours « co-localisés ».

Dans cette étude, nous avons réalisé une recherche sur grille afin de trouver les paramètres de régularisation λ et α maximisant le score de segmentation. En pratique, une gamme logarithmique de paramètres est explorée, allant de 10^{-1} à 10^3 pour λ et de 10^{-2} à 10^3 pour α .

4.5.3 ROF $_h$ -S versus segmentation à contours « libres » ou à contours « co-localisés »

4.5.3.1 Performances

Les performances de segmentation et d'estimations des méthodes ROF $_h$ -S, de segmentation à contours « libres » et de segmentation à contours « co-localisés » sont présentées pour les sept configurations décrites à la Section 4.5.2.1 à la Figure 4.6 et dans le Tableau 4.1. Les performances affichées sont calculées pour le choix d'hyperparamètres (λ, α) maximisant le *score de segmentation*.

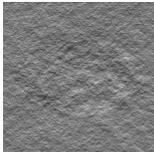
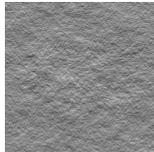
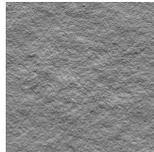
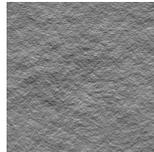
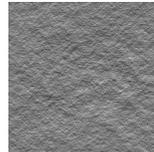
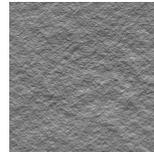
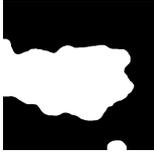
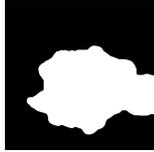
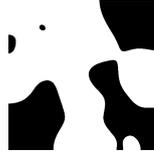
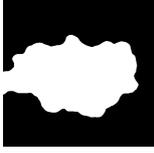
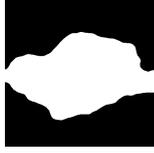
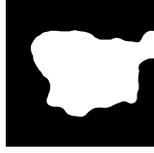
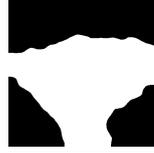
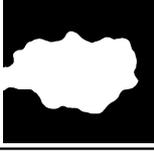
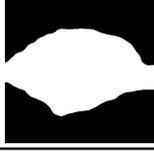
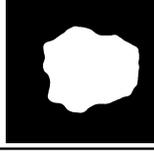
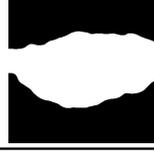
	I	II	III	IV	V	VI
	$\Delta\Sigma^2 = 0,1$ $\Delta H = 0,2$	$\Delta\Sigma^2 = 0,15$ $\Delta H = 0,1$	$\Delta\Sigma^2 = 0,1$ $\Delta H = 0,1$	$\Delta\Sigma^2 = 0,05$ $\Delta H = 0,1$	$\Delta\Sigma^2 = 0,1$ $\Delta H = 0,05$	$\Delta\Sigma^2 = 0,1$ $\Delta H = 0,025$
Texture						
ROF $_h$ -S						
Contours « libres »						
Contours « co-localisés »						

FIGURE 4.6 – **Comparaison des meilleures segmentations.** Les textures fractales par morceaux sont caractérisées par $H_1 = 0,5$, $\Sigma_1^2 = 0,6$ et différents sauts de régularité et de variance locales ΔH , $\Delta\Sigma^2$ suivant la géométrie présentée en Figure 4.5a et les configurations de la Figure 4.5b. Première ligne : segmentation ROF $_h$ -S, $S^{\hat{h}}_{\text{ROF}}$. Deuxième ligne : segmentation à contours « libres », $S^{\hat{h}^L}$. Troisième ligne : segmentation à contours « co-localisés », $S^{\hat{h}^C}$.

Les Configurations I, III, V, VI correspondent à une diminution progressive de l'écart de régularité locale, $\Delta H = H_2 - H_1$ entre les deux textures, pour un écart de variance locale fixé, $\Delta\Sigma^2 = 0,1$. La

		I	II	III	IV	V	VI
Régularité locale	$\Delta H = 0,2$	$\Delta H = 0,1$	$\Delta H = 0,1$	$\Delta H = 0,1$	$\Delta H = 0,1$	$\Delta H = 0,05$	$\Delta H = 0,025$
Variance locale	$\Delta \Sigma^2 = 0,1$	$\Delta \Sigma^2 = 0,15$	$\Delta \Sigma^2 = 0,1$	$\Delta \Sigma^2 = 0,05$	$\Delta \Sigma^2 = 0,1$	$\Delta \Sigma^2 = 0,1$	$\Delta \Sigma^2 = 0,1$
ROF _h -S	Score	86,7 ± 2,1%	79,5 ± 1,2%	78,5 ± 1,1%	77,5 ± 2,9%	69,9 ± 7,1%	59,5 ± 2,4%
	$\widehat{\Delta H}$	0,21 ± 0,07	0,05 ± 0,02	0,05 ± 0,06	0,07 ± 0,04	0,01 ± 0,06	0,05 ± 0,07
Contours « libres »	Score	91,6 ± 1,7%	91,5 ± 2,0%	90,2 ± 1,9%	84,2 ± 4,5%	84,3 ± 3,2%	74,7 ± 8,2%
	$\widehat{\Delta H}$	0,21 ± 0,06	0,07 ± 0,03	0,10 ± 0,02	0,04 ± 0,07	0,05 ± 0,02	0,11 ± 0,28
Contours « co- localisés »	Score	91,7 ± 1,7%	91,9 ± 4,0%	91,1 ± 1,5%	85,5 ± 3,8%	86,1 ± 4,3%	74,3 ± 8,2%
	$\widehat{\Delta H}$	0,20 ± 0,05	0,06 ± 0,04	0,10 ± 0,02	0,08 ± 0,04	0,05 ± 0,02	0,06 ± 0,04

TABLEAU 4.1 – **Meilleures performances de segmentation pour différentes configurations de textures fractales, moyennées sur 5 réalisations.** Les textures fractales par morceaux sont caractérisées par $H_1 = 0,5$, $\Sigma_1^2 = 0,6$ et différents sauts de variance et régularité $\Delta \Sigma^2$, ΔH suivant la géométrie présentée en Figure 4.5a et les configurations de la Figure 4.5b. Première ligne : segmentation ROF_h-S. Deuxième ligne : segmentation à contours « libres ». Troisième ligne : segmentation à contours « co-localisés ».

difficulté de la tâche de segmentation est donc croissante. Tandis que les segmentation obtenues par les trois méthodes, ROF_h-S, à contours « libres » et à contours « co-localisés », sont très similaires pour les configurations *faciles*, les performances de ROF_h-S diminuent drastiquement avec ΔH . Les méthodes à contours « libres » et à contours « co-localisés » au contraire voient leurs performances se maintenir, y compris dans les configuration plus *difficiles*. Dans la Configuration V, présentant un écart de régularité très faible, la méthode à contours « co-localisés » parvient à détacher correctement l'ellipse centrale du fond, ce qui témoigne de sa grande sensibilité. En outre, les performances d'estimation de ΔH sont notablement plus satisfaisantes pour les méthodes à contours « libres » et à contours « co-localisés », que pour la méthode ROF_h-S.

Les Configurations II, III, IV présentent une diminution de l'écart de variance locale, $\Delta \Sigma^2$, entre les deux textures pour un écart de régularité locale fixé $\Delta H = 0,1$, ce qui induit donc une difficulté croissante. Notons que l'écart $\Delta H = 0,1$ est suffisamment faible pour représenter une situation intéressante, où l'on constate que ROF_h-S échoue et que les méthodes plus sophistiquées, à contours « libres » et à contours « co-localisés », sont requises. De manière attendue, ROF_h-S ne prenant en compte que la régularité locale (dont l'estimation n'est pas impactée par le changement de variance [35, 30]), ses performances ne sont pas améliorées par l'augmentation de variance entre les configurations IV et II. Au contraire, les méthodes à contours « libres » et à contours « co-localisés » s'avèrent capables de tirer profit d'un écart de variance $\Delta \Sigma^2$, *via* la prise en compte de la puissance locale, pour fournir des segmentations plus précises. L'amélioration de la qualité de segmentation lorsque l'écart de variance augmente semble plus importante pour la méthode à contours « co-localisés », ce qui est cohérent avec le fait que cette méthode favorise des contours de régularité et de puissance locales co-localisés.

Ces résultats permettent de tirer plusieurs conclusions. D'une part, utiliser l'information auxiliaire apportée par un éventuel écart de variance locale, $\Delta \Sigma^2$, permet d'améliorer les résultats de segmentation et d'estimation, notamment dans le cas où l'écart de régularité locale est faible. Ainsi, les méthodes à contours « libres » et à contours « co-localisés » atteignent toujours des performances de segmentation et d'estimation supérieures à la méthode ROF_h-S ne travaillant que sur la régularité locale. D'autre part, la méthode à contours « co-localisés », qui impose un couplage fort entre la géométrie de la carte de régularité locale \widehat{h}^C et celle de la carte de puissance locale \widehat{v}^C , atteint une plus grande précision, à la fois dans les segmentations obtenues à la Figure 4.6, mais également dans les estimations d'écart de régularité locale du Tableau 4.1.

4.5.3.2 Coûts de calcul

Le Tableau 4.2 compare le coût de calcul de la résolution des problèmes de minimisation associés aux trois méthodes $\text{ROF}_h\text{-S}$, à contours « libres » et à contours « co-localisés », par les deux classes d'algorithmes proximaux *forward-backward* et primal-dual. Les versions standards et accélérées des différents algorithmes sont mises en œuvre pour chaque méthode de segmentation et pour deux configurations différentes, la Configuration I (facile) et la Configuration III (difficile). Les coûts de calcul, mesurés à la fois en terme de nombre d'itérations nécessaires et de temps de calcul sont moyennés sur les cinq réalisations de chaque configuration de textures.

Comme attendu, les stratégies d'accélération s'accompagnent d'une diminution systématique du nombre d'itérations nécessaires, pouvant aller jusqu'à une réduction d'un facteur vingt dans le cas des algorithmes primaux-duaux, ce qui se traduit généralement par un temps de calcul moindre. On note néanmoins que cette diminution du temps de calcul n'est pas toujours observée dans le cas de l'algorithme FISTA dont la stratégie de sur-relaxation augmente le nombre d'opérations élémentaires par itération.

En outre, la méthode $\text{ROF}_h\text{-S}$, ne prenant en compte que la régularité locale, est beaucoup plus économe en coût de calcul, à la fois en terme d'itérations et de temps de calcul, que les méthodes à contours « libres » et à contours « co-localisés » estimant conjointement la régularité et la puissance locale. Pour la méthode $\text{ROF}_h\text{-S}$, l'algorithme FISTA est préféré à l'algorithme primal-dual accéléré. En effet, si les coûts de calcul associés sont identiques pour la Configuration I, FISTA s'avère dix fois plus rapide pour la Configuration III. En revanche, l'algorithme primal-dual accéléré est préféré pour les méthodes à contours « libres » et à contours « co-localisés », qui est systématique plus efficace à la fois en terme d'itérations qu'en terme de temps de calcul. Enfin, la méthode à contours « libres » est trois à quatre fois moins coûteuse que la méthode à contours « co-localisés », à la fois en terme d'itérations et de coût de calcul, ce qui peut s'expliquer par le fait que la pénalisation à contours « libre » est, par construction, moins contraignante que la pénalisation à contours « co-localisés ».

		Configuration I			Configuration III		
		$\text{ROF}_h\text{-S}$	Contours « libres »	Contours « co-localisés »	$\text{ROF}_h\text{-S}$	Contours « libres »	Contours « co-localisés »
Itérations (10^3 it.)	FBD	96 ± 48	> 250	> 250	241 ± 18	> 250	> 250
	FISTA	1,7 ± 0,4	50,2 ± 21,0	231 ± 37	3,7 ± 0,7	48,1 ± 3,4	> 250
	PD	31,8 ± 17,0	> 250	> 250	201 ± 69	> 250	> 250
	PDacc	1,5 ± 0,4	31,4 ± 4,6	125 ± 67	45,2 ± 43	40,5 ± 2,8	121 ± 42
Temps (s)	FBD	1090 ± 520	4840 ± 15	4210 ± 76	2,010 ± 73	4810 ± 215	4200 ± 76
	FISTA	16 ± 4	1030 ± 410	4800 ± 560	30 ± 5	989 ± 64	5110 ± 340
	PD	297 ± 150	4180 ± 69	4110 ± 43	1580 ± 490	4150 ± 18	4100 ± 15
	PDacc	15 ± 4	619 ± 96	2420 ± 1,300	349 ± 330	785 ± 59	2320 ± 790

TABLEAU 4.2 – Nombres d'itérations et temps de calcul nécessaires pour atteindre la condition de convergence de l'Équation (4.71) pour les différents algorithmes présentés et leurs versions standards, sans stratégies d'accélération, illustrés sur deux Configurations : I ($\Delta H = 0,2$, $\Delta \Sigma^2 = 0,1$) et III ($\Delta H = 0,1$, $\Delta \Sigma^2 = 0,1$). **FBD** : Algorithme *forward-backward* sur le dual, **FISTA** : accélération inertielle de l'algorithme *forward-backward*, **PD** : Algorithme primal-dual de Chambolle-Pock, **PDacc** : Algorithme primal-dual accéléré grâce à la forte-convexité de la fonction objectif.

4.5.3.3 Bilan

Les résultats présentés à la Figure 4.6 et au Tableau 4.1 montrent clairement le bénéfice tiré de l'utilisation conjointe de la régularité et de la puissance locale pour la segmentation de textures

fractales, notamment pour distinguer des textures dont les attributs sont proches.

Réaliser une minimisation rapide implique de passer de l'utilisation de l'algorithme de type *forward-backward* FISTA, le plus rapide pour la méthode $\text{ROF}_h\text{-S}$, à un algorithme primal-dual de Chambolle-Pock accéléré pour les méthodes à contours « libres » et à contours « co-localisés ».

Dans les configurations les plus difficiles, nous avons constaté que la segmentation à contours « co-localisés » obtenait des résultats légèrement meilleurs que la méthode à contours « libres », au prix d'une augmentation non négligeable du coût de calcul. En ce sens, la méthode de segmentation à contours « libres » peut représenter un bon compromis entre de bonnes performances de segmentation et un effort de calcul raisonnable.

4.5.4 Segmentation d'assemblages de textures réelles

De premiers tests sur des données réelles sont réalisés en utilisant la base de données de textures fournie par l'Université du Maryland, HighResolution (UMD HR)². Cette base de données contient cinquante classes de textures *homogènes*, chacune composée de quarante photographies d'un même type de texture dans des conditions de prise de vue (exposition, orientation, ...) différentes. À partir de deux textures choisies arbitrairement, par exemples celles des Figures 4.7a et 4.7b, des images texturées *homogènes par morceaux*, comme celle de la Figure 4.7d, sont construites par assemblage selon le modèle du masque elliptique de la Figure 4.7c, utilisé à la section précédente également.

Remarque. Les textures homogènes sont au préalable centrées et normalisées en variance, afin d'éviter que le problème de segmentation de texture ne se réduise à la détection d'un saut de moyenne ou de variance.

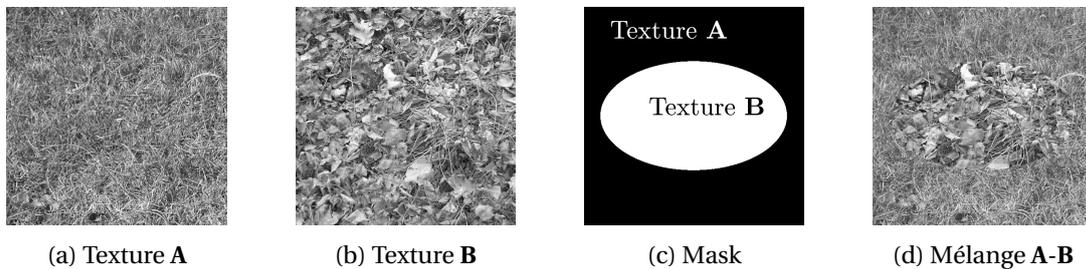


FIGURE 4.7 – **Assemblage de textures réelles.** Deux échantillons de textures réelles ((a) et (b)) sont extraits de la base de données UMD HR et assemblés selon la géométrie du masque (c) pour former une image (d) texturée homogène par morceaux.

Les approches de segmentation $\text{ROF}_h\text{-S}$ (voir Section 2.3.4), à contours « libres » et à contours « co-localisés » (introduites à la Section 4.2.2), sont appliquées à ces images texturées homogènes par morceaux, ainsi que la méthode de Yuan *et coll.* [36] (décrite à la Section 2.3.3.1), reposant sur l'extraction d'histogrammes locaux et un partitionnement *via* une factorisation matricielle.

Les résultats de segmentation obtenus avec chacune des méthodes sont présentés à la Figure 4.8. Les performances de segmentation affichées à la Figure 4.8 sont mesurées en terme de proportion de pixels bien classés. Afin d'ajuster les paramètres de régularisation intervenant dans les méthodes $\text{ROF}_h\text{-S}$, à contours « libres » et à contours « co-localisés », une recherche sur grille est effectuée, et est retenue la segmentation atteignant le meilleure score de segmentation. Pour la méthode de Yuan *et coll.* [36], la taille de la fenêtre sur laquelle sont calculés les histogrammes locaux est ajustée par une recherche exhaustive, pour maximiser le nombre de pixels bien classés.

2. <http://legacydirs.umiaccs.umd.edu/fer/website-texture/texture.htm>

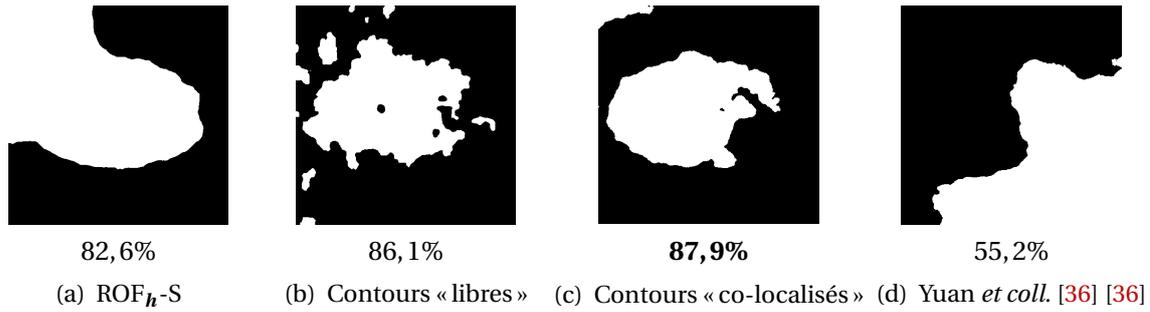


FIGURE 4.8 – **Performances de segmentation sur des textures réelles.** Segmentation de l'image texturée homogène par morceaux de la Figure 4.7d par les méthodes : ROF_h-S (a), à contours « libres » (b), à contours « co-localisés » (c) et de Yuan *et coll.* [36] (d). Les performances sont indiquées en proportion de pixels bien classés.

On constate clairement à la Figure 4.8d que la méthode de Yuan *et coll.* [36] échoue à effectuer cette segmentation, tandis que les trois méthodes reposant sur la régularité locale, ROF_h-S (Figure 4.8a), à contours « libres » (Figure 4.8b) et à contours « co-localisés » (Figure 4.8c), réussissent à saisir, au moins en partie, l'ellipse centrale. Si la méthode ROF_h-S (Figure 4.8a) détecte une partie du contour de l'ellipse, les méthodes à contours « libres » (Figure 4.8b) et à contours « co-localisés » (Figure 4.8c) aboutissent à des segmentations bien plus satisfaisantes. De plus, la méthode à contours « co-localisés », non seulement atteint un score de segmentation légèrement plus élevée que la méthode à contours « libres », mais présente également des contours plus lisses, ce qui, nous le verrons dans l'application de la Section 6.3, peut s'avérer crucial dans les problèmes réels [26]. Des conclusions similaires sont observées sur d'autres échantillons de la base de données UMD HR.

Il faut noter que l'amélioration des performances de segmentation et de la qualité des frontières obtenues est au prix d'un accroissement du temps de calcul. En effet, tandis que la méthode de Yuan *et coll.* [36] fournit en une seconde la segmentation de la Figure 4.8d, les méthodes ROF_h-S, à contours « libres » et à contours « co-localisés » nécessitent respectivement 10 s, 600 s et 1800 s pour les segmentations de la Figure 4.8, ce qui est en accord avec les évaluations du coût de calcul présentées au Tableau 4.2.

4.6 Apprentissage profond pour la segmentation de textures fractales

Le travail présenté dans cette section a fait l'objet d'un article accepté pour la conférence EU-SIPCO 2020 [28].

La Section 2.4 a souligné l'importance de l'apprentissage profond dans les méthodes récentes de traitements d'images [32, 17, 21, 19, 22], et notamment pour la segmentation de texture [10, 1], à la Section 2.4.3. L'objectif de cette Section est d'élaborer, à partir de [1], une procédure de segmentation de textures fractales *supervisée*, afin de la comparer aux méthodes de segmentation à « contours libres » et à « contours co-localisés » construites à la Section 4.2. En effet, la construction à faible coût d'une base de données d'entraînement de grande taille est rendue possible par la procédure de synthèse de textures fractales homogènes par morceaux proposée au Chapitre 3. Nous nous intéresserons non seulement aux performances de segmentation, mais également à leur robustesse, tant en terme de taille de la base d'entraînement, qu'en terme de différence entre les jeux de données d'entraînement et de test. Nous discuterons aussi de manière détaillée la complexité de chacune de ces méthodes de segmentation, et les efforts de calcul qu'elles requièrent.

4.6.1 Mise en œuvre

4.6.1.1 Architectures proposées

Trois réseaux de neurones convolutionnels de profondeurs et de complexités différentes sont construits, sur la base des travaux de Long *et coll.* [21], introduisant le réseau FCN-8s pour la segmentation d'image, puis d'Andrzejczyk *et coll.* [1], adaptant ce réseau à la segmentation de texture. Un premier réseau, dont l'architecture, calquée sur celle du réseau proposé par [1], est composée de cinq blocs de convolution et trois « connexions résiduelles » (*skip connections*), induisant un total de $8 \cdot 10^7$ poids, sert de point de départ. Il représentera dans la suite l'état-de-l'art pour la segmentation supervisée de textures par réseau de neurone convolutionnel. Deux réseaux supplémentaires sont construits, par simplification progressive du réseau à $8 \cdot 10^7$ poids, comportant respectivement $2 \cdot 10^6$ et $4 \cdot 10^5$ poids.

Les trois réseaux que nous allons considérer, à $8 \cdot 10^7$, $2 \cdot 10^6$ et $4 \cdot 10^5$ poids, partagent la même structure globale, dérivant de celle du réseau FCN-8s [21] pour la segmentation, présenté à la Section 2.4.3. Pour décrire cette structure globale, nous nous appuyons sur la Figure 4.9, représentant le réseau le plus simple, à $4 \cdot 10^5$ poids. Chaque réseau est composé d'une succession de blocs de convolution avec activations ReLU et mise en commun réduisant la résolution, qui sont notés BLOCS 1, 2 sur la Figure 4.9, et d'un bloc final de déconvolution, noté BLOC F sur la Figure 4.9, caractérisé par un nombre de filtres plus important et des filtres plus larges, de 5×5 pixels. Pour chaque réseau, le BLOC F contient une étape de décrochage (*dropout*), où le réseau se voit amputé de la moitié de ses poids afin d'éviter les effets de surapprentissage. Plusieurs « connexions résiduelles » fusionnent ensuite les sorties des blocs superficiels et la sortie du bloc final, à l'aide de sur-échantillonnages adéquats. Enfin, une activation de type « maximum doux » (*softmax*), correspond à la décision final de segmentation, assignant un vecteur de probabilités d'appartenance aux différentes classes à chaque pixel. Tous ces réseaux partagent la même taille de filtre par défaut, 3×3 pixels, un même « décalage » (*stride*), $s = 1$, et des étapes de « mise en commun » par *Max pooling* sur des voisinages de 2×2 pixels.

La différence entre les réseaux à $8 \cdot 10^7$, $2 \cdot 10^6$ et $4 \cdot 10^5$ poids réside dans leur complexité, et ils seront donc désignés dans la suite par leur nombre de poids, afin de souligner la hiérarchie de complexités entre eux. De la sorte, nous comparerons les réseaux suivants :

- Le réseau très profond, constituant l'état-de-l'art [1], à $8 \cdot 10^7$ poids, dont le premier bloc de convolution comporte deux couches, de 64 filtres chacune, le second bloc deux couches également, de 128 filtres chacune, le troisième bloc trois couches, de 256 filtres chacune, le quatrième bloc trois couches, de 512 filtres chacune, enfin, le bloc final est composé d'une couche « complètement connectée » à 4096 filtres. Trois « connexions résiduelles » (*skip connections*) relient successivement les troisième, deuxième puis premier blocs à la sortie du réseau, au moyen de couches de déconvolutions adéquates, de chacun k filtres.
- Un réseau moyennement profond à $2 \cdot 10^6$ poids, dont le premier bloc de convolution est composé de deux couches, de 32 filtres chacune, le second bloc de deux couches également, de 64 filtres chacune, et le troisième bloc de convolution comporte trois couches, de 128 filtres chacune. Le bloc final est quant à lui composé de couches contenant 512 filtres. Deux « connexions résiduelles » (*skip connections*) relient successivement le deuxième puis le premier bloc à la sortie du réseau, au moyen de couches de déconvolutions adéquates, de chacun k filtres.
- Un réseau faiblement profond à $4 \cdot 10^5$ poids, dont le premier bloc est composé de deux couches de convolution, de 32 filtres chacune et le second de deux couches de convolution également, de 64 filtres chacune. Le bloc final contient une couche de convolution de 256 filtres, et est relié à la sortie du premier bloc *via* une « connexion résiduelle » (*skip connection*), réalisant une déconvolution à k filtres.

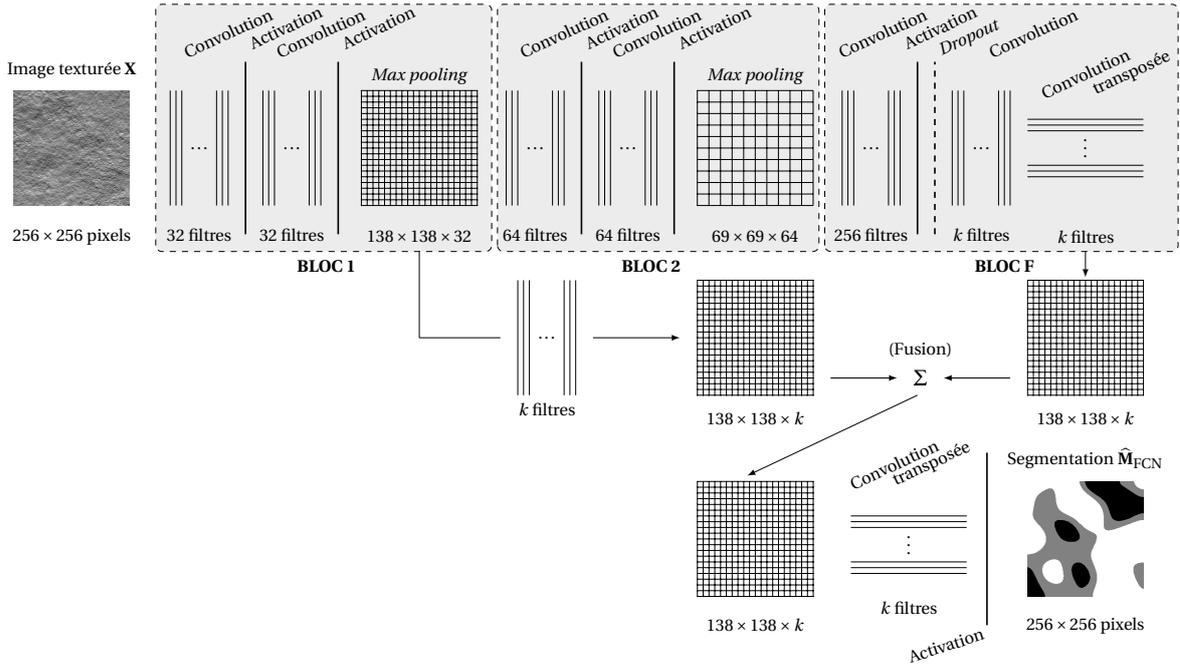


FIGURE 4.9 – Réseau de neurones convolutionnel pour la segmentation d'image. Schéma d'un réseau de neurones convolutionnel avec des connexions entre blocs de convolution (architecture la plus simple, contenant $4 \cdot 10^5$ poids).

4.6.1.2 Problèmes considérés et bases de données d'entraînement

Nous allons étudier la segmentation de textures fractales à $k = 2, 3$ et 4 classes, dans deux « configurations » distinctes, soit au total *six problèmes*, qui sont résumés à la Figure 4.10. Les images considérées sont supposées suivre le modèle de champ gaussien auto-similaire par morceaux décrit à la Section 3.3, c'est-à-dire qu'une image X est composée de k textures, caractérisées par leurs régularité et variance locales $(H_i, \Sigma_i^2)_{i=1}^k$, assemblées selon une partition $\Omega = \sqcup_{i=1}^k \Omega_i$. Chaque point du plan (H, Σ^2) représenté à la Figure 4.10, correspond à un couple de régularité et de variance locales caractérisant une texture. Les k points de couleur bleu sur la Figure 4.10 représentent les caractéristiques des k textures à distinguer dans la Configuration I, tandis que les points de couleur rouge correspondent aux caractéristiques des k textures à segmenter dans la Configuration II. Ces deux jeux d'attributs permettent d'étudier les performances de segmentation dans deux situations différentes. En effet, comme on peut le remarquer sur la Figure 4.10, dans la Configuration I, les textures à segmenter présente des écarts de régularité locale, $\Delta H \equiv H_i - H_{i'}$, grands, et de petits écarts de variance locale, $\Delta \Sigma^2 \equiv \Sigma_i^2 - \Sigma_{i'}^2$. Inversement, la Configuration II correspond à de faibles écarts de régularité locale et de grands écarts de variance locale.

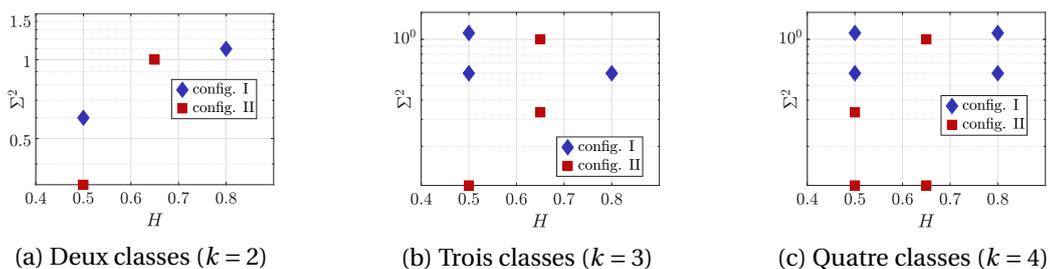


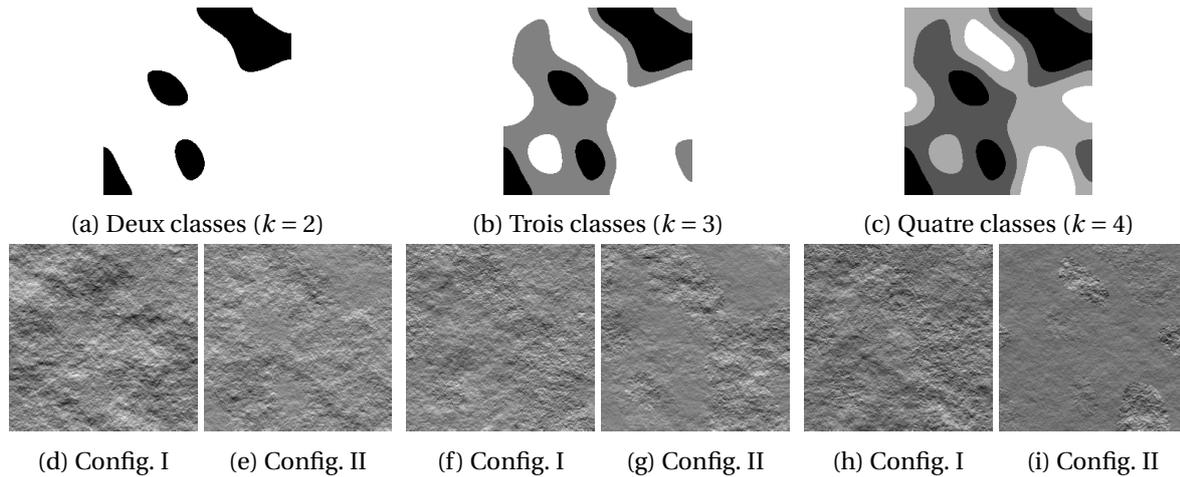
FIGURE 4.10 – Configurations utilisées pour la synthèse de textures caractérisées par leur régularité et leur variance locales $\{(H_i, \Sigma_i^2), i = 1, \dots, k\}$.

Afin d'entraîner les réseaux de neurones décrit ci-dessus, six bases d'entraînement

$$\{(X^{(p)}, M^{(p)}) \in \mathbb{R}^{N_1 \times N_2} \times \{1, \dots, k\}^{N_1 \times N_2}, \quad p = 1, \dots, P\}, \quad (4.74)$$

sont construites, où chaque image $X^{(p)}$ contient k textures fractales différentes, caractérisées par le jeu de régularités et variances locales $(H_i, \Sigma_i^2)_{i=1}^k$ de la Configuration I (en bleu sur la Figure 4.10) ou de la Configuration II (en rouge sur la Figure 4.10), assemblées selon une partition $\Omega^{(p)} = \sqcup_{i=1}^k \Omega_i^{(p)}$, choisie *aléatoirement*, encodée sous forme d'une carte d'étiquettes, telle que $M^{(p)}(\underline{n}) = i$ équivaut à $\underline{n} \in \Omega_i^{(p)}$.

Remarque (Partition aléatoire). Pour construire et d'entraîner un réseau pour une tâche de segmentation de *texture* il est nécessaire de disposer d'une base de données d'entraînement ne contenant *que* des informations de texture, en évitant le plus possible que les images d'entraînement partagent des formes spécifiques. En effet, si ce n'est pas le cas, la décision de segmentation encodée par le réseau pourrait prendre en compte des éléments de structure géométrique qui n'ont aucune raison, dans notre étude, d'être présents dans la base de données de test ou dans les données réelles à étudier. Les segmentations sous-jacentes, aléatoires, comme celles présentées aux Figures 4.11a, 4.11b et 4.11c, ont été obtenues en réalisant un débruitage par Variation Totale sur un bruit blanc gaussien de $N_1 \times N_2$ pixels et en appliquant ensuite un seuillage à $k = 2, 3$ ou 4 classes.



4.6.1.3 Entraînement

À partir de la base d'entraînement $\{(X^{(p)}, M^{(p)}), p = 1, \dots, P\}$, les poids des différents réseaux sont appris au moyen d'un algorithme de descente stochastique avec rétropropagation des gradients, pour chaque problème de segmentation à $k = 2, 3$ et 4 classes, et dans chaque Configuration I et II. Pour le problème de segmentation à $k = 2$ classes, la fonction de perte considérée est l'entropie croisée *binnaire* de l'Équation (2.128), tandis que dans les cas $k = 3, 4$ il s'agit de l'entropie croisée *catégorielle* de l'Équation (2.140).

En pratique, la minimisation de la fonction de perte globale est effectuée grâce à l'algorithme ADAM, stabilisé par une stratégie AMSGrad [31]. Le taux d'apprentissage (*learning rate*) est fixé à $2 \cdot 10^{-4}$. Enfin, pour l'implémentation de la descente de gradient *stochastique*, la base d'apprentissage est découpée en paquets (*batches*) de 20 images chacun. À chaque itération, ou *epoch*, de l'algorithme de gradient stochastique, toutes les images du jeu d'entraînement sont lues une et une seule fois.

Afin d'illustrer la convergence de l'algorithme ADAM dans la phase d'apprentissage, nous proposons de suivre l'évolution de la « précision » (*accuracy*), en terme de proportion de pixels bien

classés, sur une base d'images de test $\{(X^{(q)}, M^{(q)}), q = 1, \dots, Q\}$ en fonction, d'une part du nombre d'*epochs*, à la Figure 4.12a, d'autre part en fonction du nombre d'images traitées, à la Figure 4.12b, égal au nombre d'*epochs* multiplié par la taille de la base d'entraînement. Dans le cas de la segmentation à deux classes, dans la Configuration I, l'évolution de la précision pendant la phase d'entraînement de chacun des trois réseaux est tracée en fonction du nombre d'*epochs* pour deux bases d'entraînement de $P = 2000$ et $P = 20$ images. Pour cela, à intervalle régulier, le réseau est gelé, et la précision globale est calculée sur le jeu de données de test.

En pratique, la phase d'entraînement des réseaux de neurones a été réalisée sur les machines du Centre Blaise Pascal³, gérées par le système SIDUS (Single Instance Distributing Universal System) élaboré par Emmanuel Quemener, en utilisant les ressources de calcul sur GPU mises à notre disposition, ce qui a permis d'entraîner des réseaux profonds sur des bases de données conséquentes en quelques heures.

La Figure 4.12a montre clairement que, durant la phase d'entraînement sur $P = 2000$ (resp. $P = 20$) images, les trois réseaux de neurones convergent, et atteignent un maximum stable de la précision, après respectivement 30 et 3000 *epochs*. Des courbes équivalentes sont obtenues pour les autres problèmes de segmentation à $k = 3$ ou 4 classes, et dans la Configuration II, validant l'approche utilisée.

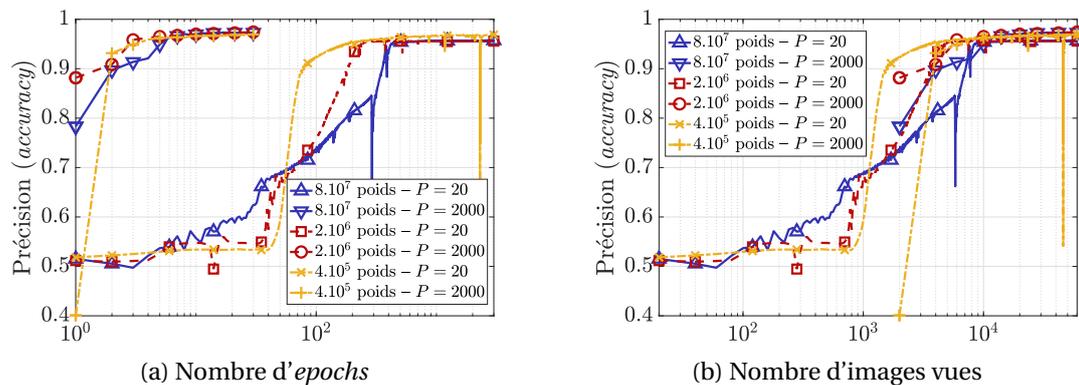


FIGURE 4.12 – Évolution du score de segmentation des trois réseaux au cours de l'entraînement sur la Config. I, avec deux classes $k = 2$.

4.6.2 Segmentation de textures synthétiques à contours aléatoires

4.6.2.1 Jeux de données tests et procédures de comparaison

Afin d'évaluer les performances de segmentation des trois réseaux de neurones, et de les comparer avec les performances de la méthode de segmentation à contours « libres », introduite à l'Équation 4.5, pour chaque problème de segmentation à $k = 2, 3$ ou 4 classes, et pour chaque Configuration I ou II, une base de données de test de 100 images, accompagnées des segmentations sous-jacentes associées, est générée sur le même modèle que la base d'entraînement décrite à la Section 4.6.1.2.

Pour la méthode à contours « libres », nous allons interpréter le réglage des paramètres de régularisation λ et α comme une phase « d'entraînement ». Pour une comparaison honnête avec les réseaux de neurones, ce réglage fin ne doit pas être réalisé pour chaque image à segmenter. La méthode de génération de contours aléatoires décrite à la Section 4.6.1.2 produisant des régions dont les tailles respectives sont équilibrées, nous allons considérer que, pour k fixé, et pour une configuration donnée, le couple (λ, α) optimal est approximativement le même. Par conséquent,

3. <http://www.cbp.ens-lyon.fr/>

le réglage de λ et α est effectué en explorant une grille de 20×20 paramètres sur une *unique* image de la base d'entraînement, pour chaque problème de segmentation à $k = 2, 3$ ou 4 classes, et pour chaque Configuration I ou II, et en choisissant le couple de paramètre maximisant le score de segmentation. La méthode de segmentation à contours « libres » est ensuite appliquées aux 100 images du jeu de test avec ce réglage *fixé* de paramètres de régularisation

4.6.2.2 Performances

Les performances de segmentation, mesurées une fois la convergence atteinte, en terme de proportion de pixels bien classés sur la base de tests de 100 images texturées à contours aléatoires décrite ci-dessus, sont reportées dans le Tableau 4.3. Pour une segmentation à $k = 2$ classes le Tableau 4.3 montre que les réseaux de neurones et la méthode de segmentation à contours « libres » atteignent tous deux des scores de segmentation très élevés, à la fois dans la Configuration I et dans la Configuration II. Pour $k = 3$ classes, les performances de segmentation des réseaux de neurones se maintiennent pour les deux configurations, tandis que la méthode à contours « libres » n'atteint un score de segmentation satisfaisant que dans la Configuration II. Enfin, pour le problème à $k = 4$ classes, les réseaux de neurones accusent une légère diminution du score de segmentation, tandis que les performances de la méthode à contours « libres » sont significativement dégradées. De manière surprenante, les trois réseaux de neurones de complexités décroissantes, à la fois en terme de profondeur (nombre de couches) et de largeur (nombre de filtres par couche), obtiennent des scores de segmentation similaires, malgré une division par un facteur 200 du nombre de poids entre le réseau de l'état-de-l'art et le réseau le plus simple que nous proposons, illustré à la Figure 4.9.

	2 classes	3 classes	4 classes
Entraîné sur la Config. I, testé sur la Config. I			
Segmentation à contours « libres »	93,2 ± 0,8%	69,3 ± 2,8%	58,6 ± 1,5%
Entraîné sur 2000 images			
Réseau à $8 \cdot 10^7$ poids / P = 2000	97,3 ± 0,6%	97,8 ± 0,3%	97,1 ± 0,4%
Réseau à $2 \cdot 10^6$ poids / P = 2000	97,4 ± 0,6%	98,1 ± 0,3%	96,8 ± 0,5%
Réseau à $4 \cdot 10^5$ poids / P = 2000	96,9 ± 0,7%	98,0 ± 0,3%	96,5 ± 0,5%
Entraîné sur 20 images			
Réseau à $8 \cdot 10^7$ poids / P = 20	95,5 ± 0,9%	97,5 ± 0,4%	95,4 ± 0,8%
Réseau à $2 \cdot 10^6$ poids / P = 20	95,4 ± 1,1%	97,4 ± 0,5%	95,9 ± 0,7%
Réseau à $4 \cdot 10^5$ poids / P = 20	96,6 ± 0,7%	98,0 ± 0,4%	96,5 ± 0,5%
Entraîné sur la Config. II, testé sur la Config. II			
Segmentation à contours « libres »	97,8 ± 0,2%	95,2 ± 3,1%	64,9 ± 1,4%
Entraîné sur 2000 images			
Réseau à $8 \cdot 10^7$ poids / P = 2000	99,1 ± 0,2%	98,3 ± 0,3%	95,7 ± 0,5%
Réseau à $2 \cdot 10^6$ poids / P = 2000	99,0 ± 0,2%	98,5 ± 0,3%	95,6 ± 0,5%
Réseau à $4 \cdot 10^5$ poids / P = 2000	99,1 ± 0,2%	98,4 ± 0,3%	95,2 ± 0,6%
Entraîné sur 20 images			
Réseau à $8 \cdot 10^7$ poids / P = 20	98,8 ± 0,2%	97,9 ± 0,3%	94,5 ± 0,7%
Réseau à $2 \cdot 10^6$ poids / P = 20	98,6 ± 0,3%	97,4 ± 0,4%	93,0 ± 0,9%
Réseau à $4 \cdot 10^5$ poids / P = 20	98,8 ± 0,3%	98,3 ± 0,3%	94,8 ± 0,6%

TABLEAU 4.3 – **Performances** Score de segmentation (% de pixels bien classés).

4.6.2.3 Robustesse

Robustesse vis-à-vis de la taille de la base d'entraînement. Afin d'étudier l'impact du nombre d'images d'entraînement, la phase d'apprentissage est effectuée une seconde fois pour chaque réseau, chaque problème à $k = 2, 3$ ou 4 classes et chaque Configuration I ou II, en utilisant uniquement $P = 20$ images d'entraînement. La première conséquence de cette réduction drastique de la base d'entraînement est l'augmentation du nombre d'*epochs* nécessaires pour atteindre la convergence, comme le montre la Figure 4.12a. Comme le montre la Figure 4.12b, le paramètre pertinent par rapport auquel mesurer la convergence n'est pas le nombre d'*epochs*, mais plutôt le nombre d'images prises en compte pour l'ajustement des poids.

En outre, le Tableau 4.3 ne montre qu'une très légère dégradation des performances lorsque le jeu de données de taille plus faible est utilisé. Ce point est particulièrement intéressant puisqu'une réduction de la taille de la base d'entraînement se traduit par des coûts de mémoire bien moindres. Une interprétation possible de cette robustesse vis-à-vis d'une amputation de la base d'entraînement, est que chaque image comportant 256×256 pixels contient en réalité de nombreux « patches » représentatifs des textures étudiées, et donc autant d'exemples de la structure à apprendre. Ainsi 20 images contiendraient suffisamment d'information pour effectuer l'apprentissage des réseaux, et notamment du réseau le plus complexe à $8 \cdot 10^7$ poids.

Robustesse vis-à-vis de la configuration des données d'entraînement. En pratique, la construction d'une base de données d'entraînement repose sur un compromis entre le fait de sélectionner *peu* de données *exactement* représentatives des données à analyser ou de sélectionner *beaucoup* de données *approximativement* représentatives des données à traiter. Pour modéliser une possible inadéquation entre les bases de données d'entraînement et de test, les trois réseaux sont entraînés sur une configuration donnée et testés sur l'autre configuration. De même, les paramètres de régularisation de la méthode à contours « libres » sont ajustés sur une image d'une première configuration et appliqués aux images de la seconde configuration.

Les résultats obtenus pour chaque méthode sont rapportés au Tableau 4.3. Lorsqu'ils sont entraînés dans la Configuration I (grand ΔH , faible $\Delta \Sigma^2$), et testés sur la Configuration II (faible ΔH , grand $\Delta \Sigma^2$), les réseaux de neurones réalisent des segmentations à $k = 2$ classes plus satisfaisantes que celles obtenus par la méthode à contours « libres » aux paramètres ajustés à la Configuration I. Néanmoins, on note que lorsque le nombre de classes k augmente les scores de segmentation atteints par réseaux de neurones se dégradent fortement, ce qui n'est pas le cas pour la méthode à contours « libres ».

Pour un ajustement des poids et des paramètres de régularisation dans Configuration II, puis une analyse d'image correspondant à la Configuration I, la méthode à contours « libres » s'avère bien plus robuste que les réseaux de neurones pour réaliser la segmentation à $k = 2$ classes. On remarque de plus qu'aucune procédure ne parvient à réaliser correctement la segmentation à $k = 4$ classes.

En conclusion, ni les réseaux de neurones, ni la méthode à contours « libres » ne montre de robustesse systématique par rapport aux données d'entraînement. Notons également que les réseaux les plus complexes ne sont pas plus robustes que les réseaux de moindre taille.

4.6.2.4 Estimation d'interfaces

Un point notable, souligné par [1], est que, bien que les scores de segmentation obtenus soient très élevés, les réseaux de neurones tendent à produire des segmentations aux contours très irréguliers. Cet effet est illustré à la Figure 4.13. Or, certaines applications, comme par exemple l'étude des interfaces liquide-gaz [26], nécessitent une mesure précise de la longueur des contours et s'ac-

	2 classes	3 classes	4 classes
Entraîné sur la Config. I, testé sur la Config. II			
Segmentation à contours « libres »	79,2 ± 2,9%	95,2 ± 1,2%	66,3 ± 1,1%
Entraîné sur 2000 images			
Réseau à $8 \cdot 10^7$ poids / P = 2000	91,2 ± 2,1%	65,7 ± 7,2%	55,6 ± 3,4%
Réseau à $2 \cdot 10^6$ poids / P = 2000	87,9 ± 2,5%	69,0 ± 7,6%	50,8 ± 4,0%
Réseau à $4 \cdot 10^5$ poids / P = 2000	81,8 ± 3,8%	65,2 ± 7,2%	46,4 ± 3,7%
Entraîné sur 20 images			
Réseau à $8 \cdot 10^7$ poids / P = 20	91,4 ± 1,6%	63,3 ± 7,1%	54,7 ± 3,3%
Réseau à $2 \cdot 10^6$ poids / P = 20	92,4 ± 1,6%	65,6 ± 7,4%	44,4 ± 3,4%
Réseau à $4 \cdot 10^5$ poids / P = 20	86,3 ± 2,6%	64,9 ± 7,2%	48,4 ± 3,8%
Entraîné sur la Config. II, testé sur la Config. I			
Segmentation à contours « libres »	90,9 ± 2,8%	66,7 ± 2,5%	52,0 ± 1,5%
Entraîné sur 2000 images			
Réseau à $8 \cdot 10^7$ poids / P = 2000	56,2 ± 13,5%	73,5 ± 8,2%	50,9 ± 3,9%
Réseau à $2 \cdot 10^6$ poids / P = 2000	55,1 ± 14,0%	74,9 ± 8,2%	51,3 ± 4,3%
Réseau à $5 \cdot 10^5$ poids / P = 2000	55,5 ± 13,8%	72,6 ± 8,1%	50,2 ± 3,8%
Entraîné sur 20 images			
Réseau à $8 \cdot 10^7$ poids / P = 20	57,1 ± 13,3%	71,1 ± 8,2%	52,6 ± 3,8%
Réseau à $2 \cdot 10^6$ poids / P = 20	55,3 ± 14,0%	71,7 ± 8,4%	49,6 ± 4,2%
Réseau à $5 \cdot 10^5$ poids / P = 20	62,3 ± 11,5%	71,0 ± 8,2%	54,1 ± 3,7%

TABLEAU 4.4 – **Robustesse** Score de segmentation (% de pixels bien classés).

commodent mieux de contours moins bien localisés mais plus lisses que des contours précis mais accidentés tels que ceux de la Figure 4.13c.

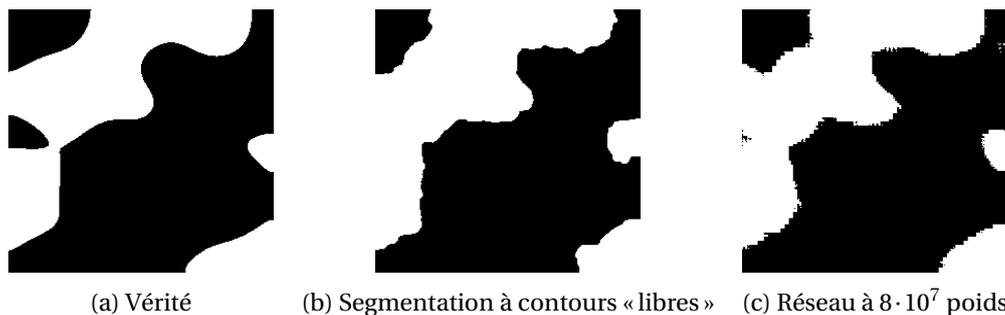


FIGURE 4.13 – **Segmentations à $k = 2$ classes** (entraîné et testé sur la Config. I), montrant des contours irréguliers sur les segmentations obtenues par les réseaux de neurones.

Dans le cas de la segmentation à $k = 2$ classes, la longueur d'interface est mesurée comme le nombre de pixels à la frontière entre les deux régions. La Tableau 4.5 compare l'erreur relative par rapport à la longueur d'interface totale pour les segmentations obtenues par la méthode à contours « libres » et par les trois réseaux de neurones et montre que la méthode de segmentation à contours « libres » estime systématiquement de manière beaucoup plus fiable la longueur d'interface. De plus, on note que lorsque les réseaux de neurones sont entraînés sur des données différentes des données de test, l'estimation de périmètre devient particulièrement mauvaise, ce qui n'est pas le cas pour la méthode à contours « libres » qui s'avère sur ce point bien plus robuste.

Train/Test	Config. I/I	Config. II/II	Config. I/II	Config. II/I
Segmentation à contours « libres »	14 ± 4	13 ± 2	48 ± 22	16 ± 4
Réseau à $8 \cdot 10^7$ poids	33 ± 3	36 ± 2	112 ± 24	68 ± 18
Réseau à $2 \cdot 10^6$ poids	37 ± 3	41 ± 3	213 ± 72	77 ± 4
Réseau à $4 \cdot 10^5$ poids	24 ± 3	18 ± 2	337 ± 131	62 ± 9

TABLEAU 4.5 – **Erreur relative sur la longueur d’interface pour la segmentation à $k = 2$ classes.** Les réseaux de neurones convolutionnels sont entraînés sur $P = 2000$ images. (Des résultats similaires sont obtenus pour un entraînement sur $P = 20$ images.)

4.6.2.5 Complexité et effort de calcul

Plusieurs raisons, liées à la fois aux procédures mise en œuvres et aux machines utilisées rendent impossible la comparaison exacte du temps de calcul nécessaire à la segmentation par réseaux de neurones et par la méthode à contours « libres ». En effet, intrinsèquement, tout l’effort de calcul requis pour les méthodes de segmentation par réseaux de neurones sont concentrées dans la phase d’entraînement. Une fois cette étape réalisée il est possible de traiter extrêmement rapidement des volumes de données conséquents. La situation est très différentes dans les méthodes non supervisées, en général bien moins coûteuses que la phase d’entraînement, mais à répéter pour chaque image à segmenter. En outre, l’entraînement des réseaux de neurones est effectué par calcul sur GPU, grâce aux infrastructures fournies par Le Centre Blaise Pascal de l’ENS de Lyon, tandis que les méthodes non supervisées (notamment à contours « libres ») sont implémentées en Matlab et mises en œuvre *via* du calcul CPU. Nous allons néanmoins proposer une heuristique de comparaison, afin de fixer quelques ordres de grandeur.

Pour chaque réseau de neurones, \mathcal{W} poids sont appris, *via* la prise en compte de P images d’entraînement segmentées par l’algorithme ADAM réalisant \mathcal{J} *epochs*, nous allons donc considérer que le *coût* induit s’écrit $\mathcal{C} = \mathcal{W} \times P \times \mathcal{J}$. Pour la méthode à contours « libres », $\mathcal{W} = 2$ hyperparamètres doivent être ajustés par l’étude de $P = 1$ image segmentée, pour laquelle l’Algorithme 9 est appelé $20 \times 20 = 400$ fois, pour 400 couples (λ, α) de paramètres de régularisation. Chaque appel de l’Algorithme 9 nécessite au plus $T_{\max} = 2.5 \cdot 10^4$ itérations, ce qui conduit à $\mathcal{J} = 10^7$. Le coût de calcul \mathcal{C} ainsi calculé est présenté au Tableau 4.6, ce qui permet d’illustrer la quantité bien moindre d’efforts de calcul nécessaire pour la mise en œuvre de la méthode non supervisée à contours « libres », par rapport à l’apprentissage supervisé de réseaux de neurones.

	\mathcal{W}	P	\mathcal{J}	\mathcal{C}
Segmentation à contours « libres »	2	1	10^7	$2 \cdot 10^7$
Réseau à $8 \cdot 10^7$ poids / $P = \{20, 2000\}$	$8 \cdot 10^7$	$\{20, 2000\}$	$\{3000, 30\}$	$4,8 \cdot 10^{12}$
Réseau à $2 \cdot 10^6$ poids / $P = \{20, 2000\}$	$2 \cdot 10^6$	$\{20, 2000\}$	$\{3000, 30\}$	$1,2 \cdot 10^{11}$
Réseau à $4 \cdot 10^5$ poids / $P = \{20, 2000\}$	$4 \cdot 10^5$	$\{20, 2000\}$	$\{3000, 30\}$	$2,4 \cdot 10^{10}$

TABLEAU 4.6 – **Coût de calcul \mathcal{C} pour la phase d’entraînement.**

4.6.3 Bilan de l’apprentissage supervisé pour la segmentation de textures fractales

Dans des conditions très favorables, où l’on dispose d’une base de données *exactement* représentative des images à analyser, nous avons pu observer la supériorité des approches de segmentation supervisées par réseau de neurones sur les méthodes non supervisée, telle que la méthode à contours « libres », au prix néanmoins d’une augmentation significative des ressources de calcul et de mémoire nécessaires.

Cependant, dans des conditions détériorées où la base d'entraînement est constituées de textures fractales dont les paramètres caractéristiques sont différents de ceux des images de test, les performances de segmentation des réseaux de neurones se dégradent, parfois fortement, y compris pour les réseaux les plus complexes. De plus, nous avons pu constater leur faible capacité à produire des contours réguliers, excluant ainsi toute mesure fiable de longueur d'interface. Il semble donc difficile d'envisager leur application directe pour l'étude de textures réelles.

De récents travaux, comme par exemple l'algorithme LISTA proposé par [18], se sont intéressés à des approches hybrides entre l'apprentissage profond et les méthodes de régularisation par algorithmes proximaux permettant de tirer le meilleur parti des avantages respectifs de ces deux approches et pourraient faire l'objet de futures extensions de ce travail.

Références

- [1] V. Andrearczyk and P. Whelan. Texture segmentation with fully convolutional networks. *preprint arXiv:1703.05230*, 2017. 87, 113, 114, 119
- [2] J.-F. Aujol and C. Dossal. Stability of over-relaxations for the forward-backward algorithm, application to FISTA. *SIAM J. Optim.*, 25(4) :2408–2433, 2015. 92
- [3] H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*, volume 408. Springer, 2011. 90, 91, 92, 93, 104, 105
- [4] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1) :183–202, 2009. 92
- [5] S. Boyd, S. P. Boyd, and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004. 92, 93
- [6] X. Cai and G. Steidl. Multiclass segmentation by iterated rof thresholding. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 237–250, Lund, Sweden, 2013. Springer. 106
- [7] A. Chambolle and C. Dossal. On the convergence of the iterates of "FISTA". *J. Optim. Theory Appl.*, 166(3) :25, Aug. 2015. URL <https://hal.inria.fr/hal-01060130>. 100, 108
- [8] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Image Vis. Comput.*, 40(1) :120–145, 2011. 92, 94, 96, 101, 102
- [9] C. Chau, P. Combettes, and V. Pesquet, J.-C. Wajs. A variational formulation for frame-based inverse problems. *Inverse Problems*, 23(4) :1495–1518, Jun. 2007. 91
- [10] M. Cimpoi, S. Maji, I. Kokkinos, and A. Vedaldi. Deep filter banks for texture recognition, description, and segmentation. *Int. J. Comp. Vis.*, 118(1) :65–94, 2016. 113
- [11] P. L. Combettes. Solving monotone inclusions via compositions of nonexpansive averaged operators. *Optimization*, 53(5-6) :475–504, 2004. 92
- [12] P. L. Combettes and J.-C. Pesquet. A douglas-rachford splitting approach to nonsmooth convex variational signal recovery. 1(4) :564, 2007. 90
- [13] P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In H. H. Bauschke, R. S. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz, editors, *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer-Verlag, New York, 2011. 90

- [14] P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Model. & Simul.*, 4(4) :1168–1200, 2005. 92
- [15] P. L. Combettes, Đ. Dũng, and B. C. Vũ. Dualization of signal recovery problems. *Set-Valued and Variational Analysis*, 18(3-4) :373–404, 2010. 91
- [16] C.-A. Deledalle, N. Papadakis, J. Salmon, and S. Vaïter. CLEAR : Covariant LEAst-Square Refitting with applications to image restoration. *SIAM J. Imaging Sci.*, 10(1) :243–284, 2017. 86
- [17] J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Y. Jia, and E. Shelhamer. Caffe : Convolutional architecture for fast feature embedding. In *Proc. IEEE Int. Conf. Comput. Vis. and Patt. Rec.*, Columbus, Ohio, USA, 2014. Citeseer. 113
- [18] K. Gregor and Y. LeCun. Learning fast approximations of sparse coding. In *International Conference on Machine Learning*, pages 399–406, Haifa, Israel, 2010. 122
- [19] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proc. IEEE Int. Conf. Comput. Vis. and Patt. Rec.*, pages 447–456, Boston, Massachusetts, USA, 2015. 113
- [20] S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Stat.*, 22(1) :79–86, 1951. 94
- [21] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proc. IEEE Int. Conf. Comput. Vis. and Patt. Rec.*, pages 3431–3440, Boston, Massachusetts, 2015. 113, 114
- [22] P. Luc, C. Couprie, S. Chintala, and J. Verbeek. Semantic segmentation using adversarial networks. *NIPS Workshop on Adversarial training*, 2016. 113
- [23] S. Mallat. *A Wavelet Tour of Signal Processing, Third Edition : The Sparse Way*. Academic Press, Inc., Orlando, FL, USA, 3rd edition, 2008. ISBN 0123743702, 9780123743701. 107
- [24] J.-J. Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société mathématique de France*, 93 :273–299, 1965. 90
- [25] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3) : 127–239, 2014. 90
- [26] B. Pascal, N. Pustelnik, P. Abry, M. Serres, and V. Vidal. Joint estimation of local variance and local regularity for texture segmentation. application to multiphase flow characterization. In *Proc. IEEE Int. Conf. Image Proc. (ICIP)*, pages 2092–2096, Athens, Greece, 2018. IEEE. 86, 87, 113, 119
- [27] B. Pascal, N. Pustelnik, and P. Abry. Nonsmooth convex joint estimation of local regularity and local variance for fractal texture segmentation. *preprint arXiv :1910.05246*, 2019. 86, 87
- [28] B. Pascal, V. Mauduit, P. Abry, and N. Pustelnik. Scale-free Texture Segmentation : Expert Feature-based versus Deep Learning strategies. 2021. 86, 113
- [29] N. Pustelnik, C. Chaux, and J.-C. Pesquet. Parallel ProXimal Algorithm for image restoration using hybrid regularization. *IEEE Trans. Image Process.*, 20(9) :2450–2462, 2011. 91
- [30] N. Pustelnik, H. Wendt, P. Abry, and N. Dobigeon. Combining local regularity estimation and total variation optimization for scale-free texture segmentation. *IEEE Trans. Computational Imaging*, 2(4) :468–479, 2016. URL <http://ieeexplore.ieee.org/abstract/document/7523285/>. 110

- [31] S. J. Reddi, S. Kale, and S. Kumar. On the convergence of adam and beyond. *preprint arXiv :1904.09237*, 2019. 116
- [32] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat : Integrated recognition, localization and detection using convolutional networks. *ICLR*, 2014. 113
- [33] D. Serre. Matrices : Theory and applications. *Graduate texts in mathematics*, 2000. 100, 103
- [34] H. Wendt, P. Abry, and S. Jaffard. Bootstrap for Empirical Multifractal Analysis. *IEEE Signal Process. Mag.*, 24(4) :38–48, 2007. 108
- [35] H. Wendt, S. G. Roux, P. Abry, and S. Jaffard. Wavelet leaders and bootstrap for multifractal analysis of images. *Signal Process.*, 89(6) :1100–1114, 2009. 108, 110
- [36] J. Yuan, D. Wang, and A. M. Cheryadat. Factorization-based texture segmentation. *IEEE Trans. Image Process.*, 24(11) :3488–3497, Nov. 2015. ISSN 1057-7149, 1941-0042. doi : 10.1109/TIP.2015.2446948. URL <http://ieeexplore.ieee.org/document/7127013/>. 106, 112, 113

Chapitre 5

Stratégies de type Stein pour le réglage des hyperparamètres

« Les études scolaires sont sans doute un chemin vers la sainteté aussi bon que tout autre. [...] Celui qui traverse les années d'étude sans développer en soi cette attention a perdu un grand trésor. »

S. Weil, *Attente de Dieu*

Sommaire

5.1 Introduction	127
5.1.1 Moindres carrés	128
5.1.2 Pénalisation	128
5.1.3 Estimation de l'erreur quadratique	129
5.1.4 Réglage des hyperparamètres	130
5.2 SURE (Stein Unbiased Risk Estimate) généralisé	131
5.2.1 Modèle d'observation	132
5.2.2 Problème d'estimation	132
5.2.3 Estimateur de Stein généralisé de l'erreur quadratique	133
5.2.4 Stratégies de différences finies et de Monte Carlo	133
5.3 SUGAR (Stein's Unbiased Gradient estimator of the Risk) généralisé	135
5.3.1 Différentiation de l'estimateur SURE généralisé	135
5.3.2 Estimateurs séquentiels et différentiation récursive	136
5.3.3 Minimisation automatique de l'erreur quadratique	138
5.4 Sélection d'hyperparamètres pour la segmentation de textures	143
5.4.1 Formulation en terme du Modèle (5.1)	143
5.4.2 Propriétés des estimateurs de la régularité locale	144
5.4.3 Aspects pratiques du calcul de $\hat{R}_{v,\epsilon}$ et $\partial_{\Lambda}\hat{R}_{v,\epsilon}$	146
5.4.3.1 Structure de covariance des observations	146
5.4.3.2 Produit matriciel $\mathcal{S}\epsilon$	146
5.4.3.3 Opérateur linéaire \mathbf{A}	146
5.4.3.4 Calcul du terme constant $\text{Tr}(\mathbf{A}\mathcal{S}\mathbf{A}^*)$	147
5.5 Performances de la segmentation automatique	147
5.5.1 Réglages	147
5.5.1.1 Textures	147
5.5.1.2 Analyse multi-échelles	148
5.5.1.3 Mesure des performances	148
5.5.2 Questions algorithmiques	148

5.5.2.1	Algorithme primal-dual avec différentiation récursive	148
5.5.2.2	Gamme d'échelles	149
5.5.2.3	Paramètres de différences finies et de Monte Carlo	149
5.5.2.4	Algorithme de quasi-Newton	149
5.5.3	Covariance des logarithmes des coefficients dominants	150
5.5.3.1	Procédures de calcul de la covariance	150
5.5.3.2	Matrice de covariance partielle <i>versus</i> complète pour l'estimation de l'erreur quadratique	151
5.5.3.3	Matrice de covariance estimée <i>versus</i> « vraie »	152
5.5.4	Sélection automatique des hyperparamètres	152
5.5.4.1	Convergence effective de l'Algorithme de quasi-Newton	152
5.5.4.2	Performance de segmentation avec sélection automatique de Λ	154
5.6	Conclusion	155

Les travaux présentés dans ce chapitre ont été réalisés dans le cadre d'une collaboration avec Samuel Vaiter, Chargé de Recherche CNRS à l'Institut de Mathématiques de Bourgogne. Ils ont fait l'objet d'un article soumis au *Journal of Mathematical Imaging and Vision* [48]. Une boîte à outils MATLAB [GSUGAR20] permettant la mise en œuvre de la stratégie de réglage des hyperparamètres automatique et piloté par les données décrite dans ce chapitre pour les méthodes de segmentation de texture ROF_h, à contours « libres » et à contours « co-localisés » est disponible en ligne¹.

Le principal obstacle à l'utilisation des méthodes de segmentation à contours « libres » et à contours « co-localisés » proposées au Chapitre 4 sur des volumes de données, repose sur la sélection des paramètres de régularisation λ et α . En effet, les performances très satisfaisantes présentées à la Section 4.5, ont nécessité l'exploration d'une grille fine de paramètres, pour une durée de calcul de plusieurs heures par image. Ce chapitre s'intéresse donc aux stratégies de réglage automatique des paramètres qui permettront d'éviter cette coûteuse recherche sur grille. En outre, les approches de type Stein que nous allons explorer visent à estimer et minimiser l'erreur quadratique, sans connaître *a priori* la vérité terrain, ce qui sera crucial pour envisager des applications à des données réelles au Chapitre 6. Plutôt que de focaliser nos efforts sur les méthodes à contours « libres » et à contours « co-localisés », nous traitons le problème dans un cadre général et illustrons les performances de la solution proposée sur le problème de segmentation de texture. Ainsi, le formalisme développé bénéficie d'une grande souplesse et peut s'appliquer à d'autres problèmes d'estimation, comme nous le verrons avec le débruitage de signaux linéaires par morceaux au Chapitre 6.

Nous présenterons à la Section 5.1 le problème général d'estimation paramétrique considéré, ainsi que les difficultés soulevées par le problème de la sélection des hyperparamètres. Nous concentrerons notre attention sur les approches s'appuyant sur les travaux pionniers de Stein [54] qui consistent à construire un estimateur non biaisé de l'erreur d'estimation, appelé SURE, et valable pour des observations corrompues par un bruit gaussien i.i.d.. La Section 5.2 propose un estimateur de l'erreur quadratique SURE *généralisé*, i.e., permettant de prendre en compte un modèle de bruit gaussien quelconque, en s'appuyant sur une stratégie de Différences Finies, combinée à un échantillonnage Monte Carlo. L'estimateur que nous proposons ne faisant intervenir que la matrice de covariance, mais pas son inverse (difficile à calculer lorsque la structure de covariance est non triviale), il est utilisable en pratique pour des observations contenant des corrélations. En outre, nous introduisons un opérateur de projection, rendant le modèle suffisamment adaptable pour traiter des applications très variées.

Dans le but de mettre en œuvre une stratégie rapide de recherche de paramètre optimaux, l'estimateur SURE généralisé par Différences Finies et stratégie de Monte Carlo est dérivé par rapport aux paramètres de régularisation, conduisant à un estimateur du gradient de l'erreur, SUGAR

1. <https://github.com/bpascal-fr/gsugar>

généralisé, par différences finies et stratégie de Monte Carlo, dont le caractère asymptotiquement non biaisé est démontré à la Section 5.3. Les estimateurs SURE et SUGAR généralisés sont utilisés pour mettre en œuvre un schéma d'optimisation de quasi-Newton pour le réglage automatique des paramètres, présenté à la Section 5.3.3. De plus, le cas des estimateurs séquentiels est discuté à la Section 5.3.2.

Puis, à la Section 5.4, la procédure entière proposée est particularisée au problème de segmentation de textures à partir d'attributs fractals présenté au Chapitre 4, et détaillé dans les articles [46, 47]. Pour le modèle de segmentation de texture les données observées, sont les logarithmes des coefficients d'ondelettes dominants de l'image à segmenter, dont le bruit d'estimation présente des corrélations à la fois entre échelles et spatiales, ce qui correspond à une matrice de covariance non diagonale. Les estimateurs de l'erreur et du gradient de l'erreur, SURE et SUGAR généralisés, sont tous deux évalués grâce à une stratégie de Différences Finies, couplée à une méthode de Monte Carlo, dont toutes les étapes sont explicitées dans le cadre de la segmentation de texture. Enfin, la Section 5.5 est dédiée à un large panel de simulations numériques permettant de valider les performances de segmentation de textures avec sélection automatique des hyperparamètres. L'importance de prendre en compte l'intégralité de la structure de covariance dans les approches de Stein est particulièrement soulignée.

5.1 Introduction

De nombreux problèmes en traitement du signal et des images consistent à trouver le meilleur estimateur possible, $\hat{\mathbf{x}}$, d'une quantité $\bar{\mathbf{x}} \in \mathcal{H}$, à partir d'une observation $\mathbf{y} \in \mathcal{G}$ (où \mathcal{H} et \mathcal{G} sont des espaces de Hilbert isomorphes à \mathbb{R}^N et \mathbb{R}^P respectivement), pouvant être corrompue par un opérateur linéaire Φ qui représente une déformation ou une perte d'information, et par un bruit additif gaussien de moyenne nulle, $\zeta \sim \mathcal{N}(\mathbf{0}_P, \mathcal{S})$, dont la matrice de covariance $\mathcal{S} \in \mathbb{R}^{P \times P}$ est connue. On supposera, sans perte de généralité, que \mathcal{S} est *invertible*, en se ramenant au besoin à un espace de dimension plus faible. Cela nous amène au modèle d'observation suivant :

$$\mathbf{y} = \Phi \bar{\mathbf{x}} + \zeta. \quad (5.1)$$

Plusieurs exemples relevant de la catégorie des problèmes inverses incluent : la restauration d'images [13, 50], l'*inpainting* [15], la décomposition en texture et géométrie [3], mais également la segmentation de texture, comme nous l'avons proposé récemment [47] et exposé au Chapitre 4. Une voie largement explorée pour l'estimation de la quantité sous-jacente $\bar{\mathbf{x}}$ est la régression linéaire [37, 10], qui fournit un estimateur non biaisé $\hat{\mathbf{x}}_{\text{RL}}$. Cependant, les estimées correspondantes souffrent d'une grande variance, ce qui conduit à des erreurs importantes, notamment en présence du bruit ζ [6].

Une alternative repose sur la construction d'*estimateurs paramétriques*

$$\begin{aligned} \mathcal{G} \times \mathbb{R}^L &\longrightarrow \mathcal{H} \\ (\mathbf{y}, \Lambda) &\longmapsto \hat{\mathbf{x}}(\mathbf{y}; \Lambda) \end{aligned} \quad (5.2)$$

qui, en permettant d'induire un biais d'estimation, conduisent à une diminution drastique de la variance d'estimation. À partir de connaissances préalables à propos de la quantité véritable $\bar{\mathbf{x}}$, comme par exemple la parcimonie de la quantité $\bar{\mathbf{x}}$ [57], ou de sa dérivée [60, 52, 33, 32], ou encore de sa transformée en ondelettes [24], il est possible de construire des *estimateurs paramétriques* réalisant un compromis entre l'accord avec le modèle (5.1) et les contraintes de régularité de l'estimée. En général, ce compromis est piloté par un petit nombre $L = \mathcal{O}(1)$ de paramètres, stockés dans un vecteur $\Lambda \in \mathbb{R}^L$. Une classe très populaire d'estimateurs paramétriques repose sur une pénalisation du terme de moindres carrés mesurant la fidélité au modèle d'observation (5.1), for-

mulé comme un problème de minimisation

$$\hat{\mathbf{x}}(\mathbf{y}; \Lambda) \in \underset{\mathbf{x} \in \mathcal{H}}{\text{Argmin}} \frac{1}{2} \|\mathbf{y} - \Phi \mathbf{x}\|_{\mathcal{W}}^2 + \|\mathbf{U}_\Lambda \mathbf{x}\|_q^q \quad (5.3)$$

avec $\|\cdot\|_{\mathcal{W}}$ la distance de Mahalanobis associée à $\mathcal{W} \in \mathbb{R}^{P \times P}$, définie comme

$$\|\mathbf{y} - \Phi \mathbf{x}\|_{\mathcal{W}} \triangleq \sqrt{(\mathbf{y} - \Phi \mathbf{x})^\top \mathcal{W} (\mathbf{y} - \Phi \mathbf{x})}. \quad (5.4)$$

$\mathbf{U}_\Lambda : \mathcal{H} \rightarrow \mathcal{Q}$, un opérateur linéaire paramétré par Λ , et $\|\cdot\|_q$, la norme ℓ_q , avec $q \geq 1$, dans l'espace de Hilbert \mathcal{Q} , permettent d'imposer des contraintes sur la forme de l'estimée $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$.

5.1.1 Moindres carrés

Tandis que les « moindres carrés ordinaires » (*Ordinary Least Squares*, OLS) utilisent usuellement le carré de la norme ℓ_2 dans leur attache aux données, i.e., $\mathcal{W} = \mathbf{I}_P$, les moindres carrés généralisés [55] font usage de la structure de covariance du bruit *via* $\mathcal{W} = \mathcal{S}^{-1}$, qui, dans le cas gaussien, caractérise entièrement la statistique du bruit. Cette approche généralisée équivaut à une décorrélation des données et à une égalisation des niveaux de bruit qui serait effectuées avant de faire la régression. Or, le théorème de Gauss-Markov [1] assure que minimiser les moindres carrés généralisés avec $\mathcal{W} = \mathcal{S}^{-1}$ fournit le meilleur estimateur *linéaire* de $\bar{\mathbf{x}}$, ce qui plaide en faveur de l'utilisation de la distance de Mahalanobis, pondérée par l'inverse de la matrice de covariance, dans le terme d'attache aux données des moindres carrés pénalisés.

Néanmoins, en pratique, les moindres carrés généralisés ne requièrent pas seulement la connaissance de la matrice de covariance, mais également d'être capable de l'inverser. Pour des observations décorrélées, \mathcal{S} est diagonale et, à condition qu'elle soit bien conditionnée, son inverse est facilement calculable numériquement. Au contraire, calculer \mathcal{S}^{-1} peut être extrêmement difficile pour des données corrélées, puisque dans ce cas la matrice \mathcal{S} n'est plus diagonale, et que, de plus, sa taille vaut le carré de la dimension de l'espace de Hilbert \mathcal{G} .

Donc, afin de prendre en compte un bruit gaussien ζ potentiellement corrélé, utiliser les moindres carrés ordinaires est souvent nécessaire, même s'ils ne bénéficient pas des mêmes garanties théoriques que les moindres carrés généralisés. Malgré tout, le cœur de cette contribution est de montrer que la connaissance de \mathcal{S} , loin d'être inutile, peut être mise à profit dans l'estimation de l'erreur quadratique.

5.1.2 Pénalisation

De nombreux estimateurs classiques peuvent s'exprimer sous la forme des moindres carrés pénalisés de l'Équation (5.3) grâce à un choix approprié de la norme ℓ_q et de l'opérateur linéaire \mathbf{U}_Λ . Le filtrage linéaire est obtenu pour $q = 2$ [27], la forme du filtre est alors représentée par l'opérateur \mathbf{U}_Λ [33], et les propriétés du filtre, comme par exemple sa largeur de bande du filtre sont pilotée par les hyperparamètres Λ . Il est par exemple fréquent, en traitement d'images, d'imposer des contraintes sur le gradient spatial de l'intensité lumineuse de l'image, en utilisant l'opérateur \mathbf{D} de différences finies horizontales et verticales et un paramètre de régularisation $\Lambda = \lambda > 0$ ($L = 1$). Par exemple, la régularité de l'estimée est favorisée grâce à l'utilisation du carré de la norme ℓ_2 , réalisant une régularisation de Tikhonov [60, 33], dans laquelle $q = 2$ et $\|\mathbf{U}_\Lambda \mathbf{x}\|_q^q \triangleq \lambda \|\mathbf{D} \mathbf{x}\|_2^2$. Une autre pénalisation standard est la Variation Totale anisotrope [52] $\|\mathbf{U}_\Lambda \mathbf{x}\|_q^q \triangleq \lambda \|\mathbf{D} \mathbf{x}\|_1$, correspondant à $q = 1$, où la norme ℓ_1 force la parcimonie du gradient spatial.

5.1.3 Estimation de l'erreur quadratique

Le but du Problème (5.3) est d'obtenir une estimation fidèle, $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$, de la quantité véritable $\bar{\mathbf{x}}$, l'erreur étant mesurée par l'*erreur quadratique* définie comme :

$$\mathbb{E} \|\mathbf{B}\hat{\mathbf{x}}(\mathbf{y}; \Lambda) - \mathbf{B}\bar{\mathbf{x}}\|_{\mathcal{W}}^2 \quad (5.5)$$

avec \mathbf{B} un opérateur linéaire, qui permet de considérer différents types d'erreurs. Par exemple, lorsque $\mathbf{B} = \mathbf{\Pi}$ est un projecteur orthogonal sur un sous-espace de \mathcal{H} [26], l'*erreur quadratique projetée* (5.5) mesure l'erreur d'estimation de la quantité projetée $\mathbf{\Pi}\bar{\mathbf{x}}$. Ce cas inclut l'erreur quadratique usuelle, lorsque $\mathbf{B} = \mathbf{I}_N$. Inversement, lorsque $\mathbf{B} = \mathbf{\Phi}$, l'erreur (5.5) mesure la qualité de la prédiction $\hat{\mathbf{y}}(\mathbf{y}; \Lambda) \triangleq \mathbf{\Phi}\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$, par rapport à l'observation non bruitée, $\bar{\mathbf{y}} \triangleq \mathbf{\Phi}\bar{\mathbf{x}}$, dans l'espace \mathcal{G} des observées. On parle alors d'*erreur de prédiction*.

En suivant le théorème de Gauss-Markov [1] déjà mentionné ci-dessus, dans le cas où $\mathbf{B} = \mathbf{\Phi}$, le choix le plus direct pour la distance de Mahalanobis dans la définition de l'erreur quadratique (5.5) est $\mathcal{W} = \mathcal{S}^{-1}$. Néanmoins, en pratique, ce choix est difficilement réalisable car il nécessite la connaissance de \mathcal{S}^{-1} , qui peut être non triviale à calculer, voire numériquement inaccessible dans le cas d'un bruit corrélé dont la matrice de covariance est non diagonale. Par conséquent, nous nous concentrerons uniquement sur l'erreur quadratique ordinaire, i.e., sur $\mathcal{W} = \mathbf{I}_p$. Il est intéressant de noter que ces deux approches, bien que construites à partir de deux définitions de l'erreur différentes, ont plusieurs propriétés communes, qui seront mentionnées brièvement dans la suite (voir Remarque 5.2.3).

Le principal obstacle au calcul, et *a fortiori* à la minimisation, de l'erreur (5.5) réside dans le fait, qu'en pratique, aucun accès à la véritable quantité $\bar{\mathbf{x}}$ n'est possible. Par conséquent, mesurer l'erreur quadratique (5.5) nécessite en premier lieu de construire un estimateur de l'erreur moyenne $\mathbb{E} \|\mathbf{B}\hat{\mathbf{x}}(\mathbf{y}; \Lambda) - \mathbf{B}\bar{\mathbf{x}}\|_{\mathcal{W}}^2$, n'utilisant pas $\bar{\mathbf{x}}$.

Ce problème a tout d'abord été traité dans le cas d'un bruit additif gaussien dont les composantes sont indépendantes et identiquement distribuées, i.e., pour une matrice de covariance scalaire $\mathcal{S} = \rho^2 \mathbf{I}_p$ connue, par Stein [54, 58], grâce à une ingénieuse intégration par partie, menant à l'estimateur SURE (*Stein Unbiased Risk Estimate*) [23, 43, 51, 59], initialement formulé pour l'*erreur de prédiction*,

$$\|(\mathbf{\Phi}\hat{\mathbf{x}}(\mathbf{y}; \Lambda) - \mathbf{y})\|_{\mathcal{W}}^2 + 2\rho^2 \text{Tr}(\mathbf{\Phi} \partial_{\mathbf{y}} \hat{\mathbf{x}}(\mathbf{y}; \Lambda)) - p\rho^2, \quad (5.6)$$

dont l'espérance vaut l'erreur quadratique (5.5) avec $\mathbf{B} = \mathbf{\Phi}$. Ces dernières années, l'estimateur SURE (5.6) a été largement utilisé pour des applications en statistique et en traitement du signal et des images [23, 11, 49]. Récemment, il a été étendu au cas d'un bruit dont les composantes sont indépendantes mais pas nécessairement identiquement distribuées [16, 64], ce qui correspond à une matrice de covariance diagonale $\mathcal{S} = \text{diag}(\rho_1^2, \dots, \rho_p^2)$, puis au cas d'un bruit gaussien éventuellement corrélé, dont la matrice de covariance \mathcal{S} est très générale. Cependant, autant que nous le sachions, très peu de validations numériques ont été menées sur l'estimateur SURE dans le cas d'un bruit gaussien dont la matrice de covariance est non scalaire. Une exception notable est l'article de Chaux *et coll.* [16], dans lequel les résultats sont illustrés dans des expériences numériques sur des données multi-composantes décorréelées mais de niveaux de bruit différents d'une composante à l'autre. Le bruit étant supposé indépendant, cette situation correspond à une matrice de covariance diagonale $\mathcal{S} = \text{diag}(\rho_1^2, \dots, \rho_p^2)$, avec ρ_i^2 la variance du bruit sur la $i^{\text{ème}}$ composante. De plus, dans le cas d'un bruit ni identiquement distribué, ni indépendant, des estimateurs SURE généralisés ont été proposés, par exemple pour les familles exponentielles [34, 26] ou le bruit de Poisson [35, 41, 38].

Dans la pratique, l'évaluation de l'estimateur de Stein (5.6), ou de ses versions généralisées, peut s'avérer un problème à part entière, notamment à cause de la présence de la Jacobienne de

l'estimateur par rapport aux observations, $\partial_{\mathbf{y}}\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$. En effet, d'une part, cette matrice, de taille $P \times N$, est très coûteuse à stocker et à manipuler. D'autre part, dans le cas où $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ est obtenu à partir d'un algorithme proximal [4, 17, 45, 18] résolvant le Problème (5.3), l'absence d'expression close de l'estimateur $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ empêche un accès direct à sa Jacobienne. Ces deux points ont conduit à la construction d'estimateurs de Stein plus sophistiqués, mais plus faciles à évaluer. Dans le cas d'un bruit gaussien i.i.d., les stratégies de type Monte Carlo [30, 51, 21] fournissent un estimateur SURE Monte Carlo [51], où la manipulation de la Jacobienne de taille $P \times N$, est remplacée par la manipulation d'un vecteur de taille P . Afin de pallier le problème de l'accès à la Jacobienne, Vonesch *et coll.* ont proposé dans [62] de réaliser une différentiation récursive à l'intérieur du schéma d'optimisation résolvant (5.3), qui bénéficie de quelques garanties théoriques [28]. Bien que restant en partie heuristique, cette approche a montré son efficacité dans une grande variété d'applications [21]. Une alternative à l'estimation de la Jacobienne, étudiée dans le cas d'un bruit gaussien i.i.d. [53, 51], consiste à utiliser une stratégie de différences finies, pour proposer des estimateurs SURE par Différences Finies (éventuellement couplée à une stratégie de Monte Carlo) [51, 21] ne requérant que le calcul de l'estimateur $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ mais pas de sa Jacobienne. La validité des estimateurs SURE avec stratégies de Différences Finies et/ou de Monte Carlo repose à la fois sur le modèle de bruit et sur les propriétés de l'estimateur $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$, ainsi leur extension à des modèles de bruits plus généraux demande une étude détaillée.

5.1.4 Réglage des hyperparamètres

L'Équation (5.3) montre clairement que l'estimée $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ dépend fortement du choix des paramètres de régularisation Λ . Ainsi, la façon de régler le plus finement possible ces paramètres est un problème de longue date en traitement du signal et des images. Une formulation usuelle du problème consiste à minimiser l'erreur quadratique par rapport aux paramètres de régularisation Λ , et de résoudre

$$\underset{\Lambda}{\text{minimize}} \mathbb{E} \left\| \mathbf{B}\hat{\mathbf{x}}(\mathbf{y}; \Lambda) - \mathbf{B}\bar{\mathbf{x}} \right\|_{\mathcal{W}}^2. \quad (5.7)$$

Comme souligné par Eldar [26], l'estimateur $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ minimisant le critère SURE (5.6), permet d'atteindre une erreur quadratique (5.7) plus faible que l'estimateur sélectionné par une approche classique de maximum *a posteriori* (MAP) appliquée au problème (5.1).

La manière la plus directe de résoudre (5.7) consiste à calculer l'estimateur SURE (5.6) sur une grille de paramètres [24, 51, 26], et de sélectionner le paramètre de la grille pour lequel il est minimal. Cependant, les méthodes de recherche sur grille induisent d'importants coûts de calcul, et ce pour plusieurs raisons. Tout d'abord, la taille de la grille augmente algébriquement avec le nombre L de paramètres de régularisation, et une recherche exhaustive est donc souvent hors de portée. Récemment, des méthodes de recherche aléatoires ont été proposées pour améliorer l'efficacité de la recherche sur grille [7]. Malgré tout, pour $L \geq 3$, cette stratégie reste très coûteuse, si ce n'est inaccessible. De plus, une difficulté supplémentaire apparaît lorsque $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ est calculé à partir d'un algorithme proximal de résolution de (5.3). En effet, lorsque le terme de pénalisation $\|\mathbf{U}_{\Lambda}\mathbf{x}\|_q^q$ est non différentiable, l'algorithme proximal résolvant (5.3) souffre d'un taux de convergence faible, ce qui rend l'évaluation de l'estimateur de Stein (5.6) en chaque point de la grille très exigeant en temps de calcul. Bien que des schémas accélérés aient été proposés [5, 14], la recherche sur grille reste très coûteuse dès lors que $L \geq 2$, empêchant son usage en pratique.

Lorsqu'une formule explicite de l'estimateur de Stein est disponible, une minimisation exacte de l'estimateur SURE par rapport aux paramètres de régularisation Λ est parfois possible. C'est le cas par exemple lorsque (5.3) est construit à partir d'une pénalisation de Tikhonov pour laquelle Thompson *et coll.* [56], Galatsanos *et coll.* in [29], et Desbat *et coll.* in [22] ont tiré parti de la formule explicite, linéaire, de $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ pour déduire le « meilleur » paramètre de régularisation, i.e., pour résoudre (5.7). Une autre expression explicite bien connue est celle du « seuillage doux », qui est largement utilisé pour le débruitage par réduction des coefficients d'ondelettes, par exemple

dans [24, 40]. Il faut noter que la méthode de validation croisée généralisée [31] s'appuie également sur l'existence d'une expression explicite de l'estimée $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ pour le réglage des paramètres, mais d'une manière légèrement différente, en travaillant sur l'erreur de prédiction, résolvant (5.7) pour $\mathbf{B} = \Phi$. La validation croisée généralisée et les estimateurs de type Stein ont été comparés indépendamment par Li [39], Thompson [56], et Desbat [22]. En outre, des méthodes bayésiennes ont également été proposées pour gérer un grand nombre de paramètres de régularisation simultanément, i.e., $L \gg 1$, parmi lesquelles la méthode SBMO (*Sequential Model-Based Optimization*), qui réalise un échantillonnage judicieux du domaine des hyperparamètres [8]. Ces méthodes sont particulièrement adaptées à l'apprentissage statistique, puisqu'elles permettent de manipuler une très grande quantité de paramètres sans requérir la connaissance du gradient de la fonction de coût [9].

Afin d'aller plus loin que les méthodes d'échantillonnage (éventuellement aléatoires), des approches élaborées à partir de schémas de minimisation ont été proposées, nécessitant des estimateurs de l'erreur quadratique suffisamment réguliers, ainsi que la possibilité d'accéder aux dérivées de ces estimateurs par rapport à Λ . À partir d'une expression explicite C^∞ de l'estimateur de Poisson de l'erreur quadratique, Deledalle *et coll.* [20] ont construit un algorithme de Newton pour la résolution de (5.7). Néanmoins, cette stratégie n'est pas généralisable, étant donné qu'il est très rare que l'accès à toutes les dérivées de l'estimateur de l'erreur soit possible. Dans le cas d'un bruit gaussien, indépendant et identiquement distribué Chaux *et coll.* [16] proposent, et évaluent numériquement, un algorithme empirique de minimisation alternée pour le choix automatique des paramètres de régularisation, sans garantie de convergence néanmoins. Deledalle *et coll.* [21] se sont également intéressés au cas d'un bruit gaussien indépendant et identiquement distribué dans le cadre des estimateurs construits comme la solution de (5.3). Ils ont formulé des conditions suffisantes pour que $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ soit différentiable par rapport aux paramètres Λ , puis en ont déduit la différentiabilité de l'estimateur SURE. Cela les a amenés à l'élaboration d'un estimateur du gradient de l'erreur quadratique appelé SUGAR (*Stein Unbiased GrAdient estimator of the Risk*) dans le but de mettre en œuvre une descente de quasi-Newton pour résoudre (5.7) à l'aide d'une stratégie de Broyden, Fletcher Goldfarb et Shanno (BFGS). L'efficacité de cette approche pour le réglage automatique des hyperparamètres a été illustrée dans le cadre d'une méthode de déconvolution spatiale et spectrale pour des données multi-spectrales de grande taille corrompue par un bruit gaussien indépendant et identiquement distribué [2].

Pour les Sections 5.2 et 5.3 nous nous intéressons à l'estimation et la minimisation de l'erreur quadratique dans le cas d'un modèle de bruit gaussien ζ , de matrice de covariance \mathcal{S} quelconque, en utilisant le formalisme des moindres carrés ordinaires, c'est-à-dire (5.5) avec $\mathcal{W} = \mathbf{I}_p$, permettant de gérer des données présentant non seulement plusieurs niveaux de bruit, mais également des corrélations.

5.2 SURE (*Stein Unbiased Risk Estimate*) généralisé

Cette Section détaille l'extension de l'estimateur SURE (5.6), pour $\mathcal{W} = \mathbf{I}_p$, au cas où les observations sont corrompues par un bruit gaussien corrélé, conduisant à la construction d'un estimateur de Stein généralisé, par Différences Finies et méthode de Monte Carlo, $\hat{R}_{v,\varepsilon}(\mathbf{y}; \Lambda | \mathcal{S})$, défini à l'Équation (5.17).

Notations. La Jacobienne par rapport aux observations \mathbf{y} d'un estimateur paramétrique différentiable $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ est notée $\partial_{\mathbf{y}} \hat{\mathbf{x}}(\mathbf{y}; \Lambda)$.

5.2.1 Modèle d'observation

Dans ce chapitre, nous considérons des observations \mathbf{y} , supposées suivre le Modèle (5.1), comme stipulé ci-après à l'Hypothèse 5.1, avec un opérateur de dégradation Φ supposé de rang plein selon l'Hypothèse 5.2.

Hypothèse 5.1 (Gaussianité). Le bruit additif $\zeta \in \mathcal{G}$ est gaussien : $\zeta \sim \mathcal{N}(\mathbf{0}_P, \mathcal{S})$, où $\mathbf{0}_P$ est le vecteur nul de \mathcal{G} et $\mathcal{S} \in \mathbb{R}^{P \times P}$ est la matrice de covariance du bruit, avec $P = \dim(\mathcal{G})$. Par conséquent, la densité de probabilité associée au Modèle (5.1) s'écrit

$$\frac{1}{\sqrt{(2\pi)^P |\det \mathcal{S}|}} \exp\left(-\frac{\|\mathbf{y} - \Phi \bar{\mathbf{x}}\|_{\mathcal{S}^{-1}}^2}{2}\right). \quad (5.8)$$

Hypothèse 5.2 (Rang plein). L'opérateur linéaire $\Phi : \mathcal{H} \rightarrow \mathcal{G}$ est de rang plein, ou de manière équivalente $\Phi^* \Phi$ est inversible.

5.2.2 Problème d'estimation

Soit $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ un estimateur paramétrique de la quantité véritable $\bar{\mathbf{x}} \in \mathcal{H}$, défini de manière unique à partir des observations $\mathbf{y} \in \mathcal{G}$ et contrôlé par des hyperparamètres $\Lambda \in \mathbb{R}^L$.

Remarque. Par exemple, $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ peut correspondre à l'estimateur des moindres carrés ordinaires pénalisés, défini à l'Équation (5.3). Dans ce cas l'Hypothèse de rang plein 5.2 assure l'unicité du minimiseur. Cependant, nous souhaitons souligner que les résultats Sections 5.2 et 5.3 peuvent s'appliquer au Problème (5.7) dans un cadre plus général.

Nous explorons également la possibilité que la quantité d'intérêt puisse être une projection de $\bar{\mathbf{x}}$ sur un sous-espace \mathcal{I} de \mathcal{H} . En effet, dans les travaux présentés au Chapitre 4, pour le problème de segmentation de textures, la variable principale est la régularité locale \mathbf{h} , tandis que la variable \mathbf{v} de puissance locale est uniquement auxiliaire. La possibilité de se concentrer sur une portion des données pour le calcul de l'erreur est néanmoins utile dans d'autres applications. Dans le cas de l'imagerie médicale, il est courant de souhaiter se concentrer sur une zone d'intérêt, la projection consiste alors à définir l'erreur quadratique sur une sous région de l'image, i.e., en prenant en compte un nombre restreint de pixels. L'étude des images (hyper)spectrales fournit un autre exemple, la projection permettant dans ce cas de se concentrer sur quelques canaux contenant l'information d'intérêt. Dans un cadre plus large, on pourra également penser aux problèmes de physiques dans lesquels seules quelques variables possèdent une interprétation physique et sont mesurables, tandis que les autres sont des quantités auxiliaires.

Définition 5.1. Soit $\Pi : \mathcal{H} \rightarrow \mathcal{H}$ l'opérateur linéaire réalisant la projection orthogonale sur le sous-espace \mathcal{I} contenant l'information pertinente de $\bar{\mathbf{x}}$. À partir de l'Hypothèse 5.2 et de l'opérateur de projection Π , l'opérateur linéaire $\mathbf{A} : \mathcal{G} \rightarrow \mathcal{H}$ défini comme la composition

$$\mathbf{A} \triangleq \Pi (\Phi^* \Phi)^{-1} \Phi^*, \quad (5.9)$$

réalise la régression linéaire associée au Modèle (5.1), suivie de la projection orthogonale sur \mathcal{I} .

L'erreur quadratique est définie comme l'*erreur d'estimation projetée*, portant sur la quantité d'intérêt $\Pi \bar{\mathbf{x}}$, commise par l'estimateur $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$, et mesurée grâce au carré d'une norme ℓ_2 :

$$R[\hat{\mathbf{x}}](\Lambda) \triangleq \mathbb{E}_{\zeta} \|\Pi \hat{\mathbf{x}}(\mathbf{y}; \Lambda) - \Pi \bar{\mathbf{x}}\|_2^2. \quad (5.10)$$

Le but de ce travail est le réglage automatique des paramètres de régularisation Λ minimisant l'erreur quadratique (5.10) définie ci-dessus. Cependant en pratique, l'erreur quadratique exacte $R[\hat{\mathbf{x}}](\Lambda)$ étant inconnue, *a fortiori* le choix optimal de paramètres Λ^\dagger , vérifiant

$$\Lambda^\dagger \in \underset{\Lambda \in \mathbb{R}^L}{\text{Argmin}} R[\hat{\mathbf{x}}](\Lambda) \quad (5.11)$$

est inaccessible. Pour cela, nous proposons une procédure détaillée pour approcher finement Λ^\dagger , grâce à la minimisation d'un estimateur de Stein généralisé approximant $R[\hat{\mathbf{x}}](\Lambda)$.

5.2.3 Estimateur de Stein généralisé de l'erreur quadratique

Les approches de type Stein pour l'estimation de l'erreur reposent de façon majeure, sur l'hypothèse suivante, portant sur l'estimateur $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$.

Hypothèse 5.3 (Régularité et intégrabilité). L'estimateur $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ est continu et faiblement différentiable [28, Définition 4.1] par rapport aux observations \mathbf{y} . De plus, les quantités $\langle \mathbf{A}^* \mathbf{\Pi} \hat{\mathbf{x}}(\mathbf{y}; \Lambda), \boldsymbol{\zeta} \rangle$ et $\partial_{\mathbf{y}} \hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ sont intégrables par rapport à $\boldsymbol{\zeta}$ pour la densité gaussienne :

$$\frac{1}{\sqrt{(2\pi)^P |\det \mathcal{S}|}} \exp\left(-\frac{\|\boldsymbol{\zeta}\|_{\mathcal{S}^{-1}}^2}{2}\right).$$

Lorsque le bruit $\boldsymbol{\zeta}$ est gaussien, avec une matrice de covariance éventuellement non triviale, la généralisation de l'estimateur de Stein initial (5.6) est fournie au Théorème 5.1, constituant le point de départ de ce travail.

Théorème 5.1. *Cas particulier de [26, Théorème 1] Considérons le Modèle (5.1), ainsi que les Hypothèses 5.1 (Gaussianité), 5.2 (Rang plein), 5.3 (Régularité et intégrabilité), le projecteur $\mathbf{\Pi}$ et l'opérateur linéaire \mathbf{A} défini à l'Équation (5.9). Le lemme de Stein généralisé s'applique, et conduit à*

$$\mathbb{R}[\hat{\mathbf{x}}](\Lambda) = \mathbb{E}_{\boldsymbol{\zeta}} \left[\|\mathbf{A}(\Phi \hat{\mathbf{x}}(\mathbf{y}; \Lambda) - \mathbf{y})\|_2^2 + 2\text{Tr}(\mathcal{S} \mathbf{A}^* \mathbf{\Pi} \partial_{\mathbf{y}} \hat{\mathbf{x}}(\mathbf{y}; \Lambda)) - \text{Tr}(\mathbf{A} \mathcal{S} \mathbf{A}^*) \right], \quad (5.12)$$

la quantité entre les crochets de l'espérance correspondant à l'estimateur SURE généralisé.

Démonstration. Une preuve détaillée est fournie en Annexe C.1. □

Remarque. Il est intéressant de noter que, lorsqu'on considère la distance de Mahalanobis avec $\mathcal{W} = \mathcal{S}^{-1}$ dans la définition de l'erreur (5.5), l'estimateur de Stein possède la même structure globale, mais, au lieu de faire intervenir la matrice \mathcal{S} , il s'écrit en fonction de son inverse \mathcal{S}^{-1} :

$$\tilde{\mathbb{R}}[\hat{\mathbf{x}}](\Lambda) = \mathbb{E}_{\boldsymbol{\zeta}} \left[\|\mathbf{A}(\Phi \hat{\mathbf{x}}(\mathbf{y}; \Lambda) - \mathbf{y})\|_{\mathcal{S}^{-1}}^2 + 2\text{Tr}(\mathbf{A}^* \mathbf{\Pi} \partial_{\mathbf{y}} \hat{\mathbf{x}}(\mathbf{y}; \Lambda)) - \text{Tr}(\mathbf{A} \mathbf{A}^*) \right]. \quad (5.13)$$

5.2.4 Stratégies de différences finies et de Monte Carlo

Dans l'expression de SURE généralisé (5.12), la quantité $\text{Tr}(\mathcal{S} \mathbf{A}^* \mathbf{\Pi} \partial_{\mathbf{y}} \hat{\mathbf{x}}(\mathbf{y}; \Lambda))$, appelée les *degrés de liberté*, concentre la majeure partie des difficultés rencontrées lors du calcul de l'estimateur de Stein généralisé pour les problèmes de traitement de données, comme le montre la littérature prolifique s'intéressant à ce problème, y compris dans le cas où $\mathcal{S} \propto \mathbf{I}_P$ [36, 59, 25, 61]. En effet, il fait intervenir le produit d'une matrice de taille $P \times P$, $\mathcal{S} \mathbf{A}^* \mathbf{\Pi}$, avec la matrice Jacobienne $\partial_{\mathbf{y}}(\hat{\mathbf{x}}(\mathbf{y}; \Lambda))$, également de taille $P \times P$.

Dans une large majorité d'applications $P \gg 1$. Par conséquent, non seulement, le produit de deux matrices de taille $P \times P$ est extrêmement coûteux en terme d'effort de calcul, mais la matrice Jacobienne, à cause de sa grande taille, s'avère également particulièrement coûteuse à calculer (ou même à estimer). Une stratégie en deux temps, combinant différences finies et échantillonnage Monte Carlo, permet, grâce à l'Hypothèse 5.4 formulée ci-dessous, de surmonter ces difficultés et de construire un estimateur de Stein de l'erreur, utilisable en pratique, noté $\hat{\mathbb{R}}_{v,\varepsilon}(\mathbf{y}; \Lambda | \mathcal{S})$, et à défini à l'Équation (5.17).

Hypothèse 5.4 (Lipschitzianité par rapport aux observations). Soit $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ un estimateur de $\bar{\mathbf{x}}$, dépendant d'observations \mathbf{y} , et paramétré par Λ .

(i) La fonction $\mathbf{y} \mapsto \hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ est uniformément L_1 -lipschitzienne.

(ii) $\forall \Lambda \in \mathbb{R}^L$, $\hat{\mathbf{x}}(\mathbf{0}_P; \Lambda) = \mathbf{0}_N$, avec $\mathbf{0}_N$ (resp. $\mathbf{0}_P$) le vecteur nul de \mathcal{H} (resp. \mathcal{G}).

Étape 1. Estimation de la trace par échantillonnage Monte Carlo :

Pour estimer le terme de *degrés de liberté* dans la pratique, la première étape est de remarquer qu'il est beaucoup moins coûteux de calculer le produit de la matrice $\mathcal{S}\mathbf{A}^*\mathbf{\Pi}$, de taille $P \times P$, avec $\partial_{\mathbf{y}}(\hat{\mathbf{x}}(\mathbf{y}, \mathbf{\Lambda}))[\boldsymbol{\varepsilon}] \in \mathbb{R}^P$, la Jacobienne *appliquée* à un vecteur $\boldsymbol{\varepsilon} \in \mathbb{R}^P$. De plus, un calcul rapide montre que, si $\boldsymbol{\varepsilon} \in \mathbb{R}^P$ est une variable aléatoire gaussienne normalisée, i.e., $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}_P, \mathbf{I}_P)$, et $\mathbf{M} \in \mathbb{R}^{P \times P}$ une matrice quelconque, alors

$$\text{Tr}(\mathbf{M}) = \mathbb{E}_{\boldsymbol{\varepsilon}} \langle \mathbf{M}\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon} \rangle.$$

Ainsi, en s'inspirant de la suggestion de [30, 51, 21], l'accès à $\partial_{\mathbf{y}}(\hat{\mathbf{x}}(\mathbf{y}, \mathbf{\Lambda}))[\boldsymbol{\varepsilon}]$ permet de calculer

$$\begin{aligned} \text{Tr}(\mathcal{S}\mathbf{A}^*\mathbf{\Pi}\partial_{\mathbf{y}}\hat{\mathbf{x}}(\mathbf{y}; \mathbf{\Lambda})) &= \mathbb{E}_{\boldsymbol{\varepsilon}} \langle \mathcal{S}\mathbf{A}^*\mathbf{\Pi}\partial_{\mathbf{y}}\hat{\mathbf{x}}(\mathbf{y}; \mathbf{\Lambda})[\boldsymbol{\varepsilon}], \boldsymbol{\varepsilon} \rangle \\ &= \mathbb{E}_{\boldsymbol{\varepsilon}} \langle \mathbf{A}^*\mathbf{\Pi}\partial_{\mathbf{y}}\hat{\mathbf{x}}(\mathbf{y}; \mathbf{\Lambda})[\boldsymbol{\varepsilon}], \mathcal{S}\boldsymbol{\varepsilon} \rangle, \end{aligned} \quad (5.14)$$

en utilisant le caractère symétrique de la matrice de covariance \mathcal{S} . Enfin, $\langle \mathbf{A}^*\mathbf{\Pi}\partial_{\mathbf{y}}\hat{\mathbf{x}}(\mathbf{y}; \mathbf{\Lambda})[\boldsymbol{\varepsilon}], \mathcal{S}\boldsymbol{\varepsilon} \rangle$ fournit donc un estimateur des *degrés de liberté*.

Étape 2. Dérivée première par différences finies :

La seconde étape consiste à trouver une manière d'estimer la dérivée $\partial_{\mathbf{y}}(\hat{\mathbf{x}}(\mathbf{y}, \mathbf{\Lambda}))[\boldsymbol{\varepsilon}]$ quand on ne dispose pas d'un accès direct à la Jacobienne $\partial_{\mathbf{y}}(\hat{\mathbf{x}}(\mathbf{y}, \mathbf{\Lambda}))$. Dans ce cas, la dérivée peut être estimée en utilisant la variable aléatoire gaussienne normalisée, $\boldsymbol{\varepsilon}$, et un pas $v > 0$ et en effectuant un développement de Taylor à l'ordre un (voir le cas gaussien i.i.d. [51, Théorème 2]) :

$$\hat{\mathbf{x}}(\mathbf{y} + v\boldsymbol{\varepsilon}; \mathbf{\Lambda}) - \hat{\mathbf{x}}(\mathbf{y}; \mathbf{\Lambda}) \underset{v \rightarrow 0}{\simeq} \partial_{\mathbf{y}}(\hat{\mathbf{x}}(\mathbf{y}; \mathbf{\Lambda})) [v\boldsymbol{\varepsilon}] \iff \partial_{\mathbf{y}}\hat{\mathbf{x}}(\mathbf{y}; \mathbf{\Lambda})[\boldsymbol{\varepsilon}] = \lim_{v \rightarrow 0} \frac{1}{v} (\hat{\mathbf{x}}(\mathbf{y} + v\boldsymbol{\varepsilon}; \mathbf{\Lambda}) - \hat{\mathbf{x}}(\mathbf{y}; \mathbf{\Lambda})), \quad (5.15)$$

Il s'ensuit

$$\begin{aligned} \text{Tr}(\mathcal{S}\mathbf{A}^*\mathbf{\Pi}\partial_{\mathbf{y}}\hat{\mathbf{x}}(\mathbf{y}; \mathbf{\Lambda})) &= \mathbb{E}_{\boldsymbol{\varepsilon}} \lim_{v \rightarrow 0} \frac{1}{v} \langle \mathcal{S}\mathbf{A}^*\mathbf{\Pi}(\hat{\mathbf{x}}(\mathbf{y} + v\boldsymbol{\varepsilon}; \mathbf{\Lambda}) - \hat{\mathbf{x}}(\mathbf{y}; \mathbf{\Lambda})), \boldsymbol{\varepsilon} \rangle \\ &= \mathbb{E}_{\boldsymbol{\varepsilon}} \lim_{v \rightarrow 0} \frac{1}{v} \langle \mathbf{A}^*\mathbf{\Pi}(\hat{\mathbf{x}}(\mathbf{y} + v\boldsymbol{\varepsilon}; \mathbf{\Lambda}) - \hat{\mathbf{x}}(\mathbf{y}; \mathbf{\Lambda})), \mathcal{S}\boldsymbol{\varepsilon} \rangle. \end{aligned} \quad (5.16)$$

À partir de la Formule (5.16) et de l'Hypothèse 5.4, le théorème suivant fournit un estimateur asymptotiquement non biaisé de l'erreur quadratique, par Différences Finies et méthode de Monte Carlo (DFMC), qui peut être utilisé dans une grande variété de problèmes d'estimation.

Théorème 5.2. Soit \mathbf{y} des observations suivant le Modèle (5.1), \mathbf{A} l'opérateur linéaire défini à l'Équation (5.9) et un estimateur $\hat{\mathbf{x}}(\mathbf{y}; \mathbf{\Lambda})$ vérifiant les Hypothèses 5.1 (Gaussianité), 5.2 (Rang plein), 5.3 (Régularité et intégrabilité), et 5.4 (Lipschitzianité par rapport à \mathbf{y}). L'estimateur SURE généralisé DFMC, s'écrivant

$$\hat{\mathbf{R}}_{v,\boldsymbol{\varepsilon}}(\mathbf{y}; \mathbf{\Lambda}|\mathcal{S}) \triangleq \|\mathbf{A}(\Phi\hat{\mathbf{x}}(\mathbf{y}; \mathbf{\Lambda}) - \mathbf{y})\|_2^2 + \frac{2}{v} \langle \mathbf{A}^*\mathbf{\Pi}(\hat{\mathbf{x}}(\mathbf{y} + v\boldsymbol{\varepsilon}; \mathbf{\Lambda}) - \hat{\mathbf{x}}(\mathbf{y}; \mathbf{\Lambda})), \mathcal{S}\boldsymbol{\varepsilon} \rangle - \text{Tr}(\mathbf{A}\mathcal{S}\mathbf{A}^*), \quad (5.17)$$

est un estimateur asymptotiquement non biaisé de l'erreur quadratique $\mathbf{R}[\hat{\mathbf{x}}](\mathbf{\Lambda})$ quand $v \rightarrow 0$, ce qui signifie que

$$\lim_{v \rightarrow 0} \mathbb{E}_{\boldsymbol{\zeta}, \boldsymbol{\varepsilon}} \hat{\mathbf{R}}_{v,\boldsymbol{\varepsilon}}(\mathbf{y}; \mathbf{\Lambda}|\mathcal{S}) = \mathbf{R}[\hat{\mathbf{x}}](\mathbf{\Lambda}). \quad (5.18)$$

Démonstration. La démonstration du Théorème 5.2 est détaillée à l'Annexe C.2. \square

Remarque. L'utilisation d'une méthode de Monte Carlo est conseillée par [21, Section 2.3] afin de réduire la complexité du calcul effectif de l'estimateur SURE, en remplaçant le produit de deux matrices de taille $P \times P$, très coûteux, par le produit d'une matrice de taille $P \times P$, par un vecteur de taille P . Néanmoins, le produit de $\mathbf{S} \in \mathbb{R}^{P \times P}$ avec $\boldsymbol{\varepsilon} \in \mathbb{R}^P$, tout comme le produit de $\mathbf{A}^*\mathbf{\Pi} \in \mathbb{R}^{P \times N}$ avec

$\hat{\mathbf{x}}(\mathbf{y}; \Lambda) \in \mathbb{R}^N$, peuvent s'avérer extrêmement coûteux. Heureusement, nous verrons que, dans les problèmes de traitement de données (comme par exemple la segmentation de texture présentée au Chapitre 4), à la fois la matrice de covariance \mathcal{S} et l'opérateur linéaire \mathbf{A} , à travers l'opérateur de déformation Φ , bénéficient de propriétés de parcimonie suffisantes pour que ces calculs puissent être réalisés à un coût raisonnable.

5.3 SUGAR (*Stein's Unbiased Gradient estimator of the Risk*) généralisé

Notations. Dans cette section, nous nous intéressons au gradient de l'estimateur SURE généralisé par rapport aux hyperparamètres Λ . Son calcul repose sur le gradient par rapport aux hyperparamètres Λ de l'estimateur paramétrique différentiable $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$, qui est noté $\partial_\Lambda \hat{\mathbf{x}}(\mathbf{y}; \Lambda)$.

Pour un pas de différences finies, $v > 0$, suffisamment petit, il est raisonnable d'espérer que la solution du Problème (5.7), Λ^\dagger , minimisant l'erreur quadratique véritable, soit bien approximée par les hyperparamètres $\hat{\Lambda}_{v,\varepsilon}^\dagger$ minimisant l'erreur estimée :

$$\hat{\Lambda}_{v,\varepsilon}^\dagger(\mathbf{y}|\mathcal{S}) \in \underset{\Lambda \in \mathbb{R}^L}{\text{Argmin}} \hat{R}_{v,\varepsilon}(\mathbf{y}; \Lambda|\mathcal{S}). \quad (5.19)$$

Le Problème (5.19) pourrait alors être résolu par la mise en œuvre d'une recherche systématique sur une grille d'hyperparamètres suffisamment fine. Pour cela l'Algorithme 13 est appelé pour un ensemble d'hyperparamètres $\{\Lambda_1, \Lambda_2, \dots\}$ formant une grille de \mathbb{R}^L . Cependant, l'exploration d'une telle grille pourrait s'avérer très coûteuse en temps de calcul, notamment si l'évaluation de $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ est elle-même déjà coûteuse, ce qui est par exemple le cas lorsque $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ est un estimateur « séquentiel », i.e., obtenu à partir d'un schéma d'optimisation, comme celui de l'Algorithme 12. De plus, la taille d'une grille régulière dans \mathbb{R}^L , de pas fixé, augmente algébriquement avec L . Tout cela rend la recherche sur grille inenvisageable en pratique dès que $L > 2$.

Cette section s'intéresse à la sélection automatique d'hyperparamètres dans le cas d'un bruit *corrélé*. Un estimateur $\partial_\Lambda \hat{R}_{v,\varepsilon}(\mathbf{y}; \Lambda|\mathcal{S}) \in \mathbb{R}^L$, par différences finies et méthode de Monte Carlo, du gradient de l'erreur par rapport aux hyperparamètres Λ , est construit à l'Équation (5.20). De plus, le caractère asymptotiquement non biaisé de cet estimateur du gradient, $\partial_\Lambda \hat{R}_{v,\varepsilon}(\mathbf{y}; \Lambda|\mathcal{S})$, est démontré.

Nous fournissons à l'Algorithme 12, un exemple d'estimateur séquentiel, construit à partir d'un schéma primal-dual accéléré pour la résolution du Problème (5.3), accompagné de sa différentiation récursive qui permet de calculer à la fois $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ et son gradient par rapport à Λ , $\partial_\Lambda \hat{\mathbf{x}}(\mathbf{y}; \Lambda) \in \mathbb{R}^{N \times L}$.

En s'appuyant sur ce formalisme, la recherche sur grille très coûteuse peut être évitée, l'estimation de Λ^\dagger étant réalisée par une descente de quasi-Newton, décrite à l'Algorithme 15, qui minimise l'erreur quadratique estimée $\hat{R}_{v,\varepsilon}(\mathbf{y}; \Lambda|\mathcal{S})$, grâce à l'utilisation de son gradient $\partial_\Lambda \hat{R}_{v,\varepsilon}(\mathbf{y}; \Lambda|\mathcal{S})$.

5.3.1 Différentiation de l'estimateur SURE généralisé

Afin de construire un estimateur SUGAR généralisé du gradient de l'erreur quadratique par rapport aux hyperparamètres Λ , il convient tout d'abord de s'assurer de la régularité de l'estimateur $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ par rapport à Λ .

Hypothèse 5.5 (Lipschitzianité par rapport à Λ). Soit $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ un estimateur de $\bar{\mathbf{x}}$, qui dépend des observations \mathbf{y} , et est paramétré par Λ . La fonction $\Lambda \mapsto \hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ est uniformément L_2 -lipschitzienne, et sa constante de Lipschitz, L_2 , est indépendante de \mathbf{y} .

Remarque. Comme discuté dans [21], lorsque l'estimateur $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ peut s'exprimer comme l'opérateur proximal de la jauge d'un ensemble compact², l'Hypothèse 5.5 est vérifiée. C'est également le cas par transitivité si l'estimateur $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ s'écrit comme la composition de telles jauges. Donc, dans le cas où l'estimateur est défini à partir de (5.3) avec $\mathcal{G} = \mathcal{H}$, $\Phi = \mathbf{I}_{\mathcal{H}}$, et $\mathbf{U}_{\Lambda} = \lambda \mathbf{I}_{\mathcal{H}}$, pour tout $q \geq 1$ le caractère lipschitzien par rapport à Λ est assuré. De plus, pour la régularisation de Tikhonov, i.e., $q = 2$ et $\|\mathbf{U}_{\Lambda} \mathbf{x}\|_q^q = \lambda \|\mathbf{D} \mathbf{x}\|_2^2$ dans (5.3), si $\Phi = \mathbf{I}_{\mathcal{H}}$ et $\mathbf{D}^* \mathbf{D}$ est diagonalisable, avec des valeurs propres strictement positives et l'Hypothèse 5.5 est alors vérifiée. Néanmoins, excepté dans ces deux exemples connus et documentés, la vérification de l'Hypothèse 5.5 dans un cadre générique de moindres carrés pénalisés (5.3) est un problème difficile, qui fera l'objet d'un travail futur.

Proposition 5.1. *Soit \mathbf{y} des observations obéissant au Modèle (5.1), \mathbf{A} l'opérateur linéaire défini à l'Équation (5.9), et un estimateur $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ vérifiant les Hypothèses 5.1 (Gaussianité), 5.2 (Rang plein), 5.3 (Régularité et intégrabilité), 5.4 (Lipschitzianité par rapport à \mathbf{y}), et 5.5 (Lipschitzianité par rapport à Λ). Alors, l'estimateur SURE par différences finies et méthode de Monte Carlo, $\hat{R}_{v,\varepsilon}(\mathbf{y}; \Lambda | \mathcal{S})$, défini à l'Équation (5.17), est différentiable au sens faible à la fois par rapport aux observations \mathbf{y} et aux paramètres Λ , et son gradient par rapport à Λ , en tant qu'élément de \mathbb{R}^L , est donné par :*

$$\partial_{\Lambda} [\hat{R}_{v,\varepsilon}(\mathbf{y}; \Lambda | \mathcal{S})] \triangleq 2 (\mathbf{A} \Phi \partial_{\Lambda} \hat{\mathbf{x}}(\mathbf{y}; \Lambda))^* \mathbf{A} (\Phi \hat{\mathbf{x}}(\mathbf{y}; \Lambda) - \mathbf{y}) + \frac{2}{v} (\mathbf{A}^* \Pi (\partial_{\Lambda} \hat{\mathbf{x}}(\mathbf{y} + v \boldsymbol{\varepsilon}; \Lambda) - \partial_{\Lambda} \hat{\mathbf{x}}(\mathbf{y}; \Lambda)))^* \mathcal{S} \boldsymbol{\varepsilon}. \quad (5.20)$$

Démonstration. L'estimateur SURE par différences finies et méthode de Monte Carlo, $\hat{R}_{v,\varepsilon}(\mathbf{y}; \Lambda | \mathcal{S})$, défini par la Formule (5.17), est une combinaison de fonctions continues et différentiables au sens faible, à la fois par rapport aux observations \mathbf{y} et aux paramètres Λ , composées avec des opérateurs linéaires (bornés). Par conséquent, $\hat{R}_{v,\varepsilon}(\mathbf{y}; \Lambda | \mathcal{S})$ est également continu et différentiable au sens faible. De plus, les règles usuelles de dérivation s'appliquent, et conduisent à l'expression (5.20) de l'estimateur SUGAR généralisé du gradient de l'erreur quadratique. \square

Théorème 5.3. *Soit \mathbf{y} des observations obéissant au Modèle (5.1), \mathbf{A} l'opérateur linéaire défini à l'Équation (5.9), et un estimateur $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ vérifiant les Hypothèses 5.1 (Gaussianité), 5.2 (Rang plein), 5.3 (Régularité et intégrabilité), 5.4 (Lipschitzianité par rapport à \mathbf{y}), et 5.5 (Lipschitzianité par rapport à Λ). Alors, l'estimateur SUGAR par différences finies et méthode de Monte Carlo généralisé, $\partial_{\Lambda} \hat{R}_{v,\varepsilon}(\mathbf{y}; \Lambda)$, défini à l'Équation (5.20), est un estimateur asymptotiquement non biaisé du gradient de l'erreur quadratique lorsque $v \rightarrow 0$, c'est-à-dire*

$$\partial_{\Lambda} \mathbf{R}[\hat{\mathbf{x}}](\Lambda) = \lim_{v \rightarrow 0} \mathbb{E}_{\boldsymbol{\zeta}, \boldsymbol{\varepsilon}} \partial_{\Lambda} \hat{R}_{v,\varepsilon}(\mathbf{y}; \Lambda | \mathcal{S}). \quad (5.21)$$

Démonstration. La démonstration du Théorème 5.3 est détaillée en Annexe C.3. \square

Remarque. L'estimateur du gradient de l'erreur quadratique par différences finies et méthode de Monte Carlo, $\partial_{\Lambda} \hat{R}_{v,\varepsilon}(\mathbf{y}; \Lambda | \mathcal{S})$, défini à l'Équation (5.20), fait intervenir la matrice Jacobienne $\partial_{\Lambda} \hat{\mathbf{x}}(\mathbf{y}; \Lambda) \in \mathbb{R}^{N \times L}$, qui peut être une matrice de très grande taille, ce qui induit des difficultés dans sa manipulation pratique. Cependant, dans la plupart des applications, les hyperparamètres forment un vecteur $\Lambda \in \mathbb{R}^L$, de faible dimension, $L = \mathcal{O}(1) \ll N$. Ainsi, il est raisonnable d'espérer pouvoir stocker et manipuler la matrice Jacobienne, $\partial_{\Lambda} \hat{\mathbf{x}}(\mathbf{y}; \Lambda) \in \mathbb{R}^{N \times L}$, à des coûts en terme de mémoire et de capacité de calcul du même ordre de grandeur que les coûts induits par le stockage et la manipulation de $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ (voir la Section 5.3.2).

5.3.2 Estimateurs séquentiels et différentiation récursive

L'évaluation de $\partial_{\Lambda} \hat{R}_{v,\varepsilon}(\mathbf{y}; \Lambda)$ à partir de la formule de l'Équation (5.20) nécessite de connaître la Jacobienne $\partial_{\Lambda} \hat{\mathbf{x}}(\mathbf{y}; \Lambda)$. Cependant, lorsqu'aucune expression explicite de l'estimateur $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$

2. Pour un ouvert non vide, convexe, $\mathcal{C} \subset \mathcal{G}$, contenant $\mathbf{0}_{\mathcal{G}}$, la « jauge » de \mathcal{C} est définie comme la fonction $\mathbf{y} \mapsto \gamma_{\mathcal{C}}(\mathbf{y}) \triangleq \inf\{\omega > 0 \mid \mathbf{y} \in \omega \mathcal{C}\}$.

n'est disponible, calculer le gradient $\partial_{\Lambda} \hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ peut s'avérer une tâche complexe. Une large classe d'estimateurs $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ ne bénéficiant pas d'expression explicite, est celle des estimateurs obtenus comme la limite d'une suite

$$\hat{\mathbf{x}}(\mathbf{y}; \Lambda) = \lim_{k \rightarrow \infty} \mathbf{x}^{[k]}(\mathbf{y}; \Lambda), \quad (5.22)$$

par exemple lorsque $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ est défini comme la solution d'un problème de minimisation tel que (5.3). On parle alors d'estimateur *séquentiel*.

Dans le cas où $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ est un estimateur séquentiel, étant donnée une observation \mathbf{y} , il est uniquement possible d'échantillonner la fonction $\Lambda \mapsto \hat{\mathbf{x}}(\mathbf{y}; \Lambda)$, pour un ensemble *discret* d'hyperparamètres $\{\Lambda_1, \Lambda_2, \dots\}$, en faisant mettre en œuvre l'algorithme de minimisation pour chaque paramètre $\Lambda_1, \Lambda_2, \dots$. Une stratégie de calcul du gradient par différences finies, comme celle mise en œuvre pour le calcul de la Jacobienne de $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$, décrite à l'Équation (5.15), n'est pas suffisamment stable pour être incluse dans un schéma de minimisation. En effet, c'est un fait avéré en traitement du signal qu'un estimateur de la dérivée d'une fonction construit à partir d'un jeu d'échantillons de cette fonction manque de robustesse. Contrairement au cas de la différentiation par rapport à la variable de grande taille $\mathbf{y} \in \mathbb{R}^P$, il est possible d'envisager l'utilisation d'outils plus sophistiqués pour la dérivation par rapport à la variable $\Lambda \in \mathbb{R}^L$, de « petite » taille $L = \mathcal{O}(1)$. En supposant que les itérations de l'algorithme permettant de calculer $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ sont suffisamment régulières, une stratégie de différentiation récursive [21] fournit une suite de Jacobiennes $\partial_{\Lambda} \hat{\mathbf{x}}^{[t]}(\mathbf{y}; \Lambda)$, au moyen de différentiations « en chaîne », dont le principe est présenté à la Proposition 5.2.

Proposition 5.2. [28, Section 4.2.2] Soit $\Psi : \mathbb{R}^{N \times L} \rightarrow \mathbb{R}^N$ une fonction différentiable des variables $(\mathbf{x}, \Lambda) \in \mathcal{H} \times \mathbb{R}^L$, et $(\mathbf{x}^{[t]})_{k \in \mathbb{N}}$ l'estimateur séquentiel défini à partir des itérations de la forme

$$\mathbf{x}^{[t+1]} = \Psi(\mathbf{x}^{[t]}; \Lambda). \quad (5.23)$$

Le gradient de $\mathbf{x}^{[t]}$ par rapport à Λ peut être calculé en utilisant la règle de différentiation « en chaîne » suivante :

$$\partial_{\Lambda} \mathbf{x}^{[t+1]} = \partial_{\mathbf{x}} \Psi(\mathbf{x}^{[t]}; \Lambda) [\partial_{\Lambda} \mathbf{x}^{[t]}] + \partial_{\Lambda} \Psi(\mathbf{x}^{[t]}; \Lambda), \quad (5.24)$$

où $\partial_{\mathbf{x}} \Psi(\mathbf{x}; \Lambda) [\boldsymbol{\delta}]$ désigne la différentielle de Ψ par rapport à la variable \mathbf{x} appliquée au vecteur $\boldsymbol{\delta}$, et $\partial_{\Lambda} \Psi(\mathbf{x}; \Lambda)$ le gradient de Ψ par rapport aux paramètres Λ . En toute généralité, la différentiabilité de Ψ doit être comprise au sens faible.

Afin d'illustrer ce principe, nous allons nous concentrer sur la résolution du Problème (5.3), pour laquelle les algorithmes proximaux [45, 17, 4] sont plébiscités. Parmi ces algorithmes, nous avons choisi le schéma primal-dual proposé dans [14, Algorithme 2], tirant parti des expressions explicites des opérateurs proximaux à la fois du terme de fidélité aux données et de la pénalisation³. L'algorithme de Chambolle-Pock, particularisé au Problème (5.3), est détaillé à l'Algorithme 12. En outre, Φ étant de rang plein en vertu de l'Hypothèse 5.2, en notant $\text{Sp}(\Phi^* \Phi)$ le spectre de $\Phi^* \Phi$, $\mu = \min \text{Sp}(\Phi^* \Phi)$ est strictement positif. Par conséquent, le terme d'attache aux données de (5.3) s'avère μ -fortement convexe, et l'algorithme primal-dual peut être accéléré grâce à l'étape de mise à jour des pas de descente (5.32) de l'Algorithme 12 [14, Algorithme 2]. La stratégie de différentiation récursive fournissant $\partial_{\Lambda} \mathbf{x}^{[t]}(\mathbf{y}; \Lambda)$ est présentée dans la seconde partie de l'Algorithme 12. Le lecteur intéressé trouvera plusieurs schémas détaillés de différentiation récursive dans [21].

Remarque. Deux cas particuliers de différentiation « en chaîne » sont utilisés à l'Algorithme 12, et plus généralement pour la différentiation récursive des schémas proximaux :

3. voir <http://proximity-operator.net> pour de nombreuses expressions explicites d'opérateurs proximaux.

- (i) *L'opérateur linéaire* $\Psi(\mathbf{x}; \Lambda) \triangleq \mathbf{U}_\Lambda \mathbf{x}$. En supposant que $(\mathbf{x} \mapsto \mathbf{U}_\Lambda \mathbf{x})_\Lambda$ est une famille d'opérateurs linéaires, avec une paramétrisation différentiable par Λ , la règle de dérivation « en chaîne »

$$\partial_\Lambda \mathbf{x}^{[t+1]} = \mathbf{U}_\Lambda \partial_\Lambda \mathbf{x}^{[t]} + (\partial_\Lambda \mathbf{U}_\Lambda) \mathbf{x}^{[t]}, \quad (5.25)$$

puisque la différentielle d'un opérateur linéaire \mathbf{U}_Λ par rapport à \mathbf{x} est lui-même. Cette règle de dérivation « en chaîne » des opérateurs linéaires est appliquée aux étapes (5.34) et (5.36), de l'Algorithme 12.

- (ii) *L'opérateur proximal* $\Psi(\mathbf{x}; \Lambda) \triangleq \text{prox}_{\tau \|\cdot\|_{2,1}}(\mathbf{x})$. L'opérateur proximal en exemple étant indépendant de Λ , la règle de différentiation « en chaîne » se simplifie en

$$\partial_\Lambda \mathbf{x}^{[t+1]} = \partial_{\mathbf{x}} \text{prox}_{\tau \|\cdot\|_{2,1}}(\mathbf{x}^{[t]}) [\partial_\Lambda \mathbf{x}^{[t]}] \quad (5.26)$$

où la dérivée de $\text{prox}_{\tau \|\cdot\|_{2,1}}$, le seuillage mixte (*soft-thresholding*) $\ell_2 - \ell_1$, par rapport à $\mathbf{x} = (x_1, x_2)$, et appliqué à $\boldsymbol{\delta} = (\delta_1, \delta_2)$ possède l'expression explicite suivante :

$$\partial_{\mathbf{x}} \text{prox}_{\tau \|\cdot\|_{2,1}}(\mathbf{x}) [\boldsymbol{\delta}] = \begin{cases} \mathbf{0} & \text{if } \|\mathbf{x}\|_2 \leq \tau \\ \boldsymbol{\delta} - \frac{\tau}{\|\mathbf{x}\|_2} \left(\boldsymbol{\delta} - \frac{\langle \boldsymbol{\delta}, \mathbf{x} \rangle}{\|\mathbf{x}\|_2^2} \mathbf{x} \right) & \text{else.} \end{cases} \quad (5.27)$$

La différentiation « en chaîne » de l'opérateur proximal $\text{prox}_{\tau \|\cdot\|_{2,1}}$ par rapport à la variable \mathbf{x} est utilisée aux étapes (5.35) et (5.37) de l'Algorithme 12.

Définition 5.2 (SURE et SUGAR généralisés pour les estimateurs séquentiels). Soit $\hat{\mathbf{x}}(\boldsymbol{\ell}; \Lambda)$ un estimateur séquentiel au sens de la Formule (5.22). L'estimée de l'erreur associée, $\hat{R}_{v,\varepsilon}(\mathbf{y}; \Lambda | \mathcal{S})$, et l'estimée de son gradient, $\partial_\Lambda \hat{R}_{v,\varepsilon}(\mathbf{y}; \Lambda | \mathcal{S})$ sont calculées à partir d'un algorithme différencié récursivement, tel que l'Algorithme 12, qui est exécuté deux fois : une première fois à partir des observations \mathbf{y} , puis une seconde fois à partir des observations perturbées $\mathbf{y} + v\boldsymbol{\varepsilon}$. Ensuite, l'estimateur SURE généralisé est calculé en appliquant la Formule (5.17), tandis que l'application de la Formule (5.20) fournit l'estimateur SUGAR généralisé. Ces étapes sont résumées aux Algorithmes 13, pour l'estimation de $\hat{R}_{v,\varepsilon}(\mathbf{y}; \Lambda | \mathcal{S})$ seul, et 14, pour l'estimation de $\hat{R}_{v,\varepsilon}(\mathbf{y}; \Lambda | \mathcal{S})$ et de son gradient $\partial_\Lambda \hat{R}_{v,\varepsilon}(\mathbf{y}; \Lambda | \mathcal{S})$ mises sous la forme de deux routines, appelées respectivement « SURE » et « SUGAR ».

5.3.3 Minimisation automatique de l'erreur quadratique

La dimension de $\Lambda \in \mathbb{R}^L$ étant raisonnablement petite (voir Remarque 5.3.1), le Problème (5.19) est résolu numériquement grâce à un algorithme de descente de quasi-Newton, utilisant l'estimée du gradient de l'erreur, $\partial_\Lambda \hat{R}_{v,\varepsilon}(\mathbf{y}; \Lambda | \mathcal{S})$, fournie par le Théorème 5.3.

Une version simplifiée de l'algorithme de descente de quasi-Newton, particularisé pour le Problème (5.19), est proposée à l'Algorithme 15. Cet algorithme génère une suite, $(\Lambda^{[m]})_{m \in \mathbb{N}}$, convergeant vers un minimiseur de $\hat{R}_{v,\varepsilon}(\mathbf{y}; \Lambda | \mathcal{S})$. Tout d'abord, une descente de gradient est réalisée (5.51), selon une direction $\mathbf{d}^{[m]}$ obtenue comme le produit d'une approximation de type BFGS de l'inverse de la matrice hessienne de l'erreur par rapport à Λ , $\mathbf{H}^{[m]}$ et du gradient $\partial_\Lambda \hat{R}_{v,\varepsilon}(\mathbf{y}; \Lambda | \mathcal{S})$, calculé à partir de la routine SUGAR de l'Algorithme 14. Le pas de descente $\alpha^{[m]}$ est déterminé grâce à une recherche unidirectionnelle [44, Chapitre 3], effectuée à l'étape (5.50) de l'Algorithme 15, qui est stoppé dès que les conditions de Wolfe sont satisfaites [19]. Enfin, l'approximation de la matrice hessienne inverse, $\mathbf{H}^{[m]}$, est mise à jour selon la Définition 5.3.

Remarque. La recherche unidirectionnelle, i.e., l'étape (5.50) de l'Algorithme 15, est la plus coûteuse en temps de calcul. En effet, les routines SURE et SUGAR sont appelées pour plusieurs paramètres de la forme $\Lambda^{[m]} + \alpha \mathbf{d}^{[m]}$, chaque appel requérant d'exécuter l'algorithme primal-dual différencié 12 deux fois : une fois avec l'observation \mathbf{y} et une fois avec l'observation perturbée $\mathbf{y} + v\boldsymbol{\varepsilon}$.

Algorithme 12 Schéma primal-dual accéléré pour la résolution du Problème (5.3) avec différenciation récursive par rapport aux paramètres Λ .

Entrée : Observations \mathbf{y} , paramètres de régularisation Λ

Paramètres : Module de forte-convexité de l'attache aux données $\mu = \min \text{Sp}(\Phi^* \Phi)$

Initialisation : $\mathbf{x}^{[0]} \in \mathcal{H}$ (variable primale)

$\mathbf{w}^{[0]} \leftarrow \mathbf{x}^{[0]}$ (variable primale auxiliaire)

$\mathbf{z}^{[0]} \in \mathcal{Q}$ (variable duale)

Calculer $\partial_{\Lambda} \mathbf{x}^{[0]}$, $\partial_{\Lambda} \mathbf{w}^{[0]}$ et $\partial_{\Lambda} \mathbf{z}^{[0]}$ (jacobiennes par rapport à Λ)

Pas de descente $\boldsymbol{\tau}^{[0]} = (\tau_1^{[0]}, \tau_2^{[0]})$ tels que $\tau_1^{[0]} \tau_2^{[0]} \|\mathbf{U}_{\Lambda}\|^2 < 1$

for $m = 1$ **to** T_{\max} **do**

{Primal-dual accéléré}

$$\tilde{\mathbf{z}}^{[t]} = \mathbf{z}^{[t]} + \tau_1^{[t]} \mathbf{U}_{\Lambda} \mathbf{w}^{[t]} \quad (5.28)$$

$$\mathbf{z}^{[t+1]} = \text{prox}_{\tau_1^{[t]}(\|\cdot\|_q^*)}(\tilde{\mathbf{z}}^{[t]}) \quad (5.29)$$

$$\tilde{\mathbf{x}}^{[t]} = \mathbf{x}^{[t]} - \tau_2^{[t]} \mathbf{U}_{\Lambda}^* \mathbf{z}^{[t+1]} \quad (5.30)$$

$$\mathbf{x}^{[t+1]} = \text{prox}_{\tau_2^{[t]} \|\mathbf{y} - \Phi \cdot\|_2^2}(\tilde{\mathbf{x}}^{[t]}) \quad (5.31)$$

$$\theta^{[t]} = \left(1 + 2\mu \tau_2^{[t]}\right)^{-\frac{1}{2}}, \quad (5.32)$$

$$\begin{aligned} \tau_1^{[t+1]} &= \tau_1^{[t]} / \theta^{[t]}, \quad \tau_2^{[t+1]} = \theta^{[t]} \tau_2^{[t]} \\ \mathbf{w}^{[t+1]} &= \mathbf{x}^{[t]} + \theta^{[t]} (\mathbf{x}^{[t+1]} - \mathbf{x}^{[t]}) \end{aligned} \quad (5.33)$$

{Primal-dual accéléré différentié}

$$\partial_{\Lambda} \tilde{\mathbf{z}}^{[t]} = \partial_{\Lambda} \mathbf{z}^{[t]} + \tau_1^{[t]} \mathbf{U}_{\Lambda} \partial_{\Lambda} \mathbf{w}^{[t]} + \tau_1^{[t]} \partial_{\Lambda} \mathbf{U}_{\Lambda} \mathbf{w}^{[t]} \quad (5.34)$$

$$\partial_{\Lambda} \mathbf{z}^{[t+1]} = \partial_{\tilde{\mathbf{z}}} \text{prox}_{\tau_1^{[t]}(\|\cdot\|_q^*)}(\tilde{\mathbf{z}}^{[t]}) [\partial_{\Lambda} \tilde{\mathbf{z}}^{[t]}] \quad (5.35)$$

$$\partial_{\Lambda} \tilde{\mathbf{x}}^{[t]} = \partial_{\Lambda} \mathbf{x}^{[t]} - \tau_2^{[t]} \mathbf{U}_{\Lambda}^* \partial_{\Lambda} \mathbf{z}^{[t+1]} - \tau_2^{[t]} \partial_{\Lambda} \mathbf{U}_{\Lambda} \mathbf{z}^{[t+1]} \quad (5.36)$$

$$\partial_{\Lambda} \mathbf{x}^{[t+1]} = \partial_{\tilde{\mathbf{x}}} \text{prox}_{\tau_2^{[t]} \|\mathbf{y} - \Phi \cdot\|_2^2}(\tilde{\mathbf{x}}^{[t]}) [\partial_{\Lambda} \tilde{\mathbf{x}}^{[t]}] \quad (5.37)$$

$$\partial_{\Lambda} \mathbf{w}^{[t+1]} = \partial_{\Lambda} \mathbf{x}^{[t]} + \theta^{[t]} (\partial_{\Lambda} \mathbf{x}^{[t+1]} - \partial_{\Lambda} \mathbf{x}^{[t]}) \quad (5.38)$$

end for

Sortie : Solution *en temps fini* du Problème (5.3) $\hat{\mathbf{x}}(\mathbf{y}; \Lambda) \triangleq \hat{\mathbf{x}}^{[T_{\max}]}$

Jacobienne par rapport aux paramètres *en temps fini* $\partial_{\Lambda} \hat{\mathbf{x}}(\mathbf{y}; \Lambda) \triangleq \partial_{\Lambda} \hat{\mathbf{x}}^{[T_{\max}]}$

Définition 5.3 (Mise à jour de Broyden, Fletcher, Goldfarb, et Shanno (BFGS)). Soit $\mathbf{d}^{[m]}$ la direction de descente et $\mathbf{u}^{[m]}$ l'accroissement du gradient à l'itération m , la mise à jour BFGS de l'approximation de l'inverse de la hessienne, $\mathbf{H}^{[m]}$, s'écrit

$$\mathbf{H}^{[m+1]} = \left(\mathbf{I}_L - \frac{\mathbf{d}^{[m]} (\mathbf{u}^{[m]})^{\top}}{(\mathbf{u}^{[m]})^{\top} \mathbf{d}^{[m]}} \right) \mathbf{H}^{[m]} \left(\mathbf{I}_L - \frac{\mathbf{u}^{[m]} (\mathbf{d}^{[m]})^{\top}}{(\mathbf{u}^{[m]})^{\top} \mathbf{d}^{[m]}} \right) + \alpha^{[m]} \frac{\mathbf{d}^{[m]} (\mathbf{d}^{[m]})^{\top}}{(\mathbf{u}^{[m]})^{\top} \mathbf{d}^{[m]}}. \quad (5.55)$$

cette étape constitue une routine, appelée « BFGS », définie par

$$\mathbf{H}^{[m+1]} \triangleq \text{BFGS}(\mathbf{H}^{[m]}, \mathbf{d}^{[m]}, \mathbf{u}^{[m]}). \quad (5.56)$$

Pour une discussion détaillée de l'implémentation de l'algorithme de quasi-Newton, des stratégies BFGS utilisant peu de mémoire, de la gestion des contraintes de bords, comme la positivité des paramètres par exemple, et plusieurs considérations algorithmiques, le lecteur pourra se référer à [44, Sections 8.1, 9.1 et 9.2] ou [12, 19].

Algorithme 13 SURE généralisé pour un estimateur séquentiel $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$, en vue d'une recherche sur une grille.

Entrée : Observations \mathbf{y} , paramètres de régularisation Λ , matrice de covariance \mathcal{S}

Paramètres : Vecteur de Monte Carlo $\boldsymbol{\varepsilon} \in \mathbb{R}^P \sim \mathcal{N}(\mathbf{0}_P, \mathbf{I}_P)$, pas de différences finies $\nu > 0$

{Solution de (5.3) à partir de l'Algorithme 12}

$$\hat{\mathbf{x}}(\mathbf{y}; \Lambda) = \text{PD}(\mathbf{y}, \Lambda) \quad (5.39)$$

$$\hat{\mathbf{x}}(\mathbf{y} + \nu \boldsymbol{\varepsilon}; \Lambda) = \text{PD}(\mathbf{y} + \nu \boldsymbol{\varepsilon}, \Lambda) \quad (5.40)$$

{SURE (5.17) par différences finies et stratégie Monte Carlo}

$$\hat{R}_{\nu, \boldsymbol{\varepsilon}}(\mathbf{y}; \Lambda | \mathcal{S}) = \|\mathbf{A}(\Phi \hat{\mathbf{x}}(\mathbf{y}; \Lambda) - \mathbf{y})\|_2^2 + \frac{2}{\nu} \langle \mathbf{A}^* \boldsymbol{\Pi}(\hat{\mathbf{x}}(\mathbf{y} + \nu \boldsymbol{\varepsilon}; \Lambda) - \hat{\mathbf{x}}(\mathbf{y}; \Lambda)), \mathcal{S} \boldsymbol{\varepsilon} \rangle - \text{Tr}(\mathbf{A} \mathcal{S} \mathbf{A}^*) \quad (5.41)$$

Sortie : Estimée de l'erreur quadratique $\hat{R}_{\nu, \boldsymbol{\varepsilon}}(\mathbf{y}; \Lambda | \mathcal{S})$

Routines :

$$\hat{R}_{\nu, \boldsymbol{\varepsilon}}(\mathbf{y}; \Lambda | \mathcal{S}) = \text{SURE}(\mathbf{y}, \Lambda, \mathcal{S}, \nu, \boldsymbol{\varepsilon})$$

Les conditions de convergence des algorithmes de quasi-Newton portent sur le comportement des dérivées secondes de la fonction objectif [44, Section 8.4]. Or, la plupart du temps, aucune information sur la dérivée seconde par rapport aux hyperparamètres des estimateurs séquentiel n'est connue, et par conséquent on ne dispose d'aucune information sur la différentiabilité à l'ordre deux de l'estimateur SURE généralisé. La convergence de l'Algorithme 15 sera donc vérifiée numériquement à la Section 5.5. En outre, par construction, les algorithmes de quasi-Newton sont très sensibles à l'initialisation. Un soin particulier doit donc être apporté, à la fois au choix des hyperparamètres initiaux, et à l'initialisation de la matrice \mathbf{H} . Ces points seront détaillés à la Section 5.5.2.4, pour le problème de segmentation de textures décrit au Chapitre 4.

Remarque. Étant donné un estimateur paramétrique $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$, éventuellement obtenu par un autre schéma numérique que celui de l'Algorithme 12, les Algorithmes 14 et 15 peuvent être utilisés, à condition de disposer d'une routine équivalente à ∂PD , calculant $\partial_{\Lambda} \hat{\mathbf{x}}(\mathbf{y}; \Lambda)$.

Algorithme 14 SURE et SUGAR généralisés pour un estimateur séquentiel $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$, en vue d'un réglage automatique par algorithme de quasi-Newton.

Entrée : Observations \mathbf{y} , paramètres de régularisation Λ , matrice de covariance \mathcal{S}

Paramètres : Vecteur de Monte Carlo $\boldsymbol{\varepsilon} \in \mathbb{R}^P \sim \mathcal{N}(\mathbf{0}_P, \mathbf{I}_P)$, pas de différences finies $\nu > 0$

{Solution de (5.3) à partir de l'Algorithme 12}

$$\hat{\mathbf{x}}(\mathbf{y}; \Lambda) = \text{PD}(\mathbf{y}, \Lambda) \quad (5.42)$$

$$\hat{\mathbf{x}}(\mathbf{y} + \nu \boldsymbol{\varepsilon}; \Lambda) = \text{PD}(\mathbf{y} + \nu \boldsymbol{\varepsilon}, \Lambda) \quad (5.43)$$

{SURE (5.17) par différences finies et stratégie Monte Carlo}

$$\hat{R}_{\nu, \boldsymbol{\varepsilon}}(\mathbf{y}; \Lambda | \mathcal{S}) = \|\mathbf{A}(\Phi \hat{\mathbf{x}}(\mathbf{y}; \Lambda) - \mathbf{y})\|_2^2 + \frac{2}{\nu} \langle \mathbf{A}^* \boldsymbol{\Pi}(\hat{\mathbf{x}}(\mathbf{y} + \nu \boldsymbol{\varepsilon}; \Lambda) - \hat{\mathbf{x}}(\mathbf{y}; \Lambda)), \mathcal{S} \boldsymbol{\varepsilon} \rangle - \text{Tr}(\mathbf{A} \mathcal{S} \mathbf{A}^*) \quad (5.44)$$

{Solution de (5.3) et sa dérivée Λ à partir de l'Algorithme 12}

$$(\hat{\mathbf{x}}(\mathbf{y}; \Lambda), \partial_{\Lambda} \hat{\mathbf{x}}(\mathbf{y}; \Lambda)) = \partial \text{PD}(\mathbf{y}, \Lambda) \quad (5.45)$$

$$(\hat{\mathbf{x}}(\mathbf{y} + \nu \boldsymbol{\varepsilon}; \Lambda), \partial_{\Lambda} \hat{\mathbf{x}}(\mathbf{y} + \nu \boldsymbol{\varepsilon}; \Lambda)) = \partial \text{PD}(\mathbf{y} + \nu \boldsymbol{\varepsilon}, \Lambda) \quad (5.46)$$

{Estimateurs (5.17) et (5.20) par différences finies et stratégie Monte Carlo}

$$\hat{R}_{\nu, \boldsymbol{\varepsilon}}(\mathbf{y}; \Lambda | \mathcal{S}) = \|\mathbf{A}(\Phi \hat{\mathbf{x}}(\mathbf{y}; \Lambda) - \mathbf{y})\|_2^2 + \frac{2}{\nu} \langle \mathbf{A}^* \boldsymbol{\Pi}(\hat{\mathbf{x}}(\mathbf{y} + \nu \boldsymbol{\varepsilon}; \Lambda) - \hat{\mathbf{x}}(\mathbf{y}; \Lambda)), \mathcal{S} \boldsymbol{\varepsilon} \rangle - \text{Tr}(\mathbf{A} \mathcal{S} \mathbf{A}^*) \quad (5.47)$$

$$\partial_{\Lambda} \hat{R}_{\nu, \boldsymbol{\varepsilon}}(\mathbf{y}; \Lambda | \mathcal{S}) = 2(\mathbf{A} \Phi \partial_{\Lambda} \hat{\mathbf{x}}(\mathbf{y}; \Lambda))^* \mathbf{A}(\Phi \hat{\mathbf{x}}(\mathbf{y}; \Lambda) - \mathbf{y}) + \frac{2}{\nu} (\mathbf{A}^* \boldsymbol{\Pi}(\partial_{\Lambda} \hat{\mathbf{x}}(\mathbf{y} + \nu \boldsymbol{\varepsilon}; \Lambda) - \partial_{\Lambda} \hat{\mathbf{x}}(\mathbf{y}; \Lambda)))^* \mathcal{S} \boldsymbol{\varepsilon} \quad (5.48)$$

Sortie : Estimée de l'erreur quadratique $\hat{R}_{\nu, \boldsymbol{\varepsilon}}(\mathbf{y}; \Lambda | \mathcal{S})$

Estimée du gradient de l'erreur $\partial_{\Lambda} \hat{R}_{\nu, \boldsymbol{\varepsilon}}(\mathbf{y}; \Lambda | \mathcal{S})$

Routines :

$$\begin{aligned} \hat{R}_{\nu, \boldsymbol{\varepsilon}}(\mathbf{y}; \Lambda | \mathcal{S}) &= \text{SURE}(\mathbf{y}, \Lambda, \mathcal{S}, \nu, \boldsymbol{\varepsilon}) \\ \partial_{\Lambda} \hat{R}_{\nu, \boldsymbol{\varepsilon}}(\mathbf{y}; \Lambda | \mathcal{S}) &= \text{SUGAR}(\mathbf{y}, \Lambda, \mathcal{S}, \nu, \boldsymbol{\varepsilon}) \end{aligned}$$

Algorithme 15 Sélection automatique des hyperparamètres minimisant l'erreur quadratique.

Entrée : Observations \mathbf{y} , hyperparamètres Λ , matrice de covariance \mathcal{S}

Paramètres : Vecteur de Monte Carlo $\boldsymbol{\epsilon} \in \mathbb{R}^P \sim \mathcal{N}(\mathbf{0}_P, \mathbf{I}_P)$, pas de différences finies $\nu > 0$

Initialisation : Choisir $\Lambda^{[0]} \in \mathbb{R}^L$, $\mathbf{H}^{[0]} \in \mathbb{R}^{L \times L}$

$\partial_\Lambda \widehat{\mathbf{R}}^{[0]} \leftarrow \text{SUGAR}(\mathbf{y}, \Lambda^{[0]}, \mathcal{S}, \nu, \boldsymbol{\epsilon})$

for $t = 0$ **to** $M_{\max} - 1$ **do**

{Direction de descente à partir du gradient de l'erreur :}

$$\mathbf{d}^{[m]} = -\mathbf{H}^{[m]} \partial_\Lambda \widehat{\mathbf{R}}^{[m]} \quad (5.49)$$

{Recherche unidirectionnelle du pas de descente :}

$$\alpha^{[m]} \in \underset{\alpha \in \mathbb{R}}{\text{Argmin}} \widehat{\mathbf{R}}(\Lambda^{[m]} + \alpha \mathbf{d}^{[m]}) \quad \text{avec } \widehat{\mathbf{R}}(\Lambda) = \text{SURE}(\mathbf{y}, \Lambda, \mathcal{S}, \nu, \boldsymbol{\epsilon}) \quad (5.50)$$

{Descente de quasi-Newton sur Λ :}

$$\Lambda^{[m+1]} = \Lambda^{[m]} + \alpha^{[m]} \mathbf{d}^{[m]} \quad (5.51)$$

{Mise à jour du gradient :}

$$\partial_\Lambda \widehat{\mathbf{R}}^{[m+1]} = \text{SUGAR}(\mathbf{y}, \Lambda^{[m+1]}, \mathcal{S}, \nu, \boldsymbol{\epsilon}) \quad (5.52)$$

{Accroissement du gradient :}

$$\mathbf{u}^{[m]} = \partial_\Lambda \widehat{\mathbf{R}}^{[m+1]} - \partial_\Lambda \widehat{\mathbf{R}}^{[m]} \quad (5.53)$$

{Mise à jour BFGS update de l'inverse de la hessienne (5.55) :}

$$\mathbf{H}^{[m+1]} = \text{BFGS}(\mathbf{H}^{[m]}, \mathbf{d}^{[m]}, \mathbf{u}^{[m]}) \quad (5.54)$$

end for

Sortie : Solution *en temps fini* du Problème (5.19) $\widehat{\Lambda}_{\nu, \boldsymbol{\epsilon}}^{\text{BFGS}}(\mathbf{y}|\mathcal{S}) \triangleq \Lambda^{[M_{\max}]}$

Estimée avec sélection automatique de Λ $\widehat{\mathbf{x}}_{\nu, \boldsymbol{\epsilon}}^{\text{BFGS}}(\mathbf{y}|\mathcal{S}) \triangleq \text{PD}(\mathbf{y}, \Lambda^{[M_{\max}]})$

5.4 Sélection d'hyperparamètres pour la segmentation de textures

5.4.1 Formulation en terme du Modèle (5.1)

Nous considérons, pour le problème de segmentation de textures du Chapitre 4, que les observations sont les logarithmes des coefficients d'ondelettes dominants, c'est-à-dire

$$\mathbf{y} = \boldsymbol{\ell} = \log_2 \boldsymbol{\mathcal{L}} \in \mathbb{R}^P, \quad \text{avec } P = N_j N_1 N_2 \quad (5.57)$$

calculés sur l'image X de $N_1 \times N_2$ pixels à segmenter, sur un nombre d'échelles N_j et stockés sous forme d'un *vecteur* dans lequel les ℓ_j sont concaténés, octave par octave et selon l'ordre lexicographique pour les pixels. Le comportement log-linéaire, décrit à l'Équation (4.1), est mis sous la forme du Modèle (5.1) en introduisant la quantité d'intérêt :

$$\bar{\mathbf{x}} = \begin{pmatrix} \bar{\mathbf{h}} \\ \bar{\mathbf{v}} \end{pmatrix} \in \mathbb{R}^N, N = 2N_1 N_2, \quad (5.58)$$

qui correspond aux « véritables » régularité et variance locales, représentées sous forme vectorisée. L'opérateur linéaire de déformation, Φ , est l'opérateur défini à l'Équation (4.3), et le terme de bruit dans la Relation (4.1) est écrit explicitement :

$$\ell_j(\underline{n}) = \bar{v}(\underline{n}) + j\bar{h}(\underline{n}) + \zeta_j(\underline{n}), \quad \text{quand } 2^j \simeq 0. \quad (5.59)$$

Pour une large classe de textures, le bruit $\zeta_j(\underline{n})$ peut être considéré comme un bruit gaussien de moyenne nulle [63, Section 3.2]. Ainsi, les logarithmes des coefficients dominants $\boldsymbol{\ell}$ suivent le modèle (5.8) et l'Hypothèse 5.1 est vérifiée.

La Proposition 4.6 assure alors que l'opérateur linéaire $\Phi^* \Phi$ est inversible, ou de manière équivalente que Φ est de rang plein, i.e., que l'Hypothèse 5.2 est vérifiée. De plus, l'expression de l'inverse de $\Phi^* \Phi$ à l'Équation (4.34) se traduit, pour des variables vectorisées, par la forme matricielle par bloc suivante :

$$\Phi^* \Phi = \frac{1}{S_2 S_0 - S_1^2} \begin{pmatrix} S_0 \mathbf{I}_{N/2} & -S_1 \mathbf{I}_{N/2} \\ -S_1 \mathbf{I}_{N/2} & S_2 \mathbf{I}_{N/2} \end{pmatrix}, \quad (5.60)$$

qui permettra dans la suite de fortement simplifier le calcul des estimateurs SURE et SUGAR généralisés.

L'attribut discriminant en segmentation de textures est la régularité locale, \mathbf{h} , tandis que la puissance locale, \mathbf{v} , est un attribut auxiliaire. Par conséquent, la qualité des résultats sera mesurée par l'erreur quadratique *projetée* (5.10) sur la régularité locale, particularisée pour la segmentation de textures en :

$$R(\hat{\mathbf{h}})(\Lambda) \triangleq \mathbb{E}_{\zeta} \left\| \hat{\mathbf{h}}(\boldsymbol{\ell}; \Lambda) - \bar{\mathbf{h}} \right\|_2^2, \quad (5.61)$$

avec $\hat{\mathbf{h}}(\boldsymbol{\ell}; \Lambda)$ l'estimateur de régularité locale pouvant être obtenu, par Variation Totale avec seuillage, noté $\hat{\mathbf{h}}^{\text{ROF}}$ à Équation (2.123), par la résolution du problème à « contours libres », noté $\hat{\mathbf{h}}^{\text{L}}$ à l'Équation (4.5) ou par la résolution du problème à « contours co-localisés », noté $\hat{\mathbf{h}}^{\text{C}}$ à l'Équation (4.7). Le problème ROF $_{\mathbf{h}}$ (2.123) possède un unique hyperparamètre $\Lambda = \lambda \in \mathbb{R}_+$, tandis que les Problèmes (4.5) et (4.7), à contours « libres » ou « co-localisés » sont pilotés par deux paramètres de régularisation $\Lambda = (\lambda, \alpha) \in \mathbb{R}_+^2$. Enfin, l'opérateur de projection à considérer s'écrit, sous forme matricielle,

$$\Pi \triangleq \begin{pmatrix} \mathbf{I}_{N/2} & \mathbf{Z}_{N/2} \\ \mathbf{Z}_{N/2} & \mathbf{Z}_{N/2} \end{pmatrix} \quad \text{de sorte que} \quad \Pi \begin{pmatrix} \bar{\mathbf{h}} \\ \bar{\mathbf{v}} \end{pmatrix} = \begin{pmatrix} \bar{\mathbf{h}} \\ \mathbf{0}_{N/2} \end{pmatrix} \quad (5.62)$$

où $\mathbf{I}_{N/2}$ (resp. $\mathbf{Z}_{N/2}$) désigne la matrice identité (resp. nulle) de taille $N/2 \times N/2$ et $\mathbf{0}_{N/2}$ le vecteur nul de $\mathbb{R}^{N/2}$.

5.4.2 Propriétés des estimateurs de la régularité locale

Afin de pouvoir appliquer le formalisme décrit aux Section 5.2 et 5.3 aux estimateurs de la régularité locale $\hat{\mathbf{h}}^{\text{ROF}}(\boldsymbol{\ell}; \lambda)$, $\hat{\mathbf{h}}^{\text{L}}(\boldsymbol{\ell}; \lambda, \alpha)$ et $\hat{\mathbf{h}}^{\text{C}}(\boldsymbol{\ell}; \lambda, \alpha)$, il faut s'assurer que les hypothèses de régularité et d'intégrabilité requises sur les estimateurs sont vérifiées. Pour cela, nous allons commencer par formuler les Problèmes (2.123), (4.5) et (4.7) sous la forme de moindres carrés pénalisés (5.3).

Lemme 5.1. *Les Problèmes (2.123), (4.5) et (4.7) peuvent se formuler comme un problème de moindres carrés pénalisés, sur le modèle du Problème (5.3), et peuvent donc s'écrire*

$$\hat{\mathbf{x}}(\mathbf{y}; \mathbf{A}) = \underset{\mathbf{x} \in \mathcal{H}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{y} - \Phi \mathbf{x}\|_{\mathcal{W}}^2 + \|\mathbf{U}_{\Lambda} \mathbf{x}\|_{2,1}. \quad (5.63)$$

Démonstration. L'attache aux données des Problèmes (4.5) et (4.7) est exactement le terme de moindres carrés pénalisés avec $\mathcal{W} = \mathbf{I}$, $\mathbf{y} = \boldsymbol{\ell}$ et $\mathbf{x} = (\mathbf{h}, \mathbf{v})$. Quant au Problème (2.123), en remarquant que

$$\mathbf{A}(\Phi(\mathbf{h}, \mathbf{v}) - \boldsymbol{\ell}) = \begin{pmatrix} \mathbf{h} - \hat{\mathbf{h}}^{\text{RL}}(\boldsymbol{\ell}) \\ \mathbf{0} \end{pmatrix} \quad (5.64)$$

où \mathbf{A} est l'opérateur linéaire défini à l'Équation (5.9), son attache aux données

$$\frac{1}{2} \|\mathbf{h} - \hat{\mathbf{h}}^{\text{RL}}(\boldsymbol{\ell})\|_2^2 = \frac{1}{2} \|\Phi(\mathbf{h}, \mathbf{v}) - \boldsymbol{\ell}\|_{\mathbf{A}^* \mathbf{A}}^2 \quad (5.65)$$

peut se reformuler grâce à la distance de Mahalanobis avec $\mathcal{W} = \mathbf{A}^* \mathbf{A}$.

Pour la pénalisation, dans le cas du Problème (2.123), $\mathbf{U}_{\Lambda} \triangleq \lambda \mathbf{D}$, avec \mathbf{D} l'opérateur de différences horizontales et verticales agissant sur \mathbf{h} à l'Équation (2.41), pour la segmentation à contours « libres » $\mathbf{U}_{\Lambda} \triangleq \mathbf{U}_{(\lambda, \alpha)}^{\text{L}}$, introduit à l'Équation (4.55), et enfin, pour la segmentation à contours « co-localisés » $\mathbf{U}_{\Lambda} \triangleq \mathbf{U}_{(\lambda, \alpha)}^{\text{C}}$, introduit à l'Équation (4.56). \square

Remarque (Distance de Mahalanobis pour la reformulation du Problème ROF $_{\mathbf{h}}$ (2.123)). La formulation du Problème (2.123) à l'aide de la distance de Mahalanobis va nous permettre, d'une part, d'étudier globalement la régularité des trois estimateurs $\hat{\mathbf{h}}^{\text{ROF}}(\boldsymbol{\ell}; \lambda)$, $\hat{\mathbf{h}}^{\text{L}}(\boldsymbol{\ell}; \lambda, \alpha)$ et $\hat{\mathbf{h}}^{\text{C}}(\boldsymbol{\ell}; \lambda, \alpha)$. D'autre part, elle permet de souligner que, dans ce modèle, l'observation est bien $\log_2 \mathcal{L} = \boldsymbol{\ell}$. L'identification précise des quantités constituant les *observations* est crucial pour la mise en œuvre du formalisme de Stein pour le réglage (automatique) des hyperparamètres. En effet, pour la stratégie de différences finies utilisée dans le calcul des estimateurs $\hat{\mathbf{R}}_{\mathbf{v}, \varepsilon}(\boldsymbol{\ell}; \Lambda | \mathcal{S})$ et $\partial_{\Lambda} \hat{\mathbf{R}}_{\mathbf{v}, \varepsilon}(\boldsymbol{\ell}; \Lambda | \mathcal{S})$, c'est bien la variable $\boldsymbol{\ell}$ qui est perturbée, y compris dans le cas de du Problème ROF $_{\mathbf{h}}$ (2.123). Ainsi, dans la pratique, pour la méthode ROF $_{\mathbf{h}}$ -S, ce calcul fait intervenir $\hat{\mathbf{h}}^{\text{RL}}(\boldsymbol{\ell})$ et $\hat{\mathbf{h}}^{\text{RL}}(\boldsymbol{\ell} + \mathbf{v}\varepsilon)$, ainsi que la matrice \mathcal{S} de covariance du bruit sur $\boldsymbol{\ell}$.

Proposition 5.3. *L'unique solution $\hat{\mathbf{h}}^{\text{ROF}}(\boldsymbol{\ell}; \Lambda)$ (resp. $\begin{pmatrix} \hat{\mathbf{h}}^{\text{L}}(\boldsymbol{\ell}; \Lambda) \\ \hat{\mathbf{v}}_{\text{L}}(\boldsymbol{\ell}; \Lambda) \end{pmatrix}$, $\begin{pmatrix} \hat{\mathbf{h}}^{\text{C}}(\boldsymbol{\ell}; \Lambda) \\ \hat{\mathbf{v}}_{\text{C}}(\boldsymbol{\ell}; \Lambda) \end{pmatrix}$) du Problème (2.123) (resp. (4.5), (4.7)) est continue et différentiable au sens faible par rapport au logarithme des coefficients dominants $\boldsymbol{\ell}$, et intégrable par rapport à la densité de probabilité gaussienne (Hypothèse 5.3). En outre, $\hat{\mathbf{h}}^{\text{ROF}}(\boldsymbol{\ell}; \Lambda)$ (resp. $\hat{\mathbf{h}}^{\text{L}}(\boldsymbol{\ell}; \Lambda)$ et $\hat{\mathbf{v}}_{\text{L}}(\boldsymbol{\ell}; \Lambda)$, $\hat{\mathbf{h}}^{\text{C}}(\boldsymbol{\ell}; \Lambda)$ et $\hat{\mathbf{v}}_{\text{C}}(\boldsymbol{\ell}; \Lambda)$) sont des fonctions uniformément L_1 -lipschitziennes par rapport à $\boldsymbol{\ell}$ (Hypothèse 5.4).*

Démonstration. L'attache aux données du Problème (2.123) est, de manière directe, 1-fortement convexe, par conséquent le Problème (2.123) possède une unique solution, $\hat{\mathbf{h}}^{\text{ROF}}$. De plus, comme nous l'avons montré dans [47, Proposition 2], et à la Proposition 4.9 du Chapitre 4, les fonctions objectifs avec pénalisation à contours « libres » et à contours « co-localisées »

$$(\mathbf{h}, \mathbf{v}) \mapsto \underbrace{\frac{1}{2} \sum_{j=j_1}^{j_2} \|j\mathbf{h} + \mathbf{v} - \boldsymbol{\ell}_j\|_2^2}_{\|\Phi(\mathbf{h}, \mathbf{v}) - \boldsymbol{\ell}\|_2^2} + \lambda (\alpha \text{TV}(\mathbf{h}) + \text{TV}(\mathbf{v})) \quad \text{et} \quad (\mathbf{h}, \mathbf{v}) \mapsto \underbrace{\frac{1}{2} \sum_{j=j_1}^{j_2} \|j\mathbf{h} + \mathbf{v} - \boldsymbol{\ell}_j\|_2^2}_{\|\Phi(\mathbf{h}, \mathbf{v}) - \boldsymbol{\ell}\|_2^2} + \lambda \text{TV}_{\alpha}(\mathbf{h}, \mathbf{v}) \quad (5.66)$$

sont non seulement convexes, car somme de deux termes convexes, mais également fortement convexes. En effet, rappelons que le calcul des valeurs propres de $\Phi^* \Phi$ montre que le terme de moindres carrés, i.e. l'attache aux données, est μ_F -fortement convexe, avec

$$\mu_F = \min \text{Sp}(\Phi^* \Phi) > 0, \quad (5.67)$$

où $\text{Sp}(\Phi^* \Phi)$ désigne le spectre de l'opérateur linéaire (borné) $\Phi^* \Phi$. Par conséquent, les fonctions objectives considérées ont chacune un unique minimum, ce qui permet d'assurer que les applications $(\ell, \Lambda) \mapsto \hat{\mathbf{h}}^{\text{ROF}}(\ell; \Lambda)$, $(\ell, \Lambda) \mapsto \hat{\mathbf{h}}^{\text{L}}(\ell; \Lambda)$ et $(\ell, \Lambda) \mapsto \hat{\mathbf{h}}^{\text{C}}(\ell; \Lambda)$ sont correctement définies, ce qui est le premier point à vérifier pour la mise en œuvre du formalisme décrit aux Sections 5.2 et 5.3, comme discuté à la Remarque 5.2.2.

En outre, d'après le Lemme 5.1, les Problèmes (2.123), (4.5) et (4.7) peuvent se formuler comme un problème de moindres carrés pénalisés, sur le modèle du Problème (5.3), et peuvent donc s'écrire

$$\hat{\mathbf{x}}(\mathbf{y}; \Lambda) = \underset{\mathbf{x} \in \mathcal{H}}{\text{argmin}} \frac{1}{2} \|\mathbf{y} - \Phi \mathbf{x}\|_{\mathcal{W}}^2 + \mathcal{J}_\Lambda(\mathbf{x}), \quad (5.68)$$

où $\mathcal{J}_\Lambda(\mathbf{x}) = \|\mathbf{U}_\Lambda \mathbf{x}\|_{2,1}$ est construit à partir d'un opérateur linéaire \mathbf{U}_Λ qui dépend du (ou des) paramètre(s) de régularisation λ (et α). \mathcal{J}_Λ est convexe, propre, semi-continue inférieurement, donc, en suivant le raisonnement développé par [61, Section 8.7(ii)], le Problème (5.68) peut se réécrire comme un problème d'optimisation sous contrainte :

$$(\hat{\mathbf{x}}(\mathbf{y}; \Lambda), \hat{\mathbf{z}}(\mathbf{y}; \Lambda)) = \underset{\mathbf{x} \in \mathcal{H}, \mathbf{z} \in \mathcal{G}}{\text{argmin}} \frac{1}{2} \|\mathbf{y} - \mathbf{z}\|_{\mathcal{W}}^2 + \mathcal{J}_\Lambda(\mathbf{x}), \quad \text{tel que } \mathbf{z} = \Phi \mathbf{x} \quad (5.69)$$

$$\iff \hat{\mathbf{z}}(\mathbf{y}; \Lambda) = \underset{\mathbf{z} \in \mathcal{G}}{\text{argmin}} \frac{1}{2} \|\mathbf{y} - \mathbf{z}\|_{\mathcal{W}}^2 + (\Phi \mathcal{J}_\Lambda)(\mathbf{z}) \quad (5.70)$$

$$\iff \hat{\mathbf{z}}(\mathbf{y}; \Lambda) = \text{prox}_{\Phi \mathcal{J}_\Lambda}(\mathbf{y}) \quad (5.71)$$

où

$$(\Phi \mathcal{J}_\Lambda)(\mathbf{z}) \triangleq \min_{\{\mathbf{x} | \Phi \mathbf{x} = \mathbf{z}\}} \mathcal{J}_\Lambda(\mathbf{x}) \quad (5.72)$$

désigne la pré-image de \mathcal{J}_Λ sous l'action de l'opérateur linéaire Φ . Or, $\Phi \mathcal{J}_\Lambda$ est convexe, propre et semi-continue inférieurement, donc l'Équation (5.71) correspond à l'opérateur proximal de $\Phi \mathcal{J}_\Lambda$. De plus, il est connu qu'un opérateur proximal possède la propriété de non-expansivité [4, Proposition 12.27], i.e., est 1-lipschitzien, ce qui est donc le cas de l'estimateur $\hat{\mathbf{z}}(\mathbf{y}; \Lambda)$ de l'Équation (5.71). Enfin, grâce à l'Équation (5.69), $\hat{\mathbf{z}}(\mathbf{y}; \Lambda) = \Phi \hat{\mathbf{x}}(\mathbf{y}; \Lambda)$, et, puisque Φ est de rang maximal d'après la Proposition 4.6,

$$\hat{\mathbf{x}}(\mathbf{y}; \Lambda) = (\Phi^* \Phi)^{-1} \Phi^* \hat{\mathbf{z}}(\mathbf{y}; \Lambda). \quad (5.73)$$

L'opérateur linéaire Φ étant borné, il s'ensuit que l'estimateur $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ est uniformément L_1 -lipschitzien, avec $L_1 = \|\Phi\|^{-1}$, ce qui démontre l'Hypothèse 5.4 (i).

L'estimateur $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ étant uniformément 1-lipschitzien, il est continu et différentiable au sens faible par rapport à \mathbf{y} [28, Théorème 5, Section 4.2.3]. Par conséquent, $\langle \mathbf{A}^* \Pi \hat{\mathbf{x}}(\mathbf{y}; \Lambda), \boldsymbol{\zeta} \rangle$ et $\partial_\Lambda \hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ sont tous deux intégrables par rapport à la densité gaussienne et l'Hypothèse 5.3 est vérifiée.

Finalement, en posant $\mathbf{y} = \mathbf{0}_P$, pour tout $\Lambda \in \mathbb{R}^L$, $\hat{\mathbf{x}}(\mathbf{0}_P; \Lambda) = \mathbf{0}_N$ atteint le minimum de la fonctionnelle minimisée dans le Problème (5.69). La solution de ce problème étant unique d'après la Proposition 4.6, $\mathbf{0}_N$ est l'unique solution de (5.69) et l'Hypothèse 5.4 (ii) est vérifiée. \square

S'il est raisonnable de supposer que le résultat d'uniforme lipschitzianité par rapport aux hyperparamètres mentionné à la Remarque 5.3.1 s'étend aux estimateurs de moindres carrés pénalisés, à notre connaissance, aucune démonstration n'a été proposée pour montrer la lipschitzianité par rapport aux paramètres de régularisation de la solution d'un problème de moindres carrés pénalisés. La vérification de l'Hypothèse 5.5 pour les estimateurs de la régularité locale $\hat{\mathbf{h}}^{\text{ROF}}(\ell; \Lambda)$, $\hat{\mathbf{h}}^{\text{L}}(\ell; \Lambda)$ et $\hat{\mathbf{h}}^{\text{C}}(\ell; \Lambda)$ reste ouverte et pourrait faire l'objet d'un travail futur.

5.4.3 Aspects pratiques du calcul de $\widehat{R}_{v,\varepsilon}$ et $\partial_{\Lambda}\widehat{R}_{v,\varepsilon}$

Dans cette section, les points techniques rencontrés dans la mise en œuvre de l’Algorithme 14, sont détaillés et résolus dans le cadre de la segmentation de texture.

5.4.3.1 Structure de covariance des observations

Le bruit additif $\zeta_{j,\underline{n}}$ introduit à l’Équation (5.59) étant gaussien l’Hypothèse 5.1 est vérifiée. La matrice de covariance \mathcal{S} du bruit ζ s’écrit

$$\mathcal{S}_j^{j'}(\underline{n}, \underline{n}') \triangleq \mathbb{E}\zeta_j(\underline{n})\zeta_{j'}(\underline{n}') = \mathcal{C}_j^{j'} \Xi_j^{j'}(\underline{n} - \underline{n}'), \quad (5.74)$$

où

$$\mathcal{C}_j^{j'} \triangleq \mathbb{E}\zeta_j(\underline{n})\zeta_{j'}(\underline{n}), \quad \mathcal{C}_j^{j'} \text{ indépendant de } \underline{n} \quad (5.75)$$

quantifie la covariance inter-échelles, et $\Xi_j^{j'}$ décrit les corrélations spatiales, stationnaires, dont la longueur de corrélation est de l’ordre de $\max(2^j, 2^{j'})$ [63, Section 3.2].

5.4.3.2 Produit matriciel $\mathcal{S}\varepsilon$

Comme mentionné à la Remarque 5.2.4, en général, le calcul direct du produit $\mathcal{S}\varepsilon$, nécessaire à l’évaluation pratique de l’estimateur SURE par différences finies et méthode de Monte Carlo (5.17), est rendu impossible par la grande taille de la matrice \mathcal{S} . Néanmoins, dans le cas des logarithmes des coefficients dominants, la stationnarité des corrélations spatiales confère à la matrice \mathcal{S} une structure par blocs (5.74), où chaque bloc est constitué d’une matrice circulante, et le produit $\mathcal{S}\varepsilon$ peut alors être calculé efficacement grâce à la Formule (5.76), rendant l’estimateur SURE (5.17) utilisable en pratique.

En effet, pour un vecteur $\varepsilon \in \mathbb{R}^{N_{j_1} \times N_1 \times N_2} = (\varepsilon_j)_{j=j_1}^{j_2}$, avec $\varepsilon_j \in \mathbb{R}^{N_1 \times N_2}$,

$$(\mathcal{S}\varepsilon)_j(\underline{n}) = \sum_{j'=j_1}^{j_2} \sum_{\underline{n}' \in \Omega} \mathcal{S}_j^{j'}(\underline{n}, \underline{n}') \varepsilon_{j'}(\underline{n}') = \sum_{j'=j_1}^{j_2} \mathcal{C}_j^{j'} \sum_{\underline{n}' \in \Omega} \Xi_j^{j'}(\underline{n} - \underline{n}') \varepsilon_{j'}(\underline{n}') = \sum_{j'=j_1}^{j_2} \mathcal{C}_j^{j'} (\Xi_{j,j'} * \varepsilon_j)(\underline{n}). \quad (5.76)$$

qui est la somme de N_j produits de convolution, notés $*$, d’un vecteur de grande dimension $\varepsilon \in \mathbb{R}^{N_1 \times N_2}$ avec des fenêtres de support fini, de petite dimension $\mathcal{C}_j^{j'} \Xi_{j,j'}$. Ainsi, le calcul du produit $\mathcal{S}\varepsilon$ s’avère bien moins coûteux qu’un produit matriciel usuel entre une matrice de taille $P \times P$ et un vecteur de taille P .

5.4.3.3 Opérateur linéaire \mathbf{A}

De la même manière, les matrices Φ^* (4.32), $(\Phi^* \Phi)^{-1}$ (5.60) et Π (5.62), qui agissent pixel par pixel, échelle par échelle, sans aucun couplage, sont en réalité très parcimonieuses et un calcul direct fournit la structure par blocs suivante

$$\mathbf{A} = \Pi (\Phi^* \Phi)^{-1} \Phi^* = \frac{1}{S_0 S_2 - S_1^2} \begin{pmatrix} (S_0 + j_1 S_1) \mathbf{I}_{N/2} & \cdots & (S_0 + j_2 S_1) \mathbf{I}_{N/2} \\ \mathbf{Z}_{N/2} & \cdots & \mathbf{Z}_{N/2} \end{pmatrix}, \quad \text{où } S_m = \sum_{j=j_1}^{j_2} j^m, \quad (5.77)$$

contenant très peu de coefficients non nuls. Ainsi, les produits $\mathbf{A}^* \Pi \widehat{\mathbf{x}}(\mathbf{y}; \Lambda)$ and $\mathbf{A}^* \Pi \partial_{\Lambda} \widehat{\mathbf{x}}(\mathbf{y}; \Lambda)$, impliqués dans l’évaluation des estimateurs SURE (5.17) et SUGAR (5.20) par différences finies et méthode de Monte Carlo sont peu coûteux à calculer. Réalisés pixel par pixel, ils ne nécessitent que $\mathcal{O}(N)$ opérations.

5.4.3.4 Calcul du terme constant $\text{Tr}(\mathbf{A}\mathcal{S}\mathbf{A}^*)$

Évaluer $\hat{R}_{v,\varepsilon}(\ell; \mathbf{A}|\mathcal{S})$, par exemple à l'étape (5.44) de l'Algorithme 14 fournissant les estimateurs SURE et SUGAR généralisés, requiert le calcul de la trace d'une matrice de taille $N \times N$, avec N pouvant être de l'ordre 10^6 , comme par exemple en traitement d'images.

Dans l'application du formalisme SURE et SUGAR généralisé à la segmentation de texture, il est possible de combiner la structure de la matrice de covariance \mathcal{S} , décrite à l'Équation (5.74), avec l'expression parcimonieuse de l'opérateur linéaire \mathbf{A} de l'Équation (5.77) pour obtenir une expression compacte du troisième terme de l'estimateur SURE (5.17) pouvant être calculé à moindre coût, comme détaillé à la Proposition 5.4.

Proposition 5.4 (Terme constant de l'estimateur SURE). *Soit \mathcal{S} la matrice de covariance des logarithmes des coefficients dominants, dont la structure de covariance, parcimonieuse, est décrite à l'Équation (5.74) et \mathbf{A} l'opérateur linéaire défini par la Formule (5.9), à partir de l'opérateur $\mathbf{\Phi}$ de l'Équation (4.3) et de la projection $\mathbf{\Pi}$ de l'Équation (5.62). Alors, le terme constant apparaissant dans la Formule (5.17) de l'estimateur SURE par différences finies et méthode Monte Carlo s'écrit*

$$\text{Tr}(\mathbf{A}\mathcal{S}\mathbf{A}^*) = \frac{N/2}{(S_0S_2 - S_1^2)^2} \left(\sum_{j,j'} \left(S_1^2 \mathcal{C}_j^{j'} - 2S_0S_1 j' \mathcal{C}_j^{j'} + S_0^2 j j' \mathcal{C}_j^{j'} \right) \right), \quad (5.78)$$

où les quantités $\{S_m, m = 0, 1, 2\}$, introduites à l'Équation (2.91), sont rappelées à l'Équation (5.77), et $\mathcal{C}_j^{j'}$ désigne la covariance entre les échelles 2^j et $2^{j'}$, décrite à l'Équation (5.75).

Démonstration. La démonstration est détaillée à l'Annexe C.4. □

5.5 Performances de la segmentation automatique

Dans cette section, les performances de sélection d'hyperparamètres optimaux sont évaluées au moyen de simulations numériques dans le cadre de la segmentation de textures. Pour cela nous nous concentrons sur la méthode de segmentation de textures à contours « libres », i.e., *via* la résolution du Problème (4.5). La Section 5.5.1 détaille l'ensemble des réglages nécessaires pour la mise en œuvre des simulations et la Section 5.5.2 se concentre sur les questions algorithmiques. Puis, le rôle central de la matrice \mathcal{S} est mis en évidence à la Section 5.5.3. La Section 5.5.3.2 évalue l'impact de l'utilisation d'une matrice de covariance partielle par rapport à l'utilisation de la matrice de covariance complète de l'Équation (5.74), tandis que l'impact de l'utilisation d'une matrice \mathcal{S} estimée est illustré à la Section 5.5.3.3. Enfin, la Section 5.5.4 présente les performances du réglage automatique des hyperparamètres par le biais de l'Algorithme 15.

5.5.1 Réglages

5.5.1.1 Textures

Par soucis de simplicité, nous considérons la segmentation à $k = 2$ régions, avec le masque elliptique de la Figure 5.1a. Des textures synthétiques de résolution $N_1 \times N_2 = 256 \times 256$, caractérisées par deux configurations d'attributs :

- Configuration “D”, « difficile », dont une réalisation est présentée à la Figure 5.1b

$$\left(\bar{H}_1, \bar{\Sigma}_1^2 \right) = (0.5, 0.6) \text{ (fond)}, \quad \left(\bar{H}_2, \bar{\Sigma}_2^2 \right) = (0.75, 0.7) \text{ (ellipse centrale)}.$$

- Configuration “E”, « facile », dont une réalisation est présentée à la Figure 5.1c

$$\left(\bar{H}_1, \bar{\Sigma}_1^2 \right) = (0.5, 0.6) \text{ (fond)}, \quad \left(\bar{H}_2, \bar{\Sigma}_2^2 \right) = (0.9, 1.1) \text{ (ellipse centrale)}.$$

sont générées grâce à l'Algorithme 4.

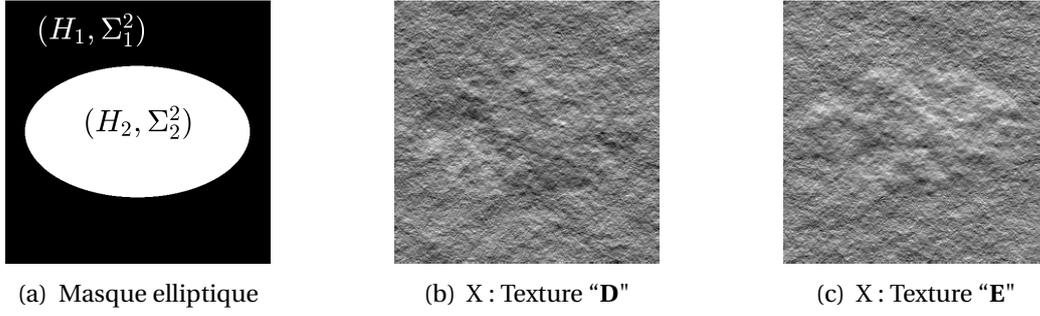


FIGURE 5.1 – (a) Masque pour les textures synthétiques composé de deux régions : le fond (en noir) dont la texture est caractérisée par une régularité locale uniforme $\bar{\mathbf{h}} \equiv \bar{H}_1$ et une variance locale uniforme également $\bar{\sigma}^2 \equiv \bar{\Sigma}_1^2$, puis l'ellipse centrale (en blanc) qui correspond à une texture homogène caractérisée par $\bar{\mathbf{h}} \equiv \bar{H}_2$ et $\bar{\sigma}^2 \equiv \bar{\Sigma}_2^2$. (b) et (c) textures fractales synthétiques homogènes par morceaux utilisées pour le calcul des performances, de résolution 256×256 pixels.

5.5.1.2 Analyse multi-échelles

Une transformée en ondelettes bidimensionnelle, non décimée, est calculée aux échelles $(2^j)_{j=1}^J$, avec une ondelette mère construite comme le produit tensoriel d'ondelettes de Daubechies asymétriques à trois moment nuls [42].

5.5.1.3 Mesure des performances

De même que dans nos précédents travaux [46, 47], pour une image texturée X donnée, et $(\ell_j)_{j=j_1}^{j_2}$ les logarithmes de ses coefficients d'ondelettes dominants, deux critères de performance sont utilisés :

- L'« erreur quadratique à un échantillon » sur la régularité locale, calculée à partir d'une seule réalisation de ℓ , i.e. sur les logarithmes des coefficients dominants d'une image X

$$\mathcal{R}(\ell; \Lambda) \triangleq \left\| \hat{\mathbf{h}}(\ell; \Lambda) - \bar{\mathbf{h}} \right\|_2^2, \quad (5.79)$$

où $\hat{\mathbf{h}}(\ell; \Lambda)$ est l'estimateur de la régularité locale considéré et $\bar{\mathbf{h}}$ est la véritable carte de régularité locale.

- L'« erreur de segmentation », définie comme la proportion de pixels mal classés

$$\mathcal{P}(\ell; \Lambda) \triangleq \left| \bar{\Omega}_1 \cap \hat{\Omega}_2(\ell; \Lambda) \right| + \left| \hat{\Omega}_1(\ell; \Lambda) \cap \bar{\Omega}_2 \right|, \quad (5.80)$$

où $\cup_i \hat{\Omega}_i(\ell; \Lambda)$ est la partition estimée, obtenue à la suite du seuillage de l'estimée $\hat{\mathbf{h}}(\ell; \Lambda)$.

Remarque. Par définition de l'erreur quadratique à l'Équation (5.61) et de l'« erreur quadratique à un échantillon » à l'Équation (5.79),

$$\mathbb{E}_\zeta \mathcal{R}(\ell; \Lambda) = R[\hat{\mathbf{h}}](\Lambda). \quad (5.81)$$

En pratique, un seul échantillon des observations, ici les logarithmes des coefficients dominants ℓ , est disponible, rendant par conséquent l'erreur quadratique exacte inaccessible. Ainsi, pour les expériences détaillées dans la suite, c'est l'« erreur quadratique à un échantillon » qui est utilisée comme référence à laquelle comparer l'estimateur SURE généralisé.

5.5.2 Questions algorithmiques

5.5.2.1 Algorithme primal-dual avec différentiation récursive

La minimisation de la fonctionnelle à contours « libres », i.e., la résolution du Problème (4.5), est effectuée au moyen du schéma primal-dual accéléré de l'Algorithme 12, de variable primale

$\mathbf{x} \triangleq (\mathbf{h}, \mathbf{v})$, tirant parti de la forte convexité du terme d'attache aux données, dont la particularisation au Problème (4.5) est donnée à l'Algorithme 9. Le nombre maximal d'itérations est fixé à $T_{\max} = 5 \cdot 10^5$, et un seuil à 10^{-4} sur le gap de dualité normalisé sert de critère d'arrêt à l'algorithme, comme détaillé à la Section 4.4.6.

5.5.2.2 Gamme d'échelles

L'estimation d'attributs *locaux* constants par morceaux nécessite de se concentrer sur les échelles les plus fines, afin de détecter précisément les contours. Ainsi, idéalement, le terme de moindres carrés porterait uniquement sur les deux échelles les plus fines de la transformée en ondelettes, c'est-à-dire sur les octaves $j_1 = 1$ à $j_2 = 2$. Cependant, la stratégie d'accélération déployée à l'Algorithme 12 est d'autant plus efficace que le module de forte convexité de l'attache aux données μ_F est élevé. Les valeurs de μ_F , rappelées au Tableau 5.1, pour $j_1 = 1$ fixé, montrent alors que la vitesse de convergence, pilotée par μ_F , augmente avec le nombre d'échelles considérées. Un bon compromis consiste à choisir $j_2 = 3$.

	$j_2 = 2$	$j_2 = 3$	$j_2 = 4$	$j_2 = 5$	$j_2 = 6$
μ_F	0,29	0,72	1,20	1,69	2,20

TABLEAU 5.1 – Module de forte-convexité μ_F du terme d'attache aux données dans le Problème (4.5), calculé à partir de la Formule (5.67), pour $j_1 = 1$ fixé et j_2 variable. L'entrée en gras correspond au choix effectué pour les expériences menées dans la suite de ce chapitre.

5.5.2.3 Paramètres de différences finies et de Monte Carlo

Le vecteur utilisé pour la méthode de Monte Carlo, $\boldsymbol{\varepsilon} \in \mathbb{R}^P$, où $P = N_j N_1 N_2$, est généré aléatoirement de manière i.i.d. selon une distribution gaussienne normalisée de moyenne nulle $\mathcal{N}(\mathbf{0}_P, \mathbf{I}_P)$. L'heuristique de [21, Section 3.2] pour le choix du pas de différences finies est adapté au cas du bruit corrélé, et conduit au choix :

$$\mathbf{v} = \frac{2}{P^\beta} \max \left(\sqrt{C_j^j}, j \in \{1, \dots, J\} \right), \quad \beta = 0.3, \quad (5.82)$$

où C_j^j est la variance de $\boldsymbol{\ell}_j$, le logarithme des coefficients dominants à l'échelle 2^j .

Les dérivées des estimées par rapport aux paramètres de régularisation, $\partial_\Lambda \hat{\mathbf{h}}(\boldsymbol{\ell}; \Lambda)$, $\partial_\Lambda \hat{\mathbf{v}}(\boldsymbol{\ell}; \Lambda)$, sont obtenues grâce à la différentiation récursive de l'Algorithme primal-dual 12, particularisé au Problème (4.5) de segmentation à contours « libres ».

5.5.2.4 Algorithme de quasi-Newton

Afin de réaliser la minimisation de l'erreur quadratique estimée, i.e., de résoudre le Problème (5.19), nous utilisons l'algorithme de quasi-Newton implémenté dans la toolbox GRANSO⁴ (*Gradient-based Algorithm for Non-Smooth Optimization*), à partir de l'algorithme de quasi-Newton avec stratégie BFGS proposé dans [19, Procédure 2], consistant en un algorithme quasi-Newton BFGS à faible utilisation de mémoire, avec des contraintes de boîtes permettant d'imposer la positivité des hyperparamètres λ et α . La structure générale de cet algorithme, particularisé au Problème (5.19), est présentée à l'Algorithme 15. Le nombre maximal d'itérations de l'Algorithme BFGS 15 est fixé à $M_{\max} = 250$, tandis que le critère d'arrêt est basé sur la norme du gradient, et fixé à 10^{-6} .

4. <http://www.timmitchell.com/software/GRANSO/>

Comme mentionné à la Section 5.3.3, l'initialisation est cruciale pour la convergence des algorithmes de quasi-Newton. Par conséquent, nous proposons une initialisation adaptée au Problème (4.5) de segmentation à contours « libres ». Les hyperparamètres initiaux sont choisis de sorte que les ordres de grandeur de l'attache aux données et de la pénalisation de (4.5) soient identiques. Pour cela, on considère que l'ordre de grandeur de l'attache aux données est contrôlé par la variance du bruit, de sorte que

$$\mathbb{E} \sum_{j=j_1}^{j_2} \|j\bar{\mathbf{h}} + \bar{\mathbf{v}} - \boldsymbol{\ell}_j\|_2^2 = \text{tr}(\mathcal{S}), \quad (5.83)$$

tandis que la pénalisation est évaluée sur les estimées de la régularité et de la variance locales obtenues par régression linéaires, $(\hat{\mathbf{h}}^{\text{RL}}, \hat{\mathbf{v}}^{\text{RL}})$, introduites à l'Équation (2.90). Ainsi, les hyperparamètres initiaux de l'Algorithme 15 sont choisis de sorte que

$$\boldsymbol{\Lambda}^{[0]} = (\lambda^{[0]}, \alpha^{[0]}), \quad \text{avec} \quad \lambda^{[0]} = \frac{\text{tr}(\mathcal{S})}{2\text{TV}(\hat{\mathbf{v}}^{\text{RL}}(\boldsymbol{\ell}))}, \quad \text{et} \quad \alpha^{[0]} = \frac{\text{TV}(\hat{\mathbf{v}}^{\text{RL}}(\boldsymbol{\ell}))}{\text{TV}(\hat{\mathbf{h}}^{\text{RL}}(\boldsymbol{\ell}))}. \quad (5.84)$$

L'approximation de l'inverse de la matrice hessienne est initialisée de manière à garantir que $\boldsymbol{\Lambda}^{[1]} = (1 \pm \kappa)\boldsymbol{\Lambda}^{[0]}$. Pour cela elle est choisie diagonale avec les coefficients suivants :

$$\mathbf{H}^{[0]} = \text{diag} \left(\left| \frac{\kappa\lambda^{[0]}}{\partial_\lambda \widehat{\mathbf{R}}_{\mathbf{v},\varepsilon}(\boldsymbol{\ell}; \boldsymbol{\Lambda}^{[0]} | \mathcal{S})} \right|, \left| \frac{\kappa\alpha^{[0]}}{\partial_\alpha \widehat{\mathbf{R}}_{\mathbf{v},\varepsilon}(\boldsymbol{\ell}; \boldsymbol{\Lambda}^{[0]} | \mathcal{S})} \right| \right). \quad (5.85)$$

En pratique, $\kappa = 0.5$ est fixé pour toutes les expériences. Ce choix de matrice $\mathbf{H}^{[0]}$ permet d'éviter que la première itération de l'Algorithme 15 de quasi-Newton aboutisse à des hyperparamètres $\boldsymbol{\Lambda}^{[1]}$ très éloignés des valeurs « naturelles » mises en évidence à l'Équation (5.84), ce qui pourrait induire un coût de calcul élevé pour atteindre les hyperparamètres optimaux.

5.5.3 Covariance des logarithmes des coefficients dominants

5.5.3.1 Procédures de calcul de la covariance

Aucune forme explicite de la matrice de covariance \mathcal{S} en fonction des attributs de régularité et de variance locales n'est disponible. Ainsi, même pour des textures synthétiques dont toutes les propriétés, et notamment les cartes de régularité et variance locales, $\bar{\mathbf{h}}$ et $\bar{\mathbf{v}}$, on ne dispose pas d'un accès direct à la structure de covariance véritable. Par conséquent, il est nécessaire de se tourner vers une estimation. À partir d'un échantillon $\boldsymbol{\ell}$, calculé à partir d'une réalisation \mathbf{X} de la texture considérée, la matrice de covariance « estimée » est calculée en utilisant un estimateur classique de covariance *via* une moyenne spatiale

$$\widehat{\mathcal{F}}_j^{j'}(\underline{n}, \underline{n}') \triangleq \frac{1}{|\Omega|} \sum_{\underline{n} \in \Omega} \ell_j(\underline{n}) \ell_{j'}(\underline{n} + \delta \underline{n}) - \left(\frac{1}{|\Omega|} \sum_{\underline{n} \in \Omega} \ell_j(\underline{n}) \right) \left(\frac{1}{|\Omega|} \sum_{\underline{n} \in \Omega} \ell_{j'}(\underline{n}) \right), \quad (5.86)$$

pour un écart spatial $\delta \underline{n} \triangleq \underline{n}' - \underline{n}$, qui conduit à la covariance inter-échelles

$$\widehat{\mathcal{C}}_j^{j'} = \widehat{\mathcal{F}}_j^{j'}(\underline{n}, \underline{n}) \quad (5.87)$$

et aux corrélations spatiales

$$\widehat{\Xi}_j^{j'}(\delta \underline{n}) = \frac{\widehat{\mathcal{F}}_j^{j'}(\underline{n}, \underline{n}')}{\widehat{\mathcal{C}}_j^{j'}}. \quad (5.88)$$

Puis, pour chaque configuration de textures “D” and “E”, la « vraie » matrice de covariance est obtenu numériquement en moyennant les matrices de covariance estimées $\widehat{\mathcal{F}}^{(q)}$ sur $Q = 5000$ échantillons de textures, i.e.,

$$\mathcal{S} \triangleq \left\langle \widehat{\mathcal{F}}^{(q)} \right\rangle_{q=1}^Q, \quad (5.89)$$

les échantillons étant générés grâce à l'Algorithme 4, décrit au Chapitre 2.

5.5.3.2 Matrice de covariance partielle *versus* complète pour l'estimation de l'erreur quadratique

Plus la matrice de covariance \mathcal{S} est complexe, i.e., plus les observations sont corrélées, plus elle est difficile à calculer et demande d'efforts pour être manipulée. Il est donc naturel de s'interroger sur la nécessité de prendre en compte l'ensemble de la structure de covariance. Afin de répondre à cette question, deux versions *partielles* de la matrice de covariance \mathcal{S} , décrite à l'Équation (5.89), sont considérées :

1. Une matrice des *variances*, \mathcal{S}_{var} négligeant à la fois les corrélations inter-échelles et spatiales, et se réduisant donc à une matrice diagonale, dont les coefficients sont les variances \mathcal{C}_j^j des ℓ_j ,

$$\mathcal{S}_{\text{var}}^{j'}(\underline{n}, \underline{n}') = \mathcal{C}_j^j \delta_{j,j'} \delta(\underline{n}, \underline{n}'). \quad (5.90)$$

2. Une matrice de covariance *inter-échelles*, \mathcal{S}_{int} , négligeant les corrélations spatiales, et contenant donc uniquement les corrélations croisées $\mathcal{C}_j^{j'}$ entre ℓ_j et $\ell_{j'}$ *au même pixel*,

$$\mathcal{S}_{\text{int}}^{j'}(\underline{n}, \underline{n}') = \mathcal{C}_j^{j'} \delta(\underline{n}, \underline{n}'). \quad (5.91)$$

La méthode de segmentation à contours « libres » est mise en œuvre pour une grille logarithmique de 15×15 paramètres de régularisation $\Lambda = (\lambda, \alpha)$ sur un échantillon de la Texture “D” (resp. “E”). Pour chaque couple de paramètres sont calculés et représentés à la Figure 5.2 :

- L'« erreur quadratique à un échantillon », $\mathcal{R}(\ell; \Lambda)$, présentée à la Figure 5.2a (resp. 5.2c), dont le minimum, indiqué par le symbole ‘+’, correspond à la segmentation de la Figure 5.2b (resp. 5.2d).
- L'estimateur SURE généralisé calculé à partir de la matrice des *variances* \mathcal{S}_{var} , $\hat{R}_{v,\varepsilon}(\ell; \Lambda | \mathcal{S}_{\text{var}})$, présenté à la Figure 5.2e (resp. 5.2g), dont le minimum, indiqué par le symbole ‘□’, correspond à la segmentation de la Figure 5.2f (resp. 5.2h).
- L'estimateur SURE généralisé calculé à partir de la matrice de covariance *inter-échelles* \mathcal{S}_{int} , $\hat{R}_{v,\varepsilon}(\ell; \Lambda | \mathcal{S}_{\text{int}})$, présenté à la Figure 5.2i (resp. 5.2k), dont le minimum, indiqué par le symbole ‘◇’, correspond à la segmentation de la Figure 5.2j (resp. 5.2l).
- L'estimateur SURE généralisé calculé à partir de la matrice de covariance complète \mathcal{S} , $\hat{R}_{v,\varepsilon}(\ell; \Lambda | \mathcal{S})$, présenté à la Figure 5.2m (resp. 5.2o), dont le minimum, indiqué par le symbole ‘△’, correspond à la segmentation de la Figure 5.2n (resp. 5.2p).
- L'« erreur de segmentation », $\mathcal{P}(\ell; \Lambda)$, présentée à la Figure 5.2q (resp. 5.2s), dont le minimum, indiqué par le symbole ‘*’, correspond à la segmentation de la Figure 5.2r (resp. 5.2t).

Remarque. À des fins de comparaison, les minimas de chacun des cinq indicateurs (erreur et estimateurs SURE), indiqués par les symboles ‘+’, ‘□’, ‘◇’, ‘△’ et ‘*’, sont superposés sur chacune des cartes. Néanmoins, dans un souci de lisibilité, leur légende n'est indiquée qu'une seule fois, sur la carte représentant l'indicateur qu'ils minimisent.

Pour la Texture “D”, les estimateurs SURE généralisés $\hat{R}_{v,\varepsilon}(\ell; \Lambda | \mathcal{S}_{\text{var}})$ (à la Figure 5.2e) et $\hat{R}_{v,\varepsilon}(\ell; \Lambda | \mathcal{S}_{\text{int}})$ (Figure 5.2i) échouent tous deux à reproduire l'erreur quadratique à un échantillon, $\mathcal{R}(\ell; \Lambda)$ (Figure 5.2a). Par conséquent, les hyperparamètres sélectionnés $\hat{\Lambda}_{v,\varepsilon}^{\dagger}(\ell | \mathcal{S}_{\text{var}})$ (‘□’) et $\hat{\Lambda}_{v,\varepsilon}^{\dagger}(\ell | \mathcal{S}_{\text{int}})$ (‘◇’) ne coïncident pas avec les hyperparamètres minimisant réellement l'erreur quadratique, $\Lambda_{\mathcal{R}}$ (‘+’). Les segmentation correspondantes, $S\hat{h}(\ell; \hat{\Lambda}_{v,\varepsilon}^{\dagger}(\ell | \mathcal{S}_{\text{var}}))$ (Figure 5.2f) et $S\hat{h}(\ell; \hat{\Lambda}_{v,\varepsilon}^{\dagger}(\ell | \mathcal{S}_{\text{int}}))$ (Figure 5.2j), diffèrent significativement de la segmentation $S\hat{h}(\ell; \Lambda_{\mathcal{R}})$ (Figure 5.2b).

Au contraire, $\hat{R}_{v,\varepsilon}(\ell; \Lambda | \mathcal{S})$ (Figure 5.2m) reproduit parfaitement $\mathcal{R}(\ell; \Lambda)$ (Figure 5.2a). En outre, grâce au calcul exact du terme constant de l'estimateur SURE généralisé, $\text{Tr}(\mathbf{A}\mathcal{S}\mathbf{A}^*)$, à la Proposition 5.4, l'ordre de grandeur de $\mathcal{R}(\ell; \Lambda)$ est correctement reproduit par $\hat{R}_{v,\varepsilon}(\ell; \Lambda | \mathcal{S})$, comme on

peut l'observer sur les échelles de couleur de la Figure 5.2. Enfin, $\widehat{\Lambda}_{v,\varepsilon}^{\dagger}(\ell|\mathcal{S})$ (' Δ ') coïncide avec $\Lambda_{\mathcal{R}}$ ('+') , et conduit à une segmentation $S\widehat{h}(\ell; \widehat{\Lambda}_{v,\varepsilon}^{\dagger}(\ell|\mathcal{S}))$ (Figure 5.2n) similaire à $S\widehat{h}(\ell; \Lambda_{\mathcal{R}})$ (Figure 5.2b). Les mêmes observations peuvent être réalisées pour la Texture "E" aux colonnes 3, 4 de la Figure 5.2.

Au travers de ces deux exemples, l'importance cruciale de prendre en compte la *totalité* de la structure de covariance pour obtenir un estimateur SURE correct, i.e., pour que $\mathcal{R}(\ell; \Lambda)$ soit bien approximé par $\widehat{R}_{v,\varepsilon}(\ell; \Lambda|\mathcal{S})$, a été illustrée. De plus, il est apparu que $\Lambda_{\mathcal{R}}$ était correctement approchée par les hyperparamètres optimaux $\Lambda_{v,\varepsilon}^{\dagger}(\ell|\mathcal{S})$, obtenus en minimisant l'estimateur SURE calculé à partir de la matrice de covariance *complète*.

5.5.3.3 Matrice de covariance estimée *versus* « vraie »

Si, à la Section précédente nous avons utilisé la « vraie » matrice de covariance des Textures "D" et "E", pour illustrer l'importance de *toute* la structure des corrélations, en pratique cette donnée n'est pas accessible. Il faut donc s'assurer qu'une matrice de covariance estimée à partir d'un seul échantillon permet tout de même, à condition que toute la structure de covariance soit considérée, de fournir un estimateur SURE pertinent.

Pour la Texture "D", $\widehat{R}_{v,\varepsilon}(\ell; \Lambda|\widehat{\mathcal{S}})$ (Figure 5.3b) est identique à $\widehat{R}_{v,\varepsilon}(\ell; \Lambda|\mathcal{S})$ (Figure 5.3a). En outre, les hyperparamètres optimaux $\widehat{\Lambda}_{v,\varepsilon}^{\dagger}(\ell|\widehat{\mathcal{S}})$ (' Δ ') coïncident exactement avec $\widehat{\Lambda}_{v,\varepsilon}^{\dagger}(\ell|\mathcal{S})$ (' Δ '), et conduisent à des segmentations $S\widehat{h}(\ell; \widehat{\Lambda}_{v,\varepsilon}^{\dagger}(\ell|\widehat{\mathcal{S}}))$ (Figure 5.3f) et $S\widehat{h}(\ell; \widehat{\Lambda}_{v,\varepsilon}^{\dagger}(\ell|\mathcal{S}))$ (Figure 5.3e) similaires. Ces observations sont quantifiées précisément au Tableau 5.2 en terme d'erreur quadratique à un échantillon $\mathcal{R}(\ell; \Lambda)$ et de proportion de pixels mal classés. Des résultats identiques sont obtenus pour la Texture "E".

La Figure 5.3 et les résultats quantitatifs présentés au Tableau 5.2 montrent que $\widehat{R}_{v,\varepsilon}(\ell; \Lambda|\widehat{\mathcal{S}})$ fournit un bon estimateur de $\mathcal{R}(\ell; \Lambda)$, et que $\Lambda_{\mathcal{R}}$ est correctement approché par $\widehat{\Lambda}_{v,\varepsilon}^{\dagger}(\ell|\widehat{\mathcal{S}})$.

Hyperparamètres Λ	Texture "D"		Texture "E"	
	$\mathcal{R}(\ell; \Lambda)$	$\mathcal{P}(\ell; \Lambda)$	$\mathcal{R}(\ell; \Lambda)$	$\mathcal{P}(\ell; \Lambda)$
$\Lambda_{\mathcal{R}}$ '+'	2,32 10 ³	7,79%	2,66 10 ³	5,34%
$\widehat{\Lambda}_{v,\varepsilon}^{\dagger}(\ell \mathcal{S})$ ' Δ '	2,35 10 ³	5,51%	2,83 10 ³	9,58%
$\widehat{\Lambda}_{v,\varepsilon}^{\dagger}(\ell \widehat{\mathcal{S}})$ ' Δ '	2,35 10 ³	5,51%	2,96 10 ³	4,61%
$\widehat{\Lambda}_{v,\varepsilon}^{\text{BFGS}}(\ell \mathcal{S})$ ' ∇ '	2,36 10 ³	4,66%	2,83 10 ³	3,71%
$\widehat{\Lambda}_{v,\varepsilon}^{\text{BFGS}}(\ell \widehat{\mathcal{S}})$ ' ∇ '	2,36 10 ³	6,22%	2,83 10 ³	3,27%

TABLEAU 5.2 – Performance en terme d'erreur quadratique et d'erreur de segmentation de la recherche sur grille *versus* l'Algorithme de quasi-Newton 15 pour les deux types de Textures "D" et "E".

5.5.4 Sélection automatique des hyperparamètres

Dans cette section, nous allons nous attacher à illustrer l'efficacité de l'Algorithme 15 pour la sélection *automatique* des hyperparamètres.

5.5.4.1 Convergence effective de l'Algorithme de quasi-Newton

La convergence de l'Algorithme de quasi-Newton 15, s'appuyant sur les estimateurs SURE et SUGAR généralisés calculés par l'Algorithme 14, est vérifiée empiriquement, en comparant les hyperparamètres sélectionnés automatiquement, $\widehat{\Lambda}_{v,\varepsilon}^{\text{BFGS}}$, avec les hyperparamètres obtenus par recherche sur grille exhaustive $\widehat{\Lambda}_{v,\varepsilon}^{\dagger}$, réalisée en appelant l'Algorithme 13 sur une grille de 15 × 15

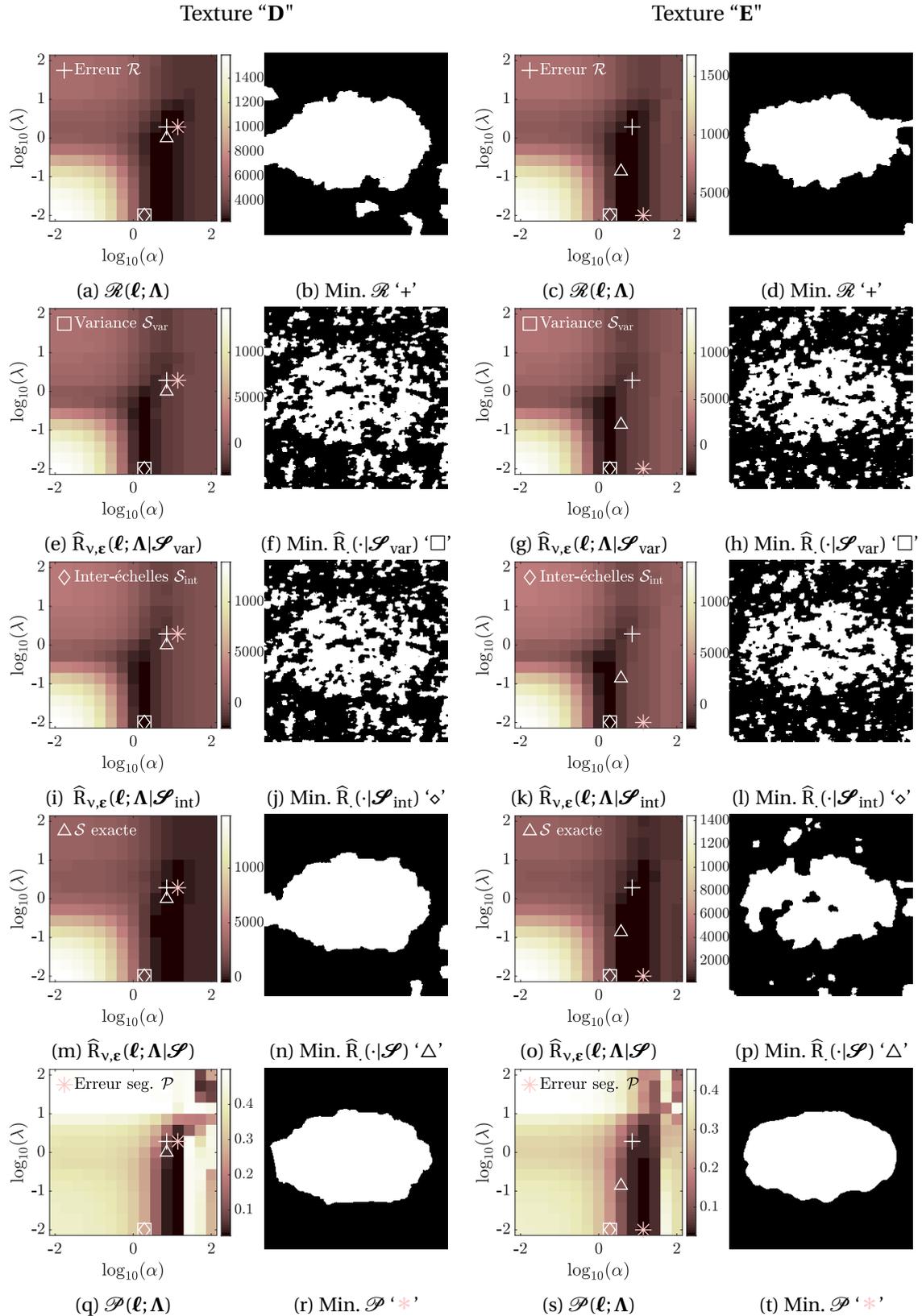


FIGURE 5.2 – Cartes d’erreur pour la segmentation de texture par la méthode à contours « libre » sur une grille d’hyperparamètres (λ, α) , accompagnées des segmentation optimales associées pour les textures homogènes par morceaux “D” (colonnes 1, 2) et “E” (colonnes 3, 4). Les erreurs quadratiques estimées $\hat{R}_{v,\epsilon}(\ell; \Lambda | \mathcal{S})$, calculées respectivement à partir de la matrice des *variances* \mathcal{S}_{var} (deuxième ligne), de la matrice des covariances *inter-échelles* \mathcal{S}_{int} (troisième ligne), ou de la matrice de covariance *complète* \mathcal{S} (quatrième ligne) sont comparées.

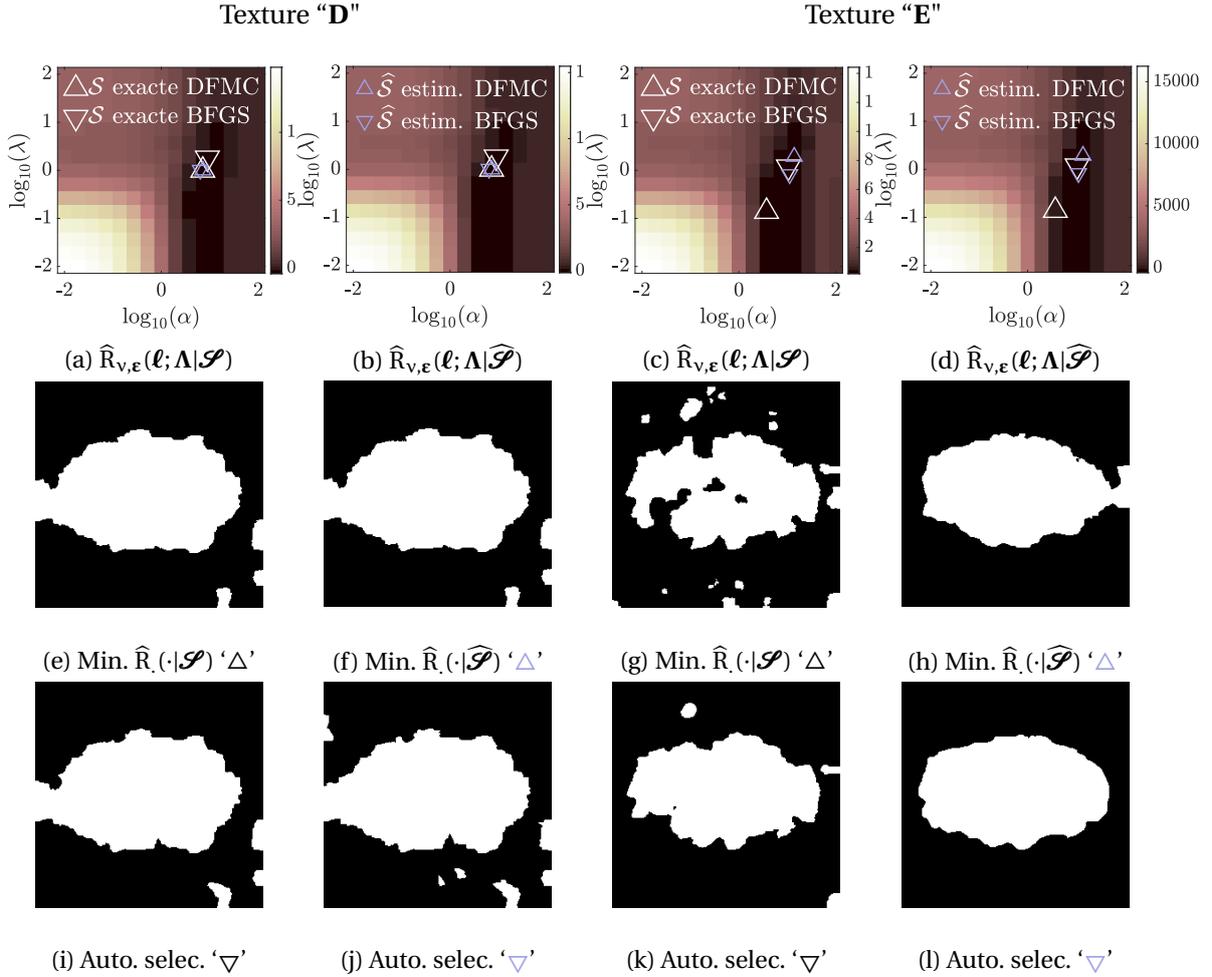


FIGURE 5.3 – Estimateur SURE généralisé calculé ou bien à partir de la « vraie » matrice de covariance \mathcal{S} (5.89), ou à partir de la matrice de covariance estimée $\hat{\mathcal{S}}$ (5.86) pour la segmentation de texture dans les configurations "D" et "E" (première ligne). Segmentations obtenues en minimisant l'estimateur SURE associé, placé au dessus (deuxième ligne). Segmentations obtenues par sélection automatique des paramètres de régularisation grâce à l'Algorithme 15, faisant usage de l'estimateur SUGAR généralisé (troisième ligne).

couples de paramètres de régularisation (λ, α) .

Les Figures 5.3a et 5.3b montrent que $\hat{\Lambda}_{v,\varepsilon}^{\text{BFGS}}(\ell | \mathcal{S})$ ('}\nabla\text{'}) et $\hat{\Lambda}_{v,\varepsilon}^{\text{BFGS}}(\ell | \hat{\mathcal{S}})$ ('}\nabla\text{'}) coïncident respectivement avec $\hat{\Lambda}_{v,\varepsilon}^{\dagger}(\ell | \mathcal{S})$ ('}\triangle\text{'}) et $\hat{\Lambda}_{v,\varepsilon}^{\dagger}(\ell | \hat{\mathcal{S}})$ ('}\triangle\text{'}) dans le cas de la Texture "D". Les mêmes conclusions peuvent être tirées des Figures 5.3c et 5.3d pour la Texture "E".

La Figure 5.3 et les résultats quantitatifs fournis au Tableau 5.2 témoignent de la convergence de l'Algorithme 15, où les estimateurs SURE et SUGAR sont calculés à partir de \mathcal{S} (resp. $\hat{\mathcal{S}}$) vers le minimum de $\hat{R}_{v,\varepsilon}(\ell; \Lambda | \mathcal{S})$ (resp. $\hat{R}_{v,\varepsilon}(\ell; \Lambda | \hat{\mathcal{S}})$).

En terme de coût de calcul, tandis que l'Algorithme 15 réalise en moyenne 40 appels à l'Algorithme 12, la recherche sur grille effectuée à la Section 5.5.3.2 nécessite 225 appels. On saisit l'avantage majeur de la sélection automatique par algorithme de quasi-Newton.

5.5.4.2 Performance de segmentation avec sélection automatique de Λ

Afin de valider les performances de segmentation de la méthode à contours « libres », de l'Équation (4.5) avec sélection automatique des paramètres de régularisation, dix réalisations des tex-

tures “D” et “E” sont générées à partir de l’Algorithme 4. Pour chacune, l’Algorithme 15 est mis en œuvre à deux reprises, une première fois en utilisant la « vraie » matrice de covariance \mathcal{P} dans le calcul des estimateurs SURE et SUGAR, une seconde fois en utilisant la matrice de covariance estimée $\widehat{\mathcal{P}}$.

Puisqu’aucune recherche sur grille n’est réalisée, on ne connaît pas la valeur minimale de l’erreur. Par conséquent les performances seront mesurée en terme d’erreur quadratique à un échantillon *normalisée*, $\tilde{\mathcal{R}}$ définie comme

$$\tilde{\mathcal{R}}(\ell|\mathcal{P}) = \frac{\mathcal{R}(\ell; \widehat{\Lambda}_{v,\varepsilon}^{\text{BFGS}}(\ell|\mathcal{P}))}{\|\widehat{\mathbf{h}}^{\text{RL}}(\ell) - \bar{\mathbf{h}}\|_2^2} = \frac{\|\widehat{\mathbf{h}}_{v,\varepsilon}^{\text{BFGS}}(\ell|\mathcal{P}) - \bar{\mathbf{h}}\|_2^2}{\|\widehat{\mathbf{h}}^{\text{RL}}(\ell) - \bar{\mathbf{h}}\|_2^2}, \quad (5.92)$$

qui quantifie l’amélioration de la qualité de l’estimation de régularité locale, entre l’estimée par régression linéaire $\widehat{\mathbf{h}}^{\text{RL}}$ et l’estimée régularisée à contours « libres » avec sélection automatique des paramètres de régularisation, $\widehat{\mathbf{h}}^{\text{L}}(\ell; \widehat{\Lambda}_{v,\varepsilon}^{\text{BFGS}}(\ell|\mathcal{P}))$.

Les performances moyennées sur dix réalisations, présentées dans le Tableau 5.3, montrent une diminution de l’erreur quadratique d’un facteur 16 pour la Texture “D” et d’un facteur 14 pour la Texture “E”, entre l’estimée par régression linéaire et l’estimée régularisée à contours « libres » avec sélection automatique des paramètres de régularisation. L’erreur de segmentation correspondante est de 6% pour la Texture “D”, et de 3% pour la Texture “E”, ce qui coïncide avec les erreurs de segmentation obtenues par recherche sur grille présentées au Tableau 5.2 et appuie la pertinence de la sélection automatique. En outre, on constate que l’utilisation d’une matrice de covariance estimée ne dégrade aucunement les performances obtenues avec la « vraie » matrice de covariance.

Ainsi, l’Algorithme 15, où les estimateurs SURE et SUGAR sont calculés à partir de la matrice de covariance estimée, $\widehat{\mathcal{P}}$ (5.86), fournissent une méthode de segmentation à contours « libres » performante, ne requérant aucun réglage de paramètres et s’appuyant *uniquement* sur l’image en entrée pour le calcul de la matrice de covariance.

	Texture “D”		Texture “E”	
	\mathcal{P}	$\widehat{\mathcal{P}}$	\mathcal{P}	$\widehat{\mathcal{P}}$
$\tilde{\mathcal{R}}(\ell \cdot)$	0,060 ± 0,003	0,057 ± 0,002	0,071 ± 0,003	0,073 ± 0,004
$\mathcal{P}(\ell; \widehat{\Lambda}_{v,\varepsilon}^{\text{BFGS}}(\ell \cdot))$ (%)	5,4 ± 0,7	6,8 ± 1,5	3,3 ± 0,7	2,8 ± 0,3

TABLEAU 5.3 – Performances moyennées de la méthode de segmentation à contours « libres » avec sélection automatique des paramètres de régularisation.

5.6 Conclusion

Dans ce chapitre, nous avons considéré le problème général de la sélection automatique des hyperparamètres optimaux pour les estimateurs paramétriques tels que le filtrage linéaire paramétrique ou les moindres carrés pénalisés et avons construit une procédure de réglage automatique, pilotée par les données, bénéficiant d’une assise théorique solide ainsi que d’une mise en œuvre efficace.

Dans ce but, l’estimateur classique de Stein a été réécrit pour prendre en compte un bruit additif gaussien corrélé, de structure de covariance quelconque. Sur ce point, nous avons proposé une formulation de l’estimateur SURE où apparaît uniquement la matrice de covariance, mais ne

faisant pas intervenir son inverse, ce qui rend le formalisme proposé plus adapté à des matrices de covariance non diagonales. En outre, l'utilisation d'un terme de moindres carrés *ordinaires* dans le Problème (5.3), permet de tirer partie de la forte-convexité de l'attache aux données pour construire un schéma de minimisation accéléré.

En dérivant l'estimateur SURE généralisé obtenu par rapport aux hyperparamètres, un estimateur SUGAR généralisé, approchant le gradient de l'erreur quadratique, est proposé. Son caractère asymptotiquement non biaisé est démontré, lui conférant validation théorique grâce à des conditions de régularité sur les estimateurs paramétriques considérés.

Le cas des estimateurs paramétriques séquentiels est largement discuté dans le cas du schéma de minimisation primal-dual pour la résolution du problème des moindres carrés pénalisés et un algorithme avec différentiation récursive est proposé.

Les estimateurs SURE et SUGAR généralisés sont ensuite mis à profit au sein d'un algorithme de quasi-Newton de manière à effectuer une minimisation automatique de l'erreur quadratique, *via* la sélection d'hyperparamètres optimaux. Un algorithme explicite réalisant cette sélection automatique est décrit.

Afin de valider les performances de la procédure de sélection automatique des paramètres de régularisation formulée dans un cadre général, elle est particularisée au problème de segmentation de texture à contours « libres », décrit au Chapitre 4. Ce problème requiert en effet le formalisme généralisé décrit dans ce chapitre, car les observations y sont par nature multi-échelles, corrélées à la fois entre les échelles et spatialement, conduisant à une matrice de covariance riche. En outre, ces corrélations n'étant pas connues *a priori*, elles doivent être estimées directement sur les données.

Des simulations numériques, conduites sur dix réalisations de textures fractales synthétiques homogènes par morceaux, ont permis de montrer que la stratégie proposée pour la sélection des paramètres de régularisation amenait à des performances d'estimation et de segmentation très satisfaisantes, sans introduction de connaissance préalable ou nécessiter l'intervention d'un « expert ».

Références

- [1] A. C. Aitkin. On least squares and linear combination of observations. *Proceedings of the Royal Society of Edinburgh*, 55 :42–48, 1935. [128](#), [129](#)
- [2] R. Ammanouil, A. Ferrari, D. Mary, C. Ferrari, and F. Loi. A parallel and automatically tuned algorithm for multispectral image deconvolution. *Monthly Notices of the Royal Astronomical Society*, 490(1) :37–49, 2019. [131](#)
- [3] J.-F. Aujol, G. Gilboa, T. Chan, and S. Osher. Structure-texture image decomposition-modeling, algorithms, and parameter selection. *Int. J. Comp. Vis.*, 67(1) :111–136, 2006. [127](#)
- [4] H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*, volume 408. Springer, 2011. [130](#), [137](#), [145](#)
- [5] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1) :183–202, 2009. [130](#)
- [6] J. Berger. Minimax estimation of a multivariate normal mean under arbitrary quadratic loss. *J. Multivariate Anal.*, 6(2) :256–264, 1976. [127](#)

- [7] J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb) :281–305, 2012. [130](#)
- [8] J. Bergstra, D. Yamins, and D. D. Cox. Making a science of model search : Hyperparameter optimization in hundreds of dimensions for vision architectures. Atlanta, USA, June 16–21 2013. *Jmlr*. [131](#)
- [9] Q. Bertrand, Q. Klopfenstein, M. Blondel, S. Vaiter, A. Gramfort, and J. Salmon. Implicit differentiation of Lasso-type models for hyperparameter optimization, 2020. [131](#)
- [10] Å. Björck. *Numerical methods for least squares problems*. SIAM, 1996. [127](#)
- [11] T. Blu and F. Luisier. The SURE-LET approach to image denoising. *IEEE Trans. Image Process.*, 16(11) :2778–2786, 2007. [129](#)
- [12] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. 16(5) :1190–1208, 1995. [139](#)
- [13] J.-F. Cai, B. Dong, S. Osher, and Z. Shen. Image restoration : Total variation, wavelet frames, and beyond. *J. Amer. Math. Soc.*, 25 :1033–1089, May 2012. [127](#)
- [14] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imag. Vis.*, 40(1) :120–145, 2011. [130](#), [137](#)
- [15] T. Chan and J. Shen. Variational image inpainting. *Comm. Pure Applied Math.*, 58(5), May 2005. [127](#)
- [16] C. Chau, L. Duval, A. Benazza-Benyahia, and J.-C. Pesquet. A nonlinear Stein-based estimator for multichannel image denoising. *IEEE Trans. Signal Process.*, 56(8) :3855–3870, 2008. [129](#), [131](#)
- [17] P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In H. H. Bauschke, R. S. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz, editors, *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer-Verlag, New York, 2011. [130](#), [137](#)
- [18] L. Condat, D. Kitahara, A. Contreras, and A. Hirabayashi. Proximal splitting algorithms : Relax them all! *Preprint arXiv :1912.00137*, 2019. [130](#)
- [19] F. E. Curtis, T. Mitchell, and M. L. Overton. A BFGS-SQP method for nonsmooth, nonconvex, constrained optimization and its evaluation using relative minimization profiles. *Optim. Methods Softw.*, 32(1) :148–181, 2017. [138](#), [139](#), [149](#)
- [20] C.-A. Deledalle, F. Tupin, and L. Denis. Poisson NL means : Unsupervised non local means for Poisson noise. In *Proc. Int. Conf. Image Process.*, pages 801–804. IEEE, 2010. [131](#)
- [21] C.-A. Deledalle, S. Vaiter, J. Fadili, and G. Peyré. Stein unbiased gradient estimator of the risk (SUGAR) for multiple parameter selection. *SIAM J. Imaging Sci.*, 7(4) :2448–2487, 2014. [130](#), [131](#), [134](#), [136](#), [137](#), [149](#)
- [22] L. Desbat and D. Girard. The “minimum reconstruction error” choice of regularization parameters : Some more efficient methods and their application to deconvolution problems. *J. Sci. Comput.*, 16(6) :1387–1403, 1995. [130](#), [131](#)
- [23] D. L. Donoho and I. M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *J. American Statist. Ass.*, 90(432) :1200–1224, 1995. [129](#)
- [24] D. L. Donoho and J. M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *biometrika*, 81(3) :425–455, 1994. [127](#), [130](#), [131](#)

- [25] C. Dossal, M. Kachour, J. Fadili, G. Peyré, and C. Chesneau. The degrees of freedom of the Lasso for general design matrix. *Statistica Sinica*, pages 809–828, 2013. [133](#)
- [26] Y. C. Eldar. Generalized SURE for exponential families : Applications to regularization. *IEEE Trans. Signal Process.*, 57(2) :471–481, 2008. [129](#), [130](#), [133](#)
- [27] L. Elden. Algorithms for the regularization of ill-conditioned least squares problems. *BIT Numer. Math.*, 17(2) :134–145, 1977. [128](#)
- [28] L. C. Evans and R. F. Gariépy. *Measure theory and fine properties of functions*. Chapman and Hall/CRC, 2015. [130](#), [133](#), [137](#), [145](#)
- [29] N. P. Galatsanos and A. K. Katsaggelos. Methods for choosing the regularization parameter and estimating the noise variance in image restoration and their relation. *IEEE Trans. Image Process.*, 1(3) :322–336, 1992. [130](#)
- [30] A. Girard. A fast ‘Monte-Carlo cross-validation’ procedure for large least squares problems with noisy data. *Numerische Mathematik*, 56(1) :1–23, 1989. [130](#), [134](#)
- [31] G. H. Golub, M. Heath, and G. Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2) :215–223, 1979. [131](#)
- [32] G. H. Golub, P. C. Hansen, and D. P. O’Leary. Tikhonov regularization and total least squares. *SIAM J. Matrix Anal. Appl.*, 21(1) :185–194, 1999. [127](#)
- [33] P. C. Hansen and D. P. O’Leary. The use of the L-curve in the regularization of discrete ill-posed problems. *J. Sci. Comput.*, 14(6) :1487–1503, 1993. [127](#), [128](#)
- [34] H. M. Hudson. A natural identity for exponential families with applications in multiparameter estimation. *Ann. Stat.*, 6(3) :473–484, 1978. [129](#)
- [35] H. M. Hudson and T. C. M. Lee. Maximum likelihood restoration and choice of smoothing parameter in deconvolution of image data subject to Poisson noise. *Computational statistics & data analysis*, 26(4) :393–410, 1998. [129](#)
- [36] K. Kato. On the degrees of freedom in shrinkage estimation. *J. Multivariate Anal.*, 100(7) : 1338–1352, 2009. [133](#)
- [37] C. L. Lawson and R. J. Hanson. *Solving least squares problems*, volume 15. SIAM, 1995. [127](#)
- [38] Y. Le Montagner, E. D. Angelini, and J.-C. Olivo-Marin. An unbiased risk estimator for image denoising in the presence of mixed poisson–gaussian noise. *IEEE Trans. Image Process.*, 23 (3) :1255–1268, 2014. [129](#)
- [39] K. C. Li. From stein’s unbiased risk estimates to the method of generalized cross validation. *Ann. Stat.*, pages 1352–1377, 1985. [131](#)
- [40] F. Luisier, T. Blu, and M. Unser. A new SURE approach to image denoising : Interscale orthogonal wavelet thresholding. *IEEE Trans. Image Process.*, 16(3) :593–606, 2007. [131](#)
- [41] F. Luisier, T. Blu, and M. Unser. Image denoising in mixed poisson–gaussian noise. *IEEE Trans. Image Process.*, 20(3) :696–708, 2010. [129](#)
- [42] S. Mallat. *A Wavelet Tour of Signal Processing, Third Edition : The Sparse Way*. Academic Press, Inc., Orlando, FL, USA, 3rd edition, 2008. ISBN 0123743702, 9780123743701. [148](#)
- [43] M. Meyer and M. Woodroffe. On the degrees of freedom in shape-restricted regression. *Ann. Stat.*, pages 1083–1104, 2000. [129](#)

- [44] J. Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2006. [138](#), [139](#), [140](#)
- [45] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3) : 127–239, 2014. [130](#), [137](#)
- [46] B. Pascal, N. Pustelnik, P. Abry, M. Serres, and V. Vidal. Joint estimation of local variance and local regularity for texture segmentation. application to multiphase flow characterization. In *Proc. Int. Conf. Image Process.*, pages 2092–2096. IEEE, 2018. [127](#), [148](#)
- [47] B. Pascal, N. Pustelnik, and P. Abry. How joint fractal features estimation and texture segmentation can be cast into a strongly convex optimization problem? *Preprint arXiv :1910.05246 [math.OC]*, 2019. [127](#), [144](#), [148](#)
- [48] B. Pascal, S. Vaiteer, N. Pustelnik, and P. Abry. Automated data-driven selection of the hyperparameters for Total-Variation based texture segmentation. *Preprint arXiv :2004.09434 [stat.ML]*, 2020. [126](#)
- [49] J.-C. Pesquet, A. Benazza-Benyahia, and C. Chau. A SURE approach for digital signal/image deconvolution problems. *IEEE Trans. Signal Process.*, 57(12) :4616–4632, 2009. [129](#)
- [50] N. Pustelnik, A. Benazza-Benahyia, Y. Zheng, and J.-C. Pesquet. Wavelet-based image deconvolution and reconstruction. *Wiley Encyclopedia of Electrical and Electronics Engineering*, Feb. 2016. doi : <https://doi.org/10.1002/047134608X.W8294>. [127](#)
- [51] S. Ramani, T. Blu, and M. Unser. Monte-carlo SURE : A black-box optimization of regularization parameters for general denoising algorithms. *IEEE Trans. Image Process.*, 17(9) :1540–1554, 2008. [129](#), [130](#), [134](#)
- [52] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D : nonlinear phenomena*, 60(1-4) :259–268, 1992. [127](#), [128](#)
- [53] X. Shen and J. Ye. Adaptive model selection. *J. Am. Stat. Assoc.*, 97(457) :210–221, 2002. [130](#)
- [54] C. M. Stein. Estimation of the mean of a multivariate normal distribution. *Ann. Stat.*, pages 1135–1151, 1981. [126](#), [129](#)
- [55] T. Strutz. *Data fitting and uncertainty : A practical introduction to weighted least squares and beyond*. Vieweg and Teubner, 2010. [128](#)
- [56] A. M. Thompson, J. C. Brown, J. W. Kay, and D. M. Titterton. A study of methods of choosing the smoothing parameter in image restoration by regularization. *IEEE Trans. Pattern Anal. Match. Int.*, (4) :326–339, 1991. [130](#), [131](#)
- [57] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society : Series B (Methodological)*, 58(1) :267–288, 1996. [127](#)
- [58] R. Tibshirani and L. Wasserman. Stein's unbiased risk estimate. *Course notes from "Statistical Machine Learning"*, pages 1–12, 2015. [129](#)
- [59] R. J. Tibshirani and J. Taylor. Degrees of freedom in lasso problems. *Ann. Stat.*, 40(2) :1198–1232, 2012. [129](#), [133](#)
- [60] A. Tikhonov. Tikhonov regularization of incorrectly posed problems. *Soviet Mathematics Doklady*, 4 :1624–1627, 1963. [127](#), [128](#)
- [61] S. Vaiteer, C. Deledalle, J. Fadili, G. Peyré, and C. Dossal. The degrees of freedom of partly smooth regularizers. *Ann. Inst. Stat. Math*, 69(4) :791–832, 2017. [133](#), [145](#)

- [62] C. Vonesch, S. Ramani, and M. Unser. Recursive risk estimation for non-linear image deconvolution with a wavelet-domain sparsity constraint. In *Proc. Int. Conf. Image Process.*, pages 665–668. IEEE, 2008. [130](#)
- [63] H. Wendt. *Contributions of Wavelet Leaders and Bootstrap to Multifractal Analysis : Images, Estimation Performance, Dependence Structure and Vanishing Moments. Confidence Intervals and Hypothesis Tests*. PhD thesis, Ecole Normale Supérieure de Lyon, 2008. [143](#), [146](#)
- [64] X. Xie, S. Kou, and L. D. Brown. Sure estimates for a heteroscedastic hierarchical model. *J. American Statist. Ass.*, 107(500) :1465–1479, 2012. [129](#)

Chapitre 6

Applications à la physique

« Je m'étonne de l'existence du monde. »

L. Wittgenstein

Sommaire

6.1 Motivations	162
6.2 Frottement solide en environnement faiblement contraint	163
6.2.1 Présentation du problème	163
6.2.2 Dispositif expérimental et acquisition de données	163
6.2.3 Estimation « linéaire par morceaux »	164
6.2.4 Sélection de paramètre par minimisation de l'estimateur SURE	165
6.2.5 Analyse des résultats et discussion	167
6.3 Écoulements multiphasiques en milieu poreux	169
6.3.1 Présentation du problème	169
6.3.2 Dispositif expérimental et acquisition de données	169
6.3.3 Segmentation de texture pour les écoulements multiphasiques	170
6.3.3.1 Cadre d'étude	170
6.3.3.2 Méthodes de segmentation comparées	171
6.3.3.3 Discussion des résultats	172
6.3.3.4 Modélisation d'écoulement à partir de textures synthétiques	172
6.3.3.5 Mesure des quantités physiques	174
6.3.4 Réglage automatique des paramètres de régularisation	176
6.3.4.1 Méthodes comparées	176
6.3.4.2 Covariance et paramètres des estimateurs SURE et SUGAR	177
6.3.4.3 Recherche sur grille <i>versus</i> réglage automatique des paramètres	178
6.3.4.4 Résultats des méthodes de segmentation automatiques	179
6.4 Bilan	183

La première application aux signaux issus d'expériences de frottement solide est le fruit d'une collaboration entre Nelly Pustelnik, Valérie Vidal, Patrice Abry et Jean-Christophe Gémard durant le stage de Jules Colas. Les données expérimentales ont été acquises par Jules Colas et nous proposons de les étudier au moyen du formalisme de sélection automatique développé au Chapitre 5.

L'application des outils de segmentation de textures aux images d'écoulements multiphasiques présentée à la Section 6.3 est le fruit d'une collaboration au long cours avec des chercheurs de l'équipe Matière et Complexité du Laboratoire de Physique à l'initiative de Nelly Pustelnik et Valérie Vidal. Les premières images d'expériences sur les écoulements multiphasiques proviennent du travail de thèse de Marion Serres. Une seconde série de données est le fruit du travail de recherche

de Thomas Busser, ATER au Laboratoire de Physique, avec qui nous avons collaboré pendant plusieurs mois pour le traitement de ces images texturées. L'analyse de ces images d'écoulement a été poursuivie dans le cadre du stage de Loris Helmlinger, sous la direction de Nelly Pustelnik et Barbara Pascal. Enfin, les données les plus récentes ont été réalisées par Eden Dzik durant son stage au Laboratoire de Physique et sont toujours en cours d'analyse par Nelly Pustelnik, Valérie Vidal et Barbara Pascal.

L'étude d'une première série d'images d'écoulements multiphasiques a fait l'objet de deux communications en traitement du signal et des images [15, 16], et d'une publication en physique non linéaire [5]. La communication [16] a été étendue en un article [17], dans lequel le problème de débruitage de signaux issus d'expériences de frottement solide et le problème de segmentation d'images d'écoulements sont formulés grâce à une minimisation de fonctionnelle, auquel est appliqué le formalisme de réglage automatique des paramètres de régularisation du Chapitre 5. Une boîte à outils, implémentée en langage MATLAB, est accessible librement en ligne¹ et permet de reproduire les résultats présentés sur des signaux comme sur des images au comportement constant ou linéaire par morceaux.

6.1 Motivations

De nombreux signaux ou images collectés en physique expérimentale, dans l'étude des matériaux ou en mécanique des fluides par exemple, peuvent être considérés, en première approximation, comme homogènes, i.e., constants, linéaires, . . . , *par morceaux*. La détection des différentes phases, sur les signaux, ou régions, sur les images, est cruciale pour pouvoir extraire correctement les quantités physiques d'intérêt mais constitue souvent un défi, notamment à cause de la présence non négligeable de bruit, provenant du dispositif de mesure ou intrinsèquement du phénomène étudié. Particulièrement lorsque la puissance du signal par rapport au bruit est faible, éliminer le bruit tout en conservant les discontinuités franches entre les phases ou régions est une tâche ardue. Or, les méthodes favorisant la parcimonie que nous avons abondamment décrites dans cette thèse constituent des outils particulièrement appropriés pour l'analyse de ce type de données. En outre, le formalisme de sélection automatique des hyperparamètres développé au Chapitre 5 rend l'utilisation de telles méthodes possibles, d'une part sans connaissances préalables sur les propriétés techniques des méthodes proposées, et d'autre part sur de gros volumes de données.

Nous allons nous intéresser dans ce chapitre à deux problèmes de physique expérimentale : d'une part, le phénomène de « coller-glisser », rencontré en physique des matériaux, voir un exemple de signal de force à analyser en Figure 6.1a, et d'autre part, la mécanique des écoulements multiphasiques, à la frontières de domaines aussi variés que la géologie et la catalyse chimique, voir un exemple d'image d'écoulement zoomée à la Figure 6.1b. Ces deux exemples ont en commun de pouvoir être décrits par leur aspect homogène par morceaux, qui reflète les mécanismes physiques sous-jacents. L'étude du frottement solide requiert d'identifier correctement les phases de « coller » et les phases de « glisser », chacune associée en première approximation à un comportement *linéaire*, tandis que l'étude des écoulements multiphasiques nécessite de localiser les deux phases en présence, liquide et gazeuse, chacune associée à une *texture* homogène. Les signaux et images homogènes par morceaux sont rencontrés dans des domaines très variés de la physique non linéaire, tels que l'étude du renversement du champ magnétique dans les dynamos turbulentes [4], l'intermittence dans les matériaux granulaires [9], la détection des protéines d'ADN durant leur translocation au travers d'un pore du noyau [1], . . . auxquels on pourrait envisager d'appliquer les méthodes présentées.

1. <https://github.com/bpascal-fr/stein-piecewise-filtering>

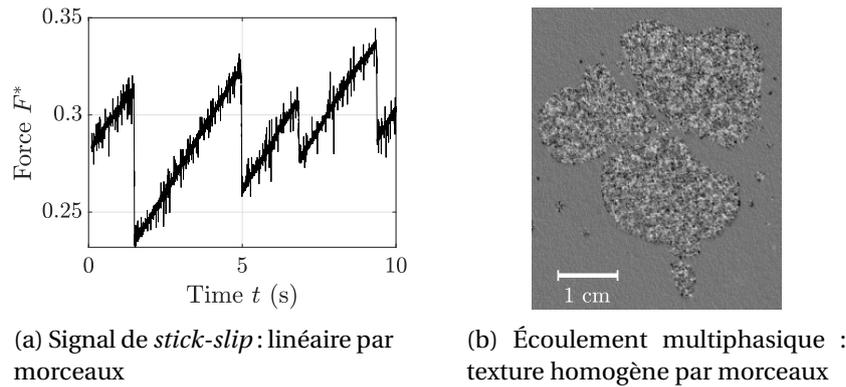


FIGURE 6.1 – **Données expérimentales illustrant des phénomènes homogènes par morceaux en physique non linéaire.** (a) : Force motrice normalisée $F^* = k\Delta x/(mg)$ (voir Section 6.2.2) en fonction du temps t dans une expérience de frottement solide en « coller-glisser ». (b) : Image directe d'une bulle de gaz dans un écoulement liquide au travers d'un milieu poreux, le gaz correspond à la région la plus texturée, tandis que le liquide correspond aux régions d'aspect plus régulier.

6.2 Frottement solide en environnement faiblement contraint

6.2.1 Présentation du problème

L'étude du frottement solide constitue un premier champ d'application des outils favorisant la parcimonie en physique expérimentale. Les expériences de frottement solide visent non seulement à étudier les caractéristiques des matériaux mais également à comprendre la dynamique de systèmes comportant des surfaces en contact, elles permettent notamment de modéliser et de prédire les tremblements de terre [14]. Lorsque deux surfaces plates en contact solide sont contraintes de glisser l'une sur l'autre, la force de cisaillement subie à la surface de contact présente généralement une allure en dents-de-scie, telle que présentée en Figure 6.1a. Le signal mesurant la force consiste alors en une alternance de phases de croissances linéaires lentes, correspondant à l'emmagasinement de l'énergie élastique par le système mobile, tandis que les surfaces de contact restent immobiles l'une par rapport à l'autre, suivies de soudain décrochements, correspondant à une brutale libération d'énergie lorsque les surfaces glissent l'une par rapport à l'autre [3]. La phase de décrochement consiste en une succession d'arcs de sinus, pouvant être approximée au premier ordre par un comportement linéaire.

Lorsque les deux solides en contact sont fortement pressés l'un contre l'autre, les deux phases de « coller » et « glisser » sont facilement identifiables. Cependant, des pressions de confinement trop importantes provoquent souvent des dommages sur les surfaces considérées, ce qui est une limitation majeure pour l'étude des phénomènes microscopiques mis en jeu, ces dommages pouvant conduire à la destruction de l'échantillon de matériau étudié. Par conséquent, afin d'étudier précisément les propriétés de frottement des matériaux, il est nécessaire de réaliser des expériences à faible pression.

6.2.2 Dispositif expérimental et acquisition de données

Une expérience classique pour étudier le frottement solide consiste à tracter une masse m , appelée un « mobile », sur un substrat, au moyen d'un ressort de raideur k , à la vitesse V . Le dispositif expérimental utilisé est illustré en Figure 6.2 où nous reproduisons le schéma de [7, Figure 2] avec l'aimable autorisation des auteurs. Le lecteur intéressé trouvera une plus ample description du dispositif expérimental dans [7]. La dynamique du mobile est appréhendée *via* la mesure d'un signal de force au point de contact entre le ressort et le mobile.

Parmi les différents régimes décrits en frottement solide, on distingue : le régime de « coller-

glisser», caractérisé par un signal en dents-de-scie alternant croissance lente et chute brutale, le régime inertiel, dans lequel le signal est périodique et ressemble à une courbe de sinus, et le régime de glissement continu, caractérisé par un signal quasi-constant auquel est superposé du bruit [3]. L'apparition d'un infime glissement vers l'avant du mobile juste avant une phase de « glisser», peut également entraîner une modification de la forme du signal. Le défi consiste à établir un *diagramme des régimes* décrivant, et donc, prédisant, la dynamique du système en fonction des paramètres m (masse), k (raideur), V (vitesse de forçage). Si, pour des masses importantes, l'identification des différents régimes est facile, les expériences effectuées à faible pression, ce qui est nécessaire pour éviter l'usure des surfaces, sont plus complexes à étudier, en raison de la présence d'un niveau de bruit élevé. Ce contexte requiert donc la mise en œuvre de méthodes de traitement du signal plus sophistiquées. Nous nous concentrons ici sur le débruitage *via* une pénalisation favorisant la linéarité par morceaux.

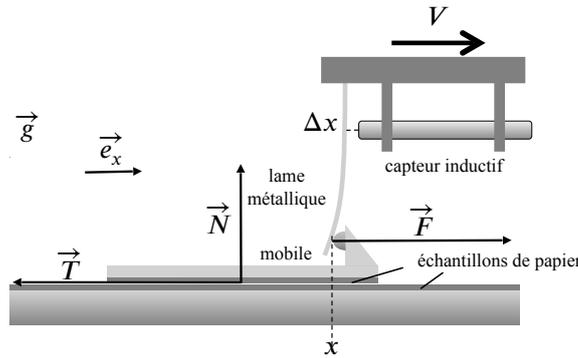


FIGURE 6.2 – Dispositif expérimental pour l'étude du frottement solide, \vec{g} désigne l'accélération de pesanteur, \vec{N} la normale et \vec{T} la tangente par rapport à la surface de contact. *Reproduction de [7, Figure 2] avec l'autorisation des auteurs.*

Toutes les signaux de force expérimentaux étudiés proviennent des travaux sur le frottement solide de [7], et ont été réalisées au Laboratoire de Physique de l'ENS de Lyon. Elles consistent en la traction d'un mobile de masse de $m = 30,7$ g, et de surface de contact de 9×6 cm² sur un substrat solide. Les deux surfaces en contact sont des échantillons de papier Canson[®], caractérisées par leur grain. Un ressort en porte-à-faux, i.e., une lame métallique de raideur k , comprise entre 168 et 2254 N/m, est tirée à vitesse constante V , comprise entre 42 et 7200 $\mu\text{m/s}$, et est en contact avec le mobile *via* une bille métallique collée sur ce dernier, assurant un contact ponctuel et un mouvement libre au niveau du point de contact. La dynamique du mobile est alors quantifiée au moyen de la déviation Δx de la lame, mesurée par un capteur à induction Baumer, IPRM 1219505/S14. Dans toutes les expériences la masse m est maintenue constante. Plusieurs valeurs des paramètres k et V sont explorées. Pour chaque expérience, le signal de force normalisée F^* , obtenu à partir de la déviation de la lame *via* $F^* = k\Delta x/(mg)$, où $g = 9.81$ m/s⁻² est l'accélération de pesanteur, est enregistré au cours du temps à une fréquence d'échantillonnage de $f_e = 2$ kHz. Les enregistrements à étudier sont alors de tailles variables, allant de 4.5×10^3 à 7.5×10^5 points expérimentaux.

6.2.3 Estimation « linéaire par morceaux »

Sur le même modèle que le débruitage par Variation Totale, présenté à la Section 2.1.1.6, permettant d'imposer la *constance* par morceaux, il est possible d'utiliser la norme ℓ_1 pour construire une méthode de régularisation favorisant la *linéarité* par morceaux. Pour cela, les signaux, que nous noterons

$$\mathbf{y} = \{F^*(nT_e), n = 0, \dots, N-1\} \in \mathbb{R}^N, \quad \text{avec } T_e \triangleq \frac{1}{f_e}, \quad (6.1)$$

où T_e désigne la période d'échantillonnage, sont analysés au moyen d'un formalisme d'optimisation pouvant s'écrire sous la forme du modèle (5.3) avec $\Phi = \mathbf{I}$, l'opérateur identité, $\|\cdot\| = \|\cdot\|_1$ la norme ℓ_1 , et $\mathbf{U}_\Lambda = \lambda \mathbf{L}$, avec $\lambda > 0$ et \mathbf{L} l'opérateur laplacien [7], défini par

$$\forall n \in \{1, \dots, N\}, \quad (\mathbf{L}\mathbf{x})(n) = \begin{cases} x(n+1) - 2x(n) + x(n-1) & \text{si } n \notin \{1, N\} \\ 0 & \text{sinon.} \end{cases} \quad (6.2)$$

Le débruitage « linéaire par morceaux » s'écrit donc sous la forme du problème suivant

$$\hat{\mathbf{x}} = \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^N} \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda \|\mathbf{L}\mathbf{x}\|_1. \quad (6.3)$$

La résolution de ce problème peut être effectuée de manière efficace grâce au schéma primal-dual accéléré présenté à l'Algorithme 12, particularisé au Problème (6.3), pour lequel le module de forte convexité est $\mu = 1$.

Le compromis entre la fidélité aux observations \mathbf{y} et la régularisation imposant la linéarité par morceaux est piloté par le paramètre de régularisation $\lambda > 0$. Dans [7], la tâche fastidieuse de déterminer le paramètre de régularisation λ optimal a été réalisée par une inspection visuelle systématique des signaux et a conduit au choix « expert » de $\hat{\lambda}^{\text{expert}} = 0.8$, commun à l'ensemble des expériences. Trois exemples de signaux bruités, \mathbf{y} , sont tracés en gris à la Figure 6.3. Les estimées linéaires par morceaux associées sont tracées en rouge et on peut remarquer que les transitions entre les régimes de « coller » et de « glisser » sont effectivement détecté avec une précision satisfaisante.

Néanmoins, le réglage « expert » du paramètre de régularisation possède plusieurs inconvénients. D'une part, une personne disposant d'une bonne connaissance préalable sur le phénomène étudié doit inspecter minutieusement des estimées débruitées avec différents paramètres de régularisation λ , et ce, *a priori*, pour chaque configuration (k, V) du diagramme des régimes. Le temps nécessaire à cette tâche limite le nombre de signaux pouvant être étudiés et par conséquent la précision avec laquelle seront connues les transitions entre différents régimes dans le plan (k, V) . D'autre part, un regard « expert » peut induire un biais de confirmation, faussant alors le choix du paramètre de régularisation λ . En outre, il peut exister une grande variabilité dans les paramètres optimaux choisis par différents experts, voire entre plusieurs réglages réalisés par un même expert. Afin de pallier ces problèmes nous proposons de particulariser le formalisme de sélection automatique d'hyperparamètres, présenté au Chapitre 5, au Problème (6.3) pour l'analyse de signaux de « coller-glisser ».

6.2.4 Sélection de paramètre par minimisation de l'estimateur SURE

Ne connaissant pas le « véritable » signal, i.e., le signal non bruité, la sélection des paramètres de régularisation optimaux ne peut se faire par la minimisation de l'erreur quadratique *exacte*. Deux stratégies s'appuyant sur les estimateurs de Stein introduits au Chapitre 5 sont donc proposées pour la sélection du paramètre de régularisation du Problème (6.3). D'une part la minimisation sur une grille de l'estimateur SURE, $\hat{R}_{v,\varepsilon}(\mathbf{y}; \lambda)$, de l'erreur quadratique, d'autre part la sélection automatique au moyen d'un algorithme de quasi-Newton faisant usage de l'estimateur du gradient de l'erreur, $\partial_\lambda \hat{R}_{v,\varepsilon}(\mathbf{y}; \lambda)$.

Le signal de force, i.e., l'observation \mathbf{y} , est considéré comme la superposition du signal pur de « coller-glisser », parfaitement linéaire par morceaux, noté $\bar{\mathbf{x}}$ et d'un bruit additif indépendant de $\bar{\mathbf{x}}$, de sorte que

$$\mathbf{y} = \bar{\mathbf{x}} + \boldsymbol{\zeta}. \quad (6.4)$$

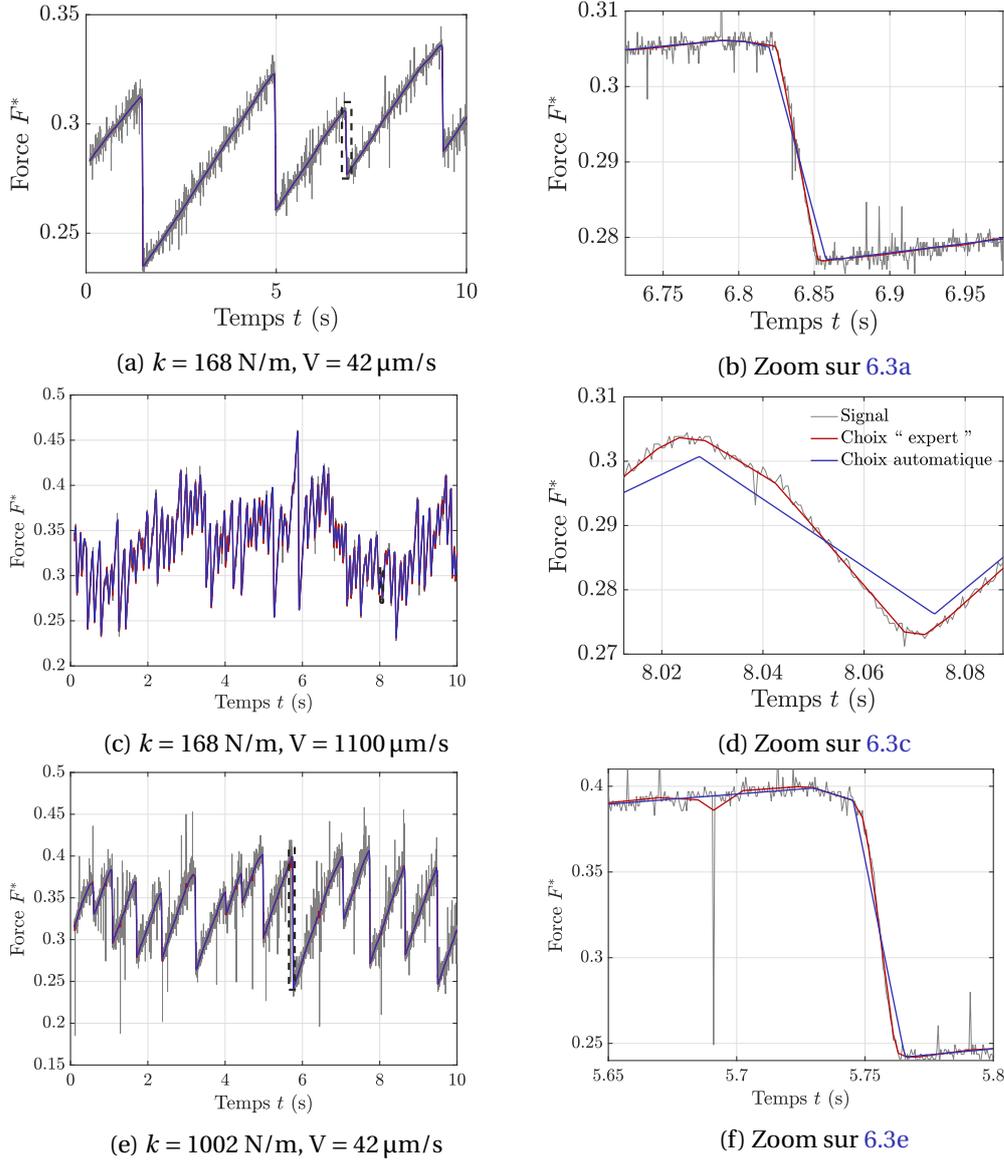


FIGURE 6.3 – **Phénomène de « coller-glisser » : évolution du signal de force.** Données expérimentales (en gris) pour trois configurations expérimentales. Débruitage « linéaire par morceaux », avec choix « expert », $\hat{\lambda}^{\text{expert}}$ du paramètre de régularisation (en rouge) et choix automatique $\hat{\lambda}_{v,\varepsilon}^{\text{BFGS}}$ (en bleu).

On suppose également que le bruit ζ est gaussien, indépendant, identiquement distribué de variance σ^2 . Pour le calcul des estimateurs de Stein, $\hat{R}_{v,\varepsilon}(\mathbf{y}; \lambda)$ et $\partial_\lambda \hat{R}_{v,\varepsilon}(\mathbf{y}; \lambda)$, l'hypothèse de bruit gaussien i.i.d. et l'application de la Formule (5.82) amènent à fixer un pas de différences finies

$$v = \frac{2\sqrt{\hat{\sigma}^2}}{N^{0,3}}, \quad (6.5)$$

où $\hat{\sigma}$ désigne l'écart-type estimé du bruit ζ . Ne disposant pas d'informations supplémentaires sur la variance du bruit, la variance estimée, $\hat{\sigma}^2$, est assimilée à la variance du signal et calculée à l'aide d'un estimateur de variance standard. Quant au vecteur de Monte Carlo, $\varepsilon \in \mathbb{R}^N$, il est généré aléatoirement selon une distribution gaussienne i.i.d., de moyenne nulle.

Dans un premier temps, l'estimateur SURE, $\hat{R}_{v,\varepsilon}(\mathbf{y}; \lambda)$, est calculé au moyen de l'Algorithme 14 pour quinze valeurs de λ , logarithmiquement réparties entre 10^{-1} et 10^3 . Trois exemples de courbes de l'erreur quadratique estimée $\hat{R}_{v,\varepsilon}(\mathbf{y}; \lambda)$ sont tracées en noir à la Figure 6.4. Le paramètre optimal par recherche sur grille, noté λ^{grille} , est défini comme le minimum de $\hat{R}_{v,\varepsilon}(\mathbf{y}; \lambda)$ sur les quinze

valeurs discrètes explorées, et indiquée par le symbole « + » sur la Figure 6.4. La valeur optimale $\hat{\lambda}_{\text{grille}}$, pour chacune des configurations (k, V) explorées, est reportée dans le Tableau 6.1a.

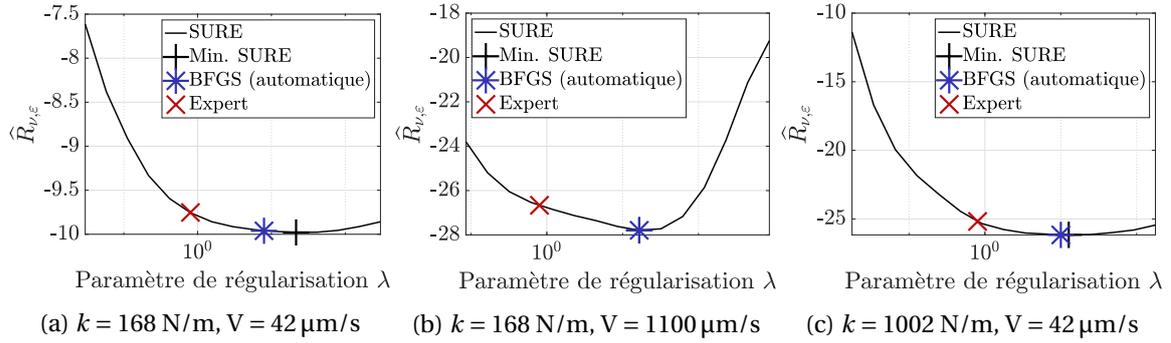


FIGURE 6.4 – **Erreur quadratique estimée** $\hat{R}_{v,\epsilon}(\mathbf{y}, \lambda)$. Recherche sur grille *versus* réglage automatique pour la sélection du paramètre de régularisation λ .

Dans un second temps, l’Algorithme de quasi-Newton 15 pour la minimisation automatique de l’erreur quadratique estimée, $\hat{R}_{v,\epsilon}(\mathbf{y}; \lambda)$, particularisé au Problème (6.3), est mis en œuvre pour chaque couple (k, V) . Le paramètre de régularisation sélectionné automatiquement est noté $\hat{\lambda}^{\text{BFGS}}$ et indiqué par le symbole « * » à la Figure 6.4. Les valeurs de $\hat{\lambda}^{\text{BFGS}}$ pour chaque configuration (k, V) sont présentées dans le Tableau 6.1b. Pour comparaison, le paramètre λ^{expert} est également représenté sur la Figure 6.4, par le symbole « x ».

$\hat{\lambda}_{\text{grille}}(k, V)$ ($\hat{R}_{v,\epsilon}$)	k (N/m)			$\hat{\lambda}^{\text{BFGS}}(k, V)$ ($\hat{R}_{v,\epsilon}$)	k (N/m)			
	168	1002	2254		168	1002	2254	
V ($\mu\text{m/s}$)	42	21,6 (-10)	12,7 (-26,2)	23,1 (-4,3)	42	7,9 (-10)	10 (-26,2)	0,2 (-4)
	1100	16,6 (-27,8)	3,7 (-17,7)	76,6 (-15,1)	1100	16,5 (-27,8)	3,4 (-17,7)	2,2 (-20,7)
	4300	8,8 (-6,9)	6,2 (-15)	2,6 (-28,4)	4300	9,1 (-6,9)	4,9 (-15,1)	3,2 (-28,6)

(a) Recherche sur grille (b) Sélection automatique

TABEAU 6.1 – **Choix du paramètre de régularisation optimal.** Paramètres de régularisation pour le Problème (6.3) d’estimation linéaire par morceaux, obtenus par recherche sur grille *versus* sélection automatique, pour différentes valeurs de la raideur k et de la vitesse de forçage V . La valeur de l’estimateur SURE, $\hat{R}_{v,\epsilon}(\mathbf{y}; \lambda)$, atteint pour le paramètre optimal $\hat{\lambda}_{\text{grille}}(k, V)$ ou $\hat{\lambda}^{\text{BFGS}}(k, V)$ est indiquée entre parenthèses afin de comparer l’efficacité de la minimisation sur grille (à l’aide de l’Algorithme 13) ou par méthode de quasi-Newton (Algorithme 15).

6.2.5 Analyse des résultats et discussion

En étudiant la Figure 6.4, on constate, tout d’abord, que les paramètres $\hat{\lambda}_{\text{grille}}$ et $\hat{\lambda}^{\text{BFGS}}$ sont parfaitement en accord. D’autre part, s’ils peuvent s’écarter du choix $\hat{\lambda}^{\text{expert}} = 0.8$ d’un ordre de grandeur, ils correspondent à des valeurs de l’erreur quadratique estimée $\hat{R}_{v,\epsilon}(\mathbf{y}; \lambda)$ qui diffèrent de 2% (Figure 6.4a) à 7% (Figure 6.4c) par rapport à la valeur atteinte pour $\hat{\lambda}^{\text{expert}}$. Cela indique que les estimées obtenues avec les réglages $\hat{\lambda}^{\text{expert}}$, $\hat{\lambda}_{\text{grille}}$ ou $\hat{\lambda}^{\text{BFGS}}$ sont de qualité équivalente, lorsque celle-ci est mesurée par l’erreur quadratique. De fait, les estimées régularisées avec $\hat{\lambda}^{\text{expert}}$ ou $\hat{\lambda}^{\text{BFGS}}$, comparées à la Figure 6.3, partagent la même allure générale et la même alternance de régimes de « coller » et de « glisser ». L’estimateur SURE est donc, non seulement, un bon critère de qualité de la solution, mais également, un critère dont la minimisation automatique *via* l’Algorithme 15, de type BFGS, est efficace. Ce point est particulièrement satisfaisant, puisque, sans

l'utilisation d'une quelconque connaissance préalable sur les données, exceptée l'hypothèse de linéarité par morceaux, et en faisant une hypothèse standard de bruit gaussien i.i.d., l'Algorithme 15 permet d'obtenir des estimées au comportement parfaitement cohérent avec la physique du problème, et ce, de manière complètement non supervisée.

Ces conclusions sont confortées par les résultats sur l'ensemble des configurations, présentés aux Tableaux 6.1a et 6.1b, où sont comparés à la fois les paramètres de régularisation obtenus par recherche sur grille ou par sélection automatique, et également les valeurs de l'estimateur SURE atteintes. On constate, dans une large majorité de configurations, un bon accord entre $\hat{\lambda}^{\text{grille}}$ et $\hat{\lambda}^{\text{BFGS}}$. Les écarts observés entre les paramètres sélectionnés par recherche sur grille et minimisation automatique, comme par exemple pour $k = 2254$ et $V = 42$ où $\hat{\lambda}^{\text{grille}} = 23,1$ et $\hat{\lambda}^{\text{BFGS}} = 0,2$, s'expliquent par le comportement de $\hat{R}_{v,\epsilon}(\mathbf{y}; \lambda)$ au voisinage de son minimum. En effet, pour $k = 2254$ et $V = 42$, la comparaison de $\hat{R}_{v,\epsilon}(\mathbf{y}; \hat{\lambda}^{\text{grille}}) = -4,3$ et $\hat{R}_{v,\epsilon}(\mathbf{y}; \hat{\lambda}^{\text{BFGS}}) = -4$ montre que la fonction $\lambda \mapsto \hat{R}_{v,\epsilon}(\mathbf{y}; \lambda)$ varie très peu sur deux ordres de grandeur. La fonction minimisée étant très plate autour de son minimum, la convergence de l'Algorithme 15 est empêchée. En outre, les Tableaux 6.1a et 6.1b montrent que la recherche sur grille et la sélection automatique conduisent à des paramètres optimaux distincts d'une configuration à l'autre. Ces résultats soulignent la capacité d'adaptation fine aux données de ces deux méthodes, qui serait pour un réglage « expert », sinon impossible, du moins particulièrement coûteux en temps et en efforts.

Le paramètre de régularisation sélectionné automatiquement, $\hat{\lambda}^{\text{BFGS}}$, étant dans la grande majorité des cas plus grand que le paramètre « expert », $\hat{\lambda}^{\text{expert}}$, les signaux débruités avec un réglage automatique du paramètre λ (en rouge sur la Figure 6.4) sont plus fortement régularisés que ceux correspondant au choix « expert » (en bleu sur la Figure 6.4). Ainsi, on constate à la Figure 6.3f, que le signal débruité associé au choix « expert », en rouge, conserve certaines discontinuités que l'on peut raisonnablement attribuer au bruit et non au phénomène physique d'intérêt, comme par exemple autour de $t = 5.7$ s, qui sont éliminées correctement lors du débruitage avec le réglage automatique $\hat{\lambda}^{\text{BFGS}}$, représenté en bleu. Cela illustre le bénéfice que l'on peut tirer d'un réglage automatique capable de s'adapter à chaque signal.

De légères différences entre le signal régularisé obtenu avec $\hat{\lambda}^{\text{expert}}$ (en rouge) et celui obtenu avec $\hat{\lambda}^{\text{BFGS}}$ (en bleu) sont visibles sur les Figures 6.3b et 6.3d, notamment dans les phases de « glisser », i.e., de décroissance rapide. Afin de décider lequel est le plus pertinent, il est nécessaire de revenir à une analyse détaillée du frottement solide. La phase de « coller », produit un signal de force normalisée F^* qui croît *exactement* linéairement. La phase de « glisser », bien qu'elle puisse être décrite en première approximation comme une décroissance linéaire abrupte, consiste en réalité en une succession d'arcs de sinusoides. Ainsi, si le signal régularisé avec le choix expert, $\hat{\lambda}^{\text{expert}}$, épouse mieux les données expérimentales en concaténant une série de courts segments linéaires, il ne capte pas la physique sous-jacente. Le signal régularisé obtenu par sélection automatique, i.e., avec $\hat{\lambda}^{\text{BFGS}}$, a une allure en dents-de-scie qui pourrait permettre une meilleur localisation des transitions entre phase de « coller » et phase de « glisser », une information cruciale pour la compréhension des phénomènes physiques à l'œuvre et l'établissement du diagramme des régimes.

En conclusion, le choix d'un réglage « expert » ou d'une sélection automatique du paramètre de régularisation λ apparaissant dans le Problème (6.3) nécessite de prendre en compte à la fois la faisabilité d'un réglage « expert », en terme de temps requis, de biais de subjectivité ou de manque de reproductibilité, et le degré de précision souhaité pour les signaux régularisés afin de pouvoir extraire de manière fiable les quantités physiques d'intérêt.

6.3 Écoulements multiphasiques en milieu poreux

6.3.1 Présentation du problème

Un second champ d'application des outils présentés dans cette thèse est l'étude des écoulements multiphasiques en milieu poreux. Comprendre et prédire la dynamique des écoulements multiphasiques est un problème central en géologie, pour la décontamination des sols ou le captage et le stockage du CO₂, mais aussi dans l'industrie, pour l'amélioration de l'extraction du pétrole ou la catalyse hétérogène [18, 11, 10, 2]. Ces processus, pour la plupart, font intervenir un écoulement gaz-liquide au travers d'un milieu poreux. La quantification de l'aire de contact entre les phases liquide et gazeuse, où ont lieu les réactions chimiques, est alors d'une importance capitale pour analyser et prédire l'efficacité des procédés mis en jeu [12].

Traditionnellement, de telles études sont conduites grâce à des lits de billes compacts, bien caractérisés. Néanmoins, de récentes expériences [21, 15] se sont intéressées à des matériaux innovants, comme les mousses solides à cellules ouvertes, particulièrement prometteuses pour des applications industrielles du fait de leur grande porosité. En effet, la pression induite à l'intérieur du réacteur est alors plus faible, et entraîne donc une diminution du coût énergétique. Dans de telles expériences, de l'eau et de l'air s'écoulent simultanément au travers de la mousse. La caractérisation de la dynamique spatio-temporelle de tels écoulements multiphasiques est en soi un problème de traitement d'image, où le défi consiste à distinguer précisément l'écoulement liquide des bulles de gaz. En effet, même lorsqu'une visualisation directe est possible, et bien que le liquide et le gaz considérés soient transparents, la présence de la mousse solide conduit à des images à la fois texturées et peu contrastées, rendant l'extraction des frontières gaz-liquide difficile.

Les méthodes issues de la morphologie mathématique utilisées jusqu'ici pour discriminer les phases gazeuse et liquide [20] présentent de sérieuses limitations : la nécessité de fixer arbitrairement les seuils utilisés pour la binarisation gaz-liquide, des contours de bulles irréguliers incompatibles avec les attendus physiques, et enfin la non détection des bulles les plus petites. En outre, les avancées récentes en imagerie haute fréquence et haute résolution fournissent des images de taille élevée et des jeux de données importants, soulevant des problèmes de mémoire et de coût de calcul.

Nous allons nous concentrer sur la segmentation gaz-liquide dans les images texturées issues de l'étude des écoulements multiphasiques en milieu poreux au travers d'expériences réalisées au Laboratoire de Physique de l'ENS de Lyon. En première approximation, le liquide et le gaz seront considérés comme deux textures fractales homogènes, telles que décrites au Chapitre 3, ouvrant la voie à l'utilisation des méthodes de segmentation à contours « libres » et à contours « co-localisés » proposées au Chapitre 4. L'objectif final est de détecter les interfaces gaz-liquide de manière suffisamment précise pour pouvoir fournir une estimation fiable de leur longueur, critère physique déterminant dans les phénomènes de catalyse.

6.3.2 Dispositif expérimental et acquisition de données

Les expériences d'écoulements multiphasiques gaz-liquide au travers d'un milieu poreux ont été réalisées dans une cellule de Hele-Shaw², orientée verticalement, quasi bidimensionnelle de hauteur 410 mm, de largeur 210 mm et d'épaisseur 1,75 mm (voir la Figure 6.5 ou [6] pour un schéma descriptif de l'expérience). Le milieu poreux est constitué d'une mousse solide à cellules ouvertes faites d'un alliage NiCrFeAl (Alantum), avec un diamètre de pore typique de 580 μm. Le gaz et le liquide sont injectés à débit constant au bas de la cellule, par une fente homogène pour

2. Une cellule de Hele-Shaw est composée de deux plaques de verres parallèles très rapprochées entre lesquelles sont injectés un ou plusieurs fluides, dans le but de réaliser un écoulement quasi bidimensionnel.

le liquide, et *via* neuf injecteurs pour le gaz.

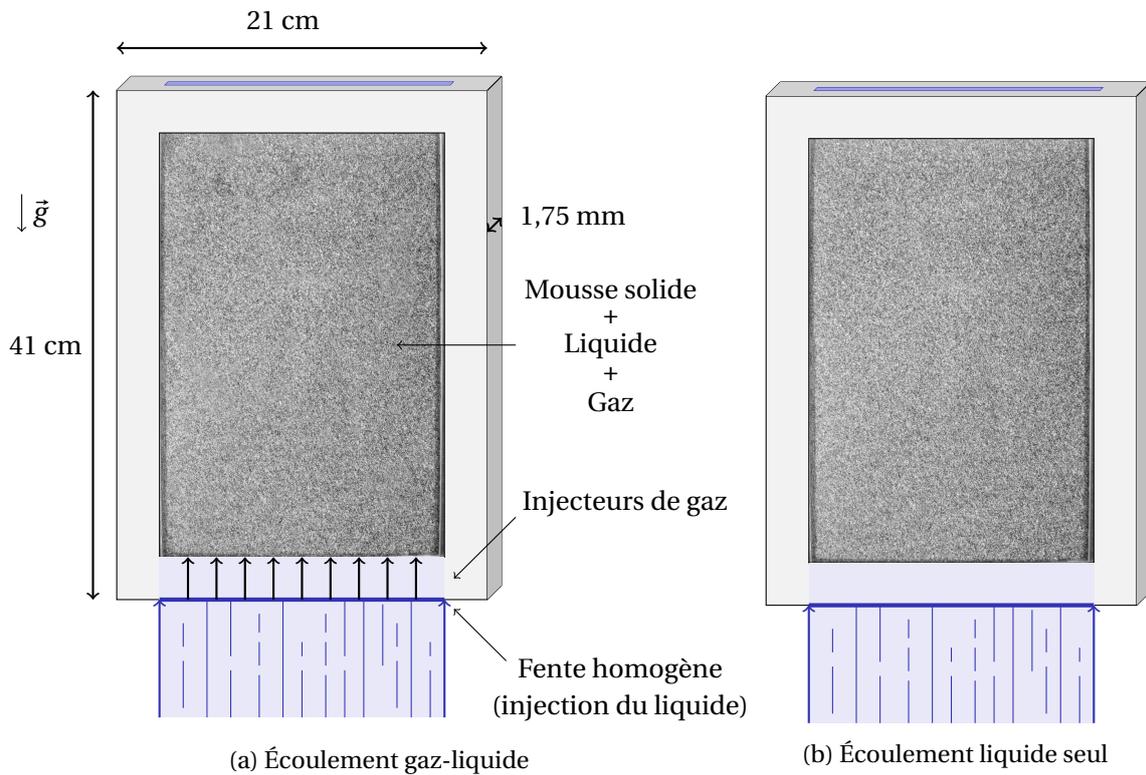


FIGURE 6.5 – Dispositif expérimental pour l'étude d'écoulements gaz-liquide en milieu poreux, \vec{g} désigne l'accélération de pesanteur et indique donc la verticale. (a) Écoulement simultané de gaz et de liquide au travers du milieu poreux. (b) Écoulement du liquide seul au travers du milieu poreux servant d'état de référence pour la normalisation des images avant leur traitement.

Les images de l'écoulement sont acquises par une caméra haute-résolution, Basler A2040-90um, de résolution 2048×2048 pixels, avec un objectif de 16 mm, à la fréquence de 100 Hz [19, 6]. Après sélection de la zone d'intérêt, les images à analyser sont de taille 1626×1160 pixels. La Figure 6.6a présente un exemple d'image d'écoulement enregistré à un instant t . À cause de la présence de la mousse, dont les pores ne sont pas résolus par la caméra, cette image est extrêmement bruitée. Afin d'augmenter le contraste entre bulles de gaz et écoulement liquide, chaque image est *normalisée* par l'« image de fond » de la Figure 6.6b, correspondant à un écoulement du liquide seul au travers du milieu poreux, *via* une division pixel par pixel $X = Y_t/Y_0$. C'est à l'image X obtenue, présentée à la Figure 6.6c, texturée, que sont appliquées les méthodes de segmentation de textures. Le zoom de la Figure 6.1b, extrait de l'image X de la Figure 6.6c, montre que la présence du gaz, au centre, se traduit par une texture irrégulière, à l'apparence claire, tandis que le liquide correspond à une texture plus lisse, d'apparence plus sombre. Cependant, la taille de pore typique de la mousse solide n'étant pas résolue par la caméra, la tâche de segmentation est difficile. Pour chaque expérience entre 50 et 3000 images sont enregistrées. De même que pour l'étude du frottement solide, on dispose de plusieurs jeux de données, correspondant à des configurations distinctes, caractérisées ici par les débits de gaz et de liquide et on souhaite caractériser les différents régimes hydrodynamiques.

6.3.3 Segmentation de texture pour les écoulements multiphasiques

6.3.3.1 Cadre d'étude

L'étude d'une première série de données a porté sur trois régimes d'écoulements, et a fait l'objet de plusieurs contributions, à la fois dans la communauté de traitement du signal et des

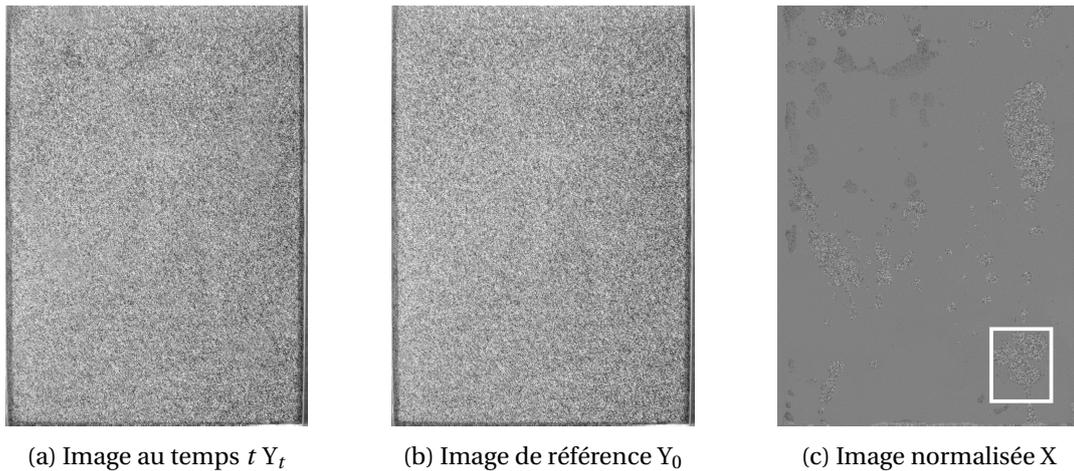


FIGURE 6.6 – Normalisation des images d'écoulement gaz-liquide au travers du milieu poreux par l'« image de fond », correspondant à l'écoulement du liquide seul au travers de la mousse solide. Toutes les méthodes de segmentation de textures sont appliquées à l'image normalisée X . Le rectangle blanc sur l'image X correspond à la zone couverte par le zoom de la Figure 6.1b.

images [15, 16], et en physique non linéaire [5]. Chaque écoulement est caractérisé par un débit de gaz Q_G et un débit de liquide Q_L . Pour chaque débit, un enregistrement vidéo est disponible, composant une série temporelle d'images texturées dont les caractéristiques varient peu d'une image à l'autre au sein d'une même série. Une image extraite de chaque série temporelle est présentée à la 1^{ère} colonne de la Figure 6.7, accompagnée de deux zooms en 2^{ème} colonne.

Nous allons illustrer la segmentation de trois images d'écoulement, représentatives de trois régimes observés lors des expériences :

- (i) un régime de faible activité, à bas débits, $Q_G = Q_L = 300$ mL/min, marqué par la présence de petites bulles,
- (ii) un régime de transition, à débits moyens, $Q_G = 400$ mL/min, $Q_L = 700$ mL/min, dans lequel on observe à la fois des petites bulles et de plus grosses bulles formées par coalescence,
- (iii) un régime de haute activité, à haut débit de gaz, $Q_G = 1200$ mL/min, $Q_L = 300$ mL/min, fortement dominé par les grosses bulles de gaz.

6.3.3.2 Méthodes de segmentation comparées

Quatre méthodes de segmentation sont comparées : la méthode de Yuan [22], présentée à la Section 2.3.3.1, la méthode $ROF_{\mathbf{h}}-S$, présentée à la Section 2.3.4 effectuant une régularisation par Variation Totale de l'estimée de la régularité locale $\hat{\mathbf{h}}^{RL}$ obtenue par régression linéaire, et les méthodes à contours « libres » et à contours « co-localisés », introduites au Chapitre 4.

L'analyse multi-échelle nécessaire à l'obtention des coefficients d'ondelettes dominants \mathcal{L} , est effectuée *via* une transformée en ondelettes bidimensionnelle, non décimée, calculée aux échelles $\{2^j, j = j_1, \dots, j_2\}$, avec une ondelette mère construite comme le produit tensoriel d'ondelettes de Daubechies asymétriques à trois moment nuls [13]. Cherchant à détecter des structures de (très) petites tailles, nous nous concentrons sur les échelles les plus fines, i.e., $j_1 = 1$ et $j_2 = 2$.

Dans cette section, de premiers résultats obtenus grâce à un réglage des hyperparamètres effectué manuellement sont présentés. Pour chaque méthode, le choix des paramètres optimaux, la taille de la fenêtre sur laquelle sont calculés les histogrammes pour Yuan, les paramètres de régularisation, λ pour $ROF_{\mathbf{h}}-S$, et (λ, α) pour les méthodes à contours « libres » et à contours « co-localisés », se fait en appliquant la méthode pour une gamme de paramètres et en sélectionnant

la segmentation qui apparaît la plus pertinente du point de vue de la physique du problème et des paramètres connus, tels que le volume de gaz injecté. Ainsi, le choix de paramètres repose sur la validation des expérimentateurs, et on parle alors de choix « expert ». Les résultats présentés en Figure 6.7 correspondent à la segmentation de « meilleure qualité » obtenue pour chaque méthode. La stratégie de réglage automatique des paramètres de régularisation introduite au Chapitre 5 sera mise en œuvre pour les méthodes ROF_h-S, à contours « libres » et à contours « co-localisés » à la section suivante.

6.3.3.3 Discussion des résultats

Deux types de bulles de gaz, d'apparences différentes, sont pointées sur les images d'écoulement de la 1^{ère} colonne de la Figure 6.7 comme des bulles « sombres » ou « claires » et semblent posséder des propriétés de textures différentes. En effet, si la méthode de Yuan [22], à la 3^{ème} colonne de la Figure 6.7, détecte parfaitement les bulles « claires », sans générer aucun artefact, et en fournissant des contours très réguliers, elle s'avère cependant totalement aveugle aux bulles sombres. Au contraire, les trois méthodes s'appuyant sur la régularité locale, ROF_h-S en 4^{ème} colonne, à contours « libres » en 5^{ème} colonne et à contours « co-localisés » en 6^{ème} colonne, détectent toutes les deux types de bulles. La méthode à contours « co-localisés » présente l'avantage de détecter les petites bulles présentes tout en évitant de générer des artefacts, nombreux pour la méthode ROF_h-S, et plus rares, mais présents, dans la segmentation à contours « libres ». En outre, on constate un gain significatif dans la régularité des contours grâce à la méthode de segmentation à contours « libres », par rapport à la méthode ROF_h-S. Ce gain est encore plus marqué pour la segmentation à contours « co-localisés ».

Ainsi, la méthode à contours « co-localisés » fournit une segmentation de qualité entre liquide et bulles de gaz, avec des frontières suffisamment régulières pour envisager une estimation robuste du périmètre d'interface gaz-liquide. Or, dans une expérience bi-dimensionnelle, ce périmètre, multiplié par l'épaisseur de la cellule, donne une estimation de la surface de contact entre le gaz et le liquide, paramètre physique crucial, par exemple dans les problèmes de catalyse. Si, en terme de qualité de segmentation, la méthode à contours « co-localisés » est privilégiée, c'est au prix d'un coût de calcul important. En effet, si la méthode de Yuan nécessite à peine 1 s de calcul et ROF_h-S environ 10 s, les méthodes à contours « libres » et à contours « co-localisés » nécessitent en moyenne respectivement 10 min et 30 min.

6.3.3.4 Modélisation d'écoulement à partir de textures synthétiques

Le fait que la méthode de Yuan soit aveugle aux bulles « sombres », tandis que toutes les bulles, « sombres » ou « claires » sont détectées par les méthodes s'appuyant sur la régularité locale, a conduit à formuler l'hypothèse que la différence de texture entre le gaz et le liquide correspondait à un changement de *régularité locale*, tandis que la différence entre bulles « sombres » et bulles « claires » résidait dans un changement de *variance locale*.

Afin d'appuyer cette hypothèse, nous proposons une brève modélisation d'un écoulement multiphasique grâce aux textures fractales homogènes par morceaux décrites au Chapitre 3. Pour cela, nous utilisons le masque de la Figure 6.8a, où chaque région correspond sur la Figure 6.8b à une texture fractale homogène caractérisée par sa régularité locale H et sa variance locale Σ^2 . Le fond, en noir, modélise le liquide et correspond à $H_1 = 0.4$ et $\Sigma_1^2 = 10^{-2}$, des bulles « sombres », en gris, sont modélisées par $H_2 = 0.9$ et $\Sigma_2^2 = \Sigma_1^2 = 10^{-2}$ (changement de régularité locale par rapport au fond sans changement de variance locale) et enfin, des bulles « claires », en blanc, correspondent à $H_3 = H_2 = 0.9$ et $\Sigma_3^2 = 10^{-1}$ (changement de régularité locale par rapport au fond et changement de variance locale par rapport aux « bulles sombres »). On cherche à détecter les « bulles de gaz », indépendamment de leur type, c'est-à-dire à retrouver le masque « gaz-liquide » de la Figure 6.8c, correspondant à un changement de *régularité locale*.

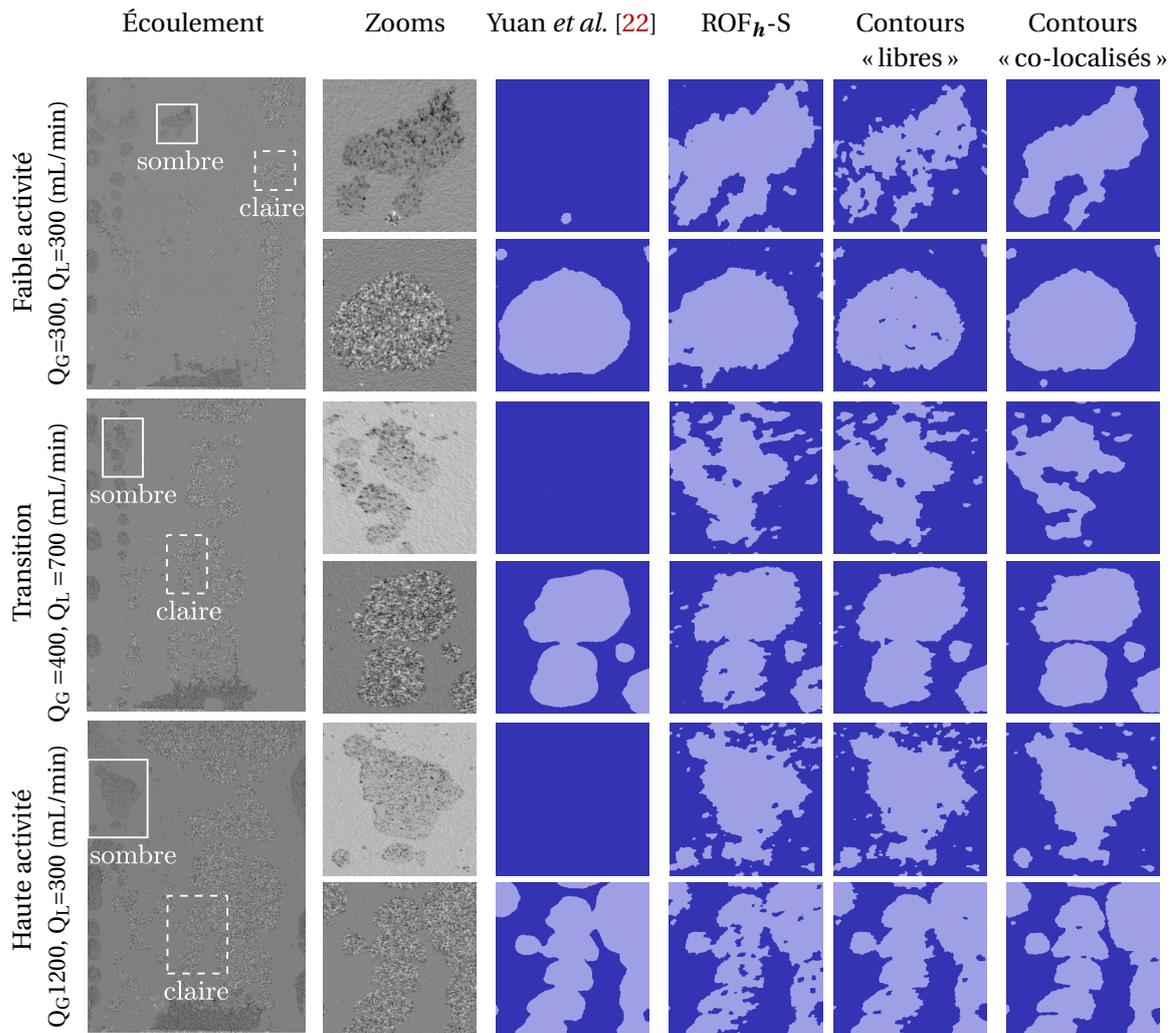


FIGURE 6.7 – Comparaison des différentes méthodes de segmentation pour plusieurs régimes d'écoulements. (Seuls des zooms sont affichés mais les algorithmes sont appliqués aux images entières, de 1677×1160 pixels, cf. 1^{ère} colonne.)

Les quatre méthodes, Yuan à $k = 2$ régions, $\text{ROF}_{\mathbf{h}\text{-S}}$, à contours « libres » et à contours « co-localisés » sont appliquées à la texture de la Figure 6.8b. Puisqu'on cherche à distinguer les deux phases par un changement de régularité locale, le seuillage itératif *a posteriori*, décrit à l'Algorithme 11, en Section 4.5.1, est appliqué à $\hat{\mathbf{h}}^{\text{ROF}}$, $\hat{\mathbf{h}}^{\text{L}}$ et $\hat{\mathbf{h}}^{\text{C}}$ pour obtenir des segmentations à $k = 2$ classes. Les paramètres optimaux, la taille de la fenêtre sur laquelle sont calculés les histogrammes pour Yuan, les paramètres de régularisation, λ pour $\text{ROF}_{\mathbf{h}\text{-S}}$, et (λ, α) pour les méthodes à contours « libres » et à contours « co-localisés », sont sélectionnés en parcourant une grille de paramètres sur laquelle est maximisé le pourcentage de pixels bien classés, au sens de la répartition « gaz-liquide » de la Figure 6.8c.

Les trois méthodes construites à partir d'attributs fractals, $\text{ROF}_{\mathbf{h}\text{-S}}$, à contours « libres » et à contours « co-localisés » détectent les deux types de bulles « claires » et « sombres », ce qui est cohérent avec les observations de la section précédente. Les contours des bulles sont bien plus réguliers pour la méthodes à contours « libres » et plus encore pour la méthode à contours « co-localisés », comme observé sur les données réelles. En outre, sur des données synthétiques il est possible de quantifier les performances de segmentation. On remarque alors que le score de classification est plus élevé pour les méthodes à contours « libres » et à contours « co-localisés », témoignant de leur plus grande pertinence. En comparaison, la méthode de Yuan [22] détecte très précisément les bulles claires, et leurs contours sont très réguliers, mais n'est pas du tout adaptée à la détection

des bulles sombres, ce qui correspond aux résultats présentés à la Figure 6.7.

Cette modélisation par des textures synthétiques permet de reproduire fidèlement les observations faites sur les données réelles de la Figure 6.7. Ces constatations tendent donc à confirmer l'hypothèse d'un changement de régularité locale entre la texture du gaz et celle du liquide, tandis que les bulles « sombres » ou « claires » correspondent à un changement de variance locale.

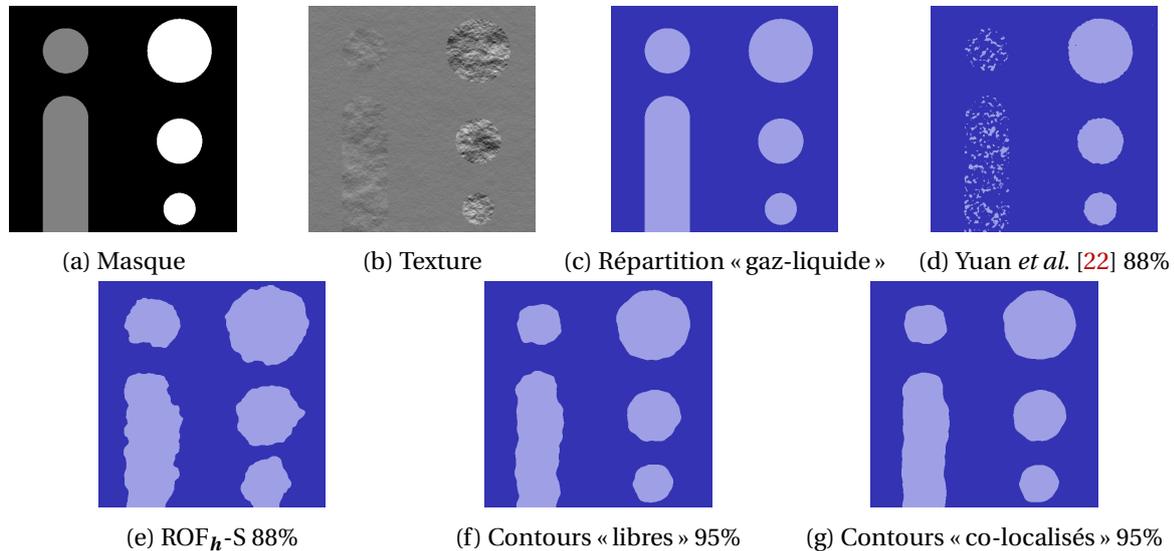


FIGURE 6.8 – Différentes méthodes de segmentation appliquées à une texture synthétique et scores de classification.

6.3.3.5 Mesure des quantités physiques

Les résultats présentés aux Sections 6.3.3.3 et 6.3.3.4 témoignent des bonnes performances de segmentation des méthodes à contours « libres » et à contours « co-localisés ». Ces deux méthodes ont donc été mises en œuvre, dans le cadre du stage de Loris Helmlinger, sur une série temporelle complète, correspondant à un écoulement multiphasique avec un débit de gaz $Q_G = 700$ mL/min et de liquide $Q_L = 300$ mL/min, et comparées avec la méthode de segmentation à l'aide d'outils de morphologie mathématique, initialement utilisée par les physiciens [20].

Deux quantités physiques intéressent particulièrement les expérimentateurs : d'une part, la fraction de gaz dans la cellule, qui peut être mesurée en calculant l'aire occupée par le gaz dans les images segmentées, et d'autre part, la surface de contact entre le gaz et le liquide, à laquelle on peut accéder en calculant le périmètre de l'interface multiplié par l'épaisseur de la cellule (car l'écoulement est quasi bidimensionnel). En effet, la fraction de gaz dans la cellule peut servir de paramètre de contrôle, tandis que dans le cadre d'une application en chimie des processus, on souhaite maximiser la surface de contact entre les différentes phases afin de rendre plus efficaces d'éventuelles réactions chimiques entre ces phases.

Pour chacune des méthodes, la morphologie mathématique [20], la segmentation à contours « libres » (4.5) ou à contours « co-localisés » (4.7), les paramètres (les seuils de binarisation en morphologie mathématique ou les paramètres de régularisation pour les méthodes par minimisation de fonctionnelle) un réglage « expert » fin est réalisé sur la première image, puis les paramètres sont conservés pour traiter l'ensemble de la série temporelle.

Les résultats de segmentation de la première image de la série sont présentés en Figure 6.9. On remarque que la méthode de segmentation s'appuyant sur la morphologie mathématique,

en Figure 6.9b, peine à détecter l'ensemble des bulles : certaines bulles au centre n'apparaissent que sous forme d'artefacts par exemple. Au contraire, les méthodes de segmentation à contours « libres », en Figure 6.9c, et à contours « co-localisés », en Figure 6.9d, détectent correctement la présence des bulles de gaz mais ont tendance à les fusionner, comme par exemple en haut de la cellule.

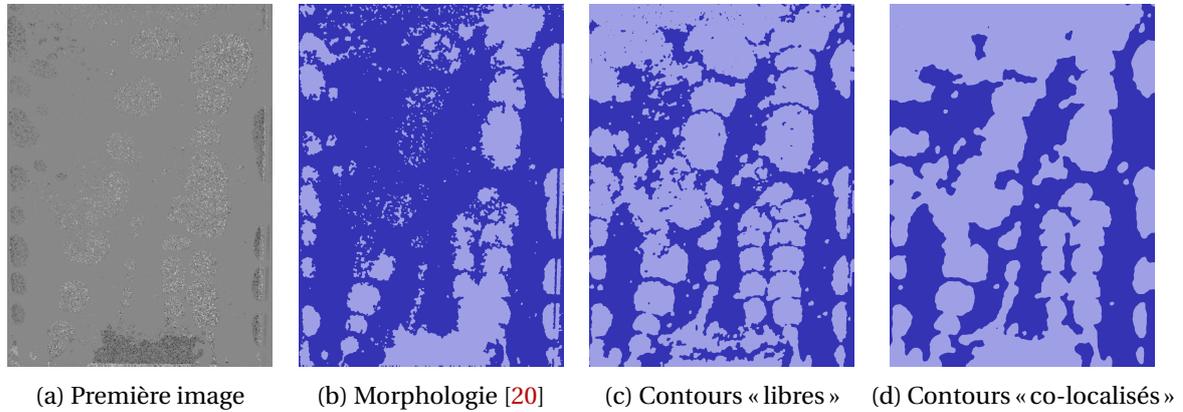


FIGURE 6.9 – Segmentation gaz-liquide sur une image d'écoulement multiphasique avec un débit de gaz $Q_G = 700$ mL/min et de liquide $Q_L = 300$ mL/min à l'aide de méthodes issues de la morphologie mathématique (b) *versus* à l'aide de la méthode à contours « libres » (c) ou à contours « co-localisés » (d).

À partir des segmentations des Figures 6.9b, 6.9c et 6.9d, la fonction MATLAB `regionprops` permet de mesurer l'aire occupée par le gaz et le périmètre de l'interface gaz-liquide. L'évolution au cours du temps de la fraction de gaz dans la cellule est représentée à la Figure 6.10a, et l'évolution du périmètre d'interface à la Figure 6.10b.

La Figure 6.10a montre une grande différence dans l'estimation de la fraction de gaz entre la méthode issue de la morphologie mathématique et les méthodes à contours « libres » et à contours « co-localisés ». Les segmentations présentées à la Figure 6.9 permettent d'avancer que l'aire estimée par les méthodes à contours « libres » (en bleu à la Figure 6.10a) et à contours « co-localisés » (en rouge à la Figure 6.10a) est très probablement surestimée, notamment à cause de la fusion de toutes les bulles de gaz en haut de cellule, observable sur les Figures 6.9c et 6.9d. Au contraire, l'aire estimée sur les segmentations effectuées par morphologie mathématique (en jaune sur la Figure 6.10b) semble légèrement sous-estimée, à cause notamment des bulles centrales très morcelées de la Figure 6.9b.

La détection d'un moins grand nombre de bulles par la méthode issue de la morphologie mathématique, entraîne une mesure de périmètre d'interface (en jaune à la Figure 6.10b) moindre. Les périmètres d'interface mesurés sur les segmentations obtenues par les méthodes à contours « libres » (en bleu à la Figure 6.10b) et à contours « co-localisés » (en rouge à la Figure 6.10b) partagent la même dynamique. La méthode à contours « co-localisés », favorisant des contours lisses, conduit logiquement à des mesures de périmètre plus faibles que la méthode à contours « libres ».

À partir de ces premières mesures quantitatives il est possible d'avancer que, si la segmentation obtenues par morphologie mathématique [20] fournit des mesures de fraction de gaz dans la cellule plus cohérentes avec les attendus des physiciens, les méthodes de segmentation proposées, à contours « libres » et à contours « co-localisés », semblent plus à même d'estimer correctement le périmètre de l'interface gaz-liquide, qui est une quantité physique d'intérêt, par exemple pour l'optimisation des processus de catalyse. Une étude plus systématique, en cours, permettra de comparer les mesures de la Figure 6.10 pour différentes séries temporelles, caractérisées par des débits de gaz et de liquide variés.

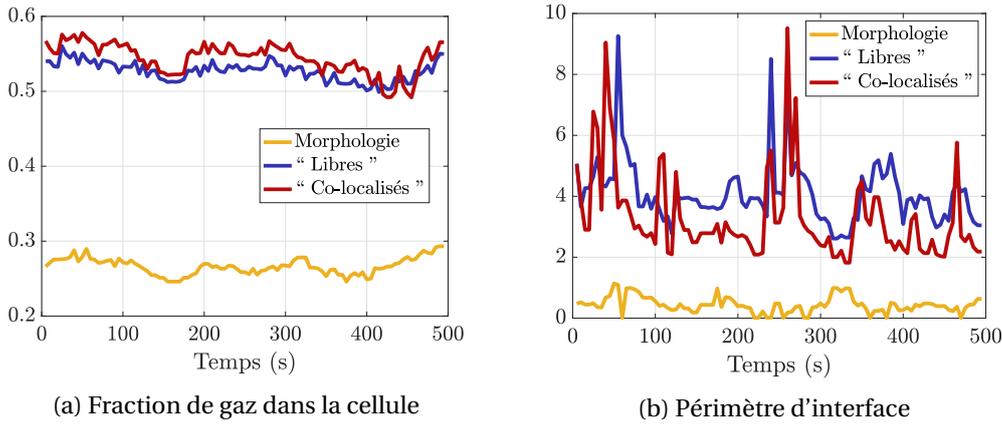


FIGURE 6.10 – Évolution temporelle de la fraction de gaz dans la cellule et du périmètre de l'interface gaz-liquide mesurées sur les segmentations obtenues *via* les outils de morphologie mathématique [20], la méthode à contours « libres » (4.5) et la méthode à contours « co-localisés » (4.7).

6.3.4 Réglage automatique des paramètres de régularisation

La Section 6.3.3 a illustré la pertinence des méthodes de segmentation s'appuyant sur des attributs fractals pour l'étude des écoulements multiphasiques en milieu poreux. Néanmoins, si un réglage « expert » des paramètres de régularisation peut être réalisé finement sur quelques images puis conservé pour la série temporelle complète avec une certaine robustesse, comme illustré à la Section 6.3.3.5, cette procédure ne permet pas de traiter des séries temporelles présentant une plus grande variabilité entre les images. De plus, le problème de la reproductibilité du réglage « expert » persiste. Afin de pallier ces problèmes, nous mettons en œuvre le formalisme de réglage automatique des paramètres du Chapitre 5, particularisé à la segmentation de texture à partir d'attributs fractals à la Section 5.4. Outre la plus grande objectivité de ce réglage automatique, il devient alors envisageable de traiter une série temporelle de plusieurs centaines d'images, chacune régularisée avec des paramètres adaptés.

6.3.4.1 Méthodes comparées

Nous allons considérer quatre méthodes de segmentation de texture à partir d'attributs fractals, avec réglage automatique des paramètres de régularisation. D'une part, le débruitage par Variation Totale de l'estimée $\hat{\mathbf{h}}^{\text{RL}}$, noté ROF $_{\mathbf{h}}$ -S, et correspondant à la résolution du Problème (2.123), la méthode de segmentation à contours « libres », correspondant au Problème (4.5), et la méthode à contours « co-localisés », correspondant au Problème (4.7) sont insérés dans le formalisme de sélection automatique des paramètres proposé au Chapitre 5, en prenant en compte la matrice de covariance des logarithmes des coefficients dominants *complète*, \mathcal{S} , décrite et calculée à la Section 5.5.3. D'autre part, dans la perspective de se doter d'une méthode « état-de-l'art » pour la segmentation de texture avec réglage automatique des paramètres, l'estimée de la régularité locale $\hat{\mathbf{h}}^{\text{RL}}$ est considérée comme une image bruitée X à laquelle est appliquée la régularisation par Variation Totale classique de l'Équation (2.46), avec seuillage *a posteriori* (voir Section 2.1.1.6), noté ROF $_X$ -S. Le réglage automatique du paramètre de régularisation λ est alors réalisé *via* une stratégie de Stein standard sous l'hypothèse d'un bruit gaussien i.i.d. sur $\hat{\mathbf{h}}^{\text{RL}}$ [8].

Afin de comparer ces quatre approches, nous proposons de les écrire sous la forme unifiée du modèle

$$\mathbf{y} = \Phi \bar{\mathbf{x}} + \boldsymbol{\zeta}, \quad (6.6)$$

introduit initialement à l'Équation (5.1), et résolu par les moindres carrés ordinaires pénalisés

pour une norme $\ell_{2,1}$

$$\hat{\mathbf{x}}(\mathbf{y}; \Lambda) = \operatorname{argmin}_{\mathbf{x} \in \mathcal{H}} \frac{1}{2} \|\mathbf{y} - \Phi \mathbf{x}\|_{\mathcal{W}}^2 + \|\mathbf{U}_\Lambda \mathbf{x}\|_{2,1} \quad (6.7)$$

formulés à l'Équation (5.3). Le Tableau 6.2 présente les choix à effectuer dans le Modèle (6.6) et dans le Problème (6.7) pour retrouver chacune des méthodes. Pour les méthodes ROF $_{\mathbf{h}}$ -S, à contours « libres » et à contours « co-localisés », le Lemme 5.1, formulé au Chapitre 5, présente et justifie la formulation sous forme unifiée, qui est reprise dans le Tableau 6.2 ci-dessous. Nous insistons sur le fait que ces quatre approches ne diffèrent pas seulement par la fonctionnelle minimisée, mais également par la modélisation du bruit, i.e., la matrice de covariance utilisée pour calculer les estimées de l'erreur quadratique, SURE, et de son gradient par rapport aux paramètres de régularisation, SUGAR. Ainsi, les méthodes ROF $_{\mathbf{x}}$ -S et ROF $_{\mathbf{h}}$ -S correspondent à la minimisation de la *même* fonctionnelle, mais le réglage automatique des paramètres, qui dépend du modèle de bruit considéré, est différent.

Méthode (Problème)	Figures 6.11, 6.12	Observée \mathbf{y}	Opérateur Φ	Quantité \mathbf{x}	Bruit ζ	Fonctionnelle (6.7) \mathcal{W}	$\ \mathbf{U}_\Lambda \mathbf{x}\ $
ROF $_{\mathbf{x}}$ -S (2.46)	(e), (f) (g), (h)	$\hat{\mathbf{h}}^{\text{RL}}$	\mathbf{I}	\mathbf{h}	$\mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$	\mathbf{I}	$\lambda \text{TV}(\mathbf{h})$
ROF $_{\mathbf{h}}$ -S (2.123)	(i), (j) (k), (l)	$\log_2(\mathcal{L})$	Φ (4.3)	\mathbf{h}	$\mathcal{N}(\mathbf{0}, \mathcal{S})$	$\mathbf{A}^* \mathbf{A}$ (5.9)	$\lambda \text{TV}(\mathbf{h})$
Contours « libres » (4.5)	(m), (n) (o), (p)	$\log_2(\mathcal{L})$	Φ (4.3)	(\mathbf{h}, \mathbf{v})	$\mathcal{N}(\mathbf{0}, \mathcal{S})$	\mathbf{I}	$G_{(\lambda, \alpha)}^{\text{L}}(\mathbf{h}, \mathbf{v})$
Contours « co- localisés » (4.7)	(q), (r) (s), (t)	$\log_2(\mathcal{L})$	Φ (4.3)	(\mathbf{h}, \mathbf{v})	$\mathcal{N}(\mathbf{0}, \mathcal{S})$	\mathbf{I}	$G_{(\lambda, \alpha)}^{\text{C}}(\mathbf{h}, \mathbf{v})$

TABLEAU 6.2 – **Méthodes automatiques de segmentation de texture grâce à des attributs fractals.** L'estimateur de régularité locale par régression linéaire est noté $\hat{\mathbf{h}}^{\text{RL}}$, \mathcal{L} désigne la concaténation des coefficients d'ondelettes dominants de l'image d'écoulement, TV désigne la Variation Totale, les pénalisation $G_{(\lambda, \alpha)}^{\text{L}}$ (à contours « libres ») et $G_{(\lambda, \alpha)}^{\text{C}}$ (à contours « co-localisés ») sont définies au Chapitre 4, et \mathcal{S} , la matrice de covariance des logarithmes des coefficients dominants est décrite et estimée à la Section 5.5.3.

L'analyse multi-échelles nécessaire à l'obtention des coefficients d'ondelettes dominants \mathcal{L} , est effectuée, comme à la section précédente, sur les octaves $j_1 = 1$ et $j_2 = 2$.

6.3.4.2 Covariance et paramètres des estimateurs SURE et SUGAR

Pour le calcul des estimateurs SURE et SUGAR, il est nécessaire de disposer (d'une estimée) de la covariance du bruit altérant l'observation \mathbf{y} , ainsi que de fixer le pas de différences finies \mathbf{v} . Pour cela, la méthode « état-de-l'art » ROF $_{\mathbf{x}}$ -S doit être distinguée des méthodes proposées s'appuyant sur les propriétés de la régularité locale (et de la puissance locale).

Dans le cas de la méthode ROF $_{\mathbf{x}}$ -S, $\hat{\mathbf{h}}^{\text{RL}}$, obtenue par régression linéaire sur les logarithmes des coefficients dominants, est vue comme une image, de taille $N_1 \times N_2 = N$ pixels, correspondant à la « vraie » régularité locale $\bar{\mathbf{h}}$, inconnue, corrompue par un bruit gaussien i.i.d. $\zeta \in \mathbb{R}^N \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$. La variance du bruit est alors estimée *via* un estimateur de variance classique, appliqué à $\hat{\mathbf{h}}^{\text{RL}}$ et

l'application de la Formule (5.82) conduit au choix suivant pour le pas de différences finies

$$v = \frac{2\sqrt{\sigma^2}}{N^{0,3}}. \quad (6.8)$$

Le vecteur de Monte Carlo $\epsilon \in \mathbb{R}^N$ est tiré aléatoirement, de manière i.i.d. selon une loi gaussienne normalisée.

Pour les méthodes ROF_h-S, à contours « libres » et à contours « co-localisés », la matrice de covariance $\widehat{\mathcal{P}}$ est estimée sur les logarithmes des coefficients dominants pour *chaque* image à étudier, en appliquant la procédure décrite à l'Équation (5.86). Le pas de différences finies est alors fixé en appliquant la formule (5.82)

$$v = \frac{2\sqrt{\max \widehat{\mathcal{P}}}}{P^{0,3}} \quad (6.9)$$

où $P = N_j N_1 N_2$, avec le nombre d'échelles $N_j = 2$. Le vecteur de Monte Carlo $\epsilon = (\epsilon_j)_{j=j_1}^{j_2} \in \mathbb{R}^P$ est tiré aléatoirement, de manière i.i.d. selon une loi gaussienne normalisée.

6.3.4.3 Recherche sur grille *versus* réglage automatique des paramètres

Pour la comparaison du réglage des paramètres de régularisation sur grille ou de manière automatique, deux images d'écoulement de 1626×1160 pixels, présentées aux Figures 6.11a et 6.12a, sont considérées, appelées respectivement image n° 1 et image n° 2. Ces images sont extraites d'une série temporelle d'images d'une expérience menée avec un débit de gaz $Q_G = 100$ mL/min et un débit de liquide $Q_L = 100$ mL/min, elles sont successives dans la vidéo de l'écoulement et espacées d'une durée de 5 s. De même que dans le cas des signaux de « coller-glisser », on ne connaît pas la « véritable » carte de régularité locale sous-jacente et on ne peut donc pas accéder à l'erreur quadratique exacte. Il est donc nécessaire de se tourner vers un estimateur SURE de l'erreur quadratique, sur lequel il est possible de réaliser, une recherche de minimum sur une grille, ou une minimisation automatique grâce à un estimateur SUGAR du gradient de l'erreur quadratique.

La recherche des paramètres optimaux sur une grille étant très coûteuse en temps de calcul, elle est effectuée, pour chaque image, sur un zoom de 281×231 pixels, illustré à la Figure 6.11c pour l'image n° 1 (resp. 6.12c pour l'image n° 2). Pour les méthodes de débruitage par Variation Totale, ROF_X-S, et ROF_h-S, quinze valeurs du paramètre λ , logarithmiquement réparties, sont explorées. Quant aux méthodes à contours « libres » et à contours « co-localisés », elles sont appliquées pour une grille de 15×15 couples de paramètres (λ, α) , logarithmiquement espacés également. Puis, l'Algorithme 15 de sélection automatique des paramètres est mis en œuvre, une première fois pour le zoom extrait de l'image, ce qui permettra de comparer le résultat à la recherche sur grille, et une seconde fois sur l'image entière, afin d'obtenir une segmentation globale gaz-liquide.

Pour chaque image, n° 1 en Figure 6.13 et n° 2 en Figure 6.14, le paramètre λ (ou le couple de paramètres (λ, α)) obtenu comme le minimum sur la grille explorée, $\widehat{\lambda}^{\text{grille}}$ (ou $(\lambda^{\text{grille}}, \alpha^{\text{grille}})$), est indiqué par le symbole « + », tandis que le résultats de la sélection automatique, $\widehat{\lambda}^{\text{BFGS}}$ (ou $(\widehat{\lambda}^{\text{BFGS}}, \widehat{\alpha}^{\text{BFGS}})$) est indiqué par le symbole « * ».

Pour les méthodes ROF_X-S et ROF_h-S appliquées aux images n° 1 et n° 2, les Figures 6.13a, 6.13b et 6.14a, 6.14b, montrent que les paramètres $\widehat{\lambda}^{\text{grille}}$ et $\widehat{\lambda}^{\text{BFGS}}$ coïncident presque exactement, excepté une légère sous-estimation par l'Algorithme 15 du paramètre optimal pour la méthode ROF_h appliqué au zoom de l'image n° 2 (Figure 6.14b). Quant aux paramètres optimaux sur une grille ou sélectionnés automatiquement pour les méthodes à contours « libres » et à contours « co-localisés »

appliquées aux images n° 1 et n° 2, s'ils ne coïncident parfaitement que dans un seul cas (Figure 6.14c), ils restent cohérents, au sens où ils se trouvent tous dans une région très plate de l'estimateur SURE contenant le véritable minimum de l'erreur quadratique estimée (Figures 6.13c, 6.13d et 6.14d).

Si, pour les méthodes ROF_{χ} -S et ROF_h -S, à un seul paramètre, le nombre d'appels au schéma primal-dual minimisant la fonction objectif est similaire entre la recherche sur grille et le réglage automatique par l'Algorithme 15, pour les méthodes à contours « libres » et à contours « co-localisés », à deux paramètres de régularisation, la minimisation sur la grille considérée nécessite 225 appels au schéma primal-dual, tandis que l'Algorithme 15 n'en nécessite en moyenne que 50. Ainsi, dès que plusieurs paramètres doivent être réglés, on tire un bénéfice certain à mettre en œuvre une stratégie de sélection automatique plutôt qu'une coûteuse recherche sur grille.

6.3.4.4 Résultats des méthodes de segmentation automatiques

Les estimées et les segmentations présentées aux Figures 6.11 et 6.12 correspondent aux résultats obtenus avec un réglage automatique des paramètres de régularisation par l'Algorithme 15, pour l'image entière de 1626×1160 pixels (pour laquelle la recherche sur grille est hors de portée en raison du temps de calcul nécessaire) dans la partie gauche (1^{ère} et 2^{ème} colonnes), et pour le zoom de 281×231 pixels, indiqué par le rectangle blanc, dans la partie droite (3^{ème} et 4^{ème} colonnes).

Les méthodes ROF_{χ} -S et ROF_h -S, dont les résultats de segmentation et d'estimation sont présentés respectivement en 2^{ème} et 3^{ème} lignes des Figure 6.13 et 6.14, présentent des contours très irréguliers et de nombreux artefacts, comme nous avons pu le constater également à la Section 6.3.3. Une longueur d'interface calculée à partir de ces résultats serait fortement surestimée, tandis qu'une estimation de l'aire totale de gaz serait au contraire sous-estimée. Les méthodes à contours « libres » et à contours « co-localisés », dont les résultats de segmentation et d'estimation sont présentés respectivement en 4^{ème} et 5^{ème} lignes des Figures 6.13 et 6.14 confirment leur capacité à obtenir des contours plus réguliers et à éliminer les artefacts tout en détectant des tailles de bulles variées, observées à la Section 6.3.3.

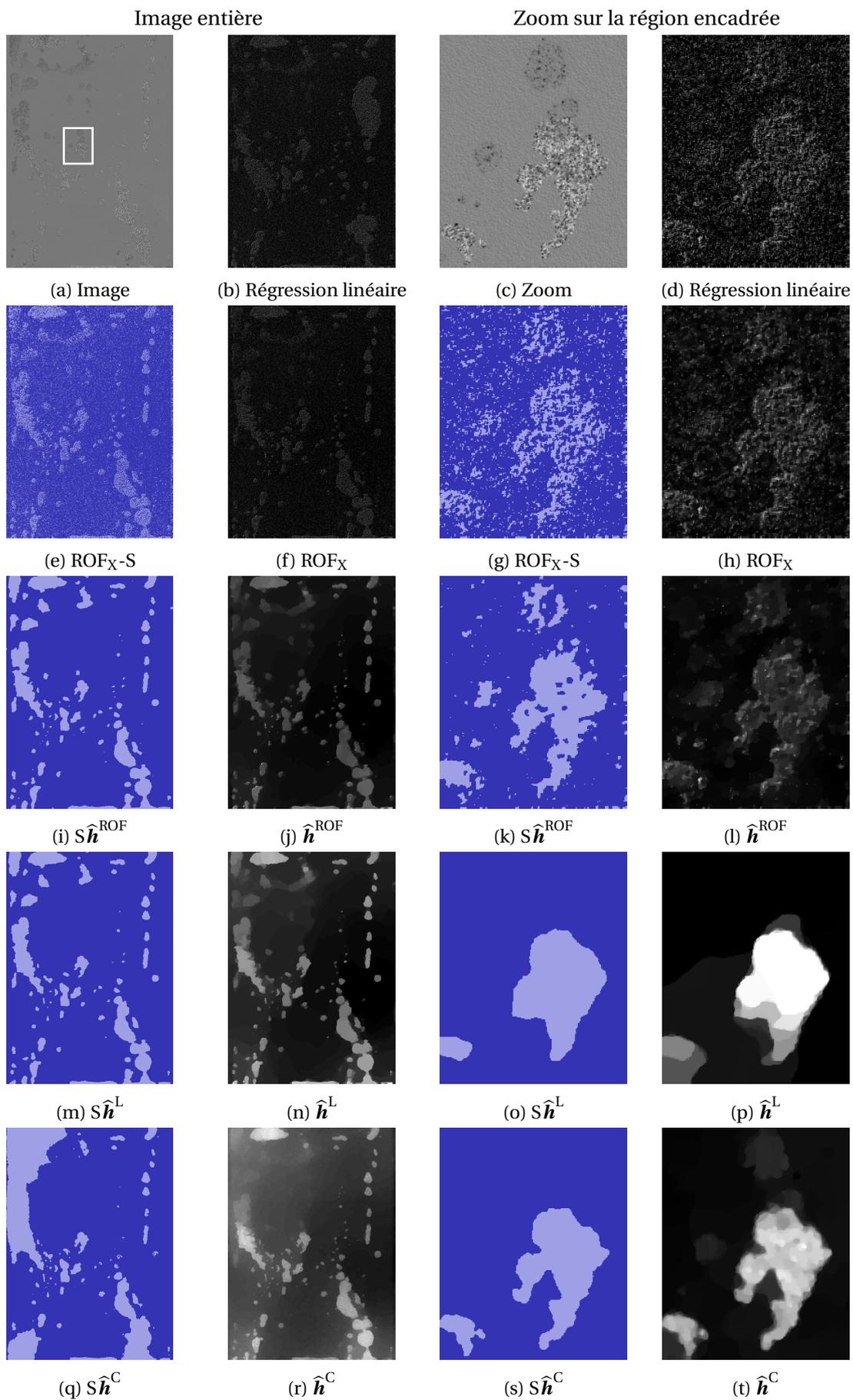


FIGURE 6.11 – **Segmentation de texture pour l'étude découlement multiphasique au travers d'un milieu poreux.** Comparaison des méthodes de segmentation avec sélection automatique des paramètres de régularisation résumés au Tableau 6.2 sur l'image n° 1.

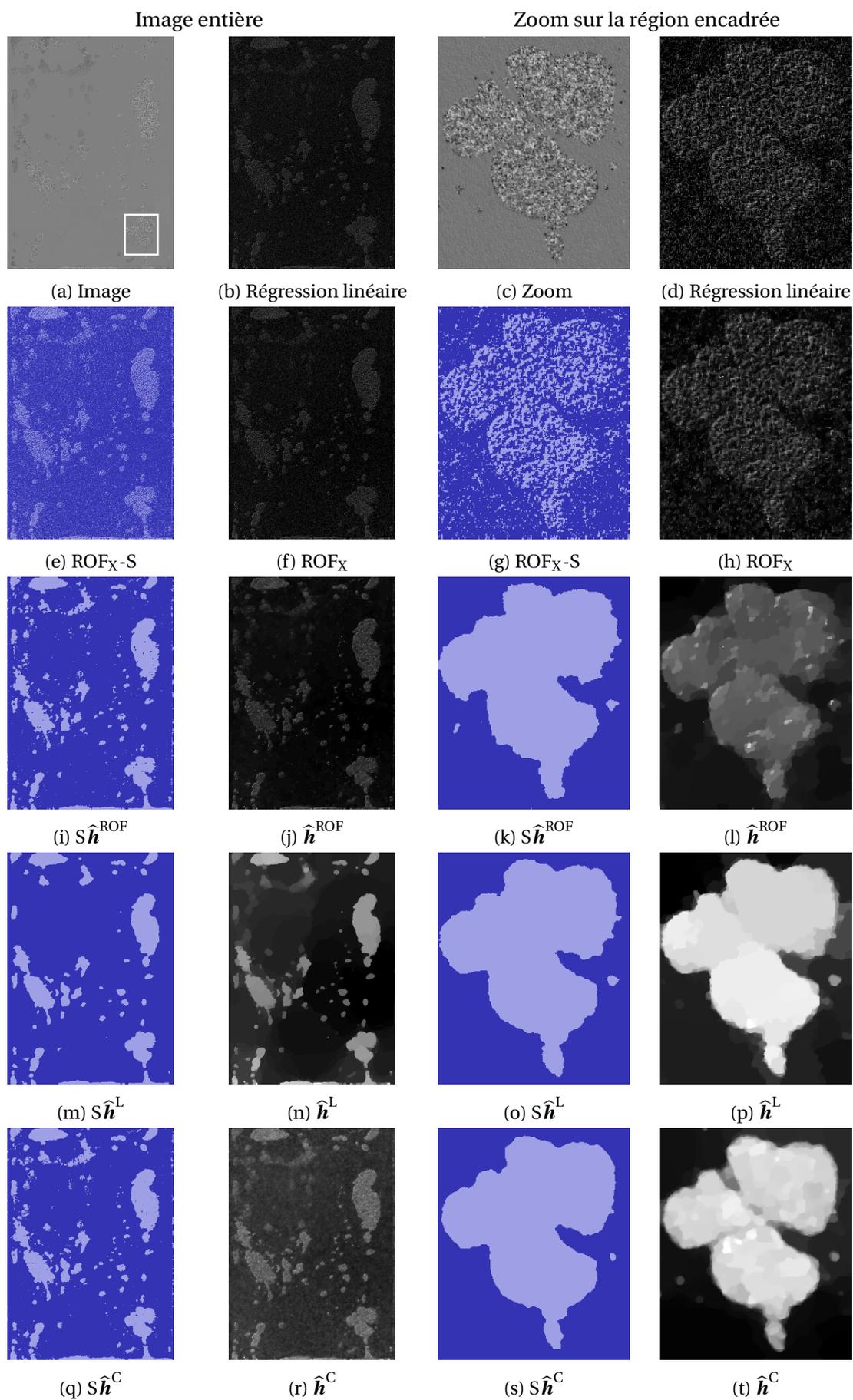


FIGURE 6.12 – Segmentation de texture pour l'étude d'écoulement multiphasique au travers d'un milieu poreux. Comparaison des méthodes de segmentation avec sélection automatique des paramètres de régularisation résumés au Tableau 6.2 sur l'image n° 2.

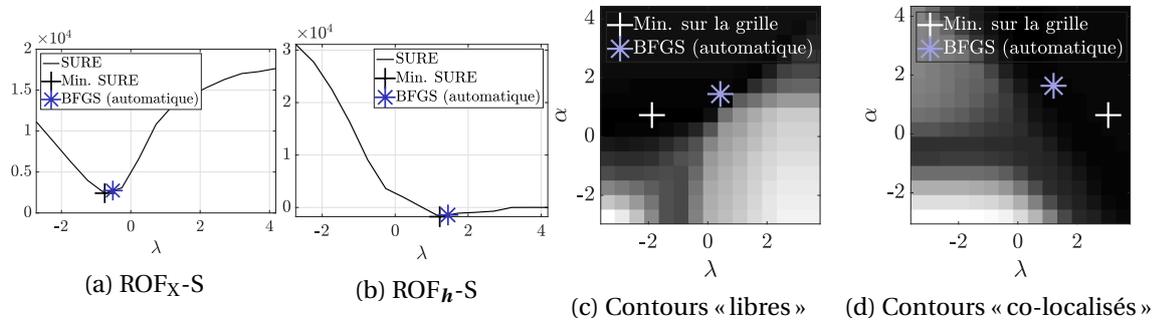


FIGURE 6.13 – Recherche sur grille *versus* réglage automatique des paramètres de régularisation pour la segmentation d'image d'écoulement multiphasique n° 1.

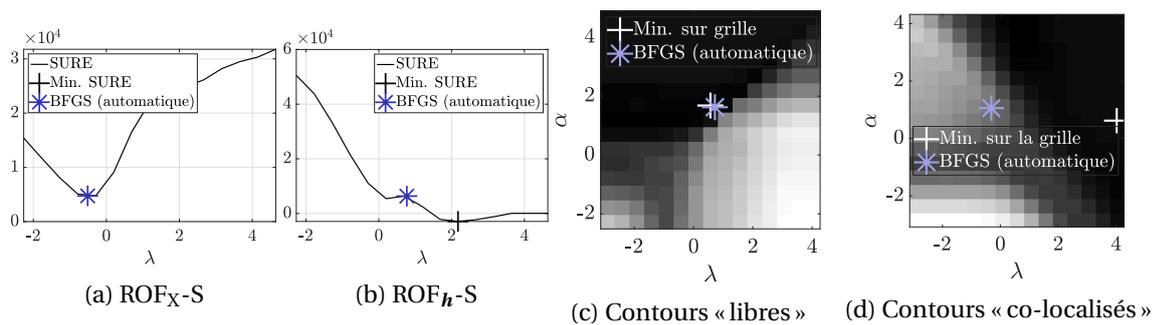


FIGURE 6.14 – Recherche sur grille *versus* réglage automatique des paramètres de régularisation pour la segmentation d'image d'écoulement multiphasique n° 2.

6.4 Bilan

Au travers de deux exemples de physique non-linéaire, le frottement solide et les écoulements multiphasiques en milieu poreux, nous avons illustré l'utilité des méthodes de débruitage imposant un comportement homogène par morceaux pour le traitement de données de physique expérimentale. En outre, la formulation grâce à une fonctionnelle régularisante est particulièrement adaptable à des problèmes *a priori* extrêmement éloignés en terme de phénomènes physiques mis en jeu, et portant sur des données de type complètement différents (signaux et images).

De plus, en particulierisant le formalisme développé au Chapitre 5, ces méthodes de débruitage *non supervisées* ne nécessitent aucun réglage de paramètres par l'utilisateur, leur conférant plusieurs avantages significatifs, dont les principaux sont la maniabilité, la reproductibilité et la capacité à gérer de gros volumes de données.

Références

- [1] T. Auger, J. Mathe, V. Viasnoff, G. Charron, J.-M. Di Meglio, L. Auvray, and F. Montel. Zero-mode waveguide detection of flow-driven dna translocation through nanopores. *Phys. Rev. Lett.*, 113 :028302, 2014. [162](#)
- [2] A. Babchin, I. Brailovsky, P. Gordon, and G. Sivashinsky. Fingering instability in immiscible displacement. *Phys. Rev. E*, 77 :026301, 2008. [169](#)
- [3] T. Baumberger and C. Caroli. Solid friction from stick-slip down to pinning and aging. *Adv. Phys.*, 55(3-4) :279–348, 2006. [163](#), [164](#)
- [4] M. Berhanu, R. Monchaux, S. Fauve, N. Mordant, F. Petrelis, A. Chiffaudel, F. Daviaud, B. Dubrulle, L. Marie, F. Ravelet, M. Bourgoïn, P. Odier, J.-F. Pinton, and R. Volk. Magnetic field reversals in an experimental turbulent dynamo. *Eur. Phys. Lett.*, 77 :59001, 2007. [162](#)
- [5] T. Busser, B. Pascal, N. Pustelnik, P. Abry, M. Serres, R. Philippe, and V. Vidal. Ecoulement gaz-liquide dans un milieu poreux confiné : caractérisation par analyse d'images. 2019. [162](#), [171](#)
- [6] T. Busser, M. Serres, R. Philippe, and V. Vidal. Hydrodynamics of gas-liquid co-current flow through a thin sheet of highly porous open cell solid foam. *in revision at Chem. Eng. Sci.*, 2020. [169](#), [170](#)
- [7] J. Colas, N. Pustelnik, C. Oliver, P. Abry, J.-C. Gémard, and V. Vidal. Nonlinear denoising for characterization of solid friction under low confinement pressure. *Phys. Rev. E*, 100 :032803, 2019. [163](#), [164](#), [165](#)
- [8] C.-A. Deledalle, S. Vaïter, J. Fadili, and G. Peyré. Stein unbiased gradient estimator of the risk (SUGAR) for multiple parameter selection. *SIAM J. Imaging Sci.*, 7(4) :2448–2487, 2014. [176](#)
- [9] T. Divoux, H. Gayvallet, and J.-C. Gémard. Creep motion of a granular pile induced by thermal cycling. *Phys. Rev. Lett.*, 101 :148303, 2008. [162](#)
- [10] V. Hessel, P. Angeli, A. Gavrilidis, and H. Löwe. Gas-liquid and gas-liquid-solid microstructured reactors : Contacting principles and applications. *Industrial & engineering chemistry research*, 44 :9750–9769, 2005. [169](#)
- [11] Q. Kang, I. N. Tsimpanogiannis, D. Zhang, and P. C. Lichtner. Numerical modeling of pore-scale phenomena during CO₂ sequestration in oceanic sediments. *Fuel Process. Technol.*, 86 :1647–1665, 2005. [169](#)

- [12] M. T. Kreutzer, F. Kapteijn, J. A. Moulijn, and J. J. Heiszwolf. Multiphase monolith reactors : Chemical reaction engineering of segmented flow in microchannels. *Chemical Engineering Science*, 60 :5895–5916, 2005. 169
- [13] S. Mallat. *A Wavelet Tour of Signal Processing, Third Edition : The Sparse Way*. Academic Press, Inc., Orlando, FL, USA, 3rd edition, 2008. ISBN 0123743702, 9780123743701. 171
- [14] C. Marone. Laboratory-derived friction laws and their application to seismic faulting. *Ann. Rev. Earth Planet. Sci.*, 26 :643–696, 1998. 163
- [15] B. Pascal, N. Pustelnik, P. Abry, M. Serres, and V. Vidal. Joint estimation of local variance and local regularity for texture segmentation. Application to multiphase flow characterization. In *Proc. Int. Conf. Image Process.*, pages 2092–2096, Athens, Greece, 2018. IEEE. 162, 169, 171
- [16] B. Pascal, T. Busser, N. Pustelnik, P. Abry, and V. Vidal. Segmentation d’images texturées en grande dimension. application à l’analyse d’écoulements multiphasiques. In *GRETSI 2019 XXVIIème Colloque francophone de traitement du signal et des images*, 2019. 162, 171
- [17] B. Pascal, N. Pustelnik, P. Abry, J.-C. Géminard, and V. Vidal. Parameter-free and fast non-linear piecewise filtering. Application to experimental physics. *Preprint arxiv :2006.03297 [physics.data-an]*, 2020. 162
- [18] K. R. Reddy and J. A. Adams. Effects of soil heterogeneity on airflow patterns and hydrocarbon removal during in situ air sparging. *J. Geotech. Geoenviron. Eng.*, 127(3) :234–247, 2001. 169
- [19] M. Serres. Etude hydrodynamique d’un écoulement gaz-liquide dans un milieu poreux confiné. *PhDThesis, École Normale Supérieure de Lyon, Université de Lyon*, page 203, 2017. 170
- [20] M. Serres, M.-L. Zanota, R. Philippe, and V. Vidal. On the stability of Taylor bubbles inside a confined highly porous medium. *Int. J. Multiph. Flow*, 85 :157–163, 2016. 169, 174, 175, 176
- [21] M. Serres, T. Maison, R. Philippe, and V. Vidal. A phenomenological model for bubble coalescence in confined highly porous media. *Int. J. Multiph. Flow*, 105 :134–141, 2018. 169
- [22] J. Yuan, D. Wang, and A. M. Cheriyyadat. Factorization-based texture segmentation. *IEEE Trans. Image Process.*, 24(11) :3488–3497, Nov. 2015. ISSN 1057-7149, 1941-0042. doi : 10.1109/TIP.2015.2446948. URL <http://ieeexplore.ieee.org/document/7127013/>. 171, 172, 173, 174

Chapitre 7

Conclusion

« *La bêtise consiste à vouloir conclure.* »

G. Flaubert

Bilan

Nous nous sommes intéressés dans cette thèse à la segmentation de textures à partir d'attributs fractals. Après avoir positionné le problème dans le champ du traitement d'images au **Chapitre 2**, nous avons présenté une première méthode de segmentation de textures s'appuyant sur la *régularité locale*, mesurée *via* l'exposant de Hölder local. Cette méthode procède en deux étapes disjointes : tout d'abord, une estimation par régression linéaire sur les logarithmes des coefficients dominants de la régularité locale en chaque pixel, puis, un débruitage par Variation Totale. Outre l'implémentation d'une version accélérée de cette méthode à l'Algorithme 5, nous lui avons également adjoint une procédure de post-traitement afin d'obtenir une segmentation par un seuillage itératif, ce qui constitue une méthode que nous avons appelée $\text{ROF}_h\text{-S}$ et considérée comme le point de départ de ce travail.

Constatant les limites de cette procédure, où erreur d'estimation et biais de régularisation se cumulent, nous avons construit au **Chapitre 4** deux variantes d'une vraisemblance pénalisée, permettant de réaliser simultanément l'estimation et la régularisation des attributs de texture. La segmentation repose alors sur la minimisation des deux fonctionnelles associées, dites à contours « libres » et à contours « co-localisés ». La démonstration de la forte-convexité des deux fonctionnelles et le calcul exact de leur module de forte-convexité ont permis la mise en œuvre de schémas d'optimisation rapides. Afin de valider les performances des procédures de segmentation construites, nous avons introduit un champ gaussien stationnaire asymptotiquement auto-similaire à partir duquel nous avons élaboré une stratégie de synthèse de textures fractales homogènes par morceaux, d'attributs et de segmentation prescrits au **Chapitre 3**. Grâce à ces textures synthétiques, des simulations numériques réalisées pour des problèmes de segmentation de textures de difficulté croissante ont montré que les méthodes à contours « libres » et à contours « co-localisés » proposées obtenaient des performances d'estimation et de segmentation supérieures à celles de la méthode $\text{ROF}_h\text{-S}$, notamment lorsque les textures à segmenter sont caractérisées par des régularités et des variances locales similaires. Les algorithmes rapides pour la segmentation de texture par les méthodes $\text{ROF}_h\text{-S}$, à contours « libres » et à contours « co-localisés » sont implémentés en MATLAB et disponibles en ligne dans la toolbox [GSUGAR20]¹.

Les textures synthétiques du **Chapitre 3** ont également été utilisées pour construire un formalisme de segmentation de textures fractales à partir de réseaux de neurones convolutionnels,

1. <https://github.com/bpascal-fr/gsugar>

présenté au **Chapitre 4**. Trois réseaux de complexité croissante ont été construits et entraînés. Nous avons pu constater leur capacité à réaliser des segmentations précises, y compris pour le réseau le moins profond. Néanmoins, nous avons noté que ces méthodes supervisées étaient peu robustes à une légère inadéquation entre données de test et d'entraînement, ce qui limite leur application à des données réelles. En outre, nous avons quantifié le fait que les contours obtenus *via* les méthodes de segmentation à partir de réseaux de neurones sont fortement irréguliers, et par conséquent que les mesures de périmètre d'interface entre deux régions sont alors peu fiables. Au contraire, les méthodes de segmentation à contours « libres » et à contours « co-localisés » présentent des frontières suffisamment lisses pour pouvoir effectuer des mesures de périmètre pertinentes.

Le **Chapitre 5** a abordé la question générale de la sélection des hyperparamètres dans les méthodes s'appuyant sur une vraisemblance pénalisée, telles que, par exemple, les méthodes à contours « libres » et à contours « co-localisés ». Nous avons développé un formalisme de réglage des hyperparamètres grâce à des estimateurs de l'erreur quadratique et du gradient de l'erreur quadratique par rapport aux hyperparamètres, dont nous avons montré le caractère asymptotiquement non biaisé sous l'hypothèse que les données observées subissent une déformation linéaire et sont corrompues par un bruit additif gaussien, de covariance quelconque. Des simulations numériques, menées pour le réglage des paramètres de régularisation de la méthode de segmentation à contours « libres », ont montré, d'une part, que l'estimateur de l'erreur quadratique proposé reproduisait très précisément le comportement de l'erreur, et d'autre part, que la minimisation automatique s'appuyant sur l'estimateur du gradient de l'erreur fournissaient des paramètres de régularisation optimaux au sens de l'erreur quadratique. En outre, nous montrons que l'estimation de la covariance du bruit directement à partir des données n'altère pas l'efficacité de ce formalisme. Des algorithmes opérationnels pour le réglage automatique et piloté par les données des paramètres de régularisation des méthodes de segmentation de texture ROF_h, à contours « libres » et à contours « co-localisés » décrites au Chapitre 4 sont réunis dans une boîte à outils MATLAB [GSUGAR20], disponible en ligne².

Le formalisme décrit au **Chapitre 5**, permettant le réglage automatique et piloté par les données des hyperparamètres, a permis, au **Chapitre 6** de traiter des données issues d'expériences de physique non linéaire de manière plus efficace et plus objective que le réglage « expert » requis usuellement dans ce type d'applications. De plus, grâce à la mise à disposition de deux boîtes à outils [STEIN20] et [GSUGAR20], ces méthodes peuvent être prise en main par des physiciens sans connaissance préalable sur les techniques de traitement du signal et de l'image sous-jacentes.

Perspectives

Algorithmes proximaux par blocs. Si les algorithmes rapides développés ont permis de traiter efficacement des images réelles de haute résolution, issues de l'étude des écoulements multiphasiques, le temps de calcul nécessaire au traitement d'une série temporelle entière reste conséquent. Le nombre de couples de débits gaz-liquide que l'on peut explorer est donc limité. L'établissement d'un diagramme des phases précis nécessiterait d'accélérer plus efficacement encore les schémas numériques proximaux réalisant la segmentation. Une alternative aux stratégies d'accélération FISTA et primal-dual s'appuyant sur la forte-convexité, sont les stratégies d'algorithmes *par blocs*, dont nous avons étudié un exemple pour la résolution du problème ROF_h par algorithme *forward-backward* dans [5, ICASSP18]. Nous avons notamment illustré sur cet exemple que l'accélération obtenue en considérant des algorithmes *par blocs* était fortement influencée par la manière de définir ces blocs. Les résultats de convergence démontrés par [8] suggèrent également que le choix des probabilités de mise à jour des blocs à chaque itération influence la vitesse

2. <https://github.com/bpascal-fr/g sugar>

de convergence. Nous étudions actuellement la possibilité de décrire le choix optimal des blocs et des probabilités de mise à jour en fonction du problème considéré, ce qui pourrait permettre de proposer des algorithmes stochastiques pour la minimisation des fonctionnelles à contours « libres » et à contours « co-localisés » et d'obtenir une convergence plus rapide des schémas numériques.

Segmentation de textures multifractales. Dans cette thèse nous avons considéré des textures caractérisées par une régularité locale uniforme, mesurée *via* l'exposant de Hölder, et par leur variance locale, dites textures *monofractales*. Or, il existe un formalisme plus général, qui correspond au modèle de textures *multifractales*, dans lequel chaque texture est caractérisée par une *distribution* d'exposants de Hölder, appelé le spectre multifractal [1], ou spectre des singularités, noté $\mathcal{D}(h)$. Ce spectre est défini, pour un champ $f : Y \subset \mathbb{R}^2 \rightarrow \mathbb{R}$, comme la dimension de Hausdorff, notée $\dim_{\mathcal{H}}$, de l'ensemble des points où le champ f admet une régularité locale h , i.e.,

$$\mathcal{D}(h) = \dim_{\mathcal{H}} \{ \underline{x} \in Y, h(\underline{x}) = h \}. \quad (7.1)$$

Pour la caractérisation des textures multifractales, [9, Section 2.6.2] et [2, Section 5.4] ont proposé de définir des attributs *multifractals* en considérant une expression paramétrique du spectre $\mathcal{D}(h)$:

$$\mathcal{D}(h) = 2 + \frac{c_2}{2} \left(\frac{h - c_1}{c_2} \right)^2 - \frac{c_3}{6} \left(\frac{h - c_1}{c_2} \right)^3 + \dots, \quad (7.2)$$

c_1 correspondant au maximum de $\mathcal{D}(h)$, c_2 à sa largeur et c_3 à son coefficient d'asymétrie. Les paramètres c_1, c_2, c_3, \dots forment alors un ensemble d'attributs de texture. Dans la limite d'un spectre de largeur nulle, on retrouve le modèle de texture *monofractale*. Ainsi, décrire une texture *multifractale* nécessite *a minima* de considérer les *deux* premiers coefficients c_1, c_2 apparaissant dans le développement de l'Équation (7.2).

Le lien entre la régularité locale et le comportement au travers des échelles des coefficients d'ondelettes dominants, exposé à la Proposition 2.2, permet de lier la statistique des coefficients dominants aux paramètres apparaissant dans le développement du spectre multifractal [1] $\mathcal{D}(h)$ de l'Équation (7.2). En effet, il a été montré par [9, Section 2.6.2] et [2, Section 5.4] que les log-cumulants des coefficients dominants, notés $C_p(j)$ à l'ordre p et à l'échelle j , présentent le comportement linéaire au travers des échelles suivant :

$$\forall p \geq 1, \quad C_p(j) = c_{0,p} + c_p j \quad (7.3)$$

où les coefficients $\{c_p, p \geq 1\}$ caractérisent le spectre multifractal et peuvent donc être estimés par régression linéaire au travers des échelles sur les log-cumulants $C_p(j)$.

Le formalisme multifractal le plus simple consiste à considérer uniquement les deux premiers termes du développement (7.2), et décrit un spectre caractérisé par les paramètres c_1, c_2 . L'estimation de ces attributs multifractals nécessite de considérer les deux premiers log-cumulants, qui fournissent les relations suivantes :

$$C_1(j) = \mathbb{E} \log_2 \mathcal{L}_j(\cdot) = c_{0,1} + c_1 j, \quad C_2(j) = \mathbb{V} \log_2 \mathcal{L}_j(\cdot) = c_{0,2} + c_2 j, \quad (7.4)$$

où \mathbb{E} désigne l'espérance et \mathbb{V} la variance. En pratique, les log-cumulants $C_1(j)$ et $C_2(j)$ sont calculés par moyennage spatial sur les pixels, ce qui permettent d'accéder aux attributs c_1, c_2 *globaux* de la texture en réalisant une régression linéaire sur $C_1(j)$ et $C_2(j)$.

Pour effectuer une segmentation, il est nécessaire d'obtenir des estimations *locales* des attributs. Pour le coefficients c_1 , en considérant que

$$\log_2 \mathcal{L}_j(\underline{n}) = c_{0,1}(\underline{n}) + c_1(\underline{n}) j \quad (7.5)$$

il est possible de définir un coefficient $c_1(\underline{n})$ ponctuel en effectuant la même régression linéaire au travers des échelles que celle que nous avons utilisé pour estimer la régularité locale pixel par pixel à la Proposition 2.3. Pour l'estimation du coefficient c_2 , qui implique un moment d'ordre deux, il n'existe aucune procédure *strictement* réduite à un pixel. Il faut alors concéder un moyennage des logarithmes des coefficients dominants sur un voisinage de pixels, noté \mathcal{V} , de sorte que $c_2(\underline{n})$ est calculé à partir de la relation :

$$\text{Var}_{\underline{n}' \in \mathcal{V}(\underline{n})} \log_2 \mathcal{L}_j(\underline{n}') = c_{0,2}(\underline{n}) + c_2(\underline{n})j, \quad (7.6)$$

où $\text{Var}_{\underline{n}' \in \mathcal{V}(\underline{n})}$ désigne la variance sur les pixels \underline{n}' appartenant au voisinage \mathcal{V} centré en \underline{n} . Si le voisinage \mathcal{V} est trop grand, la localisation nécessaire à la segmentation est perdue, s'il est trop faible, le paramètre c_2 souffre d'une variance d'estimation trop grande et ne discrimine plus les différentes textures.

Pour un voisinage \mathcal{V} fixé, une régression linéaire sur les Équations (7.5) et (7.6) fournit des estimées bruitées, $\hat{\mathbf{c}}_1^{\text{RL}}$ et $\hat{\mathbf{c}}_2^{\text{RL}}$, qui peuvent ensuite être régularisées. En s'appuyant par exemple une pénalisation de type Variation Totale, on obtient un problème de minimisation convexe :

$$(\hat{\mathbf{c}}_1, \hat{\mathbf{c}}_2) = \underset{\mathbf{c}_1, \mathbf{c}_2 \in \mathbb{R}^{\mathbb{N}_1 \times \mathbb{N}_2}}{\text{argmin}} \frac{1}{2} \|\mathbf{c}_1 - \hat{\mathbf{c}}_1^{\text{RL}}\|^2 + \frac{\mu}{2} \|\mathbf{c}_2 - \hat{\mathbf{c}}_2^{\text{RL}}\|^2 + \lambda \text{TV}_\alpha(\mathbf{c}_1, \mathbf{c}_2). \quad (7.7)$$

Cette méthode présente néanmoins plusieurs inconvénients. D'une part, elle implique non seulement deux paramètres de régularisation λ, α , mais également un paramètre μ qui permet de gérer une éventuelle différence d'ordre de grandeur entre les coefficients c_1 et c_2 . D'autre part, tout comme la méthode ROF_h-S, cette procédure accumule l'erreur d'estimation et le biais de régularisation, ce qui est d'autant plus dommageable, que l'estimation du paramètre c_2 entraîne un bruit d'estimation particulièrement fort. Enfin, en cherchant à estimer des attributs de texture définis en chaque pixel, $c_1(\underline{n})$ et $c_2(\underline{n})$, le caractère intrinsèquement *non ponctuel* du spectre multifractal est perdu.

Pour pallier les deux premières limitations, il serait nécessaire de considérer directement les logarithmes des coefficients d'ondelettes dominants, comme nous l'avons fait pour l'estimation et la régularisation simultanées des attributs *monofractals* au Chapitre 4, pour construire une vraisemblance pénalisée adaptée à la segmentation de textures *multifractales*. La difficulté centrale réside dans le fait que l'information est contenue dans le comportement *statistique* des logarithmes des coefficients d'ondelettes. Pour cela, nous pourrions envisager d'utiliser des outils mesurant des distances entre *distributions*, tels que la distance de Wasserstein [6, Proposition 2.2], dont la définition pour des ensembles discrets est rappelée à la Définition 2.12, qui seraient alors mis à profit pour comparer des distributions de logarithmes de coefficients dominants en différentes localisations de l'image. En effet, ces outils ont récemment été utilisés avec succès pour des problèmes de segmentation d'images, éventuellement texturées, [3, 7] (voir Section 2.3.1.1).

Les textures multifractales présentent des statistiques complexes, et la plupart des méthodes classiques de comparaison de distributions échouent à distinguer les changements subtils dans la distribution des intensités lumineuses de deux textures multifractales distinctes. Nous postulons donc que l'application de la distance de Wasserstein, non pas dans le domaine de l'image directement, mais à une représentation multi-échelle adaptée aux textures multifractales, permettrait de capter le plus précisément possible les différences entre deux textures multifractales et ainsi de mieux les segmenter le cas échéant. En outre, par construction, une distance de Wasserstein permettrait une meilleure prise en compte du caractère non ponctuel du spectre $\mathcal{D}(h)$.

Tout comme nous avons pu nous inspirer des méthodes de débruitage par Variation Totale usuelles en traitement d'images pour la minimisation des fonctionnelles à contours « libres » et à contours « co-localisés », des avancées récentes dans la formulation et la résolution numérique de

problèmes s'appuyant sur une distance de Wasserstein pénalisée [4, 7], et plus généralement sur les problèmes variationnels faisant intervenir la distance de Wasserstein [6, Chapitre 9], ouvrent la voie vers une mise en *oeuvre* de ces stratégies pour la segmentation de textures multifractales.

Références

- [1] S. Jaffard. Wavelet techniques in multifractal analysis. *Fractal Geometry and Applications : A Jubilee of Benoit Mandelbrot*, M. Lapidus et M. van Frankenhuysen Eds, *Proceedings of Symposia in Pure Mathematics (AMS)*, 72(2) :91–152, 2004. [187](#)
- [2] S. Jaffard, P. Abry, and H. Wendt. *Irregularities and Scaling in Signal and Image Processing : Multifractal Analysis*, pages 31–116. World scientific publishing, Singapore, 2015. ISBN 9789814366069. [187](#)
- [3] M. Jung, G. Peyré, and L. D. Cohen. Nonlocal active contours. *SIAM J. Imaging Sci.*, 5(3) : 1022–1054, 2012. [188](#)
- [4] J. Lellmann, D. A. Lorenz, C. Schonlieb, and T. Valkonen. Imaging with Kantorovich–Rubinstein Discrepancy. *SIAM J. Imaging Sci.*, 7(4) :2833–2859, 2014. [189](#)
- [5] B. Pascal, N. Pustelnik, P. Abry, and J.-C. Pesquet. Block-coordinate proximal algorithms for scale-free texture segmentation. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1253–1257, Calgary, Alberta, Canada, Apr. 15-20 2018. IEEE. [186](#)
- [6] G. Peyré and M. Cuturi. *Computational Optimal Transport*. arXiv :1803.00567, 2018. [188](#), [189](#)
- [7] J. Rabin and N. Papadakis. Convex color image segmentation with optimal transport distances. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 256–269. Springer, 2015. [188](#), [189](#)
- [8] S. Salzo and S. Villa. Parallel Random Block-Coordinate Forward-Backward Algorithm : A Unified Convergence Analysis. *arXiv preprint arXiv :1906.07392*, 2019. [186](#)
- [9] H. Wendt. *Contributions of Wavelet Leaders and Bootstrap to Multifractal Analysis : Images, Estimation Performance, Dependence Structure and Vanishing Moments. Confidence Intervals and Hypothesis Tests*. PhD thesis, Ecole Normale Supérieure de Lyon, 2008. [187](#)

Annexe A

Propriétés du champ brownien fractionnaire

Pour $0 < H < 1$, et Σ réel positif, le champ bidimensionnel b_H défini par la représentation harmonique

$$b_H(\underline{x}) \stackrel{(d)}{=} \Sigma \int_{\mathbb{R}^2} \frac{e^{-i\langle \underline{x}, \underline{v} \rangle} - 1}{C_H^{1/2} \|\underline{v}\|^{H+1}} \tilde{w}(d\underline{v}), \quad (\text{A.1})$$

où $\underline{v} = (v_1, v_2)$ désigne la fréquence spatiale, C_H est une constante de normalisation et $\tilde{w}(d\underline{v})$ est une mesure de Wiener (aléatoire) sur \mathbb{R}^2 [1], est appelé *champ brownien Fractionnaire*.

Propriétés. *Le champ b_H , défini à l'Équation (A.1) possède les propriétés suivantes :*

- (i) *Il est H-auto-similaire.*
- (ii) *Il est à accroissements stationnaires (Définition 3.2).*
- (iii) *Il est isotrope, c'est-à-dire invariant, au sens des distributions, sous les rotations du plan \mathbb{R}^2 .*
- (iv) *Sa structure de covariance est donnée par*

$$\mathbb{E}[b_H(\underline{x})b_H(\underline{x}')] = \frac{\Sigma^2}{2} (\|\underline{x}\|^{2H} + \|\underline{x}'\|^{2H} - \|\underline{x} - \underline{x}'\|^{2H}), \quad (\text{A.2})$$

où nous appellerons *variance typique* la quantité Σ^2 , qui correspond à la variance sur le cercle unité \mathcal{S} .

Démonstration. (i) Soit $a > 0$,

$$b_H(a\underline{x}) \stackrel{(d)}{=} \Sigma \int_{\mathbb{R}^2} \frac{e^{-i\langle a\underline{x}, \underline{v} \rangle} - 1}{C_H^{1/2} \|\underline{v}\|^{H+1}} \tilde{w}(d\underline{v}), \quad (\text{A.3})$$

$$\stackrel{(d)}{=} \Sigma \int_{\mathbb{R}^2} \frac{e^{-i\langle \underline{x}, a\underline{v} \rangle} - 1}{C_H^{1/2} \|\underline{v}\|^{H+1}} \tilde{w}(d\underline{v}), \quad (\text{A.4})$$

$$\underline{v}' \triangleq a\underline{v} \stackrel{(d)}{=} \Sigma \int_{\mathbb{R}^2} \frac{e^{-i\langle \underline{x}, \underline{v}' \rangle} - 1}{C_H^{1/2} \|\underline{v}'\|^{H+1} a^{-(H+1)}} \tilde{w}\left(\frac{d\underline{v}'}{a}\right), \quad (\text{A.5})$$

$$\stackrel{(d)}{=} a^H \Sigma \int_{\mathbb{R}^2} \frac{e^{-i\langle \underline{x}, \underline{v}' \rangle} - 1}{C_H^{1/2} \|\underline{v}'\|^{H+1}} \tilde{w}(d\underline{v}'), \quad (\text{A.6})$$

$$\stackrel{(d)}{=} a^H b_H(\underline{x}). \quad (\text{A.7})$$

(ii) En appliquant la Formule (3.3) et la Définition 3.2 :

$$\begin{aligned}
 b_H(\underline{x} + \underline{\delta}) - b_H(\underline{\delta}) &\stackrel{(d)}{=} \Sigma \int_{\mathbb{R}^2} \frac{e^{-i\langle \underline{x} + \underline{\delta}, \underline{v} \rangle} - 1}{C_H^{1/2} \|\underline{v}\|^{H+1}} \tilde{w}(\underline{d}\underline{v}) - \Sigma \int_{\mathbb{R}^2} \frac{e^{-i\langle \underline{\delta}, \underline{v} \rangle} - 1}{C_H^{1/2} \|\underline{v}\|^{H+1}} \tilde{w}(\underline{d}\underline{v}), \\
 &\stackrel{(d)}{=} \Sigma \int_{\mathbb{R}^2} \frac{e^{-i\langle \underline{x}, \underline{v} \rangle} - 1}{C_H^{1/2} \|\underline{v}\|^{H+1}} e^{-i\langle \underline{\delta}, \underline{v} \rangle} \tilde{w}(\underline{d}\underline{v}), \\
 &\stackrel{(d)}{=} \Sigma \int_{\mathbb{R}^2} \frac{e^{-i\langle \underline{x}, \underline{v} \rangle} - 1}{C_H^{1/2} \|\underline{v}\|^{H+1}} \tilde{w}(\underline{d}\underline{v}), \\
 &\stackrel{(d)}{=} b_H(\underline{x}) - b_H(\underline{0}), \tag{A.8}
 \end{aligned}$$

en utilisant l'invariance de la mesure de Wiener $\tilde{w}(\underline{d}\underline{v})$ sous la multiplication par un nombre complexe de module 1.

- (iii) La mesure brownienne $\tilde{w}(\underline{d}\underline{v})$ étant elle-même isotrope, on déduit de la représentation harmonique (3.3) l'isotropie du champ brownien fractionnaire.
- (iv) Commençons par noter que le champ b_H est réel, $b_H(\underline{x}) \stackrel{(d)}{=} \overline{b_H(\underline{x})}$, et calculons la variance du champ au point \underline{x} . On supposera, sans perte de généralité puisque le champ est isotrope, que $\underline{x} = \|\underline{x}\| \underline{e}_1$, où $\underline{e}_1 = (1, 0)$. Ainsi, développons le calcul de la variance du champ b_H :

$$\begin{aligned}
 \mathbb{E} [b_H(\underline{x}) b_H(\underline{x})] &= \mathbb{E} [b_H(\underline{x}) \overline{b_H(\underline{x})}], \\
 &= \mathbb{E} \left[\Sigma^2 \int_{\mathbb{R}^2} \int_{\mathbb{R}^2} \left(\frac{e^{-i\langle \underline{x}, \underline{v} \rangle} - 1}{C_H^{1/2} \|\underline{v}\|^{H+1}} \right) \left(\frac{e^{i\langle \underline{x}, \underline{v}' \rangle} - 1}{C_H^{1/2} \|\underline{v}'\|^{H+1}} \right) \tilde{w}(\underline{d}\underline{v}) \tilde{w}(\underline{d}\underline{v}') \right], \\
 &= \Sigma^2 \int_{\mathbb{R}^2} \int_{\mathbb{R}^2} \left(\frac{e^{-i\langle \underline{x}, \underline{v} \rangle} - 1}{C_H^{1/2} \|\underline{v}\|^{H+1}} \right) \left(\frac{e^{i\langle \underline{x}, \underline{v}' \rangle} - 1}{C_H^{1/2} \|\underline{v}'\|^{H+1}} \right) \mathbb{E} [\tilde{w}(\underline{d}\underline{v}) \tilde{w}(\underline{d}\underline{v}')], \\
 &= \Sigma^2 \int_{\mathbb{R}^2} \left(\frac{e^{-i\langle \underline{x}, \underline{v} \rangle} - 1}{C_H^{1/2} \|\underline{v}\|^{H+1}} \right) \left(\frac{e^{i\langle \underline{x}, \underline{v} \rangle} - 1}{C_H^{1/2} \|\underline{v}\|^{H+1}} \right) \frac{\underline{d}\underline{v}}{2\pi}, \\
 &= \frac{\Sigma^2}{C_H} \int_{\mathbb{R}^2} \frac{2 - 2 \cos(\langle \underline{x}, \underline{v} \rangle)}{\|\underline{v}\|^{2H+2}} \frac{\underline{d}\underline{v}}{2\pi}, \\
 \underline{v}' \Delta = \|\underline{x}\| \underline{v} &= \frac{\Sigma^2}{C_H} \int_{\mathbb{R}^2} \frac{2 - 2 \cos(\underline{v}'_1)}{\|\underline{v}'\|^{2H+2}} \frac{\|\underline{x}\|^{2H+2}}{\|\underline{x}\|^2} \frac{\underline{d}\underline{v}'}{2\pi}, \\
 &= \Sigma^2 \|\underline{x}\|^{2H}. \tag{A.9}
 \end{aligned}$$

Puis notons que

$$\mathbb{E} [b_H(\underline{x}) b_H(\underline{x}')] = \frac{1}{2} \left(\mathbb{E} [b_H(\underline{x})^2] + \mathbb{E} [b_H(\underline{x}')^2] - \mathbb{E} [(b_H(\underline{x}) - b_H(\underline{x}'))^2] \right) \tag{A.10}$$

$$= \frac{1}{2} \left(\Sigma^2 \|\underline{x}\|^{2H} + \Sigma^2 \|\underline{x}'\|^{2H} - \mathbb{E} [(b_H(\underline{x}) - b_H(\underline{x}'))^2] \right). \tag{A.11}$$

Il reste alors à calculer le dernier terme

$$\begin{aligned}
 \mathbb{E} [(b_H(\underline{x}) - b_H(\underline{x}'))^2] &= \mathbb{E} \left[\Sigma^2 \int_{\mathbb{R}^2} \int_{\mathbb{R}^2} \left(\frac{e^{-i\langle \underline{x}, \underline{v} \rangle} - e^{-i\langle \underline{x}', \underline{v} \rangle}}{C_H^{1/2} \|\underline{v}\|^{H+1}} \right) \left(\frac{e^{i\langle \underline{x}, \underline{v}' \rangle} - e^{i\langle \underline{x}', \underline{v}' \rangle}}{C_H^{1/2} \|\underline{v}'\|^{H+1}} \right) \tilde{w}(\underline{d}\underline{v}) \tilde{w}(\underline{d}\underline{v}') \right], \\
 &= \frac{\Sigma^2}{C_H} \int_{\mathbb{R}^2} \frac{(e^{-i\langle \underline{x}, \underline{v} \rangle} - e^{-i\langle \underline{x}', \underline{v} \rangle})(e^{i\langle \underline{x}, \underline{v} \rangle} - e^{i\langle \underline{x}', \underline{v} \rangle})}{\|\underline{v}\|^{2H+2}} \frac{\underline{d}\underline{v}}{2\pi}, \\
 &= \frac{\Sigma^2}{C_H} \int_{\mathbb{R}^2} \frac{(2 - 2 \cos(\langle \underline{x} - \underline{x}', \underline{v} \rangle))}{\|\underline{v}\|^{2H+2}} \frac{\underline{d}\underline{v}}{2\pi}, \\
 &= \Sigma^2 \|\underline{x} - \underline{x}'\|^{2H}, \tag{A.12}
 \end{aligned}$$

en se ramenant aux calculs développés en (A.9).

□

Références

- [1] S. Cohen and J. Istaş. *Fractional fields and applications*. Springer, 2013. I

Annexe B

Synthèse de textures fractales par convolution

Cette description est le fruit d'échanges avec Laurent Chevillard (chercheur au Laboratoire de Physique de l'ENS Lyon) et Jason Reneuve (ancien doctorant sous la direction de L. Chevillard).

Une alternative à la représentation harmonique des champs Gaussiens (comme (3.3) pour le champ Brownien et (3.9) pour le drap Brownien) est une représentation dans l'espace direct. Ce type de représentation est l'analogie continu de la formule (3.56), où un filtrage linéaire « mélange » un bruit blanc Gaussien de manière à reproduire la structure de covariance. L'avantage d'une représentation continue est sa plus grande souplesse. Il n'est plus nécessaire pour imposer la structure de covariance souhaitée d'effectuer de coûteuse factorisation matricielle. Néanmoins, une discrétisation est ensuite requise *a posteriori* pour obtenir un champ discret, ce qui conduit à la synthèse approximée d'un champ Gaussien. Détaillons cette construction, discutée dans [1].

Soit $\mathcal{D} \in L^2(\mathbb{R}^2)$ un noyau tel que pour tout $\underline{x} \in \mathbb{R}^2$, la fonction $\underline{y} \mapsto \mathcal{D}(\underline{x}, \underline{y})$ soit de carré intégrable. On définit le champ Gaussien

$$f(\underline{x}) = \int_{\mathbb{R}^2} \mathcal{D}(\underline{x}, \underline{y}) w(d\underline{y}), \quad (\text{B.1})$$

où $w(d\underline{y})$ désigne une mesure de bruit blanc Gaussien. La structure de covariance du champ f s'écrit

$$\begin{aligned} \mathbb{E}[f(\underline{x})f(\underline{x}')] &= \int_{\mathbb{R}^2} \int_{\mathbb{R}^2} \mathcal{D}(\underline{x}, \underline{y}) \mathcal{D}(\underline{x}', \underline{y}') \mathbb{E}[w(d\underline{y})w(d\underline{y}')] \\ &= \int_{\mathbb{R}^2} \mathcal{D}(\underline{x}, \underline{y}) \mathcal{D}(\underline{x}', \underline{y}) \delta(\underline{y} - \underline{y}') d\underline{y} d\underline{y}' \\ &= \int_{\mathbb{R}^2} \mathcal{D}(\underline{x}, \underline{y}) \mathcal{D}(\underline{x}', \underline{y}) d\underline{y} \end{aligned} \quad (\text{B.2})$$

$$(\text{B.3})$$

où $\delta(\underline{y} - \underline{y}')$ désigne la distribution delta de Dirac. Ce champ f est de variance finie en tout point \underline{x} , grâce à l'hypothèse d'intégrabilité sur le noyau, et sa variance peut être exprimée sous la forme de l'intégrale :

$$\mathbb{E}[f(\underline{x})^2] = \int_{\mathbb{R}^2} |\mathcal{D}(\underline{x}, \underline{y})|^2 d\underline{y} < \infty. \quad (\text{B.4})$$

Remarque (Stationnarité). Lorsque le noyau s'écrit $\mathcal{D}(\underline{x}, \underline{y}) = \mathcal{D}(\underline{x} - \underline{y})$, la covariance ne dépendant que de la différence $\underline{x}' - \underline{x}$, le champ f est alors stationnaire au sens large à l'ordre deux. Nous allons nous placer dans ce cas pour la suite de la description.

Heuristique. Afin que le champ f satisfasse la propriété d'auto-similarité définie en (3.1), un bon candidat pour le noyau \mathcal{D} serait

$$\mathcal{D}_H(\underline{x} - \underline{y}) \sim \|\underline{x} - \underline{y}\|^{H-1}. \quad (\text{B.5})$$

En effet, de manière formelle, on aurait alors, pour tout $a > 0$,

$$f(a\underline{x}) = \int_{\mathbb{R}^2} \|a\underline{x} - \underline{y}\|^{H-1} w(d\underline{y}) = \int_{\mathbb{R}^2} \|a\underline{x} - a\underline{u}\|^{H-1} aw(d\underline{u}) = a^H \int_{\mathbb{R}^2} \|\underline{x} - \underline{u}\|^{H-1} w(d\underline{u}) = a^H f(\underline{x}), \quad (\text{B.6})$$

et le champ f vérifierait alors la propriété d'auto-similarité (3.1). Néanmoins, une fonction puissance $\underline{x} \mapsto \|\underline{x}\|^{H-1}$ n'étant jamais intégrable en même temps en l'origine et à l'infini, ce type de noyau n'est pas de carré intégrable sur \mathbb{R}^2 .

Afin de construire un champ de variance finie, on introduit une régularisation ε autour de l'origine et une longueur de coupure L , qui permettent de définir le noyau $\mathcal{D}_H^{L,\varepsilon} \in L^2(\mathbb{R}^2)$:

$$\mathcal{D}_H^{L,\varepsilon}(\underline{x}) \triangleq \exp(-\|\underline{x}\|^2/2L^2) \sqrt{\|\underline{x}\|^2 + \varepsilon^2}^{H-d/2}. \quad (\text{B.7})$$

Définition B.1 (Modélisation de champ Gaussien stationnaire par convolution). Pour un paramètre d'auto-similarité $0 < H < 1$ donné, et le noyau $\mathcal{D}_H^{L,\varepsilon}(\underline{x})$ de l'Équation (B.7), on définit le champ Gaussien $f_H^{L,\varepsilon}$ par convolution avec un noyau :

$$f_H^{L,\varepsilon}(\underline{x}) = \int_{\mathbb{R}^2} \exp(-\|\underline{x} - \underline{y}\|^2/2L^2) \sqrt{\|\underline{x} - \underline{y}\|^2 + \varepsilon^2}^{H-d/2} w(d\underline{y}), \quad (\text{B.8})$$

de variance

$$\mathbb{E} \left[f_H^{L,\varepsilon}(\underline{x})^2 \right] = \int_{\mathbb{R}^D} \exp(-\|\underline{y}\|^2/L^2) \sqrt{\|\underline{y}\|^2 + \varepsilon^2}^{2H-2} d\underline{y}. \quad (\text{B.9})$$

Remarque. À cause de la régularisation à l'origine et de la longueur de coupure, ce champs n'est qu'asymptotiquement auto-similaire, dans la limite où $L \rightarrow +\infty$ et $\varepsilon \rightarrow 0$.

En pratique un tel champ est simulé en discrétisant le noyau $\mathcal{D}_H^{L,\varepsilon}$ et en le convoluant avec un bruit blanc discret, comme présenté dans l'Algorithme 16. On obtient alors le champ discret $F_H^{L,\varepsilon}$, Gaussien fractionnaire de paramètre d'auto-similarité H . Trois exemples pour des longueurs de coupure différentes sont proposées en Figure B.1.

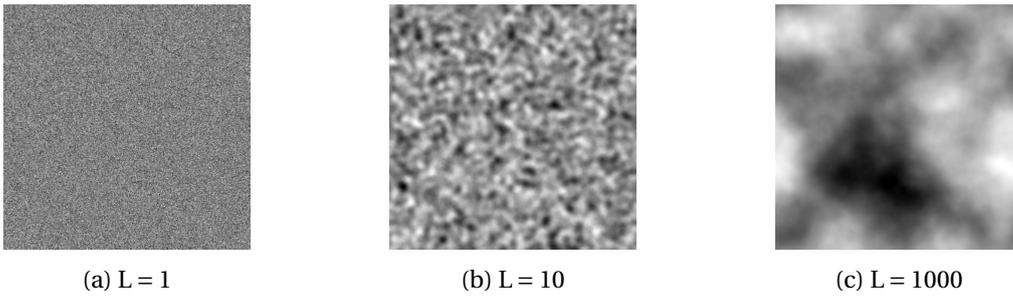


FIGURE B.1 – Champs Gaussiens fractionnaires discrets, $F_H^{L,\varepsilon}$, de paramètre $H = 0.7$ générés par la méthode de la convolution dans le domaine de l'image décrite à l'Algorithme 16 pour différentes longueurs de coupures. Le domaine spatial sur lequel est défini le champ est $[0, 1000]^2$ et la régularisation en l'origine est $\varepsilon = 2$.

Remarque. Il aurait également été possible d'utiliser des patches de textures définies à partir d'un noyau $\mathcal{D}_H^{L,\varepsilon}$, décrites à la Définition B.1, dont trois exemples sont proposés en Figure B.1. Néanmoins, nous verrons dans la suite de cette thèse que les textures g_H de la Définition 3.5 sont plus proches, en terme de rendu visuel, des données réelles sur lesquelles nous avons travaillé. Soulignons également que, si le réglage de L et de ε dans la définition par convolution de $F_H^{L,\varepsilon}$, permet d'obtenir des textures d'aspects extrêmement variés, la question du choix optimal de longueur de coupure et de régularisation reste ouverte.

Algorithme 16 Champ Gaussien autosimilaire par convolution dans le domaine spatial**Paramètres** : L longueur de coupure, ε régularisation en l'origine**Attribut** : H paramètre d'auto-similarité{Bruit blanc de taille $N_1 \times N_2$ }

$$W \sim \mathcal{N}(0, \mathbf{I}) \in \mathbb{R}^{N_1 \times N_2} \quad (\text{B.10})$$

{Noyau $\mathcal{D}_H^{L,\varepsilon}$ échantillonné sur une grille $\Omega = \{1, \dots, N_1\} \times \{1, \dots, N_2\}$ }

$$\forall \underline{n} \in \Omega, \quad D_H(\underline{n}) \triangleq \mathcal{D}_H^{L,\varepsilon}(\underline{n}) \quad (\text{B.11})$$

{Multiplication des transformées de Fourier}

$$\tilde{W} = \text{fft}(W), \quad \tilde{D}_H = \text{fft}(D_H), \quad \tilde{F}_H^{L,\varepsilon} \triangleq \tilde{W} \cdot \tilde{D}_H, \quad (W \cdot D_H)(\underline{q}) \triangleq W(\underline{q})D_H(\underline{q}) \quad (\text{B.12})$$

{Calcul du champ dans l'espace direct}

$$F_H^{L,\varepsilon} \triangleq \text{ifft}(\tilde{F}_H^{L,\varepsilon}) \quad (\text{B.13})$$

Références

- [1] R. M. Pereira, C. Garban, and L. Chevillard. A dissipative random velocity field for fully developed fluid turbulence. *J. Fluid Mech.*, 794 :369–408, 2016. [V](#)

Annexe C

Estimateurs de Stein de l'erreur quadratique et de son gradient

C.1 Démonstration du Theorème 5.1

Démonstration. Par soucis de simplicité, nous définissons le « prédicteur » à la Définition C.1, et la prédiction véritable à la Définition C.2.

Définition C.1 (*Predicteur*). À partir de l'estimateur $\hat{\mathbf{x}}(\mathbf{y}; \Lambda)$ de la quantité sous-jacente $\bar{\mathbf{x}}$ il est possible de considérer de manière *équivalente* un estimateur de *prédiction*

$$\hat{\mathbf{y}}(\mathbf{y}; \Lambda) \triangleq \Phi \hat{\mathbf{x}}(\mathbf{y}; \Lambda), \quad (\text{C.1})$$

que nous appellerons « prédicteur ».

En effet, d'après l'Hypothèse 5.2, $\Phi^* \Phi$ est inversible, et la Relation (C.1) peut donc être inversée en calculant

$$\hat{\mathbf{x}}(\mathbf{y}; \Lambda) = (\Phi^* \Phi)^{-1} \Phi^* \hat{\mathbf{y}}(\mathbf{y}; \Lambda). \quad (\text{C.2})$$

Définition C.2 (*Prédiction véritable*). L'observation non bruitée est définie par

$$\bar{\mathbf{y}} \triangleq \mathbb{E}_{\zeta} \mathbf{y} = \Phi \bar{\mathbf{x}}. \quad (\text{C.3})$$

Ainsi, l'erreur quadratique définie à l'Équation (5.10) peut s'exprimer en utilisant l'opérateur \mathbf{A} défini à l'Équation (5.9), comme

$$R[\hat{\mathbf{x}}](\Lambda) = \mathbb{E}_{\zeta} \|\Pi \hat{\mathbf{x}}(\mathbf{y}; \Lambda) - \Pi \bar{\mathbf{x}}\|_2^2 \quad (\text{C.4})$$

$$= \mathbb{E}_{\zeta} \|\Pi (\Phi^* \Phi)^{-1} \Phi^* (\hat{\mathbf{y}}(\mathbf{y}; \Lambda) - \bar{\mathbf{y}})\|_2^2, \quad (\text{C.5})$$

$$\stackrel{(5.9)}{=} \mathbb{E}_{\zeta} \|\mathbf{A} (\hat{\mathbf{y}}(\mathbf{y}; \Lambda) - \bar{\mathbf{y}})\|_2^2$$

qui sera plus aisé à exprimer dans la suite en terme des observations bruitées, \mathbf{y} , et non bruitées, $\bar{\mathbf{y}}$, et de la « prédiction » $\hat{\mathbf{y}}$.

Par construction, la matrice \mathbf{A} , définie en (5.9), réalise à la fois

- La projection sur le sous-espace d'intérêt \mathcal{I} de \mathcal{H} via l'opérateur linéaire Π .
- Le passage de la quantité prédite $\hat{\mathbf{y}}$ à l'estimée de la quantité sous-jacente $\hat{\mathbf{x}}$, en utilisant la Relation (C.2).

Dans la suite, par soucis de concision, la dépendance de $\hat{\mathbf{x}}$ en $(\mathbf{y}; \Lambda)$ est implicite. À partir du Modèle (5.1) et de l'Hypothèse 5.1 sur la densité de probabilité du bruit, deux relations utiles peuvent être obtenus :

$$\mathbb{E}_{\zeta} \|\mathbf{A}(\mathbf{y} - \bar{\mathbf{y}})\|_2^2 \stackrel{(C.3)}{=} \mathbb{E}_{\zeta} \|\mathbf{A}(\mathbf{y} - \Phi \hat{\mathbf{x}})\|_2^2 = \mathbb{E}_{\zeta} \|\mathbf{A}\zeta\|_2^2 \stackrel{\text{Hyp. 5.1}}{=} \text{Tr}(\mathbf{A}\mathcal{S}\mathbf{A}^*), \quad (C.6)$$

$$\mathbb{E}_{\zeta} \langle \mathbf{A}\mathbf{y}, \mathbf{A}(\mathbf{y} - \bar{\mathbf{y}}) \rangle \stackrel{\mathbb{E}(\mathbf{y} - \bar{\mathbf{y}}) = 0}{=} \mathbb{E}_{\zeta} \langle \mathbf{A}(\mathbf{y} - \bar{\mathbf{y}}), \mathbf{A}(\mathbf{y} - \bar{\mathbf{y}}) \rangle \stackrel{\text{Hyp. 5.1}}{=} \text{Tr}(\mathbf{A}\mathcal{S}\mathbf{A}^*). \quad (C.7)$$

L'erreur quadratique se développe alors en :

$$\begin{aligned} \mathbf{R}[\hat{\mathbf{x}}](\Lambda) &\triangleq \mathbb{E}_{\zeta} \|\mathbf{A}(\hat{\mathbf{y}} - \bar{\mathbf{y}})\|_2^2 \\ &= \mathbb{E}_{\zeta} [\|\mathbf{A}(\hat{\mathbf{y}} - \mathbf{y})\|_2^2 + \|\mathbf{A}(\mathbf{y} - \bar{\mathbf{y}})\|_2^2 + 2\langle \mathbf{A}(\hat{\mathbf{y}} - \mathbf{y}), \mathbf{A}(\mathbf{y} - \bar{\mathbf{y}}) \rangle] \\ &\stackrel{(C.6)}{=} \mathbb{E}_{\zeta} [\|\mathbf{A}(\hat{\mathbf{y}} - \mathbf{y})\|_2^2 + 2\langle \mathbf{A}\hat{\mathbf{y}}, \mathbf{A}(\mathbf{y} - \bar{\mathbf{y}}) \rangle - 2\langle \mathbf{A}\mathbf{y}, \mathbf{A}(\mathbf{y} - \bar{\mathbf{y}}) \rangle] + \text{Tr}(\mathbf{A}\mathcal{S}\mathbf{A}^*) \\ &\stackrel{(C.7)}{=} \mathbb{E}_{\zeta} [\|\mathbf{A}(\hat{\mathbf{y}} - \mathbf{y})\|_2^2 + 2\langle \mathbf{A}\hat{\mathbf{y}}, \mathbf{A}(\mathbf{y} - \bar{\mathbf{y}}) \rangle] - 2\text{Tr}(\mathbf{A}\mathcal{S}\mathbf{A}^*) + \text{Tr}(\mathbf{A}\mathcal{S}\mathbf{A}^*) \\ &= \mathbb{E}_{\zeta} [\|\mathbf{A}(\hat{\mathbf{y}} - \mathbf{y})\|_2^2 + 2\langle \mathbf{A}\hat{\mathbf{y}}, \mathbf{A}(\mathbf{y} - \bar{\mathbf{y}}) \rangle] - \text{Tr}(\mathbf{A}\mathcal{S}\mathbf{A}^*) \\ &= \mathbb{E}_{\zeta} [\|\mathbf{A}(\Phi \hat{\mathbf{x}} - \mathbf{y})\|_2^2 + 2\langle \mathbf{A}^* \mathbf{A} \Phi \hat{\mathbf{x}}, (\mathbf{y} - \Phi \hat{\mathbf{x}}) \rangle] - \text{Tr}(\mathbf{A}\mathcal{S}\mathbf{A}^*) \\ &= \mathbb{E}_{\zeta} \|\mathbf{A}(\Phi \hat{\mathbf{x}} - \mathbf{y})\|_2^2 + 2\mathbb{E}_{\zeta} \langle \mathbf{A}^* \mathbf{A} \Phi \hat{\mathbf{x}}, \zeta \rangle - \text{Tr}(\mathbf{A}\mathcal{S}\mathbf{A}^*) \\ &= \mathbb{E}_{\zeta} \|\mathbf{A}(\Phi \hat{\mathbf{x}} - \mathbf{y})\|_2^2 + 2\mathbb{E}_{\zeta} \langle \mathbf{A}^* \Pi \hat{\mathbf{x}}, \zeta \rangle - \text{Tr}(\mathbf{A}\mathcal{S}\mathbf{A}^*), \end{aligned}$$

en réinjectant l'expression de \mathbf{A} (5.9) en fonction de Φ et Π .

Le deuxième terme, $\mathbb{E}_{\zeta} \langle \mathbf{A}^* \Pi \hat{\mathbf{x}}(\mathbf{y}; \Lambda), \zeta \rangle$, correspond aux *degrés de liberté* [1]. L'Hypothèse 5.3 assure que ce terme est bien défini et s'écrit

$$\mathbb{E}_{\zeta} \langle \mathbf{A}^* \Pi \hat{\mathbf{x}}(\mathbf{y}; \Lambda), \zeta \rangle = \frac{1}{\sqrt{(2\pi)^P |\det(\mathcal{S})|}} \int \langle \mathbf{A}^* \Pi \hat{\mathbf{x}}(\mathbf{y}; \Lambda), \zeta \rangle \exp\left(-\frac{\zeta^* \mathcal{S}^{-1} \zeta}{2}\right) d\zeta, \quad (C.8)$$

nécessitant, par conséquent, le lemme de Stein généralisé pour être estimé.¹

À cause des termes extra-diagonaux de \mathcal{S}^{-1} , l'intégration par parties (IPP) nécessaire pour transformer (C.8) doit être justifiée de manière plus détaillée. Par conséquent, la généralisation du lemme de Stein aux variables aléatoires ζ à valeurs dans \mathcal{G} n'est pas directe.

Nous proposons donc, tout d'abord, de diagonaliser la matrice \mathcal{S}^{-1} , qui est bien diagonalisable car symétrique, pour obtenir

$$\mathcal{S}^{-1} = \mathcal{V}^* \mathcal{D} \mathcal{V},$$

avec \mathcal{V} une matrice orthogonale, dont les colonnes sont les vecteurs propres de \mathcal{S}^{-1} , et $\mathcal{D} = \text{diag}(\beta_1, \dots, \beta_P)$ contenant les valeurs propres, positives, de \mathcal{S}^{-1} . Ensuite, en posant $\boldsymbol{\vartheta} = \mathcal{V} \zeta$, on a

$$\mathbb{E}_{\boldsymbol{\vartheta}} \langle \mathbf{A}^* \Pi \hat{\mathbf{x}}(\mathbf{y}; \Lambda), \boldsymbol{\vartheta} \rangle = \frac{1}{\sqrt{(2\pi)^P |\det(\mathcal{S})|}} \int \langle \mathbf{A}^* \Pi \hat{\mathbf{x}}(\mathbf{y}; \Lambda), \mathcal{V}^{-1} \boldsymbol{\vartheta} \rangle \exp\left(-\frac{\boldsymbol{\vartheta}^* \mathcal{D} \boldsymbol{\vartheta}}{2}\right) |\det(\mathcal{V}^{-1})| d\boldsymbol{\vartheta}.$$

avec $\boldsymbol{\vartheta}^* \mathcal{D} \boldsymbol{\vartheta} = \sum_{p=1}^P \beta_p |\vartheta_p|^2$.

1. Le lemme de Stein assure, pour une variable aléatoire $\zeta \sim \mathcal{N}(0, \sigma^2)$, que si $f : \mathbb{R} \rightarrow \mathbb{R}$ est une fonction telle que $\mathbb{E}_{\zeta}[\zeta f(\zeta)]$ et $\mathbb{E}_{\zeta}[f'(\zeta)]$ existent, alors $\mathbb{E}_{\zeta}[\zeta f(\zeta)] = \sigma^2 \mathbb{E}_{\zeta}[f'(\zeta)]$. Sa démonstration repose sur une intégration par partie appropriée.

Puisque \mathcal{V} est orthogonale, $\mathcal{V}^{-1} = \mathcal{V}^*$ et $|\det(\mathcal{V}^{-1})| = 1$, ce qui induit

$$\begin{aligned} \mathbb{E}_{\boldsymbol{\theta}} \langle \mathbf{A}^* \boldsymbol{\Pi} \hat{\mathbf{x}}(\mathbf{y}; \boldsymbol{\Lambda}), \boldsymbol{\theta} \rangle &= \frac{1}{\sqrt{(2\pi)^P |\det(\mathcal{S})|}} \int \langle \mathcal{V} \mathbf{A}^* \boldsymbol{\Pi} \hat{\mathbf{x}}(\mathbf{y}; \boldsymbol{\Lambda}), \boldsymbol{\theta} \rangle \exp\left(-\frac{\boldsymbol{\theta}^* \mathcal{D} \boldsymbol{\theta}}{2}\right) d\boldsymbol{\theta} \\ &= \frac{1}{\sqrt{(2\pi)^P |\det(\mathcal{S})|}} \int \sum_{p=1}^P (\mathcal{V} \mathbf{A}^* \boldsymbol{\Pi} \hat{\mathbf{x}}(\mathbf{y}; \boldsymbol{\Lambda}))_p \vartheta_p \exp\left(-\frac{\sum_{p=1}^P \beta_p |\vartheta_p|^2}{2}\right) d\vartheta_1 \dots d\vartheta_P \\ &\stackrel{(\text{IPP})}{=} \mathbb{E}_{\boldsymbol{\theta}} \left[\sum_{p=1}^P \frac{1}{\beta_p} \frac{\partial (\mathcal{V} \mathbf{A}^* \boldsymbol{\Pi} \hat{\mathbf{x}}(\mathbf{y}; \boldsymbol{\Lambda}))_p}{\partial \vartheta_p} \right] \\ &= \mathbb{E}_{\boldsymbol{\theta}} \left[\text{Tr} \left(\mathcal{D}^{-1} \frac{\partial (\mathcal{V} \mathbf{A}^* \boldsymbol{\Pi} \hat{\mathbf{x}}(\mathbf{y}; \boldsymbol{\Lambda}))}{\partial \boldsymbol{\theta}} \right) \right], \end{aligned}$$

où $\frac{\partial (\mathcal{V} \mathbf{A}^* \boldsymbol{\Pi} \hat{\mathbf{x}}(\mathbf{y}; \boldsymbol{\Lambda}))}{\partial \boldsymbol{\theta}}$ désigne la matrice Jacobienne de la quantité $\mathcal{V} \mathbf{A}^* \boldsymbol{\Pi} \hat{\mathbf{x}}(\mathbf{y}; \boldsymbol{\Lambda})$ par rapport à la variable $\boldsymbol{\theta} \triangleq \mathcal{V}^{-1} \boldsymbol{\zeta}$. Afin de revenir à la variable $\boldsymbol{\zeta}$, nous utilisons l'Équation (5.1) reliant \mathbf{y} et $\boldsymbol{\zeta}$, appliquons le changement de variable inverse $\boldsymbol{\zeta} \triangleq \mathcal{V} \boldsymbol{\theta}$ et obtenons

$$\frac{\partial (\mathcal{V} \mathbf{A}^* \boldsymbol{\Pi} \hat{\mathbf{x}}(\mathbf{y}; \boldsymbol{\Lambda}))}{\partial \boldsymbol{\theta}} = \mathcal{V} \frac{\partial (\mathbf{A}^* \boldsymbol{\Pi} \hat{\mathbf{x}}(\mathbf{y}; \boldsymbol{\Lambda}))}{\partial \boldsymbol{\zeta}} \mathcal{V}^{-1} = \mathcal{V} \frac{\partial (\mathbf{A}^* \boldsymbol{\Pi} \hat{\mathbf{x}}(\mathbf{y}; \boldsymbol{\Lambda}))}{\partial \mathbf{y}} \mathcal{V}^{-1},$$

grâce à $\partial_{\boldsymbol{\zeta}} \mathbf{y} = \mathbf{I}_P$, \mathbf{I}_P désignant la matrice identité de taille $P \times P$.

La cyclicité de la trace et l'orthogonalité de \mathcal{V} conduisent finalement à l'expression explicite des degrés de liberté suivante :

$$\begin{aligned} \mathbb{E}_{\boldsymbol{\theta}} \langle \mathbf{A}^* \boldsymbol{\Pi} \hat{\mathbf{x}}(\mathbf{y}; \boldsymbol{\Lambda}), \boldsymbol{\theta} \rangle &= \mathbb{E}_{\boldsymbol{\zeta}} \left[\text{Tr} \left(\mathcal{D}^{-1} \mathcal{V} \frac{\partial (\mathbf{A}^* \boldsymbol{\Pi} \hat{\mathbf{x}}(\mathbf{y}; \boldsymbol{\Lambda}))}{\partial \mathbf{y}} \mathcal{V}^{-1} \right) \right], \\ &= \mathbb{E}_{\boldsymbol{\zeta}} \left[\text{Tr} \left(\mathcal{V}^{-1} \mathcal{D}^{-1} \mathcal{V} \frac{\partial (\mathbf{A}^* \boldsymbol{\Pi} \hat{\mathbf{x}}(\mathbf{y}; \boldsymbol{\Lambda}))}{\partial \mathbf{y}} \right) \right], \\ &= \mathbb{E}_{\boldsymbol{\zeta}} \left[\text{Tr} \left(\mathcal{S} \frac{\partial (\mathbf{A}^* \boldsymbol{\Pi} \hat{\mathbf{x}}(\mathbf{y}; \boldsymbol{\Lambda}))}{\partial \mathbf{y}} \right) \right], \\ &= \mathbb{E}_{\boldsymbol{\zeta}} [\text{Tr} (\mathcal{S} \mathbf{A}^* \boldsymbol{\Pi} \partial_{\mathbf{y}} \hat{\mathbf{x}}(\mathbf{y}; \boldsymbol{\Lambda}))] \end{aligned}$$

□

C.2 Démonstration du Théorème 5.2

Démonstration. Remarquons tout d'abord que $\mathbf{y} \mapsto \hat{\mathbf{x}}(\mathbf{y}; \boldsymbol{\Lambda})$ est lipschitzienne, en vertu de l'Hypothèse 5.4, elle est donc Lebesgue différentiable presque partout et sa dérivée de Lebesgue est égale à sa dérivée au sens faible presque partout [2]. Ainsi, en partant du résultat du Théorème 5.1, toute la difficulté réside dans le fait de trouver une domination adéquate du terme de *degrés de liberté*, puisqu'il s'agit du seul terme faisant intervenir le pas de différences finies, v .

En appliquant successivement la méthode de Monte Carlo et la stratégie de différences finies, présentées à la Section 5.2.4 on obtient

$$\begin{aligned} &\text{Tr} (\mathcal{S} \mathbf{A}^* \boldsymbol{\Pi} \partial_{\mathbf{y}} \hat{\mathbf{x}}(\mathbf{y}; \boldsymbol{\Lambda})) \tag{C.9} \\ &\stackrel{\text{Monte Carlo}}{=} \mathbb{E}_{\boldsymbol{\varepsilon}} \left\langle \mathcal{S} \mathbf{A}^* \boldsymbol{\Pi} \frac{\partial \hat{\mathbf{x}}(\mathbf{y}; \boldsymbol{\Lambda})}{\partial \mathbf{y}} [\boldsymbol{\varepsilon}], \boldsymbol{\varepsilon} \right\rangle \\ &\stackrel{\text{Finite Difference}}{=} \mathbb{E}_{\boldsymbol{\varepsilon}} \left\langle \mathbf{A}^* \boldsymbol{\Pi} \lim_{v \rightarrow 0} \frac{\hat{\mathbf{x}}(\mathbf{y} + v\boldsymbol{\varepsilon}; \boldsymbol{\Lambda}) - \hat{\mathbf{x}}(\mathbf{y}; \boldsymbol{\Lambda})}{v}, \mathcal{S} \boldsymbol{\varepsilon} \right\rangle. \end{aligned}$$

Puis, en utilisant la densité de probabilité d'une variable $\boldsymbol{\varepsilon}$, gaussienne centrée réduite, l'espérance ci-dessus s'écrit

$$\mathbb{E}_{\boldsymbol{\varepsilon}} \left\langle \mathbf{A}^* \boldsymbol{\Pi} \lim_{\nu \rightarrow 0} \frac{\widehat{\mathbf{x}}(\mathbf{y} + \nu \boldsymbol{\varepsilon}; \Lambda) - \widehat{\mathbf{x}}(\mathbf{y}; \Lambda)}{\nu}, \mathcal{S}_{\boldsymbol{\varepsilon}} \right\rangle = \int_{\mathbb{R}^P} \lim_{\nu \rightarrow 0} \left\langle \mathbf{A}^* \boldsymbol{\Pi} \frac{\widehat{\mathbf{x}}(\mathbf{y} + \nu \boldsymbol{\varepsilon}; \Lambda) - \widehat{\mathbf{x}}(\mathbf{y}; \Lambda)}{\nu}, \mathcal{S}_{\boldsymbol{\varepsilon}} \right\rangle \frac{e^{-\frac{\|\boldsymbol{\varepsilon}\|^2}{2}} d\boldsymbol{\varepsilon}}{(2\pi)^{P/2}}. \quad (\text{C.10})$$

Puis, on en déduit les majorations suivantes :

$$\begin{aligned} & \left| \left\langle \mathbf{A}^* \boldsymbol{\Pi} \frac{\widehat{\mathbf{x}}(\mathbf{y} + \nu \boldsymbol{\varepsilon}; \Lambda) - \widehat{\mathbf{x}}(\mathbf{y}; \Lambda)}{\nu}, \mathcal{S}_{\boldsymbol{\varepsilon}} \right\rangle \right| e^{-\frac{\|\boldsymbol{\varepsilon}\|^2}{2}} \\ & \stackrel{(\text{CS})}{\leq} \left\| \mathbf{A}^* \boldsymbol{\Pi} \frac{\widehat{\mathbf{x}}(\mathbf{y} + \nu \boldsymbol{\varepsilon}; \Lambda) - \widehat{\mathbf{x}}(\mathbf{y}; \Lambda)}{\nu} \right\| \|\mathcal{S}_{\boldsymbol{\varepsilon}}\| e^{-\frac{\|\boldsymbol{\varepsilon}\|^2}{2}} \\ & \stackrel{(\text{OB})}{\leq} \|\mathbf{A}^*\| \|\boldsymbol{\Pi}\| \left\| \frac{\widehat{\mathbf{x}}(\mathbf{y} + \nu \boldsymbol{\varepsilon}; \Lambda) - \widehat{\mathbf{x}}(\mathbf{y}; \Lambda)}{\nu} \right\| \|\mathcal{S}\| \|\boldsymbol{\varepsilon}\| e^{-\frac{\|\boldsymbol{\varepsilon}\|^2}{2}} \\ & \stackrel{(\text{Hyp. 5.4 : } L_1\text{-Lip.})}{\leq} \|\mathbf{A}^*\| \|\boldsymbol{\Pi}\| L_1 \|\boldsymbol{\varepsilon}\| \|\mathcal{S}\| \|\boldsymbol{\varepsilon}\| e^{-\frac{\|\boldsymbol{\varepsilon}\|^2}{2}}, \end{aligned} \quad (\text{C.11})$$

avec $\|\boldsymbol{\varepsilon}\|^2 e^{-\frac{\|\boldsymbol{\varepsilon}\|^2}{2}}$ intégrable sur \mathbb{R}^P , (CS) désignant la propriété de Cauchy-Schwarz et (OB) la majoration par la norme d'un Opérateur Borné. De plus, la domination étant indépendante de ν , la limite et l'intégrale selon la variable $\boldsymbol{\varepsilon}$ peuvent être interverties, ce qui conduit à

$$\begin{aligned} & \int_{\mathbb{R}^P} \lim_{\nu \rightarrow 0} \left\langle \mathbf{A}^* \boldsymbol{\Pi} \frac{\widehat{\mathbf{x}}(\mathbf{y} + \nu \boldsymbol{\varepsilon}; \Lambda) - \widehat{\mathbf{x}}(\mathbf{y}; \Lambda)}{\nu}, \mathcal{S}_{\boldsymbol{\varepsilon}} \right\rangle \frac{e^{-\frac{\|\boldsymbol{\varepsilon}\|^2}{2}} d\boldsymbol{\varepsilon}}{(2\pi)^{P/2}} \\ & = \lim_{\nu \rightarrow 0} \int_{\mathbb{R}^P} \left\langle \mathbf{A}^* \boldsymbol{\Pi} \frac{\widehat{\mathbf{x}}(\mathbf{y} + \nu \boldsymbol{\varepsilon}; \Lambda) - \widehat{\mathbf{x}}(\mathbf{y}; \Lambda)}{\nu}, \mathcal{S}_{\boldsymbol{\varepsilon}} \right\rangle \frac{e^{-\frac{\|\boldsymbol{\varepsilon}\|^2}{2}} d\boldsymbol{\varepsilon}}{(2\pi)^{P/2}}, \end{aligned} \quad (\text{C.12})$$

et

$$\left| \lim_{\nu \rightarrow 0} \int_{\mathbb{R}^P} \left\langle \mathbf{A}^* \boldsymbol{\Pi} \frac{\widehat{\mathbf{x}}(\mathbf{y} + \nu \boldsymbol{\varepsilon}; \Lambda) - \widehat{\mathbf{x}}(\mathbf{y}; \Lambda)}{\nu}, \mathcal{S}_{\boldsymbol{\varepsilon}} \right\rangle \frac{e^{-\frac{\|\boldsymbol{\varepsilon}\|^2}{2}} d\boldsymbol{\varepsilon}}{(2\pi)^{P/2}} \right| \leq \|\mathbf{A}^*\| \|\boldsymbol{\Pi}\| L_1 \|\mathcal{S}\| \int_{\mathbb{R}^P} \|\boldsymbol{\varepsilon}\|^2 \frac{e^{-\frac{\|\boldsymbol{\varepsilon}\|^2}{2}} d\boldsymbol{\varepsilon}}{(2\pi)^{P/2}} < \infty. \quad (\text{C.13})$$

Puis, l'Équation (C.12) signifie que

$$\text{Tr}(\mathcal{S} \mathbf{A}^* \boldsymbol{\Pi} \partial_{\mathbf{y}} \widehat{\mathbf{x}}(\mathbf{y}; \Lambda)) = \lim_{\nu \rightarrow 0} \mathbb{E}_{\boldsymbol{\varepsilon}} \left\langle \mathbf{A}^* \boldsymbol{\Pi} \frac{\widehat{\mathbf{x}}(\mathbf{y} + \nu \boldsymbol{\varepsilon}; \Lambda) - \widehat{\mathbf{x}}(\mathbf{y}; \Lambda)}{\nu}, \mathcal{S}_{\boldsymbol{\varepsilon}} \right\rangle. \quad (\text{C.14})$$

En outre, la majoration obtenue à l'Équation (C.13) étant indépendante de $\boldsymbol{\zeta}$, puisqu'en vertu de l'Hypothèse 5.4 L_1 est uniforme en \mathbf{y} , ainsi que du pas ν , la limite sur ν et l'espérance par rapport au bruit gaussien $\boldsymbol{\zeta}$ peuvent être interverties

$$\begin{aligned} \mathbb{E}_{\boldsymbol{\zeta}} \text{Tr}(\mathcal{S} \mathbf{A}^* \boldsymbol{\Pi} \partial_{\mathbf{y}} \widehat{\mathbf{x}}(\mathbf{y}; \Lambda)) & = \mathbb{E}_{\boldsymbol{\zeta}} \lim_{\nu \rightarrow 0} \mathbb{E}_{\boldsymbol{\varepsilon}} \left\langle \mathbf{A}^* \boldsymbol{\Pi} \frac{\widehat{\mathbf{x}}(\mathbf{y} + \nu \boldsymbol{\varepsilon}; \Lambda) - \widehat{\mathbf{x}}(\mathbf{y}; \Lambda)}{\nu}, \mathcal{S}_{\boldsymbol{\varepsilon}} \right\rangle \\ & = \lim_{\nu \rightarrow 0} \mathbb{E}_{\boldsymbol{\zeta}, \boldsymbol{\varepsilon}} \left\langle \mathbf{A}^* \boldsymbol{\Pi} \frac{\widehat{\mathbf{x}}(\mathbf{y} + \nu \boldsymbol{\varepsilon}; \Lambda) - \widehat{\mathbf{x}}(\mathbf{y}; \Lambda)}{\nu}, \mathcal{S}_{\boldsymbol{\varepsilon}} \right\rangle. \end{aligned} \quad (\text{C.15})$$

On a donc finalement montré le caractère asymptotiquement non biaisé de l'estimateur des *degrés de liberté* DFMC, et par suite ce caractère s'étend à l'estimateur SURE DFMC défini à l'Équation (5.17). \square

C.3 Démonstration du Théorème 5.3

Démonstration. Rappelons que l'estimateur SUGAR par différences finies et méthode de Monte Carlo est composé de deux termes, notés $(\partial 1)$ et $(\partial 2)$ dans la suite :

$$\partial_{\Lambda} \widehat{R}_{\nu, \boldsymbol{\varepsilon}}(\mathbf{y}; \Lambda | \mathcal{S}) \triangleq \underbrace{2(\mathbf{A} \boldsymbol{\Phi} \partial_{\Lambda} \widehat{\mathbf{x}}(\mathbf{y}; \Lambda))^*}_{(\partial 1)} \mathbf{A}(\boldsymbol{\Phi} \widehat{\mathbf{x}} - \mathbf{y}) + \underbrace{\frac{2}{\nu} \left\langle \mathbf{A}^* \boldsymbol{\Pi} (\partial_{\Lambda} \widehat{\mathbf{x}}(\mathbf{y} + \nu \boldsymbol{\varepsilon}; \Lambda) - \partial_{\Lambda} \widehat{\mathbf{x}}(\mathbf{y}; \Lambda)), \mathcal{S}_{\boldsymbol{\varepsilon}} \right\rangle}_{(\partial 2)}.$$

Puisque l'estimateur $\hat{\mathbf{x}}(\mathbf{y}; \mathbf{\Lambda})$ est faiblement différentiable par rapport à $\mathbf{\Lambda}$, c'est également le cas de l'erreur quadratique $R[\hat{\mathbf{x}}](\mathbf{\Lambda})$. Donc, pour toute fonction test continument différentiable, $\varphi : \mathbb{R}^L \rightarrow \mathbb{R} \in C^1(\mathbb{V})$, dont le support, compact, est noté $\mathbb{V} \subset \mathbb{R}^L$, et toute composante $l \in \{1, \dots, L\}$ du gradient de l'erreur quadratique $\partial_{\mathbf{\Lambda}} R[\hat{\mathbf{x}}](\mathbf{\Lambda})$

$$\begin{aligned}
 \int_{\mathbb{R}^L} (\partial_{\mathbf{\Lambda}} R[\hat{\mathbf{x}}](\mathbf{\Lambda}))_l \varphi(\mathbf{\Lambda}) d\mathbf{\Lambda} &= \int_{\mathbb{V}} (\partial_{\mathbf{\Lambda}} R[\hat{\mathbf{x}}](\mathbf{\Lambda}))_l \varphi(\mathbf{\Lambda}) d\mathbf{\Lambda} & (C.16) \\
 &\stackrel{\text{(Différentiabilité faible)}}{=} - \int_{\mathbb{V}} R[\hat{\mathbf{x}}](\mathbf{\Lambda}) (\partial_{\mathbf{\Lambda}} \varphi(\mathbf{\Lambda}))_l d\mathbf{\Lambda} \\
 &\stackrel{\text{(Définition (5.10))}}{=} - \int_{\mathbb{V}} \mathbb{E}_{\zeta} \|\mathbf{\Pi} \hat{\mathbf{x}}(\mathbf{y}; \mathbf{\Lambda}) - \mathbf{\Pi} \mathbf{x}\|_2^2 (\partial_{\mathbf{\Lambda}} \varphi(\mathbf{\Lambda}))_l d\mathbf{\Lambda} \\
 &\stackrel{\text{(Théorème 5.1)}}{=} - \int_{\mathbb{V}} \mathbb{E}_{\zeta, \varepsilon} \lim_{\nu \rightarrow 0} \hat{R}_{\nu, \varepsilon}(\mathbf{y}; \mathbf{\Lambda} | \mathcal{S}) (\partial_{\mathbf{\Lambda}} \varphi(\mathbf{\Lambda}))_l d\mathbf{\Lambda} \\
 &\stackrel{\text{(Théorème 5.2)}}{=} - \int_{\mathbb{V}} \lim_{\nu \rightarrow 0} \mathbb{E}_{\zeta, \varepsilon} \hat{R}_{\nu, \varepsilon}(\mathbf{y}; \mathbf{\Lambda} | \mathcal{S}) (\partial_{\mathbf{\Lambda}} \varphi(\mathbf{\Lambda}))_l d\mathbf{\Lambda} \\
 &\stackrel{\text{(CD 1)}}{=} - \lim_{\nu \rightarrow 0} \int_{\mathbb{V}} \mathbb{E}_{\zeta, \varepsilon} \hat{R}_{\nu, \varepsilon}(\mathbf{y}; \mathbf{\Lambda} | \mathcal{S}) (\partial_{\mathbf{\Lambda}} \varphi(\mathbf{\Lambda}))_l d\mathbf{\Lambda} \\
 &\stackrel{\text{(Fu 1)}}{=} - \lim_{\nu \rightarrow 0} \mathbb{E}_{\zeta, \varepsilon} \int_{\mathbb{V}} \hat{R}_{\nu, \varepsilon}(\mathbf{y}; \mathbf{\Lambda} | \mathcal{S}) (\partial_{\mathbf{\Lambda}} \varphi(\mathbf{\Lambda}))_l d\mathbf{\Lambda} \\
 &\stackrel{\text{(Proposition 5.1)}}{=} \lim_{\nu \rightarrow 0} \mathbb{E}_{\zeta, \varepsilon} \int_{\mathbb{V}} (\partial_{\mathbf{\Lambda}} \hat{R}_{\nu, \varepsilon}(\mathbf{y}; \mathbf{\Lambda} | \mathcal{S}))_l \varphi(\mathbf{\Lambda}) d\mathbf{\Lambda} \\
 &\stackrel{\text{(Fu 2)}}{=} \lim_{\nu \rightarrow 0} \int_{\mathbb{V}} \mathbb{E}_{\zeta, \varepsilon} (\partial_{\mathbf{\Lambda}} \hat{R}_{\nu, \varepsilon}(\mathbf{y}; \mathbf{\Lambda} | \mathcal{S}))_l \varphi(\mathbf{\Lambda}) d\mathbf{\Lambda} \\
 &\stackrel{\text{(CD 2)}}{=} \int_{\mathbb{V}} \lim_{\nu \rightarrow 0} \mathbb{E}_{\zeta, \varepsilon} (\partial_{\mathbf{\Lambda}} \hat{R}_{\nu, \varepsilon}(\mathbf{y}; \mathbf{\Lambda} | \mathcal{S}))_l \varphi(\mathbf{\Lambda}) d\mathbf{\Lambda}.
 \end{aligned}$$

(CD 1) Afin d'appliquer le théorème de convergence dominée et d'invertir la limite sur ν et l'intégrale sur \mathbb{V} , φ étant une fonction test à support compact, il est nécessaire de borner $\mathbb{E}_{\zeta, \varepsilon} \hat{R}_{\nu, \varepsilon}(\mathbf{y}; \mathbf{\Lambda} | \mathcal{S})$, de façon indépendante à la fois de ν et de $\mathbf{\Lambda}$. En utilisant la densité de probabilité du bruit

$$\mathbb{E}_{\zeta, \varepsilon} \hat{R}_{\nu, \varepsilon}(\mathbf{y}; \mathbf{\Lambda} | \mathcal{S}) = \int_{\zeta} \int_{\varepsilon} \hat{R}_{\nu, \varepsilon}(\mathbf{y}; \mathbf{\Lambda} | \mathcal{S}) \mathcal{G}_{\mathcal{S}}(\zeta) \mathcal{G}_{\mathbf{I}}(\varepsilon) d\zeta d\varepsilon \quad (C.17)$$

où $\mathcal{G}_{\mathcal{S}}$ (resp. $\mathcal{G}_{\mathbf{I}}$) désigne la densité de probabilité gaussienne de matrice de covariance \mathcal{S} (resp. \mathbf{I})

$$\mathcal{G}_{\mathcal{S}}(\zeta) \triangleq \frac{\exp(-\|\zeta\|_{\mathcal{S}^{-1}}^2/2)}{\sqrt{(2\pi)^P |\det \mathcal{S}|}} \quad \left(\text{resp. } \mathcal{G}_{\mathbf{I}}(\zeta) \triangleq \frac{\exp(-\|\zeta\|_2^2/2)}{\sqrt{(2\pi)^P}} \right). \quad (C.18)$$

L'estimateur $\hat{R}_{\nu, \varepsilon}(\mathbf{y}; \mathbf{\Lambda} | \mathcal{S})$ se décompose en trois termes

$$\hat{R}_{\nu, \varepsilon}(\mathbf{y}; \mathbf{\Lambda} | \mathcal{S}) \triangleq \underbrace{\|\mathbf{A}(\Phi \hat{\mathbf{x}} - \mathbf{y})\|_2^2}_{(1)} + \frac{1}{\nu} \underbrace{\langle \mathbf{A}^* \mathbf{\Pi} (\hat{\mathbf{x}}(\mathbf{y} + \nu \varepsilon; \mathbf{\Lambda} | \mathcal{S}) - \hat{\mathbf{x}}(\mathbf{y}; \mathbf{\Lambda} | \mathcal{S})), \mathcal{S} \varepsilon \rangle}_{(2)} - \underbrace{\text{Tr}(\mathbf{A} \mathcal{S} \mathbf{A}^*)}_{(3)}, \quad (C.19)$$

qui seront bornés séparément.

(1) Dans un premier temps, combiner les affirmations (i) et (ii) de l'Hypothèse 5.4, conduit à

$$\|\hat{\mathbf{x}}(\mathbf{y}; \mathbf{\Lambda}) - \hat{\mathbf{x}}(\mathbf{0}_P; \mathbf{\Lambda})\| \leq L_1 \|\mathbf{y} - \mathbf{0}_P\| \implies \|\hat{\mathbf{x}}(\mathbf{y}; \mathbf{\Lambda})\| \leq L_1 \|\mathbf{y}\|, \quad (C.20)$$

qui peut être utilisé pour borner le premier terme (1) de (C.19) par

$$\begin{aligned}
 \|\mathbf{A}(\Phi \hat{\mathbf{x}} - \mathbf{y})\| &\leq \|\mathbf{A}\| \|\Phi \hat{\mathbf{x}} - \mathbf{y}\| \\
 &\leq \|\mathbf{A}\| (\|\Phi \hat{\mathbf{x}}\| + \|\mathbf{y}\|) \\
 &\leq \|\mathbf{A}\| (\|\Phi\| \|\hat{\mathbf{x}}\| + \|\mathbf{y}\|) \\
 &\leq \|\mathbf{A}\| (\|\Phi\|_{L_1} \|\mathbf{y}\| + \|\mathbf{y}\|) \\
 &\leq \|\mathbf{A}\| (\|\Phi\|_{L_1} + 1) \|\mathbf{y}\|.
 \end{aligned} \quad (C.21)$$

Puisque, par définition, $\zeta = \mathbf{y} - \Phi \mathbf{x}$, $\mathbf{y} \mapsto \|\mathbf{y}\|$ est intégrable par rapport à la distribution de probabilité gaussienne $\mathcal{G}_{\mathcal{S}}(\zeta)$, et la majoration ci-dessus étant indépendante de \mathbf{v} , le théorème de convergence dominée s'applique.

(2) La majoration de l'Équation (C.11) induit

$$\left| \left\langle \mathbf{A}^* \Pi \frac{\widehat{\mathbf{x}}(\mathbf{y} + \mathbf{v}\boldsymbol{\varepsilon}; \Lambda) - \widehat{\mathbf{x}}(\mathbf{y}; \Lambda)}{\mathbf{v}}, \mathcal{S}\boldsymbol{\varepsilon} \right\rangle \right| \leq \|\mathbf{A}^*\| \|\Pi\|_{L_1} \|\boldsymbol{\varepsilon}\| \|\mathcal{S}\boldsymbol{\varepsilon}\|, \quad (\text{C.22})$$

et $\|\boldsymbol{\varepsilon}\|^2$ étant intégrable par rapport à $\mathcal{G}_{\mathbf{I}}(\boldsymbol{\varepsilon})$, le théorème de convergence dominée s'applique.

(3) Le troisième terme étant constant sa majoration est triviale.

En réunissant les majorations (1), (2) et (3) :

$$\begin{aligned} |\mathbb{E}_{\zeta, \boldsymbol{\varepsilon}} \widehat{\mathbf{R}}_{\mathbf{v}, \boldsymbol{\varepsilon}}(\mathbf{y}; \Lambda | \mathcal{S})| &\leq \int_{\zeta} \int_{\boldsymbol{\varepsilon}} \|\mathbf{A}\| (\|\Phi\|_{L_1} + 1) \|\mathbf{y}\| \mathcal{G}_{\mathcal{S}}(\zeta) \mathcal{G}_{\mathbf{I}}(\boldsymbol{\varepsilon}) d\zeta d\boldsymbol{\varepsilon} \\ &+ \int_{\zeta} \int_{\boldsymbol{\varepsilon}} \|\mathbf{A}^*\| \|\Pi\|_{L_1} \|\mathcal{S}\boldsymbol{\varepsilon}\|^2 \mathcal{G}_{\mathcal{S}}(\zeta) \mathcal{G}_{\mathbf{I}}(\boldsymbol{\varepsilon}) d\zeta d\boldsymbol{\varepsilon} + \int_{\zeta} \int_{\boldsymbol{\varepsilon}} \text{Tr}(\mathbf{A} \mathcal{S} \mathbf{A}^*) \mathcal{G}_{\mathcal{S}}(\zeta) \mathcal{G}_{\mathbf{I}}(\boldsymbol{\varepsilon}) d\zeta d\boldsymbol{\varepsilon} \leq \infty, \end{aligned} \quad (\text{C.23})$$

cette majoration étant indépendante à la fois de \mathbf{v} et de Λ , le théorème de convergence dominée s'applique.

(Fu 1) La domination obtenue à l'Équation (C.23) étant indépendante Λ , le théorème de Fubini permet de réaliser l'interversion.

(Fu 2) Le premier terme de (C.16), noté $(\partial 1)$ peut être majoré facilement par une fonction intégrable. En effet, d'après l'Hypothèse 5.5, $\partial_{\Lambda} \widehat{\mathbf{x}}(\mathbf{y}; \Lambda)$ est borné uniformément par L_2 , indépendamment de \mathbf{y} . Il s'ensuit en appliquant l'inégalité de Cauchy-Schwarz et la majoration de l'Équation (C.21) que

$$\begin{aligned} 2 \|(\mathbf{A} \Phi \partial_{\Lambda} \widehat{\mathbf{x}}(\mathbf{y}; \Lambda))^* \mathbf{A} (\Phi \widehat{\mathbf{x}} - \mathbf{y})\| &\leq 2 \|\mathbf{A} \Phi \partial_{\Lambda} \widehat{\mathbf{x}}(\mathbf{y}; \Lambda)\| \|\mathbf{A} (\Phi \widehat{\mathbf{x}} - \mathbf{y})\|, \\ &\leq 2 \|\mathbf{A}\| \|\Phi\|_{L_2} \|\mathbf{A}\| (\Phi_{L_1} + 1) \|\mathbf{y}\|. \end{aligned}$$

donc, puisque $\|\mathbf{y}\|$ est intégrable par rapport à $\mathcal{G}_{\mathcal{S}}(\mathbf{y} - \Phi \mathbf{x}) \mathcal{G}_{\mathbf{I}}(\boldsymbol{\varepsilon})$, et que la majoration est indépendante de \mathbf{v} , les théorèmes de Fubini et de convergence dominée s'appliquent.

Le deuxième terme de (C.16), noté $(\partial 2)$, correspond à la dérivée des *degrés de liberté*, et peut se réécrire comme :

$$\frac{2}{\mathbf{v}} \langle \mathbf{A}^* \Pi (\partial_{\Lambda} \widehat{\mathbf{x}}(\mathbf{y} + \mathbf{v}\boldsymbol{\varepsilon}; \Lambda) - \partial_{\Lambda} \widehat{\mathbf{x}}(\mathbf{y}; \Lambda)), \mathcal{S}\boldsymbol{\varepsilon} \rangle \triangleq \frac{2}{\mathbf{v}} \langle u(\zeta + \mathbf{v}\boldsymbol{\varepsilon}; \Lambda) - u(\zeta; \Lambda), \boldsymbol{\varepsilon} \rangle$$

où $u(\mathbf{z}; \Lambda) \triangleq \mathcal{S} \mathbf{A}^* \Pi \partial_{\Lambda} \widehat{\mathbf{x}}(\Phi \mathbf{x} + \mathbf{z}; \Lambda)$. Puisque $\partial_{\Lambda} \widehat{\mathbf{x}}(\mathbf{y}; \Lambda)$ est uniformément borné par L_2 , indépendamment de \mathbf{y} , et que tous les opérateurs linéaires sont bornés, $\Lambda \mapsto u(\mathbf{z}; \Lambda)$ est borné par une constante $L_u > 0$, indépendante de \mathbf{z} . Donc

$$\begin{aligned} \mathbb{E}_{\zeta, \boldsymbol{\varepsilon}} \left[\frac{2}{\mathbf{v}} \langle \mathbf{A}^* \Pi (\partial_{\Lambda} \widehat{\mathbf{x}}(\mathbf{y} + \mathbf{v}\boldsymbol{\varepsilon}; \Lambda) - \partial_{\Lambda} \widehat{\mathbf{x}}(\mathbf{y}; \Lambda)), \mathcal{S}\boldsymbol{\varepsilon} \rangle \right] &= \int_{\zeta} \int_{\boldsymbol{\varepsilon}} \frac{2}{\mathbf{v}} \langle u(\zeta + \mathbf{v}\boldsymbol{\varepsilon}; \Lambda) - u(\zeta; \Lambda), \boldsymbol{\varepsilon} \rangle \mathcal{G}_{\mathcal{S}}(\zeta) \mathcal{G}_{\mathbf{I}}(\boldsymbol{\varepsilon}) d\zeta d\boldsymbol{\varepsilon} \\ &= \int_{\zeta} \int_{\boldsymbol{\varepsilon}} \frac{2}{\mathbf{v}} \langle u(\zeta + \mathbf{v}\boldsymbol{\varepsilon}; \Lambda), \boldsymbol{\varepsilon} \rangle \mathcal{G}_{\mathcal{S}}(\zeta) \mathcal{G}_{\mathbf{I}}(\boldsymbol{\varepsilon}) d\zeta d\boldsymbol{\varepsilon} - \int_{\zeta} \int_{\boldsymbol{\varepsilon}} \frac{2}{\mathbf{v}} \langle u(\zeta; \Lambda), \boldsymbol{\varepsilon} \rangle \mathcal{G}_{\mathcal{S}}(\zeta) \mathcal{G}_{\mathbf{I}}(\boldsymbol{\varepsilon}) d\zeta d\boldsymbol{\varepsilon} \\ &= \int_{\zeta} \int_{\boldsymbol{\varepsilon}} \frac{2}{\mathbf{v}} \langle u(\zeta; \Lambda), \boldsymbol{\varepsilon} \rangle (\mathcal{G}_{\mathcal{S}}(\zeta - \mathbf{v}\boldsymbol{\varepsilon}) - \mathcal{G}_{\mathcal{S}}(\zeta)) \mathcal{G}_{\mathbf{I}}(\boldsymbol{\varepsilon}) d\zeta d\boldsymbol{\varepsilon}. \end{aligned} \quad (\text{C.24})$$

De plus, nous disposons des majorations suivantes

$$\begin{aligned} & \left\| \frac{2}{\nu} \langle u(\boldsymbol{\zeta}; \boldsymbol{\Lambda}), \boldsymbol{\varepsilon} \rangle (\mathcal{G}_{\mathcal{F}}(\boldsymbol{\zeta} - \nu \boldsymbol{\varepsilon}) - \mathcal{G}_{\mathcal{F}}(\boldsymbol{\zeta})) \mathcal{G}_{\mathbf{I}}(\boldsymbol{\varepsilon}) \right\| \\ & \leq \frac{2}{\nu} L_u \|\boldsymbol{\varepsilon}\| |\mathcal{G}_{\mathcal{F}}(\boldsymbol{\zeta} - \nu \boldsymbol{\varepsilon}) - \mathcal{G}_{\mathcal{F}}(\boldsymbol{\zeta})| \mathcal{G}_{\mathbf{I}}(\boldsymbol{\varepsilon}). \end{aligned} \quad (\text{C.25})$$

Ainsi, à un changement de variable unitaire près (voir Section C.1, avec $\boldsymbol{\Theta} = \mathcal{V}^{-1} \boldsymbol{\varepsilon}$), la matrice de covariance peut être supposée *diagonale*, avec des termes diagonaux $(s_i^2)_{i=1}^P$, de sorte que

$$\mathcal{G}_{\mathcal{F}}(\boldsymbol{\zeta}) = \prod_{i=1}^P \frac{\exp(-|\zeta_i|^2 / 2s_i^2)}{\sqrt{2\pi s_i^2}} \quad (\text{C.26})$$

les densités gaussiennes unidimensionnelles sont alors définies par

$$g_{s_i^2}(\zeta_i) \triangleq \frac{\exp(-|\zeta_i|^2 / 2s_i^2)}{\sqrt{2\pi s_i^2}}. \quad (\text{C.27})$$

En appliquant l'inégalité de Taylor,

$$|g_{s_i^2}(\zeta_i - \nu \varepsilon_i) - g_{s_i^2}(\zeta_i)| \leq \int_{(0, \nu \varepsilon_i)} |g'_{s_i^2}(\zeta_i - \tau)| d\tau, \quad (\text{C.28})$$

où $(0, \nu \varepsilon_i)$ désigne l'intervalle ordonné, permettant de prendre en compte que ε_i peut être négatif,

$$(0, \nu \varepsilon_i) = \begin{cases} [0, \nu \varepsilon_i] & \text{if } \varepsilon_i \geq 0 \\ [\nu \varepsilon_i, 0] & \text{else} \end{cases} \quad (\text{C.29})$$

puis

$$\begin{aligned} & \int_{\zeta_i} |g_{s_i^2}(\zeta_i - \nu \varepsilon_i) - g_{s_i^2}(\zeta_i)| d\zeta_i \\ & \leq \int_{\zeta_i} \int_{(0, \nu \varepsilon_i)} |g'_{s_i^2}(\zeta_i - \tau)| d\tau d\zeta_i \\ & \leq \int_{(0, \nu \varepsilon_i)} \left(\int_{\zeta_i} |g'_{s_i^2}(\zeta_i - \tau)| d\zeta_i \right) d\tau \\ & = \left(\int_{\mathbb{R}} |g'_{s_i^2}(t)| dt \right) \nu |\varepsilon_i| < +\infty \end{aligned}$$

la dérivée de la densité gaussienne étant intégrable sur \mathbb{R} .

En revenant à l'intégrale par rapport aux variables $\boldsymbol{\zeta}, \boldsymbol{\varepsilon} \in \mathbb{R}^P$ de l'Équation (C.24)

$$\begin{aligned} & \left\| \int_{\boldsymbol{\zeta}} \int_{\boldsymbol{\varepsilon}} \frac{2}{\nu} \langle u(\boldsymbol{\zeta}; \boldsymbol{\Lambda}), \boldsymbol{\varepsilon} \rangle (\mathcal{G}_{\mathcal{F}}(\boldsymbol{\zeta} - \nu \boldsymbol{\varepsilon}) - \mathcal{G}_{\mathcal{F}}(\boldsymbol{\zeta})) \mathcal{G}_{\mathbf{I}}(\boldsymbol{\varepsilon}) d\boldsymbol{\zeta} d\boldsymbol{\varepsilon} \right\| \\ & \leq \int_{\boldsymbol{\zeta}} \int_{\boldsymbol{\varepsilon}} \frac{2}{\nu} L_u \|\boldsymbol{\varepsilon}\| |\mathcal{G}_{\mathcal{F}}(\boldsymbol{\zeta} - \nu \boldsymbol{\varepsilon}) - \mathcal{G}_{\mathcal{F}}(\boldsymbol{\zeta})| \mathcal{G}_{\mathbf{I}}(\boldsymbol{\varepsilon}) d\boldsymbol{\zeta} d\boldsymbol{\varepsilon} \\ & = \int_{\boldsymbol{\varepsilon}} \frac{2}{\nu} L_u \|\boldsymbol{\varepsilon}\| \prod_{i=1}^P \left(\int_{\zeta_i} |g_{s_i^2}(\zeta_i - \nu \varepsilon_i) - g_{s_i^2}(\zeta_i)| d\zeta_i \right) \mathcal{G}_{\mathbf{I}}(\boldsymbol{\varepsilon}) d\boldsymbol{\varepsilon} \\ & \leq \int_{\boldsymbol{\varepsilon}} \frac{2}{\nu} L_u \|\boldsymbol{\varepsilon}\| \prod_{i=1}^P \left(\left(\int_{\mathbb{R}} |g'_{s_i^2}(t)| dt \right) \nu |\varepsilon_i| \right) \mathcal{G}_{\mathbf{I}}(\boldsymbol{\varepsilon}) d\boldsymbol{\varepsilon} \\ & \stackrel{|\varepsilon_i| \leq \|\boldsymbol{\varepsilon}\|}{\leq} \int_{\boldsymbol{\varepsilon}} 2\nu^{P-1} \|L_u \boldsymbol{\varepsilon}\| \|\boldsymbol{\varepsilon}\|^P \prod_{i=1}^P \left(\int_{\mathbb{R}} |g'_{s_i^2}(t)| dt \right) \mathcal{G}_{\mathbf{I}}(\boldsymbol{\varepsilon}) d\boldsymbol{\varepsilon} \\ & \stackrel{(0 < \nu \leq 1)}{\leq} \int_{\boldsymbol{\varepsilon}} 2L_u \|\boldsymbol{\varepsilon}\|^{P+1} \prod_{i=1}^P \left(\int_{\mathbb{R}} |g'_{s_i^2}(t)| dt \right) \mathcal{G}_{\mathbf{I}}(\boldsymbol{\varepsilon}) d\boldsymbol{\varepsilon} < +\infty. \end{aligned} \quad (\text{C.30})$$

En effet, $\|\cdot\|$ désignant la norme euclidienne ($\forall i \mid \varepsilon_i \leq \|\varepsilon\|$). De plus, nous nous intéressons à la limite $v \rightarrow 0$, sans perte de généralité, nous pouvons supposer que $0 > v \leq 1$, et par conséquent $v^{p-1} \leq 1$. Le fait que toute puissance de $\|\varepsilon\|$ soit intégrable par rapport à $\mathcal{G}_1(\varepsilon)$, combiné à la compacité du support \mathbb{V} de φ , permet de conclure en appliquant le théorème de Fubini pour intervertir $\int_{\mathbb{V}}$ et $\mathbb{E}_{\zeta, \varepsilon}$.

(CD 2) La majoration ci-dessus est indépendante de v . De plus, l'Hypothèse de Lipschitzianité 5.5, assure l'existence P- p .s. de

$$\lim_{v \rightarrow 0} \mathbb{E}_{\zeta, \varepsilon} \left(\partial_{\Lambda} \widehat{R}_{v, \varepsilon}(\mathbf{y}; \Lambda \mid \mathcal{S}) \right)_l$$

et le théorème de convergence dominée s'applique et permet d'échanger $\lim_{v \rightarrow 0}$ avec $\int_{\mathbb{V}}$ ce qui termine la démonstration. □

C.4 Démonstration de la Proposition 5.4

Démonstration. Grâce à la cyclicité de la trace,

$$\text{Tr}(\mathbf{A} \mathcal{S} \mathbf{A}^*) = \text{Tr}(\mathbf{A}^* \mathbf{A} \mathcal{S}),$$

puis en utilisant $\mathbf{A} \triangleq \mathbf{\Pi}(\mathbf{\Phi}^* \mathbf{\Phi})^{-1} \mathbf{\Phi}^*$,

$$\mathbf{A}^* \mathbf{A} = \mathbf{\Phi}(\mathbf{\Phi}^* \mathbf{\Phi})^{-1} \mathbf{\Pi}^* \mathbf{\Pi}(\mathbf{\Phi}^* \mathbf{\Phi})^{-1} \mathbf{\Phi}^*.$$

Afin de calculer les produits matriciels, les matrices sont mises sous forme de matrices par blocs, ce qui conduit à

$$\begin{aligned} (\mathbf{\Phi}^* \mathbf{\Phi})^{-1} \mathbf{\Pi}^* \mathbf{\Pi}(\mathbf{\Phi}^* \mathbf{\Phi})^{-1} &= \frac{1}{(F_0 F_2 - F_1^2)^2} \begin{pmatrix} F_0 \mathbf{I}_{N/2} & -F_1 \mathbf{I}_{N/2} \\ -F_1 \mathbf{I}_{N/2} & F_2 \mathbf{I}_{N/2} \end{pmatrix} \begin{pmatrix} \mathbf{I}_{N/2} & \mathbf{Z}_{N/2} \\ \mathbf{Z}_{N/2} & \mathbf{I}_{N/2} \end{pmatrix} \begin{pmatrix} F_0 \mathbf{I}_{N/2} & -F_1 \mathbf{I}_{N/2} \\ -F_1 \mathbf{I}_{N/2} & F_2 \mathbf{I}_{N/2} \end{pmatrix} \\ &= \frac{1}{(F_0 F_2 - F_1^2)^2} \begin{pmatrix} F_0 \mathbf{I}_{N/2} & -F_1 \mathbf{I}_{N/2} \\ -F_1 \mathbf{I}_{N/2} & F_2 \mathbf{I}_{N/2} \end{pmatrix} \begin{pmatrix} F_0 \mathbf{Z}_{N/2} & -F_1 \mathbf{I}_{N/2} \\ \mathbf{Z}_{N/2} & \mathbf{I}_{N/2} \end{pmatrix} \\ &= \frac{1}{(F_0 F_2 - F_1^2)^2} \begin{pmatrix} F_0^2 \mathbf{I}_{N/2} & -F_0 F_1 \mathbf{I}_{N/2} \\ -F_0 F_1 \mathbf{I}_{N/2} & F_1^2 \mathbf{I}_{N/2} \end{pmatrix}. \end{aligned}$$

En utilisant à nouveau la cyclicité de la trace

$$\begin{aligned} \text{Tr}(\mathbf{A} \mathcal{S} \mathbf{A}^*) &= \text{Tr} \left(\mathbf{\Phi}(\mathbf{\Phi}^* \mathbf{\Phi})^{-1} \mathbf{\Pi}^* \mathbf{\Pi}(\mathbf{\Phi}^* \mathbf{\Phi})^{-1} \mathbf{\Phi}^* \mathcal{S} \right) \\ &= \text{Tr} \left(\mathbf{\Phi} \frac{1}{(F_0 F_2 - F_1^2)^2} \begin{pmatrix} F_0^2 \mathbf{I}_{N/2} & -F_0 F_1 \mathbf{I}_{N/2} \\ -F_0 F_1 \mathbf{I}_{N/2} & F_1^2 \mathbf{I}_{N/2} \end{pmatrix} \mathbf{\Phi}^* \mathcal{S} \right) \\ &= \text{Tr} \left(\frac{1}{(F_0 F_2 - F_1^2)^2} \begin{pmatrix} F_0^2 \mathbf{I}_{N/2} & -F_0 F_1 \mathbf{I}_{N/2} \\ -F_0 F_1 \mathbf{I}_{N/2} & F_1^2 \mathbf{I}_{N/2} \end{pmatrix} \mathbf{\Phi}^* \mathcal{S} \mathbf{\Phi} \right) \end{aligned}$$

Puis, en considérant l'action de $\mathbf{\Phi}$ et de $\mathbf{\Phi}^*$, explicitée à l'Équation (4.3) la représentation matricielle suivante est obtenue

$$\mathbf{\Phi} = \begin{pmatrix} \mathbf{I}_{N/2} & \mathbf{I}_{N/2} \\ 2\mathbf{I}_{N/2} & \mathbf{I}_{N/2} \\ \vdots & \vdots \\ \mathbf{J}\mathbf{I}_{N/2} & \mathbf{I}_{N/2} \end{pmatrix} \in \mathbb{R}^{\mathbf{J}N_1 N_2 \times 2N_1 N_2}. \quad (\text{C.31})$$

\mathcal{S} est alors décomposée en J^2 blocs $\mathcal{S}_j^{j'} = \mathcal{C}_j^{j'} \Xi_j^{j'} \in \mathbb{R}^{N/2 \times N/2}$ pour obtenir

$$\Phi^* \mathcal{S} \Phi = \begin{pmatrix} \sum_{j,j'} j j' \mathcal{S}_j^{j'} & \sum_{j,j'} j' \mathcal{S}_j^{j'} \\ \sum_{j,j'} j \mathcal{S}_j^{j'} & \sum_{j,j'} \mathcal{S}_j^{j'} \end{pmatrix}, \quad 1 \leq j, j' \leq J. \quad (\text{C.32})$$

Les deux membres de l'Égalité (C.31) sont multipliés par $(F_0 F_2 - F_1^2)^2$, pour amener à

$$\begin{aligned} (F_0 F_2 - F_1^2)^2 \text{Tr}(\mathbf{A} \mathcal{S} \mathbf{A}^*) &= \text{Tr} \left(\begin{pmatrix} F_0^2 \mathbf{I}_{N/2} & -F_0 F_1 \mathbf{I}_{N/2} \\ -F_0 F_1 \mathbf{I}_{N/2} & F_1^2 \mathbf{I}_{N/2} \end{pmatrix} \begin{pmatrix} \sum_{j,j'} j j' \mathcal{S}_j^{j'} & \sum_{j,j'} j' \mathcal{S}_j^{j'} \\ \sum_{j,j'} j \mathcal{S}_j^{j'} & \sum_{j,j'} \mathcal{S}_j^{j'} \end{pmatrix} \right) \\ &= \text{Tr} \begin{pmatrix} \sum_{j,j'} F_0^2 j j' \mathcal{S}_j^{j'} - F_0 F_1 j' \mathcal{S}_j^{j'} & \sum_{j,j'} F_0 j' \mathcal{S}_j^{j'} - F_0 F_1 \mathcal{S}_j^{j'} \\ \sum_{j,j'} F_1^2 j \mathcal{S}_j^{j'} - F_0 F_1 j j' \mathcal{S}_j^{j'} & \sum_{j,j'} F_1^2 \mathcal{S}_j^{j'} - F_0 F_1 j' \mathcal{S}_j^{j'} \end{pmatrix} \\ &= \text{Tr} \left(\sum_{j,j'} \left(F_0^2 j j' \mathcal{S}_j^{j'} - F_0 F_1 j' \mathcal{S}_j^{j'} + F_1^2 \mathcal{S}_j^{j'} - F_0 F_1 j' \mathcal{S}_j^{j'} \right) \right). \end{aligned} \quad (\text{C.33})$$

Nous remarquons alors que

$$\text{Tr}(\mathcal{S}_j^{j'}) \triangleq \sum_{\underline{n} \in \Omega} \mathcal{S}_{j,\underline{n}}^{j',\underline{n}} = \sum_{\underline{n} \in \Omega} \mathcal{C}_j^{j'} = \frac{N}{2} \mathcal{C}_j^{j'}. \quad (\text{C.34})$$

puisque le filtre $\Xi_j^{j'}$ est supposé normalisé, au sens où sa valeur maximale est un. Et finalement

$$\text{Tr}(\mathbf{A} \mathcal{S} \mathbf{A}^*) = \frac{N/2}{(F_0 F_2 - F_1^2)^2} \sum_{j,j'} \left(F_0^2 j j' \mathcal{C}_j^{j'} - F_0 F_1 j' \mathcal{C}_j^{j'} + F_1^2 \mathcal{C}_j^{j'} - F_0 F_1 j' \mathcal{C}_j^{j'} \right). \quad (\text{C.35})$$

□

Références

- [1] B. Efron. How biased is the apparent error rate of a prediction rule? *J. Am. Stat. Assoc.*, 81(394) : 461–470, 1986. [X](#)
- [2] L. C. Evans and R. F. Gariepy. *Measure theory and fine properties of functions*. Chapman and Hall/CRC, 2015. [XI](#)