



HAL
open science

Restauration et reconnaissance des formes dans des images avec bruit de Poisson : application à l'analyse des neurones en microscopies de fluorescence

François-Xavier Dupé

► To cite this version:

François-Xavier Dupé. Restauration et reconnaissance des formes dans des images avec bruit de Poisson : application à l'analyse des neurones en microscopies de fluorescence. Traitement des images [eess.IV]. Université de Caen, 2010. Français. NNT: . tel-01075453

HAL Id: tel-01075453

<https://hal.science/tel-01075453>

Submitted on 17 Oct 2014

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

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



Université de Caen
Basse-Normandie

UNIVERSITÉ de CAEN/BASSE-NORMANDIE
U.F.R. de Sciences
ÉCOLE DOCTORALE SIMEM

THÈSE

Présentée par

M. François-Xavier Dupé

et soutenue

le 25 janvier 2010

en vue de l'obtention du
DOCTORAT de l'UNIVERSITÉ de CAEN
spécialité : **Traitement du Signal et des Images**
(Arrêté du 07 août 2006)

Restauration et reconnaissance des formes dans des images avec bruit de Poisson : application à l'analyse des neurones en microscopie de fluorescence.

MEMBRES du JURY

Mme. Laure Blanc-Féraud, Directrice de recherche, CNRS	Rapporteur
M. Jean-Philippe Vert, Maître de recherche, ENSMP	Rapporteur
M. Jean-Michel Jolion, Professeur d'université, Université de Lyon	Examineur
M. Jean-Luc Starck, Directeur de recherche, CEA Saclay	Examineur
M. Gabriel Peyré, Chargé de recherche, CNRS	Examineur
M. Luc Brun, Professeur d'université, ENSICAEN	Directeur de Thèse
M. Alain Buisson, Professeur d'université, Université de Caen	Co-directeur de Thèse
M. Jalal Fadili, Maître de conférences, ENSICAEN	Co-directeur de Thèse

Fais que je sois modéré en tout, mais insatiable dans mon amour de la science. Éloigne de moi l'idée que je peux tout. Donne-moi la force, la volonté et l'occasion d'élargir de plus en plus mes connaissances. Je peux aujourd'hui découvrir dans mon savoir des choses que je ne soupçonnais pas hier, car l'Art est grand mais l'esprit de l'homme pénètre toujours plus avant.

Extrait de *la prière médicale* attribuée à
Maïmonide (Traduction de Soulier [1985]).

Remerciements

Voici venu le moment des remerciements, j'espère n'oublier personne. Premièrement je tiens à remercier ma famille qui m'a toujours soutenue dans les études et cru en moi. Tout particulièrement, merci à mon père et ma mère pour leur soutien et leur présence et ce, même s'ils ne comprennent pas la totalité de mon travail.

Merci à Laure Blanc-Féraud et à Jean-Philippe Vert d'avoir accepté d'être les rapporteurs de la thèse. Merci à Jean-Michel Jolion, d'avoir accepté d'être le président du jury ainsi qu'examineur. Merci à Gabriel Peyré et Jean-Luc Starck d'avoir accepté d'être examinateurs.

Ensuite, un grand merci à mes encadrants, Luc Brun, Jalal Fadili et Alain Buisson, pour les conseils, leurs ouvertures. Je sens que certaines discussions vont, moi aussi, me manquer. Un grand merci aux autres doctorants (nouveaux et anciens) avec qui les rapports ont été toujours dans la bonne humeur (ils se reconnaîtront) : Guillaume, Haz-Edine, Pierre, Romain, Vinh, François, Jean-Hugues, Baptiste. Je remercie aussi les autres membres du laboratoire, dont Marinette Revenu pour m'avoir accueilli dans l'équipe image au départ. Régis Clouard et Daniel Carré pour m'avoir appris beaucoup en informatique. David Tschumperlé pour avoir écrit CImg qui a été le socle de toute ma deuxième partie de thèse. Sébastien Fourey pour toujours avoir été libre lorsque j'avais une question en géométrie discrète et aussi pour m'avoir orienté lors de divers problèmes relatifs aux squelettes. Enfin, merci à tous ceux dont le nom n'apparaît pas ici, mais qui mon aider directement et indirectement pour arriver à l'aboutissement que représente cette thèse.

Table des matières

Introduction générale	1
Notations et sigles	7
Partie I Optimisation convexe et représentations parcimonieuses pour la restauration sous contraintes	11
1 Problématique et état de l'art	13
1.1 Modèle général de dégradation	14
1.2 Algorithmes de déconvolution	15
1.2.1 Paradigme bayésien	15
1.2.2 Estimation bayésienne	16
1.2.3 Attache aux données gaussienne	17
1.2.4 Attache aux données poissonnienne	23
1.2.5 Déconvolution aveugle	27
1.3 Conclusion	27
2 Notre cadre	29
2.1 Elements d'analyse convexe	29
2.1.1 Définitions	29
2.1.2 Opérateur et calcul proximaux	34
2.2 Décomposition d'opérateurs maximaux monotones	37
2.2.1 Les origines de la décomposition	38
2.2.2 Schéma de décomposition explicite-implicite	38
2.2.3 Schéma de décomposition de Douglas-Rachford	42
2.2.4 L'opérateur proximal de $f \circ \mathbf{A}$	43
2.2.5 L'opérateur proximal de $f_1 + f_2 \circ \mathbf{A}$	45
2.3 Formulation du problème d'optimisation	48

2.4	Caractérisation	48
2.5	Premier type d'algorithme	48
2.5.1	Itération extérieure	50
2.5.2	Itération intérieure	50
2.5.3	Robustesse aux erreurs	51
2.6	Second type d'algorithme	51
2.7	Conclusion	53
3	Microscopie confocale de fluorescence	55
3.1	Principe	55
3.1.1	Historique	55
3.1.2	Principe de fonctionnement	56
3.1.3	Modes d'utilisation du microscope confocal	57
3.2	Sources de dégradations	58
3.2.1	Système optique	58
3.2.2	Bilan des bruits	64
3.3	Conclusion : modèle de formation de l'image	65
4	Application à la déconvolution sous bruit de Poisson	67
4.1	Représentations parcimonieuses	68
4.1.1	Définitions	68
4.1.2	Pourquoi la parcimonie?	69
4.1.3	Redondance et diversité morphologique	69
4.2	Formulation du problème d'optimisation	71
4.2.1	Avec stabilisation	72
4.2.2	Sans stabilisation	75
4.3	Premier type d'algorithme : cas stabilisé	76
4.3.1	Algorithme	76
4.3.2	Complexité temporelle	78
4.3.3	A priori analyse	78
4.4	Second type d'algorithme : cas non stabilisé	79
4.4.1	Algorithme	79
4.4.2	Complexité temporelle	82
4.4.3	A priori analyse	82
4.5	Choix du paramètre de régularisation	84
4.6	Expérimentations	87
4.6.1	Protocole expérimental	87

4.6.2	Discussion des résultats	88
4.7	Conclusion	95
Partie II Noyaux sur graphes pour la classification de formes		103
5	Problématique et état de l'art	105
5.1	Codages des formes	106
5.1.1	Contour d'une forme	106
5.1.2	Moments géométriques d'une forme	106
5.1.3	Codages par graphes	107
5.2	Définitions relatives aux graphes	108
5.3	Graphes caractérisant une forme	110
5.3.1	Définition d'un squelette	110
5.3.2	Codage d'un squelette par graphes	112
5.4	Similarité entre graphes	115
5.4.1	Méthodes spectrales	116
5.4.2	Appariement de graphes	117
5.4.3	Distance d'édition	118
5.4.4	Clique maximale dans un graphe d'association	119
5.5	Noyaux sur graphes	121
5.5.1	Noyau sur les chemins	122
5.5.2	Marches aléatoires	123
5.5.3	Motifs engendrés par des arbres	124
5.5.4	Sacs de chemins	125
5.5.5	Distance d'édition	127
5.5.6	Estimation des paramètres des noyaux	128
5.6	Conclusion	128
6	Sacs de chemins avec édition	131
6.1	Extraction du graphe	131
6.1.1	Structure du graphe	132
6.1.2	Attributs	132
6.2	Création d'un sac de chemins	136
6.2.1	Pertinence d'une arête	139
6.2.2	Filtrage selon la pertinence	139
6.2.3	Sac de chemins couvrant un graphe	140
6.3	Noyaux sur sacs de chemins	143

6.3.1	Noyau moyenne pondérée	144
6.3.2	Noyaux et détection de nouveauté	145
6.4	Noyau hiérarchique de comparaison de chemins	148
6.4.1	Observations	148
6.4.2	Opérations d'édition	149
6.4.3	Comparaison hiérarchique des chemins	152
6.5	Prise en compte des trous	153
6.5.1	Noyau sur faces	154
6.5.2	Noyau sur ensembles de faces	155
6.5.3	Combinaisons de noyaux	155
6.6	Expérimentations	156
6.6.1	Protocole expérimental	157
6.6.2	Discussion des résultats	159
6.7	Conclusion et perspectives	165
 Partie III Détection automatique des épines dendritiques		171
7	État de l'art sur l'analyse des épines dendritiques	173
7.1	Les épines dendritiques	173
7.1.1	Historique et présentation	173
7.1.2	Problématiques	175
7.2	Extraction automatique des épines	175
7.2.1	Restauration des images	176
7.2.2	Segmentation	178
7.2.3	Extraction des épines	180
7.2.4	Catégorisation des épines	180
7.2.5	Mesures	182
7.3	Conclusion	182
8	Détection et analyse des épines	185
8.1	Restauration	185
8.2	Segmentation	187
8.3	Extraction des épines	187
8.4	Classification des épines	191
8.5	Mesures biologiques sur les épines	191
8.6	Conclusion	192

Conclusion générale	195
Production scientifique	199
Références bibliographiques	225
Index	227

Introduction générale

*Je paye pour avoir donné la certitude à l'homme
Qu'il fait le monde à son image
Je paye infiniment pour l'homme à mon image
Pour l'homme de mon souffle et de mes bras rompus
Pour l'homme du dedans de mon œil
L'homme enfantant son destin selon sa pensée
Modelant le monde à sa présence
Et de son désir accompli faisant le marchepied d'un désir
nouveau
Je paye pour ce qui fait l'homme sans limite
Et c'est pourquoi ce péage n'a pas de raison jamais d'en finir
Louis Aragon, *Prométhée, Les Poètes**

La recherche sur la compréhension du vivant cherche à répondre à certaines questions fondamentales sur la condition humaine. Par l'étude des autres organismes vivants, l'homme cherche à mieux comprendre son propre corps. Notre époque est plus que jamais dominée par les thèmes impliquant le corps comme la maladie ou la génétique. Ainsi, de nombreuses disciplines scientifiques ont relevé le défi de créer un maximum d'outils pour en étudier le fonctionnement le plus profondément possible. La nature interdisciplinaire de cette recherche implique une grande complexité dans les méthodes mises en œuvre ; les biologistes et les médecins formant le bout de la chaîne par l'exploitation des outils. Durant ces dernières années, cette recherche destinée à l'exploration du vivant n'a pas cessé de s'accélérer. Parmi les avancées, les systèmes d'acquisitions jouent un rôle central puisqu'ils permettent l'observation de l'infiniment petit ou grand avec une précision allant grandissante. En outre, l'exploitation des acquisitions est cruciale pour la compréhension de notre environnement. Cependant, malgré leurs qualités, les systèmes d'acquisitions restent contraints par les systèmes physiques impliqués. Par conséquent, les données acquises doivent être restaurées pour pallier à ces limites et offrir l'ensemble de l'information disponible.

Plus spécifiquement, la recherche sur le fonctionnement du cerveau s'inscrit parfaitement dans les thématiques de notre société actuelle. Cet organe formidablement complexe est impliqué à la fois dans le défi de l'allongement de l'espérance de vie et celui des maladies neurodégénératives (Alzheimer, Creutzfeldt-Jakob, Parkinson, etc) toujours plus nombreuses. Au sein du groupement Cyceron à Caen, une équipe de chercheurs s'intéresse plus particulièrement

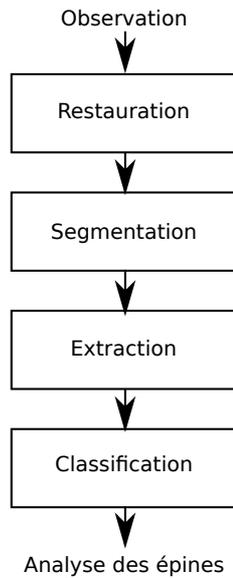


FIG. 0.1: La chaîne de traitements pour les observations d'épines dendritiques.

aux neurones et aux maladies neurodégénératives. Un neurone est essentiellement constitué en trois parties : un corps, le soma, un axe principal, l'axone formant la sortie du neurone et un grand nombre d'axes d'entrées, les dendrites. Ces dernières présentent des excroissances qui intéressent tout particulièrement les chercheurs : les épines dendritiques. Celles-ci possèdent un rôle important dans la communication inter-neuronale et leur morphologie est une caractéristique importante de leur performance en tant que transmetteur. La morphologie des épines change souvent au cours du temps selon l'environnement et l'activité neuronale. Ainsi les épines dendritiques sont fortement liées à certaines maladies neurodégénératives et leur évolution peut être révélatrice d'une pathologie.

Objectifs de la thèse

L'observation de ces petites structures est effectuée par l'intermédiaire d'un microscope confocal de fluorescence. Ce système d'acquisition est connu pour donner des images précises et de qualités. Cependant, elles sont entachées de bruit provenant de la physique liée aux capteurs et faiblement déformées par le système optique. De plus, par sa plasticité, l'épine dendritique est une structure complexe, difficile à décrire par un modèle. Avec la grande quantité de données fournies par une acquisition, l'exploitation manuelle des données est généralement laborieuse. Ainsi se pose la problématique de la création d'une méthode d'analyse automatique et approfondie des épines dendritiques.

Pour répondre à cette question, Koh et al. [2002] formulent les premiers la chaîne de traitements présentée par la figure 0.1. Celle-ci définit les étapes nécessaires pour analyser les épines

dendritiques à partir d'une observation :

Restauration A cause des dégradations présentes dans les images acquises, une étape de déconvolution est nécessaire pour estimer l'image originale à partir de l'observation. Elle tente de corriger le mieux possible les erreurs engendrées par les dégradations.

Segmentation Ensuite, afin de manipuler la structure dendritique comme un objet, nous devons la détacher de l'image, d'où une segmentation.

Extraction Une fois l'axe dendritique repéré, nous pouvons identifier les excroissances connectées ou non à l'axe répondant à des critères *simples* pour être des épines dendritiques.

Classification Généralement, deux questions sont posées pour chaque épine détectée : « Est-ce bien une épine ? » et si oui, « Quelle est son type ? ». Pour être efficace, cette étape demande des informations a priori de la part des spécialistes en biologie. Selon la modélisation retenue et les données disponibles, un classifieur adapté est produit puis utilisé.

Analyse Les épines étant classées, des informations spécifiques peuvent être calculées, par exemple la longueur, la largeur et le volume des épines et selon leurs types, la longueur et la largeur du cou et de la tête.

Bien qu'indépendante, la qualité de chaque étape dépend évidemment des précédentes. Il est donc recommandé de soigner les algorithmes mis en œuvre. En conséquence, la performance de l'algorithme de restauration des images se répercute sur tout le reste de la chaîne. De même, la classification doit être capable d'utiliser toutes les informations disponibles sur les épines dendritiques afin de conduire à une analyse correcte.

Nous nous focalisons dans ce travail de thèse sur ces deux étapes, car ce sont les plus importantes : l'une par ses répercussions sur l'ensemble de la chaîne, l'autre car elle doit filtrer les éléments aberrants avant l'analyse. En effet, d'un côté une bonne restauration est essentielle pour recouvrer les structures dégradées et ainsi conduire à des résultats plus précis. De plus la forte dégradation des images induit une perte d'information empêchant une utilisation immédiate. De l'autre côté, la classification, à cause de la grande plasticité des épines, demande des outils capables d'exploiter leur morphologie. Par conséquent, nous avons choisi de diviser cette thèse en trois parties : la première partie concerne la restauration d'images, la deuxième, la classification de formes et la troisième, l'insertion de nos contributions dans la chaîne de traitements (figure 0.1).

Plan de la thèse

Comme indiqué précédemment, la restauration sous contraintes d'images dégradées par un bruit de Poisson fait l'objet de la première partie. Nous commençons par présenter un aperçu des algorithmes exploités par la déconvolution sous bruit gaussien et poissonnien, pour restaurer les images issues d'un microscope confocal (chapitre 1). Avant de détailler la conception

d'algorithme de déconvolution, nous proposons ensuite un cadre général (chapitre 2) pour la restauration d'images basé sur la minimisation d'une fonction d'énergie composée de trois termes : 1) un terme d'attache aux données convexe et continûment différentiable, 2) un terme de régularisation convexe et non-différentiable et 3) un terme reflétant l'appartenance de l'image à un ensemble convexe fermé de contraintes données. Le problème d'optimisation est résolu par des méthodes issues de l'analyse convexe et des schémas itératifs pour la décomposition d'opérateurs maximaux monotones. Ce cadre exploite un opérateur maximal monotone particulier : l'opérateur proximal qui revient à une régularisation de Moreau-Yosida d'une fonction et permet de généraliser des schémas d'optimisation.

Ensuite, nous étudions le fonctionnement d'un microscope confocal afin de modéliser la formation de l'image (chapitre 3). En effet, les images produites par un microscope confocal de fluorescence souffrent d'une double dégradation. La première provient du système optique, généralement modélisé par une convolution avec une fonction d'étalement du point (PSF) dont nous présentons deux modélisations. L'autre est due à la nature poissonnienne des statistiques impliquées par le processus de comptage au niveau du détecteur. Enfin, nous montrons que le modèle de formation de l'image le plus simple induit une déconvolution sous bruit de Poisson pour la restauration.

Afin de concevoir de tels algorithmes, le chapitre 4 exploite le cadre du chapitre 2. Pour prendre en compte la nature statistique du bruit, deux approches pour construire le terme d'attache aux données sont proposées : soit nous appliquons la transformée stabilisatrice de la variance d'Anscombe pour retourner dans un cadre gaussien, soit nous utilisons le log de vraisemblance du bruit de Poisson. La première approche conduit à un terme non-linéaire, mais celui-ci est de gradient Lipschitz contrairement au log de vraisemblance du bruit de Poisson. Ensuite, le terme de régularisation est construit afin de promouvoir la parcimonie sur les coefficients représentant l'image. Ici aussi, nous proposons deux façons de formuler le problème : dans le domaine des coefficients des transformées (a priori dit synthèse) et dans le domaine des pixels (a priori dit analyse). Le troisième terme est une contrainte reflétant la positivité de l'image à estimer. Selon les approches, nous utilisons l'un des algorithmes d'optimisations proposés pour résoudre le problème sous-jacent. Ensuite, pour estimer le paramètre de régularisation, nous proposons un critère de sélection de modèle basé sur une validation croisée généralisée. Enfin, des résultats de comparaisons avec l'état de l'art sont présentés.

Dans la deuxième partie, nous considérons les espaces à noyaux reproduisants appliqués à la classification de forme. Nous commençons par présenter dans le chapitre 5 plusieurs méthodes de représentations et de comparaisons de formes. Ensuite nous approfondissons celles basées sur les graphes et les différentes approches pour les comparer. Enfin, nous présentons les noyaux définis positifs exploités dans le cadre de la classification. Les noyaux sur graphes permettent d'effectuer des produits scalaires dans un espace de Hilbert particulier où les données sont

plus facilement séparables. Dans le cas des graphes, cette projection dans un espace de Hilbert munit l'espace des graphes d'une métrique ou d'une pseudo-métrique permettant d'appliquer aux graphes de nombreux algorithmes de classification.

Le chapitre 6 introduit une nouvelle représentation des formes proche des graphes de chocs. En s'inspirant des travaux de Suard et al. [2006], nous présentons des extensions des noyaux sur graphes basées sur des sacs de chemins. Ce chapitre développe chacune des étapes nécessaires à la conception de tels noyaux : création des sacs de chemins, noyau sur sacs et noyau sur chemins. Nous proposons plusieurs méthodes pour construire un sac de chemins en exploitant une mesure de pertinence des chemins. Puis, plusieurs noyaux sur sacs de chemins sont introduits, chacun induisant un modèle de données particulier. Cependant, tous ces noyaux reposent sur un noyau sur chemins, or dans le cadre de la reconnaissance des formes, le graphe est obtenu en codant le squelette de la forme. Celui-ci résulte d'un processus homotopique et correspond à un graphe planaire dans le cas d'une forme 2D. Ce graphe n'est pas toujours un graphe simple et souffre du bruit structurel induit par la sensibilité du processus de squelettisation aux déformations du contour. Nous introduisons donc un noyau sur chemins incluant une notion d'édition afin de le rendre robuste au bruit. Cependant, les formes peuvent présenter un ou plusieurs trous caractéristiques (par exemple, les ciseaux) codés dans un graphe planaire par des faces. Ainsi nous proposons un noyau sur faces pour prendre en compte cette information. Ce noyau est ensuite combiné à un noyau sur sacs de chemins pour le compléter. Enfin, nos approches sont comparées à l'état de l'art au niveau de la robustesse au bruit et des performances de classification.

La dernière partie combine les outils précédents dans le cadre de la reconnaissance et analyse des épines dendritiques. Le chapitre 7 présente les différentes approches utilisées au sein de la chaîne de traitements (figure 0.1) dédiée aux épines. Puis, dans le chapitre 8 nous intégrons dans cette chaîne les outils des chapitres précédents 4 et 6. Les algorithmes de restaurations nous permettent de recouvrir la structure et de préserver les épines. Cette structure tubulaire peut être représentée par un graphe. Nous pouvons utiliser le cadre et les noyaux sur graphes en adaptant la représentation à la 3D tubulaire. Avec les noyaux, les épines peuvent être classées selon leur morphologie à la place d'un vecteur d'attributs. Nous proposons plusieurs approches pour effectuer des mesures sur les épines selon leur type en tenant compte de la perte de données le long de l'axe des profondeurs. En conclusion, nous présentons les différentes perspectives offertes par cette thèse.

Contributions

Le chapitre 2 propose un cadre complet pour la conception d'algorithme d'optimisation convexe. Nos contributions reposent sur la combinaison de différents schémas de minimisa-

tion pour résoudre des problèmes formés par une somme de trois fonctions semi-continues inférieurement. Deux approches sont exposées, toujours basées sur les mêmes schémas de minimisation, pour gérer la composition avec un opérateur linéaire borné. Pour chaque formulation de problème, nous avons établi les conditions d'existence et d'unicité d'une solution ainsi que la convergence des algorithmes.

Ce cadre est exploité dans le chapitre 4 pour construire quatre algorithmes de déconvolution reposant sur la notion de parcimonie et le concept de diversité morphologique. Ces quatre algorithmes sont les déclinaisons de deux possibilités : la stabilisation ou non du bruit et le choix d'un a priori synthèse ou analyse. Selon les combinaisons, nous utilisons différents résultats du chapitre 2 pour résoudre le problème d'optimisation induit.

Dans le chapitre 6, nous étendons les noyaux sur graphes par sacs de chemins introduits par Suard et al. [2006]. D'abord, nous proposons plusieurs approches pour construire les sacs de chemins, avec un faible pourcentage de chemins pertinents ou en effectuant une couverture contrainte de graphe. Puis nous proposons de nouveaux noyaux pour comparer les sacs en essayant de modéliser la propagation des données dans l'espace engendré par le noyau sur chemins. Ensuite, un mécanisme d'édition de chemins est présenté pour rendre les comparaisons robuste au bruit structurel et plus souple. Enfin, nous décrivons un noyau sur faces afin de prendre en compte les trous caractéristiques de certaines formes (par exemple les ciseaux). Pour chaque noyau, nous avons établi sa défini positivité et sa complexité temporelle.

Les contributions précédentes sont mises à profit dans le chapitre 8 pour former une chaîne de traitements. Nous décrivons les méthodes utilisées dans chacune des étapes adaptées à notre représentation. Pour cela, nous étendons la représentation des graphes introduite chapitre 6 à la 3D tubulaire. Cette dernière, par l'intermédiaire des noyaux sur graphes, offre toutes les possibilités des machines à noyaux pour effectuer de la fouille de données sur des ensembles d'épines dendritiques.

Notations et sigles

Notations

Fonctions et signaux

$f(t)$	Fonction continue dans le temps, $t \in \mathbb{R}$.
$f(\mathbf{t})$ ou $f(t_1, \dots, t_d)$	Fonction d -D continue dans le temps, $\mathbf{t} \in \mathbb{R}^d$.
$f[k]$	Signal discret dans le temps, $k \in \mathbb{Z}$ ou k -ième entrée d'un vecteur de dimension finie.
\hat{f}	Transformée de Fourier de f .
f^*	Conjuguée complexe d'un signal ou d'une fonction.
$lhs = \mathcal{O}(rhs)$	lhs est de l'ordre de rhs ; il existe une constante $C > 0$ telle que $lhs \leq Crhs$.
$lhs \sim rhs$	lhs est équivalent à rhs ; $lhs = \mathcal{O}(rhs)$ et $rhs = \mathcal{O}(lhs)$.
\mathcal{H}	Ensemble convexe de dimension finie n .
$\Gamma_0(\mathcal{H})$	Classe des fonctions convexes, propres et semi-continues inférieurement de \mathcal{H} vers $\mathbb{R} \cup \{+\infty\}$.
$2^{\mathcal{H}}$	Ensemble des sous-parties de \mathcal{H} .

Opérateurs sur les signaux ou fonctions

*	Convolution continue.
\dagger	Inf-convolution (régularisation de Moreau-Yosida).
\otimes	Convolution discrète circulaire.
\odot	Multiplication terme à terme de deux vecteurs (produit de Hadamard).

Matrices, opérateurs linéaires et normes

Symboles en majuscules et en gras	Matrices ou opérateurs linéaires, e.g. \mathbf{M} .
\cdot^T	Transposée d'un vecteur ou d'une matrice.
$\mathbb{1}$	Le vecteur composé uniquement de 1.
\mathbf{M}^*	Adjoint de \mathbf{M} (en tant que matrice ou opérateur).
$\mathbf{M}[i, j]$	Entrée de la ligne i et de la colonne j de la matrice \mathbf{M} .
$\text{diag}(\mathbf{M})$	La matrice diagonale avec les mêmes éléments diagonaux que la matrice \mathbf{M} .
$\text{Im}(\mathbf{M})$	Image de \mathbf{M} .
$\ker \mathbf{M}$	Noyau de \mathbf{M} .
$\text{trace}(\mathbf{M})$	Trace de la matrice \mathbf{M} .
\mathbf{M}^+	Pseudo-inverse de \mathbf{M} .
$\langle \cdot, \cdot \rangle$	Produit scalaire (dans un espace pré-hilbertien).
$\ \cdot\ $	Norme associée.
$\ \cdot\ _p$	$p \geq 1$, ℓ_p -norme d'un signal.
$\ \cdot\ _0$	ℓ_0 quasi-norme d'un signal ; nombre d'éléments non-nuls.
$\ \cdot\ _{\text{TV}}$	(Semi)-norme discrète de la variation totale.
$\bar{\nabla}$	Gradient discret d'une image.
$\overline{\text{div}}$	Opérateur discret de divergence (adjoint de $\bar{\nabla}$).
$\ \cdot\ $	Norme spectrale d'un opérateur linéaire.
$\ \cdot\ _F$	Norme de Frobenius d'une matrice.
\mathbf{ST}_γ	Opérateur de seuillage doux de seuil γ .
prox_f	Opérateur proximal d'une fonction f dans $\Gamma_0(\mathcal{H})$.

Variables aléatoires et vecteurs

$\varepsilon \sim \mathcal{N}(\mu, \sigma^2)$	ε est additif, blanc et gaussien avec une moyenne μ et une variance σ^2 .
$\varepsilon \sim \mathcal{P}(\lambda)$	ε est distribué selon une loi de Poisson avec une intensité (moyenne) λ .
$\mathbb{E}[\cdot]$	L'opérateur d'espérance.
$\text{Var}[\cdot]$	L'opérateur de variance.

Suites et schémas itératifs

N_{ext}	Nombre d'itérations pour le schéma extérieur.
------------------	---

N_{int}

Nombre d'itérations pour le schéma intérieur.

Sigles

1D, 2D, 3D

uni-, bi-, tri-dimensionnel.

ACP

Analyse en composantes principales.

AIC

Critère d'information d'Akaike (*Akaike information criterion*).

BIC

Critère d'information bayésien (*Bayesian information criterion*).

BPDN

Basic Pursuit Denoising.

DR

Schéma de Douglas-Rachford.

DSP

Densité spectrale de puissance.

EDP

Equation aux dérivées partielles.

FB

Schéma explicite-implicite (Forward-Backward) à un pas.

GCV

Validation croisée généralisée (*Generalized Cross Validation*).

GFP

Protéine à fluorescence verte (*Green fluorescent protein*).

MAE

Erreur absolue moyenne (*Mean Absolute Error*).

MAP

Maximum a posteriori.

MMSE

Espérance conditionnelle a posteriori (*Minimum Mean Square Error*).

MSE

Erreur quadratique moyenne (*Mean Square Error*).

pdf

Densité de probabilité.

PMT

Photomultiplicateur.

PSF

Fonction d'étalement du point (*Point Spread Function*).

RL

Algorithme de Richardson-Lucy.

SKIZ

Squelette par zones d'influences.

SVDD

Descripteur de données à vecteurs de support (*Support Vector Data Descriptor*).

SVM

Séparateur à vaste marge (*Support Vector Machine*).

TCL

Théorème central limite.

VST

Transformée stabilisatrice de la variance (*Variance Stabilizing Transform*).

Première partie

Optimisation convexe et
représentations parcimonieuses pour
la restauration sous contraintes

1 Problématique et état de l'art

*Mais pour moi, que tes pensées sont difficiles,
ô Dieu, que la somme en est imposante !
Je les compte, il en est plus que sable ;
ai-je fini, je suis encore avec toi.*

Psaume 139(138) 17-18

La déconvolution est un problème présent, depuis longtemps, au sein de nombreuses disciplines liées au traitement du signal et des images (l'imagerie biomédicale [Sarder et Nehorai, 2006; Pawley, 2005] ou l'astronomie [Starck et Murtagh, 2006] par exemples). La recherche dans la déconvolution des images astronomiques a récemment connu un regain d'intérêt, partiellement déclenché par la présence d'aberrations optiques dans les premières acquisitions produites par le télescope spatial Hubble. En imagerie biomédicale, les chercheurs se reposent de plus en plus sur la déconvolution pour améliorer la qualité des images acquises avec un microscope confocal [Pawley, 2005]. Celle-ci devient alors cruciale pour exploiter pleinement les images acquises et en extraire le contenu scientifique.

Une imposante littérature est disponible sur la problématique de la déconvolution [Andrews et Hunt, 1977; Stark, 1987; Jansson, 1997]. En présence du bruit de Poisson, plusieurs méthodes ont été proposées dont le filtre inverse de Tikhonov-Miller et l'algorithme de Richardson-Lucy (RL) [Lucy, 1974; Lagendijk et Biemond, 1991; van der Voort et Strasters, 1995; Sarder et Nehorai, 2006; Starck et Murtagh, 2006]. Ce dernier a été fortement utilisé dans un grand nombre d'applications, car il est adapté au bruit de Poisson. Cependant, il amplifie le bruit après quelques itérations, mais l'introduction d'une régularisation permet de contrôler cette amplification. Par exemple, la variation totale a été utilisée comme terme de régularisation par Dey et al. [2004]. En astronomie, plusieurs auteurs ont préconisé l'utilisation des ondelettes pour former un terme de régularisation [Starck et Murtagh, 1994; Starck et al., 1995; Jammal et Bijaoui, 2004]. Dans le contexte de l'imagerie biomédicale, une régularisation basée sur les ondelettes a aussi été exploitée par Boutet de Monvel et al. [2001, 2003]. De leur côté, Vonesch et Unser [2007] combinent l'itération de Landweber seuillée introduite par Daubechies et al. [2004] avec un terme de promotion de la parcimonie dans le domaine des ondelettes. Cette dernière approche suppose implicitement que le bruit contaminant les images suit une loi gaussienne, ce qui bien évidemment n'a de sens que si le nombre de photons reçus est élevé (selon le théorème

central limite). Et même si le régime du TCL est vérifié, le bruit reste dépendant du signal (cf. 1.2.3).

Ce chapitre présente brièvement le modèle de dégradation généralement adopté pour le système d'acquisition du microscope confocal. Puis nous proposons une revue non exhaustive des différentes approches déployées pour la conception d'algorithmes de déconvolution. Elle se décline en deux parties selon les deux modèles de dégradations dominants dans la littérature : le modèle gaussien et le modèle poissonnien.

1.1 Modèle général de dégradation

Le microscope confocal présente deux principales sources de bruit : le système optique et le détecteur (voir le chapitre 3 pour plus de détails). Le système optique est modélisé par la fonction d'étalement du point (PSF) notée h . Ce modèle peut être obtenu soit en connaissant la formule du modèle ou une approximation [Gibson et Lanni, 1991; Booth et Wilson, 2001; Zhang et al., 2007a], ou enfin en mesurant expérimentalement les distorsions induites par le système [Pawley, 2005], soit en l'estimant conjointement avec l'image originale (cas de la déconvolution aveugle ou myope). Les principales sources de bruit proviennent de la nature physique des phénomènes observés. Ainsi le comptage des photons au niveau du capteur est soumis à un processus poissonnien, conduisant à une importante dégradation des données lorsque le nombre de photons est faible.

L'image observée est formée par un ensemble de n pixels. Ceux-ci représentent une collection discrète de comptages bornés $y = (y[i])_{1 \leq i \leq n}$, i.e. $y \in \ell_\infty(\Omega)$, sur un domaine discret $\Omega = \{0, \dots, \sqrt{n} - 1\}^2$. Chaque comptage est la réalisation d'une variable aléatoire de Poisson indépendante avec une moyenne de $(h \otimes x)[i]$, où \otimes représente l'opérateur de convolution circulaire et h la PSF. Le modèle de dégradation d'une image originale x est donc :

$$\forall i, 1 \leq i \leq n, \quad y[i] \sim \mathcal{P}((h \otimes x)[i]). \quad (1.1)$$

Le problème de la déconvolution consiste alors à retrouver l'image x à partir de l'image de comptage observée y . Pour des raisons de simplification de notation, l'équation de dégradation est réécrite :

$$y \sim \mathcal{P}(\mathbf{H}x), \quad (1.2)$$

où \mathbf{H} est l'opérateur de convolution circulaire avec h . Nous notons par \mathbf{H}^* l'opérateur de convolution avec le symétrique de la PSF h .

1.2 Algorithmes de déconvolution

Cette section se découpe en deux sous-parties suivant les deux approches les plus courantes pour la déconvolution sous bruit de Poisson. La première approche approxime le bruit de Poisson par un bruit blanc additif et gaussien, permettant d'utiliser les nombreuses méthodes disponibles dans le cas gaussien. La deuxième approche est plus directe et cherche à formuler le problème de déconvolution en modélisant fidèlement la statistique du bruit de Poisson. Pour chacune des deux approches, nous proposons un aperçu général des principaux algorithmes utilisés dans le cadre de la déconvolution en biologie et/ou en astronomie.

1.2.1 Paradigme bayésien

Soient X et Y deux variables aléatoires, X représente l'image réelle et Y l'observation à travers d'un système d'acquisition. L'image des observations Y est considérée comme une fonction déterministe et/ou aléatoire de l'image des X . La forme explicite de cette fonction dépend essentiellement du système d'acquisition. Nous notons \mathcal{X} l'espace des images à estimer et \mathcal{Y} celui des observations. Estimer X en ayant observé Y est souvent difficile, car généralement la solution n'est pas unique. De plus, elle n'est pas stable en présence de bruit (deux conditions d'Hadamard sont violées). Il faut par conséquent imposer des contraintes supplémentaires pour réduire le nombre de candidats à la solution dans l'espace \mathcal{X} .

Le paradigme bayésien offre un cadre théorique flexible pour imposer des contraintes sur X au travers d'une loi de probabilité sur l'espace \mathcal{X} . Cette loi est appelée *loi a priori*. Le second ingrédient est la loi de l'observation Y sachant X . Celle-ci est essentiellement obtenue par la loi du bruit contaminant les mesures dans le modèle de formation de l'image. A partir de la loi a priori et de la conditionnelle a priori, nous pouvons définir la densité de probabilité (pdf) jointe définie sur l'espace produit $\mathcal{X} \times \mathcal{Y}$ par la règle de Bayes :

$$p_{X,Y}(x, y) = p_{Y|X}p_X(x) . \quad (1.3)$$

Avec les réalisations y , il est aisé de prouver que la pdf *conditionnelle a posteriori* est :

$$p_{X|Y}(x|y) = \frac{p_{Y|X}(y|x)p_X(x)}{\int_{\mathcal{X}} p_{Y|X}(y|x)p_X(x)dx} . \quad (1.4)$$

Nous avons supposé tacitement que le dénominateur ne s'annule jamais.

Le choix de la loi a priori ne peut être traité en général. Souvent, il convient de dire qu'un a priori quelconque de pdf $p_X(x)$ aura toujours une forme gibbsienne équivalente pour une fonction d'énergie $U(x) = -\log p_X(x)$ donnée (théorème de Hammersley-Clifford).

Nous définissons alors le score de vraisemblance conditionnelle a posteriori à partir de (1.4) :

$$U(x|y) = -\log p_{X|Y}(x|y) = \underbrace{-\log p_{Y|X}(y|x)}_{\text{terme de fidélité } -\ell(y|x)} + \underbrace{U(x)}_{\text{terme de régularité}} - c(y). \quad (1.5)$$

Le premier terme reflète l'attache aux données (fidélité aux observations), et peut être vu comme le score de vraisemblance de l'observation Y sachant le paramètre pertinent X . Dans beaucoup de situations en traitement d'images, ce terme est étroitement lié au modèle du bruit. Le second terme de (1.5) traduit les contraintes (de régularité par abus de langage parfois) imposées sur la solution à produire. Cette formulation regroupe un grand nombre de situations pratiques et unifie leur traitement dans le cadre statistique bayésien. Notons par ailleurs que cette écriture partage beaucoup de similitudes avec les modèles variationnels.

1.2.2 Estimation bayésienne

Nous avons à notre disposition une image observée Y , de laquelle nous voulons déduire l'image non-observée X . Nous avons vu que l'approche bayésienne permet d'ajuster la solution \tilde{x} aux données tout en obéissant à des contraintes (de « qualité »). Il existe alors différentes manières de choisir l'estimée \tilde{x} de x tout en assurant ce compromis entre fidélité et régularité. C'est ce que nous allons exposer dans un cadre général en faisant appel à la théorie de l'estimation bayésienne.

En estimation bayésienne, un estimateur est défini comme une application :

$$\begin{aligned} \mathcal{D} : \mathcal{Y} &\rightarrow \mathcal{X} \\ y &\rightarrow \tilde{x} = \mathcal{D}(y) \end{aligned}$$

pour laquelle nous espérons que \tilde{x} soit la plus proche possible de x . Le terme proche suppose que nous disposons d'une métrique pour quantifier l'écart entre x et son estimée. Définissons la fonction coût $L(x, \tilde{x})$ positive, vérifiant $L(x, \tilde{x}) = 0$ (une pseudo-distance puisque l'inégalité triangulaire n'est pas forcément vérifiée). Le risque bayésien est le risque moyen relativement à la probabilité jointe de X et \tilde{x} :

$$\begin{aligned} R(x, \mathcal{D}(y)) &= \mathbb{E}_{X, \tilde{X}} [L(x, \mathcal{D}(y))] = \int_{\mathcal{X} \times \mathcal{Y}} L(x, \mathcal{D}(y)) p_{X,Y}(x, y) dx dy \\ &= \int_{\mathcal{Y}} \int_{\mathcal{X}} L(x, \mathcal{D}(y)) p_{Y|X}(y|x) p_X(x) dx dy. \end{aligned} \quad (1.6)$$

Un estimateur bayésien est celui minimisant ce risque. Ainsi, chaque estimateur bayésien différent correspond à une fonction de coût spécifique. D'ailleurs les performances d'un estimateur bayésien sont affectées non seulement par la qualité de l'a priori, mais aussi par le choix de la fonction de coût.

1.2.2.1 Maximum a posteriori (MAP)

Soit la fonction de coût par tout ou rien suivante :

$$L(x, \tilde{x}) = \begin{cases} 0 & \text{si } x = \tilde{x} \\ 1 & \text{si } x \neq \tilde{x} \end{cases} . \quad (1.7)$$

Proposition 1.2.1. *L'estimateur bayésien correspondant à la fonction de coût ci-dessus est le maximum a posteriori, i.e. :*

$$\tilde{x}_{\text{MAP}} = \arg \max_{x \in \mathcal{X}} p_{X|Y}(x|y) . \quad (1.8)$$

1.2.2.2 Espérance conditionnelle a posteriori (MMSE)

Considérons maintenant le cas de l'erreur quadratique pour la fonction de coût :

$$L(x, \mathcal{D}(y)) = (x - \mathcal{D}(y))^2 . \quad (1.9)$$

Proposition 1.2.2. *L'estimateur bayésien correspondant à la fonction de coût ci-dessus est l'espérance conditionnelle a posteriori (Minimum Mean Square Error ou MMSE), i.e. :*

$$\tilde{x}_{\text{MMSE}} = \mathbb{E}[X|Y = y] = \frac{\int_{\mathcal{X}} x p_{Y|X}(y|x) p_X(x) dx}{\int_{\mathcal{X}} p_{Y|X}(y|x) p_X(x) dx} . \quad (1.10)$$

L'inférence bayésienne sert souvent de base théorique pour les méthodes de déconvolution qui vont suivre. En effet, de part ses propriétés et sa relative simplicité, l'estimateur du maximum a posteriori est régulièrement choisi lorsqu'un a priori sur la solution est à ajouter au problème.

1.2.3 Attache aux données gaussienne

Pour des raisons de complexité analytique et de temps de calcul, les premiers algorithmes utilisés pour la déconvolution supposent un bruit gaussien. En effet, de nombreuses méthodes existent déjà pour la déconvolution associée à un bruit gaussien. De plus certaines sont linéaires, donc rapides et utilisables dans un environnement de calcul limité. Le modèle de dégradation dans le cas gaussien est le suivant :

$$y = \mathbf{H}x + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2), \quad (1.11)$$

avec σ l'écart-type du bruit.

L'erreur commise par ce modèle de dégradation par rapport au véritable modèle dépend fortement du taux de comptage, c'est à dire du nombre moyen de photons émis par l'objet observé. En effet, soit $y \sim \mathcal{P}(\lambda)$, par le théorème central limite, lorsque la moyenne λ est grande, y tend en distribution vers une variable normale de moyenne λ et de même variance λ . Deux difficultés majeures découlent toutefois : lorsque la moyenne est faible, cette approximation conduit à des erreurs non-négligeables, même si λ est grand, la variance asymptotique est linéaire en fonction du signal, alors que σ dans (1.11) est supposé constant.

1.2.3.1 Maximum de vraisemblance ou filtrage inverse

L'estimateur le plus direct et simple dans le cas gaussien est celui du maximum de vraisemblance :

$$\tilde{x}_{MV} = \arg \min_{x \in \mathcal{X}} \frac{\|y - \mathbf{H}x\|^2}{2\sigma^2} . \quad (1.12)$$

Cet estimateur est aussi connu sous le nom de méthode des moindres carrés. (1.12) peut être résolue analytiquement de deux manières différentes : soit dans le domaine temporel, soit dans le domaine fréquentiel.

Dans le domaine temporel, la solution est directement donnée par inversion matricielle :

$$\tilde{x}_{MV} = (\mathbf{H}^* \mathbf{H})^{-1} \mathbf{H}^* y = \mathbf{H}^+ y , \quad (1.13)$$

pourvu que $\mathbf{H}^* \mathbf{H}$ soit inversible. Dans le domaine de Fourier, la convolution devient une multiplication, et une estimation de l'image originale est obtenue par une division terme à terme :

$$\hat{x} = \left(\begin{array}{c} \hat{y}[i] \\ \hat{h}[i] \end{array} \right)_{1 \leq i \leq n} , \quad (1.14)$$

si $\forall i, \hat{y}[i] \neq 0$. Cette méthode est aussi connue sous le nom de filtrage inverse.

La simplicité de ces deux techniques est contrebalancée par leurs comportements face au bruit. En effet, celui-ci est fortement amplifié sur certaines fréquences où $\hat{h}[i]$ est petit, conduisant souvent à un résultat difficile à exploiter. De façon générale, un problème surgit lorsque l'opérateur \mathbf{H} est mal conditionné. Pour stabiliser partiellement la solution, une heuristique consiste à seuiller ou tronquer les valeurs dans (1.14).

1.2.3.2 Filtre de Wiener

Afin d'obtenir une meilleure restauration, nous posons des a priori sur l'image réelle : nous supposons que x est gaussien, et que x et ε sont mutuellement indépendants. Nous pouvons alors montrer que pour un tel a priori l'estimateur de Wiener coïncide avec l'estimateur bayésien MMSE (et aussi avec le MAP). L'estimateur de Wiener pour des signaux cyclostationnaires est :

$$\hat{x}[i] = \frac{\hat{h}[i] \hat{y}[i]}{|\hat{h}[i]|^2 + \frac{S_\varepsilon}{S_{\tilde{x}}}} , \quad (1.15)$$

où S_ε et $S_{\tilde{x}}$ sont respectivement les densités spectrales de puissance (DSP) de ε et de \tilde{x} .

Le filtre de Wiener est connu par sa rapidité, mais il produit aussi des artefacts tels que les effets de Gibbs. De plus, l'estimateur dépend de la DSP de l'image inconnue \tilde{x} qui est rarement (voire jamais) accessible. Plusieurs alternatives sont disponibles. L'une d'entre elles consiste à estimer localement un rapport signal-sur-bruit en chaque fréquence. L'autre, plus attractive,

consiste à itérer le filtre de Wiener et utiliser l'estimée de l'itération t pour en déduire la DSP en $t + 1$ et donc une nouvelle estimée [Hillery et Chin, 1991].

1.2.3.3 Régularisation de Tikhonov

L'une des régularisations plus connues en traitement d'images est celle de Tikhonov. Elle consiste en une contrainte quadratique reflétant le caractère lisse de l'image à restaurer, le problème d'optimisation devient alors :

$$\tilde{x} \in \arg \min_{x \in \mathbb{R}^n} \|y - \mathbf{H}x\|^2 + \gamma \left\| \Delta^{(s)} x \right\|^2, \quad (1.16)$$

où γ est un paramètre représentant l'équilibre entre la régularisation et l'attache aux données, $\Delta^{(s)}$ est un opérateur différentiel d'ordre s , par exemple $s = 2$ correspond au laplacien. La solution de cette équation est obtenue analytiquement :

$$\tilde{x} = \left(\mathbf{H}^* \mathbf{H} + \gamma \Delta^{(s)*} \Delta^{(s)} \right)^{-1} \mathbf{H}^* y, \quad (1.17)$$

pourvu que le terme entre parenthèses soit inversible. Notons que cette solution peut être calculée efficacement si $\Delta^{(s)}$ et \mathbf{H} sont diagonalisés dans une même base, ce qui est le cas pour le laplacien ($s = 2$) et la convolution, tous deux diagonalisés dans la base de Fourier.

Cette équation de déconvolution conduit à de bons résultats, mais les images restaurées sont souvent trop lisses. En effet, la régularisation de Tikhonov préserve mal les contours. Le paramètre de régularisation γ peut être estimé par des techniques telles que la validation croisée généralisée [Golub et al., 1979].

1.2.3.4 L'algorithme de Jansson-Van Cittert

Les méthodes linéaires sont généralement rapides, mais sont limitées au niveau de l'expressivité. En effet il n'est pas toujours facile de préserver la linéarité en ajoutant une régularisation. De plus, l'estimateur du maximum de vraisemblance (1.12) demande une inversion matricielle, or cette opération est sensible au conditionnement des matrices. Toutefois, ce n'est pas l'inverse de la matrice qui est utile au calcul, mais le produit matriciel entre cette matrice inverse et un vecteur donné.

L'algorithme de Van Cittert est historiquement l'une des premières méthodes itératives de déconvolution contrainte. Il calcule l'estimateur du maximum de vraisemblance (1.12), sans passer par une inversion. Il débute par $x_0 = y$ puis itère :

$$x_{t+1} = x_t + \tau(y - \mathbf{H}x_t), \quad (1.18)$$

où τ est un paramètre de descente dépendant de \mathbf{H} .

Cet algorithme est modifié par Jansson [1970] pour le rendre davantage robuste au bruit en ajoutant des contraintes sur la solution. Par exemple, si l'image d'origine est supposée bornée : $\forall i, 1 \leq i \leq n, a \leq x[i] \leq b$, l'itération devient alors :

$$x_{t+1} = x_t + r(x_t)(y - \mathbf{H}x_t), \quad (1.19)$$

où r est une fonction d'échelle dont la formule est :

$$r(x) = c \left(1 - 2 \frac{\|x - \frac{a+b}{2} \mathbb{1}\|}{a-b} \right).$$

avec c une constante suffisamment petite pour permettre à l'itération de converger vers une solution et $\mathbb{1}$ le vecteur constitué uniquement de 1.

En plus des défauts inhérents à l'estimateur, cet algorithme pose des problèmes de convergence. En effet, il diverge parfois impliquant que le nombre d'itérations peut être interprété comme un paramètre de régularisation.

1.2.3.5 L'algorithme de Landweber

L'algorithme de Landweber [1951] utilise une procédure itérative pour calculer le maximum de vraisemblance. Nous avons fait remarquer que (1.13) et (1.14) reposent sur une inversion, or une telle opération demande des matrices bien conditionnées, ce qui n'est pas toujours le cas. Pour éviter d'effectuer cette opération, Landweber [1951] utilise une descente de gradient pour minimiser (1.12) :

$$x_{t+1} = x_t + \tau \mathbf{H}^*(y - \mathbf{H}x_t), \quad (1.20)$$

où τ est le pas de descente tel que $0 < \tau < 2/\|\mathbf{H}\|^2$. L'itération converge vers un minimum global qui n'est cependant pas unique si $\hat{h}[i]$ s'annule pour i donné [Ciarlet, 1985]. Le nombre d'itérations joue, ici aussi, un rôle important dans cette méthode et peut être interprété comme un paramètre de régularisation, car le bruit a tendance à être amplifié sur certaines fréquences après quelques itérations.

Des contraintes peuvent être ajoutées en utilisant une descente de gradient projeté :

$$x_{t+1} = \mathcal{P}_{\mathcal{C}}(x_t + \tau \mathbf{H}^*(y - \mathbf{H}x_t)), \quad (1.21)$$

où $\mathcal{P}_{\mathcal{C}}$ est le projecteur sur un ensemble de contraintes \mathcal{C} . Les contraintes les plus courantes sont la positivité, la restriction du support ou bien une contrainte de limitation de la bande de fréquences. L'itération converge vers un minimum global (non-nécessairement unique) si \mathcal{C} est un ensemble convexe fermé [Ciarlet, 1985].

1.2.3.6 L'algorithme itératif contraint de Tikhonov-Miller

L'algorithme de Landweber n'intègre pas de régularisation. Celui de Tikhonov-Miller [Lagendijk et Biemond, 1991; van der Voort et Strasters, 1995] utilise le maximum a posteriori en combinant le terme de régularisation de Tikhonov avec une contrainte de positivité et propose donc de résoudre le problème d'optimisation suivant :

$$\tilde{x} \in \arg \min_{x \in \mathbb{R}^n} \|y - \mathbf{H}x\|^2 + \gamma \left\| \Delta^{(s)} x \right\|^2 + \iota_{\mathcal{C}}(x) \quad (1.22)$$

où $\Delta^{(s)}$ est un opérateur linéaire différentiel d'ordre s , γ le paramètre d'équilibre entre le terme de régularisation et l'attache aux données et $\iota_{\mathcal{C}}$ est l'indicatrice d'un ensemble convexe fermé \mathcal{C} , ici l'orthant positif. Dans le cas pratique, l'opérateur $\Delta^{(s)}$ utilisé est le laplacien ($s = 2$), car il est diagonalisable dans la base de Fourier.

La solution est obtenue avec la méthode du gradient conjugué projeté :

$$x_{t+1} = \mathcal{P}_{\mathcal{C}}(x_t + \tau_t d_t) \quad (1.23)$$

avec d_t la descente de gradient conjugué à l'itération t et τ_t le pas de descente à cette même itération. Cette direction de descente est calculée pour être optimale afin de converger rapidement vers une solution :

$$d_t = r_t + \frac{\|r_t\|^2}{\|r_{t-1}\|^2} d_{t-1}, \quad (1.24)$$

avec r_t la direction de descente la plus profonde à l'itération t :

$$r_t = \frac{1}{2}(\mathbf{H}^* \mathbf{H} + \gamma \Delta^{(s)*} \Delta^{(s)}) x_t - \mathbf{H}^* y. \quad (1.25)$$

Le pas τ peut être obtenu de différentes façons : pas optimal, pas adaptatif, etc.

A notre connaissance, il n'existe pas de résultat de convergence pour la descente de gradient conjugué projeté. Ceci implique que la suite $(x_t)_{t \in \mathbb{N}}$ peut ne pas converger vers une solution. De plus, avec la régularisation de Tikhonov, les images estimées sont souvent trop lisses et plus γ est élevé plus l'image estimée sera lisse. A noter, qu'une valeur optimale du paramètre de régularisation γ peut être estimée en utilisant, par exemple, la validation croisée généralisée [Golub et al., 1979; Kempen, 1999].

Une approche similaire a été proposée par Carrington [Carrington, 1990; Carrington et al., 1995] pour résoudre le problème (1.22), dans le cas particulier où $s = 0$ (opérateur identité). Ceci revient en fait à un filtre de Wiener contraint par la positivité.

1.2.3.7 Régularisation basée sur la parcimonie

Les régularisations précédentes ne sont pas toujours adaptées aux images à restaurer. Dans le contexte de la déconvolution avec un bruit blanc gaussien, il a été récemment préconisé

d’utiliser une régularisation basée sur la promotion de la parcimonie sur les coefficients des ondelettes orthogonales [Figueiredo et Nowak, 2003; Daubechies et al., 2004; Combettes et Wajs, 2005]. La généralisation aux trames a ensuite été soumise par plusieurs auteurs [Fadili et al., 2006; Teschke, 2007; Chaux et al., 2007b]. Par exemple, Fadili et Starck [2006] développent un algorithme de déconvolution basé sur un seuillage itératif dans un dictionnaire redondant de transformées. Dans un cadre un peu différent, Starck et al. [2003] combinent à la fois transformées d’ondelettes et de curvelets.

D’un point de vue strict, le terme de parcimonie est défini en utilisant la pseudo-norme ℓ_0 qui calcule le nombre de coefficients non-nuls dans le vecteur. Cependant, la fonction objectif associée est alors non-convexe et induit un problème NP-difficile. Afin d’obtenir une formulation convexe et donc plus facile à résoudre, celui-ci est relaxé en remplaçant la pseudo-norme ℓ_0 par la norme convexe ℓ_1 . Les solutions deviennent alors plus simples à caractériser, toutefois l’équivalence entre ces deux problèmes (relaxé et non-relaxé) n’est pas toujours acquise. Cependant, d’importants liens entre eux ont été montrés et sous certaines conditions, ils sont équivalents (voir [Bruckstein et al., 2009] pour une revue détaillée).

Soient Φ l’opérateur de synthèse à partir des coefficients et Φ^* l’opérateur d’analyse. $\Phi \in \mathbb{R}^{m \times n}$ est aussi le dictionnaire contenant les différentes transformées choisies (voir section 4.1 pour plus de détails). Le problème d’optimisation considéré est alors le suivant :

$$\tilde{\alpha} \in \arg \min_{\alpha \in \mathbb{R}^m} \frac{1}{2} \|y - \mathbf{H}\Phi\alpha\|_2^2 + \gamma \|\alpha\|_1, \quad (1.26)$$

avec γ le paramètre de régularisation. L’image estimée \tilde{x} est alors obtenue en appliquant l’opérateur de reconstruction sur l’estimée finale : $\tilde{x} = \Phi\tilde{\alpha}$. Cette méthode conduit à une estimée au sens du maximum a posteriori et suppose implicitement que les coefficients $(\alpha[i])_{1 \leq i \leq m}$ sont identiquement distribués avec une densité laplacienne $\propto e^{-\gamma|\alpha[i]|}$. Une solution est alors obtenue avec l’itération proximale explicite-implicite suivante (voir section 2.2.2) pour plus de détail :

$$\alpha_{t+1} = \mathbf{ST}_{\gamma\tau_t/2}(\alpha_t + \tau_t\Phi^*\mathbf{H}^*(y - \mathbf{H}\Phi\alpha_t)), \quad (1.27)$$

où \mathbf{ST}_β est l’opérateur de seuillage doux de seuil β et $(\tau_t)_{t \in \mathbb{N}} \in (0, 2/(\|\mathbf{H}\|^2 \|\Phi\|^2))$ est une suite bornée dont les valeurs sont suffisamment petites pour permettre la convergence. Cet algorithme est aussi connu sous le nom de Landweber seuillé [Figueiredo et Nowak, 2003; Daubechies et al., 2004].

Dans le cas où le dictionnaire contient une transformée orthogonale d’ondelettes (plus précisément les ondelettes de Shannon), Vonesch et Unser [2008] ont proposé une technique pour déterminer les pas de descente τ_t selon chaque sous-bande de la transformée. Celle-ci a été appliquée avec succès pour la déconvolution d’images issues d’un microscope à champ large.

1.2.4 Attache aux données poissonnienne

Les méthodes précédentes sont toutes basées sur un modèle de dégradation faisant intervenir un bruit gaussien et non un bruit de Poisson. Nous avons discuté de la justification de cette approche dans la section 1.2.3. Expérimentalement, il est considéré qu'à partir d'un taux de comptage moyen de 30, l'approximation gaussienne peut être utilisée correctement [Starck et Murtagh, 2006]. Pour des signaux faibles, donc de faibles taux de comptage, il est nécessaire d'être plus en adéquation avec la loi de Poisson.

La recherche, pour résoudre les problèmes inverses, a récemment conduit à quelques algorithmes de restauration dans le cadre poissonnien. Une de ces méthodes [Nowak et Kolaczyk, 2000] introduit un cadre bayésien multi-échelle basé sur une factorisation multi-échelle de la fonction de vraisemblance de la loi de Poisson, et associée à un partitionnement spatial dyadique récursif de l'image des intensités sous-jacentes. La solution du maximum a posteriori est calculée en utilisant un algorithme EM (*expectation-maximization*). Cependant, l'analyse multi-échelle proposée par les auteurs [Nowak et Kolaczyk, 2000] est possible uniquement avec les ondelettes de Haar qui engendrent des artefacts typiques d'effet de bloc.

Similairement, le travail de Cavalier et Koo [2002] sur les estimateurs basés sur le seuillage dur dans le cadre des données tomographiques montre que pour un opérateur particulier (l'opérateur de Radon) une extension de la méthode de décomposition ondelette-vaguelette (WVD ou *Wavelet-Vaguelet Decomposition*) [Donoho, 1995] est théoriquement possible pour les données poissonniennes. Toutefois les auteurs ne fournissent aucun algorithme pour calculer l'estimée.

Inspirée de la méthode WVD, Antoniadis et Bigot [2006] explorent une approche alternative via des décompositions, basées sur des ondelettes, combinées avec une stratégie de seuillage qui s'adresse à des problèmes d'adaptativité. Plus précisément, leur cadre étend les techniques basées sur les ondelettes de Galerkin [Cohen et al., 2004] au bruit de Poisson. La positivité des intensités estimées est assurée en travaillant sur le logarithme de l'intensité. Cette méthode est cependant limitée aux bases d'ondelettes orthogonales.

Le reste des algorithmes que nous développons dans la suite sont parmi les plus utilisées dans le cadre du bruit de Poisson. Ils sont généralement basés sur l'algorithme dit de Richardson-Lucy ou bien tentent de retrouver un cadre gaussien en stabilisant le bruit.

1.2.4.1 L'algorithme de Richardson-Lucy

Dans le cas du bruit de Poisson à réalisations indépendantes, la formule de la vraisemblance est :

$$p(y|x) = \prod_i \frac{((\mathbf{H}x)[i])^{y[i]} \exp(-(\mathbf{H}x)[i])}{y[i]!} . \quad (1.28)$$

Par le passage au logarithme de la vraisemblance et en calculant le gradient par rapport à x , il est aisé de prouver que l'estimée du maximum de vraisemblance satisfait l'équation non-linéaire :

$$\mathbf{H}^* \left(\frac{y[i]}{(\mathbf{H}x)[i]} \right)_{1 \leq i \leq n} - \mathbf{I} = 0, \quad (1.29)$$

avec \mathbf{I} l'opérateur identité (ici dans \mathbb{R}^n).

Cette dernière équation combinée à l'itération du point fixe de Picard conduit à l'algorithme de Richardson-Lucy [Lucy, 1974] :

$$x_{t+1} = \left[\mathbf{H}^* \left(\frac{y[i]}{(\mathbf{H}x_t)[i]} \right)_{1 \leq i \leq n} \right] \odot x_t, \quad (1.30)$$

où \odot est une multiplication terme à terme de deux vecteurs. Les avantages de cette méthode sont la conservation de la dynamique de l'image, de la positivité et le lien avec le modèle de dégradation. Mais, elle n'est pas régularisée et le bruit est amplifié après quelques itérations. Le nombre d'itérations peut alors faire office de paramètre de régularisation. Cependant un faible nombre d'itérations conduira à une faible déconvolution.

Cette itération peut également être interprétée dans un cadre EM [Lange, 1999]. Elle est très utilisée en astronomie. De nombreuses méthodes de déconvolution sous bruit de Poisson sont des versions régularisées de RL. Parmi les différents termes de régularisations proposés, nous trouvons la variation totale [Dey et al., 2004], la régularisation de Tikhonov-Miller [Kempen, 1999; Kempen et van Vliet, 2000], le support multi-échelle ou encore l'entropie [Starck et Murtagh, 2006].

1.2.4.2 L'algorithme de Richardson-Lucy avec la régularisation de Tikhonov-Miller

De même que pour le cas gaussien, l'une des premières régularisations introduites dans le cas poissonnien est celle de Tikhonov-Miller [Kempen, 1999; Kempen et van Vliet, 2000]. Une approche par l'estimateur du maximum a posteriori est développée conduisant au problème d'optimisation suivant :

$$\tilde{x} \in \arg \min_{x \in \mathbb{R}^n} -\ell_{\text{poisson}}(y|x) + \gamma \|\bar{\nabla}x\|^2, \quad (1.31)$$

avec ℓ_{poisson} le log de vraisemblance lié à la loi de Poisson qui est déduit de (1.28) et $\bar{\nabla}$ l'opérateur gradient discret d'une image.

Une solution est alors obtenue en utilisant l'itération de Picard sur le gradient :

$$x_{t+1} = \left[\mathbf{H}^* \left(\frac{y[i]}{(\mathbf{H}x_t)[i]} \right)_{1 \leq i \leq n} \right] \odot \left(\frac{x_t[i]}{1 - 2\gamma \Delta x_t[i]} \right)_{1 \leq i \leq n}, \quad (1.32)$$

avec γ le paramètre de régularisation, Δ l'opérateur laplacien et \odot la multiplication terme à terme de deux vecteurs.

Cet algorithme est rapide et facile à mettre en place. Toutefois, cette régularisation, comme dans le cas gaussien, ne préserve pas les contours et tend à produire une image estimée trop lisse. De plus, le choix du paramètre γ peut être assez difficile à trouver en pratique, bien que des critères de sélection de modèles peuvent être mis en place, mais avec moins de facilité que dans le cas gaussien puisque (1.32) est non-linéaire.

1.2.4.3 L'algorithme de Richardson-Lucy avec une régularisation basé sur la variation totale

La régularisation de Tikhonov ne préservant pas les contours, Dey et al. [2004] proposent de la remplacer par la variation totale conduisant au problème d'optimisation :

$$\tilde{x} \in \arg \min_{x \in \mathbb{R}^n} -\ell_{\text{poisson}}(y|x) + \gamma \|x\|_{\text{TV}} , \quad (1.33)$$

avec $\|\cdot\|_{\text{TV}}$ la semi-norme TV. Cette norme n'étant pas différentiable, la solution ne peut pas être calculée avec les méthodes habituelles. Dey et al. [2004] utilisent une formulation continue de (1.33) et passent par le calcul des variations. Ensuite, ils discrétisent l'EDP et obtiennent l'itération suivante :

$$x_{t+1} = \left[\mathbf{H}^* \left(\frac{y[i]}{(\mathbf{H}x_t)[i]} \right)_{1 \leq i \leq n} \right] \odot \left(\frac{x_t[i]}{1 - \gamma \overline{\text{div}} \left(\frac{\overline{\nabla} x_t[i]}{|\overline{\nabla} x_t[i]|} \right)} \right)_{1 \leq i \leq n} , \quad (1.34)$$

avec γ le paramètre de régularisation, $\overline{\nabla}$ l'opérateur gradient discret pour une image, $\overline{\text{div}}$ l'opérateur de divergence discret (opérateur adjoint de $\overline{\nabla}$) et \odot la multiplication terme à terme de deux vecteurs.

Cette méthode a l'avantage de bien mieux préserver les contours et les zones homogènes. En revanche les zones texturées ne sont pas préservées. Le choix du paramètre de régularisation reste assez délicat. En effet, pour certaines valeurs de γ , le résultat final peut devenir négatif, ce qui est en contradiction avec le modèle de dégradation (1.1) (la loi de Poisson est définie uniquement pour des données positives). De plus, le passage du continu au discret peut être source d'erreurs numériques.

1.2.4.4 L'algorithme de Richardson-Lucy régularisé par le support multi-résolution

Une autre méthode pour supprimer le bruit est d'opérer un seuillage dans un domaine donné comme celui de la transformée en ondelettes. L'algorithme suivant repose sur une heuristique :

le bruit présent dans le résidu (l'image des différences entre l'estimée courante convoluée par h et l'observation) domine essentiellement les petits coefficients dans le domaine des ondelettes.

Introduit par Murtagh et al. [1994], le support multi-résolution est une technique de suppression du bruit basée sur une vue multi-échelle de l'image. Cette suppression est effectuée par l'intermédiaire d'un seuillage dans le domaine des ondelettes. L'objectif de cette méthode est d'enlever le bruit tout en préservant les structures présentes dans l'image. Pour cela, l'itération de RL (1.30) est réécrite en utilisant le résidu $r_t = y - \mathbf{H}x_t$:

$$x_{t+1} = \left[\mathbf{H}^* \left(\frac{r_t[i] + (\mathbf{H}x_t)[i]}{(\mathbf{H}x_t)[i]} \right)_{1 \leq i \leq n} \right] \odot x_t . \quad (1.35)$$

Le principe de cet algorithme est de retirer, à chaque itération le bruit présent dans le résidu. Pour cela, le bruit de Poisson est stabilisé par l'application de la transformée stabilisatrice de la variance d'Anscombe [Anscombe, 1948] : $\text{ansc}(t) = 2\sqrt{t + \frac{3}{8}}$; le résidu stabilisé s'écrit donc : $(\text{ansc}(y[i]) - \text{ansc}((\mathbf{H}x_t)[i]))_{1 \leq i \leq n}$. Le bruit est retiré du résidu stabilisé en effectuant un seuillage dans le domaine des ondelettes et en construisant le support multi-résolution, puis le résidu débruité \bar{r}_t est reconstruit à partir de ce support.

Enfin, l'itération de RL régularisée est réécrite en utilisant le résidu débruité :

$$x_{t+1} = \left[\mathbf{H}^* \left(\frac{\bar{r}_t[i] + (\mathbf{H}x_t)[i]}{(\mathbf{H}x_t)[i]} \right)_{1 \leq i \leq n} \right] \odot x_t . \quad (1.36)$$

Les structures sont bien préservées et une bonne partie de la photométrie est retrouvée. Toutefois, le seuillage dans le domaine des ondelettes peut parfois engendrer des artefacts dans l'image restaurée. De plus, aucun résultat de convergence vers une solution n'est disponible. Remarquons aussi que, contrairement aux autres méthodes présentées jusqu'ici, celle-ci est la première à ne pas reposer sur un cadre bayésien stricto sensu, mais plutôt sur une approche heuristique d'amélioration d'un algorithme existant.

1.2.4.5 Stabilisation de la variance

Un autre point de vue pour prendre en compte le bruit de Poisson est d'appliquer au préalable une transformée stabilisatrice de la variance, par exemples la transformée d'Anscombe [Anscombe, 1948] ou celle de Fisz [Fisz, 1955], afin de retourner dans le cas d'une dégradation gaussienne (1.11). Ainsi Chaux et al. [2007a] proposent de combiner une de ces transformées avec un algorithme de déconvolution sous bruit gaussien régularisé par promotion de la parcimonie (section 1.2.3.7). Ils supposent donc que les observations stabilisées ont été linéairement dégradées et contaminées par un bruit gaussien additif. Toutefois, ceci n'est pas valide pour la transformée d'Anscombe car les résultats standards (comme nous verrons dans le chapitre 4) conduisent à une équation de dégradation non-linéaire (due à la présence d'une racine carrée).

La dégradation est linéaire avec Fisz mais elle est basée sur la transformée de Haar qui n'est pas invariante par translation. La technique de *cycle spinning* permet d'obtenir cette invariance, cependant l'algorithme final est long en pratique car il faut résoudre plusieurs fois le problème de déconvolution.

1.2.5 Déconvolution aveugle

Lorsqu'aucune information sur la PSF n'est disponible, il faut déduire l'information à partir de l'image observée elle-même. Dans le cadre de la déconvolution sous bruit de Poisson, plusieurs approches ont été proposées pour le cas aveugle, lorsqu'aucune paramétrisation n'est disponible ou myope, lorsque les informations sont partiellement disponibles [Molina et al., 2006; Burger et Scherzer, 2001; Conchello et Yu, 1996].

Récemment, une approche par minimisation alternée a été développée par Pankajakshan et al. [2009] pour restaurer les images issues d'un microscope confocal. Celle-ci combine l'approximation de la PSF par des gaussiennes [Zhang et al., 2007a], avec l'algorithme de Richardson-Lucy régularisé avec la variation totale (section 1.2.4.3).

Toutefois les méthodes de déconvolution aveugle ou myope produisent des résultats moins précis que ceux obtenus avec une PSF mesurée ou modélisée [Pawley, 2005]. Cette observation est justifiée dans le sens où la déconvolution aveugle (ou myope) est un problème plus complexe que la simple déconvolution avec un ensemble de solutions beaucoup plus large. De plus, la fonction objectif n'étant pas convexe, atteindre un minimum global (ou même local voire un point stationnaire) n'est pas toujours possible ou peut être extrêmement coûteux en pratique. Des heuristiques sont souvent appliquées pour simplifier, au détriment de la qualité de la solution.

1.3 Conclusion

La chronologie de développement des algorithmes de déconvolution renseigne sur les deux principales problématiques : la gestion du bruit et la régularisation de l'espace des solutions. Pour lier ces deux informations, l'approche par l'estimateur MAP est privilégiée, conduisant à la résolution d'un problème d'optimisation. Remarquons qu'avec le temps, les méthodes linéaires ont été abandonnées au profit de celles itératives plus lentes, mais plus robustes et performantes.

Au niveau de la gestion du bruit, nous constatons que les algorithmes basés sur Richardson-Lucy sont considérés comme produisant de bonnes déconvolutions [Pawley, 2005; Starck et Murtagh, 2006] grâce à une attache aux données fidèle au modèle du bruit. Toutefois, la non-linéarité du log de vraisemblance de cette loi tend à rendre complexe l'ajout d'une ré-

gularisation. Les méthodes de déconvolution avec bruit de Poisson ont d'ailleurs fortement évolué, notamment grâce aux développements aussi bien en théorie de l'optimisation qu'en régularisation (e.g. parcimonie).

Dans les chapitres suivants, nous proposons un cadre théorique pour résoudre de nombreux problèmes d'optimisation convexes. Puis en exploitant ce cadre, plusieurs méthodes pour la déconvolution sous bruit de Poisson sont présentées avec des termes d'attaches aux données fidèles à la statistique du bruit, une contrainte de positivité et une régularisation puisant dans la notion de parcimonie.

2 Notre cadre

La science suscite un monde, non plus par une impulsion magique, immanente à la réalité, mais bien par une impulsion rationnelle, immanente à l'esprit.

Gaston Bachelard, *Le nouvel esprit scientifique*

Beaucoup de problématiques liées au traitement du signal et des images se ramènent à de l'optimisation. La popularité de l'estimation bayésienne, et en particulier l'estimateur du maximum a posteriori, n'est pas sans incidence sur l'importance de l'analyse convexe en traitement du signal. Dans ce chapitre, nous proposons un cadre général pour résoudre les problèmes dont la fonction objectif s'écrit comme la somme de trois fonctions convexes semi-continues inférieurement. De plus, si l'une est différentiable et possède un gradient Lipschitz, nous pouvons exploiter cette propriété.

Premièrement, nous introduisons quelques notions, puis présentons l'opérateur proximal, une généralisation de la projection sur un ensemble convexe fermé introduite par Moreau [1962]. Ensuite, les conditions d'existence et d'unicité des solutions sont démontrées. Nous proposons plusieurs schémas de résolution basés sur les opérateurs proximaux, dont nous prouvons la convergence vers une solution. Enfin, nous présentons un dernier schéma, pour le cas général où la fonction se décompose en une somme finie de fonctions convexes semi-continues inférieurement.

2.1 Elements d'analyse convexe

Dans ce chapitre, et les chapitres suivants de cette partie, nous nous plaçons dans un espace de Hilbert $\mathcal{H} \subseteq \mathbb{R}^n$ de dimension finie n . Nous notons $\langle \cdot, \cdot \rangle$ le produit scalaire dans \mathcal{H} et $\|\cdot\|$ la norme associée. \mathbf{I} représente l'opérateur identité de \mathcal{H} . La norme (induite) d'un opérateur $\mathbf{A} : \mathcal{H} \rightarrow \mathcal{H}'$ est définie par $\|\mathbf{A}\| = \sup_{x \in \mathcal{H}} \frac{\|\mathbf{A}x\|}{\|x\|}$. Soient la norme ℓ_p définie par $\|\cdot\|_p = (\sum_i |x[i]|^p)^{1/p}$, $p \geq 1$, $\|\cdot\|_\infty = \max_i |x[i]|$, et \mathcal{B}_p^ρ la boule ℓ_p fermée de rayon $\rho > 0$.

2.1.1 Définitions

Nous commençons par donner quelques notions et définitions essentielles en analyse convexe. D'abord, nous introduisons la notion d'ensemble convexe ainsi que trois ensembles liés à un tel ensemble.

Définition 2.1.1 (Ensemble convexe).

Un ensemble fermé $\mathcal{E} \subseteq \mathcal{H}$ est dit convexe si et seulement si :

$$\forall x, y \in \mathcal{E}, \quad 0 \leq \zeta \leq 1 \Rightarrow \zeta x + (1 - \zeta)y \in \mathcal{E}.$$

Définition 2.1.2 (Enveloppe affine).

L'enveloppe affine d'un ensemble convexe $\mathcal{E} \subseteq \mathcal{H}$ est définie par :

$$\text{aff}(\mathcal{E}) = \left\{ x \in \mathcal{H} \mid \forall i, y_i \in \mathcal{E}, x = \sum_{i=1}^n a_i y_i \quad \text{et} \quad \sum_{i=1}^n a_i = 1 \right\}.$$

Définition 2.1.3 (Intérieur relatif).

Pour un ensemble convexe $\mathcal{E} \subseteq \mathcal{H}$, son intérieur relatif $\text{ri} \mathcal{E}$ est défini comme l'intérieur de \mathcal{E} pour la topologie relative à l'enveloppe affine de \mathcal{E} . C'est à dire, $x \in \text{ri} \mathcal{E}$ si et seulement si :

$$x \in \text{aff} \mathcal{E} \quad \text{et} \quad \exists \rho > 0 \text{ tel que } (\text{aff} \mathcal{E}) \cap \mathcal{B}_2^\rho(x) \subset \mathcal{E}.$$

Les définitions suivantes concernent les fonctions opérant dans des ensembles convexes.

Définition 2.1.4 (Coercivité).

Une fonction $f : \mathcal{H} \rightarrow (-\infty, +\infty]$ est dite coercive si et seulement si :

$$\lim_{\|x\| \rightarrow +\infty} f(x) = +\infty.$$

Définition 2.1.5 (Domaine d'une fonction).

Le domaine $\text{dom}(f)$ d'une fonction $f : \mathcal{H} \rightarrow \mathbb{R}$ est défini par :

$$\text{dom}(f) = \{x \in \mathcal{H} \mid f(x) < +\infty\}.$$

Définition 2.1.6 (Epigraphe d'une fonction).

L'épigraphe $\text{epi}(f)$ d'une fonction f est défini par :

$$\text{epi}(f) = \{(x, \zeta) \in \mathcal{H} \times \mathbb{R} \mid f(x) \leq \zeta\}.$$

Définition 2.1.7 (Fonction convexe).

Une fonction est convexe si et seulement si son épigraphe est convexe, c'est à dire si et seulement si

$$\forall x, y \in \mathcal{H}, \quad 0 \leq \zeta \leq 1, \quad f(\zeta x + (1 - \zeta)y) \leq \zeta f(x) + (1 - \zeta)f(y).$$

La fonction est dite strictement convexe si et seulement si cette inégalité est stricte.

Ensuite, nous introduisons une notion clef en analyse convexe.

Définition 2.1.8 (Semi-continuité inférieure).

Une fonction $f : \mathcal{H} \rightarrow \mathbb{R}$ est dite semi-continue inférieurement en $x_0 \in \mathcal{H}$ si et seulement si :

$$\liminf_{x \rightarrow x_0} f(x) \geq f(x_0), \quad x \in \mathcal{H}.$$

Par extension, une fonction est semi-continue inférieurement sur un ensemble \mathcal{E} inclus dans \mathcal{H} si et seulement si elle est semi-continue inférieurement en tout point de \mathcal{E} . La semi-continuité est une propriété plus faible que la continuité. Elle joue cependant un rôle important pour caractériser l'existence de solutions dans les problèmes d'optimisation [Lemaréchal et Hiriart-Urruty, 1996].

Nous notons $\Gamma_0(\mathcal{H})$ la classe des fonctions semi-continues inférieurement, convexes dans \mathcal{H} , à valeurs dans $(-\infty, +\infty]$ et à domaine non-vide.

Les deux définitions suivantes introduisent deux fonctions caractéristiques d'un ensemble.

Définition 2.1.9 (Fonction indicatrice d'un ensemble).

Soit \mathcal{E} un sous-espace convexe et non-vide de \mathcal{H} . La fonction indicatrice de \mathcal{E} est définie par :

$$\iota_{\mathcal{E}}(x) = \begin{cases} 0, & \text{si } x \in \mathcal{E} \\ +\infty, & \text{sinon.} \end{cases} \quad (2.1)$$

Définition 2.1.10 (Fonction d'appui (ou support) d'un ensemble).

Soit \mathcal{E} un sous-espace convexe et non-vide de \mathcal{H} . La fonction support de \mathcal{E} est définie par :

$$\forall x \in \mathcal{H}, \quad \varsigma_{\mathcal{E}}(x) = \sup \{ \langle y, x \rangle \mid y \in \mathcal{E} \} . \quad (2.2)$$

Définition 2.1.11 (Conjuguée d'une fonction ou transformée de Legendre-Fenchel).

Soit une fonction $f \in \Gamma_0(\mathcal{H})$, sa conjuguée est la fonction convexe fermée f^* définie par :

$$f^*(u) = \sup_{x \in \text{dom}(f)} \langle u, x \rangle - f(x). \quad (2.3)$$

Nous avons alors $f^* \in \Gamma_0(\mathcal{H})$ et pour la biconjuguée $f^{**} = f$.

Ces trois dernières définitions conduisent directement au résultat suivant :

Lemme 2.1.12.

La conjuguée de l'indicatrice d'un ensemble convexe \mathcal{E} , $\iota_{\mathcal{E}}$, est la fonction support de ce même ensemble $\varsigma_{\mathcal{E}}$. Par conséquent, la conjuguée de la norme ℓ_p est l'indicatrice de la boule duale unité \mathcal{B}^1 .

Puis, nous introduisons une opération qui sera par la suite utilisée pour généraliser la notion de projection.

Définition 2.1.13 (Convolution infimale ou inf-convolution).

Soient deux fonctions f_1 et f_2 dans $\Gamma_0(\mathcal{H})$. Leur inf-convolution est la fonction dans $\Gamma_0(\mathcal{H})$ définie par :

$$(f_1 \overset{\dagger}{\vee} f_2)(x) = \inf_{y \in \mathcal{H}} f_1(y) + f_2(x - y) . \quad (2.4)$$

Un résultat important de l'inf-convolution est présenté par le lemme suivant :

Lemme 2.1.14 (Lemaréchal et Hiriart-Urruty [1996, Théorème X.2.3.1]).

Soient deux fonctions f_1 et f_2 dans $\Gamma_0(\mathcal{H})$, alors :

$$(f_1 \overset{\dagger}{\vee} f_2)^* = f_1^* + f_2^* . \quad (2.5)$$

Cela revient à dire que la transformée de Legendre-Fenchel équivaut à la transformée de Fourier dans l'algèbre $(\max, +)$.

Il reste à introduire les outils nécessaires pour étudier la différentiabilité d'une fonction. La notion suivante est une généralisation de celle de la dérivée.

Définition 2.1.15 (Sous-différentielle).

La sous-différentielle d'une fonction $f \in \Gamma_0(\mathcal{H})$ en x est définie par l'ensemble :

$$\partial f(x) = \{u \in \mathcal{H} \mid \forall y \in \mathcal{H}, f(y) \geq f(x) + \langle u, y - x \rangle\} . \quad (2.6)$$

Un élément u de $\partial f(x)$ est appelé un sous-gradient. Si f est (Gâteaux-)différentiable en x , alors le seul sous-gradient en x est le gradient de f en x : $\partial f(x) = \{\nabla f(x)\}$. Si une fonction est strictement convexe alors l'inégalité dans (2.6) est stricte pour $y \neq x$ [Lemaréchal et Hiriart-Urruty, 1996, Proposition VI.6.1.3].

Définition 2.1.16 (Dérivée directionnelle).

Une fonction f admet une dérivée unidirectionnelle en x dans la direction d si

$$f'(x, d) = \lim_{\zeta \downarrow 0} \frac{f(x + \zeta d) - f(x)}{\zeta}$$

existe avec des valeurs dans $[-\infty, +\infty]$. Elle est bidirectionnelle si et seulement si $f'(x, -d)$ existe et $f'(x, -d) = -f'(x, d)$.

La dérivée directionnelle d'une fonction f en x , $f'(x, \cdot)$, est la fonction support associée à sa sous-différentielle $\partial f(x)$, c'est à dire :

$$\partial f(x) = \{u \in \mathcal{H} \mid \langle u, d \rangle \leq f'(x, d) \text{ pour tout } d \in \mathcal{H}\} . \quad (2.7)$$

Définition 2.1.17 (Convexité forte).

Une fonction f est fortement convexe de module $c > 0$ si et seulement si :

$$\forall x, z \in \mathcal{H}, \forall u \in \partial f(x), \quad f(z) \geq f(x) + \langle u, z - x \rangle + \frac{c}{2} \|z - x\|^2 . \quad (2.8)$$

Définition 2.1.18 (Lipschitz continuité).

Un opérateur $\mathbf{A} : \mathcal{H} \rightarrow \mathcal{H}'$ est κ -Lipschitz continue si et seulement s'il vérifie :

$$\forall x, y \in \mathcal{H}, \|\mathbf{A}(x) - \mathbf{A}(y)\| \leq \kappa \|x - y\|. \quad (2.9)$$

Un opérateur 1-Lipschitz est aussi dit non-expansif.

Définition 2.1.19 (Opérateur monotone).

Un opérateur $\mathbf{A} : \mathcal{H} \rightarrow 2^{\mathcal{H}'}$, à valeurs dans l'ensemble des sous-parties de \mathcal{H}' , est dit monotone si et seulement si

$$\forall (x, u), (y, v) \in \mathcal{H} \times \mathcal{H}' \text{ tels que } u \in \mathbf{A}(x), v \in \mathbf{A}(y), \quad \langle x - y, u - v \rangle \geq 0.$$

Un opérateur monotone est maximal si en tant que graphe dans $\mathcal{H} \times \mathcal{H}'$, il n'est pas inclus dans tout autre opérateur monotone sur \mathcal{H} .

Il est facile de voir que, par définition dans (2.6), la sous-différentielle est un opérateur maximal monotone. De plus, pour un opérateur maximal monotone \mathbf{A} , il existe un unique point $J_{\mathbf{A}}(x)$ tel que

$$x - J_{\mathbf{A}}(x) \in \mathbf{A}(J_{\mathbf{A}}(x)),$$

d'où la définition de la résolvante.

Définition 2.1.20 (Résolvante).

La résolvante $J_{\zeta \mathbf{A}}$ d'un opérateur $\mathbf{A} : \mathcal{H} \rightarrow 2^{\mathcal{H}'}$ est définie par :

$$J_{\zeta \mathbf{A}} = (\mathbf{I} + \zeta \mathbf{A})^{-1}.$$

La résolvante d'un opérateur monotone \mathbf{A} est à valeur unique et si, de plus, \mathbf{A} est maximal, alors $J_{\mathbf{A}}$ est à domaine plein [Eckstein et Bertsekas, 1992].

Pour finir, nous résumons dans le lemme suivant plusieurs résultats caractérisant le minimum d'une fonction $f \in \Gamma_0(\mathcal{H})$:

Lemme 2.1.21 (Rockafellar [1970]; Ekeland et Teman [1974]; Rockafellar [1981]).

- (i) *Condition de minimalité* : x^* est un minimum global de $f \in \Gamma_0(\mathcal{H})$ sur \mathcal{H} si et seulement si $0 \in \partial f(x^*)$ et $f'(x^*, d) \geq 0$ pour tout d . Le minimum est unique si f est strictement convexe.
- (ii) *Dualité de Fenchel-Rockafellar* : soient $f \in \Gamma_0(\mathcal{H})$ et $g \in \Gamma_0(\mathcal{H}')$, $\mathbf{A} := \mathbf{F} \cdot -y : \mathcal{H} \rightarrow \mathcal{H}'$ un opérateur affine borné, et \mathcal{H}' un espace de dimension finie. Supposons que $0 \in \text{ri}(\text{dom}(g) - \mathbf{A}(\text{dom}(f)))$. Alors

$$\inf_{x \in \mathcal{H}} f(x) + g \circ \mathbf{A}(x) = - \min_{u \in \mathcal{H}'} f^*(-\mathbf{F}^* u) + g^*(u) + \langle u, y \rangle, \quad (2.10)$$

avec la relation entre x^* et u^* , respectivement les solutions des problèmes primal et dual :

$$f(x^*) + f^*(-\mathbf{F}^*u^*) = \langle -\mathbf{F}^*u^*, x^* \rangle, \quad (2.11)$$

$$g(\mathbf{A}x^*) + g^*(u^*) = \langle u^*, \mathbf{A}x^* \rangle, \quad (2.12)$$

ce qui équivaut à :

$$x^* \in \partial f^*(-\mathbf{F}^*u^*) \text{ et } u^* \in \partial g(\mathbf{A}x^*) . \quad (2.13)$$

La dualité est un outil primordial pour résoudre certaines formulations complexes dans le domaine primal. La composition avec un opérateur affine borné étant un élément clef dans le cadre de la déconvolution, mais aussi dans le cadre d'utilisation de trames (voir chapitre 4), ce résultat va nous permettre d'étendre les possibilités des algorithmes d'optimisation.

2.1.2 Opérateur et calcul proximaux

Maintenant que nous avons introduit quelques définitions essentielles, nous pouvons présenter l'opérateur proximal. Celui-ci est une généralisation de la notion de projection sur un ensemble convexe fermé et forme la pierre angulaire des schémas d'optimisation que nous exposerons ultérieurement.

Définition 2.1.22 (Moreau [1962]).

Soit $f \in \Gamma_0(\mathcal{H})$, l'opérateur proximal associé à f , noté $\text{prox}_f : \mathcal{H} \rightarrow \mathcal{H}$, est défini comme l'inf-convolution avec la fonction $x \mapsto \|x\|^2/2$. La valeur de l'opérateur en un point x est donc donnée par la solution de $\inf_{y \in \mathcal{H}} f(y) + \|x - y\|^2/2$ qui atteint son minimum en un unique point noté $\text{prox}_f x$.

Remarquons que l'opérateur proximal de la fonction indicatrice d'un ensemble convexe fermé \mathcal{E} est alors le projecteur sur \mathcal{E} . Nous introduisons aussi l'opérateur de réflexion, défini par $\text{rprox}_f = 2 \text{prox}_f - \mathbf{I}$.

Il est facile d'établir le lemme suivant :

Lemme 2.1.23.

Soient $\forall x, p \in \mathcal{H}$, alors

$$p = \text{prox}_f x \iff x - p \in \partial f(p) \iff \langle y - p, x - p \rangle + f(p) \leq f(y), \forall y \in \mathcal{H}. \quad (2.14)$$

Ce qui implique $\text{prox}_f = J_{\partial f}$; prox_f est donc la résolvante de la sous-différentielle de f .

Les opérateurs proximaux sont liés aux enveloppes de Moreau :

Définition 2.1.24 (Enveloppe de Moreau).

L'enveloppe de Moreau d'ordre $\zeta \in (0, +\infty)$ d'une fonction $f \in \Gamma_0(\mathcal{H})$ est la fonction continue et convexe :

$${}^{(\zeta)}f : \mathcal{H} \rightarrow \mathbb{R} : x \mapsto \inf_{y \in \mathcal{H}} f(y) + \frac{1}{2\zeta} \|x - y\|^2 . \quad (2.15)$$

L'enveloppe de Moreau d'ordre ζ de f peut être aussi définie comme l'inf-convolution avec $x \mapsto \frac{1}{2\zeta} \|x\|^2$.

Remplacer une fonction par son enveloppe de Moreau s'appelle également la *régularisation de Moreau-Yosida* [Lemaréchal et Hiriart-Urruty, 1996].

2.1.2.1 Propriétés

L'opérateur proximal possède un grand nombre de propriétés, dont nous avons résumé une partie dans la proposition suivante :

Proposition 2.1.25 (Moreau [1962, 1963, 1965]; Combettes et Wajs [2005]).

Soient $f \in \Gamma_0(\mathcal{H})$ et $x \in \mathcal{H}$. Alors nous avons :

1. *Perturbation quadratique* : soit $g = f + \zeta \|\cdot\|^2/2 + \langle \cdot, u \rangle + \beta$, avec $u \in \mathcal{H}$, $\zeta \in [0, +\infty[$ et $\beta \in \mathbb{R}$. Alors $\text{prox}_g x = \text{prox}_{f/(\zeta+1)}((x - u)/(\zeta + 1))$.
2. *Translation* : soit $g = f(\cdot - z)$, avec $z \in \mathcal{H}$. Alors $\text{prox}_g x = z + \text{prox}_f(x - z)$.
3. *Changement d'échelle* : soit $g = f(\cdot/\zeta)$, avec $\zeta \in \mathbb{R} \setminus \{0\}$. Alors $\text{prox}_g x = \zeta \text{prox}_{f/\zeta^2}(x/\zeta)$.
4. *Réflexion* : soit $g : x \mapsto f(-x)$. Alors $\text{prox}_g x = -\text{prox}_f(-x)$.
5. *Séparabilité* : soient $\{f_i\}_{1 \leq i \leq p}$ une famille de fonctions dans $\Gamma_0(\mathcal{H})$, et f une fonction définie sur \mathcal{H}^p avec $f(\alpha) = \sum_{i=1}^p f_i(\alpha_i)$, $\alpha_i \in \mathcal{H}$. Alors $\text{prox}_f = \{\text{prox}_{f_i}\}_{1 \leq i \leq p}$.

Le lemme suivant résume quelques propriétés des enveloppes de Moreau.

Lemme 2.1.26 (Moreau [1965]; Lemaréchal et Hiriart-Urruty [1996]; Combettes et Hirstoaga [2005]).

Soit $f \in \Gamma_0(\mathcal{H})$. Alors son enveloppe de Moreau ${}^{(\zeta)}f$ est convexe, Fréchet-différentiable avec un gradient $1/\zeta$ Lipschitz et :

$$\nabla^{(\zeta)} f = (\mathbf{I} - \text{prox}_{\zeta f})/\zeta . \quad (2.16)$$

De plus, son opérateur proximal est formé par la combinaison convexe :

$$\text{prox}_{{}^{(\zeta)}f}(x) = \frac{\zeta}{1 + \zeta} x + \frac{1}{1 + \zeta} \text{prox}_{(1+\zeta)f}(x) . \quad (2.17)$$

Lemme 2.1.27 (Décomposition de Moreau [Moreau, 1965; Rockafellar, 1970; Combettes et Wajs, 2005]).

Soit $f \in \Gamma_0(\mathcal{H})$, alors pour tout $x \in \mathcal{H}$,

$$\text{prox}_{\zeta f^*}(x) + \zeta \text{prox}_{f/\zeta}(x/\zeta) = x, \quad \forall \zeta, \quad 0 < \zeta < +\infty . \quad (2.18)$$

A partir de ce lemme, nous pouvons conclure que :

$$\text{prox}_{f^*} = \mathbf{I} - \text{prox}_f \iff \text{prox}_{f^*}(x) \in \partial f(\text{prox}_f(x)) . \quad (2.19)$$

2.1.2.2 Fonctions favorisant la parcimonie

Dans le chapitre 1, nous avons fait part d'une augmentation sensible d'algorithmes reposant sur la parcimonie pour construire un terme de régularisation. Nous considérons ici une famille de fonctions potentielles promouvant la parcimonie telles que l'opérateur proximal associé possède une forme analytique facilement calculable (pour reprendre la terminologie de [Nesterov, 2007], nous parlerons de fonction simple). Soit la fonction (additive) de pénalité suivante :

$$\Psi(\alpha) = \sum_{i=1}^T \psi_i(\alpha_i) . \quad (2.20)$$

En supposant $\forall i, 1 \leq i \leq T$ et $\psi_i : \mathbb{R} \rightarrow \mathbb{R}^+$ et vérifiant les hypothèses suivantes :

Hypothèses 2.1.28.

- (i) ψ_i est une fonction convexe, symétrique, positive, non-décroissante sur $[0, +\infty[$ et $\psi_i(0) = 0$.
- (ii) ψ_i est deux fois différentiable sur $\mathbb{R} \setminus \{0\}$.
- (iii) ψ_i est continue sur \mathbb{R} , elle n'est pas nécessairement différentiable en 0 et admet une dérivée positive à droite en 0, $\psi'_{i+}(0) = \lim_{h \downarrow 0} \frac{\psi_i(h)}{h} > 0$.

Lemme 2.1.29 (Fadili et Bullmore [2005]; Fadili et al. [2006]).

Si les hypothèses 2.1.28 sont vérifiées, alors l'opérateur proximal de $\delta\Psi(\alpha)$, $\text{prox}_{\delta\Psi}(\alpha)$ possède exactement une unique solution continue, découplée en chaque coordonnée $\alpha[i]$:

$$\text{prox}_{\delta\psi_i}(\alpha[i]) = \begin{cases} 0 & \text{si } |\alpha[i]| \leq \delta\psi'_{i+}(0) , \\ \alpha[i] - \delta\psi'_i(\alpha[i]) & \text{si } |\alpha[i]| > \delta\psi'_{i+}(0) . \end{cases} \quad (2.21)$$

Ce résultat peut être retrouvé en appliquant la propriété de séparabilité (Proposition 2.1.25 et Lemme 2.1.21(i)).

Pour le cas de la norme ℓ_1 , $\Psi(\alpha) = \sum_i |\alpha[i]| = \|\alpha\|_1$, nous retrouvons le seuillage doux :

$$\text{prox}_{\gamma\|\cdot\|_1}(\alpha) = \mathbf{ST}_\gamma(\alpha) = \left(\max \left(1 - \frac{\gamma}{|\alpha_i|}, 0 \right) \alpha_i \right)_{1 \leq i \leq T} . \quad (2.22)$$

D'autres exemples de fonctions vérifiant les conditions du lemme 2.1.29 sont :

- $\psi_i(\alpha_i) = \gamma |\alpha_i|^p, p > 1$;
- la fonction de Hubert :

$$\psi_i(\alpha_i) = \begin{cases} \alpha_i^2/2 & \text{si } |\alpha_i| \leq \gamma, \\ \gamma |\alpha_i| - \gamma^2/2 & \text{sinon.} \end{cases}$$

– [Ni et Huo, 2009]

$$\psi_i(\alpha_i) = \begin{cases} -\log \cos(\delta\alpha_i/\gamma) & \text{si } |\alpha_i| < \gamma, \\ \delta \tan(\delta)(|\alpha_i/\gamma| - 1) - \log \cos(\delta) & \text{sinon} \end{cases}, \quad 0 < \delta < \pi/2.$$

La figure 2.1 montrent les trois fonctions (à gauche) et leurs opérateurs proximaux correspondants (à droite).

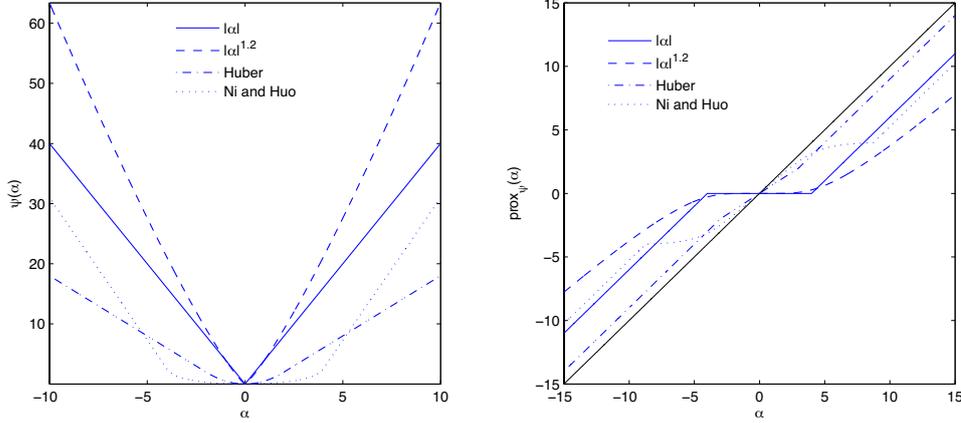


FIG. 2.1: Opérateurs proximaux associés à des fonctions promouvant la parcimonie : à gauche, les différentes fonctions et à droite les opérateurs proximaux correspondants.

2.2 Décomposition d'opérateurs maximaux monotones

Soit $f \in \Gamma_0(\mathcal{H})$ telle que $f = f_1 + f_2$, $f_1, f_2 \in \Gamma_0(\mathcal{H})$ et $\text{dom}(f_1) \cap \text{dom}(f_2) \neq \emptyset$. Nous voulons résoudre le problème d'optimisation suivant :

$$\min_{x \in \mathcal{H}} f(x) = \min_{x \in \mathcal{H}} f_1(x) + f_2(x). \quad (\text{P})$$

Soit $\mathcal{G} = \{x \in \mathcal{H} \mid 0 \in \partial f(x)\}$ l'ensemble des solutions, supposé non-vidé. Alors en combinant les conditions de minimalité avec la définition de l'opérateur proximal, nous obtenons :

$$\begin{aligned} & 0 \in \partial f(x), \\ \iff & 0 \in \partial(\beta f)(x), \forall \beta > 0, \\ \iff & x - x \in \partial(\beta f)(x), \\ \iff & x = \text{prox}_{\beta f}(x), \end{aligned} \quad (2.23)$$

où β est un scalaire positif nommé *pas proximal*. L'algorithme proximal par excellence est construit directement à partir de l'équation du point fixe (2.23) :

$$x_{t+1} = \text{prox}_{\beta f}(x_t). \quad (2.24)$$

Cette itération est fondamentale pour la recherche d'un minimal global de f . Martinet [1972] prouve que la suite $(x_t)_{t \in \mathbb{N}}$ converge vers $x \in \mathcal{G}$. Le calcul de $\text{prox}_{\beta f}$ est ici le point épineux. En effet, cet opérateur peut être assez difficile à calculer en pratique selon la fonction f .

2.2.1 Les origines de la décomposition

Les méthodes de décomposition pour les opérateurs maximaux monotones pour le problème (P) sont celles qui proposent de résoudre celui-ci sans avoir à évaluer directement prox_f . Elles exploitent plutôt une récursion impliquant individuellement les opérateurs proximaux prox_{f_1} et prox_{f_2} en supposant que ces derniers soient plus facilement évaluables. Elles ont donc fait l'objet d'une intense recherche, par leur vaste utilisation en analyse numérique et, en particulier, en théorie de l'optimisation convexe. De nombreuses applications sont possibles en construisant des algorithmes de décomposition pour l'analyse convexe et les inégalités variationnelles.

Une littérature extensive est disponible sur le sujet (voir [Eckstein, 1989] pour une revue). Toutes ces méthodes peuvent être, en grande partie, classées en trois familles principales : explicite-implicite [Gabay, 1983; Tseng, 1991, 2000], Douglas/Peaceman-Rachford [Lions et Mercier, 1979] (bien que les racines remontent aux méthodes numériques pour les équations aux dérivées partielles [Douglas et Rachford, 1956]), et implicite-implicite [Lions, 1978; Passty, 1979]. Nous invitons le lecteur à lire [Combettes, 2004; Eckstein et Svaiter, 2008] pour un cadre théorique général et pertinent pour les trois classes algorithmes pré-existants. Toutefois, à cause de difficultés de convergence, la classe implicite-implicite est peu utilisée, en pratique, dans le cas général de (P) et, pour cette raison, ne sera pas considérée ultérieurement dans cette thèse.

2.2.2 Schéma de décomposition explicite-implicite

Le schéma explicite-implicite est essentiellement une généralisation de la descente de gradient projeté, très classique en optimisation. Elle hérite donc de ses avantages et restrictions. Nous supposons ici que f_1 possède un gradient continu et κ -Lipschitz.

2.2.2.1 Schéma à un pas

Ce schéma est à un pas, dans le sens où, à chaque itération, seule la mémoire de l'itération précédente est utilisée dans le calcul de la descente. Son itération générale et sa convergence sont fournies par le théorème suivant :

Théorème 2.2.1 ([Combettes, 2004]).

Supposons que $\mathcal{G} \neq \emptyset$. Pour $t \geq 0$, soient $(\tau_t)_{t \in \mathbb{N}}$ une suite dans $(0, +\infty)$ telle que $0 < \inf_t \tau_t \leq$

$\sup_t \tau_t < 2/\kappa$, $(\beta_t)_{t \in \mathbb{N}}$ une suite dans $(0, 1]$ telle que $\inf_t \beta_t > 0$, et $(a_t)_{t \in \mathbb{N}}$ et $(b_t)_{t \in \mathbb{N}}$ deux suites dans \mathcal{H} telles que $\sum_t \|a_t\| < +\infty$ et $\sum_t \|b_t\| < +\infty$. Pour $x_0 \in \mathcal{H}$, posons pour tout $t \geq 0$:

$$x_{t+1} = x_t + \beta_t(\text{prox}_{\tau_t f_2}(x_t - \tau_t(\nabla f_1(x_t) + b_t))) + a_t - x_t \quad (2.25)$$

avec ∇f_1 le gradient de f_1 . Alors la suite $(x_t)_{t \in \mathbb{N}}$ converge vers une solution $x^* \in \mathcal{G} \neq \emptyset$ de (P).

Le rôle des suites $(a_t)_{t \in \mathbb{N}}$ et $(b_t)_{t \in \mathbb{N}}$ dans (2.25) est formel. Ces suites tiennent compte de la possibilité d'utiliser des approximations pour le calcul de deux éléments de l'algorithme, à savoir l'opérateur proximal et le gradient. Si les erreurs commises par les approximations sont sommables, alors l'itération converge vers une solution. Cette flexibilité sera utilisée, dans la suite, lors de l'utilisation d'une sous-itération pour le calcul même de l'opérateur proximal. Il nous suffira donc de sous-itérer suffisamment pour obtenir une erreur amortissable par l'itération extérieure.

Un des avantages de cette récursion est de reposer uniquement sur l'évaluation de $\text{prox}_{\tau_t f_2}$ et ∇f_1 . Notons que (2.25) peut être simplifiée en prenant $(a_t)_t \equiv 0$, $(b_t)_t \equiv 0$ et $(\beta_t)_t \equiv 1$ (puisque le rôle de $(a_t)_t$ et $(b_t)_t$ est purement formel). Nous obtenons alors l'itération suivante :

$$x_{t+1} = \text{prox}_{\tau_t f_2}(x_t - \tau_t \nabla f_1(x_t)). \quad (2.26)$$

L'algorithme 2.1 résume l'implémentation de cette récursion. Notons que ce schéma est celui utilisé par Figueiredo et Nowak [2003]; Daubechies et al. [2004] dans des cas particuliers.

Remarque 2.2.2.

Pour $f_2 = \iota_{\mathcal{E}}$ (\mathcal{E} étant un ensemble convexe fermé), $\text{prox}_{\iota_{\mathcal{E}}}$ correspond au projecteur sur \mathcal{E} et dans ce cas (2.26) devient une descente de gradient projeté.

Le point qui peut se révéler épineux dans certaines applications est l'estimation de la constante de Lipschitz κ , et donc de la suite $(\tau_t)_{t \in \mathbb{N}}$. Tseng [2000] propose une version du schéma explicite-implicite avec estimation du pas de descente à chaque itération et ajoute une étape d'extra-gradient. Il estime les valeurs de la suite $(\tau_t)_{t \in \mathbb{N}}$ au fur et à mesure des itérations en utilisant la règle d'Armijo-Goldstein. Celle-ci permet, aux prix de calculs supplémentaires, de trouver un pas adapté à chaque descente. Cette astuce produit dans tous les cas une estimation valable du pas de descente. L'algorithme 2.2 résume les différentes étapes impliquées dans ce schéma itératif. Lors de l'étape 1, un nouveau pas doit être calculé, Tseng [2000] choisit, par exemple, τ_t comme la plus grande valeur vérifiant le test parmi $\{\sigma\tau_*, \sigma^2\tau_*, \dots\}$, où $\sigma \in (0, 1)$ est fournie en entrée et τ_* est, par exemple, une constante donnée en entrée, ou bien la dernière valeur de pas vérifiant le test.

Algorithme 2.1 : Décomposition explicite-implicite à un pas pour résoudre (P).

Initialisation : Choisir un $x_0 \in \mathcal{H}$, une suite ou une constante $\tau_t \in (0, 2/\kappa)$.

Début

Pour $t = 0$ jusqu'à $N_{\text{ext}} - 1$ **faire**

1. Etape explicite de descente de gradient :

$$x_{t+1/2} = x_t - \tau_t \nabla f_1(x_t). \quad (2.27)$$

2. Etape implicite avec l'opérateur proximal :

$$x_{t+1} = \text{prox}_{\tau_t f_2}(x_{t+1/2}). \quad (2.28)$$

finpour

Fin

Sorties : Une solution du problème (P) : $x^* = x_{N_{\text{ext}}}$

Algorithme 2.2 : Décomposition explicite-implicite à un pas estimé pour résoudre (P).

Initialisation : Choisir un $x_0 \in \mathcal{H}$ et un $\zeta \in (0, 1)$.

Début

Pour $t = 0$ jusqu'à $N_{\text{ext}} - 1$ **faire**

Répéter

1. Choisir : $\tau_t \in \mathbb{R}^+$.

2. Etape explicite de descente de gradient :

$$x_{t+1/2} = x_t - \tau_t \nabla f_1(x_t). \quad (2.29)$$

3. Etape implicite avec l'opérateur proximal :

$$\bar{x}_{t+1} = \text{prox}_{\tau_t f_2}(x_{t+1/2}). \quad (2.30)$$

jusqu'à $\tau_t \|\nabla f_1(\bar{x}_{t+1}) - \nabla f_1(x_t)\| \leq \zeta \|\bar{x}_{t+1} - x_t\|$.

4. Etape d'extra-gradient

$$x_{t+1} = \bar{x}_{t+1} - \tau_t (\nabla f_1(\bar{x}_{t+1}) - \nabla f_1(x_t)). \quad (2.31)$$

finpour

Fin

Sorties : Une solution du problème (P) : $x^* = x_{N_{\text{ext}}}$.

2.2.2.2 Schéma à pas multiple

Les schémas précédents mémorisent uniquement l'itérée précédente pour ensuite mettre à jour la nouvelle itérée. La succession des gradients et des itérées forment toutefois une information importante sur la variation de la fonction. Celle-ci peut être mise à profit pour obtenir un pas de descente mieux adapté. Ainsi, Nesterov [2007] propose une version du schéma explicite-implicite avec un pas multiple utilisant la totalité des itérées précédentes.

L'algorithme 2.3 présente une version adaptée par Weiss [2009] à notre problème. Remarquons que cette méthode demande de calculer l'opérateur proximal deux fois à chaque étape, ce qui implique une complexité temporelle double par rapport à celle de l'algorithme 2.1.

Algorithme 2.3 : Décomposition explicite-implicite à pas multiple pour résoudre (P).

Initialisation : Choisir $x_0 \in \mathcal{H}$, $\tau \in (0, 2/\kappa)$, $b_0 = 0$ et $\eta_0 = 0$.

Début

Pour $t = 0$ jusqu'à $N_{\text{ext}} - 1$ **faire**

1. Premier calcul proximal :

$$v_t = \text{prox}_{b_t f_2}(x_0 - \eta_t) . \quad (2.32)$$

2. $a_t = (\tau + \sqrt{\tau^2 + 2\tau b_t}) / 2$ et $\omega_t = \frac{b_t x_t + a_t v_t}{b_t + a_t}$.

3. Deuxième calcul proximal :

$$x_{t+1} = \text{prox}_{\tau/2 f_2} \left(\omega_t + \frac{\tau}{2} \nabla f_1(\omega_t) \right) . \quad (2.33)$$

4. Accumulation des directions de descentes :

$$\eta_{t+1} = \eta_t - a_t \nabla f_1(x_{t+1}) . \quad (2.34)$$

5. Etape de mise à jour : $b_{t+1} = b_t + a_t$.

finpour

Fin

Sorties : Une solution du problème (P) : $x^* = x_{N_{\text{ext}}}$.

Identiquement à l'algorithme 2.1, le pas de référence peut être estimée au fur et à mesure des itérations. Pour cela, il suffit d'utiliser, par exemple, la règle d'Armijo-Goldstein et de répéter les étapes 2 et 3 jusqu'à satisfaire la condition donnée par la règle, à l'instar de l'algorithme 2.2.

Remarque 2.2.3.

Beck et Teboulle [2009] proposent un autre schéma à pas multiple utilisant uniquement les deux dernières itérées et demandant une seule évaluation de l'opérateur proximal par itération. La

vitesse de convergence sur la fonction objectif est théoriquement la même que celle de l'algorithme de Nesterov (en $1/t^2$). Rappelons quelle est en $1/t$ pour l'itération explicite-implicite.

Toutefois, la contrainte sur f_1 de posséder un gradient Lipschitz n'est pas toujours vérifiée, remettant en question l'utilisation du schéma implicite-explicite. Il faut alors se tourner vers d'autres méthodes.

2.2.3 Schéma de décomposition de Douglas-Rachford

Le schéma de Douglas-Rachford forme l'approche la plus générale avec des conditions d'utilisations plus faibles que précédemment. Nous supposons que $f_1, f_2 \in \Gamma_0(\mathcal{H})$, mais non-nécessairement différentiables. Le schéma de décomposition et sa convergence sont établis par le théorème suivant :

Théorème 2.2.4 (Combettes [2004]).

Supposons $\mathcal{G} \neq \emptyset$. Soient $\tau \in]0, +\infty[$, $(\beta_t)_{t \in \mathbb{N}}$ une suite dans $]0, 2[$, et $(a_t)_{t \in \mathbb{N}}$ et $(b_t)_{t \in \mathbb{N}}$ deux suites dans \mathcal{H} . Supposons que :

- (i) $\sum_{t \in \mathbb{N}} \beta_t(2 - \beta_t) = +\infty$.
- (ii) $\sum_{t \in \mathbb{N}} \beta_t(\|a_t\| + \|b_t\|) < +\infty$.

Prenons $x_0 \in \mathcal{H}$ et pour tout $t \in \mathbb{N}$, posons :

$$\begin{aligned} x_{t+\frac{1}{2}} &= \text{prox}_{\tau f_2} x_t + b_t, \\ x_{t+1} &= x_t + \beta_t \left(\text{prox}_{\tau f_1} \left(2x_{t+\frac{1}{2}} - x_t \right) + a_t - x_{t+\frac{1}{2}} \right). \end{aligned} \tag{2.35}$$

Alors la suite $(x_t)_{t \in \mathbb{N}}$ converge vers un point quelconque $x^* \in \mathcal{H}$ et $\text{prox}_{\tau f_2} x^* \in \mathcal{G}$ est un minimum de $f = f_1 + f_2$.

L'avantage de ce schéma repose sur les conditions d'utilisation, mais il suppose que les opérateurs proximaux associés à f_1 et f_2 soient simples (facilement évaluables). Notons que la formule se simplifie en prenant $(\beta_t)_t \equiv 1$, $(a_t)_t \equiv 0$ et $(b_t)_t \equiv 0$, conduisant à l'itération suivante (mise en œuvre dans l'algorithme 2.4) :

$$x_{t+1} = x_t/2 + \text{rprox}_{\nu f_1}(\text{rprox}_{\nu f_2}(x_t))/2. \tag{2.36}$$

Les suites $(a_t)_{t \in \mathbb{N}}$ et $(b_t)_{t \in \mathbb{N}}$ ont la même interprétation que dans le schéma explicite-implicite précédent. Nous pouvons donc utiliser des approximations pour le calcul des opérateurs proximaux, tant que les erreurs vérifient les conditions de sommabilité (i) – (ii) ci-dessus.

A notre connaissance, peu de résultats sur la vitesse de convergence de l'itération de DR sont connus. Chaux et al. [2009] ont montré, dans le cas où f_1 ou f_2 est fortement convexe, que la vitesse de convergence est linéaire sur les itérées.

Algorithme 2.4 : Schéma de Douglas-Rachford pour résoudre (P).

Initialisation : Choisir $x_0 \in \mathcal{H}$ et $\tau \in (0, +\infty)$.

Début

Pour $t = 0$ jusqu'à $N_{\text{ext}} - 1$ **faire**

1. Premier calcul proximal :

$$x_{t+1/2} = \text{prox}_{\tau f_2}(x_t) . \quad (2.37)$$

2. Deuxième calcul proximal :

$$x_{t+1} = x_t - x_{t+1/2} + \text{prox}_{\tau f_1}(2x_{t+1/2} - x_t) . \quad (2.38)$$

finpour

Fin

Sorties : Une solution du problème (P) : $x^* = \text{prox}_{\tau f_2} x_{N_{\text{ext}}}$.

2.2.4 L'opérateur proximal de $f \circ \mathbf{A}$

Le résultat qui suit est un résultat important dans le cadre de cette thèse. En effet, il permet d'exprimer l'opérateur proximal d'une fonction $f \in \Gamma_0(\mathcal{H})$ composée avec un opérateur affine borné $\mathbf{A} : \mathcal{H} \rightarrow \mathcal{H}'$, $x \mapsto \mathbf{F}x - y$, $y \in \mathcal{H}'$. Nous rappelons que $\mathbf{F} : \mathcal{H} \rightarrow \mathcal{H}'$ est un opérateur synthèse de trame associé à une trame de \mathcal{H}' si l'opérateur d'analyse \mathbf{F}^* satisfait la relation généralisée de Parseval avec les bornes inférieure et supérieure c_1 et c_2 :

$$c_1 \|x\|^2 \leq \|\mathbf{F}^*x\|^2 \leq c_2 \|x\|^2 \quad 0 < c_1 \leq c_2 < +\infty . \quad (2.39)$$

La trame est dite ajustée lorsque $c_1 = c_2 = c$, auquel cas $\mathbf{F}\mathbf{F}^* = c\mathbf{I}$.

Théorème 2.2.5 (Fadili et Starck [2009]).

Soit \mathbf{F} un opérateur linéaire borné tel que $\text{ri}(\text{dom}(f) \cap \text{Im}(\mathbf{A})) \neq \emptyset$. Alors $f \circ \mathbf{A} \in \Gamma_0(\mathcal{H}')$ et :

(i) \mathbf{F} est une trame ajustée. Alors,

$$\text{prox}_{f \circ \mathbf{A}}(x) = y + c^{-1} \mathbf{A}^* \text{prox}_{cf}(\mathbf{F}x - y) . \quad (2.40)$$

(ii) \mathbf{F} est une trame quelconque. Soit $\tau_t \in (0, 2/c_2)$. Nous définissons l'algorithme 2.5. Alors, $(u_t)_{t \in \mathbb{N}}$ converge vers \bar{u} et $(p_t)_{t \in \mathbb{N}}$ converge vers $\text{prox}_{f \circ \mathbf{A}} x = x - \mathbf{F}^* \bar{u}$. Plus précisément, ces deux suites convergent linéairement et la meilleure vitesse de convergence est atteinte pour $u_t \equiv 2/(c_1 + c_2)$:

$$\|p_t - \text{prox}_{f \circ \mathbf{A}}(x)\| \leq \sqrt{\frac{c_2}{c_1}} \left(\frac{c_2 - c_1}{c_2 + c_1} \right)^t \|p_0 - \text{prox}_{f \circ \mathbf{A}}(x)\| . \quad (2.41)$$

(iii) Si $c_1 = 0$, \mathbf{F} n'est pas une trame. En appliquant l'algorithme 2.5 avec $\tau_t \in (0, 2/c_2)$, alors, $(u_t)_{t \in \mathbb{N}}$ converge vers \bar{u} et $(p_t)_{t \in \mathbb{N}}$ converge vers $\text{prox}_{f \circ \mathbf{A}} x = x - \mathbf{F}^* \bar{u}$ avec une vitesse en $\mathcal{O}(1/t)$, c'est à dire qu'il existe $C > 0$ tel que :

$$\|p_t - \text{prox}_{f \circ \mathbf{A}}(x)\|^2 \leq \frac{C}{t}. \quad (2.42)$$

Une démonstration complète de ces résultats est disponible dans [Fadili et Starck, 2009]. Pour le cas où \mathbf{F} est une trame ajustée, une démonstration différente a été proposée par Combettes et Pesquet [2007].

Algorithme 2.5 : Itération explicite-implicite à un pas pour le calcul de l'opérateur proximal d'une fonction composée avec un opérateur affine.

Initialisation : Choisir $u_0 \in \mathbb{R}^n$, $p_0 = x - \mathbf{F}^* u_0$ et $\tau_t \in (0, 2/c_2)$.

Début

<p>Pour $t = 0$ jusqu'à $N_{\text{ext}} - 1$ faire</p>	$u_{t+1} = \tau_t(\mathbf{I} - \text{prox}_{f/\tau_t})(u_t/\tau_t + \mathbf{A}p_t),$	(2.43)
<p style="padding-left: 20px;">finpour</p>	$p_{t+1} = x - \mathbf{F}^* u_{t+1}.$	

Fin

Sorties : L'opérateur proximal de $f \circ \mathbf{A}$ en x : $p_{N_{\text{ext}}}$.

Lorsque que \mathbf{F} est une trame, la vitesse de convergence du schéma explicite-implicite à un pas dépend de la redondance de la trame. Plus la redondance est élevée, plus la convergence est lente. Plus précisément, le nombre d'itérations nécessaire pour obtenir une solution avec une précision ε à l'itération finale est en $\mathcal{O}\left(\frac{c_1}{c_2} \log \varepsilon^{-1}\right)$. Dans le cas général (iii) où \mathbf{F} n'est pas une trame, la vitesse de convergence vers l'opérateur proximal est seulement en $\mathcal{O}(1/t)$, donc nécessite au minimum $\mathcal{O}(1/\varepsilon)$ itérations pour atteindre la précision ε .

A la place de l'algorithme 2.5, nous pouvons utiliser une méthode à pas multiple inspiré de l'algorithme 2.3. Le schéma à pas multiple, alternative au précédent, est résumé par l'algorithme 2.6, avec $c_1 = 0$ si \mathbf{F} n'est pas une trame.

Ce schéma jouit des vitesses de convergences suivantes (voir [Fadili et Starck, 2009] pour une preuve). Supposons que $\tau \in (0, 2/c_2)$. Soit $u_0 \in \text{dom}(\mathbf{F}^*)$, et définissons $R = \|\bar{u} - u_0\|$. Si \mathbf{F} est une trame (non-ajustée), alors $\forall t \geq 1$:

$$\|p_t - \text{prox}_{f \circ \mathbf{A}} x\|^2 \leq 2c_2 R^2 \left(1 + \sqrt{\frac{c_1}{8c_2}}\right)^{-2(t-1)}. \quad (2.44)$$

Sinon, la vitesse est :

$$\|p_t - \text{prox}_{f \circ \mathbf{A}} x\|^2 \leq \frac{8c_2 R^2}{t^2}. \quad (2.45)$$

Algorithme 2.6 : Schéma itératif de Nesterov à pas multiple pour le calcul de l'opérateur proximal d'une fonction composée avec un opérateur affine.

Initialisation : Choisir $u_0 \in \text{dom}(\mathbf{F})^*$, $v_0 = 0$, ξ_0 , $\tau \in (0, 2/c_2)$.

Début

1. Poser $\rho_t = \tau(1 + c_1 v_0)$.
2. Premier calcul proximal :

$$q_t = v_t \left(\mathbf{I} - \text{prox}_{f/v_t} \right) \circ ((u_0 - \xi_t)/v_t) .$$

3. Poser $a_t = \left(\rho_t + \sqrt{\rho_t^2 + 4\rho_t v_t} \right) / 2$ et $w_t = \frac{v_t u_t + a_t q_t}{q_t + a_t}$
4. Deuxième calcul proximal :

$$u_{t+1} = \tau/2 (\mathbf{I} - \text{prox}_{2f/\tau}) \circ (2w_t/\tau + (\mathbf{F}(x - \mathbf{F}^* w_t) - y)) ,$$

$$\xi_{t+1} = \xi_t - a_t (\mathbf{F}(x - \mathbf{F}^* u_{t+1}) - y) .$$

5. Mise à jour : $v_{t+1} = v_t + a_t$.

Fin

Sorties : L'opérateur proximal de $f \circ \mathbf{A}$ en $x : x - \mathbf{F}^* u_{N_{\text{ext}}}$.

Pour le cas où \mathbf{F} n'est pas une trame, la vitesse de convergence de cette itération en $\mathcal{O}(1/t^2)$ est bien meilleure que celle proposée par l'algorithme 2.5. Même pour les autres cas, la convergence est plus rapide. Toutefois, une itération de l'algorithme 2.6 coûte deux fois plus en complexité temporelle qu'une de l'algorithme 2.5.

2.2.5 L'opérateur proximal de $f_1 + f_2 \circ \mathbf{A}$

Le résultat que nous exposons ci-après est particulièrement précieux pour la suite, notamment pour calculer l'opérateur proximal de la somme de deux fonctions, par exemple, l'une traduisant la parcimonie et l'autre une contrainte sur la solution.

Soient f_1 et $f_2 \in \Gamma_0(\mathcal{H})$, et $\mathbf{A} : \mathcal{H} \rightarrow \mathcal{H}'$ est un opérateur linéaire. Soit $f = f_1 + f_2 \circ \mathbf{A}$, nous posons les hypothèses suivantes :

- (i) $\text{Im}(\mathbf{A}) \neq \emptyset$.
- (ii) $0 \in \text{ri}(\text{dom}(f_1) - \mathbf{A} \text{dom}(f_2))$.
- (iii) Les opérateurs proximaux associés à f_1 et f_2 sont simples à calculer analytiquement.

Nous voulons résoudre la régularisation de Moreau-Yosida :

$$\inf_{y \in \mathcal{H}} \frac{1}{2} \|y - x\|^2 + f_1(y) + f_2 \circ \mathbf{A}(y) , \quad (2.46)$$

dont l'unique solution est l'opérateur proximal de f .

En utilisant la dualité de Fenchel-Rockafellar (lemme 2.1.21), nous prouvons le résultat suivant :

Lemme 2.2.6.

Sous les hypothèses (i)-(ii), le problème proximal d'optimisation primal (2.46) est équivalent au problème dual :

$$\inf_{y \in \mathcal{H}} \frac{1}{2} \|y - x\|^2 + f_1(y) + f_2 \circ \mathbf{A}(y) \iff - \min_{u \in \mathcal{H}'} \stackrel{(1)}{f_1^*}(-\mathbf{A}^*u + x) + f_2^*(u) . \quad (2.47)$$

La solution primale $y^* = \text{prox}_f(x)$ est calculée à partir de la solution duale u^* par

$$y^* = \text{prox}_{f_1}(-\mathbf{A}^*u^* + x) . \quad (2.48)$$

Démonstration. Nous supposons que les différentes hypothèses sont vérifiées. Considérons le problème d'optimisation suivant :

$$\inf_{y \in \mathcal{H}} \frac{1}{2} \|x - y\|^2 + f_1(y) + f_2 \circ \mathbf{A}(y) = \frac{1}{2} \|\cdot\| \overset{\dagger}{\vee} f(x) = \stackrel{(1)}{f}(x) , \quad (\text{Q})$$

où $\stackrel{(1)}{f}(x)$ est l'enveloppe de Moreau d'ordre 1 de f . Il y a nécessairement une solution unique par coercivité et convexité stricte : l'ensemble des solutions $\mathcal{G} = \{\text{prox}_f(x)\}$.

Soit $g_x(y) = \frac{1}{2} \|x - y\|^2 + f_1(y) \in \Gamma_0(\mathcal{H})$ ($\text{dom}(g_x) \neq \emptyset$ par hypothèse sur $\text{dom}(f_1)$). Sachant que $\text{dom}(f_1) = \text{dom}(g_x)$, nous avons $0 \in \text{ri}(\text{dom}(f_2) - \mathbf{A} \text{dom}(f_1)) = \text{ri}(\text{dom}(f_2) - \mathbf{A} \text{dom}(g_x))$. De ce fait, nous pouvons appliquer la dualité de Fenchel-Rockafellar (lemme 2.1.21) pour obtenir :

$$\inf_{y \in \mathcal{H}} g_x(y) + f_2 \circ \mathbf{A}(y) = - \min_{u \in \mathcal{H}'} g_x^*(-\mathbf{A}^*u) + f_2^*(u) , \quad (\text{R})$$

et la solution primale est telle que :

$$y^* \in \partial g_x^*(-\mathbf{A}^*u^*) .$$

Ceci revient à calculer la transformée de Legendre-Fenchel de g_x définie plus tôt. Par le lemme 2.1.14, la conjuguée d'une somme de fonctions est l'inf-convolution des conjuguées :

$$\forall v \in \mathcal{H}, g_x^*(v) = (\frac{1}{2} \|x - \cdot\|^2)^* \overset{\dagger}{\vee} f_1^*(v) = \inf_{w \in \mathcal{H}'} (\frac{1}{2} \|x - \cdot\|^2)^*(v - w) + f_1^*(w) .$$

Notons que $\text{dom}(\frac{1}{2} \|x - \cdot\|^2) = \mathcal{H}$, $\text{dom}(f_1) \neq \emptyset$ prouvant ainsi l'existence des conjuguées utilisées et de leur somme. Nous avons alors :

$$\begin{aligned} \left(\frac{1}{2} \|x - \cdot\|^2\right)^*(v) &= \frac{1}{2} \|v\|^2 + \langle v, x \rangle , \\ &= \frac{1}{2} \|v + x\|^2 - \frac{1}{2} \|x\|^2 , \end{aligned}$$

et

$$\begin{aligned} g_x^*(v) &= \inf_{w \in \mathcal{H}} \frac{1}{2} \|v - w + x\|^2 - \frac{1}{2} \|x\|^2 + f_1^*(w) , \\ &= -\frac{1}{2} \|x\|^2 + \underbrace{\inf_{w \in \mathcal{H}} \frac{1}{2} \|(v+x) - w\|^2 + f_1^*(w)}_{(1)f_1^*(v+x)} . \end{aligned}$$

Le problème (Q) devient alors :

$$\min_{u \in \mathcal{H}'} -\frac{1}{2} \|x\|^2 + (1)f_1^*(-\mathbf{A}^*u + x) + f_2^*(u) \iff \min_{u \in \mathcal{H}'} (1)f_1^*(-\mathbf{A}^*u + x) + f_2^*(u) ,$$

d'où (2.47).

Notons que la solution du problème dual n'est pas unique, sauf si f_1 est continûment différentiable et \mathbf{A} surjectif, ou f_2 est continûment différentiable. Dans ces cas, l'objectif dual est strictement convexe.

De plus, par calcul proximal (lemmes 2.1.26 et 2.1.27), nous avons :

$$\nabla^{(1)} f_1^* = \mathbf{I} - \text{prox}_{f_1^*} = \text{prox}_{f_1} , \quad (2.49)$$

d'où (2.48). □

En utilisant le schéma explicite-implicite, nous arrivons au résultat suivant :

Théorème 2.2.7.

Soit $0 < \inf_t \tau_t \leq \tau_t \leq \sup_t \tau_t < 2/\|\mathbf{A}\|^2$. Nous définissons la suite suivante :

$$u_{t+1} = \tau_t(\mathbf{I} - \text{prox}_{f_2/\tau_t})(u_t/\tau_t + \mathbf{A} \circ \text{prox}_{f_1}(-\mathbf{A}^*u_t + x)) . \quad (2.50)$$

Alors, u_t converge vers u^* et $y_t = \text{prox}_{f_1}(-\mathbf{A}^*u_t + x)$ vers $\text{prox}_f(x)$.

Démonstration. Nous rappelons que $(1)f_1^*$ est Fréchet-différentiable et à gradient 1-Lipschitz avec :

$$\nabla^{(1)} f_1^*(v) = v - \text{prox}_{f_1^*}(v) = \text{prox}_{f_1}(v) ,$$

et

$$\text{prox}_{f_2^*}(u) = u - \text{prox}_{f_2}(u) .$$

Il suffit alors d'appliquer un schéma explicite-implicite (théorème 2.2.1) pour résoudre le problème (Q). □

Le schéma explicite-implicite utilisé ici, peut être remplacé par un convergeant plus rapidement, comme celui proposé par Nesterov (section 2.2.2.2). A noter qu'un travail parallèle mais indépendant de Combettes et al. [2009] a montré un résultat similaire au nôtre avec quelques différences toutefois.

Le lemme suivant résume les vitesses de convergence lors de l'utilisation du schéma explicite-implicite (théorème 2.2.1) pour résoudre (2.47) :

Lemme 2.2.8.

Si le problème (Q) est résolu avec le schéma (2.50), alors :

- (i) si f_2 est à gradient Lipschitz, alors le schéma converge en $\mathcal{O}(1/t)$ sur les itérées duales et primales,
- (ii) si f_1 est à gradient Lipschitz, alors la convergence est en $\mathcal{O}(1/t)$ sur les itérées primales. Si de plus \mathbf{A} est surjectif, alors la convergence est linéaire sur les itérées duales et primales,
- (iii) si $\text{dom}(f_2^*)$ est borné, alors la convergence est en $\mathcal{O}(1/t)$ sur les itérées primales.

2.3 Formulation du problème d'optimisation

Soient $f_1 : \mathcal{H} \rightarrow (-\infty, +\infty]$, $f_2 : \mathcal{H} \rightarrow \mathbb{R}$ et $f_3 : \mathcal{H} \rightarrow \mathbb{R}$, trois fonctions dans $\Gamma_0(\mathcal{H})$. Notre objectif est de résoudre le problème :

$$\tilde{x} \in \arg \min_{x \in \mathcal{H}} f_1(x) + f_2(x) + f_3(x). \quad (\text{P3})$$

Nous rappelons que l'ensemble des solutions est noté \mathcal{G} .

Cette formulation en trois composantes permet de modéliser un cadre faisant intervenir un terme d'attache aux données soumis à deux termes de régularisations ou de contraintes. Une telle écriture couvre un large spectre de problématiques rencontrées en traitement du signal dont, par exemple, certaines formulations de problèmes inverses.

2.4 Caractérisation

La caractérisation de la solution du problème est donnée par des résultats classiques en optimisation :

Proposition 2.4.1.

1. *Existence* : (P3) admet au moins une solution si la fonction objectif est coercive, i.e. $\mathcal{G} \neq \emptyset$,
2. *Unicité* : (P3) admet une solution unique si au moins l'une des trois fonctions f_1 , f_2 ou f_3 est strictement convexe.

2.5 Premier type d'algorithme

Nous proposons de construire un algorithme itératif basé sur les schémas discutés en section 2.2 pour résoudre le problème (P3). Pour cela, nous commençons par décomposer en deux parties : d'un côté, une fonction seule, par exemple f_1 , de l'autre, la somme des deux fonctions

restantes; dans cette occurrence, en suivant l'exemple, $f_2 + f_3$. L'idée directrice est d'utiliser un premier schéma pour former l'itération *extérieure*, et un deuxième schéma *intérieur*, chargé de calculer l'opérateur proximal de $f_2 + f_3$, est utilisé à chaque itération. Selon la décomposition, plusieurs combinaisons sont possibles : celles-ci sont présentées dans la table 2.1. L'algorithme 2.7 offre un cadre général d'utilisation des deux itérations en supposant que le schéma extérieur soit itéré N_{ext} fois et celui intérieur N_{int} . Nous détaillons les deux parties de ce type d'algorithme.

Itération Extérieure	Itération Intérieure
FB (une des fonctions à gradient Lipschitz)	DR FB Nesterov
DR	FB DR Nesterov

TAB. 2.1: Les différentes combinaisons de schémas pour résoudre le problème (P3) : FB pour le schéma explicite-implicite à un pas (section 2.2.2.1), DR pour le schéma de Douglas-Rachford (section 2.2.3) et Nesterov pour le schéma explicite-implicite à pas multiple (section 2.2.2.2).

Algorithme 2.7 : Schéma général du premier type d'algorithme pour résoudre (P3).

Initialisation : Choisir un $x_0 \in \mathcal{H}$.

Début

1. Schéma extérieur.

Pour $t = 0$ jusqu'à $N_{\text{ext}} - 1$ **faire**

 2. Itération du schéma extérieur.

 3. Schéma intérieur.

Pour $k = 0$ jusqu'à $N_{\text{int}} - 1$ **faire**

 4. Itération du schéma intérieur.

finpour

 5. Mise à jour de l'estimée : x_{t+1} .

finpour

Fin

Sorties : Une solution du problème (P3) : $x^* = x_{N_{\text{ext}}}$.

2.5.1 Itération extérieure

Le schéma extérieur est celui dont dépendra la convergence de l'algorithme vers une solution. L'idée est d'utiliser celui le plus adapté selon les propriétés des fonctions impliquées dans la fonction objectif. Par exemple, supposons f_1 différentiable sur \mathcal{H} avec un gradient continue et κ -Lipschitz, $\kappa \in (0, +\infty)$. Alors, avec la décomposition f_1 d'un côté et $f_2 + f_3$ de l'autre, nous pouvons utiliser un schéma explicite-implicite à pas simple ou multiple (section 2.2.2) comme itération extérieure. Celle intérieure aura alors la charge de calculer $\text{prox}_{f_2+f_3}$.

En revanche, pour la décomposition f_2 et $f_1 + f_3$, le schéma de Douglas-Rachford (section 2.2.3) peut gérer l'itération extérieure. Le choix dépend de trois points : les propriétés des fonctions, la vitesse de convergence sur les itérées recherchée et le calcul des opérateurs proximaux associés aux fonctions.

2.5.2 Itération intérieure

Cette itération a pour rôle de calculer l'opérateur proximal d'une somme de fonctions. Toutefois, un tel opérateur est rarement exprimable analytiquement. Pour être général, nous appelons les deux fonctions formant la somme g_1 et g_2 , puis reformulons le calcul de l'opérateur proximal en problème d'optimisation :

$$\text{prox}_{g_1+g_2} y = \arg \min_{x \in \mathcal{H}} g_1(x) + g_2(x) + \frac{1}{2} \|x - y\|^2 . \quad (2.51)$$

Selon les propriétés des fonctions, nous choisissons un schéma explicite-implicite (si l'une des fonctions est différentiable à gradient Lipschitz) ou celui de Douglas-Rachford (si les opérateurs proximaux de chacune des fonctions sont facilement calculables).

Remarque 2.5.1.

D'autres schémas itératifs, profitant de la décomposition d'un problème en somme de fonctions, existent. Par exemple, Bauschke et Combettes [2008] en proposent un en reprenant les idées fondatrices de l'algorithme de Dykstra. Autre remarque, si g_1 ou g_2 est une fonction simple composée par un opérateur affine, nous pouvons utiliser les résultats de la section 2.2.5 pour calculer l'opérateur associé à $g_1 + g_2$.

Toutefois, il reste à calculer $\text{prox}_{g_1 + \frac{1}{2}\|x-y\|^2}$ (ou $\text{prox}_{g_2 + \frac{1}{2}\|x-y\|^2}$) en utilisant uniquement prox_{g_1} (ou prox_{g_2}). Celui-ci est donné par la proposition 2.1.25(1) : $\text{prox}_{g_1 + \frac{1}{2}\|x-y\|^2} = \text{prox}_{g_1/2} \left(\frac{\cdot+y}{2} \right)$.

Le paramètre le plus important du schéma intérieur est le nombre d'itérations (N_{int}) pour atteindre la solution. Remarquons que si l'une des fonctions g_1 ou g_2 est dérivable à gradient Lipschitz continue, nous avons tout intérêt à utiliser le schéma à pas multiple (section 2.2.2.2) afin de profiter d'une meilleure vitesse de convergence.

2.5.3 Robustesse aux erreurs

Après un certain nombre d'itérations intérieures, nous obtenons toujours une approximation $\widetilde{\text{prox}}_{f_2+f_3}$ de l'opérateur proximal $\text{prox}_{f_2+f_3}$ telle que :

$$\widetilde{\text{prox}}_{f_2+f_3} = \text{prox}_{f_2+f_3} + b_t, \quad (2.52)$$

avec b_t l'erreur commise à l'itération extérieure t en appliquant le schéma intérieur avec N_{int} itérations. Cette erreur est inévitable et peut nuire à la convergence globale de l'algorithme.

Selon les théorèmes 2.2.1 et 2.2.4 concernant le schéma explicite-implicite à un pas et celui de Douglas-Rachford, si l'erreur est sommable (i.e. $\sum_t \|b_t\| < +\infty$), alors les deux convergent vers une solution du problème (P3). Toutefois, à notre connaissance, un tel résultat de robustesse n'existe pas pour l'algorithme de Nesterov (section 2.2.2.2).

Dans le cas où la vitesse de convergence sur les itérées de l'itération intérieure est connue (typiquement dans le cas de l'algorithme composé du DR à l'extérieur et du FB ou Nesterov à l'intérieur), nous pouvons exhiber le nombre minimal d'itérations N_{int} nécessaire pour avoir une erreur sommable. Nous supposons qu'à chaque itération t , nous faisons converger le schéma intérieur jusqu'à obtenir une tolérance (i.e. l'erreur ℓ_2 entre deux itérées) $\eta_{\text{int},t}$. Soit $r(t)$ la vitesse de l'itération intérieure sur les itérées avec $r(t)$ strictement décroissante et bijective de \mathbb{N} vers $[0, +\infty)$. Le théorème suivant exprime la condition pour avoir une erreur sommable :

Théorème 2.5.2.

Nous supposons que la suite $(\eta_{\text{int},t})_{t \in \mathbb{N}}$ est sommable. Alors, si l'itération intérieure est appliquée à chaque étape t avec $N_{\text{int},t} \geq r^{-1}(\eta_{\text{int},t}/2)$ itérations, l'itération extérieure converge.

La preuve de ce théorème est une extension de celle dans [Fadili et Peyré, 2009, Proposition 3].

2.6 Second type d'algorithme

Nous avons supposé que notre fonction objectif se décompose en une somme de trois fonctions. Le premier type d'algorithme permet de résoudre le problème d'optimisation en profitant de cette séparation. Toutefois, nous pouvons rapidement constater que si la fonction objectif est la somme de plus de trois fonctions, nous sommes incapables d'en tirer pleinement parti. Au contraire, le raisonnement précédent nous conduit à construire une série de récursions emboîtées les unes dans les autres et donc produit à terme un schéma inutilisable car trop complexe. D'où le besoin de présenter un deuxième type d'algorithme.

Soit $f \in \Gamma_0(\mathcal{H})$ telle que $f = \sum_{i=1}^M f_i$, $\forall i, f_i \in \Gamma_0(\mathcal{H})$. Nous considérons le problème général suivant :

$$\arg \min_{x \in \mathcal{H}} f(x) = \arg \min_{x \in \mathcal{H}} \sum_{i=1}^M f_i(x). \quad (\text{PN})$$

Celui-ci généralise les formulations précédentes (P) et (P3). Plusieurs méthodes [Eckstein et Svaiter, 2007; Combettes et Pesquet, 2008] ont été développées pour résoudre (PN), exploitant la décomposition récursive d'opérateurs monotones maximaux.

L'algorithme 2.8 présente une version de celui proposé par Combettes et Pesquet [2008], elle-même construite à partir du schéma de Douglas-Rachford (section 2.2.3) étendu aux espaces produits. Elle demande uniquement de calculer les opérateurs proximaux associés à chacune des fonctions f_i .

Algorithme 2.8 : Schéma de décomposition pour résoudre (PN).

Initialisation : Choisir $(y_{0,i})_{1 \leq i \leq M} \in \mathcal{H}^M$, $\mu \in]0, +\infty[$ et poser $x_0 = \sum_{i=1}^M x_{0,i}/M$.

Début

Pour $t = 0$ jusqu'à $N_{\text{ext}} - 1$ **faire**

1. Calcul des opérateurs proximaux :

Pour $i = 1$ jusqu'à M **faire**

$$z_{t,i} = \text{prox}_{\mu f_i/M} y_{t,i} . \quad (2.53)$$

finpour

2. Mise à jour de l'estimée :

$$x_{t+1} = \sum_{i=1}^M \frac{z_{t,i}}{M} . \quad (2.54)$$

3. Mise à jour des variables auxiliaires :

Pour $i = 1$ jusqu'à M **faire**

$$y_{t+1,i} = y_{t,i} + 2x_{t+1} - x_t - z_{t,i} . \quad (2.55)$$

finpour

finpour

Fin

Sorties : Une solution du problème (PN) : $x^* = x_{N_{\text{ext}}}$.

Remarque 2.6.1.

Lorsque les fonctions f_i sont toutes des indicatrices d'ensembles convexes fermés dont l'intersection est non-vidée, ce schéma possède des ressemblances avec la projection moyennée.

Le côté général de cette méthode permet de résoudre de nombreux problèmes d'optimisation. Evidemment, les opérateurs proximaux des différentes fonctions impliquées sont supposés facilement calculables (fonctions simples). Toutefois, il n'existe à notre connaissance aucun résultat sur sa vitesse de convergence, contrairement à celles du premier type où dans certains cas les vitesses sont disponibles.

Comparée au schéma de Douglas-Rachford, la suite $(x_t)_{t \in \mathbb{N}}$ dans cet algorithme est minimisante (et non son image par un opérateur proximal). De plus, Combettes et Pesquet [2008] prouvent qu'il offre une même robustesse aux erreurs d'approximations sur le calcul des opérateurs, à la condition que ces erreurs soient sommables.

2.7 Conclusion

Le cadre théorique que nous avons exposé permet de résoudre un grand nombre de problèmes d'optimisation, à la condition, pour chaque fonction, que l'opérateur proximal soit facilement calculable, ou que la fonction soit dérivable avec un gradient Lipschitz. Dans le cadre des problèmes inverses, l'expressivité des opérateurs proximaux permet d'envisager de nouvelles contraintes et de les combiner.

Toutefois, la variabilité et flexibilité annoncées avec le premier type d'algorithme sont plus ou moins perdues lorsque la fonction objectif se décompose en plus de trois fonctions. Néanmoins, nous pouvons envisager une version récursive de la méthode, mais le bénéfice d'une telle procédure est incertain par rapport au deuxième type d'algorithme. Ce dernier, bien que plus général, requiert la connaissance des opérateurs proximaux des fonctions impliquées.

Une application du cadre est présentée dans le chapitre 4, pour obtenir l'estimateur du maximum a posteriori dans le cas d'une déconvolution sous bruit de Poisson. Celui-ci est formulé par un problème d'optimisation convexe, non-linéaire et non-différentiable. La fonction objectif est composée d'une somme de trois fonctions, elle entre donc dans le cadre. Selon la gestion du bruit (stabilisé ou non), nous utiliserons le premier ou le deuxième type d'algorithme, présentant ainsi un exemple pour chaque type.

3 Microscopie confocale de fluorescence

But there were simply too many gaps ; nothing was known about how brains learn. Nevertheless, it occurred to me, you might be able to figure that out - if only you knew how those brain cells were connected to each other. Then you could attempt some of what is now called "reverse engineering" - to guess what those circuit's components do from knowing both what the circuits do and how their parts are connected.

Marvin Minsky, *Memoir on Inventing the Confocal Scanning Microscope*

Dans le cadre de cette thèse, les images issues d'un microscope confocal de fluorescence à balayage laser doivent être restaurées. Cette étape est cruciale pour une pleine exploitation de leurs contenus. Cependant, la conception d'un algorithme de restauration demande de connaître le système d'acquisition et de localiser les différentes sources de dégradation. Cette connaissance permet de modéliser l'équation de formation de l'image, et ainsi nous pouvons construire un estimateur de l'image réelle à partir des observations. Dans ce chapitre, nous exposons le principe du microscope confocal avec des exemples d'utilisation. Ensuite, nous présentons les différentes sources de déformations et leur modélisation lorsqu'elle existe. Enfin, nous proposons un modèle général de formation de l'image pour un microscope confocal.

3.1 Principe

3.1.1 Historique

Le principe du microscope confocal a été inventé par Marvin Minsky en 1955 alors jeune post-doctorant à l'université de Harvard. Il y conçut un système optique confocal avec un balayage par déplacement de la platine du microscope, et un système de sténopés pour enlever la lumière issue hors du plan focal. L'idée de déplacer l'objet à observer pour effectuer le balayage présente l'énorme avantage de conserver ainsi le réglage du système optique. Ce système ingénieux permet d'obtenir des images constituées essentiellement de la lumière provenant du plan focal. L'auteur brevette son idée en 1957 [Minsky, 1957] (accepté en 1961) mais, faute de sources lumineuses suffisamment puissantes pour une pleine exploitation de l'appareil, celui-ci provoque peu de remous. Il faudra attendre l'arrivée du laser au début des années 60 et l'accélération

de la recherche sur le cerveau, pour voir le développement du système. Vers la fin des années 70, le microscope confocal à balayage laser voit enfin le jour dans le cadre de l'acquisition d'images 3D en biologie, grâce aux travaux issus du groupe de génie électrique d'Oxford. En 1988, Minsky proposa un mémoire [Minsky, 1988] relatant les circonstances ayant conduit à cette invention, maintenant abondamment utilisée.

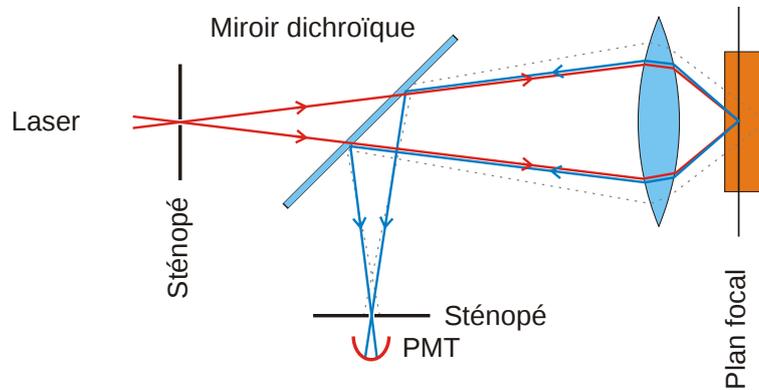


FIG. 3.1: Microscope confocal à balayage laser dans une configuration classique.

3.1.2 Principe de fonctionnement

La figure 3.1 expose le principe du microscope confocal à balayage laser : premièrement le laser émet un rayon dont la longueur d'onde excite un révélateur déposé dans l'échantillon à observer. Une fois celui-ci excité, il émet un rayonnement, sur une autre longueur d'onde, réfléchi par un miroir dichroïque vers un photo-multiplicateur (PMT) au travers d'un sténopé. Ce dernier permet d'éliminer une partie de la lumière issue des autres plans, mais en contrepartie diminue fortement la luminosité.

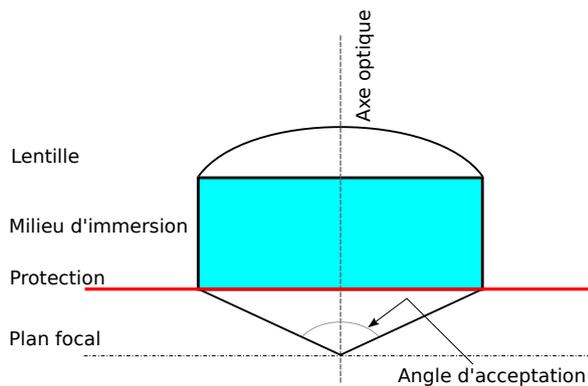


FIG. 3.2: Détails de l'objectif d'un microscope confocal.

La performance d'un microscope confocal dépend fortement de son ouverture numérique (ou NA pour *Numerical Aperture*). La figure 3.2 montre la composition de l'objectif d'un microscope confocal. L'ouverture numérique est définie par la formule $NA = n_{\text{ref}} \sin(\omega)$, n_{ref} étant l'indice de réfraction du milieu d'immersion et ω le demi-angle d'acceptation.

L'ouverture numérique permet de définir les approximations des résolutions latérales (i.e. le long du plan focal) et axiales (i.e. en profondeur) maximales du système (ou limites de Nyquist), soient les plus petites distances visibles (résolue) par le système d'acquisition :

$$\Delta_{\text{latéral}} = \frac{l_{ex}}{8NA}, \quad \Delta_{\text{axial}} = \frac{l_{ex}}{4 \left(n_{\text{ref}} - \sqrt{n_{\text{ref}}^2 - NA^2} \right)}, \quad (3.1)$$

avec n_{ref} l'indice de réfraction et l_{ex} la longueur d'onde d'excitation. Ces distances supposent que les longueurs d'ondes d'excitation et d'émissions soient approximativement les mêmes.

Par exemple, avec la protéine à fluorescence verte (GFP) comme révélateur, nous avons $l_{ex} = 488\text{nm}$ (longueur d'onde excitatrice). Donc, pour un microscope confocal avec $NA = 1.4$ et $n_{\text{ref}} = 1.515$ (indice de réfraction de l'huile), nous obtenons $\Delta_{\text{axial}} = 43.6\text{nm}$ et $\Delta_{\text{latéral}} = 130\text{nm}$. Ces bornes sont toutefois théoriques ; en pratique les hautes fréquences sont rapidement perdues près de ces bornes et une majoration de ces valeurs de 60% donne une meilleure approximation de la limite réelle.

3.1.3 Modes d'utilisation du microscope confocal

Le microscope confocal à balayage laser peut évidemment être utilisé pour obtenir des séries de plan 2D d'un échantillon, fournissant ainsi une observation 3D de l'échantillon, mais avec des données manquantes entre chaque plan à cause du pas d'échantillonnage dans la direction axiale. Cependant, d'autres modalités d'utilisation ont été développées, basées sur une utilisation approfondie des caractéristiques des révélateurs utilisés. L'élément fondamental de la microscopie par fluorescence est le fluorophore : la molécule fluorescente ajoutée à l'échantillon et qui généralement se fixe sur l'élément (protéine, molécule, etc) qui est observé. Cet élément fait office de révélateur dans le cadre de la microscopie de fluorescence. Il est à la base de trois techniques couramment utilisées dans la communauté biomédicale :

FRET (*Fluorescence Residence Energy Transfer*) est une technique consistant à attacher un fluorophore « donneur » sur une protéine et un fluorophore « accepteur » sur une autre. Si la distance entre les deux protéines est suffisamment faible, alors elles s'échangent leurs électrons provoquant une fluorescence. Cette technique permet, par exemple, de repérer les zones où deux protéines cohabitent.

FRAP (*Fluorescence Recovery After Photobleaching*) est une technique qui propose premièrement de blanchir (et donc de rendre non-fluorescent) le fluorophore avec le laser, puis

d'observer le retour de la fluorescence. Elle permet d'obtenir des informations à propos de la diffusion et de la mobilité du fluorophore et du support de fixation du fluorophore.

FLIM (*Fluorescence Lifetime Imaging*) est une technique reposant sur le temps d'excitation du fluorophore. Ce temps est dépendant de l'environnement du fluorophore, en effet, certains ions l'allongent ou le réduisent. L'observation du temps de fluorescence permet donc d'obtenir des informations sur l'environnement du fluorophore dans certaines zones de l'échantillon.

Notons que d'autres méthodes d'acquisitions avec un microscope confocal existent. Elles sont souvent basées sur des propriétés du fluorophore utilisé [Pawley, 2005] selon l'information recherchée (temps de recouvrement, zone de recouvrement, etc).

3.2 Sources de dégradations

Le schéma de la figure 3.1 montre les différents éléments participant à l'élaboration de l'image. Chaque étape ajoute des aberrations provenant d'une construction qui n'est pas parfaite, ou de phénomènes physiques difficiles à éviter. Nous proposons de faire un tour sur les éléments constituant le microscope confocal et de discuter des dégradations qu'ils apportent.

3.2.1 Système optique

Dans le cas de la microscopie de fluorescence, la lumière émise par le laser traverse le système optique puis frappe l'échantillon, avec une précision réduite par le flou introduit par l'optique. Ensuite la lumière émise par le révélateur traverse le système optique avant d'arriver au capteur. Le système optique est constitué d'un miroir, une lentille et un sténopé (voir figure 3.1). La lentille et le miroir sont susceptibles d'ajouter du flou dans l'image ainsi que des aberrations sphériques provenant des changements d'indices entre l'échantillon et le milieu d'immersion. Le sténopé, pour sa part, réduit la zone de provenance de la lumière. La figure 3.3 présente l'erreur commise sur l'angle d'acceptation en présence d'un changement de milieu.

Un des points recherché consiste à comprendre comment les rayons traversent la lentille avant de converger vers le plan focal. Généralement deux types d'optiques sont considérés : les systèmes paraxiaux et les non-paraxiaux. Paraxial signifie que la majorité des rayons arrivant sur la lentille de l'objectif forment un angle faible avec l'axe optique (permettant d'effectuer les approximations usuelles : $\sin(\omega) \approx \omega$ et $\cos(\omega) \approx 1$). La figure 3.4(a) montre le modèle de diffraction avec une telle lentille ; pour calculer le champ de diffraction au point P , les outils classiques de l'optique suffisent. Dans le cas non-paraxial (figure 3.4(b)), le calcul du champ demande d'autres outils comme le modèle de diffraction de Fresnel-Kirchhoff.

L'objet décrivant et modélisant (dont les aberrations et défauts) la formation de l'image par

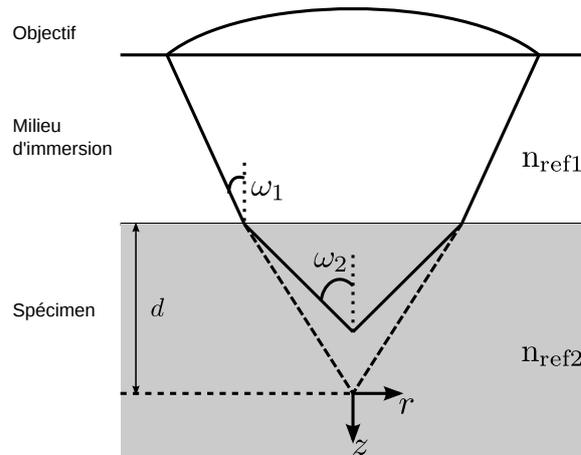
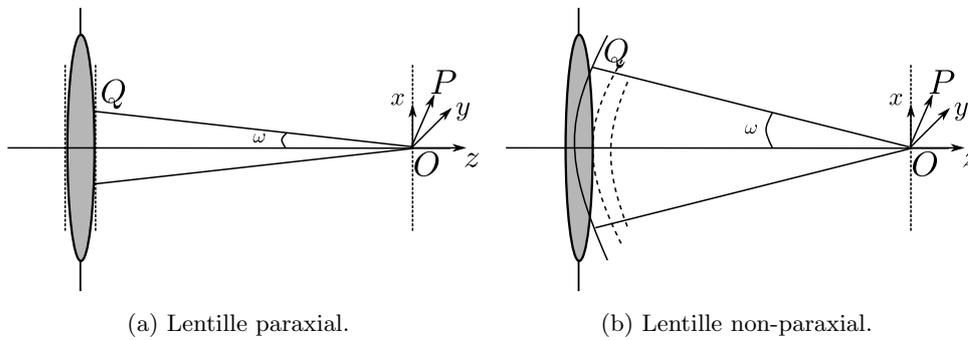


FIG. 3.3: Le changement d'indice de réfraction entre deux milieux est une source d'erreurs. Celui-ci est susceptible de provoquer des aberrations sphériques au niveau de la zone de transition entre les deux milieux.



(a) Lentille paraxial.

(b) Lentille non-paraxial.

FIG. 3.4: Les deux modèles de diffraction selon le type de lentille : (a) système paraxial, l'angle ω est proche de 0, (b) système non-paraxial, $\omega \gg 0$. Dans les deux cas, nous recherchons la formule du champ de diffraction au niveau du point P . Le point O repère l'intersection entre le plan focal et l'axe optique et Q est le début d'un rayon sortant de la lentille allant vers le plan.

le système optique s'appelle la fonction d'étalement du point (PSF). Cette fonction exprime la vue par le système optique d'un point évanescent (infiniment petit). Pratiquement, une telle fonction est dépendante d'un grand nombre de paramètres et quasiment unique pour chaque appareil. Les travaux de Gibson et Lanni [1991] ont conduit à un modèle général pour la distribution d_{conf} de l'amplitude du rayonnement des microscopes confocaux, celle-ci peut être exprimée par l'intégrale scalaire de Debye [Gu, 2000] :

$$d_{conf}(x, y, z; l) = c_0 \int_0^\omega \sqrt{\cos \theta} J_0(k\rho \sin \theta) \exp(-ikz \cos \theta) \sin \theta d\theta, \quad (3.2)$$

avec c_0 une constante complexe, J_0 la fonction de Bessel de première espèce et d'ordre 0, $\rho = \sqrt{x^2 + y^2}$, ω le demi-angle d'acceptation de l'objectif et $k = n_{ref} \frac{2\pi}{l_{ex}}$ (l_{ex} étant la longueur d'onde d'excitation et n_{ref} l'indice du milieu d'immersion). La PSF pour un microscope confocal est donc :

$$PSF_{conf}(x, y, z; l) = |d_{conf}(x, y, z; l)|^2 \int_{\{x_1^2 + y_1^2 \leq R^2\}} |d_{conf}(x - x_1, y - y_1, z; l)|^2 dx_1 dy_1, \quad (3.3)$$

R étant le rayon d'ouverture du sténopé.

Cette formule étant assez complexe à calculer, des approximations ont été proposées utilisant au maximum les informations disponibles sur l'appareil d'acquisition. Nous présentons ici deux formules d'approximations différentes selon la qualité recherchée : la méthode d'approximation de Zhang et al. [2007a] par des gaussiennes et celle de Booth et Wilson [2001] basée sur les polynômes de Zernike.

3.2.1.1 Approximation par des gaussiennes

Zhang et al. [2007a] proposent d'approcher la formule (3.3) en utilisant une contrainte ℓ_∞ pour imposer une égalité entre les pics et une contrainte de conservation de l'énergie. Deux formules sont alors obtenues selon le type de PSF modélisée (2D ou 3D) :

$$PSF_{gauss,2D}(x, y) = \exp\left(-\frac{x^2 + y^2}{2\sigma_{xy}^2}\right), \quad (3.4)$$

$$PSF_{gauss,3D}(x, y, z) = \exp\left(-\frac{x^2 + y^2}{2\sigma_{xy}^2} - \frac{z^2}{2\sigma_z^2}\right). \quad (3.5)$$

Les paramètres $\sigma_{(\cdot)}$ sont dépendants de l'appareil d'acquisition et en modélisent les caractéristiques. Les différentes valeurs possibles sont résumées dans la table 3.1, les formules se découpent en deux parties : paraxiale et non-paraxiale. Lorsqu'un microscope présente une ouverture numérique assez grande ($NA \geq 1.3$), la distribution de l'intensité lumineuse présente une forte composante non-paraxiale [Gu, 2000] ; nous utiliserons alors les formules non-paraxiales pour un tel microscope. En revanche, pour une ouverture numérique faible, la lumière est essentiellement paraxiale.

La figure 3.5(a) montre un exemple de PSF pour un microscope confocal classique : $NA = 1.4$, $l_{ex} = 488\text{nm}$ (longueur d'onde excitatrice de la protéine GFP), $l_{em} = 509\text{nm}$ (émission de la protéine GFP), $R = 500\text{nm}$ et $n_{ref} = 1.515$. La tâche formée au centre de l'image est aussi connue sous le nom de « disque d'Airy ».

Zhang et al. [2007a] précisent que cette approximation est assez proche de la « vérité terrain » dans le cas 2D, mais tend à être grossière en 3D. Une des raisons de la mauvaise approximation en 3D est la présence non-négligeable d'aberrations sphériques produites par les changements d'indices des différents milieux traversés par la lumière. Or celles-ci sont ignorées dans les formules d'approximations proposées ici. Toutefois, ce modèle est facile à utiliser et demande un minimum d'informations sur l'appareil d'acquisition. La méthode décrite ci-après prend en compte ces aberrations en 3D.

3.2.1.2 Approximation par des polynômes de Zernike

Booth et Wilson [2001] proposent de modéliser les aberrations sphériques en utilisant les polynômes de Zernike. Ceci a deux intérêts : 1) avoir un modèle plus précis de la PSF, 2) les coefficients des polynômes peuvent être mesurés directement à partir du système d'acquisition avec du matériel spécial [Pawley, 2005]. La PSF est donnée par :

$$\text{PSF}_{\text{zern}}(v, u, d) = \left| \int_0^1 L(\rho, d) \exp(iu\rho^2/2) J_0(\rho v) \rho d\rho \right|^2, \quad (3.6)$$

u, v, d étant les coordonnées optiques : $v = \frac{2\pi n_{ref} r}{l_{ex}} \sin \omega$, $u = \frac{8\pi n_{ref} z}{l_{ex}} \sin^2\left(\frac{\omega}{2}\right)$, r et z sont respectivement le déplacement radial et axial à partir du focus nominal et l_{ex} la longueur d'onde d'excitation. La fonction lentille L est donnée par :

$$L(\rho, d) = \exp(i\Upsilon(\rho, d)). \quad (3.7)$$

La fonction d'aberration Υ modélise les caractéristiques des différentes déformations provenant de l'objectif. Υ est construite par une suite infinie, mais convergeant rapidement, de polynômes de Zernike d'ordre 0 (donc radialement symétrique) :

$$\Upsilon(\rho, d) = d \frac{2\pi}{l_{ex}} n_{ref1} \sin \omega_1 \left[\sum_{\substack{j=0 \\ j \text{ pair}}}^{\infty} A_{j,0} Z_{j,0}(\rho) \right], \quad (3.8)$$

$$Z_{j,0}(\rho) = \sqrt{j+1} \sum_{s=0}^{j/2} \frac{(-1)^s (j-s)!}{s! [(j/2-s)!]^2} \rho^{j-2s} \quad \text{pour } j \text{ pair}, \quad (3.9)$$

$$A_{j,0} = B_j(\omega_1) - B_j(\omega_2), \quad (3.10)$$

$$B_j(\eta) = \left[1 - \left(\frac{j-1}{j+3} \right) \tan^4(\eta/2) \right] \frac{\tan^{j-1}(\eta/2)}{2(j-1)\sqrt{j+1}}, \quad (3.11)$$

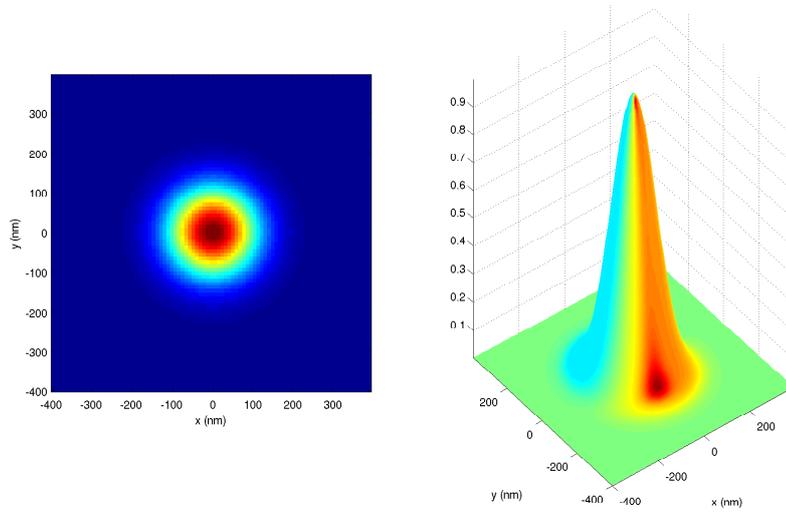
Paraxial 2D, 3D	$\sigma_{xy} = \sqrt{2} \left[\frac{c_1^2}{R^2} + \frac{4c_2 J_0(c_2) J_1(c_2) - 8J_1^2(c_2)}{R^2(J_0^2(c_2) + J_1^2(c_2) - 1)} \right]^{-\frac{1}{2}}$
Non-paraxial 2D, 3D	$\sigma_{xy} = \sqrt{2} \left[\frac{2g_{xy}(k_{em}, n_{ref}, \omega)^4 \left[\exp\left(\frac{R^2}{2g_{xy}(k_{em}, n_{ref}, \omega)^2}\right) - 1 \right] + R^2 g(k_{ex}, n_{ref}, \omega)^2}{g(k_{ex}, n_{ref}, \omega)^2 g(k_{em}, n_{ref}, \omega)^4 \left[\exp\left(\frac{R^2}{2g_{xy}(k_{em}, n_{ref}, \omega)^2}\right) - 1 \right]} \right]^{-\frac{1}{2}}$
Paraxial 3D	$\sigma_z = 2\sqrt{6} \left[\frac{c_1^2 NA^2}{R^2 n_{ref}^2} - \frac{48c_2^2 (J_0^2(c_2) + J_1^2(c_2)) - 192J_1^2(c_2)}{n_{ref}^2 k_{em}^2 R^4 (J_0^2(c_2) + J_1^2(c_2) - 1)} \right]^{-\frac{1}{2}}$
Non-paraxial 3D	$\sigma_z = \frac{g_z(k_{ex}, n_{ref}, \omega) g_z(k_{em}, n_{ref}, \omega)}{(g_z(k_{ex}, n_{ref}, \omega)^2 + g_z(k_{em}, n_{ref}, \omega)^2)^{1/2}}$

avec $k_{em} = \frac{2\pi}{l_{em}}$, $k_{ex} = \frac{2\pi}{l_{ex}}$, $c_1 = k_{ex} R NA$, $c_2 = k_{em} R NA$,

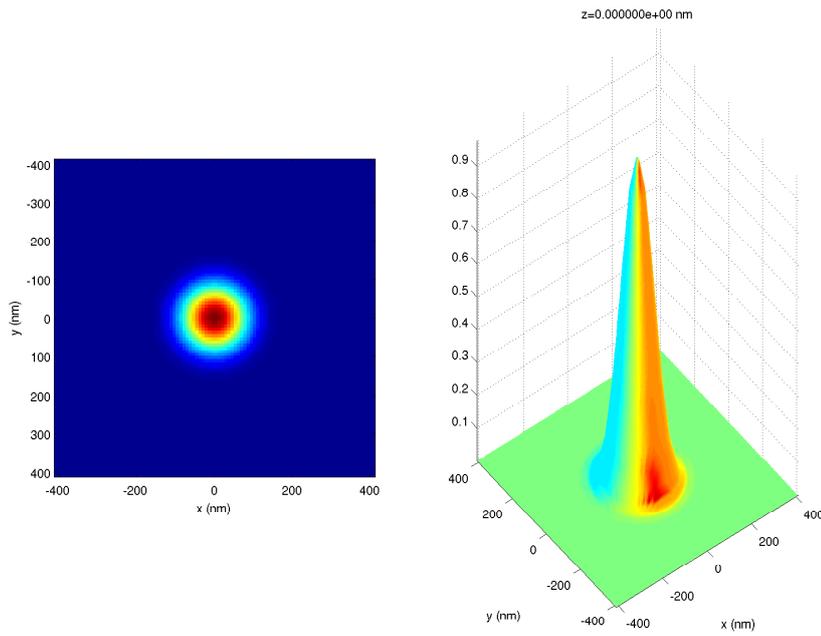
$$g_{xy}(k, n_{ref}, \omega) = \frac{1}{n_{ref} k} \left[\frac{4 - 7 \cos^{3/2} \omega + 3 \cos^{7/2} \omega}{7(1 - \cos^{3/2} \omega)} \right]^{-\frac{1}{2}} \text{ et}$$

$$g_z(k, n_{ref}, \omega) = \frac{5\sqrt{7}(1 - \cos^{3/2} \omega)}{\sqrt{6} n_{ref} k (4 \cos^5 \omega - 25 \cos^{7/2} \omega + 42 \cos^{5/2} \omega - 25 \cos^{3/2} \omega + 4)^{1/2}}.$$

TAB. 3.1: Les différents $\sigma_{(\cdot)}$ selon le type d'appareil pour l'approximation gaussienne de la PSF d'un microscope confocal. Les paramètres sont l_{ex} et l_{em} respectivement les longueurs d'ondes d'excitation et d'émission, n_{ref} l'indice de réfraction du milieu d'immersion, NA l'ouverture numérique, R le rayon d'ouverture du sténopé et ω le demi-angle d'acceptation. La fonction J_0 est la fonction de Bessel de première espèce d'ordre 0 et J_1 d'ordre 1.



(a) Approximation gaussienne



(b) Approximation avec les polynômes de Zernike

FIG. 3.5: La fonction d'étalement du point au niveau du plan focal telle que donnée par les deux méthodes d'approximation présentées : (a) avec des gaussiennes [Zhang et al., 2007a] et (b) avec des polynômes de Zernike [Booth et Wilson, 2001].

$n_{\text{ref}1}$ est l'indice de réfraction du milieu d'immersion de l'objectif, d est la profondeur du focus nominal (c'est à dire la profondeur du focus dans l'échantillon), ω_1 et ω_2 sont les demi-angles d'acceptations tels que $n_{\text{ref}1} \sin \omega_1 = n_{\text{ref}2} \sin \omega_2$ (voir figure 3.3), $n_{\text{ref}2}$ étant l'indice de réfraction du milieu contenant l'échantillon. La correction apportée par la fonction Υ peut être améliorée en ne calculant les polynômes de Zernike que pour certaines valeurs de j [Booth et Wilson, 2001]. Pour le microscope confocal, la formule finale de la fonction d'étalement du point est donc :

$$\text{PSF}_{\text{zern,conf}}(v, u, d) = \text{PSF}_{\text{zern}}(v, u, d) \text{PSF}_{\text{zern}}(v/\beta, u/\beta, d), \quad \beta = \frac{l_{ex}}{l_{em}}. \quad (3.12)$$

La figure 3.5(b) montre l'étalement du point sur le plan focal pour les mêmes valeurs que précédemment, avec $n_{\text{ref}2} = 1.6$ (indice de réfraction de l'eau). Notons que l'étalement est moins fort par rapport à l'approximation gaussienne et le pic plus prononcé.

3.2.2 Bilan des bruits

Nous venons de proposer plusieurs modèles pour les déformations provenant du système optique du microscope confocal. Nous allons maintenant nous intéresser aux autres sources de dégradations. Celles-ci sont essentiellement situées au niveau des capteurs et de l'échantillon lui-même.

La plus importante source de bruit provient du bruit de comptage au niveau du photomultiplicateur (PMT). Ce phénomène physique est un bruit intrinsèque provenant de la nature quantique des photons. Il implique un processus de comptage dont la loi de probabilité est celle de Poisson. De plus, lorsqu'un photon est émis par le fluorophore, un autre photon peut être émis selon une probabilité qui suit aussi une loi de Poisson. Ce second phénomène est appelé bruit d'émission secondaire. Ces deux processus étant indépendants, pour un compteur c , nous avons $c \sim \mathcal{P}(p_1 + p_2)$ où p_1 est le nombre moyen de photons émis provenant de l'émission primaire et p_2 de l'émission secondaire. Cette source de bruit est considérée par les experts comme la source prépondérante de fluctuations aléatoires dégradant le signal [Pawley, 2005]. La seule méthode pour réduire le bruit est d'avoir un temps d'acquisition important afin d'obtenir des taux de comptage élevés. Cependant, ce n'est pas toujours possible et conduit en plus à une augmentation du bruit de fond.

Au niveau du capteur, d'autres sources de bruits existent :

- le *courant noir* provoquant l'apparition d'électrons indésirables au niveau des compteurs, généralement négligeable et réduit en refroidissant le capteur ;
- l'environnement peut présenter des radiations susceptibles d'être prises en compte par les capteurs et provoquant alors un *bruit de fond* ;

- le *bruit de lecture* au niveau du gain du photomultiplicateur, dû à l'amplification de bruits électriques lors de la lecture, ajoute un bruit généralement supposé gaussien mais souvent négligeable.

Le fluorophore utilisé dans les préparations présente des phénomènes qui induisent aussi du bruit dans l'image :

- le *photoblanchiment* lorsque le fluorophore perd son pouvoir fluorescent après une excitation ;
- certains éléments dans les préparations sont naturellement fluorescents et émettent aussi des photons provoquant l'apparition d'objets ou artefacts dans l'image, ce phénomène est appelé : *autofluorescence* ;
- la distribution du révélateur dans l'échantillon peut être hétérogène provoquant des changements d'intensité compliquant l'analyse de l'image ;
- la lumière peut être absorbée ou dispersée par le milieu de la préparation provoquant une perte d'information proportionnelle à l'enfoncement du plan focal dans l'échantillon.

Les déformations provenant de l'échantillon et du fluorophore sont difficiles, voire impossibles, à enlever. Il convient cependant d'éviter d'augmenter l'importance de ces phénomènes, par exemple, en réglant correctement le laser pour éviter de dégrader l'échantillon.

3.3 Conclusion : modèle de formation de l'image

A partir des développements précédents, nous pouvons conclure que les principales sources de dégradations sont : 1) les aberrations optiques, 2) le bruit de Poisson au niveau des capteurs et 3) le bruit de lecture (généralement modélisé par un bruit blanc, additif et gaussien d'écart type σ). Les autres sources sont habituellement négligées car leurs influences sur le résultat restent faibles. Soient y l'image observée, x l'image réelle, g_0 le gain de l'amplificateur et h la fonction d'étalement du point, l'équation de formation de l'image est alors :

$$y \sim g_0 \mathcal{P}(x \otimes h) + \mathcal{N}(m, \sigma^2), \quad (3.13)$$

où \otimes est l'opérateur de convolution circulaire, m et σ sont respectivement la moyenne et l'écart type du bruit de lecture. Ce mélange de bruits peut être stabilisé en appliquant la transformée stabilisatrice de la variance d'Anscombe généralisée [Starck et al., 1998] :

$$A_{\text{gen}}(t) = 2\sqrt{g_0 t + \frac{3}{8}g_0^2 + \sigma^2 - g_0 m} . \quad (3.14)$$

Les paramètres (g_0, m, σ^2) peuvent être estimés à partir de l'image observée y , par exemple en traçant la variance locale en fonction de la moyenne dont la pente est g_0 :

$$\begin{aligned} \text{Var}[y[i]] &= g_0^2(x \otimes h)[i] + \sigma^2 , \\ g_0\mathbb{E}[y[i]] &= g_0^2(x \otimes h)[i] + g_0m , \text{ donc} \\ \text{Var}[y[i]] &= g_0\mathbb{E}[y[i]] + (\sigma^2 - g_0m) . \end{aligned} \tag{3.15}$$

De même, m et σ^2 peuvent être estimés à partir des zones sombres de l'image. La transformation (3.14) permet de se ramener à un cadre non-linéaire gaussien :

$$\forall 1 \leq i \leq n, \quad A_{\text{gen}}(y[i]) = A_{\text{gen}}((x \otimes h)[i]) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, 1) . \tag{3.16}$$

Toutefois, d'un point de vue rigoureux, cette équation n'est valable qu'asymptotiquement. Expérimentalement, elle est vérifiée à partir d'un taux de comptage suffisamment grand.

Souvent, le bruit de lecture est négligeable devant les autres. Une fois les valeurs de l'image divisées par le gain, nous pouvons simplifier le modèle de formation de l'image :

$$y/g_0 \sim \mathcal{P}(x \otimes h). \tag{3.17}$$

Ce modèle est celui communément utilisé pour la formation des images par un microscope confocal, et celui sur lequel nous nous appuyons dans le reste de cette thèse.

4 Application à la déconvolution sous bruit de Poisson

[...] to see things for the first time, in other words, to contemplate them, putting aside what we learned from books, false descriptions and common platitudes, has great importance in scientific research. We have to free our minds of prejudices and other people's images, and to have the firm intention to see and judge for ourselves.

Ramón y Cajal [1899], translated in english by García-López et al. [2007].

Les images acquises par un microscope confocal sont dégradées à la fois par un flou et du bruit. Nous avons discuté le modèle de formation de ces images au chapitre précédent. Celui-ci, (3.17), est celui généralement traité dans le cadre de la déconvolution sous bruit de Poisson. Lors du chapitre 1, nous avons présenté deux approches pour gérer correctement le bruit : soit il est stabilisé afin de retourner vers un cadre gaussien, soit la loi de Poisson est directement considérée dans la vraisemblance additionnelle (attache aux données).

En se plaçant dans un cadre bayésien, nous proposons d'utiliser l'estimateur MAP pour concevoir nos algorithmes. Le terme de fidélité aux données est dépendant de la gestion du bruit. Avec la stabilisation, nous retrouvons le cadre gaussien avec un terme non-linéaire possédant un gradient Lipschitz. En revanche, sans la stabilisation, le log de vraisemblance de la loi de Poisson conduit à une fonction dont le gradient n'est pas Lipschitz. Afin de réduire l'ensemble solution, le terme de régularisation est construit en exploitant la parcimonie et la diversité morphologique présente dans l'image à restaurer. Cet a priori permet de décrire la grande variété des structures composant l'image et ainsi les préserver lors de la restauration. Le problème d'optimisation est ensuite résolu en nous plaçant dans le cadre décrit dans le chapitre 2.

Le chapitre suivant se décompose en quatre parties : la première introduit la notion de représentations parcimonieuses et le concept de diversité morphologique, la deuxième présente un algorithme de déconvolution avec stabilisation de la variance pour gérer le bruit, celui-ci est ensuite traité sans stabilisation dans la troisième. Enfin, la dernière partie présente le protocole expérimental ainsi qu'une discussion sur les résultats obtenus.

4.1 Représentations parcimonieuses

Avant d'introduire les notions de représentation parcimonieuse et de diversité morphologique, nous présentons une série de définitions et terminologies utilisées dans le domaine. Durant tout ce chapitre, nous considérons une image $x \in \mathcal{H}$ de taille $\sqrt{n} \times \sqrt{n}$ réarrangée en un vecteur dans \mathbb{R}^n .

4.1.1 Définitions

Définition 4.1.1 (Atome).

Un atome est un signal élémentaire de représentation d'un signal ou d'une image.

Les exemples d'atomes les plus connus sont les sinusoides (avec la transformée de Fourier), les ondelettes, les monômes, etc.

Les atomes sont réunis afin de former des dictionnaires :

Définition 4.1.2 (Dictionnaire).

Un dictionnaire est une collection indexée d'atomes $(\varphi_i)_{i \in \mathcal{I}}$, avec \mathcal{I} un ensemble dénombrable (de cardinalité L). L'interprétation de l'index dépend du dictionnaire : fréquence pour le dictionnaire de Fourier, position pour celui de Dirac, position-échelle pour les ondelettes, etc.

Dans le cadre de cette thèse, un dictionnaire est une matrice $n \times L$ dont les colonnes sont les atomes, et les atomes sont considérés comme des vecteurs colonnes. Lorsque le dictionnaire possède plus de colonnes que de lignes, $L > n$, il est dit redondant. Ce cas conduit à la configuration où $x = \Phi\alpha$ est un système linéaire d'équations sous-déterminé.

Ainsi, l'image x peut être écrite comme la superposition d'atomes élémentaires φ_γ paramétrisés par $\gamma \in \mathcal{I}$ selon le modèle linéaire génératif suivant :

$$x = \sum_{\gamma \in \mathcal{I}} \alpha_\gamma \varphi_\gamma = \Phi\alpha, \quad |\mathcal{I}| = L \geq n. \quad (4.1)$$

Φ est donc le dictionnaire, c'est à dire la matrice $n \times L$ dont les colonnes sont les atomes élémentaires $(\varphi_\gamma)_{\gamma \in \mathcal{I}}$ tous normalisés à une norme ℓ_2 unitaire. Les dictionnaires forment généralement des opérateurs de trames (section 2.2.4).

Pour un dictionnaire donné Φ , nous différencions donc deux opérations :

Définition 4.1.3 (Analyse).

*L'opération d'analyse est celle qui associe à un signal x , un vecteur de coefficient α attaché aux atomes : $\alpha = \Phi^*x$.*

Définition 4.1.4 (Synthèse).

L'opération de synthèse est celle de reconstruction d'un signal x à partir des coefficients en superposant les atomes : $x = \Phi\alpha$.

L'analyse et la synthèse sont deux opérations linéaires bien différentes. Dans le cas redondant, Φ n'est pas inversible et la reconstruction n'est pas unique. La transformée dans le sens explicite (analyse) est alors définie par une matrice $\mathbf{T} = \Phi^* \in \mathbb{R}^{L \times n}$ avec $L \geq n$; cette matrice n'est donc pas nécessairement carrée. Dans le cas d'une base orthogonale, la transformée inverse (synthèse) est directement $\Phi = \mathbf{T}^*$. En revanche, si Φ est une trame ajustée, alors l'opérateur de trame satisfait $\Phi\Phi^* = c\mathbf{I}$, avec $c > 0$ la constante de la trame ajustée. La reconstruction par la pseudo-inverse \mathbf{T}^+ pour les trames ajustées n'est autre que Φ à une constante près : $\mathbf{T}^+ = c^{-1}\Phi$.

4.1.2 Pourquoi la parcimonie ?

Par parcimonieux, nous entendons rechercher une bonne représentation de x avec seulement quelques coefficients significatifs. Depuis une dizaine d'années, la parcimonie s'est placée comme un concept central de part sa large palette d'application en traitement du signal (restauration, extraction de caractéristiques, séparation de sources, etc). Elle a été longtemps une notion attractive grâce à ses propriétés théoriques et pratiques dans beaucoup de domaines des mathématiques appliquées (par exemples, l'analyse harmonique, l'estimation statistique, etc).

Récemment, des chercheurs de divers horizons ont préconisé l'utilisation de représentations redondantes de signaux. De telles représentations diffèrent des traditionnelles bases orthonormées car elles offrent un plus grand éventail d'éléments génératifs (i.e. les atomes). Bien sûr, l'attrait de ces représentations redondantes tient à leur capacité pour représenter *économiquement* une vaste classe de signaux. Potentiellement, cette capacité à représenter se traduit par davantage de flexibilité pour représenter un signal et d'adaptabilité à son contenu morphologique, et donc introduit plus d'efficacité dans les diverses méthodes classiques en traitements du signal (restauration, séparation, compression et estimation). Les neurosciences ont aussi souligné le rôle important de la redondance. En effet, le système visuel des mammifères semblent avoir besoin de représentations redondantes [Field, 1999; Hyvärinen et Hoyer, 2001; Olshausen et Field, 1996a; Simoncelli et Olshausen, 2001]. Dans cette configuration, la redondance peut conduire à un codage plus efficace et parcimonieux.

4.1.3 Redondance et diversité morphologique

Le contenu d'une image est souvent complexe et il n'y a pas de transformée unique qui soit optimale pour représenter efficacement toutes les caractéristiques présentes. Par exemple, la transformée de Fourier est meilleure pour obtenir une représentation parcimonieuse de textures globalement oscillantes, quand les ondelettes sont plus performantes avec des singularités isolées. Même si nous limitons notre classe de transformées aux ondelettes, nous devons décider entre la transformée en starlets qui produit de bon résultats pour des objets isotropes

(comme des cellules dans les images biomédicales, ou des étoiles et galaxies dans les images astronomiques), et la transformée en ondelettes orthogonales qui est adaptée pour des images à variations bornées [Cohen et al., 1999].

Si nous ne restreignons pas à des dictionnaires fixés reliés aux transformées implicites rapides comme les dictionnaires de Fourier ou les ondelettes, nous pouvons même concevoir un très large dictionnaire contenant beaucoup de formes différentes pour représenter les données efficacement. Selon Olshausen et Field [1996b], nous pouvons même pousser l'idée un peu plus loin, en requérant que le dictionnaire ne soit pas fixé mais plutôt appris à partir des données pour rendre parcimonieux un ensemble d'images caractéristiques (ou *patches*). Ce problème d'apprentissage de dictionnaire correspond à trouver une factorisation matricielle parcimonieuse et a été étudié par plusieurs auteurs [Field, 1999; Olshausen et Field, 1996a; Simoncelli et Olshausen, 2001; Lewicki et Sejnowski, 2000; Kreutz-Delgado et al., 2003; Aharon et al., 2006; Peyré et al., 2007]. Dans le reste de cette thèse, nous nous restreignons à des dictionnaires fixés avec des transformées rapides.

Le concept de *diversité morphologique* introduit un nouveau cadre de modélisation de données qui nous permet d'utiliser, à la fois, les représentations parcimonieuses et des algorithmes rapides exploitant la structure du dictionnaire. La diversité morphologique suppose que les données x peuvent être modélisées comme la somme de K composantes x_l qui sont *morphologiquement* différentes :

$$x = \sum_{l=1}^K x_l, \quad (4.2)$$

où chaque composante x_l est parcimonieuse dans un dictionnaire donné Φ_l qui est associé à des opérateurs implicites rapides d'analyse et de synthèse. Chaque x_l est appelé une *composante morphologique*. Par exemple, en combinant les dictionnaires de Fourier et des ondelettes, nous pouvons correctement représenter des signaux contenant des éléments stationnaires et/ou localisés. Grâce aux récentes avancées dans l'analyse harmonique moderne, de nombreux systèmes redondants, comme les ondelettes non-décimées, curvelets, contourlets, etc, ont prouvé leur efficacité pour une représentation parcimonieuse des images.

Le dictionnaire est construit en choisissant une ou plusieurs transformées. Le choix d'un dictionnaire approprié est une étape clef pour avoir une bonne représentation parcimonieuse, et donc une bonne restauration. Lorsque les transformées sont réunies pour former un dictionnaire, elles doivent être choisies pour que chacune conduise à une représentation parcimonieuse d'une partie de l'image qu'elle est sensée servir au mieux. Nous pouvons choisir les ondelettes pour les images lisses avec des singularités isotropes [Mallat, 1998, Section 9.3], les curvelets pour représenter des images C^2 au delà des contours C^2 par morceaux [Candès et Donoho, 1999; Candès et al., 2005], les *wave atoms* ou la transformée en cosinus discrets locaux pour représenter des textures localement oscillantes [Demanet et Ying, 2007; Mallat, 1998], etc. Ce

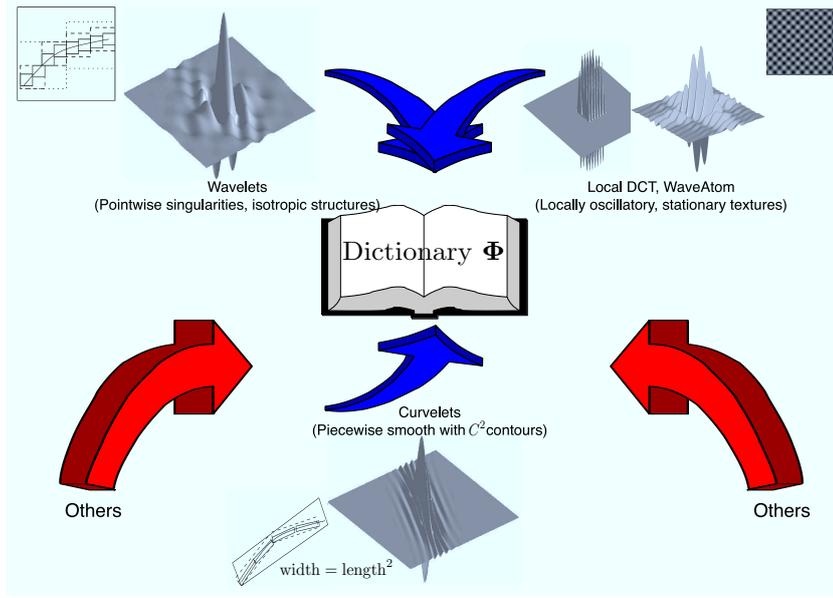


FIG. 4.1: Illustration du concept de diversité morphologique et le choix du dictionnaire qu'il implique.

processus de construction du dictionnaire est résumé graphiquement en figure 4.1.

4.2 Formulation du problème d'optimisation

En adoptant un cadre bayésien et l'estimateur usuel du maximum a posteriori (section 1.2.1), notre but est alors de minimiser la fonction objectif suivante dans le domaine des coefficients :

$$\arg \min_{\alpha \in \mathbb{R}^L} J(\alpha),$$

$$J : \alpha \mapsto \underbrace{F \circ \mathbf{H} \circ \Phi(\alpha)}_{f_1(\alpha)} + \underbrace{\gamma \sum_{i=1}^L \psi(\alpha[i])}_{f_2(\alpha)} + \underbrace{\iota_{\mathcal{C}} \circ \Phi(\alpha)}_{f_3(\alpha)}, \quad (\mathbf{P}_{F,\gamma,\psi})$$

où la fonction de pénalisation ψ est choisie de manière à promouvoir la parcimonie (voir section 2.1.2.2), $\gamma > 0$ est un paramètre de régularisation et $\iota_{\mathcal{C}}$ est la fonction indicatrice d'un ensemble convexe fermé \mathcal{C} . Nous supposons implicitement que les $(\alpha[i])_{1 \leq i \leq L}$ sont indépendants et identiquement distribués avec une densité gibbsienne $\propto e^{-\gamma\psi(\alpha[i])}$. Dans notre cas, \mathcal{C} est l'orthant positif. Le rôle du terme $\iota_{\mathcal{C}} \circ \Phi$ est d'imposer une contrainte de positivité sur l'image restaurée car les intensités poissonniennes sont positives par nature. Nous définissons aussi l'ensemble $\mathcal{C}' = \{\alpha \mid \Phi\alpha \in \mathcal{C}\}$, donc $\iota_{\mathcal{C}'} = \iota_{\mathcal{C}} \circ \Phi$. L'ensemble des solutions de ce problème est noté \mathcal{G} .

Le terme d'attache aux données est dépendant d'une fonction F , reflétant la vraisemblance déduite du modèle de bruit issu du modèle de formation de l'image. Nous proposons deux méthodes de restauration : la première repose sur une stabilisation du bruit conduisant à un cadre gaussien, mais *non-linéaire*, la deuxième consiste à formuler la vraisemblance originelle liée au bruit de Poisson sans avoir à stabiliser les observations. Nous décrivons les algorithmes itératifs pour résoudre les fonctionnelles obtenues en utilisant les schémas de décompositions d'opérateurs monotones maximaux du chapitre 2. Puis, nous discuterons des avantages et inconvénients de chacune des approches.

4.2.1 Avec stabilisation

Il est souvent intéressant de pouvoir transformer un problème en un autre plus simple à résoudre, ainsi que nous l'avons présenté dans l'état de l'art (section 1.2.4.5). Nous proposons de prendre en compte correctement la non-linéarité transformée stabilisatrice de la variance d'Anscombe [1948] dont l'équation est :

$$\forall i, 1 \leq i \leq n, \quad z_i = 2\sqrt{y[i] + \frac{3}{8}} = 2\sqrt{(h \otimes x)[i] + \frac{3}{8}} + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, 1), \quad (4.3)$$

où ε est un bruit blanc gaussien de variance unitaire. D'un point de vue rigoureux, cette équation doit être comprise dans le sens asymptotique. Cependant, nous supposons que pour des valeurs de taux de comptages $(h \otimes x)[i]$ suffisamment élevées, i.e. (4.3) est vraie. Cette affirmation peut être vérifiée graphiquement en comparant les vraisemblances de la loi de Poisson et de la stabilisation. La figure 4.2 montre que plus le taux de comptage est élevé, plus les courbes sont proches et les minima des deux fonctions coïncident. Notons par ailleurs que les constantes 2 et $3/8$ dans (4.3) peuvent être modifiées pour améliorer le comportement asymptotique de la stabilisation, et notre raisonnement est assez général pour être vrai pour d'autres choix de constantes de stabilisation. Dans [Chaux et al., 2009] une autre stabilisation basée sur une extension quadratique a été proposée.

L'expression (4.3) dit que z est non-linéairement lié à x . En considérant l'anti-log de la vraisemblance associé à (4.3), le terme d'attache aux données devient donc :

$$f_1 : \alpha \mapsto F_{\text{stab}} \circ \mathbf{H} \circ \Phi(\alpha), \text{ avec} \\ F_{\text{stab}} : \eta \in \mathbb{R}^n \mapsto \sum_{i=1}^n f_{\text{stab}}(\eta[i]), \quad f_{\text{stab}}(\eta[i]) = \frac{1}{2} \left(z[i] - 2\sqrt{\eta[i] + \frac{3}{8}} \right)^2. \quad (4.4)$$

A partir du problème $(P_{F_{\text{stab}}, \gamma, \psi})$, nous pouvons déduire :

Proposition 4.2.1.

- (i) f_1 est une fonction convexe. Elle est strictement convexe si Φ est une base orthonormale et $\ker(\mathbf{H}) = \emptyset$ (le spectre de la fonction d'étalement du point ne s'annule pas dans la bande de Nyquist).

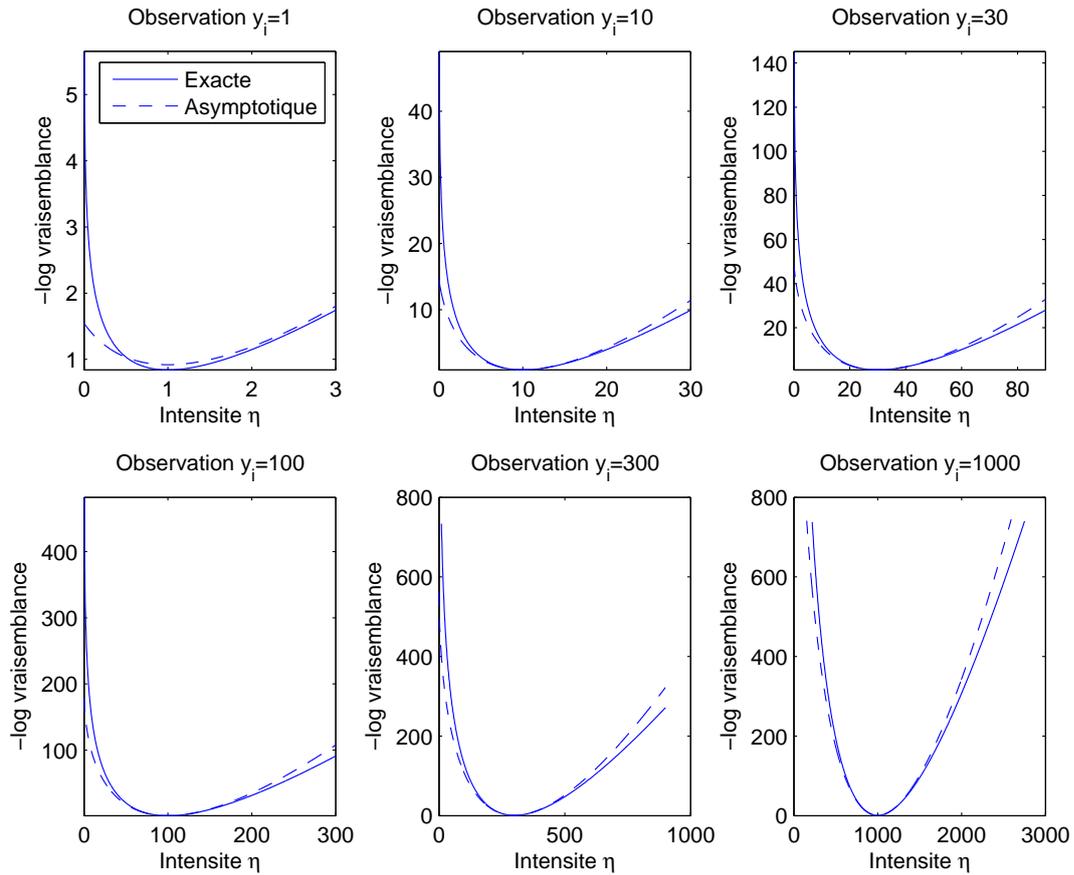


FIG. 4.2: Plus le taux de comptage est élevé, plus la vraisemblance approximée par la transformée d'Anscombe est proche de celle de la loi de Poisson.

(ii) Le gradient de f_1 est

$$\nabla f_1(\alpha) = \Phi^* \circ \mathbf{H}^* \circ \nabla F_{\text{stab}} \circ \mathbf{H} \circ \Phi(\alpha), \quad (4.5)$$

avec

$$\nabla F_{\text{stab}}(\eta) = \left(\frac{-z[i]}{\sqrt{\eta[i] + 3/8}} + 2 \right)_{1 \leq i \leq n}. \quad (4.6)$$

(iii) f_1 est continûment différentiable avec un gradient κ -Lipschitz avec :

$$\kappa \leq \left(\frac{2}{3}\right)^{3/2} 4c \|\mathbf{H}\|^2 \|z\|_\infty < +\infty, \quad (4.7)$$

où c est la borne supérieure de la trame associée à Φ .

(iv) $(\mathbf{P}_{F_{\text{stab}}, \gamma, \psi})$ est un cas particulier du problème (P3).

Démonstration.

(i) f_1 est convexe, car Φ et \mathbf{H} sont des opérateurs linéaires bornés et F_{stab} est convexe.

(ii) Le calcul du gradient de f_1 est direct à partir de (4.4).

(iii) Pour tout $\alpha, \alpha' \in \mathcal{H}$, nous avons,

$$\|\nabla f_1(\alpha) - \nabla f_1(\alpha')\| \leq \|\Phi\| \|\mathbf{H}\| \|\nabla F \circ \mathbf{H} \circ \Phi(\alpha) - \nabla F \circ \mathbf{H} \circ \Phi(\alpha')\|. \quad (4.8)$$

La fonction $-\frac{z_i}{\sqrt{\eta_i + 3/8}} + 2$ est croissance composante par composante sur $[0, +\infty)$ avec une dérivée uniformément bornée supérieurement par $\frac{z_i}{2} (8/3)^{3/2}$. Ainsi,

$$\begin{aligned} \|\nabla F \circ \mathbf{H} \circ \Phi(\alpha) - \nabla F \circ \mathbf{H} \circ \Phi(\alpha')\| &\leq \left(\frac{8}{3}\right)^{\frac{3}{2}} \frac{\|z\|_\infty}{2} \|\mathbf{H} \circ \Phi(\alpha) - \mathbf{H} \circ \Phi(\alpha')\| \\ &\leq \left(\frac{8}{3}\right)^{\frac{3}{2}} \frac{\|z\|_\infty}{2} \|\Phi\| \|\mathbf{H}\| \|\alpha - \alpha'\|. \end{aligned} \quad (4.9)$$

Sachant que $\|\Phi\|^2 = \|\Phi\Phi^*\| \leq c$ pour une trame, et z est bornée puisque nous supposons $y \in \ell_\infty(\{0, \dots, n-1\})$, nous concluons que ∇f_1 est Lipschitz continue avec une constante bornée par (4.7). □

4.2.1.1 Caractérisation de la solution

Comme J est convexe et coercive dans ce cas, nous avons :

Proposition 4.2.2.

1. *Existence* : $(\mathbf{P}_{F_{\text{stab}}, \gamma, \psi})$ possède au moins une solution, c'est à dire $\mathcal{G} \neq \emptyset$.

2. *Unicité* : $(P_{F_{\text{stab},\gamma,\psi}})$ possède une unique solution si Φ est une base orthonormée et que $\ker(\mathbf{H}) = \emptyset$, ou si ψ est strictement convexe.

Démonstration. L'existence de la solution est directe car J est coercive. Si Φ est une base orthonormée et $\ker(\mathbf{H}) = \emptyset$, alors f_1 est strictement convexe et par construction J l'est aussi ce qui conduit à l'assertion d'unicité. De même, si ψ est strictement convexe, alors f_2 l'est ainsi que J . \square

4.2.2 Sans stabilisation

Sans stabilisation, nous utilisons directement l'équation de dégradation dans le cas du bruit de Poisson :

$$y[i] \sim \mathcal{P}((h \otimes x)[i]) . \quad (4.10)$$

Nous rappelons la pdf de la loi de Poisson pour une intensité λ :

$$p(k) = e^{-\lambda} \frac{\lambda^k}{k!} . \quad (4.11)$$

Par conséquent, la vraisemblance est :

$$p(y|x) = \prod_i \frac{((\mathbf{H}x)[i])^{y[i]} \exp(-(\mathbf{H}x)[i])}{y[i]!} , \quad (4.12)$$

et le log de vraisemblance associé est :

$$\ell_{\text{poisson}}(y|x) = \sum_i (y[i] \log((\mathbf{H}x)[i]) - (\mathbf{H}x)[i] - \log(y[i]!)) , \quad (4.13)$$

ces formules sont étendues pour $y = 0$ en prenant comme convention $0! = 1$. L'anti-log de vraisemblance, nous conduit au terme d'attache aux données suivant :

$$f_1 : \alpha \mapsto F_{\text{poisson}} \circ \mathbf{H} \circ \Phi(\alpha), \text{ avec} \quad (4.14)$$

$$F_{\text{poisson}} : \eta \in \mathbb{R}^n \mapsto \sum_{i=1}^n f_{\text{poisson}}(\eta[i]),$$

$$\text{si } y[i] > 0, \quad f_{\text{poisson}}(\eta[i]) = \begin{cases} -y[i] \log(\eta[i]) + \eta[i] & \text{si } \eta[i] > 0, \\ +\infty & \text{sinon,} \end{cases}$$

$$\text{si } y[i] = 0, \quad f_{\text{poisson}}(\eta[i]) = \begin{cases} \eta[i] & \text{si } \eta[i] \in [0, +\infty], \\ +\infty & \text{sinon.} \end{cases}$$

Nous déduisons de $(P_{F_{\text{poisson},\gamma,\psi}})$ la proposition suivante :

Proposition 4.2.3.

(i) f_1 est une fonction convexe. Elle est strictement convexe si Φ est une base orthonormale et $\ker(\mathbf{H}) = \emptyset$.

(ii) Le gradient de f_1 est

$$\nabla f_1(\alpha) = \Phi^* \circ \mathbf{H}^* \circ \nabla F_{\text{poisson}} \circ \mathbf{H} \circ \Phi(\alpha), \quad (4.15)$$

avec

$$\begin{aligned} \nabla F_{\text{poisson}}(\eta) &= (g(\eta[i]))_{1 \leq i \leq n}, \\ \text{si } y[i] > 0, \quad g(\eta[i]) &= \begin{cases} 1 - \frac{y[i]}{\eta[i]} & \text{si } \eta[i] > 0, \\ +\infty & \text{sinon.} \end{cases} \\ \text{si } y[i] = 0, \quad g(\eta[i]) &= \begin{cases} 1 & \text{si } \eta[i] \geq 0, \\ +\infty & \text{sinon.} \end{cases} \end{aligned} \quad (4.16)$$

(iii) $(P_{F_{\text{poisson}}, \gamma, \psi})$ est un cas particulier du problème (P3).

La preuve est similaire à celle de la proposition 4.2.1.

Une remarque importante s'impose différenciant F_{poisson} de F_{stab} :

Remarque 4.2.4.

En effet, à l'inverse de F_{stab} , F_{poisson} n'est pas à gradient Lipschitz continu. Donc nous ne pouvons pas utiliser de schémas explicite-implicite pour résoudre $(P_{F_{\text{poisson}}, \gamma, \psi})$.

4.2.2.1 Caractérisation de la solution

J est convexe et coercive dans ce cas, nous en déduisons :

Proposition 4.2.5.

1. Existence : $(P_{F_{\text{poisson}}, \gamma, \psi})$ possède au moins une solution, c'est à dire $\mathcal{G} \neq \emptyset$.
2. Unicité : $(P_{F_{\text{poisson}}, \gamma, \psi})$ possède une unique solution si Φ est une base orthonormée et que $\ker(\mathbf{H}) = \emptyset$, ou si ψ est strictement convexe.

La preuve est similaire à celle de la proposition 4.2.2.

4.3 Premier type d'algorithme : cas stabilisé

4.3.1 Algorithme

Nous proposons d'utiliser les schémas explicite-implicite proposés dans la section 2.5 pour résoudre le problème dans le cas stabilisé. Nous avons démontré que le gradient de f_1 est Lipschitz continu et $f_2, f_3 \in \Gamma_0(\mathbb{R}^L)$, le schéma est donc adapté à ce cas. De plus, les opérateurs proximaux des fonctions composant f_2 et f_3 sont analytiquement calculables :

- (1) $\text{prox}_{\gamma\Psi}$ est donné par le lemme 2.1.29, par exemple si $\Psi = \|\cdot\|_1$, alors nous obtenons l'opérateur de seuillage doux de seuil γ ,
- (2) $\text{prox}_{\iota_C \circ \Phi}$ est calculé en utilisant le théorème 2.2.5 sachant que prox_{ι_C} est le projecteur sur l'orthant positif.

L'algorithme 4.1 propose un exemple d'utilisation de l'algorithme 2.7 combinant l'algorithme 2.1 pour l'itération extérieure et l'algorithme 2.4 pour l'intérieure. Le pas de descente τ est parfois difficile à estimer, dans ce cas il est avantageux de remplacer l'itération extérieure par l'algorithme 2.2 qui repose sur des étapes supplémentaires pour estimer le pas. Notons que l'ordre des schémas peut être inversé comme l'ont proposé Chaux et al. [2009], ce qui correspond à la dernière ligne dans le tableau 2.1.

Algorithme 4.1 : Déconvolution sous bruit de Poisson avec stabilisation du bruit, résout $(P_{F_{\text{stab},\gamma,\psi}})$.

Données : Une image observée de comptage y , un dictionnaire Φ , des nombres d'itérations N_{ext} pour le schéma extérieur et N_{int} pour le schéma intérieur, un pas de descente $\tau \in (0, (\frac{3}{2})^{3/2}/(2c \|\mathbf{H}\|^2 \|z\|_\infty))$ et un paramètre de régularisation γ .

Début

1. Initialisation

- 1.1 Appliquer la stabilisation : $\forall i, 1 \leq i \leq n, \quad z[i] = 2\sqrt{y[i] + 3/8}$.
 1.2 Solution initiale : $\alpha_0 = 0$.

2. Itération extérieure (schéma explicite-implicite)

Pour t de 0 à $N_{\text{ext}} - 1$ **faire**

- 2.1 Calcul du gradient de f_1 (4.5) : $g_t = \nabla f_1(\alpha_t)$.
 2.2 Evoluer le long de direction de descente : $\xi_t = \alpha_t - \tau g_t$.
2.3 Calcul de $\text{prox}_{f_2+f_3} \xi_t$ (schéma de Douglas-Rachford)
 2.3.1 Initialiser $p_0 = \xi_t$

Pour l de 0 à $N_{\text{int}} - 1$ **faire**

- 2.3.2 Opérateur proximal de $\gamma\Psi$: pénalité de parcimonie
 $q_l = \text{prox}_{\gamma\Psi} p_l$ (théorème 2.2.5 et section 2.1.2).
 2.3.3 Opérateur proximal de la perturbation quadratique de $\iota_C \circ \Phi$
 $p_{l+1} = \text{prox}_{\iota_C \circ \Phi + \|\cdot - \xi_t\|^2/2} q_l = \text{prox}_{\iota_C \circ \Phi} \left(\frac{q_l + \xi_t}{2} \right)$ (section 2.5.2 et lemme 2.1.29).

finpour

- 2.4 Mise à jour de l'estimée : $\alpha_{t+1} = p_{N_{\text{int}}} - \text{prox}_{f_2} p_{N_{\text{int}}}$.

finpour

Fin

Sorties : Une estimation de l'image d'origine $x^* = \Phi \alpha_{N_{\text{ext}}}$.

Dans le cas où Φ est une trame générale (pas nécessairement ajustée), nous pouvons aussi mettre à profit le théorème 2.2.7 pour le calcul de l'opérateur proximal de $f_2 + f_3$. Le schéma explicite-implicite semble ici adapté car le gradient de f_1 est connu et Lipschitz. Toutefois, il est possible d'utiliser uniquement des opérateurs proximaux et donc de construire l'algorithme final avec deux schémas de Douglas-Rachford (algorithme 2.4), ou avec le schéma généralisé (algorithme 2.8). L'opérateur proximal prox_{f_1} peut être obtenu en utilisant le théorème 2.2.5. Resterait à calculer $\text{prox}_{F_{\text{stab}}}$ ce qui est possible mais que nous ne développerons pas ici.

4.3.2 Complexité temporelle

La complexité temporelle de cet algorithme de déconvolution est dépendante de celle de l'application de l'opérateur Φ , son adjoint Φ^* , \mathbf{H} et son adjoint \mathbf{H}^* . Ces opérateurs ne sont jamais construits explicitement, au contraire ils sont écrits comme des opérateurs implicites rapides prenant un vecteur x , et retournant par exemple Φx . La multiplication par \mathbf{H} ou \mathbf{H}^* coûte deux FFTs, ce qui revient à $2n \log n$ opérations (n étant le nombre de pixels). La complexité de Φ et Φ^* dépend des transformées présentes dans le dictionnaire : par exemple, la transformée en ondelettes orthogonales coûte $\mathcal{O}(n)$ opérations, la transformée discrète en ondelettes invariantes en translation (TI-DWT) coûte $\mathcal{O}(n \log n)$, la transformée en curvelets coûte $\mathcal{O}(n \log n)$, etc. Soit V_Φ la complexité due à l'application de l'opérateur d'analyse (Φ^*) ou de synthèse (Φ). Si Φ est une trame générale (non-ajustée), nous supposons que le théorème 2.2.7 est utilisé pour l'itération intérieure. Soient N_{ext} et N_{int} le nombre d'itérations respectivement des schémas extérieur et intérieur. Nous rappelons que L est le nombre de coefficients. La complexité de l'algorithme 4.1 est résumée ci-dessous :

Complexité temporelle	
Φ base orthonormale	Φ trame
$N_{\text{ext}} (4n \log n + N_{\text{int}} (2V_\Phi + \mathcal{O}(n)))$	$N_{\text{ext}} (4n \log n + 2V_\Phi + N_{\text{int}}(2V_\Phi + \mathcal{O}(L)))$

Le cas avec une base orthonormale demande moins de multiplications par Φ et Φ^T car dans ce cas, Φ est un opérateur linéaire bijectif. Ainsi, le problème d'optimisation $(P_{F_{\text{stab}}, \gamma, \psi})$ peut être réécrit de façon équivalente en terme d'échantillons de l'image au lieu de coefficients, réduisant alors les calculs dans l'algorithme 4.1. Remarquons que si nous utilisons l'algorithme 2.3 de Nesterov pour l'itération extérieure, la complexité temporelle est multipliée par 2, mais la convergence est plus rapide.

4.3.3 A priori analyse

$(P_{F_{\text{stab}}, \gamma, \psi})$ est formulé dans le domaine des coefficients avec un a priori de type synthèse. Cette formulation permet d'obtenir des opérateurs proximaux analytiques. Une alternative

serait de reformuler le problème d'optimisation en termes dans le domaine des pixels avec un a priori dit analyse. Ceci conduit à :

$$\begin{aligned}
 & \arg \min_{\alpha \in \mathbb{R}^n} J(x), \\
 J : x \mapsto & \underbrace{F_{\text{stab}} \circ \mathbf{H}(x)}_{f'_1(x)} + \underbrace{\gamma \Psi \circ \Phi^*(x)}_{f'_2(x)} + \underbrace{\iota_{\mathcal{C}}(x)}_{f'_3(x)}, \quad (P'_{F_{\text{stab}}, \gamma, \psi})
 \end{aligned}$$

avec $\Psi(\alpha) = \sum_{i=1}^L \psi(\alpha[i])$.

La résolution de ce problème est accomplie avec les mêmes outils que précédemment. Cependant, la difficulté reste le calcul de $\text{prox}_{f'_2}$ à cause de la composition avec un opérateur linéaire injectif borné. Ceci est résolu en appliquant le théorème 2.2.7 (avec la décomposition f'_1 et $f'_2 + f'_3$). L'algorithme 4.2 propose un exemple d'utilisation avec ce dernier découpage en combinant l'algorithme 2.1 avec le théorème 2.2.7.

4.4 Second type d'algorithme : cas non stabilisé

Dans le cas non-stabilisé, le problème d'optimisation $(P_{F_{\text{poisson}}, \gamma, \psi})$ est composé de trois fonctions appartenant à $\Gamma_0(\mathcal{H})$. Le contrainte d'appartenance à un ensemble convexe fermé \mathcal{C} n'est pas utile si \mathcal{C} est l'orthant positif, cependant, pour être général, nous conservons le terme. Aucune des trois fonctions n'est différentiable avec un gradient Lipschitz et donc ne répond à la contrainte d'utilisation du schéma explicite-implicite.

4.4.1 Algorithme

Deux choix s'offrent donc à nous pour résoudre $(P_{F_{\text{poisson}}, \gamma, \psi})$: le premier type d'algorithme en combinant deux schémas de Douglas-Rachford (section 2.2.3), un schéma DR et FB (ou Nesterov) ou le deuxième type d'algorithme (section 2.6). Pour illustrer, nous avons choisi ce dernier, car il implique les mêmes calculs d'opérateurs proximaux et ne demande pas une projection finale pour obtenir une solution.

Les opérateurs proximaux des fonctions impliquées f_1 , f_2 et f_3 sont analytiquement calculables :

- (1) $\text{prox}_{\gamma \Psi}$ est donné par le lemme 2.1.29, par exemple si $\Psi = \|\cdot\|_1$, alors nous obtenons l'opérateur de seuillage doux de seuil γ ;
- (2) $\text{prox}_{\iota_{\mathcal{C}} \circ \Phi}$ est calculé en utilisant le théorème 2.2.5 ;
- (3) $\text{prox}_{F_{\text{poisson}}}$ est donné par le lemme suivant :

Lemme 4.4.1 ([Combettes et Pesquet, 2007]).

Algorithme 4.2 : Déconvolution sous bruit de Poisson avec stabilisation du bruit dans le domaine des pixels, résout $(P'_{F_{\text{stab}}, \gamma, \psi})$.

Données : Une image observée de comptage y , un dictionnaire Φ , des nombres d'itérations N_{ext} pour le schéma extérieur et N_{int} pour le schéma intérieur, deux pas de descente $\tau \in (0, (\frac{3}{2})^{3/2}/(2 \|\mathbf{H}\|^2 \|z\|_{\infty}))$ et $\nu \in (0, 2/c)$, et un paramètre de régularisation γ .

Début

1. Initialisation

1.1 Appliquer la stabilisation : $\forall i, 1 \leq i \leq n, \quad z[i] = 2\sqrt{y[i] + 3/8}$.

1.2 Solution initiale : $x_0 = 0$.

2. Itération extérieure (schéma explicite-implicite)

Pour t de 0 à $N_{\text{ext}} - 1$ **faire**

2.1 Calcul du gradient de f'_1 (4.5) : $g_t = \nabla f'_1(x_t)$.

2.2 Evoluer le long de direction de descente : $\xi_t = x_t - \tau g_t$.

2.3 Calcul de $\text{prox}_{\iota_C + \gamma \Psi \circ \Phi^*} \xi_t$ (**Théorème 2.2.7 avec** $f_1 = \iota_C, f_2 = \gamma \Psi$ **et** $\mathbf{A} = \Phi^*$)

2.3.1 Initialiser $p_0 = \Phi^* \xi_t$

Pour l de 0 à $N_{\text{int}} - 1$ **faire**

2.3.2 Opérateur proximal de ι_C : projection sur l'ensemble convexe fermé \mathcal{C}

$q_l = \mathcal{P}_{\mathcal{C}}(-\Phi p_l + \xi_t)$ (expression (2.50)).

2.3.3 Opérateur proximal de $\gamma \Psi$: pénalité de parcimonie

$q_{l+1/2} = \text{prox}_{\gamma \Psi}(p_l/n_u + \Phi^* q_l)$ (lemme 2.1.29 et (2.50)).

2.3.4 Mise à jour : $p_{l+1} = p_l + \nu(\Phi^* q_l - q_{l+1/2})$

finpour

2.4 Mise à jour de l'estimée : $x_{t+1} = \mathcal{P}_{\mathcal{C}}(-\Phi p_{N_{\text{int}}} + \xi_t)$.

finpour

Fin

Sorties : Une estimation de l'image d'origine $x^* = x_{N_{\text{ext}}}$.

Soit y l'image de comptage (i.e. l'observation), l'opérateur proximal de F_{poisson} est :

$$\text{prox}_{\beta F_{\text{poisson}}} x = \left(\frac{x[i] - \beta + \sqrt{|x[i] - \beta|^2 + 4\beta y[i]}}{2} \right)_{1 \leq i \leq n}. \quad (4.17)$$

L'algorithme 4.3 propose un exemple d'utilisation de l'algorithme 2.8. Les trois opérateurs proximaux sont calculés par :

prox_{f_1} est obtenu en combinant le théorème 2.2.5 avec le lemme 4.4.1 (la solution est obtenue en utilisant par exemple l'algorithme 2.5 ou 2.6) ;

prox_{f_2} avec le lemme 2.1.29 ;

prox_{f_3} en combinant le théorème 2.2.5 avec le projecteur sur l'orthant positif.

Algorithme 4.3 : Déconvolution sous bruit de Poisson sans stabilisation, résoudre $(P_{F_{\text{poisson}}, \gamma, \psi})$.

Données : Une image observée de comptage y , un dictionnaire Φ , le nombre d'itérations N_{ext} pour le schéma extérieur et un paramètre de régularisation γ .

Début

1. Initialisation :

$$\forall i \in \{0, 1, 2\}, \quad p_{(0,i)} = \Phi^* y.$$

Pour $t = 0$ jusqu'à $N_{\text{ext}} - 1$ **faire**

2. Calcul des différents opérateurs proximaux :

2.1 Attache aux données (algorithme 2.5 ou 2.6 et lemme 4.4.1)

$$\xi_{(t,0)} = \text{prox}_{f_1/3} P_{(t,0)}.$$

2.2 Pénalité de parcimonie (Lemme 2.1.29)

$$\xi_{(t,1)} = \text{prox}_{\gamma \Psi/3} P_{(t,1)}.$$

2.3 Projection sur l'ensemble convexe fermé \mathcal{C}' (algorithme 2.5 ou 2.6)

$$\xi_{(t,2)} = \text{prox}_{\iota_{\mathcal{C}} \circ \Phi/3} P_{(t,3)}$$

3. Nouvelle estimée :

$$\alpha_{t+1} = (\xi_{(t,0)} + \xi_{(t,1)} + \xi_{(t,2)})/3.$$

4. Mise à jour des composantes :

$$\forall i \in \{0, 1, 2\}, \quad p_{(t+1,i)} = p_{(t,i)} + 2\alpha_{t+1} - \alpha_t - \xi_{(t,i)},$$

finpour

Fin

Sorties : Une estimation de l'image d'origine $x^* = \Phi \alpha_{N_{\text{ext}}}$.

Remarque 4.4.2.

Si Φ est une trame générale, l'algorithme 4.3 aura deux itérations intérieures (une pour prox_{f_1}

et une pour prox_{f_3}) ce qui peut devenir lourd. Dans ce cas, l'utilisation de Nesterov (algorithme 2.6) pour le calcul de ces opérateurs proximaux peut devenir essentiel pour converger rapidement, et ce malgré les calculs supplémentaires.

4.4.2 Complexité temporelle

Nous supposons le même coût pour les diverses opérations que dans la section 4.3.2. Nous avons essentiellement deux cas, si Φ est une trame ajustée alors le calcul de prox_{f_3} demande une opération de synthèse et une opération d'analyse, sinon nous considérons le coût de l'algorithme 2.5 qui demande les deux opérations d'analyse et de synthèse à chaque itération intérieure. Le calcul de prox_{f_1} demande, avec le même algorithme que précédemment, deux convolutions et deux opérations d'analyse et de synthèse à chaque itération intérieure. Celui de prox_{f_2} a une complexité linéaire.

Soient N_{ext} et N_{int} le nombre d'itérations du schéma extérieur et du schéma intérieur. Nous rappelons que L est le nombre de coefficients. La complexité de l'algorithme 4.3 est résumée ci-dessous :

Complexité temporelle	
Φ trame ajustée ou base orthonormale	Φ trame générale
$N_{\text{ext}} (4V_{\Phi} + N_{\text{int}} (4n \log n) + \mathcal{O}(L))$	$N_{\text{ext}} (N_{\text{int}} (4V_{\Phi} + 4n \log n + \mathcal{O}(L)))$

L'utilisation de l'algorithme 2.6 (Nesterov) pour converger plus rapidement, demande deux fois plus d'opérations pour chaque itération intérieure.

4.4.3 A priori analyse

Comme précédemment, $(P_{F_{\text{poisson}}, \gamma, \psi})$ peut être formulé dans le domaine des pixels avec un a priori analyse :

$$\begin{aligned}
 & \arg \min_{\alpha \in \mathbb{R}^n} J(x), \\
 J : x \mapsto & \underbrace{F_{\text{poisson}} \circ \mathbf{H}(x)}_{f'_1(x)} + \underbrace{\gamma \Psi \circ \Phi^*(x)}_{f'_2(x)} + \underbrace{\iota_{\mathcal{C}}(x)}_{f'_3(x)}. \quad (P'_{F_{\text{poisson}}, \gamma, \psi})
 \end{aligned}$$

La résolution du problème est effectuée avec les mêmes outils que précédemment. Nous pouvons constater que la reformulation ne change pas beaucoup les calculs des différents opérateurs proximaux. En effet, le théorème 2.2.5 fournit les formules de calculs de $\text{prox}_{f'_1}$ et $\text{prox}_{f'_2}$. Dans ce cas la complexité de l'algorithme devient : $N_{\text{ext}} (N_{\text{int}} (2V_{\Phi} + 4n \log n) + \mathcal{O}(n))$ que Φ soit une trame ajustée ou non.

Algorithme 4.4 : Déconvolution sous bruit de Poisson sans stabilisation dans le domaine des pixels, résout $(P'_{F_{\text{poisson}}, \gamma, \psi})$.

Données : Une image observée de comptage y , un dictionnaire Φ , des nombres d'itérations N_{ext} pour le schéma extérieur et un paramètre de régularisation γ .

Début

1. Initialisation :

$$\forall i \in \{0, 1, 2\}, \quad p_{(0,i)} = y.$$

Pour $t = 0$ jusqu'à N_{ext} **faire**

2. Calcul des différents opérateurs proximaux :

2.1 Attache aux données (algorithme 2.5 ou 2.6 et lemme 4.4.1)

$$\xi_{(t,0)} = \text{prox}_{f'_1} p_{(t,0)}.$$

2.2 Pénalité de parcimonie (algorithme 2.5 ou 2.6 et lemme 2.1.29)

$$\xi_{(t,1)} = \text{prox}_{\gamma \Psi \circ \Phi^*} p_{(t,1)}.$$

2.3 Projection sur l'ensemble convexe fermé \mathcal{C}

$$\xi_{(t,2)} = \mathcal{P}_{\mathcal{C}}(p_{(t,3)})$$

3. Nouvelle estimée :

$$x_{t+1} = (p_{(t,0)} + p_{(t,1)} + p_{(t,2)})/3.$$

4. Mise à jour des composantes :

$$\forall i \in \{0, 1, 2\}, \quad p_{(t+1,i)} = p_{(t,i)} + 2x_{t+1} - x_t - \xi_{(t,i)},$$

finpour

Fin

Sorties : Une estimation de l'image d'origine $x^* = x_{N_{\text{ext}}}$.

4.5 Choix du paramètre de régularisation

Nous avons présenté des exemples d’algorithmes de déconvolution sous bruit de Poisson. Ces algorithmes reposent par leur conception sur un équilibre entre un terme d’attache aux données et un terme de pénalité. Cet équilibre est représenté par le paramètre γ qui définit l’importance accordée à chaque terme.

Le choix de γ est souvent un choix crucial dans les problèmes inverses régularisés. Pour une application et un corpus d’images donnés (par exemple un microscope confocal) ou si une vérité terrain est disponible, une approche naïve et brute consisterait à tester plusieurs valeurs de γ et à prendre celle qui conduit à la meilleure qualité visuelle de déconvolution. Toutefois, cette approche reste en réalité peu praticable.

Dans ses travaux pour résoudre la forme contrainte du débruitage par la variation totale (discrète), Chambolle [2004] utilise une série de problèmes régularisés chacun avec une valeur différente de γ . Pour obtenir une contrainte de taille σ sur les résidus, une mise à jour intuitive de γ est $\gamma_{\text{nouveau}} = \gamma_{\text{ancien}} \sigma / \|\text{residus}_{\text{ancien}}\|$. L’analyse théorique est rigoureuse pour le débruitage. En pratique, il a été observé par Chambolle [2004] que cette règle fonctionne toujours si γ est mis à jour à chaque itération de l’algorithme de résolution, et ce même avec des problèmes inverses. Transposé à notre configuration, pour une taille voulue de résidu σ , la règle de mise à jour est :

$$\gamma_{t+1} = \gamma_t \frac{\sigma}{\left\| z - 2\sqrt{\mathbf{H}\Phi\alpha_t(z) + \frac{3}{8}} \right\|}. \quad (4.18)$$

Toutefois, cette méthode demande d’avoir une idée du paramètre σ . Dans notre cas avec la transformée stabilisatrice de la variance, nous pouvons prendre $\sigma = \sqrt{n}\sqrt{1 + 2\sqrt{\frac{2}{n}}} \approx \sqrt{n}$ par les inégalités de concentration pour les variables χ_n^2 .

Nous proposons de sélectionner objectivement le paramètre de régularisation γ avec un critère adaptatif de sélection de modèle tel que la validation croisée généralisée (GCV) [Golub et al., 1979]. D’autres critères sont aussi disponibles incluant le critère d’information d’Akaike (AIC) [Akaike et al., 1973] ou le critère d’information bayésien (BIC) [Schwarz, 1978]. Le GCV tente d’obtenir une estimation de γ selon les données fournies en minimisant :

$$\text{GCV}(\gamma) = \frac{\left\| z - 2\sqrt{\mathbf{H}\Phi\alpha^*(z) + \frac{3}{8}} \right\|^2}{(n - df)^2}, \quad (4.19)$$

où $\alpha^*(z)$ représente la solution obtenue avec l’un des algorithmes 4.1, 4.2, 4.3 ou 4.4 et df le nombre effectif de degrés de liberté.

Trouver une expression analytique de df est assez compliqué dans notre cas, car nous faisons face à deux difficultés :

- (i) le modèle d’observation (4.3) est non-linéaire,

(ii) la solution $\alpha^*(z)$ n'est pas connue sous une forme analytique, mais est donnée par un algorithme itératif.

« Degrés de liberté » est une terminologie usuelle en statistiques. Dans le cas d'une régression linéaire (surdéterminée), df est le nombre de colonnes dans la matrice de dessin de la régression. Plus généralement, les degrés de liberté sont souvent utilisés pour quantifier la complexité du modèle créé par une procédure de modélisation statistique. Cependant, d'un point de vue général, il n'y a pas de correspondance exacte entre le nombre de degrés de liberté df et le nombre de paramètres d'un modèle. Dans le cas des solutions pénalisées de problèmes inverses, où l'estimateur est linéaire en l'observation, par exemple, pour la régularisation de Tikhonov également connue, en statistique, sous la dénomination de *ridge regression*, df est simplement la trace de la matrice d'influence. Néanmoins, il est en général difficile de trouver une expression analytique de df pour une procédure générale de modélisation non-linéaire telle que la nôtre. Cela reste un champ actif et difficile de recherche en statistiques.

La théorie de l'estimation du risque sans biais de Stein (SURE) [Stein, 1981] donne une définition rigoureuse des degrés de liberté pour une procédure d'estimation. En utilisant notre notation, soient α^* la solution fournie par l'algorithme de déconvolution et $z^*(z) = 2\sqrt{\mathbf{H}\Phi\alpha^*(z) + \frac{3}{8}}$ le modèle ajusté selon l'observation z . Comme $Z | \alpha \sim \mathcal{N}\left(2\sqrt{\mathbf{H}\Phi\alpha + \frac{3}{8}}, 1\right)$, alors selon Efron [1981] le nombre de degrés de liberté de notre procédure est

$$df(\gamma) = \sum_{i=1}^n \text{Cov}(z^*[i](z), z[i]), \quad (4.20)$$

une quantité aussi appelée l'estimateur optimiste de $z^*(z)$. Si l'algorithme d'estimation est tel que $z^*(z)$ est faiblement différentiable [Stein, 1981] selon z , alors sa divergence est bien définie dans le sens faible (c'est le cas si $z^*(z)$ est uniformément Lipschitz continue). Le lemme de Stein donne alors ce qui est appelé la formule de la divergence :

$$df(\gamma) = \sum_{i=1}^n \text{Cov}(z^*(z)[i], z[i]) = \mathbb{E}_Z [\text{div}(z^*(z))] = \mathbb{E}_Z \left[\sum_{i=1}^n \frac{\partial z^*(z)[i]}{\partial z[i]} \right]. \quad (4.21)$$

L'espérance \mathbb{E}_Z est prise sous la distribution de Z . Le terme df est alors la somme des sensibilités de chaque valeur ajustée selon la valeur observée. Par exemple, la dernière expression de cette formule a été utilisée dans [Jansen et al., 1997] pour du débruitage avec des ondelettes orthogonales. Cependant, trouver une forme analytique de df à partir de cette formule pour des procédures de modélisation non-linéaire est notoirement difficile. Pour surmonter cette difficulté, une méthode par rééchantillonnage aléatoire comme le *bootstrap* [Efron, 2004] peut être utilisée pour obtenir un estimateur (asymptotiquement) sans biais de df . Ye [1998] et Shen et Ye [2002] proposent d'utiliser une technique de perturbation de données pour calculer numériquement une estimée (approximativement) sans biais de df lorsque la forme analytique de

$z^*(z)$ n'est pas disponible. A partir de (4.21), l'estimateur de df prend la forme :

$$\begin{aligned}\widehat{df}(\gamma) &= \mathbb{E}_{V_0} \left[\left\langle v_0, \frac{z^*(z + \tau v_0) - z^*(z)}{\tau} \right\rangle \right], \quad V_0 \sim \mathcal{N}(0, \mathbf{I}), \\ &= \frac{1}{\tau^2} \int \langle v, z^*(z + v) \rangle \phi(v; \tau^2 \mathbf{I}) dv, \quad V \sim \mathcal{N}(0, \tau^2 \mathbf{I}),\end{aligned}\quad (4.22)$$

où $\phi(v; \tau^2 \mathbf{I})$ est la densité gaussienne multivariée de covariance $\tau^2 \mathbf{I}$. Cette formule reste valable si V est remplacé par n'importe quel vecteur de variables aléatoires avec un nombre fini de moments. L'auteur dans [Ye, 1998] prouve que c'est une estimation sans biais de df lorsque $\tau \rightarrow 0$, c'est à dire $\lim_{\tau \rightarrow 0} \mathbb{E}_Z [\widehat{df}(\gamma)] = df(\gamma)$. Ceci peut être calculé par une intégration de Monte-Carlo avec τ près de 0.6 comme proposé par Ye [1998]. Cependant, autant les méthodes par rééchantillonnage aléatoire que celle de Ye [1998], bien que générales et capables d'être utilisées pour n'importe quelle $\Psi \in \Gamma_0(\mathbb{R}^L)$, sont gourmandes en temps de calcul. Nous n'utilisons justement pas ces méthodes essentiellement pour cette raison.

Zou et al. [2007] ont récemment étudié les degrés de liberté du Lasso¹ (ou BPDN) dans le cadre proposé par la méthode SURE. Ils prouvent que pour n'importe quel γ le nombre de coefficients non-nuls dans le modèle est une estimation consistante et sans biais de df . Cependant, pour que leurs résultats tiennent rigoureusement, la matrice $\mathbf{A} = \mathbf{H}\Phi$ dans le Lasso doit être sur-déterminé $L < n$ avec un rang égal à L . Toutefois, nous pouvons prouver que leur estimateur intuitif peut être étendu au cas sous-déterminé (i.e. $L \geq n$) sous la condition (UC) de Dossal [2007, Théorème 2]. Cela conduit vers un estimateur sans biais de df , cependant la consistance est beaucoup plus difficile à prouver. En effet la matrice de Gram $\mathbf{A}^* \mathbf{A}$ doit être définie positive ce qui apparaît uniquement dans le cas spécial où Φ est une base orthogonale et $\ker(\mathbf{H}) = \emptyset$. De plus, même avec la norme ℓ_1 , étendre ce simple estimateur rigoureusement à notre cas fait face à deux sérieuses difficultés additionnelles, au-delà du fait que \mathbf{A} est sous-déterminée : la non-linéarité de l'équation du modèle de dégradation (4.3) et la contrainte de positivité dans $(P_{F, \gamma, \psi})$.

En suivant cette discussion, il apparaît clairement qu'estimer df est soit intensif en calculs (rééchantillonnage aléatoire ou par perturbation), soit analytiquement difficile à obtenir. Dans cette thèse, suivant la méthode de [Zou et al., 2007], nous conjecturons qu'un simple estimateur de df , est fourni par le cardinal du support de α^* . C'est à dire, à partir du lemme 2.1.29 et du schéma explicite-implicite (algorithme 4.1), nous posons

$$\widehat{df}(\gamma) = \text{card} \{i = 1, \dots, L \mid |\alpha^*[i]| \geq \gamma\tau\} . \quad (4.23)$$

Avec une aussi simple expression, nous pouvons exprimer le critère de sélection de modèle GCV dans (4.19). Notons que cette formule est utilisable uniquement avec les algorithmes utilisant

¹Le modèle Lasso correspond au cas de (1.26) où le modèle de dégradation est linéaire et Ψ est la norme ℓ_1 .

un a priori synthèse (par exemple, les algorithmes 4.1 et 4.3). Cependant, nous pouvons montrer qu’une formulation avec un a priori analyse se ramène à une formulation synthèse en rajoutant la contrainte supplémentaire $\mathbf{T}\mathbf{T}^+\alpha = \alpha$, où $\mathbf{T} = \Phi^*$ est l’opérateur d’analyse.

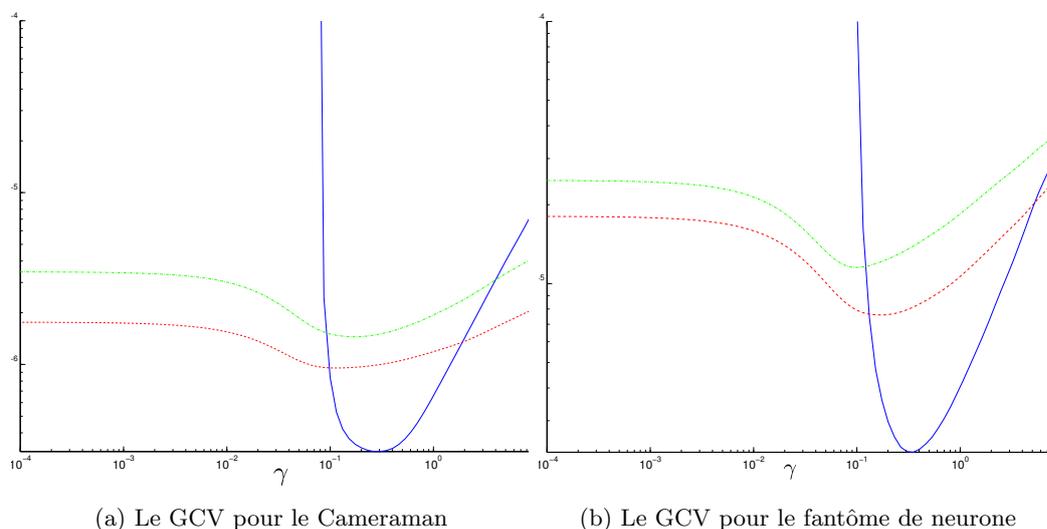


FIG. 4.3: Le GCV (—), le MAE (---) et le MSE (---) pour le cameraman (a) et l’image simulée de neurone (b). La transformée en ondelettes discrète invariante par translation a été utilisée pour l’image du Cameraman, et la transformée en curvelets avec la simulation de neurone.

Bien que cette formule soit seulement une approximation, dans toutes nos expériences, elle a assez bien fonctionné. La figure 4.3(a) et (b) montre le comportement du GCV en fonction de γ pour deux images : le cameraman (figure 4.6(a)) et l’image simulée de neurone (figure 4.4(a)). Comme la vérité terrain est connue dans la simulation, nous avons calculé pour chaque valeur de γ , l’erreur absolue moyenne (MAE) bien adaptée au bruit de Poisson laquelle est assez proche de racine carrée de la distance de Hellinger [Barron et Cover, 1991], ainsi que l’erreur quadratique moyenne (MSE) entre l’image restaurée et l’image réelle. Nous pouvons clairement observer que le GCV atteint son minimum près de ceux des erreurs MAE et MSE. Le paramètre de régularisation prédit par le critère GCV semble légèrement plus élevé que celui du MSE, ce qui peut conduire à une image restaurée légèrement sur-lissée.

4.6 Expérimentations

4.6.1 Protocole expérimental

Nous avons testé la performance de notre approche sur plusieurs images : une image simulée de neurone (128×128 pixels), une image de microscopie confocale (370×370 pixels)

de cellule de micro-vaisseaux (fourni avec ImageJ²), le Cameraman (256×256 pixels), une image de simulation astronomique de la caméra à champ large du télescope spatial Hubble d'un groupement distant de galaxies [Starck et Murtagh, 2006]. Les algorithmes 4.1 avec la stabilisation et l'a priori synthèse (PoissonStab) et 4.3 sans stabilisation et avec l'a priori synthèse (PoissonNS) ont été comparés aux algorithmes : RL régularisé avec la variation totale (section 1.2.4.3, RL-TV [Dey et al., 2004]), RL régularisé avec le support multirésolution (section 1.2.4.4, RL-MRS [Starck et al., 1995]), la reconstruction rapide par suppression d'arbre invariante par translation combinée à un algorithme EM (FTITPR [Willett et Nowak, 2004]) et la méthode proximale naïve qui suppose un bruit gaussien (NaiveGauss [Vonesch et Unser, 2007]). La fonction Ψ choisie pour les algorithmes PoissonStab et PoissonNS était la norme ℓ_1 dont l'opérateur proximal est l'opérateur de seuillage doux. Pour tous les résultats présentés, chaque algorithme a été lancé avec $N_{\text{ext}} = 200$ itérations, suffisant pour atteindre la convergence. Le schéma intérieur n'est itéré que $N_{\text{int}} = 1$ fois pour PoissonStab et $N_{\text{int}=5}$ pour les deux sous-itérations dans PoissonNS. Il semble que cela soit suffisant pour obtenir des erreurs sommables. Pour tous les résultats ci-dessous, γ pour nos deux algorithmes a été sélectionné en utilisant le critère GCV (4.19). Afin d'avoir une comparaison objective avec Vonesch et Unser [2007], γ a aussi été choisi en adaptant la formule du GCV (4.19) au cas gaussien. Pour PoissonNS, nous avons utilisé le GCV pour obtenir un ordre de grandeur, mais pour les résultats visuels, nous avons modifié la valeur afin d'obtenir des images moins lisses que celles proposées par le GCV.

4.6.2 Discussion des résultats

4.6.2.1 Données simulées

La figure 4.4(a) montre une image simulée d'un neurone avec une épine dendritique en forme de champignon. L'intensité maximale de l'image est de 30. Les versions lissées par h (la PSF simulée de h est un moyenneur 7×7) et bruitées par un bruit de Poisson sont présentées par les images en (b) et (c). Avec ce neurone et pour la méthode NaiveGauss et les nôtres, le dictionnaire Φ contient la transformée en curvelets qui correspond à une trame ajustée [Candès et al., 2005]. Les résultats de déconvolution sont présentés dans les figures 4.4(d)-(i). Comme prévu à ce niveau d'intensité, l'algorithme NaiveGauss fonctionne mal, car il ne correspond pas au modèle de dégradation à cette intensité. Il apparaît que le NaiveGauss ne régularise pas assez l'estimée et, le bruit de Poisson, dépendant du signal, n'est pas toujours contrôlé. Ce comportement du NaiveGauss, qui est prévisible à ce niveau d'intensité, sera observé sur toutes les images de test. L'algorithme RL-TV fournit un bon résultat de déconvolution. Cependant le fond est dominé par des artefacts et le neurone restauré montre des artefacts de type *stair-*

²<http://rsb.info.nih.gov/ij/>

casings typique d'une régularisation avec la variation totale. L'algorithme PoissonStab fournit une déconvolution assez bonne visuellement, sur cet exemple. Il restaure correctement l'épine, toutefois le fond n'est pas complètement nettoyé. L'algorithme PoissonNS fournit une déconvolution assez bonne également, mais un peu lisse, en revanche le fond est propre. L'algorithme RL-MRS montre aussi un bon résultat de déconvolution. Finalement sur cette image, l'algorithme FTITPR conduit à une estimée très lisse du neurone mais avec pratiquement aucune déconvolution.

Ces résultats visuels qualitatifs sont confirmés par une mesure quantitative de la qualité de la déconvolution. Nous avons utilisé l'erreur absolue moyenne (MAE) qui est bien adaptée au bruit de Poisson. Pour chaque valeur d'intensité, dix réalisations lissées et bruitées sont générées et les mesures sont calculées pour chaque algorithme de déconvolution. Les moyennes obtenues pour le MAE à partir des dix réalisations sont présentées dans la figure 4.8. D'un point de vue général, nos algorithmes fonctionnent très correctement à tous les régimes d'intensité (surtout entre les intensités moyenne et faible). L'algorithme NaiveGauss est parmi les plus mauvais pour les faibles intensités. Ses performances s'améliorent lorsque l'intensité augmente, ce qui était prévu par le théorème central limite. L'algorithme RL-MRS est efficace à faible et moyenne intensité, il est même meilleur que nos algorithmes sur les images de cellules. PoissonNS est meilleur que PoissonStab pour la plus faible intensité, ce qui est attendu, mais les performances des deux se rejoignent à partir des intensités moyennes, ce qui est aussi attendu par la validité asymptotique de la VST. L'algorithme RL-TV est le moins efficace à faible intensité. Nous pensons que les artefacts de type *staircasing*, provenant de la régularisation, sont à l'origine de ce comportement. A haute intensité, RL-TV devient compétitif et ces mesures de MAE sont comparables à celles produites par nos algorithmes.

La même expérience que ci-dessus a été conduite sur l'image de microscopie confocale de cellule (voir figure 4.5). Dans cette expérience, la PSF est la même que précédemment, afin d'avoir la même dégradation. Pour l'algorithme NaiveGauss et les nôtres, le dictionnaire Φ contenait la transformée en ondelettes invariante par translation (TI-DWT). Le résultat de déconvolution produit par l'algorithme NaiveGauss est dégradé par les artefacts. L'algorithme RL-TV conduit à une bonne restauration de petits détails isolés mais dominée par les artefacts de *staircasing*. L'algorithme FTITPR fournit une estimation un peu trop lisse, alors que les algorithmes PoissonStab et PoissonNS donnent des déconvolutions plus propres. Ce dernier produit également une image plus lisse que PoissonStab avec des structures mieux restaurées. Ces résultats visuels sont en adéquation avec les mesures de MAE (figure 4.8). En particulier, nous pouvons noter que l'algorithme RL-MRS montre le meilleur comportement, et que les performances de nos approches, comparées aux autres méthodes, sont fortement semblables à celles obtenues avec la précédente image de neurone.

La figure 4.6 montre les résultats de l'expérience sur le Cameraman avec un maximum d'in-

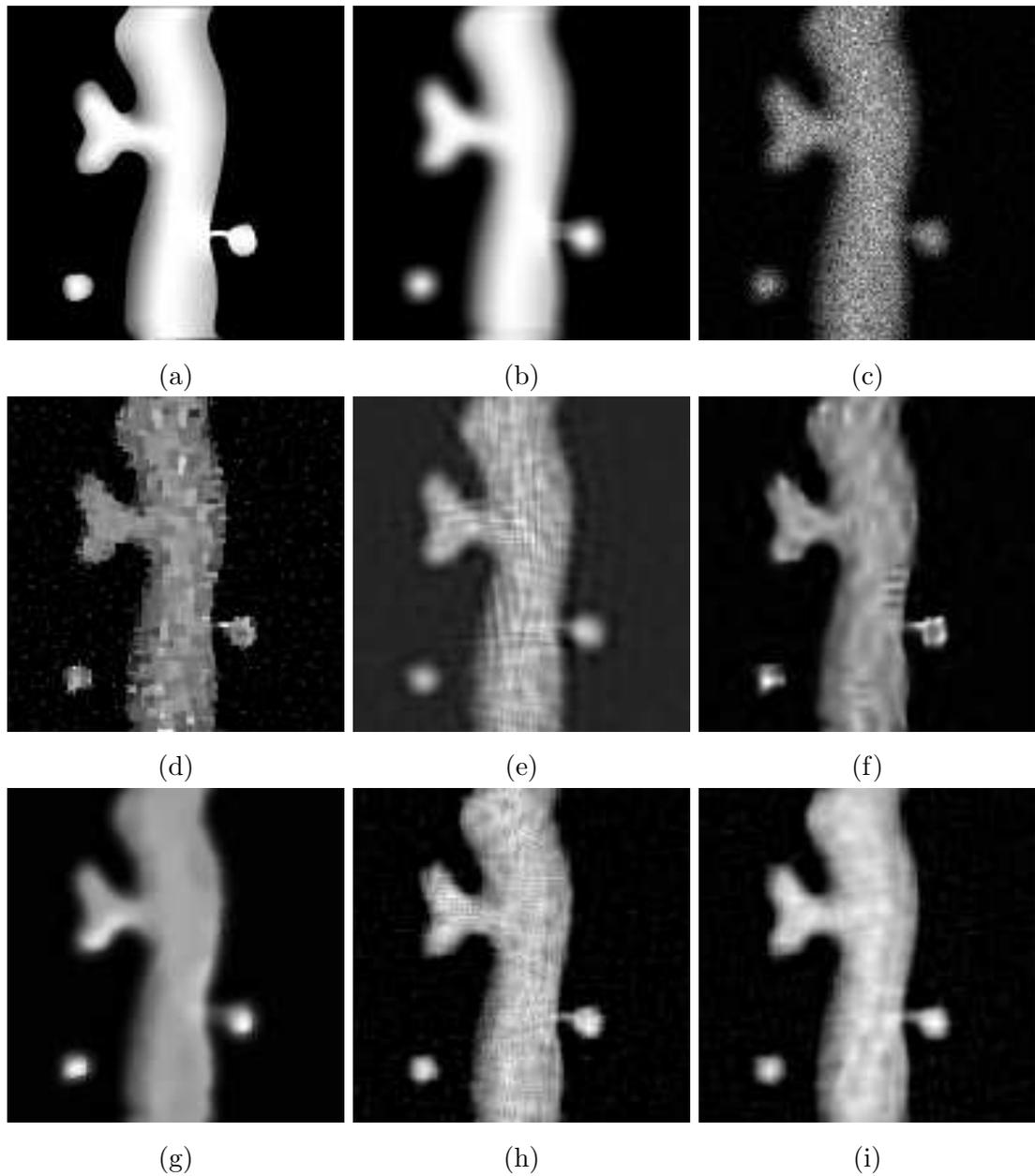


FIG. 4.4: Déconvolution d'une image simulée de neurone (Intensité ≤ 30). (a) Originale, (b) Convoluée, (c) Convoluée & bruitée, (d) RL-TV [Dey et al., 2004], (e) NaiveGauss [Vonesch et Unser, 2007], (f) RL-MRS [Starck et Murtagh, 2006], (g) FTITPR [Willett et Nowak, 2004], (h) PoissonStab, (i) PoissonNS.

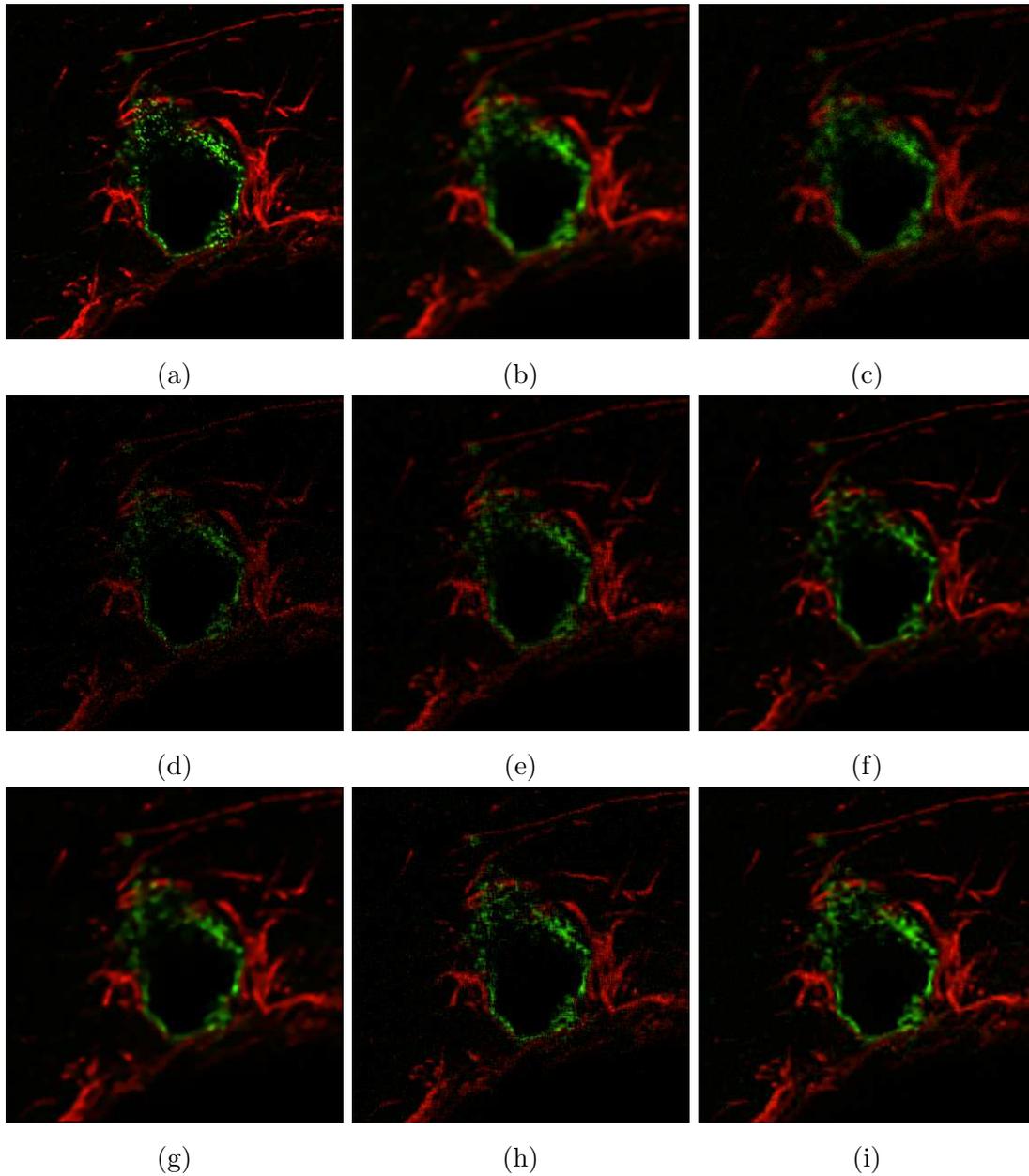


FIG. 4.5: Déconvolution d'une image microscopique de cellule (Intensité ≤ 30). (a) Originale, (b) Convoluée, (c) Convoluée & bruitée, (d) RL-TV [Dey et al., 2004], (e) NaiveGauss [Vonesch et Unser, 2007], (f) RL-MRS [Starck et Murtagh, 2006], (g) FTITPR [Willett et Nowak, 2004], (h) PoissonStab, (i) PoissonNS.

tensité de 30. La PSF était la même que précédemment. De même, le dictionnaire contenait également la transformée TI-DWT. Nous pouvons remarquer que la dégradation dans la figure 4.6(c) est assez importante. L’algorithme PoissonStab produit un résultat agréable visuellement avec un bon équilibre entre déconvolution et régularisation. Toutefois, quelques artefacts subsistent. L’algorithme PoissonNS produit le résultat le plus agréable avec peu d’artefacts et un arrière plan partiellement restauré. L’algorithme RL-MRS conduit à une déconvolution avec plus d’artefacts que nos approches et souffre d’une perte de photométrie. L’algorithme FTITPR fournit encore une estimation trop lisse avec une perte des détails. Les deux algorithmes RL-TV et NaiveGauss conduisent à un résultat dominé par des artefacts. Cette impression visuelle est en accord avec les valeurs mesurées de MAE (figure 4.8).

Afin d’avoir une idée du coût de calculs des différents algorithmes comparés, le tableau 4.1 résume les temps d’exécution pour l’image du Cameraman avec un PC Intel Core 2 Duo 2GHz, 2GB RAM. A l’exception de l’algorithme RL-MRS qui est écrit en C++, tous les autres algorithmes ont été implémentés en Matlab.

Méthode	Temps (en s)
PoissonStab ($N_{\text{int}} = 1$)	88
PoissonNS ($N_{\text{int}} = 5$)	655
NaiveGauss	71
RL-MRS	99.5
RL-TV	15.5

TAB. 4.1: Temps d’exécution pour l’image 256×256 simulée du Cameraman en utilisant la transformée TI-DWT ($N_{\text{ext}} = 200$).

Le même protocole expérimental a été appliqué à l’image simulée d’un groupement de galaxies lointaines avec la caméra à champ large du télescope spatial Hubble présentée dans la figure 4.7(a). Nous avons utilisé la fonction d’étalement du point du télescope spatial Hubble comme donnée dans [Starck et Murtagh, 2006]. L’intensité maximale dans l’image floutée était 5000. Pour l’algorithme NaiveGauss, PoissonStab et PoissonNS, le dictionnaire contenait la trame TI-DWT. Pour cette image, l’algorithme RL-MRS est clairement le meilleur car il a été conçu pour gérer le bruit de Poisson dans de telles images. La plupart des structures peu lumineuses sont restaurées par le RL-MRS et les objets larges et lumineux sont bien déconvolués. L’algorithme PoissonStab conduit aussi à un bon résultat de déconvolution et préserve la plupart des objets peu lumineux qui sont difficiles à percevoir dans l’image dégradée. Cependant, l’arrière plan est moins bien restauré que celui produit par RL-MRS. L’algorithme PoissonNS conduit à un résultat similaire, mais avec un arrière plan plus propre et des structures légèrement plus lisses. A ce régime d’intensité, l’algorithme NaiveGauss fourni des résultats



FIG. 4.6: Déconvolution du cameraman ($\text{Intensité} \leq 30$). (a) Originale, (b) Convoluée, (c) Convoluée & bruitée, (d) RL-TV [Dey et al., 2004], (e) NaiveGauss [Vonesch et Unser, 2007], (f) RL-MRS [Starck et Murtagh, 2006], (g) FTITPR [Willett et Nowak, 2004], (h) PoissonStab, (i) PoissonNS.

satisfaisants comparables aux nôtres sur les galaxies. L’algorithme FTITPR arrive à restaurer correctement la plupart des structures importantes avec un arrière plan très propre, cependant beaucoup d’objets peu lumineux sont perdus. L’algorithme RL-TV donne un résultat de déconvolution comparable au nôtre pour des objets lumineux, mais l’arrière plan est dominé par de fausses structures faiblement lumineuses.

Pour mettre en exergue l’intérêt de prendre en compte la diversité morphologique des structures géométriques présentes dans l’image à restaurer, nous avons quantifié l’influence du dictionnaire sur le résultat de déconvolution (avec PoissonStab) sur trois images de test. Premièrement, nous présentons dans la figure 4.9 les résultats d’une expérimentation sur une image simulée 128×128 , contenant des sources ponctuelles (en haut à gauche), des sources gaussiennes et des lignes. Dans cette expérience, le maximum d’intensité de l’image originale est 30 et nous avons utilisé comme PSF le filtre moyennneur 7×7 . La transformée TI-DWT montrée dans la figure 4.9(d) conduit à un bon résultat de restauration sur les structures isotropes (points et gaussiennes), mais les lignes ne sont pas bien restaurées. Ce défaut peut être contrebalancé en utilisant la transformée en curvelets (figure 4.9(e)), mais, comme prévu, les sources ponctuelles peu lumineuses dans la partie en haut à gauche sont sacrifiées. Visuellement, l’utilisation d’un dictionnaire contenant les deux transformées semble conduire au meilleur des deux premiers résultats (figure 4.9(g)).

La figure 4.10 montre le MAE (—), ici normalisé selon l’intensité maximale de l’image d’origine pour avoir des échelles comparables, en fonction du maximum d’intensité pour trois images : l’image simulée de neurone, la cellule et l’image composée de lignes, de points et de gaussiennes (LigneGauss). Comme précédemment, les dictionnaires utilisés étaient : TI-DWT (—), curvelets (--) et le dictionnaire construit par la fusion des deux transformées (-..). L’algorithme de déconvolution utilisé était PoissonStab. Pour l’image simulée de neurone, qui est très « géométriques » et lisse par morceaux, la meilleure performance est donnée par le dictionnaire TI-DWT+curvelets aux intensités moyennes et élevées ; bien que les différences entre les dictionnaires soient moins prononcées à faible niveau d’intensité. Pour l’image de la cellule qui contient plein de structures localisées, la transformée TI-DWT semble conduire au meilleur résultat, surtout lorsque l’intensité augmente. Finalement, le comportement observé pour l’image LigneGauss est l’exact opposé de celui de la cellule. Plus précisément, les dictionnaires contenant les curvelets ou la fusion TI-DWT plus les curvelets montrent les meilleurs résultats avec un avantage pour le dernier. Cependant, il faut se garder de tirer des conclusions hâtives de cet ensemble limité d’expériences sur la construction d’un dictionnaire redondant. En effet, celui-ci doit bien s’adapter au contenu morphologique de l’image à restaurer, sinon le prix de la redondance peut être élevé.

4.6.2.2 Données réelles

Nous avons appliqué nos deux méthodes sur une image réelle 512×512 de neurone acquise avec un microscope confocal. Le dictionnaire contenait les deux transformées TI-DWT, pour les structures isotropes et curvelets, pour les parties curvilignes. Pour les deux méthodes, le GCV a été utilisé pour obtenir une valeur de γ . La figure 4.11(a) montre une image observée d'une arborescence dendritique avec la protéine fluorescente GFP. La PSF a été modélisée en utilisant l'approximation gaussienne introduite dans [Zhang et al., 2007a]. La figure 4.11(b) présente l'image restaurée en utilisant PoissonStab et la figure 4.11(c) PoissonNS. Les images sont montrées dans une échelle logarithmique pour une meilleure visualisation.

Globalement, l'algorithme PoissonNS produit une meilleure restauration. En effet, les structures curvilignes sont mieux préservées qu'avec PoissonStab et moins de trous sont présents dans la structure. Nous supposons que cette amélioration provient d'une attache aux données plus fidèle au bruit. PoissonStab introduit des artefacts qui sont produits par les transformées, contrairement à PoissonNS qui donne une image lisse. Nous supposons que ces résultats sont améliorables en utilisant une PSF plus précise, par exemple celle proposée par Booth et Wilson [2001] (section 3.2.1.2).

Les épines sont bien restaurées ainsi que l'illustre les figures 4.12(b), (c), (e) et (f). L'épine dans la figure 4.12(a) est intéressante car très curviligne ; nous pouvons constater que les curvelets préservent correctement la structure (figure 4.12(b) et (c)). L'arborescence dendritique est aussi restaurée comme le présente les figures 4.12(e) et (f). Les épines reliées à l'axe ressortent parfaitement dans la restauration, de même leurs points d'attache sur l'axe sont plus précis. Remarquons que PoissonNS produit des résultats plus lisses que PoissonStab, en effet des trous présents dans la figure 4.12(e) ne le sont pas dans la figure 4.12(f). Nous supposons que le GCV conduit à une surestimation du paramètre d'équilibre pour PoissonNS.

4.7 Conclusion

Combinant un cadre théorique pour l'optimisation convexe (chapitre 2) avec une analyse du fonctionnement d'un microscope confocal (chapitre 3), nous avons conçu plusieurs algorithmes de restauration pour des images dégradées par une convolution avec une PSF (flou introduit par le système optique) et contaminées par un bruit de Poisson. L'approche bayésienne a permis de mélanger la connaissance du contenu morphologique des images avec une attache aux données fidèle à la formation de l'image.

Nous avons proposé plusieurs algorithmes différents pour résoudre le problème d'optimisation qui en découle. Deux approches sont présentées : la première consiste à stabiliser la variance du bruit pour retrouver un contexte gaussien, la seconde prend directement le bruit

de Poisson en compte. Chacune implique un contexte différent pour l'application des outils provenant du cadre du chapitre 2. L'absence de la condition d'avoir un gradient Lipschitz dans la deuxième approche réduit le panel de schémas de résolution, mais nous avons montré qu'une telle approche est toujours possible.

Notre régularisation repose sur l'a priori que les images à restaurer sont parcimonieuses dans un dictionnaire redondant dont le choix reflète la diversité morphologique des structures contenues dans l'image. Les expériences montrent l'importance de choisir un dictionnaire adapté pour la restauration. Dans le cadre de cette thèse, nous devons restaurer des images de dendrites, or ces images présentent une morphologie bien particulière (voir chapitre 7) que nous pouvons prendre en compte avec ce concept.

Selon les résultats, la stabilisation semble effective dès que les intensités deviennent moyennes et les restaurations obtenues sont comparables à celles données par des algorithmes de l'état de l'art. Une validation à plus large échelle impliquant tout ce que nous avons proposé comme algorithmes serait toutefois nécessaire pour obtenir un plus large panorama de comparaison. De plus, nous avons essentiellement testé les approches avec un a priori synthèse, qui peuvent introduire des artefacts provenant des transformées.

Le choix du paramètre de régularisation γ est crucial dans notre cadre. Le GCV tel que nous le proposons, semble conduire à une estimation valable. Néanmoins, un post-réglage manuel peut s'avérer nécessaire pour parvenir à une bonne qualité visuelle, qui reste toutefois très objective. Une autre approche serait de formuler le problème sous une forme contrainte de l'attache aux données, à l'image de ce qui est communément fait en contexte gaussien. Le choix de la taille de la contrainte peut s'avérer plus facile que celui du paramètre de régularisation.

Autre point pour des recherches futures dans notre approche : la vitesse de convergence dont nous en avons fait déjà part lors de la conclusion du chapitre 2. Cette vitesse est à négocier avec la complexité temporelle de l'algorithme. Celle-ci est largement dominée par les appels au dictionnaire et à la transformée de Fourier. Un axe de recherche serait de trouver un compromis acceptable entre une complexité plus importante, mais avec un convergence plus rapide, donnant lieu à moins d'opérations.

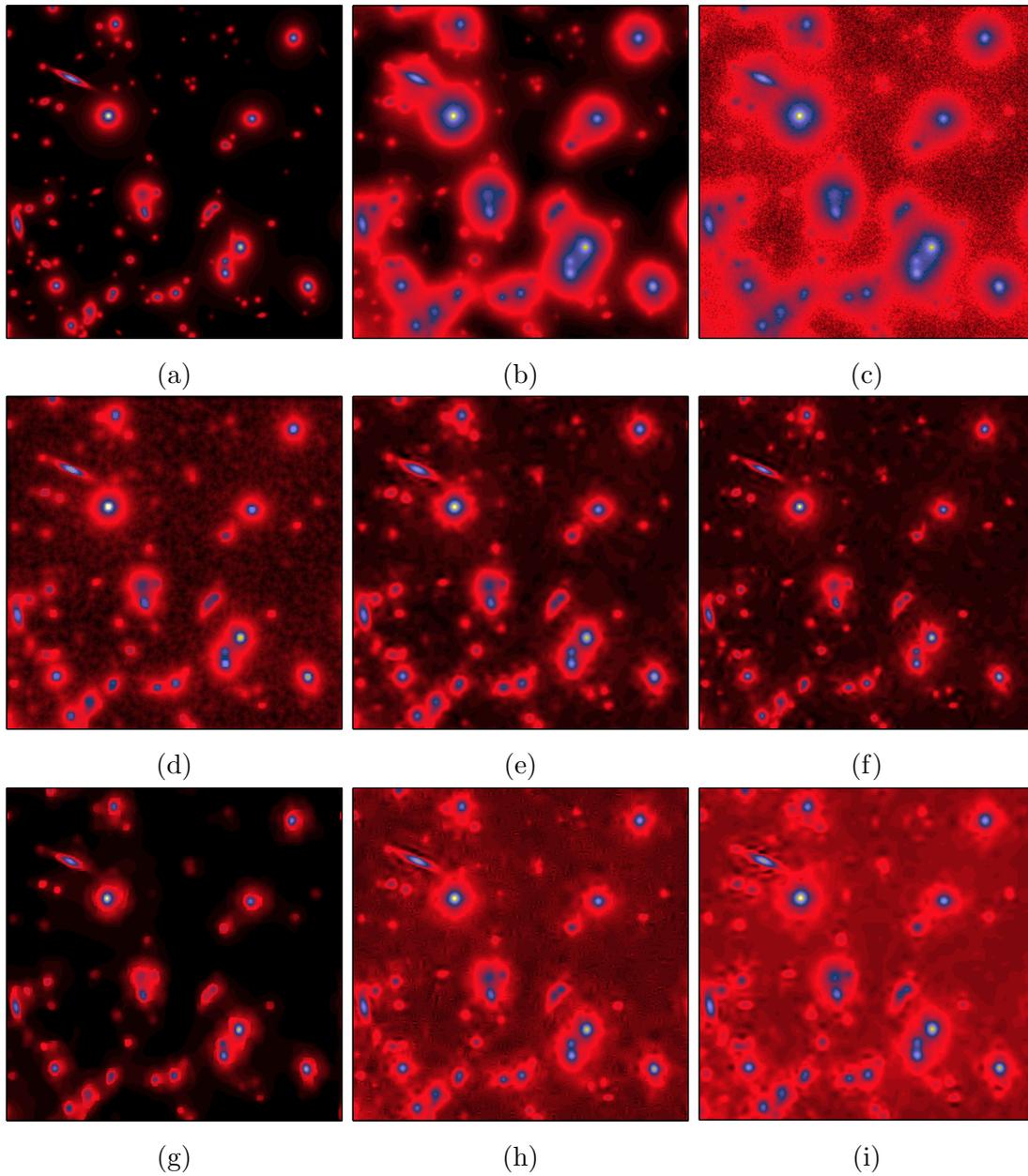


FIG. 4.7: Déconvolution de l'image simulée de ciel. (a) Originale, (b) Convoluée, (c) Convoluée & bruitée, (d) RL-TV [Dey et al., 2004], (e) NaiveGauss [Vonesch et Unser, 2007], (f) RL-MRS [Starck et Murtagh, 2006], (g) FTITPR [Willett et Nowak, 2004], (h) PoissonStab, (i) PoissonNS.

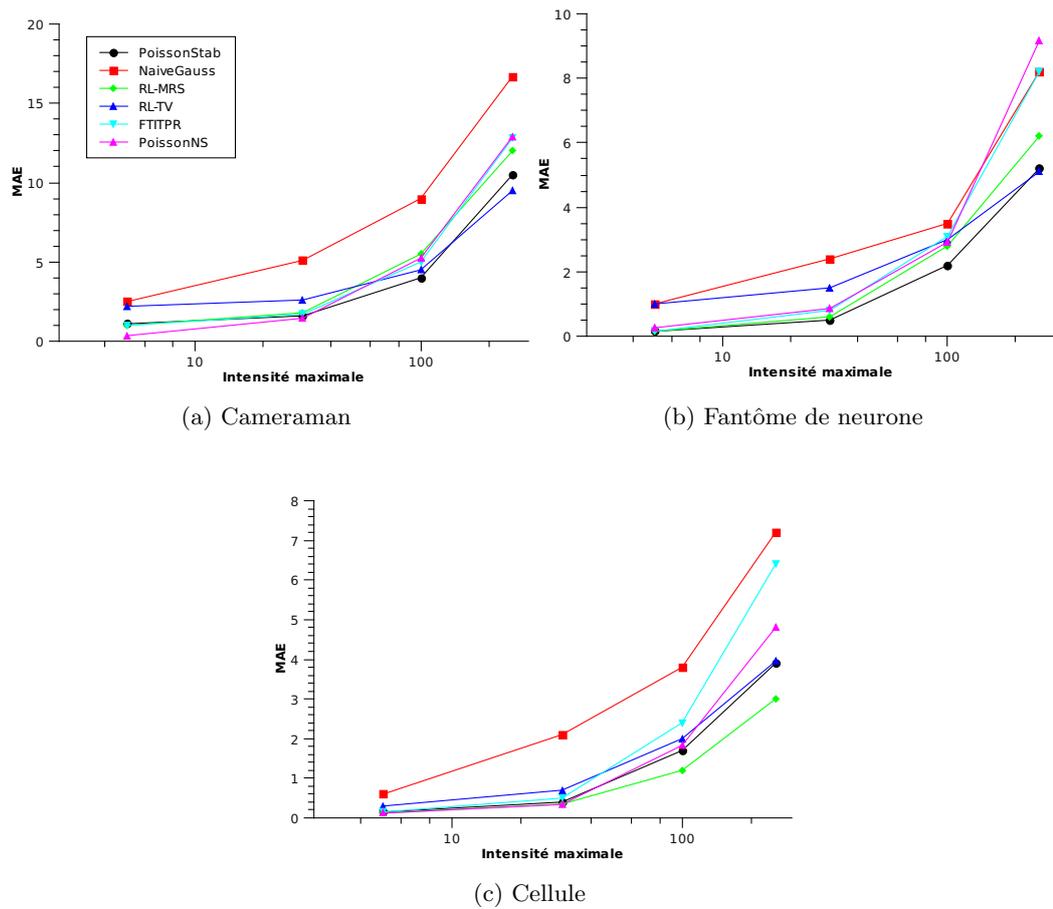


FIG. 4.8: MAE moyen pour tous les algorithmes en fonction du niveau d'intensité.

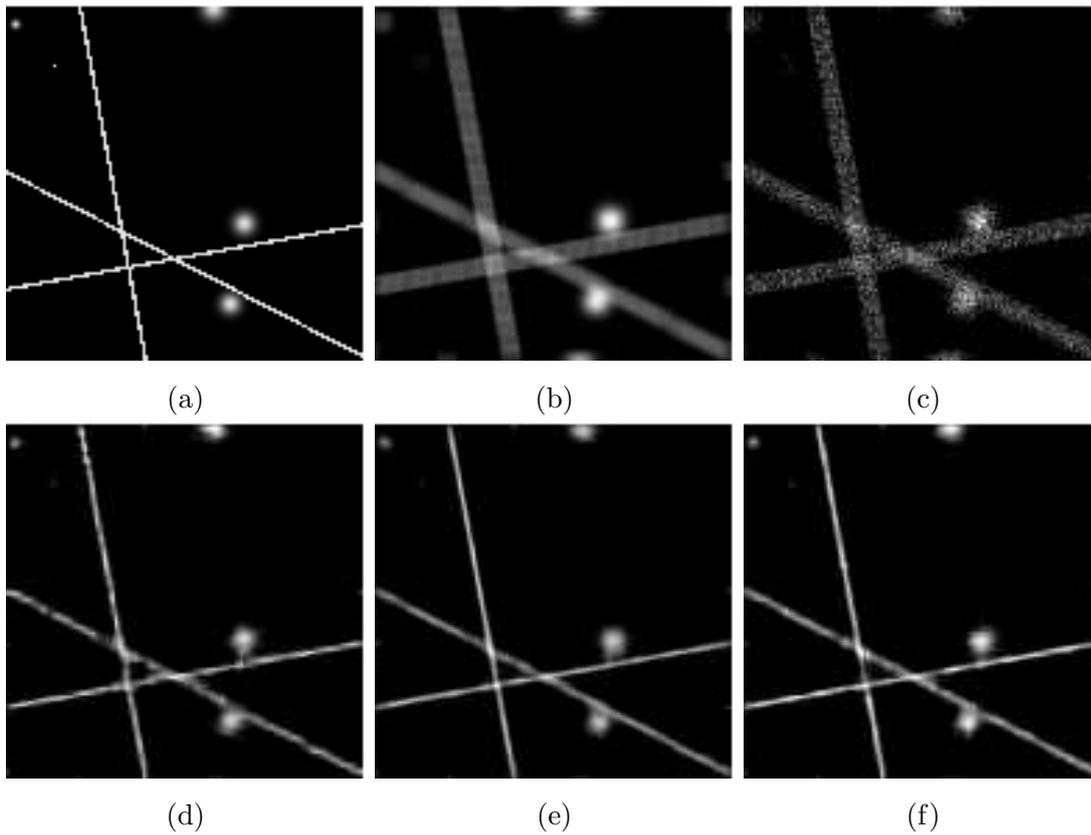


FIG. 4.9: Impact du dictionnaire sur la déconvolution d'une image simulée avec une intensité maximale de 30. (a) Originale, (b) Convoluée, (c) Convoluée & bruitée, (d) Restaurée avec TI-DWT, (e) Restaurée avec les curvelets, (f) Restaurée avec un dictionnaire contenant les deux transformées.

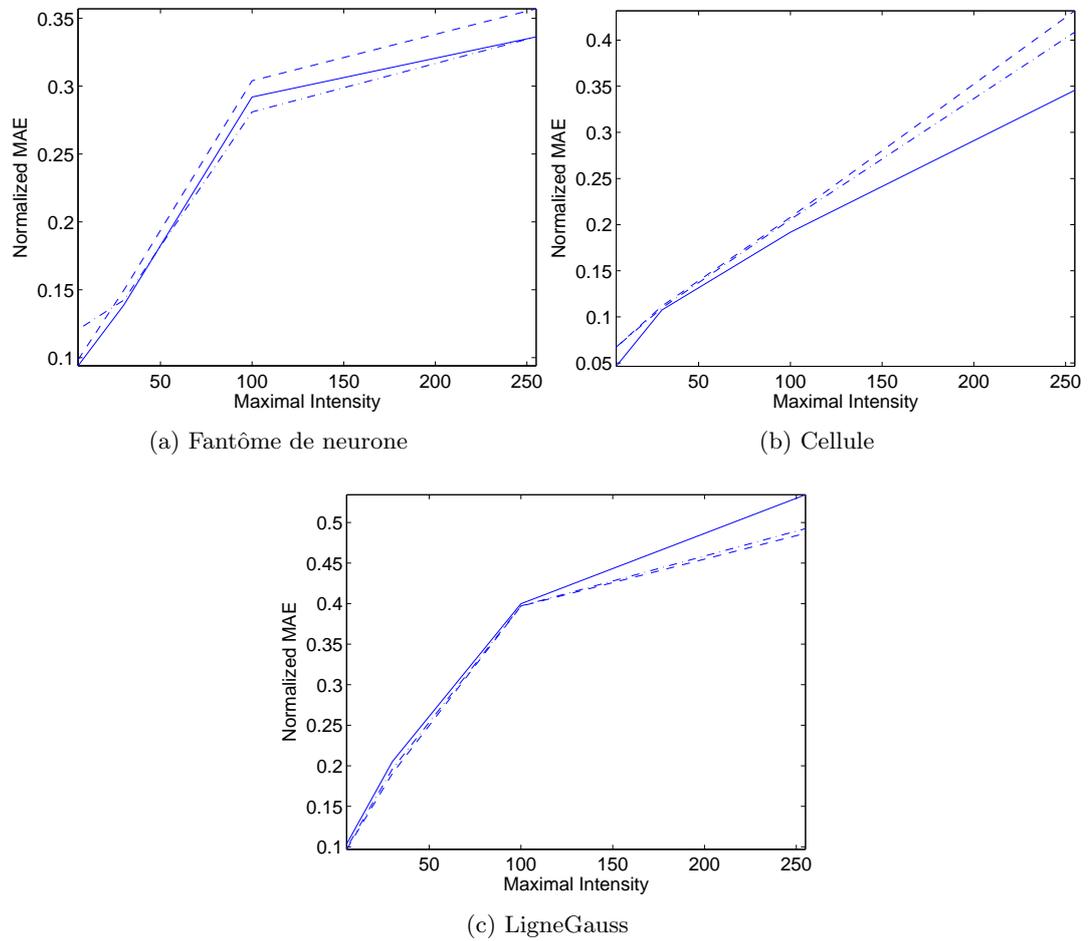
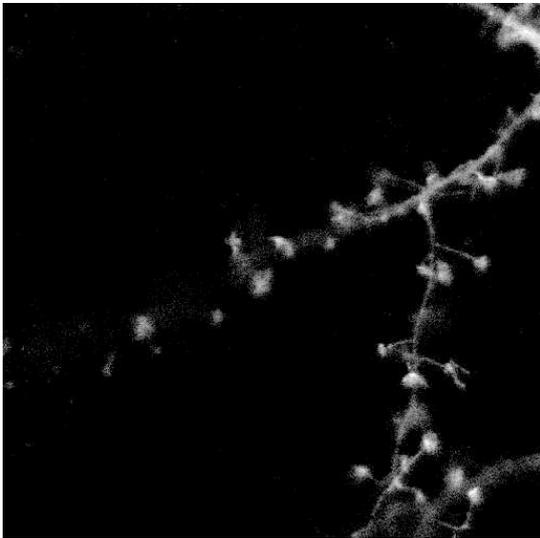
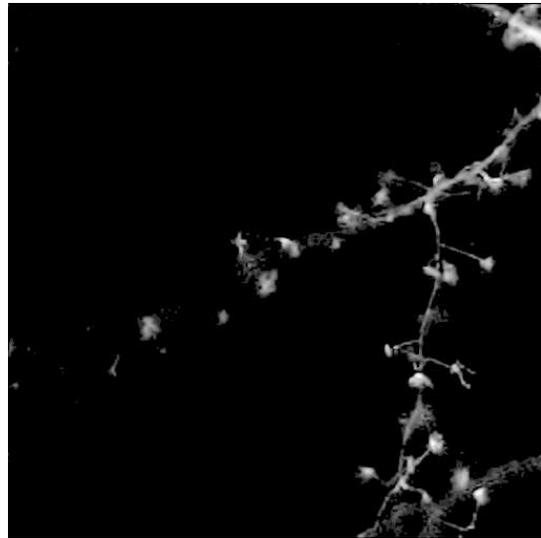


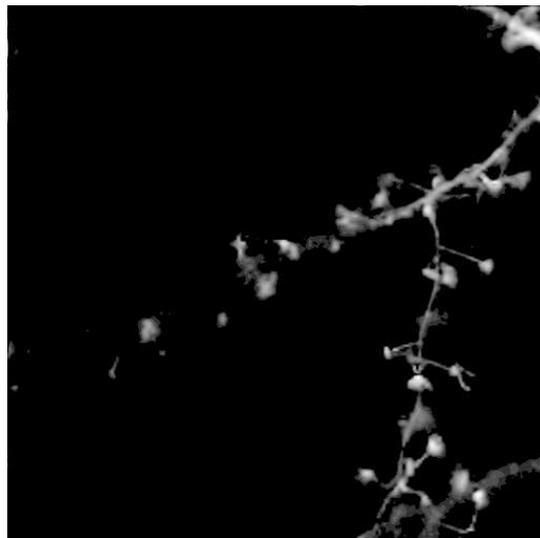
FIG. 4.10: Impact du dictionnaire sur les résultats de déconvolution en fonction du maximum d'intensité sur plusieurs images de test : (a) image simulée de neurone, (b) cellule et (c) LigneGauss. Les traits pleins représentent la transformée TI-DWT, les traits hachurées à la transformée en curvelets, et les traits en pointillés au dictionnaire construit par la fusion des deux transformées précédentes.



(a) Image bruitée originale

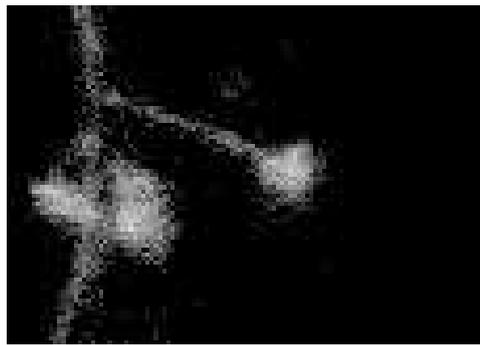


(b) Image restaurée avec PoissonStab

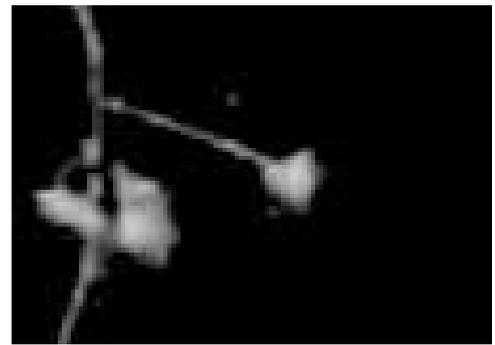


(c) Image restaurée avec PoissonNS

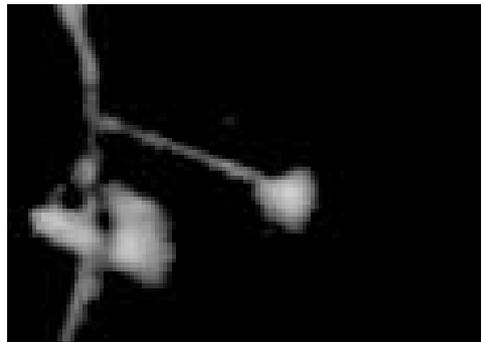
FIG. 4.11: Déconvolution d'un neurone réel. (a) Image originale bruitée, (b) Image restaurée avec PoissonStab, (c) Image restaurée avec PoissonNS.



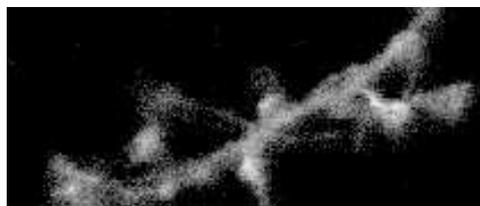
(a) Une épine dégradée



(b) L'épine restaurée (PoissonStab)



(c) L'épine restaurée (PoissonNS)



(d) Un axe dendritique dégradé



(e) L'axe restauré (PoissonStab)



(f) L'axe restauré (PoissonNS)

FIG. 4.12: Détails de la restauration de l'axe dendritique. (a) Une épine dégradée, (b) et (c) sa restauration avec respectivement PoissonStab et PoissonNS. (d) Un morceau de l'axe dendritique avec des épines, (e) et (f) sa restauration avec respectivement PoissonStab et PoissonNS.

Deuxième partie

Noyaux sur graphes pour la classification de formes

5 Problématique et état de l'art

Tout est dans tout, et vice et versa.

Alphonse Allais

La reconnaissance des formes est l'un des problèmes fondamentaux en vision par ordinateur. En effet, de nombreuses problématiques nécessitent de classer ou d'indexer des objets. « Quelle est cette forme ? », « Quelles sont les formes semblables à celle-ci ? » sont deux questions maintes fois posées. Ces questions usuellement très ardues en informatique le sont également pour le système visuel humain où même un individu entraîné peut avoir des difficultés pour distinguer deux formes.

Une représentation numérique des formes est nécessaire pour les manipuler avec les outils informatiques. La représentation numérique la plus usuelle pour une forme est un ensemble de pixels. Cette représentation a l'avantage d'être visuelle, mais elle manque de souplesse et varie selon les transformations classiques que sont la translation, la rotation et le changement d'échelle. D'autres types de représentations sont alors préférés en se basant sur le contour, les moments géométriques ou le squelette de la forme. Ces différents codages cherchent à répondre à la fois au besoin d'informations discriminantes et à celui de robustesse au bruit et aux déformations. Le choix du codage dépend alors fortement de l'application.

Une fois la représentation numérique choisie, la comparaison des formes est effectuée par un algorithme basé sur cette représentation. De nombreuses méthodes sont disponibles pour résoudre les problèmes d'appariement de formes : distance d'édition, mesure de similarité, etc. Elles sont généralement robustes aux déformations et au bruit mais ne vérifient pas les propriétés nécessaires pour être de véritables distances. Les méthodes de classification sont, elles aussi, nombreuses : réseaux de neurones, machines à noyaux (SVM, kPCA...), k plus proches voisins, etc. Certaines d'entre elles opèrent directement dans des espaces de Hilbert. D'autres sont complètement indifférentes par rapport aux propriétés de la technique de comparaison (par exemple les k plus proches voisins).

Dans ce chapitre, nous présentons différentes représentations numériques de formes. Puis, plusieurs méthodes d'appariement et de classification basées sur ces représentations sont exposées. Finalement, nous discuterons leurs avantages et inconvénients.

5.1 Codages des formes

Diverses méthodes de codage numérique de formes sont proposées dans la littérature, chacune avec un domaine de prédilection. Par exemple, les contours forment une caractéristique importante des formes et deux formes semblables possèdent généralement des contours similaires (au moins à une certaine échelle). Par ailleurs, les moments géométriques permettent de représenter une forme à l'aide d'un vecteur de données, facilitant l'utilisation d'outils mathématiques classiques. Enfin, le squelette permet de coder les formes par des graphes ouvrant ainsi toute la richesse et la complexité de cette structure de données à la description de formes.

5.1.1 Contour d'une forme

Le contour est une représentation linéaire d'une forme fermée. Si elle présente des trous, ceux-ci donnent des contours supplémentaires. Le contour peut présenter des motifs très discriminants pour certaines formes telles que les feuilles d'arbres. La comparaison des contours peut être effectuée par des différences entre les coefficients codant deux contours dans une base appropriée [Shen et Shen, 1996; Brun et Pruvot, 2008] ou par un processus d'alignement correspondant à une distance d'édition [Bunke et Buhler, 1993; Gdalyahu et Weinshall, 1999; Kaygin et Bulut, 2002]. Pour être robustes aux déformations, ces distances doivent être robustes au bruit et être capables d'ignorer les perturbations apportées par les occlusions.

Par exemple, les contours peuvent être comparés selon l'évolution de la position des points : soient $(a_i)_{i \in \mathbb{N}}, (b_i)_{i \in \mathbb{N}} \in \mathbb{R}^2$ deux suites finies représentant la succession des positions des points du contour, la distance d'édition entre les contours est classiquement définie par :

$$d(i, j) = \min \{d(i-1, j-1) + \delta(a_i, b_j), d(i-1, j) + c_r, d(i, j-1) + c'_r\} \quad (5.1)$$

avec i et j les positions courantes dans les suites $(a_i)_{i \in \mathbb{N}}$ et $(b_j)_{j \in \mathbb{N}}$, $\delta(a_i, b_j)$ une mesure de similarité entre les points a_i et b_j , c_r et c'_r des coûts de sauts.

L'avantage d'une représentation par contour est de ramener la comparaison de formes 2D à une mesure de similarité entre courbes 1D. De plus, ce type de représentation permet de capturer aisément des motifs particuliers sur les contours qui seraient ignorés par d'autres types. Cependant, contrairement à d'autres approches, le codage du bord d'un objet ne fournit pas une décomposition intuitive de celui-ci en sous-parties. Celle-ci peut pourtant s'avérer utile pour des comparaisons « naturelles » de formes articulées.

5.1.2 Moments géométriques d'une forme

Les moments permettent de décomposer une forme en une série numérique finie ou non. Certains moments possèdent des propriétés très utiles dans le cadre de la comparaison de

formes : ils sont invariants par rotations, translations et changements d'échelles. Parmi les méthodes basées sur les moments, nous trouvons : les moments géométriques, les moments de Legendre [Mukundan et Ramakrishnan, 1998], les moments de Zernike [Novotni et Klein, 2004], etc. D'autres descripteurs ont aussi été introduits comme la transformée radiale angulaire [Kim et Kim, 1999] proposée comme attribut de région par la norme MPEG-7. Cette transformée se révèle très stable aux perturbations et performante pour la comparaison de formes.

Prenons, par exemple, les moments géométriques :

$$m_{pq} = \int \int_{\Omega} x^p y^q dx dy, \quad p, q \in \mathbb{N} \quad (5.2)$$

avec m_{pq} le moment géométrique d'ordre p et q , Ω est un domaine compact de \mathbb{R}^2 dans lequel se trouve la forme. Ces moments doivent être normalisés pour être invariants aux rotations, translations et changements d'échelles. Ils peuvent être interprétés comme une projection de la forme sur les fonctions de bases $\{x^p y^q\}$, or cette base n'est pas orthogonale et n'est donc pas adaptée pour une comparaison directe des formes à l'aide de vecteurs de coefficients. En revanche, les moments de Legendre ou de Zernike proposent une projection sur une base orthogonale, fournissant ainsi une reconstruction non redondante de la forme à partir des coefficients.

Les moments permettent de représenter une forme à l'aide d'un vecteur dans l'espace transformé. Cela ouvre de fait l'utilisation de tous les classifieurs disponibles dont les séparateurs à vaste marge (SVM). Cependant, une représentation complète des formes demande des vecteurs de tailles infinies, or ils sont habituellement tronqués en fixant un nombre maximal de moments. Cette perte d'information peut amoindrir les performances des algorithmes de comparaisons. De plus, les moments fournissent essentiellement une description globale de la forme et donc ne sont pas adaptés si les différences entre deux formes tiennent sur des détails.

5.1.3 Codages par graphes

Les contours et les moments géométriques proposent une représentation de la forme qui n'offre pas un découpage direct en sous-parties indépendantes. Un tel découpage permet d'obtenir une robustesse aux déformations telles que les occlusions. En effet, une occlusion apporte et/ou supprime partiellement ou totalement des sous-parties. Un découpage de la forme permet donc une meilleure robustesse aux déformations et potentiellement au bruit.

Nous développons plus spécifiquement la représentation par graphes construits à partir de squelettes. Celle-ci permet une partition des formes en sous-parties. Le problème de comparaison de formes devient donc un problème de comparaison de graphes, or celui-ci est difficile, surtout en présence de bruit au sein de la structure des graphes.

5.2 Définitions relatives aux graphes

Nous introduisons dans cette section un certain nombre de définitions relatives aux graphes. Elles proviennent essentiellement de trois ouvrages de références : [Roy, 1969; Harary, 1969; Tutte, 1984]. Cette section étant composée de définitions, celle-ci peut être ignorée par les lecteurs déjà familiarisés avec ce vocabulaire.

Définition 5.2.1 (Graphe).

Un graphe est une paire $G = (V, E)$ d'ensembles tels que $E \subseteq V \times V$. Les éléments de V sont appelés des sommets et ceux de E des arêtes.

Les deux sommets d'une arête $e = (a, b)$ sont dit adjacents, l'arête e est dite incidente aux sommets a et b . Une arête incidente à un même sommet ($e = (a, a)$) est appelée une boucle.

Définition 5.2.2 (Sous-graphe).

Un sous-graphe d'un graphe $G = (V, E)$ est un graphe $G' = (V', E')$ tel que $V' \subseteq V$ et $E' \subseteq (V' \times V') \cap E$.

Définition 5.2.3 (Graphe simple, Multigraphe, Pseudographe).

- Un graphe G est dit simple s'il ne possède pas de boucles (sommets reliés à eux-mêmes) ni de couples de sommets joints par plusieurs arêtes.
- Un graphe G est appelé un multigraphe si un couple de sommets distincts peut être joint par plusieurs arêtes et s'il ne possède pas de boucles.
- Un graphe G est appelé un pseudographe si un couple de sommets distincts peut être joint par plusieurs arêtes et s'il possède des boucles.

Dans la suite de ce document et sauf mention contraire, le terme générique *graphe* désignera un pseudographe.

Nous considérons maintenant un ensemble de parcours spécifiques dans les graphes.

Définition 5.2.4 (Chemin).

Un chemin est une suite alternée de sommets et d'arêtes, commençant et finissant par un sommet et telle que chaque arête soit incidente aux deux sommets qui la précède et la suit immédiatement. La taille d'un chemin est définie par son nombre d'arêtes.

Définition 5.2.5 (Chemin simple).

Un chemin est dit simple si ses arêtes sont toutes distinctes.

Définition 5.2.6 (Chemin élémentaire).

Un chemin est dit élémentaire si ses sommets et arêtes sont tous distincts.

Dans la terminologie proposée par Berge [1958], les chemins sont définis pour des graphes orientés. Pour les graphes non orientés, il parle de parcours. Pour des raisons de clarté, nous avons préféré utiliser le terme « chemin » même si nos graphes ne sont pas nécessairement orientés.

Définition 5.2.7 (Chemin hamiltonien).

Un chemin est dit hamiltonien s'il passe une seule fois par tous les sommets du graphe.

La caractérisation d'un graphe possédant un chemin hamiltonien est assez complexe, il n'existe pas de critère nécessaire et suffisant pour caractériser l'existence de tels chemins. En revanche, des critères suffisants sont disponibles. Il est à noter que le problème du chemin hamiltonien qui consiste à trouver un tel chemin dans un graphe est connu pour être NP-complet. Celui-ci est lié à celui, plus connu, du voyageur de commerce.

Définition 5.2.8 (Chemin eulerien).

Un chemin est dit eulerien s'il passe une seule fois par toutes les arêtes du graphe.

Définition 5.2.9 (Cycle).

Un cycle est un chemin dont le premier sommet et le dernier sommet sont identiques. Un cycle est dit simple si les sommets (en dehors du premier et du dernier) sont distincts, et élémentaire si en plus les arêtes sont distinctes.

Définition 5.2.10 (Graphe connexe).

Un graphe est dit connexe si pour tout couple de sommets distincts, il existe un chemin joignant les deux sommets.

Définition 5.2.11 (Arbre).

Un arbre est un graphe connexe ne possédant pas de cycle élémentaire.

Définition 5.2.12 (Graphe planaire).

Un graphe planaire est un graphe $G = (V, E)$ possédant un plongement dans \mathbb{R}^2 tel que ses sommets sont des points distincts et ses arêtes sont des courbes dont l'intérieur (tous les points de la courbe sauf les deux extrémités) ne contiennent aucun point appartenant à un sommet ou à une autre arête.

Définition 5.2.13 (Face d'un graphe planaire).

Une face d'un graphe planaire G est une région de plan limitée par des arêtes et dont l'intérieur ne contient ni sommets ni arêtes.

Définition 5.2.14 (Graphe labélisé).

Un graphe labélisé $G = (V, E, \mu, \nu, L_v, L_e)$ est défini par un ensemble de sommets V , un ensemble d'arêtes $E \subseteq V \times V$ et par deux fonctions associant des labels aux sommets et aux

arêtes :

$\mu : V \rightarrow L_v$ fonction associant à un sommet un élément de l'ensemble L_v ,

$\nu : E \rightarrow L_e$ fonction associant à une arête un élément de l'ensemble L_e .

5.3 Graphes caractérisant une forme

Les graphes forment une structure de données très adaptée pour représenter un voisinage, or nous comptons décomposer les formes en sous-parties reliées entre elles. Cependant, une telle décomposition demande un outil de détection de ces sous-parties. La squelettisation, par exemple, définit implicitement une division selon la méthode utilisée. Ainsi les chocs produits lors du processus de création du squelette, en utilisant la définition de Blum [1967], fournissent une telle décomposition.

5.3.1 Définition d'un squelette

Plusieurs définitions existent, néanmoins nous avons préféré prendre celle qui sert de base pour la plupart des travaux sur l'appariement de formes [Blum, 1967; Siddiqi et al., 2002; Sebastian et al., 2004] :

Définition 5.3.1 (Squelette).

Soient X l'ensemble des points d'une forme \mathcal{F} quelconque, $Sq(X)$ l'ensemble de point tels que : pour tout point $x \in Sq(X)$, x est le centre d'une boule \mathcal{B}_x de rayon maximal $\delta > 0$ tel que $\mathcal{B}_x \subset X$ et \mathcal{B}_x est tangente en au moins deux points distincts à la bordure de la forme. Un tel ensemble $Sq(X)$ est appelé le squelette associé à la forme \mathcal{F} .

Ce squelette possède les propriétés suivantes [Blum, 1967] :

- il est fin (son épaisseur est d'un pixel),
- il est homotope à la forme (donc un trou conduit à une boucle),
- la forme peut être reconstruite à partir de son squelette et des boules de rayon maximal.

La figure 5.1 montre le squelette d'une main. Ici les différentes branches permettent de repérer les parties de la main dont les doigts. Des études [Kovács et al., 1998; Siddiqi et al., 2001] ont souligné le lien entre le squelette d'une forme et la psychologie cognitive pour la reconnaissance des formes. En effet, notre cerveau décompose naturellement les formes en sous-formes, facilitant ainsi la reconnaissance d'objets articulés ou déformés.

Blum [1967] présente un premier algorithme de calcul du squelette basé sur l'analogie du feu de forêt : imaginons que la bordure (i.e. le contour) de la forme est en feu et consomme la forme à vitesse égale et constante, le squelette est alors formé par les points où plusieurs fronts de propagation du feu se rejoignent. Cependant, cet algorithme suppose que la forme

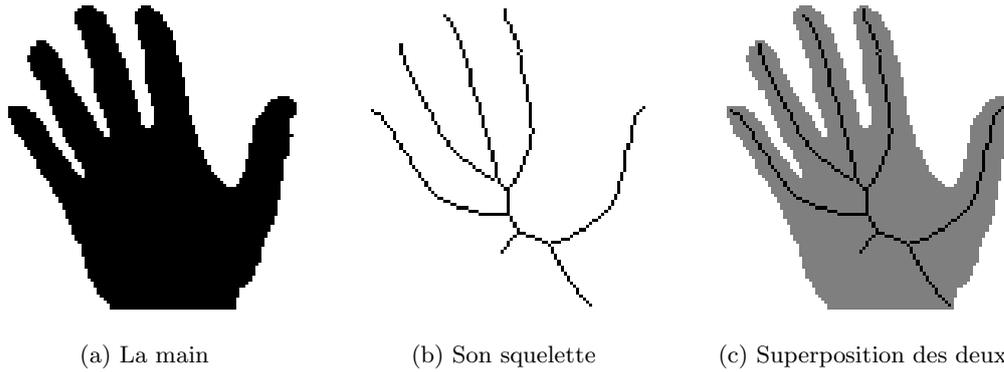


FIG. 5.1: Une main et son squelette

soit définie en continue, or généralement l'opération est effectuée dans un espace fini (donc dans le domaine discret) impliquant un bruit de quantification (souvent situé sur le contour des formes). Par définition, les algorithmes de squelettisation sont sensibles au bruit introduit par l'étape de discrétisation et aux déformations.

Plusieurs méthodes de squelettisation proposent des heuristiques pour être robuste au bruit de quantification. Par exemple, Pudney [1998] combine un seuil avec la fonction distance associée à la forme. Par ailleurs, Siddiqi et al. [2002] utilisent un algorithme basé sur le flux de la fonction distance associée à la forme et supposent que les points du squelette soient situés sur des zones particulières de ce flux définies avec l'aide d'un seuil ajustable. Cette méthode a ensuite été améliorée par Torsello et Hancock [2003] qui approfondissent et corrigent le calcul du flux. Couprie et al. [2007] définissent une fonction, robuste au bruit, comme critère d'extraction du squelette. Notons que ces trois méthodes sont aussi bien définies en 2D qu'en 3D.

La définition du squelette est parfois adaptée dans l'optique de corriger les principaux défauts que sont la sensibilité aux déformations et au bruit, tout en conservant une structure linéaire. Par exemple, Goh [2008] propose un squelette pour des formes 2D qui n'est pas homotope à la forme, mais qui conserve la propriété de reconstruction. De plus son algorithme est moins sensible aux déformations. Aslan et al. [2008] proposent un squelette déconnecté pour les formes 2D qui a l'avantage de ne pas présenter les défauts des squelettes homotopiques traditionnels. Ces méthodes servent de base à des algorithmes de comparaison robustes aux déformations et aux bruits. Cependant elles sont, pour le moment, définies uniquement pour des formes 2D et ne permettent pas de s'affranchir du bruit structurel présent dans les graphes codant les squelettes.

5.3.2 Codage d'un squelette par graphes

Parmi la littérature, deux méthodes dominent pour le codage du squelette par un graphe : le graphe *direct* où la structure du squelette est directement traduite en graphe, ou le graphe de chocs (« shock graph ») où la structure du squelette est enrichie avec les singularités de la fonction distance le long de l'axe médian. L'information structurelle portée par ces deux types de graphes peut être complétée par des attributs numériques correspondant à des descripteurs globaux ou locaux d'arêtes et de sommets :

Définition 5.3.2 (Descripteur local).

Un descripteur local décrit l'objet auquel il est associé indépendamment des autres objets ou de l'ensemble dans lequel l'objet s'inclut.

Un descripteur local associé à une branche d'un squelette sera donc défini à partir des points de la branche ou de la région associée à celle-ci (voir section 6.1.2.2).

Définition 5.3.3 (Descripteur global).

Un descripteur global décrit l'objet auquel il est associé en prenant en compte les autres objets inclus avec lui dans l'ensemble.

5.3.2.1 Le graphe direct

Le codage *direct* consiste à coder les branches du squelette par des arêtes, les jonctions et terminaisons par des sommets. Le graphe est ensuite enrichi par d'autres informations telles que la longueur des branches, la distance des points au centre de gravité pour les sommets [Ruberto, 2004; Torsello et Hancock, 2004; Suard et al., 2006; Goh, 2008], etc. Le choix des attributs répond à une recherche d'équilibre entre descripteurs locaux et descripteurs globaux tels que définis dans 5.3.2 et 5.3.3. Comme exemples de descripteurs locaux, nous trouvons la longueur d'une branche, l'évolution du rayon du cercle maximal le long d'une branche, etc. Ainsi, un grand nombre d'attributs relatifs uniquement aux branches et à leurs zones d'influences (l'ensemble des points plus proches de la branche considérée que des autres branches) sont disponibles. De même, comme exemple de descripteurs globaux, nous avons le pourcentage de la forme représentée par la zone d'influence associée à une branche, la distance des sommets au centre de gravité, etc. Ce choix dépend fortement de l'application et des formes considérées [Goh, 2008].

La figure 5.2(a) donne un exemple de graphe direct pour une main. Nous pouvons remarquer que les différentes parties de la forme, par exemples les doigts, ne ressortent pas beaucoup dans le graphe. Si nous regardons de plus près, nous constatons que l'arête codant un doigt code aussi un morceau de la paume de la main. Ce défaut est encore plus marqué dans la figure 5.2(b) où l'occlusion n'apparaît pas clairement dans le graphe. L'importance de ces défauts peut être

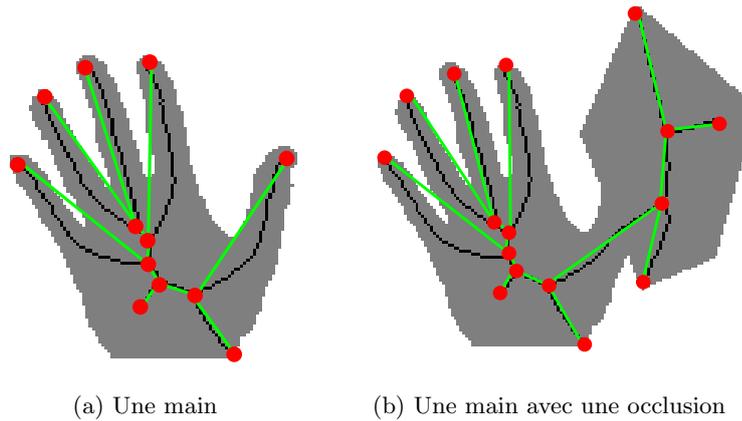


FIG. 5.2: Les graphes directs de deux mains.

amenuisée en changeant la méthode de construction du graphe, par exemple en utilisant celle proposée par le graphe de chocs.

5.3.2.2 Le graphe de chocs

Reprenons le modèle de construction du squelette par propagation du feu de forêt, les points du squelette correspondent aux rencontres entre plusieurs fronts de propagations. Le squelette correspond alors aux zones de rencontres qui sont les zones de chocs entre plusieurs fronts. Siddiqi et Kimia [1996] proposent de classer les différentes zones de chocs en quatre catégories. Ce classement peut être aussi défini à partir de l'évolution du rayon des boules de rayon maximal le long du squelette :

le choc de type 1 est formé par une branche où le rayon des boules évolue de façon continue et monotone avec une vitesse non nulle (figure 5.3(a)),

le choc de type 2 est formé par un minimum dans l'évolution des rayons le long du squelette (figure 5.3(b)),

le choc de type 3 est formé par une branche où le rayon est constant (figure 5.3(c)),

le choc de type 4 est formé par un maximum dans l'évolution des rayons le long du squelette (figure 5.3(d)).

Siddiqi et Kimia [1996] montrent que les chocs forment une grammaire unique pour chaque forme. Les auteurs proposent donc d'utiliser les graphes de chocs pour reconnaître et classer les formes. Après avoir repéré les *chocs*, le graphe est construit en codant les chocs de type 1 et 3 par des arêtes et les chocs de type 2 et 4 par des sommets. Des sommets de type 1 sont aussi ajoutés entre les changements de chocs si nécessaire. Ensuite, le graphe est enrichi au niveau

des arêtes avec l'évolution du rayon et de l'orientation, ainsi que l'instant de formation et la position des points le long de la branche considérée.

Les arêtes du graphe sont enfin orientées pour correspondre à une évolution décroissante du rayon le long du squelette. Les sommets représentant les chocs de type 2 sont des sommets dont les arêtes incidentes sont toutes dirigées vers le sommet et inversement pour les sommets représentant les chocs de type 4. Cette orientation conduit alors à la construction d'un graphe orienté sans boucle. Notons que l'orientation simplifie la lecture du graphe, mais le rend plus sensible aux erreurs de détection de chocs. En effet, la différence entre un choc de type 1 et un choc de type 3 peut devenir assez faible du fait des erreurs d'approximation ; l'introduction d'une tolérance est alors requise pour être robuste à ces erreurs.

Pour des raisons de complexité algorithmique, Siddiqi et al. [1999] proposent de convertir le graphe de chocs en un arbre de chocs afin de transformer le problème d'appariement de graphes en celui, plus facile, d'appariement d'arbres enracinés. Pour cela, les sommets de type 2 sont dupliqués et les arêtes parvenant à ces sommets séparées pour obtenir un arbre. Ensuite, un sommet fictif est créé et relié à tous les chocs dont les arêtes incidentes sont toutes dirigées vers l'extérieur. Finalement, l'arbre est enraciné à ce sommet fictif. Le désavantage de cette technique est la perte de toutes les informations sur les trous présents dans la forme.

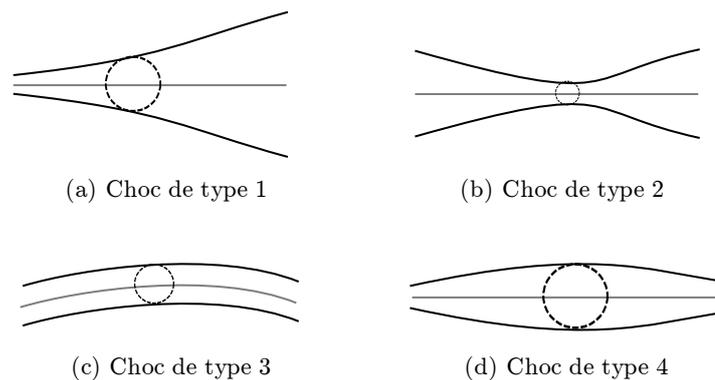


FIG. 5.3: Les différents types de chocs (représentés par des cercles)

Les chocs permettent de découper la forme en plusieurs parties. Celles-ci peuvent être vues comme des briques élémentaires. Cette décomposition possède des liens avec la psychologie cognitive [Siddiqi et al., 2001]. Cependant, la détection précise des chocs est assez délicate, car ils peuvent être localisés entre plusieurs pixels. Pour cela Siddiqi et Kimia [1996] proposent une méthode de détection sous-pixelique, mais celle-ci peut toutefois introduire des chocs erronés qui peuvent ensuite perturber les comparaisons.

La figure 5.4 montre deux graphes de chocs de mains dont l'une est déformée par une occlusion. Le graphe de chocs de la main déformée (figure 5.4(b)) montre l'intérêt des chocs. En effet, la structure du graphe permet une séparation claire des différentes sous-parties de la

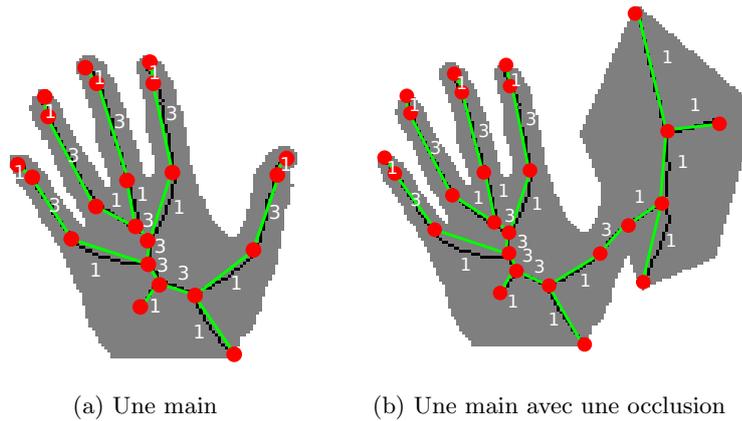


FIG. 5.4: Les graphes de chocs de deux mains.

forme, ce qui n'était pas le cas avec le graphe direct (figure 5.2). De plus, le graphe de chocs de la main (figure 5.4(a)) partage un sous-graphe assez important avec la main déformée, preuve de la grande similarité entre les deux formes.

Une extension à la 3D de la grammaire des chocs a été proposée par Leymarie et Kimia [2001] en introduisant de nouvelles catégories de chocs et de nouvelles règles. Le graphe ne pouvant coder toute la richesse de ce nouveau cadre, les auteurs utilisent un hypergraphe afin de modéliser toutes les adjacences créées par les surfaces du squelette.

5.4 Similarité entre graphes

Les formes étant codées par des graphes, le calcul de similarité entre deux formes est transformé en calcul de similarité entre deux graphes. Ce calcul de similarité implique des problèmes NP-complets : plus grand sous-graphe commun [Diestel, 2005], plus grande clique dans le graphe d'association [Pelillo, 1998] ou distance d'édition [Siddiqi et al., 1999; Sebastian et al., 2004; Torsello et Hancock, 2004]. Les algorithmes doivent de plus être robustes au bruit structurel provenant du processus de squelettisation et des déformations subies par les formes.

La complexité temporelle des algorithmes oblige usuellement à considérer des problèmes plus « simples », mais suffisant pour une tâche de classification. De telles méthodes sont souvent basées sur des heuristiques dépendantes des données à classer, du coup perdant la généralité d'une approche directe pour un gain de temps et de performance. En effet, une approche spécifique permet de mettre en place des mécanismes fortement discriminant dans un cadre applicatif précis.

5.4.1 Méthodes spectrales

Une première approche puise dans la théorie spectrale [Shokoufandeh et al., 1999; Shokoufandeh et Dickinson, 2000; Wilson et Zhu, 2008] pour comparer les graphes. Celle-ci propose des outils puissants et efficaces pour la comparaison de graphes. De plus, des liens entre la théorie spectrale et les marches aléatoires ont été développés [Chung, 1997], montrant par exemple que certaines valeurs propres correspondent à des chemins particuliers dans les graphes.

Nous présentons comme exemple une méthode de comparaison utilisant les valeurs propres des matrices codant les graphes. Supposons que deux graphes soient représentés par l'intermédiaire de deux matrices X_1 et X_2 . La décomposition de ces deux matrices en valeurs propres et vecteurs propres donne : $X_1 = \mathbf{W}_1 \mathbf{\Lambda}_1 \mathbf{W}_1^T$ et $X_2 = \mathbf{W}_2 \mathbf{\Lambda}_2 \mathbf{W}_2^T$ avec $\mathbf{\Lambda}_i = \text{diag}(\lambda_i^1, \dots, \lambda_i^{|V_{G_i}|})$, $i \in \{1, 2\}$ les matrices diagonales contenant les valeurs propres dans l'ordre décroissant et \mathbf{W}_i , $i \in \{1, 2\}$ les matrices formées par les vecteurs propres. Les spectres des matrices sont alors formés par les vecteurs contenant les valeurs propres : $s_i = \{\lambda_i^1, \dots, \lambda_i^{|V_{G_i}|}\}$. La distance entre les deux graphes est ensuite directement déduite de la distance euclidienne entre les deux spectres :

$$d_{\text{spectre}}(G_1, G_2) = \sqrt{\sum_i (s_1[i] - s_2[i])^2}. \quad (5.3)$$

Si les deux vecteurs n'ont pas la même taille, le plus petit vecteur est mis à la taille du plus grand en comblant la fin du vecteur avec des zéros.

Les performances de cette méthode dépendent de la représentation matricielle des graphes. Wilson et Zhu [2008] proposent plusieurs représentations dont :

- la matrice d'adjacence : soit un graphe $G = (V, E)$, la matrice d'adjacence associée à G est :

$$\mathbf{A}(u, v) = \begin{cases} 1, & \text{si } (u, v) \in E \\ 0, & \text{sinon} \end{cases}, \quad (5.4)$$

- la matrice du laplacien combinatoire : soit un graphe $G = (V, E)$, la matrice du laplacien combinatoire associée à G est :

$$\mathbf{L}(u, v) = \begin{cases} -1, & \text{si } (u, v) \in E \\ d_u, & \text{si } u = v \\ 0, & \text{sinon} \end{cases}, \quad (5.5)$$

avec d_u le degré du sommet u .

D'autres représentations sont disponibles ; nous conseillons de lire [Wilson et Zhu, 2008] pour plus d'informations.

Cette théorie propose des méthodes de comparaisons de graphes simples et rapides. Cependant, quelque soit la représentation choisie, il existe des graphes différents possédant les mêmes

spectres, ces graphes sont dit cospectraux. Le choix de la représentation va définir la quantité de graphes cospectraux ainsi que la résistance au bruit structurel.

Les désavantages tiennent essentiellement en deux points : 1) malgré les liaisons existantes entre structure et valeurs spectrales, il est difficile de savoir ce qui est réellement comparé au niveau structurel, 2) les représentations matricielles des graphes sont souvent limitées aux graphes faiblement attribués.

5.4.2 Appariement de graphes

La méthode précédente compare des vecteurs ou des valeurs déduits des graphes, mais perd de ce fait une partie de leur information structurelle. Une autre technique consiste à appairer explicitement les graphes, pour un coût algorithmique beaucoup plus élevé car le problème d'appariement de graphes est NP-complet [Bang-Jensen et Gutin, 2008]. Néanmoins, en passant par des heuristiques, il devient possible d'obtenir une solution approchée avec un coût combinatoire plus faible.

Par exemple, Gold et Rangarajan [1996] proposent un algorithme d'alignement de graphes robuste au bruit basé sur une optimisation non-linéaire. Le principe réside en la résolution du problème d'optimisation suivant, avec $G_1 = (V_1, E_1)$ et $G_2 = (V_2, E_2)$ deux graphes :

$$\min_{\mathbf{M} \in \mathbb{R}^{|V_1| \times |V_2|}} -\frac{1}{2} \sum_{a \in V_1} \sum_{i \in V_2} \sum_{b \in V_1} \sum_{j \in V_2} \mathbf{M}[a, i] \mathbf{M}[b, j] \mathbf{C}[(a, i), (b, j)], \quad (5.6)$$

$$\text{sous les contraintes : } \forall a \in |V_1| \sum_{i \in V_2} \mathbf{M}[a, i] \leq 1, \quad (5.7)$$

$$\forall i \in |V_2| \sum_{a \in V_1} \mathbf{M}[a, i] \leq 1, \quad (5.8)$$

$$\forall a, i \in V_1 \times V_2 \quad \mathbf{M}[a, i] \in \{0, 1\}, \quad (5.9)$$

la matrice \mathbf{C} étant définie par :

$$\mathbf{C}[(a, i), (b, j)] = \begin{cases} 0 & \text{si } (a, b) \notin E_1 \text{ ou } (i, j) \notin E_2 \\ c((a, b), (i, j)) & \text{sinon} \end{cases}, \quad (5.10)$$

avec $c(., .)$ une fonction de comparaison entre deux arêtes. Pour expliquer le fonctionnement de cet algorithme, nous proposons un exemple fourni par les auteurs : soient \mathbf{O} et \mathbf{P} les matrices d'adjacences respectivement associées à G_1 et G_2 . Lorsque $c((a, i), (b, j)) = c(\mathbf{O}[a, b], \mathbf{P}[i, j]) = 1 - 3|\mathbf{O}[a, b] - \mathbf{P}[i, j]|$, le problème de comparaison de graphes devient le problème de recherche du plus grand sous-graphe commun. La fonction objectif devient alors :

$$-\frac{1}{2} \sum_{a \in V_1} \sum_{i \in V_2} \sum_{b \in V_1} \sum_{j \in V_2} \mathbf{M}[a, i] \mathbf{M}[b, j] \mathbf{O}[a, b] \mathbf{P}[i, j]. \quad (5.11)$$

Un graphe G est le plus grand sous-graphe commun entre deux graphes G_1 et G_2 si et seulement si la plus petite valeur de (5.11) est le nombre d'arêtes dans G . Ceci peut être interprété comme l'application de la règle du rectangle (figure 5.5) pour l'isomorphisme de sous-graphe : les quatre termes (soient les quatre flèches formant le rectangle) doivent valoir 1 (c'est à dire que les quatre flèches existent) pour qu'un isomorphisme soit considéré.

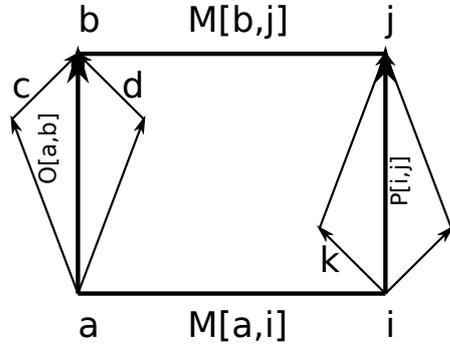


FIG. 5.5: La règle du rectangle pour l'isomorphisme de sous-graphe.

Ce problème d'optimisation est donc NP-complet pour un grand nombre de fonctions $c(.,.)$, Gold et Rangarajan [1996] proposent d'utiliser des heuristiques pour obtenir une solution sous-optimale convenable en utilisant une approche dérivée de *softmax* [Bridle, 1990].

Siddiqi et al. [1999] proposent une méthode proche de celle de Reyner [1977] qui effectue un appariement robuste des graphes pour en déduire une mesure de similarité. Celle-ci est basée sur un algorithme récursif et parcourt les arbres de chocs (section 5.3.2.2) en profondeur en choisissant à chaque étape, c'est à dire à chaque sommet, la meilleure arête selon un critère pour continuer le parcours.

Une autre approche est de calculer les similarités entre graphes en comparant des chemins particuliers. Par exemple, Bai et Latecki [2008] présentent un algorithme pour une comparaison rapide des graphes issus d'un squelette : ils commencent par appairier les sommets terminaux (sommets de degré 1), puis considèrent uniquement, pour chaque appariement, le chemin le plus court entre les deux sommets. L'évolution du rayon de la boule maximale le long de ces deux chemins est ensuite comparée avec un algorithme robuste basé sur un alignement de chaînes de caractères. Ceci a l'avantage de pouvoir prendre en compte les boucles présentes dans le graphe. Cependant, cette technique est spécifique à la comparaison de formes et ne garantit pas que l'ensemble du graphe sera parcouru.

5.4.3 Distance d'édition

L'appariement de deux graphes induit une mesure de la différence entre ceux-ci. Cette distance peut être exprimée comme le coût minimal des opérations nécessaires pour transformer

l'un des graphes en l'autre. Cet autre point de vue permet de concevoir des algorithmes qui cherchent plutôt un coût de transformation qu'un appariement optimal.

Les graphes issus des squelettes présentant un bruit structurel non négligeable, l'utilisation des distances d'édits pour calculer la similarité entre les graphes paraît logique. L'ensemble des opérations élémentaires (insertion, suppression, substitution de sommet ou d'arête) permettant de passer d'un graphe à un autre est appelé un chemin d'édition. Soient G et G' deux graphes, la formule générale d'une distance d'édition [Riesen et al., 2007a] est définie par :

$$d_{\text{edit}}(G, G') = \min_{(e_1, \dots, e_k) \in \Gamma(G, G')} \sum_{i=1}^k c(e_i) \quad (5.12)$$

avec $\Gamma(G, G')$ l'ensemble des chemins d'édition transformant G en G' , $(e_1, \dots, e_k) \in \Gamma(G, G')$ un chemin d'édition composé d'une succession d'opérations élémentaires e_i et $c(\cdot)$ la fonction retournant le coût d'une opération. Le calcul demande des heuristiques afin de réduire la complexité temporelle. Par exemple, Klein et al. [2001] calculent une distance d'édition entre les arbres en utilisant des cycles eulériens dans une vue combinatoire des arbres (un peu à la manière des cartes combinatoires). Ces cycles sont ensuite codés par des chaînes de caractères comparées par une distance entre chaînes. L'ensemble des comparaisons est combiné afin d'obtenir une distance d'édition entre arbres. D'autres approches ont été proposées pour approximer la distance d'édition, par exemple, avec une programmation quadratique [Neuhaus et Bunke, 2007a] ou un graphe d'association bipartite [Riesen et al., 2007b].

5.4.4 Clique maximale dans un graphe d'association

Avant de présenter les méthodes basées sur la recherche de cliques maximales, nous avons besoin d'introduire de nouvelles définitions.

Définition 5.4.1 (Clique).

Une clique est un sous-graphe dont les sommets sont deux-à-deux adjacents.

Définition 5.4.2 (Clique maximale).

Une clique est dite maximale s'il n'existe pas dans le graphe de clique plus grande la contenant. Une clique maximale est dite maximum s'il n'existe pas de clique de taille supérieure dans le graphe.

Définition 5.4.3 (Graphe d'association [Barrow et Burstall, 1976]).

Le graphe d'association entre deux graphes $G_1 = (V_1, E_1)$ et $G_2 = (V_2, E_2)$ est le graphe $G = (V, E)$ tel que $V = V_1 \times V_2$ et pour tous les sommets (u, w) et (v, z) dans V , nous avons :

$$((u, w), (v, z)) \in E \iff u \neq v, w \neq z \text{ et } (u, v) \in E_1 \iff (w, z) \in E_2$$

Pelillo [1998] proposent de comparer deux graphes en recherchant la ou les cliques maximales dans le graphe d'association associé aux deux arbres. Le principe repose sur le théorème suivant liant la recherche d'une clique maximale au problème de recherche d'isomorphisme de graphes :

Théorème 5.4.4 (Massaro et Pelillo [2003]).

Soient deux graphes $G_1 = (V_1, E_1)$ et $G_2 = (V_2, E_2)$ avec $G = (V, E)$ le graphe d'association correspondant. Toutes les cliques maximales (maximum) dans G sont en correspondance un-à-un avec des sous-graphes maximaux (maximum) communs entre G_1 et G_2 .

La figure 5.6 montre un exemple de graphe d'association. Toutes les cliques maximales sont de degré 2 et proposent un appariement de sous-graphes.

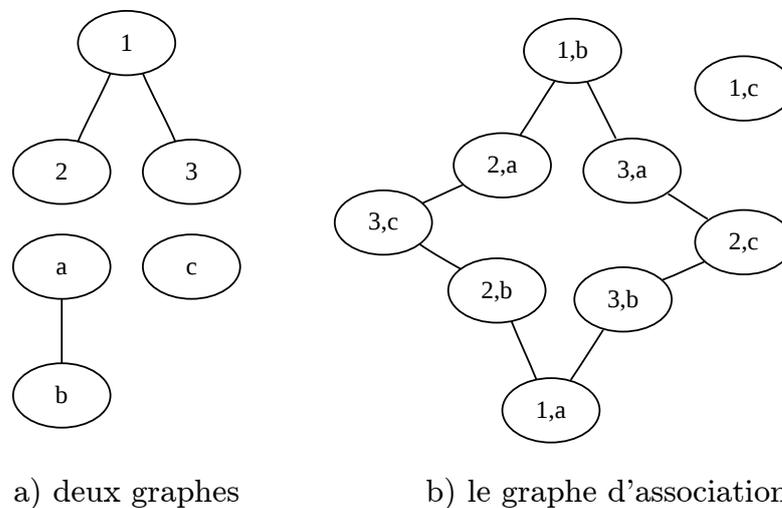


FIG. 5.6: Exemple de graphe d'association

Cependant ce problème est NP-complet et, pour le résoudre, Pelillo et al. [1999]; Pavan et Pelillo [2007] font appel à la théorie des jeux pour construire une heuristique permettant d'obtenir une solution sous-optimale acceptable. D'autres heuristiques ont été proposées, par exemple, Metzen et al. [2007] combinent l'approche de Pelillo et al. [1999] avec des approches basées sur les axes centraux pour recalibrer des images 3D de vaisseaux sanguins.

Le problème d'appariement de graphes peut être étendu aux graphes pondérés, le problème devient alors une recherche de sous-graphe maximal de poids le plus élevé [Locatelli et al., 2004]. Une extension du graphe d'association pour le cas particulier des arbres enracinés a aussi été discutée par Pelillo et al. [1999] pour comparer des arbres de chocs (section 5.3.2.2). Le principal défaut de cette méthode est d'effectuer uniquement des appariements exacts impliquant une sensibilité au bruit structurel.

5.5 Noyaux sur graphes

Les graphes sont des représentations complexes et non-vectorielles, leur comparaison n'est pas aisée et implique une combinatoire élevée (section 5.4). De plus, l'espace des graphes ne possède pas les outils mathématiques traditionnellement utilisés pour les tâches de classifications et de comparaisons. Par exemple, la définition du graphe moyen d'un ensemble de graphes est très délicate.

Une solution est de projeter les graphes dans un espace plus souple possédant des outils mathématiques classiques. Une telle projection est implicitement appliquée au sein des méthodes à noyaux. Celles-ci proposent de construire directement ou indirectement un produit scalaire dans un espace de Hilbert. Soit K un noyau sur graphe, celui-ci projette les graphes dans un espace de Hilbert \mathcal{H} par l'intermédiaire d'une fonction de projection $\mathcal{P}_{\mathcal{H}}$, puis en calcul le produit scalaire :

$$K(x, y) = \langle \mathcal{P}_{\mathcal{H}}(x), \mathcal{P}_{\mathcal{H}}(y) \rangle , \quad (5.13)$$

avec x, y deux graphes. La fonction $\mathcal{P}_{\mathcal{H}}$ n'a pas besoin d'être explicitement construite pour créer un noyau.

Cependant, pour qu'un noyau corresponde à un produit scalaire dans un espace de Hilbert, celui-ci doit vérifier la condition de Mercer, c'est à dire être défini positif [Aronszajn, 1950].

Définition 5.5.1 (Mercer [1909]).

Un noyau K est dit défini positif sur $\mathcal{X} \times \mathcal{X}$, \mathcal{X} étant un espace non vide, si et seulement si il est symétrique ($K(x, y) = K(y, x)$) et vérifie la condition suivante :

$$\forall x_1, \dots, x_n \in \mathcal{X}, \forall c \in \mathbb{R}^n, \quad \sum_i \sum_j c_i K(x_i, x_j) c_j \geq 0. \quad (5.14)$$

Si cette somme est égale à 0 uniquement si $\forall i c_i = 0$, le noyau est alors strictement défini positif dans le domaine \mathcal{X} .

Si le noyau n'est pas strictement défini positif, il est aussi dit semi-défini positif. L'espace de projection créé par un noyau est appelé *espace reproductible du noyau* et celui-ci est unique pour chaque noyau [Aronszajn, 1950]. Notons qu'un noyau strictement défini positif induit une métrique dans cet espace.

Les noyaux ont été utilisés par l'intermédiaire de *l'astuce des noyaux* (« kernel trick ») pour généraliser un grand nombre de classifieurs linéaires et d'outils tels que le séparateur à vaste marge (SVM), l'analyse en composantes principales (ACP), etc. L'ensemble de ces algorithmes forme la famille des *machines à noyaux*.

La construction d'un noyau est souvent effectuée par l'assemblage de noyaux de bases (Tab. 5.1) en utilisant les propriétés suivantes :

Proposition 5.5.2 (Berg et al. [1984]).

Soient K_1, K_2 deux noyaux définis positifs sur $\mathcal{X} \times \mathcal{X}$, où \mathcal{X} est un espace non vide, alors :

1. L'ensemble des noyaux définis positifs est un cône convexe et fermé, c'est à dire, (a) si $\alpha_1, \alpha_2 \geq 0$, alors $\alpha_1 K_1 + \alpha_2 K_2$ est défini positif, et (b) si $K(x, x') := \lim_{n \rightarrow \infty} K_n(x, x')$, K_n étant une suite de noyaux définis positifs, existe pour tout x, x' , alors K est défini positif.
2. Le produit $K_1 K_2$ est défini positif.
3. Supposons que pour $i = 1, 2$, K_i soit défini positif sur $\mathcal{X}_i \times \mathcal{X}_i$, avec \mathcal{X}_i un ensemble non vide. Alors le produit de tenseurs $K_1 \otimes K_2$ (i.e. $K_1 K_2$) et la somme directe $K_1 \oplus K_2$ (i.e. $K_1 + K_2$) sont des noyaux définis positifs sur $(\mathcal{X}_1 \times \mathcal{X}_2) \times (\mathcal{X}_1 \times \mathcal{X}_2)$.

Linéaire	$K(x, y) = x^T y$
Gaussien	$K(x, y) = \exp\left(-\frac{\ x-y\ ^2}{2\sigma^2}\right)$, $\sigma \in \mathbb{R}^*$
Polynomial	$K(x, y) = (x^T y)^d + c$, $c \in \mathbb{R}$, $d \in \mathbb{N}$
Rationnel	$K(x, y) = 1 - \frac{\ x-y\ ^2}{\ x-y\ ^2 + b}$, $b \in \mathbb{R}^*$
Cosinus	$K(x, y) = \frac{x^T y}{\ x\ \ y\ }$

TAB. 5.1: Les principaux noyaux de bases

De nombreux noyaux ont été conçus pour comparer différents types d'objets : histogrammes, chaînes de caractères, arbres, graphes, etc [Hofmann et al., 2008]. Par exemple, Haussler [1999] généralise la construction des noyaux sur des données structurées en introduisant les noyaux de convolutions. Cortes et al. [2004] développent une théorie de construction de noyaux à partir de transducteurs pondérés et précisent sous quelles conditions de tels noyaux sont définis positifs.

Nous nous intéressons plus particulièrement aux noyaux sur graphes. Ramon et Gärtner [2003] ont démontré que la construction d'un noyau strictement défini positif sur le domaine des graphes revient à résoudre un problème NP-complet lié au problème de recherche de sous-graphes communs entre deux graphes. En revanche, Gärtner [2002] construisent des noyaux à base de marches aléatoires qui comparent l'ensemble des chemins de deux graphes. Depuis les travaux pionniers de Kashima et al. [2003] et de Gärtner [2002], de nombreux noyaux sur graphes ont fait leur apparition, quasiment tous basés sur les noyaux de convolutions [Haussler, 1999]. Un aperçu détaillé de ces noyaux est produit dans la section suivante.

5.5.1 Noyau sur les chemins

Parmi les noyaux qui vont suivre beaucoup reposent sur un noyau sur chemins. Ce noyau compare les motifs produits par les labels des sommets et des arêtes le long des chemins.

Kashima et al. [2003] proposent un noyau basé sur un produit de tenseurs entre noyaux sur les labels :

$$K_{\text{chemin}}(h, h') = \begin{cases} 0, & \text{si } |h| \neq |h'|, \\ K_v(\varphi(v_1), \varphi(v'_1)) \prod_{i=2}^{|h|} K_e(\psi(e_{v_{i-1}v_i}), \psi(e_{v'_{i-1}v'_i})) K_v(\varphi(v_i), \varphi(v'_i)), & \text{sinon,} \end{cases} \quad (5.15)$$

avec φ (resp. ψ) la fonction qui retourne le vecteur de données associé aux sommets (resp. arêtes). Les noyaux K_v et K_e désignent les noyaux sur les données associées respectivement aux sommets et aux arêtes. Le noyau K_{chemin} est défini positif si et seulement si K_v et K_e sont définis positifs. Les noyaux utilisés pour construire K_v et K_e sont souvent basés sur les noyaux gaussien (table 5.1).

5.5.2 Marches aléatoires

Gärtner [2002] proposent de comparer les graphes en parcourant aléatoirement les deux graphes afin de trouver des similarités. Les parcours aléatoires sont modélisés en utilisant des marches aléatoires. Kashima et al. [2003] ajoutent des lois marginales de probabilités sur les sommets et arêtes des graphes afin d'améliorer les comparaisons. Cependant, le noyau sur les marches aléatoires souffre du phénomène d'aller-retour direct entre deux sommets qui peut dégrader les comparaisons. Pour éviter ce phénomène, Mahé et al. [2004] changent les lois marginales de probabilités et prouvent que le changement revient en réalité à modifier le graphe. Une autre difficulté de ce noyau est son temps de calcul. Cependant, Vishwanathan et al. [2008], dans leur unification des différentes approches, proposent un cadre permettant de calculer rapidement les noyaux sous certaines conditions.

Le noyau générique est décrit par :

$$K_{\text{marches}}(G, G') = \sum_{w \in G} \sum_{w' \in G'} \nu(p, p') K_{\text{chemin}}(p, p') p(w|G) p(w'|G') \quad (5.16)$$

avec $p(\cdot|G)$ la loi marginale sur la distributions des chemins dans un graphe G et $\nu(\cdot, \cdot)$ une fonction définie positive assurant la convergence. Ce noyau est un noyau de convolution et donc défini positif sur le domaine des graphes si et seulement si K_{chemin} est défini positif [Haussler, 1999].

Les noyaux basés sur les marches aléatoires comparent l'ensemble des motifs construits par les chemins aléatoires dans les graphes. Deux graphes différents peuvent produire les mêmes ensembles de motifs (figure 5.7). Les marches aléatoires peuvent donc être incapables de distinguer certains graphes, ce phénomène rare peut toutefois être problématique avec de grandes bases de graphes.

De plus, le gros défaut de cette approche reste son coût dans le cas général. Lorsque le graphe est simple, des implémentations performantes [Vishwanathan et al., 2008] existent basées sur

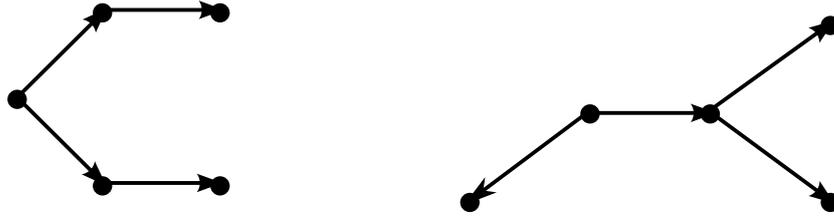


FIG. 5.7: Deux graphes possédant le même ensemble de chemins.

des marches aléatoires dans le graphe produit, cela suppose aussi que les probabilités puissent s'exprimer à l'aide d'une matrice de transition pour les arêtes et des probabilités de départ et d'arrivée pour les sommets. Un autre défaut est la sensibilité au bruit structurel, en effet, l'ajout ou la suppression d'une arête ou d'un sommet peut fortement changer les motifs formés par les marches aléatoires.

5.5.3 Motifs engendrés par des arbres

Les noyaux basés sur les marches aléatoires souffrant de la vue linéaire qu'elles représentent du graphe, Ramon et Gärtner [2003] introduisent des motifs d'arbres pour comparer les graphes sans le coté linéaire des chemins. Récemment, Mahé et Vert [2009] ont étendu cette méthode pour définir plusieurs noyaux de convolution basés sur des motifs formés par des arbres dans les graphes.

Définition 5.5.3 (Motif d'arbre [Mahé et Vert, 2009]).

Soient un graphe $G = (V_G, E_G)$ et un arbre $T = (V_T, E_T)$, avec $V_T = (n_1, \dots, n_{|T|})$. Un $|T|$ -tuple de sommets $(v_1, \dots, v_{|T|}) \in V_G^{|T|}$ est un motif d'arbre de G selon T si et seulement si

$$\begin{cases} \forall i \in [1, |T|], & l(v_i) = l(n_i), \\ \forall (n_i, n_j) \in E_T, & (v_i, v_j) \in E_G \text{ et } l((v_i, v_j)) = l((n_i, n_j)), \\ \forall (n_i, n_j), (n_i, n_k) \in E_T, & j \neq k \iff v_j \neq v_k \end{cases}$$

avec $l(\cdot)$ la fonction qui retourne les labels associés aux sommets et aux arêtes. Un tel motif $(v_1, \dots, v_{|T|})$ est noté $\text{motif}(T)$.

Soient G_1 et G_2 deux graphes, le noyau général sur graphes basé sur les motifs d'arbre est alors défini par :

$$K(G_1, G_2) = \sum_{T \in \mathcal{T}} \omega(T) \psi_T(G_1) \psi_T(G_2), \quad (5.17)$$

avec \mathcal{T} l'ensemble des arbres, $\omega : \mathcal{T} \rightarrow \mathbb{R}^+$ une fonction non-négative de pondération des arbres et ψ_T la fonction comptant le nombre d'apparition du motif associé à l'arbre T ($\text{motif}(T)$).

Ce noyau compare les graphes non-linéairement et utilise donc des informations qui sont inaccessibles pour des chemins. Toutefois, le noyau montre des résultats proches de ceux obtenus

avec les marches aléatoires [Mahé et Vert, 2009]. De plus, il souffre du même problème que le noyau précédent, c'est à dire qu'un motif peut être créé par une oscillation entre deux sommets. Le calcul des motifs repose sur la programmation dynamique et peut devenir coûteux lorsque les motifs sont long.

5.5.4 Sacs de chemins

Borgwardt et Kriegel [2005] définissent deux noyaux qui contrairement aux noyaux précédents n'utilisent pas un ensemble infini de motifs. Au contraire, le premier noyau considère tous les chemins élémentaires ou simples et le second construit l'ensemble des chemins en prenant le chemin le plus court entre chaque paire de sommets différents. Prenant la même formulation que les noyaux sur les marches aléatoires, les auteurs utilisent une convolution pour comparer les chemins de deux graphes G_1 et G_2 :

$$K_{\text{Borgwardt}}(G_1, G_2) = \sum_{w \in \mathcal{M}(G_1)} \sum_{w' \in \mathcal{M}(G_2)} K_{\text{chemin}}(w, w') \quad (5.18)$$

avec $\mathcal{M}(G_1)$ et $\mathcal{M}(G_2)$ l'ensemble des chemins selon la méthode choisie des graphes G_1 et G_2 . Borgwardt et Kriegel [2005] font remarquer, par le théorème suivant, que la complexité temporelle du premier noyau le rend difficilement utilisable dans le cas général :

Théorème 5.5.4 (Borgwardt et Kriegel [2005]).

La construction d'un sac de chemins contenant tous les chemins simples (ou élémentaires) d'un graphe G est un problème NP-complet.

Démonstration. Soit $\mathcal{M}(G)$ le sac de chemins contenant tous les chemins simples d'un graphe G . Si G possède un chemin eulerien (ou hamiltonien avec les chemins élémentaires) alors ce chemin est dans $\mathcal{M}(G)$, donc la construction de $\mathcal{M}(G)$ permet de résoudre le problème du chemin eulerien (ou hamiltonien). Or ces deux problèmes sont NP-complets [Bang-Jensen et Gutin, 2008]. Donc la construction du sac $\mathcal{M}(G)$ est un problème NP-complet. \square

Reprenant l'idée précédente, Suard et al. [2006, 2007] développent l'idée de comparer uniquement une partie des motifs présents dans les graphes. Ainsi, ils construisent des noyaux prenant uniquement deux sacs de chemins extraits de chacun des graphes. L'idée est de sélectionner directement les chemins les plus discriminants en laissant la liberté de choix au niveau des heuristiques mises en œuvre. La comparaison entre deux graphes est alors effectuée par un noyau sur les sacs de chemins. Soient G_1 et G_2 deux graphes, P_1 le sac de chemins associé à G_1 et P_2 celui associé à G_2 , Suard et al. [2006] introduisent plusieurs noyaux pour comparer les sacs de chemins :

Noyau moyenne

$$K_{\text{moyenne}}(P_1, P_2) = \frac{1}{|P_1|} \frac{1}{|P_2|} \sum_{h_1 \in P_1} \sum_{h_2 \in P_2} K_{\text{chemin}}(h_1, h_2), \quad (5.19)$$

Noyau max

$$\begin{aligned} \hat{K}_{\text{max}}(P, P') &= \frac{1}{|P|} \sum_{h \in P} \max_{h' \in P'} K_{\text{chemin}}(h, h'), \\ K_{\text{max}}(P_1, P_2) &= \frac{1}{2} \left(\hat{K}_{\text{max}}(P_1, P_2) + \hat{K}_{\text{max}}(P_2, P_1) \right), \end{aligned} \quad (5.20)$$

Noyau approximation

$$K_{\text{approx}} = \frac{1}{|P_1|} \frac{1}{|P_2|} \sum_{h_1 \in P_1} \sum_{h_2 \in P_2} \exp \left(-\frac{d_{\text{chemin}}^2(h_1, h_2)}{2\sigma_{\text{approx}}^2} \right), \quad (5.21)$$

$$\text{avec } d_{\text{chemin}}^2(h_1, h_2) = K_{\text{chemin}}(h_1, h_1) + K_{\text{chemin}}(h_2, h_2) - 2K_{\text{chemin}}(h_1, h_2).$$

Sauf K_{max} , les autres noyaux sont tous définis positifs sur les sacs de chemins. K_{max} est un noyau indéfini, cependant il semble donner de bons résultats [Suard et al., 2006]. Le noyau K_{approx} reprend l'idée de base du noyau K_{max} , c'est à dire de faire intervenir en priorité les chemins les plus proches.

Suard et al. [2006] propose un dernier noyau, basé sur une mesure de détection de changements introduite par Desobry et al. [2005] qui compare des estimations de supports de densité de probabilités. Les supports de densité de probabilités sont estimés en utilisant le SVM à une classe [Hofmann et al., 2008] (ou « one-class SVM ») calculant pour chaque distribution une moyenne w et une marge ρ . Soient $x_1, \dots, x_n \in \mathcal{X}$ les éléments d'un l'ensemble inclus dans \mathcal{X} , le support de densité de probabilités est estimé par le classifieur à une classe suivant :

$$\min_{w, \rho, \xi} \frac{1}{2} \|w\|_2^2 + \sum_{i=1}^n \xi[i] - n\nu\rho \quad (5.22)$$

$$\text{sous les contraintes : } \langle w, x_i \rangle \geq \rho - \xi[i] \text{ et } \xi[i] \geq 0,$$

avec $\nu \in [0, 1]$ le pourcentage d'éléments rejetés lors du calcul du support.

La mesure de détection de changements est alors calculée à partir des estimations des supports :

$$K_{\text{densité}}(P_1, P_2) = \rho_{P_1} \rho_{P_2} \sum_{i: h_i \in P_1} \sum_{j: h_j \in P_2} w_{P_1}[i] w_{P_2}[j] K_{\text{chemin}}(h_i, h_j), \quad (5.23)$$

avec ρ_{P_1} et w_{P_1} la marge et la moyenne associée au sac de chemins P_1 et ρ_{P_2} et w_{P_2} la marge et la moyenne associée au sac de chemins P_2 . Ce noyau peut être interprété comme le produit scalaire entre les deux fonctions de décisions estimées par le classifieur à une classe. Le noyau est défini positif, mais peut projeter deux sacs de chemins différents sur le même point. En

effet, ils peuvent présenter des marges et des moyennes identiques. D'autant plus que le SVM à une classe permet d'ignorer les points aberrants. Toutefois, le problème est mineur car de tels sacs ont beaucoup d'éléments en commun et donc proviennent de graphes similaires.

5.5.5 Distance d'édition

Au premier abord les méthodes à noyaux semblent difficiles à lier avec celles de comparaison de formes présentées dans la section 5.4. En effet, soit elles conduisent à une distance qui ne vérifie pas toutes les propriétés d'une métrique [Cortes et al., 2004; Bunke et Riesen, 2007] (ce qui correspond à un noyau indéfini), soit elles s'intéressent aux structures communes de deux graphes, mais ne cherchent pas à définir une distance ou une mesure de similarité.

Bunke et Riesen [2007] ont essayé de combler le fossé entre ces deux théories. Ils ont proposé plusieurs noyaux basés directement sur une distance d'édition, ou basés en s'inspirant du fonctionnement de celle-ci. Par exemple, ils prennent une distance d'édition et l'utilisent comme une véritable métrique. Le noyau ainsi créé est évidemment indéfini, mais le comportement de tels noyaux avec un SVM est connu [Haasdonk, 2005]. Soient d_{edit} une distance d'édition, G et G' deux graphes, les noyaux « triviaux » introduits par Neuhaus et Bunke [2007b] sont :

$$K_1(G, G') = -d_{\text{edit}}^2(G, G') \quad (5.24)$$

$$K_2(G, G') = -d_{\text{edit}}(G, G') \quad (5.25)$$

$$K_3(G, G') = \tanh(-d_{\text{edit}}(G, G')) \quad (5.26)$$

$$K_4(G, G') = \exp(-d_{\text{edit}}(G, G')) \quad (5.27)$$

$$K_5(G, G') = \exp(-d_{\text{edit}}^2(G, G')) \quad (5.28)$$

Ces noyaux sont inspirés des résultats classiques sur les noyaux construits à partir d'une métrique.

Une autre approche proposée consiste à modifier le noyau sur marches aléatoires afin de comparer uniquement les chemins dont l'alignement est correct par rapport à l'appariement effectué par une distance d'édition. Ce noyau est aussi indéfini, à cause de l'appariement. Autre méthode, Neuhaus et Bunke [2007b] proposent d'utiliser un noyau de diffusion [Kondor et Lafferty, 2002] sur une matrice de similarité, construite à partir des distances d'éditions entre les éléments d'entraînement et l'élément à classer, afin d'obtenir une matrice définie positive. Cependant, le noyau construit à partir de ce principe, calcule une matrice de diffusion pour chaque graphe à classer. Il implique donc une projection dans un espace différent à chaque utilisation. La définie positivité d'un tel noyau est donc incertaine. En revanche, un tel processus peut être utilisé pour obtenir une matrice définie positive basée sur l'ensemble des éléments.

Neuhaus et Bunke [2007b] définissent aussi deux noyaux définis positifs. Le premier est basé sur l'ensemble des chemins de décomposition des graphes. Il demande de résoudre un problème NP-complet afin de comparer les graphes. Le deuxième consiste à comparer les graphes au niveau des sommets et de leurs voisinages immédiats (les arêtes directement incidentes). Cette comparaison est effectuée en calculant toutes les combinaisons possibles d'arêtes, le noyau sur les sommets étant formé par un noyau de convolution entre les différentes combinaisons. Le noyau final est un noyau de convolution entre les sommets et donc défini positif. Cependant, celui-ci applique uniquement des comparaisons locales entre les graphes, il est donc efficace uniquement si les graphes diffèrent localement au niveau du voisinage des sommets.

5.5.6 Estimation des paramètres des noyaux

Un problème important posé par les machines à noyaux est la détermination de leurs paramètres optimaux. En effet, des noyaux complexes peuvent comporter de nombreux paramètres. De plus, la valeur de chaque paramètre est représentative des données à classer, il n'y a donc généralement pas de valeur par défaut. L'approche la plus simple est de créer une grille de recherche dans le domaine des paramètres, en utilisant des échelles logarithmiques, puis de trouver la combinaison optimale parmi les points de la grille. Cependant, autant cette technique est utilisable avec un ou deux paramètres, autant elle devient inutilisable pour un plus grand nombre. Des heuristiques peuvent évidemment être mises en place lorsque des informations sont connues sur les données, mais la solution risque d'être sous-optimale.

Dans le cas d'utilisation d'un SVM, [Keerthi et al., 2006; Chapelle et al., 2002] proposent un cadre global pour obtenir les paramètres du SVM et des noyaux. Ils utilisent les propriétés du SVM pour formuler un problème d'optimisation faisant intervenir le critère de classification (F1-score, AUC, précision, rappel, etc) et la méthode de gestion de la base d'entraînement (validation croisée, tous sauf un, etc). La solution est obtenue avec une descente de gradient, ce qui suppose que les différents noyaux soient différentiables. Notons que cette formulation permet d'obtenir une solution optimale quelque soit le point de départ.

5.6 Conclusion

De nombreuses représentations sont disponibles pour les formes, mais peu proposent un lien direct entre morphologie et codage. Parmi ces méthodes nous comptons les chaînes codant les contours et les graphes. Les moments géométriques possèdent bien une interprétation géométrique, mais celle-ci nous semble trop indirecte pour notre application. Nous avons donc choisi les graphes, car cette structure code explicitement le squelette d'une forme et établit un lien fort entre la représentation structurelle et la morphologie de la forme.

Ce lien est d'autant plus intéressant, qu'il nous permet de comparer les formes à la fois localement et globalement, ce qui semble en lien avec les expériences montrant que le système visuel humain découpe les formes en sous-parties. Le problème de comparaison de formes devenu un problème de comparaison de graphes, généralement difficile, doit utiliser cette notion de sous-parties et équilibrer les informations locales avec des informations globales sur la forme. Les graphes de chocs qui fournissent une décomposition « naturelle » de la forme en sous-parties sont, dans ce cadre, particulièrement intéressants.

Les méthodes de comparaisons de graphes présentées sont générales et donc comparent n'importe quel type de graphe, ou bien sont dédiées aux formes. Parmi ces dernières méthodes, les plus intéressantes sont, sans doute, celles qui incorporent une notion d'édition. Celle-ci leur permet en effet d'approximer le coût de déformation d'une forme en une autre et donc de refléter une comparaison visuelle des formes.

L'insertion de ces méthodes dans des outils de classification n'est pas toujours possible. Les plus simples restent souvent disponibles, mais le support mathématique est faible et limité. Les noyaux permettent justement de profiter de toute la richesse mathématique des espaces de Hilbert. Toutefois, la construction d'un noyau sur graphes est complexe, et la conservation de la définie positivité impose des règles assez strictes. L'existant peut pour l'instant se découper en deux parties : les noyaux basés sur des comparaisons de densité d'apparition de motifs (marches aléatoires ou motifs d'arbre) et ceux sur des sacs de chemins caractéristiques.

Ces noyaux sont conceptuellement assez éloignés de la distance d'édition, et donc ne sont pas vraiment adaptés à la comparaison de forme. Neuhaus et Bunke [2007b] lient les deux approches, mais sans réellement apporter la notion d'édition dans les noyaux définis positifs. Toutefois, ils montrent que les deux mondes peuvent être combinés. Dans le chapitre suivant, nous proposons d'insérer la notion d'édition dans les noyaux introduits par Suard et al. [2006]. Ainsi les noyaux sont définis positifs et ont un lien fort entre la forme et les éléments comparés.

6 Sacs de chemins avec édition

Qu'importe l'issue du chemin quand seul compte le chemin parcouru.

David le Breton

L'approche de Neuhaus et Bunke [2007b] (section 5.5.5) est d'utiliser directement ou indirectement une distance d'édition sur graphes. Cependant, les distances d'éditations sont des outils complexes qui peuvent être sur-dimensionnés par rapport à la tâche à accomplir. En revanche, Suard et al. [2006] proposent de découper les graphes en un ensemble de vues linéaires et simples et, de comparer les graphes uniquement à partir de ces vues. Cependant à la différence de l'approche de Neuhaus, ces comparaisons ne sont pas robustes au bruit.

Dans ce chapitre, nous présentons une technique d'insertion d'un processus d'édition pour comparer les chemins inspiré des distances d'éditations. Celle-ci permet d'introduire une robustesse au bruit dans les noyaux sur graphes. Ces noyaux sont générés à partir de trois briques élémentaires : 1) un sac de chemins, 2) un noyau sur sacs de chemins et 3) un noyau sur chemins.

Notre première contribution porte sur chacune des étapes : 1) nous présentons deux méthodes de construction de sacs générant peu de chemins avec un maximum d'information, 2) nous proposons de nouveaux noyaux sur sacs de chemins et 3) nous introduisons un noyau sur chemins robuste au bruit. La robustesse est assurée par une comparaison hiérarchique de chemins liée à un processus d'édition.

Les trous peuvent représenter des caractéristiques importantes d'une forme. Or ceux-ci ne sont pas particulièrement exploités dans les noyaux précédents. Afin de prendre en compte à la fois, les trous et la structure générale des formes, nous combinons un noyau sur sacs de trous avec un noyau sur graphes. Ces approches sont validées à travers un ensemble d'expérimentations. Finalement, nous présentons des perspectives liant notre travail avec des travaux récents sur les noyaux sur graphes.

6.1 Extraction du graphe

L'extraction du graphe à partir d'une forme se déroule en deux étapes : le graphe est tout d'abord construit à partir du squelette, puis enrichi par l'intermédiaire d'attributs sur les sommets et les arêtes avec des informations sur la forme.

6.1.1 Structure du graphe

La représentation par graphe de chocs, introduite par Siddiqi et Kimia [1996] (section 5.3.2.2), propose une grammaire permettant de regrouper les chocs, afin d'obtenir un graphe unique et irréductible représentant une forme. Ces regroupements de plusieurs chocs en un seul peuvent conduire à une perte d'information en cas de changements de l'accélération (Fig. 6.1(a)) dans l'évolution du rayon des cercles maximaux. De plus, l'orientation des arêtes selon l'évolution de la pente des rayons des cercles maximaux demande une estimation précise, une erreur pouvant modifier la structure du graphe.

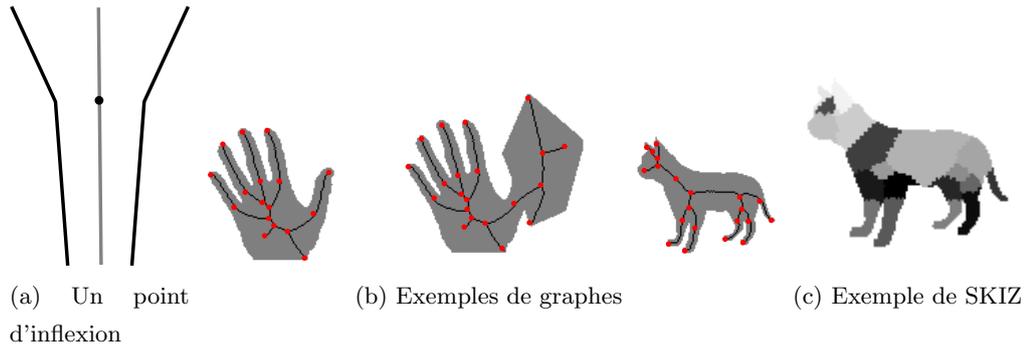


FIG. 6.1: Les graphes sont constitués en utilisant les squelettes et les points d'inflexion. Les arêtes sont liées à leur zone d'influence.

Les processus de regroupement et d'orientation des graphes de chocs peuvent donc être perturbés par des erreurs d'estimation. Donc, ces erreurs influencent la structure finale des graphes. Afin d'avoir un codage des formes riche, sans les défauts des graphes de chocs, nous proposons de prendre le graphe direct (section 5.3.2.1) et d'y ajouter des sommets aux endroits de changements significatifs de pentes. A l'instar des chocs, ces zones de changements correspondent à une frontière entre deux parties distinctes dans la forme.

La difficulté de notre codage réside alors dans la détection des changements de pentes. Pour cela, nous effectuons une régression multi-polynomiales avec des splines de premier ordre [DiMatteo et al., 2001]. Cette méthode a l'avantage d'être robuste au bruit discret et de sélectionner automatiquement les changements les plus significatifs. Le nombre de changements significatifs est estimé par un critère stochastique : le critère d'information bayésien (BIC) ou d'Akaike (AIC). Une fois la régression effectuée, les points de changements de pentes sont localisés au niveau des nœuds de liaison entre les splines (Fig. 6.2).

6.1.2 Attributs

Le graphe associé au squelette d'une forme n'apporte qu'une information structurelle : un sommet représente une jonction et/ou un changement de pente ou une terminaison, une arête

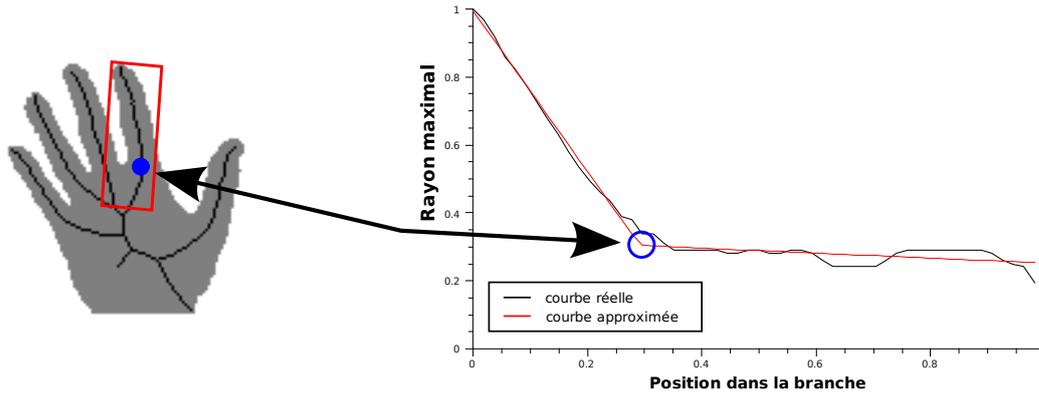


FIG. 6.2: Exemple de point d'inflexion dans une main.

représente une branche. Il convient donc d'ajouter des informations géométriques aux sommets et arêtes du graphe afin d'obtenir une représentation plus complète. D'un point de vue géométrique, un sommet représente soit la frontière entre deux parties de la forme (jonction et/ou changement de pente), soit la terminaison d'une partie (si il est de degré 1). L'arête représente une partie de la forme, c'est à dire un ensemble de points dans le cas discret. Cet ensemble de points est défini comme la *zone d'influence* de la branche, donc, par extension, de l'arête. La zone d'influence peut être obtenue par des algorithmes comme le squelette par zones d'influences (ou SKIZ) [Meyer, 1994] (Fig. 6.1(c)).

La géométrie de la forme est codée par des attributs associés aux sommets et arêtes du graphe. Les attributs sont formés par un ensemble de descripteurs liés soit à une branche pour le cas des arêtes, soit à un point pour le cas des sommets. Nous rappelons que les descripteurs sont classés en deux catégories (section 5.3.2) : les *descripteurs globaux* fournissant une information prenant en compte la forme dans sa totalité (par exemple l'aire d'une partie divisée par l'aire totale) et, les *descripteurs locaux* décrivant une information relative à une zone d'influence indépendamment du reste de la forme.

Goh [2008] a montré l'importance de trouver un équilibre entre descripteurs locaux et globaux. Cependant, cet équilibre dépend fortement des formes considérées. En effet, pour différencier des formes où la silhouette est globalement la même (par exemple un ensemble de mains), les descripteurs locaux seront plus discriminant que les descripteurs globaux. En revanche, pour différencier des formes très dissemblables (par exemple des avions et des lapins), les attributs globaux seront plus pertinents.

6.1.2.1 Descripteurs globaux

Les descripteurs globaux décrivent une partie d'une forme relativement à la globalité de celle-ci. Ils sont essentiels pour comparer globalement les formes et permettent une première

	Sommet	Arête
Plongement géométrique	terminaison, jonction et/ou inflexion du squelette	branche, partie de forme
Descripteurs globaux	distance au centre de gravité	périmètre normalisé
Descripteurs locaux		évolution du rayon du cercle maximal et évolution de l'angle d'orientation

TAB. 6.1: Résumé des descripteurs de la forme associé au graphe.

discrimination entre celles-ci. Comme ils sont globaux, ils sont sensibles aux occlusions et autres déformations, car elles font parties intégrantes de la forme. Toutefois, lorsque les formes sont dissemblables, cette discrimination peut être suffisante.

Les images étant discrètes, nous avons besoin d'un descripteur robuste au bruit discret. Torsello et Hancock [2004] proposent un tel descripteur pour les arêtes. Il repose sur une approximation du périmètre de la bordure ayant conduit à la formation de la branche. Ce descripteur permet de caractériser l'importance d'une branche dans le squelette, donc de détecter les ligatures qui sont générées par la contrainte d'homotopie du processus de squelettisation. Afin d'être invariant au changement d'échelle, le descripteur est normalisé en le divisant par le périmètre total de la forme.

Pour les sommets, nous avons choisi d'utiliser la distance au centre de gravité divisée par la racine carré de l'aire totale de la forme. Ce descripteur permet de tenir compte du degré d'anisotropie de la forme et reste peu sensible au bruit et aux petites déformations.

6.1.2.2 Descripteurs locaux

Les descripteurs locaux permettent d'avoir une description précise et indépendante du reste de la forme. Ils permettent de discriminer des formes qui sont globalement semblables. Comme ils caractérisent une partie de la forme, ils sont insensibles aux occlusions. En revanche il sont sensibles aux déformations. Pour une arête, un descripteur local correspond à une caractérisation de la branche et/ou de la zone d'influence associée à la branche. Pour un sommet, un descripteur local caractérise le point associé au sommet.

Nous proposons d'utiliser deux descripteurs locaux pour les arêtes initialement introduits par Goh [2008] : l'évolution du rayon de la boule maximale le long de la branche et l'évolution de l'orientation le long de cette même branche. L'orientation est représentée par l'angle formé par la tangente et l'axe des abscisses. Les deux évolutions sont codées par des polynômes dont le degré est fixé, au préalable, selon la précision recherchée. Pour chaque branche, nous considérons

les fonctions d'évolution $(x(t), y(t))$, $R(t)$ et $\alpha(t)$, $t \in [0, 1]$ qui représentent respectivement, la position du point au moment t , la valeur du rayon de la boule maximale à la position $(x(t), y(t))$ et l'orientation (l'angle formé par le vecteur vitesse et l'axe des abscisses) de la branche en ce même point. Afin d'être robuste aux changements d'échelle, le rayon des boules maximales est divisé par la racine carré de l'aire de la zone d'influence associée à la branche. La robustesse au bruit est assurée par un lissage gaussien des données, appliqué avant toute opération. Finalement, les descripteurs considérés sont les coefficients $(a_k)_{k \in \mathbb{N}}$ et $(b_k)_{k \in \mathbb{N}}$ des polynômes de régressions de respectivement $R(t)$ et $\alpha(t)$ au sens des moindres carrés. Notons que les degrés sont fixés globalement afin d'avoir des suites comparables, mais le degré du polynôme codant $R(t)$ n'est pas nécessairement le même que celui codant $\alpha(t)$.

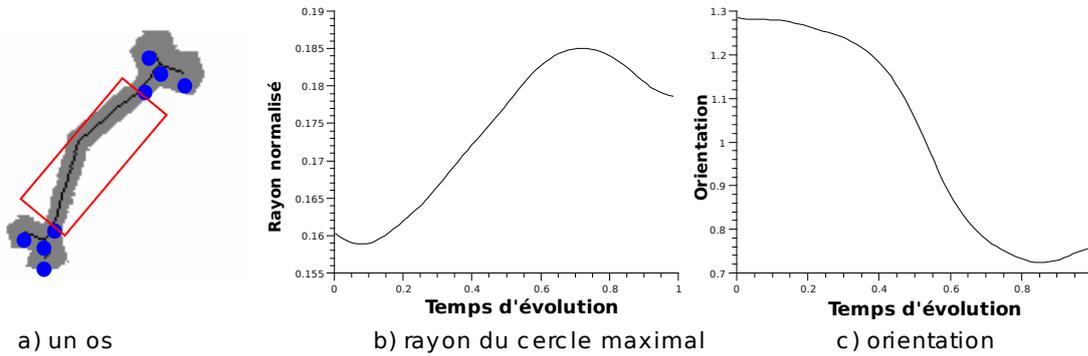


FIG. 6.3: Exemple d'évolutions de rayons et d'orientation.

La figure 6.3 montre les évolutions lissées du rayon et de l'orientation de la partie centrale de l'os (Fig.6.3a). Le rayon montre une évolution (Fig. 6.3b) qui n'est pas nulle, contrairement aux attentes vis à vis de la forme. Cette erreur d'estimation provient du bruit de quantification qui apporte une erreur importante dans l'estimation du rayon et de la position réelle du squelette. Le lissage permet cependant de produire une courbe assez propre. L'évolution de l'orientation (Fig. 6.3c) souffre des mêmes défauts, cependant la courbe montre clairement les trois zones que nous observons directement sur la forme, c'est à dire un plat suivi d'une pente descendante puis d'un nouveau plat.

6.1.2.3 Distance entre attributs

Tous les attributs proposés peuvent être représentés par des vecteurs et comparés par une distance euclidienne entre deux vecteurs. Afin d'être invariant par changement d'échelle, les attributs sont préalablement normalisés. Notons que le squelette d'une forme est intrinsèquement invariant par translation et rotation. Ces distances sont ensuite utilisées pour concevoir des noyaux gaussien, $K(a, b) = \exp\left(-\frac{d^2(a, b)}{2\sigma^2}\right)$, où d est une distance entre deux attributs et σ un paramètre dépendant du type d'attribut. Ici, la distance est la distance euclidienne asso-

ciée à l'espace vectoriel considéré. L'écart type est évidemment dépendant du descripteur. Par exemple, il n'est pas le même pour le périmètre normalisé et pour les coefficients des polynômes décrivant l'évolution du rayon des boules maximales.

6.2 Création d'un sac de chemins

Un sac de chemins contient une succession de vues linéaires d'un graphe. Les chemins simples ou élémentaires présentent l'avantage d'être en nombre finis. Nous les utilisons donc pour construire les sacs. Les problèmes de convergence rencontrés par les noyaux sur les marches aléatoires sont ainsi évités. Notons que les chemins simples peuvent parcourir n'importe quelle arête d'un graphe, à l'inverse des chemins élémentaires. Une première méthode de construction de sacs de chemins consiste à prendre tous les chemins simples (ou élémentaires) du graphe. Toutefois, le théorème 5.5.4 montre que la recherche de tous ces chemins peut être coûteuse voire impossible.

Une première approche ne considère que les chemins simples ou élémentaires jusqu'à une certaine taille. Cette technique souffre d'un défaut : l'information est fortement redondante ce qui peut affecter les performances du noyau en temps de calcul et en précision. En effet, dans le cas des formes, la redondance conduit à une sur-représentation de certaines parties au détriment d'autres. Toutefois, elle peut être discriminante dans certains cas, lorsque les formes partagent de nombreux points communs.

Considérons un sac contenant tous les chemins jusqu'à une taille de s arêtes. La construction du sac demande de connaître toutes les possibilités à partir de chaque sommet, ce qui au pire des cas revient à calculer toutes les combinaisons d'arêtes possibles de taille s . Prenons un graphe $G' = (V', E')$, nous supposons qu'il est un sous-graphe d'un graphe d -régulier $G = (V, E)$ (chacun de ces sommets est de degré d). Nous supposons aussi que G est le plus petit graphe d -régulier contenant G' . Le calcul des chemins de tous type est effectué en utilisant une méthode comme l'algorithme 6.1, dont la complexité pour les graphes d -régulier est donnée par la proposition suivante :

Proposition 6.2.1. *Si G est d -régulier le nombre d'appels à `calculChemins` (algorithme 6.2) impliqué par un appel de `tousLesChemins` (algorithme 6.1) avec `maxLength = s` est égal à :*

$$|V| \sum_{i=0}^s d^i$$

De plus le nombre d'appels terminal (avec un chemin h tel que $|h| = s$) sur chaque sommet est égal à d^s .

Démonstration. Montrons que la propriété est vrai pour $s = 1$. `tousLesChemins` appelle `|V|` fois `CalculChemin`. De plus, pour chaque sommet, nous appelons `CalculChemin` sur tous les

voisins, soit d fois. Le nombre total d'appels est donc égal à $|V|(1+d)$ ce qui coïncide avec la formule générique. De plus, chaque sommet sera un sommet terminal pour *CalculChemin*, lors d'un appel déclenché par un de ses voisins (soit d fois).

Supposons la propriété vraie à l'ordre $s-1$.

Le nombre d'appels à l'ordre s sera égal au nombre d'appels à l'ordre $s-1$ plus les nouveaux appels terminaux. Le nombre d'appel à l'ordre $s-1$ est égal à :

$$|V| \sum_{i=1}^{s-1} d^i$$

Or, puisque chaque sommet rappelle *Calculchemins* sur tous ses voisins, le nombre d'appels terminaux sur un sommet à l'ordre s est la somme des appels terminaux, à l'ordre $s-1$ de chacun de ses voisins. À l'ordre $s-1$, chaque sommet fait l'objet de d^{s-1} appels terminaux. Donc à l'ordre s , un sommet fera l'objet de $\sum_{i=1}^d d^{s-1} = d^s$ appels terminaux. Ceci étant vrai pour tout sommet, le nombre d'appels supplémentaires, impliqués par le passage de l'ordre $s-1$ à l'ordre s , est égal à $|V|d^s$. Le nombre total d'appels à l'ordre s est donc égal à :

$$|V| \sum_{i=1}^s d^i$$

et le nombre d'appels terminaux sur un sommet est égal à d^s . □

La complexité globale de l'algorithme 6.1 est donc :

$$|V| \sum_{i=1}^s d^i = |V| \frac{d^{s+1} - 1}{d - 1} = \mathcal{O}(|V|d^{s+1})$$

Par extension, nous pouvons conclure que la complexité temporelle de calcul des chemins de G' est en $\mathcal{O}(|V|d^{s+1})$. Ce résultat est en accord avec le théorème 5.5.4. Il est toutefois à tempérer, car, généralement, dans le cas des formes, les graphes ne sont pas grands et la taille des chemins recherchés est souvent petite. Néanmoins, pour la comparaison de grands graphes, l'étape de construction des sacs de chemins est un point crucial si nous souhaitons obtenir des complexités temporelles réduites. Par comparaison, le noyau sur les marches aléatoires (section 5.5.2) possède une complexité polynomiale pour tous les graphes simples [Vishwanathan et al., 2008].

Une deuxième approche consiste à prendre uniquement une certaine catégorie de chemins. Par exemple, Borgwardt et Kriegel [2005] et Suard et al. [2006] construisent les sacs en considérant, pour chaque paire de sommets distincts, le chemin élémentaire le plus court (ou le plus léger en terme de pondération) reliant les deux sommets formant la paire. Cette technique réduit fortement la taille des sacs de chemins. Cependant, elle ne garantit pas la représentation de toutes les arêtes. En effet, comme seuls les chemins les plus courts sont impliqués, une arête *lourde* peut ne jamais être parcourue si des *raccourcis* existent. Les multigraphes et les

Algorithme 6.1 : Procédure *tousLesChemins* : calcul tous les chemins d'un type donné et d'une taille inférieure à une borne, puis les mets dans un sac de chemins.

Données : Un graphe $G = (V, E)$, une taille maximale pour les chemins $maxLength$ et $tych$ le type de chemin recherché.

Début

| $S = \emptyset$.

| **Pour chaque** $v \in V$ **faire**

| | Soit h le chemin composé de l'unique sommet v .

| | Calculer les chemins partant de v : $calculChemins(G, h, S, maxLength, tych)$.

| **finprch**

Fin

Sorties : Un sac de chemins S .

Algorithme 6.2 : Procédure récursive *calculChemins* : calcul tous les chemins d'un type donné et d'une taille inférieure à une borne, à partir d'un sommet de départ, puis les mets dans un sac de chemins.

Données : Un graphe $G = (V, E)$, h un chemin, un sac de chemins S , une taille maximale pour les chemins $maxLength$ et $tych$ le type de chemin recherché.

Début

| **si** h est un chemin de type $tych$ **alors**

| | $S = S \cup \{h\}$

| **finsi**

| **si** $|h| < maxLength$ **alors**

| | Soit v_{Der} le dernier sommet de h .

| | **Pour chaque** $e \in E$ telle que $v_{Der} \in e$ **faire**

| | | Soit h' le chemin composé de h , de l'arête e et du sommet relié à v_{Der} par e .

| | | Rechercher un chemin plus long : $calculChemins(G, h', S, maxLength, tych)$.

| | **finprch**

| **finsi**

Fin

pseudographes sont encore plus sensible avec les arêtes multiples et de boucles, dont la plus grande partie risque de ne jamais apparaître dans un tel sac de chemins.

Cette heuristique réduit toutefois la complexité temporelle de la construction du sac. Celle-ci dépend essentiellement de celle de l'algorithme de calcul du chemin le plus court entre deux sommets. Nous supposons que de tels chemins sont calculables en $\mathcal{O}(|V|^2 + |E|)$ (avec l'algorithme de Dijkstra, par exemple). Dans ce cas, la complexité temporelle pour la création du sac est en $\mathcal{O}(|V|^4 + |V|^2 |E|)$. Cette méthode est donc plus adaptée pour les grands graphes. De plus, dans le cas des graphes creux, la complexité peut être améliorée en utilisant, par exemple, des tas de Fibonacci.

6.2.1 Pertinence d'une arête

Afin de comparer les chemins qualitativement, nous avons besoin de mesurer la pertinence de chaque chemin. Dans le cas des formes, une grande partie de l'information est portée par les arêtes. Nous avons donc besoin d'une mesure de pertinence des arêtes au sein des graphes.

L'approximation du périmètre, introduite par Torsello et Hancock [2004], induit une estimation de la pertinence d'une arête dans le cas de nos graphes. En effet, une ligature (i.e. un morceau de branche existant uniquement à cause de la contrainte d'homotopie) aura une pertinence faible, car son existence ne provient pas du contour de la forme. De même, une branche devant son existence à une bosse sur le contour aura elle aussi une faible pertinence, puisque le périmètre du contour générateur de cette branche est petit.

Nous pouvons maintenant introduire la pertinence d'un chemin :

Définition 6.2.2 (Pertinence d'un chemin).

La pertinence d'un chemin est la somme des pertinences des arêtes présentes dans le chemin.

6.2.2 Filtrage selon la pertinence

Les chemins étant ordonnable selon leur pertinence, une heuristique consiste à considérer tous les chemins simples (ou élémentaires) jusqu'à une taille fixée. Les chemins sont alors premièrement triés selon leur pertinence, puis seul un pourcentage p des chemins les plus pertinents est conservé. Cette technique permet d'ignorer l'information portée par les chemins les moins pertinents.

Le filtrage permet de comparer les formes en utilisant les sous-parties les plus représentatives. En effet, la perte des chemins de faible pertinence a pour conséquence une perte de la description de certains détails des formes. Lorsque les formes sont assez dissemblables, le choix d'un pourcentage assez petit peut alors être suffisant. En revanche des formes assez ressemblantes demanderont des sacs de chemins plus représentatif des formes, pour être discriminant.

Au niveau de la complexité temporelle, en supposant que nous ayons l chemins dans le sac, d'une taille maximale de s arêtes, le filtrage demande de parcourir chaque chemin pour obtenir sa pertinence (complexité au pire en $\mathcal{O}(ls)$), puis demande un tri (complexité au pire en $\mathcal{O}(l^2)$), d'où une complexité temporelle en $\mathcal{O}(l^2 + ls)$.

6.2.3 Sac de chemins couvrant un graphe

Le filtrage par pourcentage présente des limitations lorsqu'il faut séparer des classes contenant des éléments ressemblants. De plus, le compromis entre peu de chemins et bonne représentativité de la forme est difficile à obtenir avec un pourcentage. En effet, le seuil dépend du graphe, or le pourcentage est généralement le même pour tous. L'idéal serait d'avoir un sac de chemins, peu redondant, avec des chemins pertinents et une représentation quasi-complète de la forme. Cet idéal est proche du problème de couverture de graphe avec une contrainte de cardinalité minimale (sur le nombre de chemins) et de poids maximal. Certaines versions du problème permettent de contraindre les chemins à appartenir à un sac de chemins fourni au préalable [Lin et al., 2006; Guo et al., 2008].

De tels problèmes de couverture sont NP-complet [Lin et al., 2006] dans le cas général. Cependant, dans le cas spécifique des arbres, le problème de couverture peut être résolu en temps polynomial. Néanmoins, si nous ajoutons une contrainte sur l'ensemble des chemins disponibles, le problème devient NP-complet même pour les arbres [Lin et al., 2006; Guo et al., 2008]. Or, pouvoir spécifier l'entrée permet d'avoir un premier sac de chemins calculé par d'autres méthodes ; la couverture servant de filtre.

Lin propose un premier algorithme exact de couverture d'arbre avec une contrainte de cardinalité [Lin et al., 2006]. Celui-ci prend en entrée un ensemble de chemins \mathcal{P} et ressort un ensemble $\mathcal{L} \subset \mathcal{P}$ de chemins de cardinalité minimale et couvrant le graphe. Guo et al. [2008] modifient celui-ci en changeant la contrainte sur la cardinalité de \mathcal{L} en une contrainte sur son poids.

Nous avons modifié ce dernier pour combiner la contrainte de cardinalité et celle de poids maximal pour l'ensemble de sortie. Soient $T = (V, E)$ un arbre, \mathcal{P} un ensemble de chemins dans T , le problème de la couverture de poids maximal avec cardinalité minimale est :

$$\mathcal{L} = \arg \min_{l \subseteq \mathcal{P}} -\omega(l) + \lambda \text{card}(l) \quad \text{sous la contrainte : } l \text{ couvre } T, \quad (6.2)$$

avec $\omega(l)$ le poids du sous-ensemble l et $\text{card}(l)$ le nombre d'éléments dans l . La méthode de résolution est présentée par l'algorithme 6.3. Elle est basée sur le calcul de coût des différentes combinaisons (algorithme 6.4) lui-même basé sur la programmation dynamique : chaque sommet parcouru est une nouvelle étape où la combinaison calculée est dépendante des combinaisons des sommets enfants (voir l'équation (6.4)). Cette dépendance reflète le caractère récursif

Algorithme 6.3 : Algorithme de couverture d'arbre.

Données : Un arbre pondéré $T = (V, E)$ et un ensemble \mathcal{P} de chemins dans T .

Début

1. Enraciner l'arbre à un sommet arbitraire $v_s \in V$.
2. Calcul des combinaisons (matrice \mathbf{A}) à chaque sommet en utilisant l'algorithme 6.4.
3. Calcul d'un parcours préfixé de l'arbre T .
4. Récupérer la combinaison d'énergie minimale :

$\mathcal{L} = \emptyset$.

Pour chaque *sommet* v_i *dans le parcours préfixé faire*

- 4.1. Trouver la combinaison d'énergie minimale couvrant le sommet courant et tout les autres sommets déjà parcouru :

$$\hat{c} = \arg \min_{c \subseteq \mathcal{C}(v_i)} \mathbf{A}[v_i, c \cap \mathcal{L}], \quad (6.1)$$

$\mathcal{C}(v_i)$ étant l'ensemble des chemins couvrant le sommet v_i .

- 4.2. $\mathcal{L} = \mathcal{L} \cup \hat{c}$.

finprch

Fin

Sorties : Un ensemble $\mathcal{L} \subseteq \mathcal{P}$ de chemins de poids maximal et de cardinalité minimale couvrant T .

du parcours de l'arbre. Le calcul de la combinaison finale proposée dans l'algorithme 6.3 est simplement un parcours préfixé de l'arbre enraciné utilisé pour les calculs. La couverture est obtenue en calculant à chaque sommet la combinaison d'énergie minimale en considérant les résultats précédant (voir équation (6.1)). La complexité temporelle finale est en $\mathcal{O}(2^r r q)$ avec q le nombre de chemins en entrée et r le nombre maximal de chemins couvrant un sommet.

L'algorithme 6.3 calcule une couverture des sommets, donc les arêtes ne sont pas toujours couvertes. Pour avoir une couverture d'arêtes, sans modifier les étapes précédentes, nous considérons pour chaque sommet de l'arbre les chemins couvrant celui-ci et son père. Nous ajoutons donc une arête fictive au niveau de la racine pour faire le lien.

La complexité de l'algorithme de couverture est exponentiel selon la redondance des chemins recouvrant un sommet (ou une arête dans notre cas). Il est donc indispensable de se placer dans des conditions acceptables d'utilisation, c'est à dire de limiter la valeur de r . Pour cela, nous avons décidé de filtrer le sac de chemins, donné en entrée, en enlevant les chemins les moins pertinents et non nécessaires à la propriété de couverture, jusqu'à obtenir une redondance acceptable. Une fois la couverture calculée, nous ajoutons dans le sac le symétrique de chaque chemin, afin d'être invariant par symétrie. Le symétrique d'un chemin est défini comme le chemin effectuant le parcours inverse, par exemple, le symétrique du chemin $v_1 e_1 v_2$ est $v_2 e_1 v_1$.

Algorithme 6.4 : Calcul des différentes combinaisons de couverture pour un arbre.

Données : Un arbre pondéré $T = (V, E)$ enraciné au sommet v_s et un ensemble \mathcal{P} de chemins dans T .

Début

1. Initialiser \mathbf{A} contenant les coûts des combinaisons pour chaque sommet :

Pour chaque sommet v_i de T **faire**

1.1 Calculer l'ensemble $\mathcal{C}(v_i) \subset \mathcal{P}$ des chemins couvrant le sommet v_i .

1.2 Calcul le coût de chacune des combinaisons possibles :

Pour chaque combinaison c possible de chemins, $c \subseteq \mathcal{C}(v_i)$ **faire**

$$\mathbf{A}[v_i, c] = \begin{cases} +\infty, & \text{si aucune combinaison n'est sélectionnée} \\ -w(c) + \lambda \text{card}(c), & \text{sinon} \end{cases} \quad (6.3)$$

avec $w(c)$ la somme des poids des chemins impliqués dans la combinaison.

finprch

finprch

2. Mettre dans une file F un parcours suffixe de l'arbre enraciné.

3. Parcourir l'arbre en utilisant le parcours suffixe.

Tant que F n'est pas vide **faire**

3.1 Défiler la tête t de F .

3.2 Récupérer l'ensemble $\mathcal{C}(t) = \{h_1, \dots, h_n, h'_1, \dots, h'_m\} \subset \mathcal{P}$.

3.3 Parcourir les enfants de t pour mettre à jour les coûts :

Pour chaque enfant e de t dans T **faire**

3.3.1 Récupérer l'ensemble $\mathcal{C}(e) = \{h'_1, \dots, h'_m, h''_1, \dots, h''_p\} \subset \mathcal{P}$.

3.3.2 Mettre à jour les coûts de combinaisons :

Pour chaque combinaison c de chemins, $c \subseteq \mathcal{C}(t)$ **faire**

$$\mathbf{A}[t, c] = \mathbf{A}[t, c] + w(c \setminus \mathcal{C}(e)) - \lambda \text{card}(c \setminus \mathcal{C}(e)) + \begin{cases} \mathbf{A}[e, c \cap \mathcal{C}(e)] & \text{si } \mathcal{C}(e) \setminus \mathcal{C}(t) = \emptyset \\ \min_{c' \subseteq \mathcal{C}(e) \cup c} \mathbf{A}[e, c'] & \text{sinon} \end{cases} \quad (6.4)$$

finprch

finprch

fintq

Fin

Sorties : Une matrice \mathbf{A} de coût pour chaque combinaison possible.

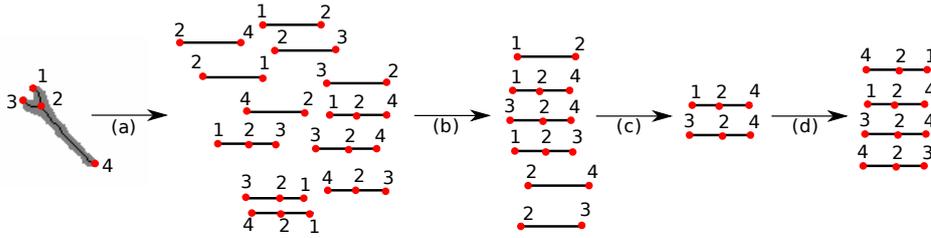


FIG. 6.4: Calcul d'une couverture : (a) extraction des chemins, (b) réduction de la redondance, (c) couverture d'arêtes, (d) ajout des symétriques.

Notons qu'un tel filtrage demande un tri, d'où une complexité temporelle finale en $\mathcal{O}(p^2q)$ avec p le nombre de chemins dans l'ensemble d'entrée et q le nombre de chemins après la réduction de redondance.

Notons que, le problème de couverture de graphe dans le cas général étant assez complexe, nous nous affranchissons du problème en considérant un arbre de recouvrement maximal du graphe. Cette astuce nous permet d'utiliser dans tous les cas l'algorithme 6.3. Bien évidemment, dans ce cas, la solution obtenue n'est plus optimale.

6.3 Noyaux sur sacs de chemins

La décomposition des graphes en sacs de chemins permet d'approximer les graphes par des ensembles de chemins. Le noyau moyenne proposé par Suard et al. [2006] (section 5.5.4) est un parfait exemple d'une vue « ensemble » des graphes. Ce noyau est strictement défini positif sur le domaine des sacs de chemins. Cependant, comme ce noyau compare toutes les paires de chemins possibles, il est sensible au bruit provoqué par des chemins très différents des autres. Il peut donc conduire à de faibles similarités entre deux sacs assez semblables. Afin d'éviter ces problèmes, Suard et al. [2006] introduisent un autre noyau basé sur une détection de changements (section 5.5.4) présentée par Desobry et al. [2005]. Cette détection utilise un SVM une-classe pour estimer une densité de probabilités avec une robustesse aux points aberrants. Nous avons ainsi deux catégories de noyaux : l'une concerne ceux construits à partir d'une convolution, l'autre ceux basés sur l'estimation d'une densité de probabilités.

La section suivante introduit trois nouveaux noyaux. Le premier améliore le noyau moyenne en pénalisant les chemins selon leurs éloignements à la moyenne. Cette pénalité permet de réduire l'apport des chemins aberrants. Les deux autres noyaux sont basés sur des estimations de densités. L'un propose une autre méthode d'utilisation de la détection de changement présentée par Desobry et al. [2005]. L'autre utilise le SVDD pour modéliser une sphère autour des données, puis compare les ensembles à partir d'une distance de Hausdorff entre les sphères.

6.3.1 Noyau moyenne pondérée

Le défaut majeur du noyau moyenne est la dilution de l'information pertinente lorsque les sacs contiennent beaucoup de chemins. En effet, cette opération provoque une répartition équitable de tous les résultats, donc un chemin aberrant participera au résultat avec la même importance que les autres chemins. En considérant un chemin aberrant comme un chemin éloigné du chemin « moyen » dans le sac, la perte d'information peut être réduite en pénalisant les chemins proportionnellement à leur éloignement au chemin « moyen ».

Nous proposons de modifier le noyau moyenne afin d'insérer cette contrainte. Le noyau obtenu K_{pondere} est appelé *noyau moyenne pondéré*. Soient T_1 et T_2 deux sacs de chemins associés respectivement aux graphes G_1 et G_2 , le noyau K_{pondere} est défini par :

$$K_{\text{pondere}}(T_1, T_2) = \sum_{t \in T_1} \sum_{t' \in T_2} (K_{\text{chemin}}(t, m_1) K_{\text{chemin}}(t', m_2))^d p(t/T_1) p(t'/T_2) K_{\text{chemin}}(t, t'), \quad (6.5)$$

avec $d \in \mathbb{R}^+$, m_1 et m_2 les chemins « moyens » des sacs T_1 et T_2 , $p(t/T_1)$ et $p(t'/T_2)$ sont les probabilités marginales associées respectivement aux chemins t et t' . Le produit scalaire entre un chemin t et le chemin « moyen » m du sac de chemins T est défini par :

$$K_{\text{chemin}}(t, m) = \frac{1}{|T|} \sum_{t_i \in T} K_{\text{chemin}}(t, t_i). \quad (6.6)$$

Proposition 6.3.1. *Le noyau K_{pondere} est défini positif dans le domaine des sacs de chemins si et seulement si le noyau sur chemin K_{chemin} est défini positif dans le domaine des chemins.*

Démonstration. Considérons pour chaque chemin $t \in T$, le 3-tuple $(p(t/T), K_{\text{chemin}}(t, m), t)$. Le noyau $K_3(a, b) = (K_{\text{chemin}}(a, m_1) K_{\text{chemin}}(b, m_2))^d p(a/T_1) p(b/T_2) K_{\text{chemin}}(a, b)$ sur les 3-tuples est basé sur un produit de tenseurs (section 5.5) composé de trois noyaux :

- un produit scalaire entre les probabilités,
- un noyau polynomial entre les similarités avec le chemin « moyen »,
- un noyau entre chemins (K_{chemin}).

K_3 est donc défini positif si et seulement si K_{chemin} l'est (section 5.5). K_{pondere} est un noyau de convolution entre les 3-tuples associés à chaque chemin basé sur K_3 . Il est donc défini positif si et seulement si K_3 l'est, c'est à dire si et seulement si K_{chemin} est défini positif. Notons que si K_{chemin} est strictement défini positif, alors K_{pondere} l'est [Haussler, 1999]. \square

Complexité temporelle Soit B_{ch} la complexité du noyau sur chemin, K_{pondere} demande pour chaque chemin h dans un sac T de connaître sa similarité avec le chemin moyen ($\mathcal{O}(|T| B_{\text{ch}})$), puis effectue la somme des comparaisons entre les chemins des deux sacs, donc par construction la complexité temporelle du noyau est en $\mathcal{O}((|T_1|^2 + |T_2|^2 + |T_1| |T_2|) B_{\text{ch}})$.

6.3.2 Noyaux et détection de nouveauté

La détection de nouveauté consiste à déterminer si un ensemble de données apporte de l'information par rapport à un autre ensemble. Si les deux ensembles sont égaux, il n'y a pas d'apport de nouveauté. Considérons les sacs de chemins, nous voulons mesurer la similarité entre deux sacs, ce qui revient à mesurer l'apport d'information d'un sac vis à vis de l'autre. La similarité entre deux sacs de chemins et la détection de changement sont donc liées.

Cependant, la présence de chemins aberrants pose un problème pendant la comparaison de deux formes. Auparavant, nous avons proposé de pénaliser les chemins éloignés du chemin « moyen » de leur sac, afin d'obtenir un noyau sur sacs de chemins strictement défini positif. Cependant, lorsque les sacs possèdent beaucoup de chemins, la pénalisation peut être affaiblie par la forte quantité de chemins. Il peut alors être plus efficace de modéliser les données par une fonction dont les paramètres sont déduits des chemins du sac. Les sacs sont alors comparés avec des distances basées sur cette modélisation.

Nous proposons deux méthodes différentes de modélisation des données, chacune fournissant au final un noyau sur sacs de chemins. La première méthode utilise un SVM une-classe [Schölkopf et al., 2001] pour estimer le support de densité des données et compare les sacs de chemins avec un a priori gaussien. La deuxième méthode suppose que les données sont isotropes et estime une hypersphère englobant les données avec le SVDD (*support vector data description*) introduit par Tax et Duin [2004].

6.3.2.1 A priori gaussien

Supposons que les chemins soient des variables aléatoires présentant une répartition gaussienne dans l'espace engendré par le noyau sur chemins K_{chemin} . Les paramètres de la fonction de densité de probabilités (pdf) peuvent alors être estimés en utilisant le classifieur à une classe ν -SVM. Cette approche a été utilisée par Desobry et al. [2005] pour mesurer la distance entre deux ensembles.

Le principe est le suivant : d'abord les données sont projetées sur la sphère unité dans l'espace engendré par le noyau. Cette projection revient à normaliser le noyau sur chemins :

$$K_{\text{norm}}(h, h') = \frac{K_{\text{chemin}}(h, h')}{\sqrt{K_{\text{chemin}}(h, h)K_{\text{chemin}}(h', h')}}. \quad (6.7)$$

Ce noyau normalisé est défini positif si et seulement si K_{chemin} est défini positif (section 5.5). Ensuite, le classifieur ν -SVM à une classe est appliqué sur les données. La sortie du SVM fournit deux variables : le vecteur de séparation w et la marge ρ . Le paramètre ν du SVM règle le pourcentage d'échantillons à rejeter lors du calcul du vecteur de séparation. Ce paramètre peut être interprété comme un contrôle sur les points aberrants. La variable w peut être interprétée comme une estimation de la moyenne et ρ de la variance. Finalement, les variables w et ρ sont

utilisées comme paramètres de la loi de Gauss relative aux chemins dans l'espace de Hilbert généré par K_{norm} .

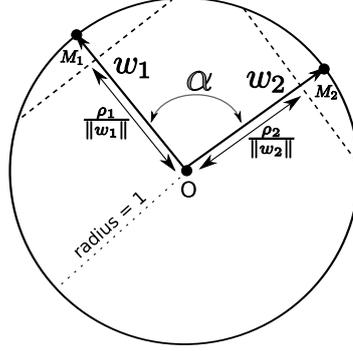


FIG. 6.5: Distance entre les données projetées sur une sphère. L'angle α représente la distance $d_{\text{sphere}}(w_1, w_2)$.

En appliquant ce processus sur deux sacs de chemins T_1 et T_2 , nous obtenons respectivement w_1 et ρ_1 pour T_1 et w_2 et ρ_2 pour T_2 . Les deux sacs sont alors comparés par la distance géodésique α entre les deux moyennes des sacs (w_1 et w_2) sur l'hypersphère (Fig. 6.5) :

$$d_{\text{sphere}}(w_1, w_2) = \arccos \left(\frac{w_1^T K_{1,2} w_2}{\|w_1\| \|w_2\|} \right), \quad (6.8)$$

où $K_{1,2}(i, j) = p(h_i/T_1)p(h_j/T_2)K_{\text{norm}}(h_i, h_j)$, $h_i \in T_1$, $h_j \in T_2$, $p(h_i/T_1)$ et $p(h_j/T_2)$ sont les deux distributions de probabilités sur les chemins et $w_1^T K_{1,2} w_2$ est le produit scalaire entre les vecteurs w_1 et w_2 . Cette distance est directement utilisée pour former un noyau :

$$K_{\text{change}}(T_1, T_2) = \exp \left(\frac{-d_{\text{sphere}}^2(w_1, w_2)}{2\sigma_{\text{moyenne}}^2} \right) \exp \left(\frac{-(\rho_1 - \rho_2)^2}{2\sigma_{\text{origine}}^2} \right). \quad (6.9)$$

Proposition 6.3.2. *Le noyau K_{change} est défini positif dans le domaine des sacs de chemins.*

Démonstration. Le noyau sur les moyennes des sacs de chemins est défini positif car d_{sphere} est une métrique [Berg et al., 1984]. Le noyau entre les variances est aussi défini positif par construction [Berg et al., 1984]. Le noyau K_{change} est constitué du produit de tenseurs entre deux noyaux définis positifs, il est donc défini positif (section 5.5). \square

Complexité temporelle Soit B_{ch} la complexité du noyau sur chemin, nous supposons que l'appel au SVM demande $\mathcal{O}(n^3)$ opérations (n étant la taille des données). Nous devons calculer les matrices de Gram pour T_1 et T_2 , puis faire 2 appels au SVM, et ensuite calculer (6.8). Donc par construction, la complexité temporelle est $\mathcal{O}(|T_1|^3 + |T_2|^3 + (|T_1|^2 + |T_2|^2)B_{\text{ch}} + |T_1||T_2|)$.

6.3.2.2 Données isotropes dans l'espace

Le noyau précédent est basé sur un a priori gaussien dans l'espace engendré par le noyau sur chemins. Cependant, cet a priori gaussien sur la répartition n'est qu'une hypothèse et elle peut ne pas être vérifiée. A la place de cette hypothèse, nous supposons, plus simplement, que les données sont majoritairement situées à une distance bornée d'un point « central ». Nous devons donc rechercher l'hypersphère englobant les chemins dans l'espace engendré par le noyau sur les chemins. Cette hypersphère est estimée en utilisant le SVDD [Tax et Duin, 2004]. La robustesse aux aberrations est assurée par un paramètre ν du SVDD qui correspond au pourcentage de points rejetés lors de l'estimation. Le SVDD permet donc une estimation robuste et fournit pour chaque ensemble le centre et le rayon de l'hypersphère.

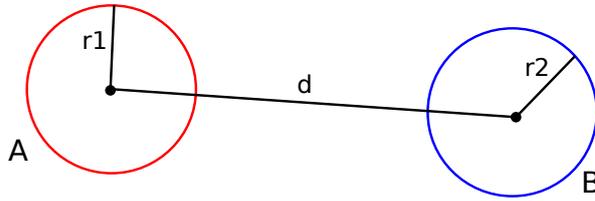


FIG. 6.6: La distance de Hausdorff entre deux cercles est $d + |r_1 - r_2|$.

Nous proposons de comparer deux sphères avec la distance de Hausdorff pour construire un noyau sur sacs de chemins. Soient T_1 et T_2 deux sacs de chemins, l'application du SVDD sur chacun des ensembles donne les variables c_1, r_1 pour T_1 et c_2, r_2 pour T_2 . La distance de Hausdorff entre les deux sphères est définie par :

$$\begin{aligned} d_{\text{SVDD}}(S_1, S_2) &= \max\left\{ \sup_{x \in S_1} \inf_{y \in S_2} \|x - y\|, \sup_{y \in S_2} \inf_{x \in S_1} \|x - y\| \right\}, \\ &= \|c_1 - c_2\| + |r_1 - r_2|, \\ &= \sqrt{\|c_1\|^2 + \|c_2\|^2 - 2c_1^T K_{1,2} c_2} + |r_1 - r_2|. \end{aligned} \quad (6.10)$$

avec $K_{1,2}(i, j) = p(h_i/T_1)p(h_j/T_2)K_{\text{chemin}}(h_i, h_j)$, $h_i \in T_1$, $h_j \in T_2$, $p(h_i/T_1)$ et $p(h_j/T_2)$ sont les deux distributions de probabilités sur les chemins et $c_1^T K_{1,2} c_2$ est le produit scalaire entre les vecteurs c_1 et c_2 . La figure 6.6 présente une illustration de la formule. La distance est directement utilisée pour former un noyau sur sacs de chemins :

$$K_{\text{SVDD}}(T_1, T_2) = K_{\text{SVDD}}(S_1, S_2) = \exp\left(-\frac{d_{\text{SVDD}}^2(S_1, S_2)}{2\sigma_{\text{SVDD}}^2}\right), \quad (6.11)$$

avec σ_{SVDD} la paramètre du noyau.

Proposition 6.3.3. *Le noyau K_{SVDD} est défini positif dans le domaine des sacs de chemins.*

Démonstration. La distance de Hausdorff est une métrique donc le noyau exponentiel construit à partir de cette métrique est défini positif [Berg et al., 1984]. K_{SVDD} est donc défini positif dans le domaine des sacs de chemins. \square

Complexité temporelle Soit B_{ch} la complexité du noyau sur chemin, nous supposons que l'appel au SVDD demande $\mathcal{O}(n^3)$ opérations (n étant la taille des données). Nous devons calculer les matrices de Gram pour T_1 et T_2 , puis faire 2 appels au SVDD, et ensuite calculer (6.10), d'où une complexité temporelle en $\mathcal{O}(|T_1|^3 + |T_2|^3 + (|T_1|^2 + |T_2|^2)B_{\text{ch}} + |T_1||T_2|)$. Nous obtenons un résultat similaire au noyau précédent, ce qui est normal car les deux noyaux font appel aux mêmes mécanismes.

6.4 Noyau hiérarchique de comparaison de chemins

Les noyaux sur sacs de chemins reposent tous sur un noyau sur chemins K_{chemin} . Une version classique de ce noyau basée sur un produit de tenseurs a été présentée (section 5.5.1). Cependant, ce noyau renvoie une valeur non nulle uniquement si les deux chemins comparés ont la même longueur. La comparaison d'un chemin avec une version perturbée de celui-ci (par ajout d'une arête ou d'un sommet) renvoie donc zéro, indiquant que les deux chemins sont extrêmement différents. Les conséquences de ce résultat contre-intuitif sur la comparaison de deux formes sont amplifiées par le processus de squelettisation. En effet, celui-ci est très sensible au bruit sur le contour et induit l'ajout ou la suppression de sommets et d'arêtes, par rapport au graphe initial.

Nous proposons d'éditer les chemins pour enlever les sommets ou arêtes, potentiellement défectueux, afin d'améliorer les comparaisons. Pour cela nous construisons des opérations d'éditions en partant de l'observation des conséquences du bruit. Ces opérations sont ensuite utilisées pour construire un noyau basé sur une comparaison hiérarchique des chemins.

6.4.1 Observations

Le processus de squelettisation est sensible aux déformations le long du contour. Elles proviennent principalement de deux phénomènes : l'élongation de la forme (ou d'une partie) (Fig. 6.7(b)) et l'apparition de bosses sur les contours (Fig. 6.7(c)). Les deux phénomènes ont des répercussions immédiates sur le squelette et donc sur le graphe. Le bruit provoqué est de nature structurelle car il implique des modifications en profondeur dans les graphes.

L'élongation consiste en une dilatation, localisée ou non, de la forme. La discrétisation des images est susceptible de créer de telles dilatations à cause des erreurs de quantifications. Au niveau du squelette, l'élongation tend à créer une branche ou allonger une branche existante. Pour de faibles déformations, les branches ainsi créées sont associées à un faible pourcentage

du contour global. Il convient donc d'utiliser cette propriété pour caractériser ces branches non pertinentes.

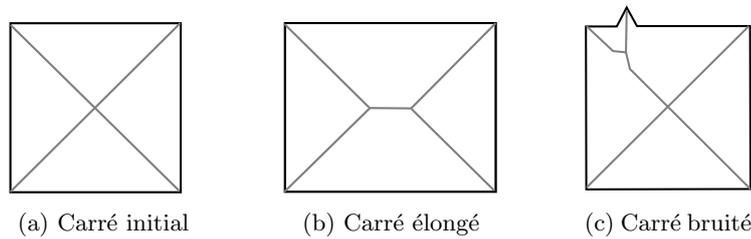


FIG. 6.7: Effets de certaines déformations de la forme sur le squelette.

Les bosses sur les contours sont souvent le résultat d'une mauvaise segmentation ou d'une erreur de discrétisation. Les répercussions au niveau du squelette sont doubles : ajout de branches et ajout de jonctions. De plus, les branches induites par les bosses correspondent souvent à des ligatures. En effet, ces liaisons entre la branche provoquée par une bosse et le reste du squelette sont généralement générées par les contraintes d'homotopie.

La combinaison de ces deux perturbations provoque de sérieuses modifications au sein des graphes. Deux formes semblables peuvent alors conduire à deux graphes très différents structurellement. Nous proposons de construire deux opérations d'édition afin d'avoir une meilleure robustesse au bruit lors de la comparaison de chemins.

6.4.2 Opérations d'édition

La bruit structurel a deux incidences sur les chemins : ajout d'arêtes et ajout de sommets. Nous avons donc besoin de deux opérations pour inverser les effets des perturbations : *suppression de sommet* et *suppression d'arête*. L'opération de suppression de sommet consiste à enlever un sommet, puis fusionner les deux arêtes incidentes au sommet dans le chemin (figure 6.8(b)), les autres arêtes incidentes sont oubliées. L'opération de contraction d'arête comprend la suppression d'une arête et la fusion des deux sommets aux extrémités s'ils sont différents (figure 6.8(c)). Dans le cas d'une boucle, l'arête est simplement supprimée.

A la suite de chaque opération, les attributs des sommets et arêtes du chemin sont mis à jour afin de répercuter les changements. La mise à jour permet de transformer les opérations d'édition en opération de transformation de la forme. En effet, la suppression d'un sommet revient à remettre en cause la partie de la forme reliée au chemin par le sommet et la suppression d'une arête revient à remettre en cause la partie de la forme à l'origine de la branche du squelette représentée par l'arête.

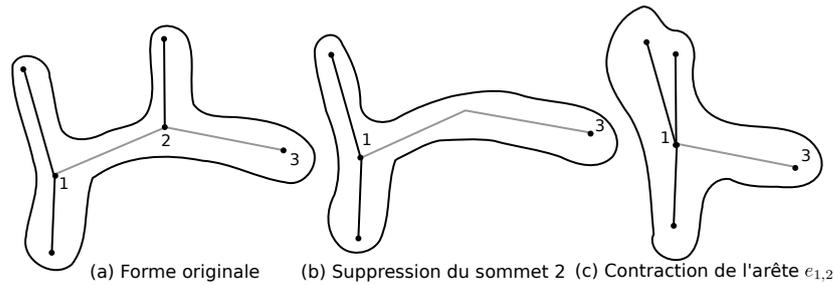


FIG. 6.8: Répercussions des opérations d'éditions sur la forme.

6.4.2.1 Pertinence des éléments d'un chemin

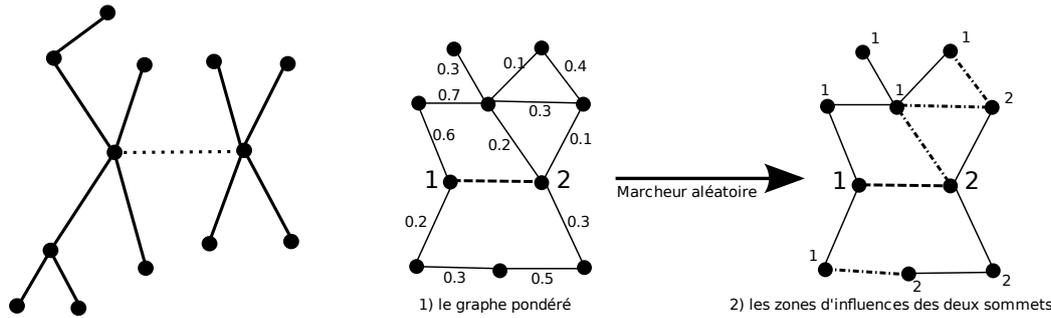
Nous supposons qu'une mesure de pertinence est disponible pour chaque arête. Une telle mesure a déjà été utilisée pour ordonner les chemins (section 6.2). Nous allons aussi l'utiliser pour associer un coût à chaque opération. Pour cela, la pertinence de chaque élément présent dans un chemin est calculée : celle d'une arête est directement disponible, celle d'un sommet est la somme des pertinences des arêtes du sous-graphe relié au chemin par ce sommet. Dans le cas des arbres, les sous-graphes sont des sous-arbres dont la pertinence peut être calculée sans ambiguïté (Fig. 6.9(a)).

Cependant, dans le cas général, avec les formes trouées, les graphes ne sont pas des arbres et le calcul de la pertinence du sous-graphe associé à un sommet n'est pas direct. En effet, les sommets d'un chemin peuvent être reliés entre eux en dehors du chemin (Fig. 6.9(b)). Nous proposons de résoudre ce problème en utilisant le marcheur aléatoire [Grady, 2006], avec comme probabilités de transition les pertinences normalisées des arêtes.

L'algorithme est le suivant : soit le chemin $h = v_1 e_1 \dots v_n$, nous associons le label i au sommet v_i du chemin. Le marcheur aléatoire associe à chaque sommet $v'_k \in V$ du graphe $G = (V, E)$ une probabilité $p_{k,i}$ pour chaque label i . $p_{k,i}$ est la probabilité que le marcheur partant du sommet v_i atteigne le sommet v'_k . Enfin, chaque sommet est labellisé avec le label associé à la probabilité la plus grande. Le sous-graphe obtenu pour chaque sommet v_i du chemin est appelé la *zone d'influence* du sommet v_i .

Cependant, cette méthode est conçue pour des graphes simples, or nos graphes ne le sont pas toujours. Toutefois, dans la mesure où nous nous intéressons uniquement à une labélisation des sommets, le graphe peut être simplifié sans perte d'information. Ainsi, les boucles sont ignorées car elles n'interviennent pas dans le découpage en zone. Puis, les arêtes multiples sont fusionnées en une seule arête dont la pertinence est la somme de leur pertinence. Donc, le graphe est simple lors du calcul des zones d'influences. Les modifications proposées sont évidemment appliquées uniquement pour le calcul des zones d'influences.

La pertinence d'un sommet du chemin est calculée en sommant 1) la pertinence des arêtes de



(a) Les sous-arbres reliés au chemin par ses sommets sont clairement définis.

(b) Le marcheur aléatoire permet de définir les zones d'influences associées à chaque sommet d'un chemin.

FIG. 6.9: Zone d'influence avec le marcheur aléatoire. Le chemin considéré est représenté en pointillés (--). Les arêtes partagées par les deux zones sont représentées par une ligne hachurée et pointée (-·-).

la zone d'influence associé au sommet et 2) une portion de la pertinence des arêtes partagées avec des sommets d'une autre zone d'influence. La portion est déterminée selon les probabilités maximales associées aux deux sommets : soient v_1 et v_2 les deux sommets d'une arête d'une pertinence w , v_1 est associée à sa zone d'influence avec une probabilité p_1 , v_2 avec une probabilité p_2 , la part de la pertinence de l'arête revenant à la zone d'influence associée à v_1 est définie par $\frac{p_1}{p_1+p_2}w$ et réciproquement $\frac{p_2}{p_1+p_2}w$ pour v_2 . La figure 6.9(b) montre un exemple de calcul de zone d'influences pour un chemin composé de deux sommets. Notons que la zone d'influence du sommet 1 est bien plus grande que celle du sommet 2.

6.4.2.2 Coût d'une opération d'édition

De manière à sélectionner les opérations les plus appropriées, un coût est associé à chaque opération. Ce coût reflète la perte d'information induite par l'édition selon l'importance des changements dans la forme. L'importance d'une arête est directement liée à sa pertinence. Toutefois, l'importance d'un sommet reflète deux points : sa pertinence (soit l'importance du sommet en tant que jonction) et la valeur de l'inflexion au niveau du sommet (soit l'importance du sommet en tant que point d'inflexion). Le coût de suppression d'un sommet exprime donc ses deux origines possibles.

Considérons un chemin $h = v_1 e_1 \dots v_n$ dans un graphe $G = (V, E)$, les coûts des opérations d'éditions au sein du chemin sont alors définis par :

- le coût d'une *suppression d'un sommet* dépend de 1) la pertinence de la zone d'influence associée au sommet et de 2) l'inflexion au niveau du sommet. La pertinence du sommet est calculée en utilisant les méthodes présentées dans la section précédente (section 6.4.2.1).

Nous proposons de quantifier l'importance de l'inflexion en utilisant l'angle formé par les pentes des deux arêtes incidentes e_i et e_{i+1} au sommet v_i dans le chemin. Or celles-ci sont représentées par des splines de premier ordre (section 6.1.1). Soient a_i la pente associée à l'arête e_i et a_{i+1} celle associée à e_{i+1} , l'angle $\beta(v_i)$ est donc formulé par :

$$\beta(v_i) = \arccos \left(\frac{1 + a_i a_{i+1}}{\sqrt{1 + a_i^2} \sqrt{1 + a_{i+1}^2}} \right). \quad (6.12)$$

Finalement, le coût de suppression d'un sommet est, en normalisant l'angle :

$$(1 - \gamma_{\text{deform}})\omega(v_i) + \frac{2}{\pi}\gamma_{\text{deform}}\beta(v_i), \quad (6.13)$$

avec $\omega(v_i)$ la pertinence de la zone d'influence du sommet v_i et γ_{deform} une variable d'équilibre.

– le coût de *contraction d'une arête* est directement égal à sa pertinence.

Une fois les coûts d'éditions définis, nous pouvons construire un noyau utilisant une comparaison hiérarchique des chemins, cette hiérarchie étant fournie par une application séquentielle des opérations d'éditions.

6.4.3 Comparaison hiérarchique des chemins

Nous proposons de construire un noyau sur chemin basé sur K_{chemin} qui compare deux chemins à partir de leurs différentes réécritures. Soit \mathcal{C} l'ensemble des chemins, la fonction d'édition $\kappa : \mathcal{C} \mapsto \mathcal{C}$ est définie comme celle appliquant l'édition de moindre coût sur le chemin d'entrée et renvoyant la réécriture obtenue. Cette fonction peut être composée pour réaliser plusieurs éditions : $\kappa^i(t) = \underbrace{\kappa \circ \kappa \circ \dots \circ \kappa}_i(t)$. Le coût cumulé des éditions appliquées pour obtenir $\kappa^i(t)$ à partir de t est noté $\text{cout}_i(t)$. Le noyau sur chemin avec éditions est alors construit comme un noyau de convolution entre les chemins et leurs réécritures pondérées selon les coûts :

$$K_{\text{edit}}(t, t') = \frac{1}{m_{\text{edit}} + 1} \sum_{k=0}^{m_{\text{edit}}} \sum_{l=0}^{m_{\text{edit}}} \exp \left(-\frac{\text{cout}_k(t) + \text{cout}_l(t')}{2\sigma_{\text{cout}}^2} \right) K_{\text{chemin}}(\kappa^k(t), \kappa^l(t')), \quad (6.14)$$

avec m_{edit} le nombre maximal d'édition et σ_{cout} le paramètre du noyau RBF entre les coûts d'éditions.

Proposition 6.4.1. K_{edit} est défini positif si et seulement si K_{chemin} est défini positif.

Démonstration. Considérons un ensemble $\mathcal{E}(t)$ qui contient les $m_{\text{edit}} + 1$ couples $(\kappa^i(t), \text{cout}_i(t))$ issus des réécritures d'un chemin t .

Le noyau entre les coûts d'édits est un produit scalaire car :

$$\begin{aligned} \exp\left(-\frac{\text{cout}_k(t) + \text{cout}_l(t')}{2\sigma_{\text{cout}}^2}\right) &= \exp\left(-\frac{\text{cout}_k(t)}{2\sigma_{\text{cout}}^2}\right) \exp\left(-\frac{\text{cout}_l(t')}{2\sigma_{\text{cout}}^2}\right) \\ &= \langle \phi(\text{cout}_k(t)), \phi(\text{cout}_l(t')) \rangle \end{aligned}$$

avec $\phi : x \rightarrow \exp\left(-\frac{x}{2\sigma_{\text{cout}}^2}\right)$. K_{edit} est donc un noyau de convolution (à une constante positive près) entre deux ensembles $\mathcal{E}(t)$ et $\mathcal{E}(t')$ (t et t' étant deux chemins) basé sur un produit de tenseurs. Celui-ci est défini par un noyau sur chemin K_{chemin} et un produit scalaire entre les coûts d'édits.

En appliquant le théorème de Haussler [1999], K_{edit} est défini positif si et seulement si K_{chemin} est défini positif. De plus, si K_{chemin} induit une métrique dans le domaine des chemins alors K_{edit} induit une métrique dans le domaine des chemins *réécrits*. \square

La particularité de ce noyau réside dans l'indépendance des processus d'édition. En effet, il ne cherche pas à transformer un chemin en un autre, mais plutôt à enlever les éléments susceptibles d'être corrompus. En cela les processus ne forment pas une distance d'édition entre deux chemins. Notre approche diffère donc fortement de celle de Neuhaus et Bunke [2006] qui construisent des noyaux directement à partir des distances d'édits.

Complexité temporelle Soient $G = (V, E)$ et $G' = (V', E')$ deux graphes, h un chemin dans G et h' un chemin dans G' . Nous notons B_{ch} la complexité temporelle du noyau K_{chemin} . La création des hiérarchies demande pour chaque application de $\kappa : 1$) des parcours des chemins ($\mathcal{O}(|h| + |h'|)$), 2) le calcul des zones d'influences ($\mathcal{O}(|V|^3 + |V'|^3)$) [Grady, 2006] et 3) la mise à jour des graphes ($\mathcal{O}(|V| + |E| + |V'| + |E'|)$). Nous obtenons donc une complexité temporelle en $\mathcal{O}(|V|^3 + |V'|^3 + m_{\text{edit}}B_{\text{ch}})$ pour K_{edit} .

6.5 Prise en compte des trous

Les faces d'un graphe issues d'un squelette dépeignent les trous présents dans la forme. Elles représentent donc des caractéristiques importantes de la forme. Cependant, les différents noyaux présentés ne gèrent pas spécifiquement cette information. Au mieux, les faces sont englobées dans quelques chemins sans emphase particulière.

Plusieurs noyaux sur cycles [Horváth et al., 2004; Horváth, 2005] ont déjà été proposés dans le cadre de la classification de molécules chimiques. Cependant, les faces sont des cycles particuliers avec une notion topologique, or ces noyaux ne prennent pas en compte l'orientation et ne sont pas restreint aux faces. Afin d'exploiter ces informations, nous construisons un noyau sur faces et donc sur les trous.

Une comparaison efficace des faces, dans le cadre de la reconnaissance des formes, demande, encore une fois, un noyau robuste au bruit structurel. Les effets des perturbations sont évidemment les mêmes que pour les chemins, c'est à dire ajout de sommets et d'arêtes. Afin de simplifier les comparaisons, nous représentons chaque face par un chemin simple fermé. Comme convention, le sommet de départ est celui le plus proche du centre de gravité de la forme, puis le chemin parcourt les arêtes de la face dans le sens trigonométrique. Ainsi, les faces sont toutes représentées de la même manière et peuvent être directement comparées avec K_{chemin} .

6.5.1 Noyau sur faces

Dans ce cadre, une comparaison robuste demande de gérer le bruit structurel et les erreurs provoquées par des sommets de départ mal alignés. En effet, le choix du sommet est parfois délicat lorsque plusieurs sommets ont des distances au barycentre de la forme équivalentes. Nous corrigeons cet effet par l'adjonction d'un processus de décalage des chemins à celui de la comparaison. Ainsi, nous nous introduisons la fonction $\mu_i(t)$ qui renvoie la version du chemin t après i décalages dans le sens horaire si i est positif et dans le sens trigonométrique sinon.

Le noyau sur faces est alors construit comme un noyau de convolution entre ensembles contenant un chemin et ses permutations :

$$K_{\text{permut}}(t, t') = \frac{1}{(2p+1)^2} \sum_{i=-p}^p \sum_{j=-p}^p \exp\left(-\frac{|i|+|j|}{2\sigma_{\text{permut}}^2}\right) K_{\text{chemin}}(\mu_i(t), \mu_j(t')), \quad (6.15)$$

avec p le nombre maximal de permutations.

Proposition 6.5.1. K_{permut} est défini positif si et seulement si le noyau sur chemin K_{chemin} .

Démonstration. Considérons $\mathcal{E}(t)$ un ensemble contenant les différents couples $(\mu_i(t), i)$. Le noyau sur les couples est un noyau basé sur un produit de tenseurs, avec un noyau sur chemin K_{chemin} et un noyau sur le nombre de permutations effectuées. Le noyau K_{permut} est donc un noyau de convolution, à la constante positive près, entre deux ensembles $\mathcal{E}(t)$ et $\mathcal{E}(t')$ où t et t' sont deux chemins simples et fermés. K_{permut} est donc défini positif si et seulement si K_{chemin} est défini positif [Haussler, 1999]. \square

Complexité temporelle Soit B_{ch} la complexité de K_{chemin} . K_{permut} demande de calculer les permutations de chaque chemin et de comparer les comparer à travers l'ensemble des possibilités, d'où une complexité temporelle en $\mathcal{O}((2p+1)^2 B_{\text{ch}})$.

Remarque 6.5.2.

La robustesse au bruit structurel peut être apportée en utilisant K_{edit} à la place de K_{chemin} dans (6.15). Une autre possibilité est d'utiliser un noyau d'alignement de chaîne de caractères robuste aux substitutions, ajouts ou suppressions d'éléments [Cuturi et al., 2006].

6.5.2 Noyau sur ensembles de faces

Finalement, le noyau de comparaison des faces est construit comme une convolution entre les ensembles de faces. Soient G_1 et G_2 deux graphes planaires, $\mathcal{K}(G_1)$ l'ensemble des faces du graphe G_1 de cardinal $|\mathcal{K}(G_1)|$, $\mathcal{K}(G_2)$ l'ensemble des faces du graphe G_2 de cardinal $|\mathcal{K}(G_2)|$. Le noyau sur faces est défini par :

$$K_{\text{faces}}(G_1, G_2) = K_{\text{faces}}(\mathcal{K}(G_1), \mathcal{K}(G_2)) = \frac{1}{|\mathcal{K}(G_1)|} \frac{1}{|\mathcal{K}(G_2)|} \sum_{t \in \mathcal{K}(G_1)} \sum_{t' \in \mathcal{K}(G_2)} K_{\text{permut}}(t, t'). \quad (6.16)$$

Proposition 6.5.3. K_{faces} est défini positif sur l'ensemble des chemins simples et fermés codant des faces si et seulement si K_{permut} est défini positif sur le domaine des chemins simples et fermés.

Démonstration. K_{faces} est un noyau de convolution entre ensembles de chemins codant des faces, il est donc défini positif si et seulement si K_{permut} est défini positif [Haussler, 1999]. \square

Les formes considérées présentent généralement peu de trous, le calcul des faces du graphe issue du squelette est alors assez direct. Dans notre application, nous commençons par repérer les trous dans les formes, puis les faces sont directement extraites du graphe en recherchant les chemins encodant les régions cernant le trou considéré. Néanmoins, dans le cas général, l'extraction de toutes les faces devient un problème complexe. Des heuristiques peuvent alors être nécessaires pour rechercher en priorité les faces les plus caractéristiques de la forme.

Complexité temporelle Soit B_{perm} la complexité de K_{permut} . K_{faces} demande de calculer les faces des graphes, puis compare les ensembles avec une convolution. Nous supposons que les chemins représentant les faces sont obtenus avec un algorithme de complexité B_{faces} . Soient G_1 et G_2 deux graphes, $\mathcal{K}(G_1)$ l'ensemble des faces de G_1 de cardinal $|\mathcal{K}(G_1)|$ et respectivement pour G_2 . K_{faces} possède donc une complexité temporelle en $\mathcal{O}((|\mathcal{K}(G_1)| + |\mathcal{K}(G_2)|)B_{\text{faces}} + |\mathcal{K}(G_1)||\mathcal{K}(G_2)|B_{\text{perm}})$.

6.5.3 Combinaisons de noyaux

Le noyau final de comparaison de graphe codant des squelettes est construit en combinant le noyau sur ensembles de faces K_{faces} à un noyau sur sacs de chemins K_{sacs} :

$$K_{\text{combinaison}}(G_1, G_2) = (1 - \gamma_{\text{combinaison}})K_{\text{sacs}}(G_1, G_2) + \gamma_{\text{combinaison}}K_{\text{faces}}(G_1, G_2), \quad (6.17)$$

avec $\gamma_{\text{combinaison}} \in [0, 1]$ une variable d'ajustement.

Proposition 6.5.4. Le noyau $K_{\text{combinaison}}$ est défini positif si et seulement si K_{sacs} et K_{faces} sont défini positif.

Démonstration. Ce résultat tient aux propriétés classiques des noyaux : multiplication de noyaux par une constante positive et somme directe entre un noyau sur sacs de chemins et un noyau sur sacs de faces (section 5.5). \square

6.6 Expérimentations

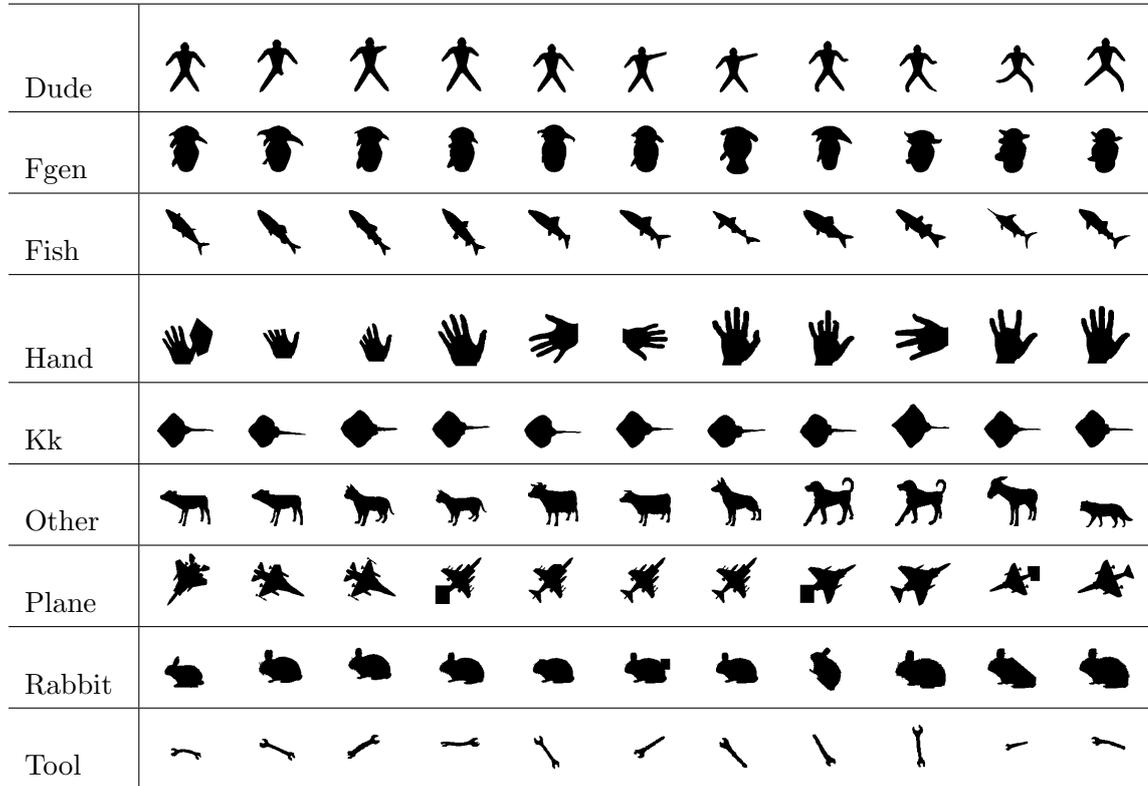


FIG. 6.10: Les 9 classes formant la base de 99 formes.

La performance de notre approche a été testée sur la base de silhouettes du laboratoire LEMS¹. Cette base de 99 formes est répartie en 9 classes de 11 formes (Fig. 6.10). Sept noyaux ont été comparés lors des expériences (résumés dans Tab. 6.2) :

- 3 d'entre eux sont construits en combinant $K_{pondere}$ avec 3 différentes méthodes de construction de sacs de chemins (5% des chemins les plus pertinents, 10% et la couverture avec Alg. 6.3) et en utilisant le processus d'édition (avec K_{edit} , section 6.4.3) : $K_{edit,5\%}$, $K_{edit,10\%}$ et $K_{edit,couverture}$.
- Un noyau combinant $K_{pondere}$ sans processus d'édition avec les sacs construits avec la couverture pour $K_{classique,couverture}$ et 5% des chemins les plus pertinents pour $K_{classique,5\%}$.

¹<http://www.lems.brown.edu/vision/software>

- Le noyau sur les marches aléatoires (section 5.5.2) implémenté en utilisant l'une des méthodes rapides proposées par Vishwanathan et al. [2008] : K_{marches} .
- Afin de comparer avec les approches basées sur les distances d'édition [Neuhaus et Bunke, 2007b], nous avons testé un dernier noyau :

$$K_{\text{Neuhaus}}(G_1, G_2) = \exp\left(\frac{-d_{\text{edit}}^2(G_1, G_2)}{2\sigma_{\text{Neuhaus}}}\right), \quad (6.18)$$

avec d_{edit} une distance d'édition adaptée à nos graphes. Pour bénéficier d'une complexité comparable aux autres noyaux, la distance d'édition était construite à partir de la méthode d'approximation de Riesen et al. [2007b], elle-même utilisée récemment par [Bunke et Riesen, 2007; Riesen et al., 2007a] pour la construction de leur noyau sur graphes. Notons que ce noyau n'est évidemment pas défini positif.

Noyaux	Méthode de construction
$K_{\text{edit,couverture}}$	K_{pondere} avec le processus d'édition et le sac de chemins contenant la couverture du graphe
$K_{\text{edit,5\%}}$	K_{pondere} avec le processus d'édition et le sac de chemins contenant 5% des chemins les plus pertinents
$K_{\text{edit,10\%}}$	K_{pondere} avec le processus d'édition et le sac de chemins contenant 10% des chemins les plus pertinents
$K_{\text{classique,couverture}}$	K_{pondere} avec le sac contenant la couverture du graphe
$K_{\text{classique,5\%}}$	K_{pondere} avec le sac contenant 5% des chemins les plus pertinents
K_{marches}	Comparaisons de tous les chemins possibles dans chaque graphe.
K_{Neuhaus}	Distance d'édition dans un noyau exponentiel

TAB. 6.2: Les différents noyaux impliqués dans les expériences.

6.6.1 Protocole expérimental

Plusieurs expériences ont été conduites. La première avec l'objectif de tester la robustesse des différentes approches à l'ajout ou la suppression de barbules et de points d'inflexions. Pour cela, nous avons fait varier deux paramètres de squelettisation :

Le seuil d'extraction du squelette qui représente l'équilibre entre le bruit et l'axe médian dans la méthode proposée par Siddiqi et al. [2002]. Plus ce seuil est élevé, moins il y a de branches, donc de barbules. Le seuil par défaut est de 0.4.

L'écart type σ_{BIC} du critère BIC utilisé lors de la détection des points d'inflexions. σ_{BIC} représente la sensibilité de la méthode au changement de pente, plus il est petit, plus les changements seront nombreux. La valeur par défaut est 0.1.

Les valeurs par défauts sont celles utilisées par les graphes *normaux* formant la base de formes. Nous avons pris pour chacun des deux paramètres 20 valeurs comprises dans $(0, 1.0]$. La classe outils (*Tool*) a été arbitrairement choisie pour tester les effets de ces paramètres. Ensuite, pour chaque valeur et chaque outil, un nouveau graphe est créé avec la valeur par défaut pour l'un des paramètres et la nouvelle valeur pour l'autre. Enfin, les distances ont été calculées avec la distance induite ($d_{\text{induite}}(G, G') = \sqrt{K(G, G) + K(G', G') - 2K(G, G')}$, K un noyau) entre ce nouveau graphe et l'ensemble de la base, puis comptons le nombre d'outils parmi les 11 formes les plus proches. Ainsi, nous obtenons un indice de la résistance au bruit de chaque noyau.

La deuxième expérience consistait en la classification de toutes les formes de la base (Fig. 6.10) en 9 classes avec l'analyse discriminante quadratique à noyau introduite par Haasdonk et Pękalska [2008]. La base d'entraînement était constituée de 5 formes choisies arbitrairement par classe. Le but de cette expérience était de comparer les performances des différents noyaux en terme de classification d'un ensemble de formes, avec ou sans processus d'édition, et selon la construction du sac de chemins. Nous avons aussi comparés nos noyaux avec deux noyaux de l'état de l'art (marches aléatoires, K_{marches} et noyau basé sur une distance d'édition, K_{Neuhaus}).

La troisième expérience reprenait l'idée de la deuxième avec le même classifieur, mais en ajoutant à la base de 99 formes (figure 6.10) 6 classes de 11 formes (figure 6.11) formant ainsi une base de 165 formes. Les nouvelles formes possédant pour la plupart d'entre elles un ou plusieurs trous caractéristiques, l'objectif de cette expérience était de montrer l'intérêt de combiner un noyau sur sac de chemins (construit avec la couverture) avec un noyau sur les faces pour classer de telles formes. Nous avons aussi testé avec ou sans le processus d'édition afin de déterminer ses avantages et inconvénients. La base d'entraînement était constituée de 5 formes choisies arbitrairement, pour chaque classe.

Pour chacune des expériences, la taille maximale des chemins était fixée à 6 arêtes, le nombre maximal d'édition à 6 et le paramètre λ (6.2) d'équilibre de l'algorithme de couverture à 1.0. Les différents σ des noyaux gaussiens et les autres paramètres présents dans les noyaux ont été estimés en effectuant un parcours de grille. Pour l'indexation, nous avons cherché à avoir pour chaque outil un maximum d'outils parmi les 11 premiers. Pour la classification, nous avons appliqué une validation croisée avec 5 répertoires (*5-folds*) en recherchant à maximiser la précision ($\frac{\text{tp}}{\text{tp}+\text{fp}}$, avec tp le nombre de vrais positifs (formes correctement classées) et fp le nombre de faux positifs (formes mal classées)). Compte tenu du grand nombre de paramètres et afin d'obtenir un résultat dans des temps raisonnables, nous supposons qu'ils sont indépendants. La valeur maximale présente dans la grille est alors recherchée en effectuant des parcours

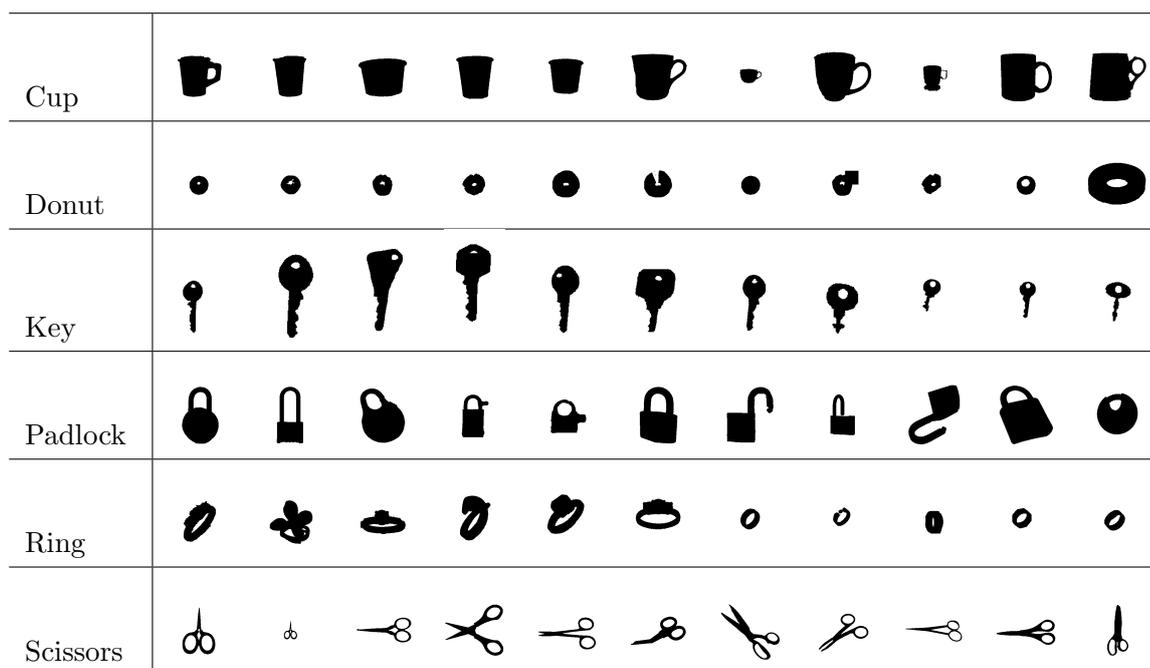


FIG. 6.11: Les 6 classes de formes dont le ou les trous forment une information importante lorsqu'ils sont présents.

linéaires pour chaque paramètre et en fixant alors les autres avec leur meilleure valeur. La table 6.3 présente les paramètres, avec leur interprétation et la plage de valeurs utilisées.

6.6.2 Discussion des résultats

6.6.2.1 Résistance au bruit

La figure 6.12 illustre le nombre moyen d'outils de la base de 99 formes parmi les 11 plus proche pour chaque outil déformé. $K_{\text{edit},5\%}$ présente une bonne robustesse aux deux bruits. Il est assez stable aux variations du seuil d'extraction du squelette, mais cette robustesse décroît lorsque le nombre de barbules est trop important (seuil faible). Les mêmes remarques sont valables pour les points d'inflexion, si le nombre de points est trop élevé, le noyau est inefficace, mais les performances se stabilisent une fois que le nombre de points est en dessous d'un certain seuil. Notons, que le noyau montre les meilleurs résultats sur l'ensemble, avec une bonne stabilité. Les faibles résultats du noyau $K_{\text{classique},5\%}$ laisse supposer que la stabilité du noyau précédent provient du processus d'édition. Remarquons que $K_{\text{classique},5\%}$ est meilleur que $K_{\text{edit},5\%}$ quand le nombre de barbules est important (figure 6.12(a)). Nous supposons que dans ce cas le processus d'édition produit l'effet inverse de celui voulu. En effet, comme il y a peu de chemins qui sont fortement dégradé, l'approche de comparaison par hiérarchie de chemins est incapable de faire ressortir l'information importante voire la dégrade davantage.

Paramètre	Interprétation	Plage de valeurs
d	Puissance dans K_{pondere} (6.5), pondération du terme de pénalisation	6 valeurs entières dans $[0, 5]$
$\sigma_{\text{perimètre}}$	Distance maximale entre les valeurs de périmètres	30 valeurs équidistantes dans $[0, 2]$
σ_{cout}	Proportionnel au coût maximal d'édition	30 valeurs équidistantes dans $[0, 2]$
$\sigma_{\text{gravité}}$	Distance maximale entre les distances aux centres de gravité	30 valeurs équidistantes dans $[0, 2]$
σ_{rayon}	Distance maximale entre les coefficients des polynômes représentant l'évolution du rayon	30 valeurs équidistantes dans $[0, 100]$
$\sigma_{\text{orientation}}$	Distance maximale entre les coefficients des polynômes représentant l'évolution de l'orientation	30 valeurs équidistantes dans $[0, 100]$
σ_{Neuhaus}	Distance maximale entre deux graphes	30 valeurs équidistantes dans $[0, 10]$
γ_{deform}	Equilibre entre le coût d'édition et celui de déformation pour les sommets (6.13)	10 valeurs équidistantes dans $[0, 1]$
σ_{permute}	Distance maximale entre deux chemins représentant des trous	30 valeurs équidistantes dans $[0.1, 2]$
p	Le nombre maximal de permutations	$\{0, 1, 2, 3, 4, 5\}$
$\gamma_{\text{combinaison}}$	Equilibre entre le noyau sur sacs de chemins et le noyau sur les faces	15 valeurs équidistantes dans $]0, 1]$

TAB. 6.3: Les plages de valeurs testées pour chacun des paramètres.

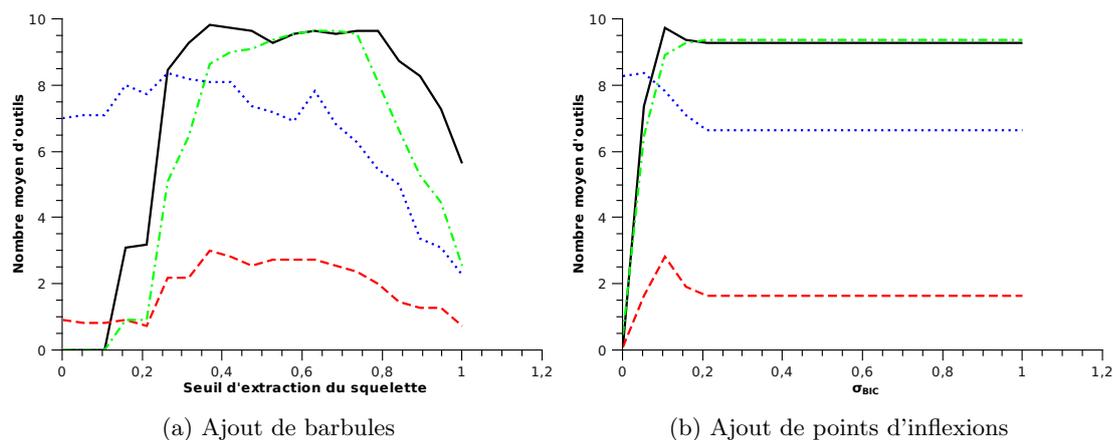


FIG. 6.12: Résistance des noyaux aux deux types de déformations : ajout de barbules et ajout de points d'inflexions. Les noyaux testés sont $K_{\text{edit},5\%}$ (—), $K_{\text{classique},5\%}$ (- -), K_{marches} (-·-) et K_{Neuhaus} (··).

K_{marches} montre des résultats très proche de $K_{\text{edit},5\%}$ voire le dépasse parfois. Cependant, il ne possède pas la robustesse de ce dernier. Ceci confirme notre opinion sur le processus d'édition. Notons que K_{marches} utilise tous les chemins présent dans les graphes, quand $K_{\text{edit},5\%}$ prend 5% des chemins parmi tous les chemins de taille allant jusqu'à six arêtes. Nous obtenons donc des résultats similaires en utilisant des parcours de graphes très différents. Nous supposons que ceci est possible, car peu de chemins sont nécessaires pour obtenir une majeure partie de l'information sur la forme.

K_{Neuhaus} présente la meilleure robustesse au bruit aussi bien avec les barbules que les points d'inflexions. En revanche, il n'atteint pas les performances de K_{marches} ou $K_{\text{edit},5\%}$ pour des graphes faiblement perturbés. La distance d'édition permet donc d'obtenir une excellente résistance au bruit, mais montre de faibles performances en terme d'indexation. Ceci provient probablement de l'algorithme d'approximation utilisé pour le calcul de la distance [Riesen et al., 2007b]. En effet, celui-ci repose sur un appariement bipartite, mais ne modélise pas toujours fidèlement le coût de déformation. L'utilisation d'une autre distance permettrait sûrement d'avoir de meilleurs résultats, mais avec un prix élevé en terme de complexité.

6.6.2.2 Classification de formes

La table 6.4 montre les matrices de confusions obtenues avec 6 des 7 noyaux. $K_{\text{edit},\text{couverture}}$ présente les meilleurs résultats avec une légère confusion sur la classe Other, mais celle-ci com-

Classes	$K_{\text{edit,couverture}}$									$K_{\text{edit,5\%}}$								
	Dude	Fgen	Fish	Hand	Kk	Other	Plane	Rabbit	Tool	Dude	Fgen	Fish	Hand	Kk	Other	Plane	Rabbit	Tool
Dude	11	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	0
Fgen	0	11	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0
Fish	0	0	11	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0
Hand	0	0	0	10	0	0	1	0	0	0	0	0	11	0	0	0	0	0
Kk	0	0	0	0	11	0	0	0	0	0	1	0	0	10	0	0	0	0
Other	0	0	0	1	0	9	1	0	0	0	0	0	2	0	9	0	0	0
Plane	0	0	0	0	0	0	11	0	0	0	0	0	2	0	0	9	0	0
Rabbit	0	1	0	0	0	0	0	10	0	0	0	0	0	1	0	10	0	0
Tool	0	0	0	0	1	0	0	0	10	0	0	0	0	1	0	0	0	10
Classes	$K_{\text{edit,10\%}}$									$K_{\text{classique,couverture}}$								
	Dude	Fgen	Fish	Hand	Kk	Other	Plane	Rabbit	Tool	Dude	Fgen	Fish	Hand	Kk	Other	Plane	Rabbit	Tool
Dude	11	0	0	0	0	0	0	0	0	9	0	0	0	0	2	0	0	0
Fgen	0	11	0	0	0	0	0	0	0	0	10	0	0	0	0	0	1	0
Fish	0	0	11	0	0	0	0	0	0	0	0	10	0	0	0	0	0	1
Hand	0	0	0	10	0	0	1	0	0	0	0	0	10	0	0	1	0	0
Kk	0	0	1	0	10	0	0	0	0	0	0	0	0	10	0	0	0	1
Other	0	0	0	1	0	10	0	0	0	0	0	0	1	0	10	0	0	0
Plane	0	0	0	2	0	0	9	0	0	0	0	0	0	0	11	0	0	
Rabbit	0	0	0	0	0	0	0	11	0	0	1	0	0	0	0	10	0	
Tool	0	0	0	0	1	0	0	0	10	0	0	0	0	0	0	0	0	11
Classes	K_{marches}									K_{Neuhaus}								
	Dude	Fgen	Fish	Hand	Kk	Other	Plane	Rabbit	Tool	Dude	Fgen	Fish	Hand	Kk	Other	Plane	Rabbit	Tool
Dude	10	0	0	0	0	1	0	0	0	11	0	0	0	0	0	0	0	0
Fgen	0	11	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0
Fish	0	0	8	1	0	1	1	0	0	0	1	10	0	0	0	0	0	0
Hand	1	0	0	9	0	0	0	1	0	0	0	0	11	0	0	0	0	0
Kk	0	1	0	0	10	0	0	0	0	0	0	0	0	10	1	0	0	0
Other	0	0	0	1	0	10	0	0	0	0	0	2	1	0	8	0	0	0
Plane	0	0	0	2	0	1	8	0	0	0	0	2	0	0	1	8	0	0
Rabbit	0	2	0	0	0	1	0	8	0	0	1	1	0	0	0	0	9	0
Tool	0	0	1	0	0	0	2	0	8	0	0	0	0	4	0	0	0	7

TAB. 6.4: Matrices de confusions avec la base de 99 formes.

posée d'animaux montrant leurs quatre pattes conduit à des graphes structurellement proche de ceux des classes Hand et Plane (Fig. 6.13). $K_{\text{edit},5\%}$ et $K_{\text{edit},10\%}$ ont des résultats très similaires et sont confus sur les classes Hand et Other, pour les mêmes raisons que précédemment. Aux vues de ces résultats, nous pouvons supposer que la couverture proposant un sac de chemins avec plus de détails conduit à une meilleure classification que la simple utilisation d'un pourcentage des chemins les plus pertinents.

$K_{\text{classique,couverture}}$ présente de bon résultats, avec une petite confusion sur les classes Dude et Other. Celle-ci peut venir de deux faits : premièrement, l'absence de processus d'édition rend le noyau vulnérable au moindre bruit et deuxièmement, la couverture apporte des petits chemins, correspondant à des détails, qui trouvent facilement un chemin assez semblable dans le sac de l'autre forme. Cette remarque explique aussi pourquoi, contrairement aux noyaux précédents, seulement deux classes sont parfaitement bien classées.

Les performances de classification de K_{marches} sont plus mauvaises que prévues par rapport aux résultats de résistances au bruit. Une confusion moyenne est présente sur les classes Fish, Plane, Rabbit et Tool. Les formes des classes Hand et Plane sont confondus dans les deux sens. Nous supposons que ces résultats sont induits par la méthode de comparaison des graphes et surtout l'effet de d'aller-retour décrit par Mahé et al. [2004]. En effet, ce phénomène conduit à des comparaisons erronées d'un point de vue structurelle au niveau des formes (les aller-retour remettent en cause le coté planaire du graphe codant une forme 2D).

Noyau	Temps (en s)
$K_{\text{edit,couverture}}$	9.46
$K_{\text{edit},5\%}$	5.2
$K_{\text{edit},10\%}$	11.7
$K_{\text{classique,couverture}}$	8.8
K_{marches}	18.3
K_{Neuhaus}	1.1

TAB. 6.5: Temps d'exécution pour le calcul de la matrice de Gram de la base d'entraînement.

Enfin, K_{Neuhaus} présente des résultats conformes à ceux donnés dans la première expérience. Les classes Other, Plane, Rabbit et Tool sont mal classées, surtout la classe Tool. Celle-ci est confondue avec la classe Kk qui présente des graphes très similaires structurellement. Les informations données par les attributs ne paraissent pas être assez discriminantes avec ce noyau, nous supposons que les raisons de ces résultats sont les mêmes que lors de la première expérience.

Ces résultats semblent montrer que la combinaison d'un processus d'édition et d'un sac de chemins suffisamment représentatif de la forme à classer conduit à de bonnes performances.

$K_{\text{edit},5\%}$ semble adapté pour la classification de formes dont les graphes sont différents structurellement et $K_{\text{edit},10\%}$ lorsque des chemins supplémentaires sont nécessaires. La couverture de graphe apportant une vision complète des formes paraît naturellement adaptée pour la classification de forme ainsi que semble montrer les matrices de confusions.

Afin d’avoir une idée du coût de calculs des différents noyaux comparés, le tableau 6.5 résume les temps d’exécution pour le calcul de la matrice de Gram de la base d’entraînement sur un quadri-processeurs (quadri-cœurs Opteron 3GHz) avec 16GB de RAM. Tous les algorithmes sont écrits en C++. Nous pouvons constater que nos noyaux sont plus rapides que K_{marches} , malgré une complexité temporelle plus élevée. Avec l’approximation de la distance d’édition proposée par Neuhaus et Bunke [2007a], K_{Neuhaus} est le plus rapide des noyaux testés.

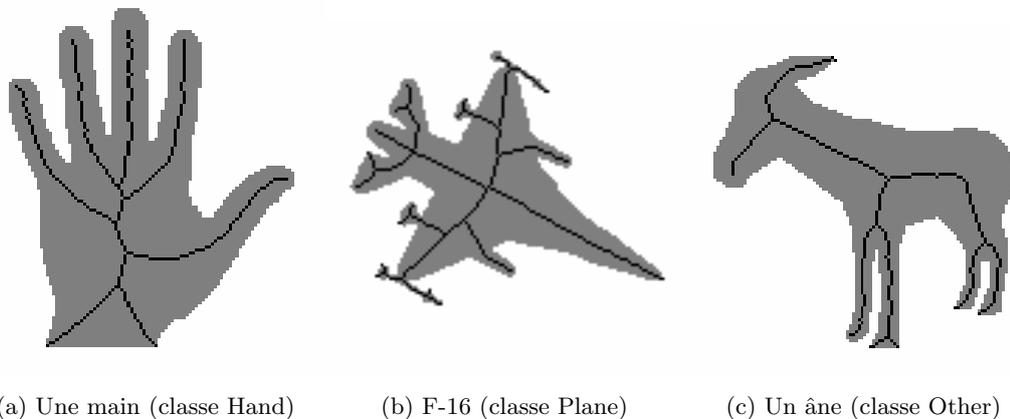


FIG. 6.13: Les graphes basés sur le squelette peuvent parfois être très proches structurellement, bien que les formes soient très différentes.

6.6.2.3 Classification de formes trouées

La table 6.6 montre les matrices de confusion obtenues avec $K_{\text{edit},\text{couverture}}$ avec puis sans K_{faces} et la table 6.7 celles avec $K_{\text{classique},\text{couverture}}$. Ces tests ont été réalisés sans permutations dans K_{faces} , car lors de la validation croisée elles n’ont pas montré d’interaction avec les résultats. Nous supposons qu’une recherche plus approfondie des paramètres optimaux donnerait des résultats différents.

$K_{\text{edit},\text{couverture}}$ combiné avec K_{faces} (les deux avec le processus d’édition) présente les meilleurs résultats (table 6.6(haut)). Les plus fortes confusions sont au niveau des classes Cup et Ring, qui sont confondues avec la classe Donut. Si nous regardons les formes (figure 6.11), nous pouvons constater que certaines sont très similaires. De plus, les comparaisons sont invariantes par translation, rotation et changement d’échelle, donc les donuts et les anneaux (Ring) montrent parfois les mêmes caractéristiques. Les différencier demande alors une autre méthode ou de

plus amples informations, la couleur par exemple. Notons que les autres confusions sont généralement sur des formes assez proches. Sans le noyau sur les faces (table 6.6(bas)), la confusion est accentuée sur certaines classes. Par exemple, Padlock est confondue avec Donut, Tool avec Padlock et Kk.

Sans processus d'édition, $K_{\text{classique,couverture}}$ avec K_{faces} (table 6.7(haut)) présente une confusion encore plus importante, essentiellement au niveau des classes Cup, Key, Other, Padlock, Ring et Scissor. Les formes trouées sont souvent confondues avec Donut, un mauvais réglage de $\gamma_{\text{combinaison}}$ pourrait expliquer ce phénomène, mais les résultats de $K_{\text{classique,couverture}}$ sans combinaison (table 6.7(bas)) montre aussi une forte confusion avec la classe Donut. Ainsi, nous pouvons supposer que sans éditions le noyau est incapable de comparer correctement les formes. Cela confirme l'importance du processus d'édition pour comparer les chemins. Néanmoins, sans la combinaison, $K_{\text{classique,couverture}}$ produit de meilleurs résultats. La confusion est en effet moins présente sur des classes comme Ring, Hand, Other et Scissor.

Ces résultats semblent confirmer les avantages du processus d'éditions pour la comparaison de chemins. De mêmes, la construction du sac de chemins avec la couverture donne suffisamment d'information pour différencier les formes lorsque c'est possible. Nous touchons ici le point faible de notre méthode. En effet, nous avons cherché à être invariant aux transformations usuelles des formes. Toutefois, nous sommes incapable de séparer des objets trop similaires sans un ajout d'information. De même, si la différence tient à un motif sur le contour, notre représentation est inefficace pour le mettre en avant. La combinaison avec K_{faces} permet de mettre un a priori sur les trous et conduit à une amélioration de la classification. Cependant deux formes similaires, l'une avec un trou et l'autre avec une cavité correspondant à une ouverture dans le trou, seront parfois considérer comme trop différentes, d'où l'importance de la variable d'équilibre.

6.7 Conclusion et perspectives

La construction de noyaux à partir de briques qui sont elles-mêmes des noyaux définis positifs fait à la fois le charme et la force de cette approche. Le domaine d'application semble infini tant il est facile de concevoir en agglomérant différents éléments un noyau adapté à une nouvelle tâche. Toutefois, les règles de construction sont peu adaptées pour des données structurées. Haussler [1999] avec ces noyaux de convolutions a introduit l'un des premiers cadres pour de telles données. Mais les différentes approches pour les graphes qui ont suivi [Kashima et Koyanagi, 2002; Kashima et al., 2003; Gärtner, 2002; Gärtner et al., 2003; Mahé et al., 2004] ne font pas vraiment apparaître la structure du graphe ou celle de l'élément représenté par celui-ci. Seuls sont pris en compte les motifs *linéaires* induit par les chemins. Mahé et Vert [2009] proposent bien d'utiliser des motifs d'arbres, mais ils n'effectuent aucune sélection et

	$K_{\text{edit,couverture}} + K_{\text{faces}}$														
Classes	Cup	Donut	Dude	Fgen	Fish	Hand	Key	Kk	Other	Padlock	Plane	Rabbit	Ring	Scissor	Tool
Cup	5	3	0	0	0	0	0	0	0	0	0	0	2	1	0
Donut	0	10	0	0	0	0	0	0	0	0	0	0	1	0	0
Dude	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0
Fgen	0	0	0	8	0	0	0	0	0	0	0	3	0	0	0
Fish	0	0	0	0	9	0	0	0	1	0	0	0	0	1	0
Hand	0	0	0	0	0	10	0	0	0	0	1	0	0	0	0
Key	0	0	0	0	0	0	8	0	0	1	0	0	1	1	0
Kk	1	0	0	0	0	0	0	10	0	0	0	0	0	0	0
Other	0	0	0	0	0	0	0	0	9	0	0	0	0	2	0
Padlock	0	3	0	0	0	0	0	0	0	8	0	0	0	0	0
Plane	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0
Rabbit	0	0	0	1	0	0	0	0	1	0	0	9	0	0	0
Ring	0	6	0	0	0	0	0	0	0	0	0	0	5	0	0
Scissor	0	0	0	0	0	1	1	0	0	0	0	0	1	8	0
Tool	0	0	0	0	0	0	0	2	0	1	0	0	0	0	8
	$K_{\text{edit,couverture}}$														
Classes	Cup	Donut	Dude	Fgen	Fish	Hand	Key	Kk	Other	Padlock	Plane	Rabbit	Ring	Scissor	Tool
Cup	5	3	0	0	0	0	0	0	0	1	0	0	1	1	0
Donut	0	10	0	0	0	0	0	0	0	0	0	0	1	0	0
Dude	0	0	9	0	0	0	0	0	0	0	0	0	1	1	0
Fgen	0	0	0	10	0	0	0	0	0	0	0	1	0	0	0
Fish	0	0	0	1	9	0	0	0	0	0	0	0	0	1	0
Hand	0	0	0	0	0	10	0	0	0	0	1	0	0	0	0
Key	0	0	0	0	0	0	7	0	0	3	0	0	0	1	0
Kk	1	0	0	0	0	0	0	10	0	0	0	0	0	0	0
Other	0	0	0	0	0	0	0	0	8	0	0	0	0	3	0
Padlock	0	4	0	0	0	0	0	0	0	7	0	0	0	0	0
Plane	0	0	0	0	0	0	0	0	0	0	10	0	0	1	0
Rabbit	0	0	0	1	0	0	0	0	2	0	0	8	0	0	0
Ring	0	5	0	0	0	0	0	0	0	0	0	0	6	0	0
Scissor	0	0	0	0	0	0	2	0	0	1	0	0	0	8	0
Tool	0	0	0	0	0	0	0	3	0	2	0	0	0	0	6

TAB. 6.6: Matrices de confusions avec la base de 165 formes avec $K_{\text{edit,couverture}}$ combiné avec un noyau sur faces (haut) et $K_{\text{edit,couverture}}$ seul (bas).

	$K_{\text{classique,couverture}} + K_{\text{faces}}$														
Classes	Cup	Donut	Dude	Fgen	Fish	Hand	Key	Kk	Other	Padlock	Plane	Rabbit	Ring	Scissor	Tool
Cup	5	4	0	0	0	0	0	0	0	1	0	0	1	0	0
Donut	0	9	0	0	0	0	0	0	0	0	0	1	1	0	0
Dude	0	0	9	0	0	0	0	0	0	0	0	0	2	0	0
Fgen	0	0	0	7	0	0	0	0	0	0	0	4	0	0	0
Fish	0	1	0	0	8	0	0	0	0	0	0	0	0	2	0
Hand	0	1	0	0	0	8	1	0	0	0	1	0	0	0	0
Key	0	1	0	0	0	0	6	0	0	0	0	0	2	2	0
Kk	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0
Other	0	0	0	0	0	0	1	0	6	0	0	1	2	1	0
Padlock	0	5	0	0	0	0	0	0	0	6	0	0	0	0	0
Plane	0	0	0	0	0	0	0	0	0	0	10	0	1	0	0
Rabbit	0	0	0	1	0	0	0	1	2	0	0	7	0	0	0
Ring	0	5	0	0	0	0	1	0	0	0	0	0	5	0	0
Scissor	0	3	0	0	0	0	1	0	0	1	0	0	0	6	0
Tool	0	0	0	0	0	0	0	1	0	1	0	0	0	0	9
	$K_{\text{classique,couverture}}$														
Cup	6	3	0	0	0	0	0	0	0	0	0	0	0	2	0
Donut	1	9	0	0	0	0	0	0	0	0	0	0	1	0	0
Dude	0	0	9	0	0	0	0	0	1	0	0	0	0	1	0
Fgen	1	0	0	8	0	0	1	0	0	0	0	1	0	0	0
Fish	0	0	0	1	9	0	0	0	0	0	0	0	1	0	0
Hand	0	0	0	0	0	11	0	0	0	0	0	0	0	0	0
Key	0	0	0	0	0	0	9	0	0	0	0	0	1	1	0
Kk	3	0	0	0	0	0	0	8	0	0	0	0	0	0	0
Other	0	0	0	0	0	1	0	0	9	0	0	0	1	0	0
Padlock	0	4	0	0	0	0	0	0	0	7	0	0	0	0	0
Plane	0	0	0	0	0	0	0	0	0	0	9	0	2	0	0
Rabbit	0	0	0	1	0	0	0	0	2	0	0	8	0	0	0
Ring	1	3	0	0	0	0	0	0	0	0	0	0	7	0	0
Scissor	0	1	0	0	0	0	1	0	0	1	0	0	0	8	0
Tool	0	5	0	0	0	0	0	0	0	0	0	0	0	0	6

TAB. 6.7: Matrices de confusions avec la base de 165 formes avec $K_{\text{classique,couverture}}$ combiné avec un noyau sur faces (haut) et $K_{\text{classique,couverture}}$ seul (bas).

leurs résultats sont équivalents à ceux obtenus avec un noyau sur les marches aléatoires.

Suard et al. [2006] avec ses noyaux sur sacs met en place un cadre intuitif pour ajouter une sélection de chemins, tout en conservant les mécanismes de comparaison des méthodes précédentes. Ce cadre permet de construire un noyau sur graphes en trois parties : un sac de chemins, un noyau sur sacs et un noyau sur chemins. Tous trois pouvant être définis indépendamment. Notre contribution porte sur chacune des trois parties. Premièrement, nous avons proposé deux techniques pour extraire un sac de chemins. Deuxièmement, trois nouveaux noyaux sur sacs sont introduits comme amélioration du noyau basé sur la moyenne des comparaisons. Enfin, nous insérons un processus d'édition au sein du noyau de comparaison de chemins afin d'obtenir une meilleure robustesse au bruit structurel.

Le sac de chemins peut être interprété comme un ensemble de vues linéaires d'un graphe. Partant du principe que pour comparer deux formes, il suffit d'avoir les éléments principaux de celles-ci, nous avons définis deux méthodes de construction de sacs basées sur une mesure de pertinence des chemins. La première consiste à prendre uniquement un pourcentage fixé de chemins en commençant par les plus importants. Toutefois, cette sélection induit une perte de données sur la forme qui peut conduire à de mauvaises comparaisons. Ainsi, nous introduisons une deuxième méthode, reposant sur une couverture contrainte de graphe, afin d'avoir un minimum de chemins importants sans perte d'informations sur la forme. Ces deux techniques sont basées uniquement sur la structure du graphe et une mesure de pertinence.

Un défaut des sacs de chemins est qu'ils ne mettent pas en avant les trous éventuels de la forme, or ceux-ci sont une caractéristique topologique importante de celle-ci. Pour cela, nous avons proposé un sac de trous pour construire un noyau sur les faces des graphes planaires. De plus, afin d'utiliser les mêmes briques, ce noyau est basé sur un noyau sur chemin, avec un processus de permutation permettant d'être robuste aux erreurs d'alignement.

Autre problème, les chemins simples n'apportent qu'une vue linéaire de la forme, les informations liées aux jonctions ne sont quasiment pas présentes. Cette information est souvent importante, car elle décrit l'agencement structurel. Une perspective est de s'inspirer des travaux de Mahé et Vert [2009], pour utiliser des motifs d'arbres ou de petits graphes pour comparer les graphes.

Une fois les sacs disponibles, nous devons les comparer. Habituellement, une moyenne est effectuée entre les différentes comparaisons de chemins. En effet cette méthode possède dans certains cas des implémentations efficaces [Vishwanathan et al., 2008]. Toutefois, nous avons souligné son manque de flexibilité, car elle ne profite pas de la distribution des données. Nos trois noyaux sur sacs tentent d'introduire d'autres points de vues pour comparer des ensembles, soit avec une pénalisation, soit avec l'ajout d'une estimation de la distribution des données.

L'insertion de la notion d'édition en sein de la construction de noyau défini positif n'est pas aisée. Notre proposition repose sur la connaissance d'une mesure d'importance des arêtes d'un

graphe. Celle-ci est alors exploitée pour extraire de chaque chemin une hiérarchie de réécritures, la comparaison est ensuite effectuée au travers de cette dernière. Les résultats montrent que ce processus apporte une flexibilité et une robustesse au bruit proche de celle d'une distance d'édition. Notons que ce processus peut s'étendre, sans trop de changements, à des structures autres que les chemins.

Cependant, cette technique a été développée pour les formes et nous supposons qu'elle pourrait s'appliquer à d'autres problématiques, par exemples, la classification d'images par l'intermédiaire des graphes de régions ou encore les molécules chimiques. Pour chacune, il nous faudra déterminer une technique de construction de sacs et adapter le processus d'édition. La principale difficulté sera alors de définir une mesure d'importance des sommets et des arêtes, si nous souhaitons conserver la méthode actuelle ; sinon la méthodologie sera à revoir afin d'intégrer la connaissance sur les sources de bruit.

Troisième partie

Détection automatique des épines
dendritiques

7 État de l'art sur l'analyse des épines dendritiques

Il n'y a point d'objet d'étude dont, chez l'homme libre, un comportement servile doive accompagner l'étude! [...] Dans l'âme, aucune étude forcée ne s'établit de façon permanente.

Platon, livre 7, *La république*

Les épines dendritiques jouent un rôle important dans les transmissions synaptiques. Elles font donc l'objet d'une importante recherche en biologie, dans le cadre de la compréhension du fonctionnement du cerveau. Les altérations de la neurotransmission étant au centre des processus dégénératifs à l'origine de certaines maladies neuro-dégénératives, elles font l'objet d'effort important de la part des chercheurs et des pouvoirs publics. Ces efforts sont aussi portés sur les outils d'analyse automatique des épines, qui permettent de compléter les observations qualitatives par des mesures quantitatives plus objectives.

Ce chapitre présente une introduction sur les épines dendritiques et leurs principales caractéristiques. Ensuite, nous posons les problématiques liées à l'analyse d'images de dendrites. Enfin, nous exposons la chaîne de traitements généralement utilisée pour l'exploitation d'images de dendrites avec des épines. Les différents maillons de cette chaîne sont détaillés en exposant les diverses méthodes proposées par l'état de l'art.

7.1 Les épines dendritiques

7.1.1 Historique et présentation

Découverte vers 1888 par Cajal, l'épine dendritique est une petite excroissance sur les dendrites (figure 7.1) sur lesquelles viennent se positionner des récepteurs aux neurotransmetteurs. Des recherches récentes [García-López et al., 2007] ont montré les liens importants entre la morphologie des épines et leur efficacité en terme de transfert de l'information neuronale. Les épines sont les sites où s'effectuent l'activation des récepteurs post synaptiques par les neurotransmetteurs relargués par les neurones présynaptique. La forme de ces structures va donc influencer sur la qualité de cette interaction.

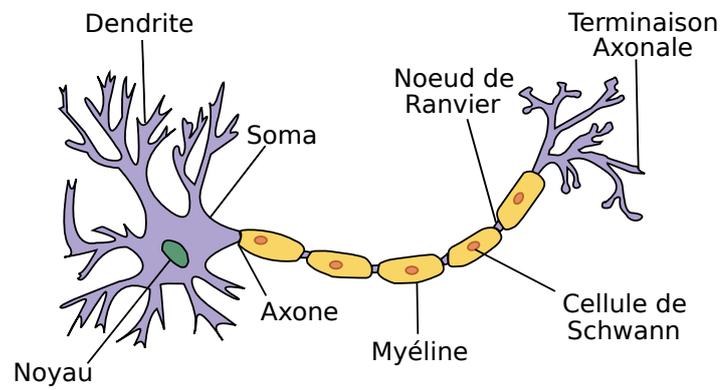


FIG. 7.1: Les épines sont des excroissances présentes sur les dendrites d'un neurone.

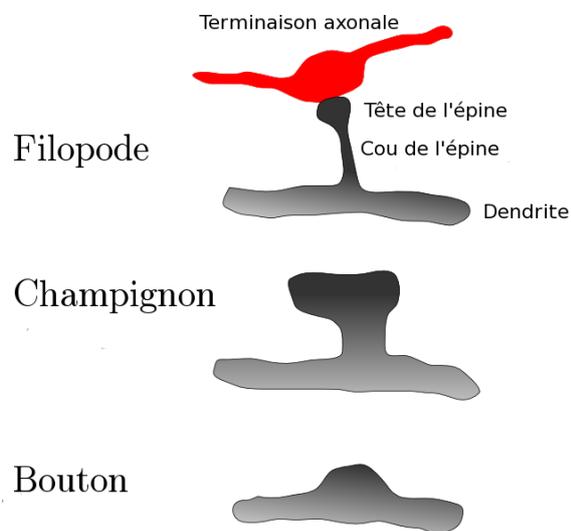


FIG. 7.2: Les épines sont généralement divisées en trois classes selon leur morphologie.

La morphologie d'une épine détermine la force, la stabilité des signaux générés par les neurotransmetteurs au niveau des synapses. Les anatomistes considèrent trois types d'épines différentes décrites par Peters et Kaiserman-Abramof [1969] selon la morphologie dominante (figure 7.2) :

l'épine en filopode qui est assez longue et fine et ne possède pas de tête,

l'épine en champignon qui possède un cou plutôt long et une tête, remarquons que le cou peut être très fin et assez court dans certain cas,

l'épine en bouton qui est courte mais large.

Cette morphologie varie dynamiquement en réponse de l'activité synaptique. Les petites épines sont moins stables et plus mobiles, donc plus plastiques que les épines larges. Cela implique que certaines épines ne sont pas incluses dans ces catégories. En effet, elles peuvent prendre des formes intermédiaires lors d'un changement.

7.1.2 Problématiques

Pour mieux comprendre les processus impliquant les épines, les biologistes cherchent à exploiter des observations microscopiques des dendrites. Les appareils d'acquisitions devenant de plus en plus précis, les images sont de plus en plus complexes à analyser. L'automatisation, d'une partie des traitements nécessaires à une analyse en profondeur, devient alors indispensable devant la quantité de données à exploiter.

Les images présentent deux principaux problèmes. Un premier avec l'acquisition, en effet les images sont dégradées à la fois par du bruit et du flou provenant du système optique. Le second problème provient de la nature même des observations : une seule dendrite peut contenir des centaines d'épines dont la morphologie évolue fortement au cours du temps. L'analyse automatique du nombre d'épines et de leurs évolutions constitue alors un problème difficile du fait de la variabilité des acquisitions.

L'observation des épines demande de pouvoir effectuer des mesures : longueur totale, longueur et largeur du cou lorsqu'il existe, longueur et largeur de la tête. Or, la précision de ces mesures est fonction de la résolution des images et de la qualité de l'acquisition. De plus, pour des raisons de rapidité de capture d'images, les volumes 3D présentent une anisotropie le long de l'axe des profondeurs (provenant du pas de descente). Ceci implique une perte d'information au niveau de l'épaisseur des échantillons.

7.2 Extraction automatique des épines

Nous avons souligné que la grande quantité d'informations à traiter demande l'automatisation de certaines étapes, pour permettre une pleine exploitation des images de dendrites. Avec le

développement du microscope confocal, les prémices de la restauration d'images de dendrites sont présentes au milieu des années 90 [Watzel et al., 1995, 1996], ainsi que les premières méthodes de détection des épines sur une dendrite [Rusakov et Stewart, 1995]. Koh et al. [2002] sont les premiers à proposer une chaîne de traitements complète dédiée à l'exploitation d'images de dendrites. Orientée 2D, cette chaîne a été reprise par divers acteurs qui ont modifié et amélioré les différentes étapes.

La figure 7.3 présente la chaîne de traitements. Celle-ci se décompose en quatre étapes successives indépendantes : restauration, segmentation, extraction et classification. La *restauration* permet d'obtenir une image plus exploitable, toutefois cette étape est parfois ignorée. Dans ce cas, les étapes suivantes sont définies pour travailler directement dans le bruit. L'étape de *segmentation* sépare la structure du reste de l'image. La complexité de cette étape dépend évidemment de la qualité de l'image d'entrée. Ensuite, l'*extraction* des épines est effectuée, cette étape est aussi celle des mesures sur les épines. Enfin une étape de *classification* est souvent présente pour catégoriser les épines et filtrer les éléments détectés. Les sous-sections suivantes développent chacune des étapes en suivant la chronologie de la chaîne de traitements.

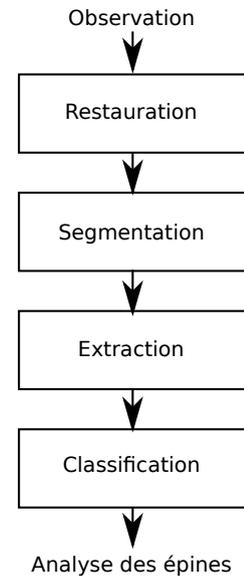


FIG. 7.3: La chaîne de traitements pour les observations d'épines dendritiques.

7.2.1 Restauration des images

Les images d'épines acquises avec un microscope confocal sont entachées de bruit. Une restauration est donc nécessaire pour obtenir une image exploitable. Cependant, la qualité de l'algorithme utilisé dépend souvent de l'étape suivante. Par exemple, Rink et Tönnies [2007] ou Zhang et al. [2007b] utilisent un filtre gaussien pour simplement réduire le bruit, car leur segmentation gère le bruit restant. Zhou et al. [2008] utilisent un filtre médian pour réduire le bruit de Poisson. Weaver et al. [2004]; Bai et al. [2007]; Janoos et al. [2009] préfèrent utiliser un logiciel propriétaire comme AutoDeblur¹. Koh et al. [2002] restaurent les images en appliquant un algorithme de déconvolution introduite par Kawata et Ichioka [1980].

Lorsque les images sont acquises par un microscope confocal à balayage laser, les deux sources de bruits principales sont le système optique qui introduit du flou et le photo-multiplicateur dégradant l'image avec un bruit de Poisson (chapitre 3). Des approches statiques peuvent être alors développées pour nettoyer les images (voir la première partie de la thèse). Dans

¹AutoDeblur est un logiciel de déconvolution produit par AutoQuant Image Inc.

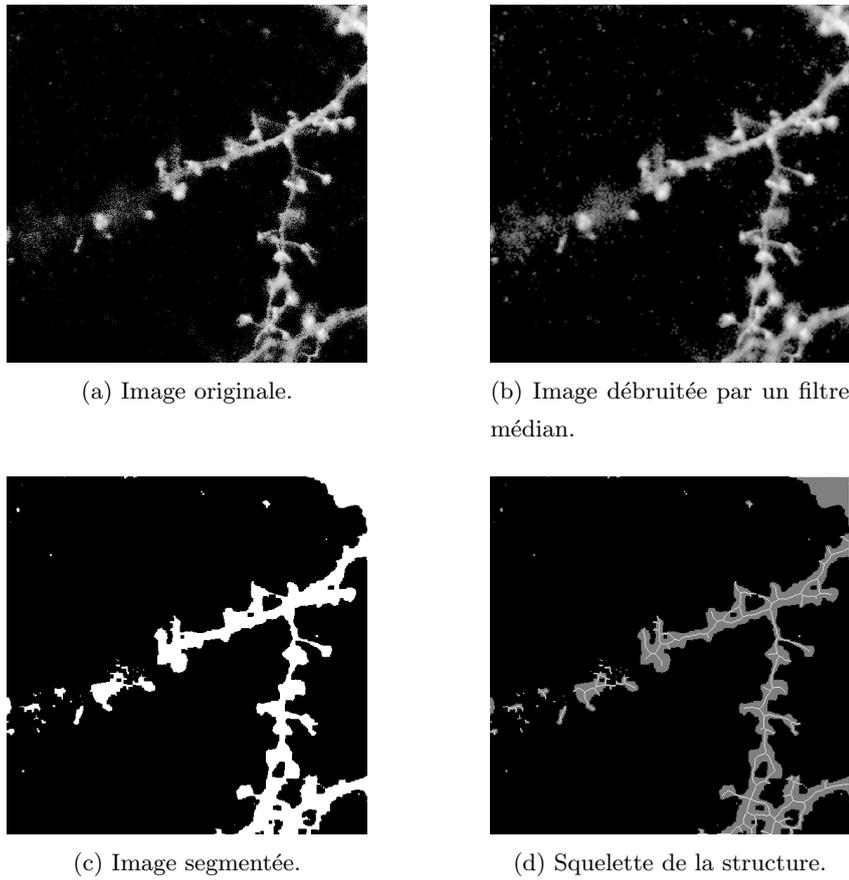


FIG. 7.4: Exemple de traitements d'une observation de dendrite. (a) L'image originale. (b) L'image débruitée en utilisant un filtre médian 3×3 . (c) L'image segmentée avec un seuillage simple. (d) Le squelette de la structure dendritique, les épines forment alors de petites branches reliées à un axe centrale représentant la dendrite.

le cas particulier de la restauration des épines dendritiques, Fudenberg et Paninski [2009] proposent un algorithme basé sur une méthode de Monte-Carlo pour obtenir une restauration sous contraintes dans le cas où l'image est observée avec un faible taux de comptage.

7.2.2 Segmentation

La segmentation permet d'extraire l'arborescence dendritique du reste de l'image. Les méthodes utilisées dépendent de la qualité des images données en entrée. Celles-ci présentent trois principales difficultés :

le bruit en effet les images ne sont pas toujours bien restaurées et peuvent présenter des artefacts. Dans d'autres cas, la segmentation est directement calculée à partir de l'observation bruitée.

le cou des épines peut être d'une épaisseur inférieure à la résolution du système d'acquisition, ce qui conduit l'observation de têtes déconnectées. Les échantillons présentent aussi des déchets cellulaires ressemblant à des têtes déconnectées. Koh et al. [2002] proposent de repérer les têtes seules en utilisant l'information présente sur l'axe dendritique, en supposant que l'épine considérée ait laissé une base (en forme de bouton) sur la dendrite.

le contraste est variable selon les zones observées et la concentration du fluorophore. La segmentation de la structure demande donc d'être robuste et adaptatif au changement de contraste au sein d'une même observation.

Face aux différentes difficultés, plusieurs méthodes ont été proposées. Elles sont séparables en deux catégories : celles basées sur un seuillage et celles basées sur des contours actifs.

Méthodes basées sur un seuillage : Koh et al. [2002] utilisent un simple seuillage, le seuil étant sélectionné en cherchant un équilibre entre suppression du bruit et conservation de la structure. Cependant, une telle méthode conduit souvent à une perte d'information dans les zones de faibles intensités. De plus, le seuil est déterminé manuellement par l'utilisateur pour chaque image. Bai et al. [2007] proposent de modéliser l'histogramme des images de dendrites par une hyperbole, partant de la remarque que plus l'intensité est forte moins il y a de pixels. En outre, l'essentiel de la forme se trouve du côté plat de l'histogramme et le fond est proche du pic. Le seuil est alors calculé en recherchant le début de la zone plate :

$$\text{seuil} = \min \left\{ i \mid - \frac{h(i) - h(i-1)}{h(i-1)} < \varepsilon \right\}, \quad (7.1)$$

avec ε un seuil choisi par l'utilisateur et h l'histogramme. Afin d'être moins sensible aux changements d'intensité, Cheng et al. [2007] proposent un seuillage local adaptatif : un pixel est conservé si son intensité est supérieure à l'intensité moyenne dans un voisinage défini autour du pixel. Afin de rendre la méthode plus robuste, les auteurs augmentent la valeur des pixels les

plus faibles pour éviter qu'ils soient directement considérés comme appartenant au fond. Cependant, les auteurs remarquent que les pixels de faible intensité proches d'un axe dendritique au contraste fort ne sont pas préservés. La conséquence de cet effet est le risque de déconnecter certaines épines. Ils proposent alors une étape supplémentaire après la segmentation : considérant comme dendrites toutes les structures trop grosses pour être des épines, toutes les intensités des pixels des dendrites sont remplacées par l'intensité minimale dans l'image observée. Ensuite, la segmentation est appliquée de nouveau sur cette nouvelle image. Une autre technique de seuillage adaptatif est proposée par Rodriguez et al. [2008]. Ils utilisent une version modifiée de la méthode ISODATA [Ridler et Calvard, 1978] qui calcule un seuil selon un voisinage local. Ils considèrent que la distribution des intensités des pixels présente un histogramme bimodal et trouvent le seuil qui sépare les deux modes de façon optimale.

Il existe également deux autres méthodes de segmentation basées sur un seuillage, mais à la différence des autres, celles-ci utilisent des outils soit pour restaurer l'image, soit pour extraire les structures présentant une forme particulière (ici les structures curvilignes). Ainsi, Janoos et al. [2009] améliorent le contraste des structures curvilignes et de leurs bordures avec un filtre non-linéaire de diffusion. La segmentation est alors effectuée simplement, par exemple, avec un seuillage général et un seuil fixe, car le contraste entre le fond de l'image et la structure est fort. Zhang et al. [2007a] utilisent un détecteur curviligne pour extraire l'axe dendritique. Ce détecteur est basé sur une succession de cinq étapes : 1) les auteurs commencent par estimer les directions des lignes en convoluant avec un noyau gaussien, puis 2) ils détectent les points formant les lignes médianes en calculant la matrice hessienne à chaque pixel, ensuite 3) les points médians sont liés pour former les lignes centrales, 4) la bordure des structures curvilignes détectées est extraite en combinant les informations sur le gradient et enfin 5) les différentes structures curvilignes sont reliées si besoin.

Méthodes basées sur des contours actifs : la segmentation avec un seuillage suppose que la structure puisse être séparée du fond uniquement avec les informations données par les intensités. Les contours actifs permettent de prendre en compte des informations a priori sur la structure et ainsi relier des éléments déconnectés. Rink et Tönnies [2006, 2007] proposent des algorithmes de contours actifs basés sur les ensembles de niveaux. Les auteurs ajoutent une énergie de *liaison* pour lier des éléments peu espacés. Cependant, cette méthode est sensible au bruit et donc demande d'avoir des images convenablement nettoyées. Zhou et al. [2008] proposent d'utiliser des laplaciens de gaussiens pour construire l'équation d'évolution des contours actifs, cette formulation améliore la robustesse au bruit et étend la zone de capture du contour.

La figure 7.4(c) montre un exemple de seuillage simple, les structures sont préservées mais il reste un peu de bruit. Pour une meilleure visualisation, le contraste des deux premières images a été rehaussé par une transformation logarithmique. Cependant, un seuillage simple peut se révéler inefficace pour la préservation des éléments faiblement lumineux, obligeant l'utilisation

de techniques plus complexes. Les contours actifs, par exemple, permettent de récupérer des liaisons perdues. Ainsi, les épines déconnectées peuvent être reliées à la dendrite.

7.2.3 Extraction des épines

Une fois l'arborescence dendritique repérée, il faut séparer les épines de l'axe principal de la dendrite. La plupart des auteurs font pour cela appel à une squelettisation, suivie d'une extraction des lignes centrales. Cette méthode a l'avantage d'être définie aussi bien en 2D qu'en 3D. Les épines sont alors repérées en recherchant les petites branches significatives reliées à l'axe principale (figure 7.4(d)). Ensuite, un ébarbage permet d'avoir rapidement une estimation de l'axe dendritique. Les éléments aberrants sont supprimés en vérifiant leurs conformités à un ensemble de critères.

Cette technique permet de repérer facilement les épines connectées à l'axe dendritique. Cependant, le cou des épines est parfois plus fin que la résolution maximale du microscope, il devient alors invisible et seul la tête apparaît dans l'image. La gestion de ces têtes flottantes est délicate, car elles peuvent être confondues avec des débris cellulaires et inversement. Par exemple, Koh et al. [2002] utilisent plusieurs critères pour les repérer comme la distance de la tête à l'axe.

7.2.4 Catégorisation des épines

Les épines sont généralement classées en trois types (figure 7.2). Pour cela, Koh et al. [2002]; Weaver et al. [2004]; Cheng et al. [2007]; Bai et al. [2007]; Rodriguez et al. [2008]; Janoos et al. [2009] utilisent des mesures et des critères simples (présence ou non d'un cou, etc) sur les épines pour les différencier. Par exemple, Koh et al. [2002] catégorisent les épines selon la longueur, le diamètre du cou et celui de la tête. Le tableau 7.1 présente les catégories selon les résultats de deux ratios, celui entre la longueur et le diamètre de la tête et celui entre le diamètre de la tête et celui du cou.

Autre exemple, Rodriguez et al. [2008] proposent un schéma de classification (figure 7.5) reposant sur quatre questions. La première question est de savoir si l'objet en question est bien une épine. Les auteurs y répondent avec une série de critères (longueur, profil 3D, etc). La deuxième question concerne l'existence d'un cou dans l'épine. La troisième question demande si le ratio entre longueur et largeur de l'épine est grand. Enfin, la quatrième question s'intéresse à la largeur de la tête. Les deux dernières questions requièrent des seuils pour répondre. Ces seuils sont réglés empiriquement selon les données.

A la différence des autres auteurs, Zhang et al. [2007b] utilisent un classifieur basé sur une analyse linéaire discriminante pour détecter les épines *proches* d'un ensemble d'épines donné en entrée. Pour cela, des imagerie d'épines sont transformées en vecteurs d'attributs contenant

L/d_t	d_t/d_c		
	$[0, 1.3)$	$[1.3, 3)$	$[3, +\infty)$
$[0, 2/3)$	Bouton	Champignon	Champignon
$[2/3, 2)$	Bouton	Bouton	Bouton
$[2, 3)$	Bouton	Champignon	Champignon
$[3, 5)$	Filopode	Champignon	Champignon
$[5, +\infty)$	Filopode	Filopode	Filopode

TAB. 7.1: Critère de catégorisation des épines selon les ratios entre la longueur L , le diamètre du cou d_c et celui de la tête d_t .

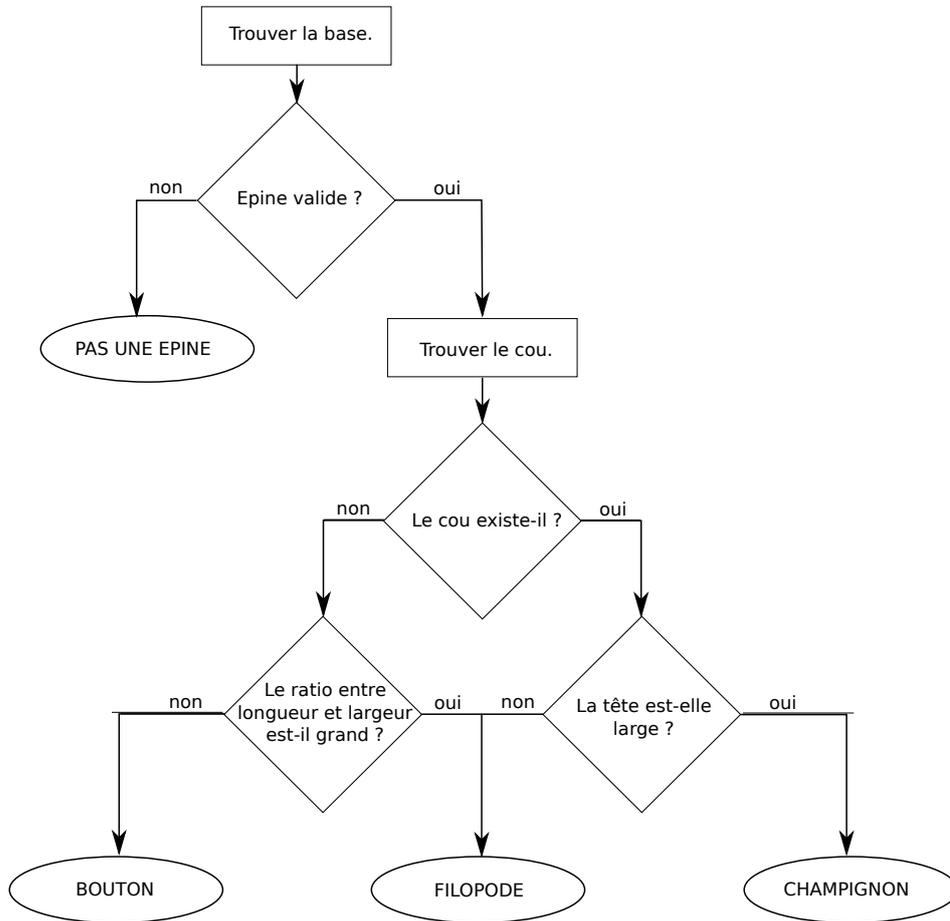


FIG. 7.5: Exemple de schéma de classification pour les épines. La validation des épines est effectuée par un ensemble de critères biologiques à respecter.

12 caractéristiques géométriques (la longueur par exemple) et 36 moments de Zernike. Les attributs corrélés sont ensuite supprimés pour plus d'efficacité (selon les auteurs, il reste environ 22 attributs). La classification est effectuée avec tout ou partie des attributs. Selon les auteurs, cette méthode n'est toutefois pas adaptée pour catégoriser les épines en fonction des trois classes principales (figure 7.2).

7.2.5 Mesures

La mesure principale est souvent la longueur des épines prise à partir du morceau de squelette les représentant. Koh et al. [2002] déterminent le volume des épines à partir des intensités des pixels :

$$\text{volume} = \frac{\text{intensité maximale de l'épine}}{\text{intensité maximale de la dendrite}} \times \text{volume focal} . \quad (7.2)$$

Rodriguez et al. [2008] utilisent un profil en plusieurs couches pour chaque épine (avec leur technologie Rayburst) et obtiennent ainsi la longueur et largeur du cou si il existe avec la longueur de l'épine et les informations sur la tête. Janoos et al. [2009] utilisent directement le nombre de voxels pour obtenir le volume. Koh et al. [2002]; Bai et al. [2007] proposent un mécanisme particulier pour gérer les épines déconnectées : 1) ils détectent les bases sur les dendrites, 2) regardent l'angle formé entre la direction principale de la base et le centre de la tête et 3) calculent la distance entre la base et la tête qui doit être en-dessous d'un seuil fixé pour que l'épine soit valide.

7.3 Conclusion

Le processus d'exploitation des observations d'épines dendritiques se traduit par une chaîne de traitements. Les différentes étapes impliquées font souvent l'objet de recherches indépendantes pour des problématiques similaires. Seules les dernières étapes sont vraiment spécifiques à l'analyse d'épines et donc dépendantes des structures présentes. Les premières étapes de restauration et de segmentation sont parfois combinées en une seule d'extraction du réseau dendritique dans le bruit. Cependant, dans le chapitre 3, nous avons développé la formation de l'image par un microscope confocal (3.17). La dégradation est très importante, le flou enlève certains éléments, comme le cou des épines et le bruit rend l'analyse complexe. Il nous semble donc qu'une étape de déconvolution est essentielle pour une exploitation du réseau dendritique.

Une fois la forme séparée du fond, l'état de l'art propose une seule technique pour repérer les épines : une squelettisation suivie d'une suppression des éléments trop petits. Les mesures pouvant ensuite être effectuées sur les éléments détectés. Les méthodes de mesures sont souvent peu développées, se contentant de la longueur. Toutefois, l'obtention de renseignement sur le

cou ou la tête demande des algorithmes robustes. Par exemple, la mesure du cou requière de séparer la tête du reste de l'épine, or ces deux parties sont parfois très proches.

Seulement deux méthodologies sont proposées pour l'étape finale de classification : soit un ensemble de critères est construit, soit un classifieur est utilisé sur des données représentant les épines. La première méthode est la plus simple à mettre en œuvre et la plus rapide, mais elle demande d'ajuster des paramètres qui peuvent varier selon la provenance des données. La deuxième permet d'utiliser plus d'attributs avec des modèles de données complexes. Bien que moins rapide, cette technique est plus flexible et n'est pas limitée à la classification en 3 catégories.

8 Détection et analyse des épines

Mais si l'ambition ultime de la science entière est bien, comme je le crois, d'élucider la relation de l'homme à l'univers, alors il faut reconnaître à la biologie une place centrale puisqu'elle est, de toutes les disciplines, celle qui tente d'aller le plus directement au cœur des problèmes qu'il faut avoir résolus avant de pouvoir seulement poser celui de la « nature humaine » en termes autres que métaphysiques.

Jacques Monod, *Le hasard et la nécessité*

Dans le chapitre précédent, nous avons présenté la chaîne de traitements et d'analyse des images de dendrites contenant des épines (figure 7.3) initialement proposée par Koh et al. [2002]. Cette chaîne a ensuite été reprise par beaucoup d'acteurs qui proposent des améliorations sur une ou plusieurs étapes. La qualité des méthodes impliquées dans chaque étape influence la qualité des étapes suivantes.

Ainsi, nous proposons premièrement d'utiliser les algorithmes présentés dans le chapitre 4 pour restaurer les images issues d'un microscope confocal. De plus, le concept de diversité morphologie permet de prendre en compte la structure fortement curviligne des réseaux dendritiques. La qualité de cette étape autorise l'utilisation de technique de segmentation simple dans un premier temps. Puis, nous utilisons la représentation par graphe introduite dans le chapitre 6 pour le réseau dendritique, puis pour les épines. Celles-ci sont extraites en recherchant les petits éléments connectés à l'axe centrale de la dendrite. Ensuite, les graphes permettent d'utiliser les méthodes à noyaux avec ceux du chapitre 6 pour classer les épines. Cette approche ouvre même la porte vers d'autres outils de fouilles de données. Enfin, les observations 3D issues d'un microscope confocal montrent souvent une forte anisotropie le long de l'axe des profondeurs. Nous proposons un ensemble de techniques pour effectuer des mesures en faisant attention au manque de données.

8.1 Restauration

L'exploitation des images correspond à l'extraction d'un maximum d'information de celles-ci. Travailler directement avec l'image brute impose d'avoir des méthodes à la fois robustes au bruit et sensibles aux détails. Cet équilibre est difficile à obtenir, car le bruit est surtout présent

sur les détails (figure 8.1). Donc, la restauration des images ouvre la porte aux méthodes moins robustes mais plus sensibles dans le reste de la chaîne de traitements de la figure 7.3.

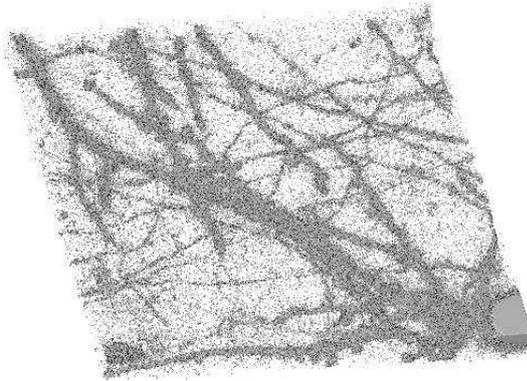


FIG. 8.1: Exemple d'image de dendrites bruitées.

Les images de dendrites présentent de fortes caractéristiques morphologiques : les dendrites sont curvilignes et les épines sont des éléments soit curvilignes, soit quasi isotropes dans le cas des têtes détachées. Nous proposons alors de construire le dictionnaire Φ avec des ondelettes pour capturer les structures isotropes et des curvelets pour les structures curvilignes (voir chapitre 4 pour plus d'information). La PSF est construite en utilisant les méthodes d'approximations proposées par Zhang et al. [2007a] et Booth et Wilson [2001] (sections 3.2.1.1 et 3.2.1.2) selon la précision recherchée. Notons que ces deux méthodes requièrent uniquement les paramètres d'acquisitions pour un modèle. Ensuite, nous appliquons l'algorithme 4.1 pour déconvoluer les images.

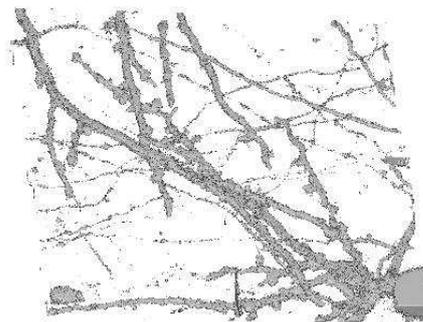


FIG. 8.2: Exemple d'image de dendrites restaurée avec l'algorithme 4.1 utilisant un dictionnaire contenant des ondelettes décimées.

La déconvolution peut être utilisée selon plusieurs modes. Si l'image 3D possède suffisamment d'information en profondeur, une déconvolution 3D permet d'obtenir de belles restaurations de l'image [Pawley, 2005]. Sinon, une déconvolution 2D est effectuée plan par plan, ignorant alors l'information de profondeur.

La figure 8.1 présente une image 3D d'un réseau dendritique (afin de visualiser l'image, celle-ci a été seuillée en retirant les valeurs inférieures à 3). Nous pouvons constater la présence de bruit un peu partout dans l'image. Nous avons restauré cette image en appliquant l'algorithme 4.1 avec un dictionnaire contenant uniquement les ondelettes décimées. La taille de chaque voxel est de $0.1\mu m \times 0.1\mu m \times 0.3\mu m$, il présente donc une forte anisotropie en Z (les images présentées par les figures 8.1 et 8.2 supposent que les voxels sont isotropes). Nous avons donc appliqué la déconvolution 2D sur chaque plan de l'image. La figure 8.2 montre le résultat de la restauration. Un grand nombre de structures sont préservées et plus fines et le bruit est maintenant moins présent dans l'image. Il reste toutefois un léger bruit le long des structures curvilignes.

8.2 Segmentation

Une fois l'image restaurée, nous devons séparer la structure du fond de l'image. Nous supposons que la restauration a correctement reconstruit les structures curvilignes, donc la bordure de ces structures est clairement séparable du fond. Nous proposons ainsi deux méthodes. Si l'image est correctement restaurée, un seuillage général devrait suffire pour séparer la forme du fond (le seuil pouvant être déduit de l'histogramme). Sinon, un seuillage adaptatif permet préserver les structures d'intensités faibles.

Après segmentation, l'image présente un grand nombre de petites structures dont les origines sont assez diverses : artefacts de restauration, déchets cellulaires, têtes d'épines détachées, etc. Nous proposons de supprimer les plus petits éléments en appliquant successivement deux opérateurs de morphologie mathématique : une érosion (réglée selon les éléments à supprimer) suivie d'une dilatation où les structures possédant au moins un élément présent après l'érosion sont complètement restaurées.

La figure 8.3 montre un résultat de segmentation après suppression des petites structures avec des opérations de morphologies mathématiques. Toutes les structures de faibles intensités ont disparu. La structure est préservée dans certaines zones présentant des épines. Ces zones correspondent souvent à celles recherchées par les biologistes.

8.3 Extraction des épines

L'extraction des épines est l'étape préliminaire pour le calcul d'information sur ces épines. Suivant l'état de l'art, nous utilisons le squelette pour repérer les éléments pouvant être des

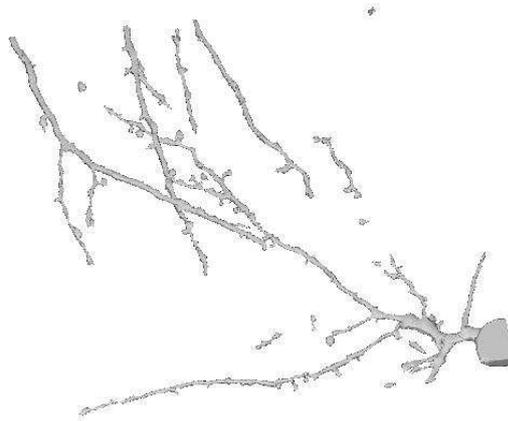


FIG. 8.3: Exemple de structures dendritiques après segmentation avec un seuillage général.

épines. En effet, le squelette permettant de représenter l'arborescence avec une structure filaire, les épines sont soit de petites branches reliées à l'axe dendritique, soit des éléments isolés dans le cas des têtes détachées.

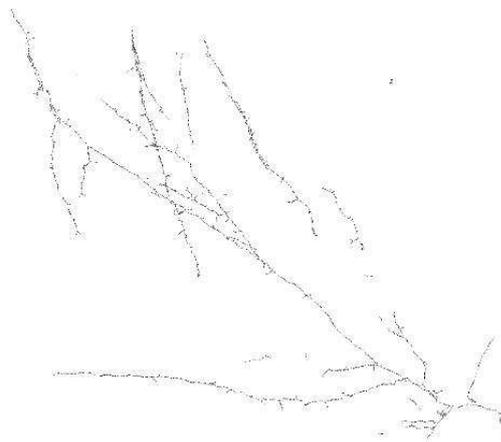


FIG. 8.4: Exemple de squelette. Les épines forment de petites branches reliées à une ligne centrale.

L'algorithme de squelettisation choisi repose sur celui proposée par Siddiqi et al. [2002] et étendu aux squelettes filiformes par Bouix et al. [2005]. Cette méthode propose un contrôle sur la sensibilité aux bruits qui peut être important selon la qualité de la segmentation. Le squelette est homotope à la forme, ainsi la moindre cavité dans la structure est conservée. Or les cavités sont souvent des zones mal segmentées. Pour boucher les trous, nous utilisons un algorithme [Aktouf et al., 2002] dérivé de la squelettisation, afin d'obtenir une structure filiforme pour représenter la structure. L'utilisation de cet algorithme peut se concevoir comme l'utilisation d'informations a priori (anatomiques) sur la structure des dendrites pour améliorer

	Sommet	Arête
plongement géométrique	terminaison, jonction et/ou inflexion du squelette	branche, partie de forme
descripteurs globaux	distance au centre de gravité	surface normalisée de la bordure
descripteurs locaux		évolution du rayon de la sphère maximale

TAB. 8.1: Résumé des descripteurs de la forme associée au graphe dans le cas de la 3D tubulaire.

la qualité de la squelettisation.

Cette structure est ensuite codée par un graphe en étendant à la 3D la construction du graphe présentée dans la section 6.1 (la structure étant linéaire la méthode est la même). Maintenant dans le domaine 3D, la branche suit le centre de la sphère de rayon maximale ; le squelette propose donc une représentation tubulaire et sphérique de l'arborescence dendritique. Les différents attributs décrits dans la section 6.1.2 possèdent une équivalence en 3D et sont résumés dans la table 8.1.

Une fois le squelette obtenu, nous extrayons le ou les axes dendritiques, c'est à dire les lignes centrales. Le principe réside dans l'observation suivante : les éléments connectés à ces lignes sont potentiellement des épines. Nous proposons de repérer les éléments en supprimant les petites branches, pour obtenir les lignes (algorithme 8.1). Afin de supprimer les éléments aberrants, nous considérons comme épines uniquement ceux répondant à des critères sur le volume, la longueur et la surface.

Algorithme 8.1 : Calcul des lignes centrales.

Entrées : Un squelette S et une taille maximale τ .

Début

```

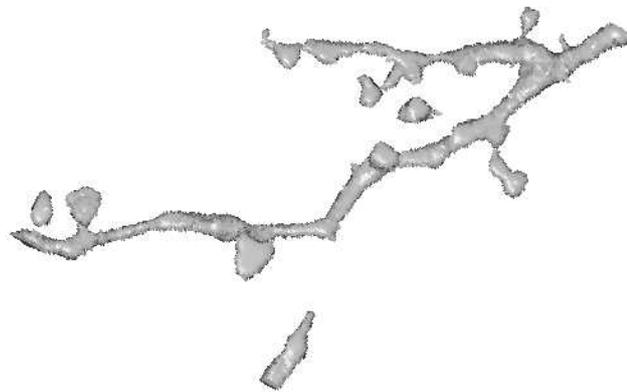
Tant que  $S$  possède une branche  $b$  de taille  $< \tau$  avec un point terminal faire
  |  $S = S \setminus \{b\}$ 
fintq

```

Fin

Sorties : L'ensemble des lignes centrales S .

La figure 8.4 présente un exemple de squelette filiforme. Nous pouvons observer les petites branches, présentes un peu partout sur la structure, qui représentent des excroissances. Un exemple d'extraction des épines présentés dans 8.5(a) est illustré par la figure 8.5(b). Les épines sont toutes détectées, cependant, un morceau de la dendrite est aussi pris. Un réglage des paramètres des filtres est suffisant pour enlever le peu d'éléments aberrant dans cet exemple.



(a) La structure



(b) Un exemple de segmentation

FIG. 8.5: Exemple de segmentation d'une zone de l'arborescence dendritique présentée dans la figure 8.3. Tous les éléments d'une couleur différente de celle de la dendrite sont considérés comme des épines.

Notons qu'ici, la zone est petite donc nous sommes peu gênés par la diversité morphologique des épines au moments de l'extraction.

8.4 Classification des épines

Les épines sont principalement classées en trois catégories (figure 7.1(b)). Ces catégories définies par les anatomistes sont basées sur des critères de tailles. Koh et al. [2002] proposent d'utiliser des ratios pour classer. Afin d'utiliser plus de critères et d'utiliser les mesures 3D, Rodriguez et al. [2008] utilisent un schéma plus souple dont certains éléments peuvent être adaptés selon les besoins. A l'inverse de ces deux méthodes, Zhang et al. [2007b] utilisent une analyse linéaire discriminante sur des vecteurs représentant les épines. Toutefois, cette méthode permet uniquement d'obtenir les épines les plus proches d'un ensemble.

Chaque épine étant représentée par un graphe, nous proposons d'utiliser les noyaux sur graphes pour classer les épines. Leur cadre théorique apporte un grand nombre d'algorithmes : ACP, SVM, distance de Mahalanobis, etc. Tous travaillent directement dans l'espace engendré par les noyaux. Dans le chapitre 6, nous avons présenté plusieurs noyaux sur graphes adaptés à la comparaison de formes. Ainsi, les épines peuvent être classées selon leur morphologie et les caractéristiques des sous-parties qui les forment. Ici réside la différence fondamentale avec l'approche de Zhang et al. [2007b]. En effet, nous préservons au maximum le lien entre la représentation et la forme. La table 8.1 résume les attributs des graphes, certains d'eux sont déjà utilisés (par exemple le surface et le volume) pour filtrer les épines. Autre intérêt de cette approche, nous n'avons pas besoin de repérer le cou ou la tête pour avoir une classification plus précise, car ces éléments apparaissent dans le graphe au niveau de la structure et des points d'inflexions.

Ainsi, un large panel d'applications sont possibles au delà la classification des épines parmi les trois principales catégories. Par exemple, comme Zhang et al. [2007b], avec un ensemble d'épines en entrée, nous pouvons déterminer les épines les plus proches morphologiquement de cet ensemble. De la fouille de données peut aussi être envisagée. Par exemple, une ACP permet de trouver le point commun entre un ensemble d'épines, à condition que l'espace engendré par le noyau soit interprétable. Ceci dépend des attributs comparés et de la construction des sacs de chemins.

8.5 Mesures biologiques sur les épines

Une fois les épines correctement identifiées, nous pouvons procéder à des mesures. Les biologistes sont intéressés par les informations suivantes : la longueur totale, la largeur de la tête, la longueur et la largeur du cou (si présent), le volume et la surface.

Le volume et la surface peuvent être directement calculés à partir de l'imagette 3D de l'épine. Toutefois, si l'anisotropie des voxels est trop forte, une erreur non-négligeable est à prévoir sur les mesures. Pour éviter ces problèmes, une première méthode consiste à rééchantillonner l'image pour obtenir des voxels isotropes. Une autre méthode, plus lourde, est de reconstruire l'image en calculant les données manquantes à partir de l'existant.

La longueur totale et la largeur de la tête (ou de l'épine le cas échéant) peuvent directement obtenue en projetant la forme sur la première composante principale pour la longueur et la deuxième composante principale pour la largeur (figure 8.6(a)). Si l'image présente une forte anisotropie le long de l'axe des profondeurs, nous proposons de faire les calculs sur une version aplatie (projection sur un plan 2D) de l'image.

Pour le cou, la longueur du cou est déduite à partir d'un profil 1D (figure 8.6(b)) de l'épine le long de la composante principale. Comme profil, nous proposons d'utiliser l'histogramme du nombre de voxels (par projection euclidienne) le long de ce vecteur. Si l'épine possède un cou, alors le profil devrait montrer un plat avec moins de voxels que la tête. La longueur de ce plat est celle du cou. La largeur du cou est obtenue comme étant la moyenne des largeurs le long du cou. Nous proposons d'estimer la longueur du cou en appliquant une régression multi-polynomiales avec des polynômes de degré 1. Le cou est alors obtenu en recherchant les polynômes de faibles pentes compris entre deux polynômes de fortes pentes.

La figure 8.7 présente un exemple de l'ensemble des mesures que nous pouvons effectuées sur une épine. La précision est ici limitée par la résolution de l'image. Nous pouvons constaté que le cou est petit et la tête large, ainsi l'épine est classée comme bouton (*stubby*) par la méthode proposée par Rodriguez et al. [2008]. Celle de Koh et al. [2002] semble mal prendre en compte les paramètres du cou.

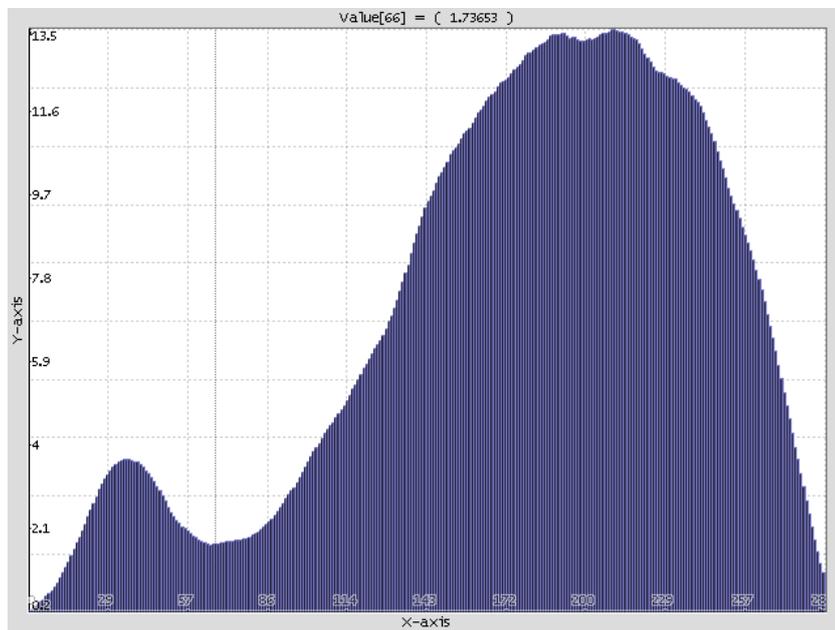
8.6 Conclusion

Nos contributions au sein de la chaîne de traitements de Koh et al. [2002] sont essentiellement en deux points. Premièrement, nous proposons un algorithme de déconvolution capable d'utiliser la diversité morphologique des images. De plus, celui-ci est adapté pour les acquisitions produites avec un microscope confocal. Les résultats montrent que cette étape est importante pour la suite, car une bonne restauration permet d'utiliser des méthodes plus précises et moins robustes au bruit. Pawley [2005] soulignent fortement l'intérêt de la déconvolution pour l'analyse en décrivant un grand nombre d'avantages dont la restauration des petites structures.

La classification des épines par des machines à noyaux forme la deuxième contribution. Les épines sont représentées par un graphe extrait à partir du squelette dont les attributs donnent des informations sur la morphologie. De plus, la structure du graphe décompose chaque épine en plusieurs parties. Les noyaux sur graphes offrent donc une comparaison proche de la forme



(a) Les deux axes principaux d'une épine



(b) Le profil 1D le long du plus grand axe

FIG. 8.6: Les deux axes principaux et le profil 1D d'une épine. Les deux axes servent de références pour le calcul de la largeur et la longueur de l'épine. Le profil 1D montre le nombre de voxels suivant l'axe de la longueur en partant de la dendrite. La première bosse est donc un morceau de la base appartenant à la dendrite, puis nous avons le cou et enfin la tête.

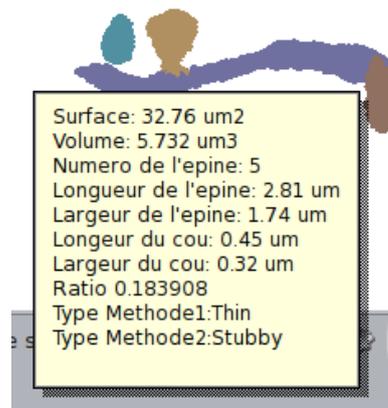


FIG. 8.7: Exemple de mesures effectuées sur l'épine de couleur marron. Pour la classification, les deux méthodes proposées ici sont celles de Koh et al. [2002] (méthode 1) et celle de Rodriguez et al. [2008] (méthode 2).

et donc permettent une catégorisation, non plus basées sur un ensemble de critères mais sur la forme elle-même. La diversité des machines à noyaux ouvre un important éventail de possibilités d'utilisation des noyaux sur graphes pour classer des épines et ainsi nous évitons les limitations de la méthode proposée par Zhang et al. [2007b].

Nous avons présenté un ensemble de techniques pour effectuer des mesures sur les épines tout en considérant l'anisotropie. Celle-ci peut être enlevée avec un algorithme de reconstruction des données manquantes. Par exemple, Schmitt et al. [2004] combinent squelette et contours actifs pour obtenir une reconstruction assez lisse du réseau dendritique. Evidemment, un tel outil conduit à la fois à une meilleure exploitation des données et donc à des mesures plus précises. En perspectives, l'utilisation du concept de diversité morphologique introduit dans le chapitre 4 semble un point de départ pour reconstruire les données manquantes. Un tel cadre a déjà été utilisé en 2D par Fadili et al. [2006], pour de l'*inpainting* et de l'agrandissement d'image.

Conclusion générale

Tout à l'air si compliqué. Tout est si simple pourtant. Si j'avais eu la lune, si l'amour suffisait, tout serait changé. Mais où étancher cette soif? Quel cœur, quel dieu auraient pour moi la profondeur d'un lac? Rien dans ce monde, ni dans l'autre, qui soit à ma mesure. Je sais pourtant, et tu le sais aussi, qu'il suffirait que l'impossible soit. L'impossible! Je l'ai cherché aux limites du monde, aux confins de moi-même.

Albert Camus, *Caligula*, scène XIV, acte IV

Cette thèse s'inscrit dans la problématique de l'analyse des épines dendritiques, sachant que ces éléments sont observés par fluorescence avec un microscope confocal. Au départ, nous avons identifié les deux principales sources de difficultés : la dégradation des observations et la morphologie très changeante des épines. Ainsi, la thèse a été divisée en trois parties. D'abord, nous avons cherché à estimer les images réelles à partir des acquisitions. Ensuite, un cadre pour la reconnaissance de formes prenant en compte la variabilité de celles-ci a été développé. Enfin, les contributions des deux parties précédentes ont été utilisées pour former une chaîne de traitements dédiée à l'analyse des épines. A travers de cette thèse, nous avons donc proposé une série de contributions dans trois domaines spécifiques : la restauration d'images, la reconnaissance de formes et l'analyse des épines dendritiques.

Notre première contribution consiste en un cadre général de résolution de problèmes d'optimisation impliquant des fonctions convexes et semi-continues inférieurement. Le principe de ce cadre est d'utiliser au maximum la décomposition de la fonction objectif en des fonctions *simples*. Nous supposons qu'elles possèdent un gradient Lipschitz continue ou un opérateur proximal exprimable analytiquement. Ainsi, plusieurs schémas de résolution sont présentés selon le type de décomposition. Un tel cadre permet de résoudre un large panel de problèmes, dont celui de la déconvolution sous bruit de Poisson. En effet, en étudiant le microscope confocal avec la fluorescence, la formation de l'image est approximable par une dégradation du système optique (flou) suivie de celle du photo-multiplicateur (bruit de Poisson). Nous exprimons l'a priori sur les structures présentes dans l'image avec le concept de diversité morphologique. Celui-ci repose sur des transformées issues de l'analyse harmonique comme la transformée en ondelettes. Ainsi, avec la notion de parcimonie, nous supposons que les structures sont représentables par peu de coefficients dans le domaine des transformées.

Nous avons proposé une série d'algorithmes de déconvolution, selon que le bruit soit stabilisé ou non. Ils exploitent un cadre théorique issu de l'analyse convexe et le concept de diversité morphologique. Toutefois, les algorithmes que nous avons proposés peuvent parfois s'avérer lents et leurs vitesses de convergence n'ont pas toujours été exhibées. De même, concernant la diversité morphologique, la construction du dictionnaire optimal pour une catégorie d'images n'est pas toujours immédiate. Ces deux problèmes forment deux futurs axes de recherche. De même que l'estimation des hyperparamètres impliqués, comme celui de la régularisation, qui, même avec la solution que nous avons proposée pour son choix, reste un point crucial en problèmes inverses régularisés.

Une fois les images restaurées, nous nous sommes intéressés aux éléments que nous avons à manipuler. Les épines dendritiques sont connues pour avoir trois principales morphologies : filopode, bouton et champignon. Toutefois, elles possèdent aussi une grande plasticité et peuvent présenter une forme intermédiaire aux trois principales. Ainsi, nous avons cherché à développer un cadre de reconnaissance des formes puisant à la fois dans les outils de classification classiques et dans les méthodes de comparaison de formes capables d'exploiter les caractéristiques morphologiques des épines. Le squelette nous a paru la représentation la plus flexible, car il est invariant par rotation, translation et changement d'échelle. Pour la classification, les machines à noyaux nous fournissent un puissant cadre pour notre problématique. En effet, un noyau défini positif dans un domaine peut être interprété comme une mesure de similarité des éléments de ce domaine. Afin d'être robuste au bruit structurel, nous avons inséré un processus d'édition à l'intérieur de la comparaison de chemins. Ce processus nous permet de comparer des chemins de longueurs différentes mettant en avant les éléments communs dans une hiérarchie de réécritures. Cette flexibilité nous a offert la possibilité de comparer les formes à partir de sacs de chemins de faibles cardinalités contenant un minimum d'information, conduisant ainsi à des gains de rapidité.

Avec le noyau proposé, les formes sont comparées via des parcours de chemins dans les graphes. Cependant, bien qu'un chemin soit capable de représenter une sous-partie d'une forme, il est incapable d'exploiter efficacement les jonctions. Ce point est la faiblesse de cette approche et peut nuire aux performances. La perspective la plus immédiate est donc de pallier à ce défaut, par exemple, en remplaçant les chemins par des morceaux de graphes. Les sacs deviendront ainsi une sorte de sacs de vignettes, chaque morceau représentant un élément primitif de la forme. Une fois encore, le concept d'édition est nécessaire pour comparer les morceaux en les déformant. Néanmoins, le calcul, le choix et la comparaison des morceaux restent à déterminer.

Lorsque les formes présentent des trous caractéristiques, nous avons proposé de combiner les noyaux sur sacs avec un noyau sur faces. Cet a priori est pertinent quand le trou est significatif dans une classe d'objets, par exemple, les clés et les ciseaux. La robustesse au bruit structurel est assurée dans un processus de permutation des chemins représentant les trous et

par le processus d'édition précédent. Toutefois, l'importance d'un trou n'est jamais mesurée, donc nous considérons ceux issus du bruit avec la même importance que les autres. Il serait intéressant de chercher un moyen de caractériser les faces importantes d'un graphe sans induire une complexité temporelle trop élevée.

Munis d'un algorithme de restauration et d'un cadre pour comparer et classer les épines, nous avons étudié la chaîne de traitements d'analyse des épines à partir d'observations. Formée par une succession d'étapes, celle-ci repose sur des outils issus de diverses problématiques : restauration, segmentation, extraction de sous-structures présentes dans l'observation (excroissances sur l'axe dendritique et têtes d'épines déconnectées), etc. L'axe dendritique étant une structure curviligne et les épines détachées étant plutôt isotropes, nous avons utilisé le cadre de déconvolution proposé dans le chapitre 4 pour restaurer les images. La bonne qualité des résultats, nous a permis d'envisager une version simple de l'algorithme de segmentation et de nous concentrer sur la détection et la catégorisation des épines. Celle-ci est effectuée en étendant notre représentation par graphe à la 3D tubulaire, ainsi les noyaux restent valables. La représentation par graphe nous conduit à une manipulation efficace de l'axe pour l'extraction des épines. Toutefois, le calcul des différentes mesures demande de gérer convenablement l'anisotropie des images provenant du processus d'acquisition. Nous proposons d'enlever la composante en profondeur, en projetant toutes les données sur un seul plan, pour effectuer certaines mesures.

Cette anisotropie pourrait être corrigée avec une méthode de reconstruction des données manquantes. De plus, notre segmentation est des plus basiques et nous perdons des informations, comme des morceaux de structures. Nous pensons qu'une meilleure approche consisterait à appliquer une méthode prenant en compte un a priori de forme. Ainsi, certaines épines déconnectées seraient reliées à l'axe dendritique qui lui-même serait préservé. Cela conduirait à de meilleures mesures. De plus, avec la reconstruction les épines présenteront une forme complète et leur comparaisons se placeraient dans de meilleures conditions. Les différentes applications des machines à noyaux sur les épines restent aussi à explorer. Par exemple, l'analyse en composantes principales (ACP) à noyaux peut donner les points communs entre les épines dans un ensemble donné. Evidemment, tout cela dépend de l'interprétation de l'espace engendré par le noyau de comparaison. En perspective, avec des sacs contenant des morceaux caractéristiques des épines et un noyau défini positif capable de les comparer, de nombreuses informations peuvent être extraites d'un ensemble d'épines rien qu'avec une ACP et les autres méthodes à noyaux offriront d'autres types de renseignements.

Production scientifique

Publications

Journaux internationaux

[1] F.-X. Dupé, Jalal M. Fadili, J.-L. Starck, "A proximal iteration for deconvolving Poisson noisy images using sparse representations", IEEE Transactions on Image Processing, Vol. 18, No. 2, pp. 310-321, 2009.

Conférences internationales avec actes

[2] F.-X. Dupé, L. Brun, "Shape Classification Using a Flexible Graph Kernel", CAIP 2009, Münster, Allemagne, 2009.

[3] F.-X. Dupé, L. Brun, "Tree covering within a graph kernel framework for shape classification.", ICIAP 2009, Salerne, Italie, 2009.

[4] F.-X. Dupé, L. Brun, "Edition within a graph kernel framework for shape recognition", GBR 2009, Venise, Italie, 2009.

[5] F.-X. Dupé, L. Brun, "Hierarchical bag of paths for kernel based shape classification", SSPR 2008, Orlando, USA, 2008.

[6] F.-X. Dupé, M.J. Fadili, J.-L. Starck, "Deconvolution of confocal microscopy images using proximal iteration and sparse representations", ISBI 2008, Paris, France, 2008.

[7] F.-X. Dupé, M.J. Fadili, J.-L. Starck, "Image deconvolution under Poisson noise using sparse representations and proximal thresholding iteration", ICASSP 2008, Las Vegas, USA, 2008.

Logiciels

La première partie a donné lieu à la création d'une boîte à outils pour Matlab : Poisson-Deconv. Celle-ci contient le code nécessaire pour refaire les expériences présentées dans la section 4.6 (sans l'algorithme non-stabilisé). De même un logiciel à destination des biologistes a aussi été développé, il propose entre autre les deux modèles de PSF dont nous avons discuté dans le chapitre 3.

Nous avons aussi développé une boîte à outils contenant tous les algorithmes testés pour les méthodes à noyaux. Cette boîte est destinée à être diffusée dans la communauté et contiendra les scripts nécessaires pour refaire les expériences de la section 6.6.

Enfin, un logiciel pour l'analyse des épines dendritiques a aussi été réalisé. Celui-ci permet d'appliquer l'ensemble des étapes de la chaîne de traitement (figure 7.3) à partir de l'étape d'extraction des épines.

Table des figures

0.1	La chaîne de traitements pour les observations d'épines dendritiques.	2
2.1	Opérateurs proximaux associés à des fonctions promouvant la parcimonie : à gauche, les différentes fonctions et à droite les opérateurs proximaux correspondants.	37
3.1	Microscope confocal à balayage laser dans une configuration classique.	56
3.2	Détails de l'objectif d'un microscope confocal.	56
3.3	Le changement d'indice de réfraction entre deux milieux est une source d'erreurs. Celui-ci est susceptible de provoquer des aberrations sphériques au niveau de la zone de transition entre les deux milieux.	59
3.4	Les deux modèles de diffraction selon le type de lentille : (a) système paraxial, l'angle ω est proche de 0, (b) système non-paraxial, $\omega \gg 0$. Dans les deux cas, nous recherchons la formule du champ de diffraction au niveau du point P . Le point O repère l'intersection entre le plan focal et l'axe optique et Q est le début d'un rayon sortant de la lentille allant vers le plan.	59
3.5	La fonction d'étalement du point au niveau du plan focal telle que donnée par les deux méthodes d'approximation présentées : (a) avec des gaussiennes [Zhang et al., 2007a] et (b) avec des polynômes de Zernike [Booth et Wilson, 2001]. . .	63
4.1	Illustration du concept de diversité morphologique et le choix du dictionnaire qu'il implique.	71
4.2	Plus le taux de comptage est élevé, plus la vraisemblance approximée par la transformée d'Anscombe est proche de celle de la loi de Poisson.	73
4.3	Le GCV (—), le MAE (---) et le MSE (---) pour le cameraman (a) et l'image simulée de neurone (b). La transformée en ondelettes discrète invariante par translation a été utilisée pour l'image du Cameraman, et la transformée en curvelets avec la simulation de neurone.	87

4.4	Déconvolution d'une image simulée de neurone (Intensité ≤ 30). (a) Originale, (b) Convoluée, (c) Convoluée & bruitée, (d) RL-TV [Dey et al., 2004], (e) NaiveGauss [Vonesch et Unser, 2007], (f) RL-MRS [Starck et Murtagh, 2006], (g) FTITPR [Willett et Nowak, 2004], (h) PoissonStab, (i) PoissonNS.	90
4.5	Déconvolution d'une image microscopique de cellule (Intensité ≤ 30). (a) Originale, (b) Convoluée, (c) Convoluée & bruitée, (d) RL-TV [Dey et al., 2004], (e) NaiveGauss [Vonesch et Unser, 2007], (f) RL-MRS [Starck et Murtagh, 2006], (g) FTITPR [Willett et Nowak, 2004], (h) PoissonStab, (i) PoissonNS.	91
4.6	Déconvolution du cameraman (Intensité ≤ 30). (a) Originale, (b) Convoluée, (c) Convoluée & bruitée, (d) RL-TV [Dey et al., 2004], (e) NaiveGauss [Vonesch et Unser, 2007], (f) RL-MRS [Starck et Murtagh, 2006], (g) FTITPR [Willett et Nowak, 2004], (h) PoissonStab, (i) PoissonNS.	93
4.7	Déconvolution de l'image simulée de ciel. (a) Originale, (b) Convoluée, (c) Convoluée & bruitée, (d) RL-TV [Dey et al., 2004], (e) NaiveGauss [Vonesch et Unser, 2007], (f) RL-MRS [Starck et Murtagh, 2006], (g) FTITPR [Willett et Nowak, 2004], (h) PoissonStab, (i) PoissonNS.	97
4.8	MAE moyen pour tous les algorithmes en fonction du niveau d'intensité.	98
4.9	Impact du dictionnaire sur la déconvolution d'une image simulée avec une intensité maximale de 30. (a) Originale, (b) Convoluée, (c) Convoluée & bruitée, (d) Restaurée avec TI-DWT, (e) Restaurée avec les curvelets, (f) Restaurée avec un dictionnaire contenant les deux transformées.	99
4.10	Impact du dictionnaire sur les résultats de déconvolution en fonction du maximum d'intensité sur plusieurs images de test : (a) image simulée de neurone, (b) cellule et (c) LigneGauss. Les traits pleins représentent la transformée TI-DWT, les traits hachurés à la transformée en curvelets, et les traits en pointillés au dictionnaire construit par la fusion des deux transformées précédentes.	100
4.11	Déconvolution d'un neurone réel. (a) Image originale bruitée, (b) Image restaurée avec PoissonStab, (c) Image restaurée avec PoissonNS.	101
4.12	Détails de la restauration de l'axe dendritique. (a) Une épine dégradée, (b) et (c) sa restauration avec respectivement PoissonStab et PoissonNS. (d) Un morceau de l'axe dendritique avec des épines, (e) et (f) sa restauration avec respectivement PoissonStab et PoissonNS.	102
5.1	Une main et son squelette	111
5.2	Les graphes directs de deux mains.	113
5.3	Les différents types de chocs (représentés par des cercles)	114
5.4	Les graphes de chocs de deux mains.	115

5.5	La règle du rectangle pour l'isomorphisme de sous-graphe.	118
5.6	Exemple de graphe d'association	120
5.7	Deux graphes possédant le même ensemble de chemins.	124
6.1	Les graphes sont constitués en utilisant les squelettes et les points d'inflexions. Les arêtes sont liées à leur zone d'influence.	132
6.2	Exemple de point d'inflexion dans une main.	133
6.3	Exemple d'évolutions de rayons et d'orientation.	135
6.4	Calcul d'une couverture : (a) extraction des chemins, (b) réduction de la redon- dance, (c) couverture d'arêtes, (d) ajout des symétriques.	143
6.5	Distance entre les données projetées sur une sphère. L'angle α représente la distance $d_{\text{sphere}}(w_1, w_2)$	146
6.6	La distance de Hausdorff entre deux cercles est $d + r_1 - r_2 $	147
6.7	Effets de certaines déformations de la forme sur le squelette.	149
6.8	Répercussions des opérations d'éditations sur la forme.	150
6.9	Zone d'influence avec le marcheur aléatoire. Le chemin considéré est représenté en pointillés (--). Les arêtes partagées par les deux zones sont représentées par une ligne hachurée et pointée (-·-).	151
6.10	Les 9 classes formant la base de 99 formes.	156
6.11	Les 6 classes de formes dont le ou les trous forment une information importante lorsqu'ils sont présents.	159
6.12	Résistance des noyaux aux deux types de déformations : ajout de barbules et ajout de points d'inflexions. Les noyaux testés sont $K_{\text{edit},5\%}$ (—), $K_{\text{classique},5\%}$ (-·-), K_{marches} (-·) et K_{Neuhaus} (·).	161
6.13	Les graphes basés sur le squelette peuvent parfois être très proches structurel- lement, bien que les formes soient très différentes.	164
7.1	Les épines sont des excroissances présentes sur les dendrites d'un neurone. . . .	174
7.2	Les épines sont généralement divisées en trois classes selon leur morphologie. .	174
7.3	La chaîne de traitements pour les observations d'épines dendritiques.	176
7.4	Exemple de traitements d'une observation de dendrite. (a) L'image originale. (b) L'image débruitée en utilisant un filtre médian 3×3 . (c) L'image segmentée avec un seuillage simple. (d) Le squelette de la structure dendritique, les épines forment alors de petites branches reliées à un axe centrale représentant la dendrite.	177
7.5	Exemple de schéma de classification pour les épines. La validation des épines est effectuée par un ensemble de critères biologiques à respecter.	181
8.1	Exemple d'image de dendrites bruitées.	186

8.2	Exemple d'image de dendrites restaurée avec l'algorithme 4.1 utilisant un dictionnaire contenant des ondelettes décimées.	186
8.3	Exemple de structures dendritiques après segmentation avec un seuillage général.	188
8.4	Exemple de squelette. Les épines forment de petites branches reliées à une ligne centrale.	188
8.5	Exemple de segmentation d'une zone de l'arborescence dendritique présentée dans la figure 8.3. Tous les éléments d'une couleur différente de celle de la dendrite sont considérés comme des épines.	190
8.6	Les deux axes principaux et le profil 1D d'une épine. Les deux axes servent de références pour le calcul de la largeur et la longueur de l'épine. Le profil 1D montre le nombre de voxels suivant l'axe de la longueur en partant de la dendrite. La première bosse est donc un morceau de la base appartenant à la dendrite, puis nous avons le cou et enfin la tête.	193
8.7	Exemple de mesures effectuées sur l'épine de couleur marron. Pour la classification, les deux méthodes proposées ici sont celles de Koh et al. [2002] (méthode 1) et celle de Rodriguez et al. [2008] (méthode 2).	194

Liste des tableaux

2.1	Les différentes combinaisons de schémas pour résoudre le problème (P3) : FB pour le schéma explicite-implicite à un pas (section 2.2.2.1), DR pour le schéma de Douglas-Rachford (section 2.2.3) et Nesterov pour le schéma explicite-implicite à pas multiple (section 2.2.2.2).	49
3.1	Les différents $\sigma_{(\cdot)}$ selon le type d'appareil pour l'approximation gaussienne de la PSF d'un microscope confocal. Les paramètres sont l_{ex} et l_{em} respectivement les longueurs d'ondes d'excitation et d'émission, n_{ref} l'indice de réfraction du milieu d'immersion, NA l'ouverture numérique, R le rayon d'ouverture du sténopé et ω le demi-angle d'acceptation. La fonction J_0 est la fonction de Bessel de première espèce d'ordre 0 et J_1 d'ordre 1.	62
4.1	Temps d'exécution pour l'image 256×256 simulée du Cameraman en utilisant la transformée TI-DWT ($N_{ext} = 200$).	92
5.1	Les principaux noyaux de bases	122
6.1	Résumé des descripteurs de la forme associé au graphe.	134
6.2	Les différents noyaux impliqués dans les expériences.	157
6.3	Les plages de valeurs testées pour chacun des paramètres.	160
6.4	Matrices de confusions avec la base de 99 formes.	162
6.5	Temps d'exécution pour le calcul de la matrice de Gram de la base d'entraînement.	163
6.6	Matrices de confusions avec la base de 165 formes avec $K_{edit,couverture}$ combiné avec un noyau sur faces (haut) et $K_{edit,couverture}$ seul (bas).	166
6.7	Matrices de confusions avec la base de 165 formes avec $K_{classique,couverture}$ combiné avec un noyau sur faces (haut) et $K_{classique,couverture}$ seul (bas).	167
7.1	Critère de catégorisation des épines selon les ratios entre la longueur L , le diamètre du cou d_c et celui de la tête d_t	181
8.1	Résumé des descripteurs de la forme associée au graphe dans le cas de la 3D tubulaire.	189

Liste des Algorithmes

2.1	Décomposition explicite-implicite à un pas pour résoudre (P).	40
2.2	Décomposition explicite-implicite à un pas estimé pour résoudre (P).	40
2.3	Décomposition explicite-implicite à pas multiple pour résoudre (P).	41
2.4	Schéma de Douglas-Rachford pour résoudre (P).	43
2.5	Itération explicite-implicite à un pas pour le calcul de l'opérateur proximal d'une fonction composée avec un opérateur affine.	44
2.6	Schéma itératif de Nesterov à pas multiple pour le calcul de l'opérateur proximal d'une fonction composée avec un opérateur affine.	45
2.7	Schéma général du premier type d'algorithme pour résoudre (P3).	49
2.8	Schéma de décomposition pour résoudre (PN).	52
4.1	Déconvolution sous bruit de Poisson avec stabilisation du bruit, résout $(P_{F_{stab},\gamma,\psi})$	77
4.2	Déconvolution sous bruit de Poisson avec stabilisation du bruit dans le domaine des pixels, résout $(P'_{F_{stab},\gamma,\psi})$	80
4.3	Déconvolution sous bruit de Poisson sans stabilisation, résout $(P_{F_{poisson},\gamma,\psi})$	81
4.4	Déconvolution sous bruit de Poisson sans stabilisation dans le domaine des pixels, résout $(P'_{F_{poisson},\gamma,\psi})$	83
6.1	Procédure <i>tousLesChemins</i> : calcul tous les chemins d'un type donné et d'une taille inférieure à une borne, puis les mets dans un sac de chemins.	138
6.2	Procédure récursive <i>calculChemins</i> : calcul tous les chemins d'un type donné et d'une taille inférieure à une borne, à partir d'un sommet de départ, puis les mets dans un sac de chemins.	138
6.3	Algorithme de couverture d'arbre.	141
6.4	Calcul des différentes combinaisons de couverture pour un arbre.	142
8.1	Calcul des lignes centrales.	189

Références bibliographiques

- M. Aharon, M. Elad, et A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006.
- H. Akaike, B. N. Petrox, et F. Caski. Information theory and an extension of the maximum likelihood principle. *Second International Symposium on Information Theory. Akademiai Kiado, Boudapest.*, pages 267–281, 1973.
- Z. Aktouf, G. Bertrand, et L. Perroton. A three-dimensional holes closing algorithm. *Pattern Recognition Letters*, 23:523–531, 2002.
- H. C. Andrews et B. R. Hunt. *Digital Image Restoration*. Prentice-Hall, 1977.
- F. J. Anscombe. The Transformation of Poisson, Binomial and Negative-Binomial Data. *Biometrika*, 35:246–254, 1948.
- A. Antoniadis et Bigot. Poisson inverse problems. *Annals of Statistics*, 34:1811–1825, 2006.
- N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, May 1950.
- C. Aslan, A. Erdem, E. Erdem, et S. Tari. Disconnected skeleton: Shape at its absolute scale. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(12):2188–2203, 2008.
- W. Bai, X. Zhou, L. Ji, J. Cheng, et S. T. C. Wong. Automatic dendritic spine analysis in two-photon laser scanning microscopy images. *Cytometry Part A*, 71A(10):818–826, 2007.
- X. Bai et L. J. Latecki. Path similarity skeleton graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(7):1282–1292, 2008.
- J. Bang-Jensen et G. Gutin. *Digraphs Theory, Algorithms and Applications*. Springer-Verlag, 2008.
- A. Barron et T. Cover. Minimum complexity density estimation. *IEEE Transactions on Information Theory*, 37:1034–1054, 1991.

- H. G. Barrow et R. Burstall. Subgraph isomorphism, matching relational structures and maximal cliques. *Information Processing Letters*, 4(4):83–84, 1976.
- H. H. Bauschke et P. L. Combettes. A dykstra-like algorithm for two monotone operators. *Pacific Journal of Optimization*, 4(3):383–391, September 2008.
- A. Beck et M. Teboulle. Fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2:183–202, 2009.
- C. Berg, J. P. R. Christensen, et P. Ressel. *Harmonic Analysis on Semigroups*. Springer-Verlag, 1984.
- C. Berge. *Théorie des graphes et ses applications*. Dunod, 1958.
- H. Blum. A transformation for extracting new descriptors of shape. Dans W. Wathen-Dunn, éditeur, *Models for the Perception of Speech and Visual Form*, pages 363–380. MIT Press, 1967.
- M. Booth et T. Wilson. Refractive-index-mismatch induced aberrations in single-photon and two-photon microscopy and the use of aberration correction. *Journal of Biomedical Optics*, 6(3):266–272, 2001.
- K. M. Borgwardt et H.-P. Kriegel. Shortest-path kernels on graphs. Dans *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM 2005), 27-30 November 2005, Houston, Texas, USA*, pages 74–81. IEEE Computer Society, 2005.
- S. Bouix, K. Siddiqi, et A. Tannenbaum. Flux driven automatic centerline extraction. *Medical Image Analysis*, 9(3):209 – 221, 2005.
- J. Boutet de Monvel, S. Le Calvez, et M. Ulfendahl. Image Restoration for Confocal Microscopy: Improving the Limits of Deconvolution, with Application of the Visualization of the Mammalian Hearing Organ. *Biophysical Journal*, 80:2455–2470, 2001.
- J. Boutet de Monvel, E. Scarfone, S. Le Calvez, et M. Ulfendahl. Image-Adaptive Deconvolution for Three-Dimensional Deep Biological Imaging. *Biophysical Journal*, 85:3991–4001, 2003.
- J. S. Bridle. *Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters*, pages 211–217. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990. ISBN 1-55860-100-7.
- A. Bruckstein, D. Donoho, et M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review*, 51(1):34–81, 2009.

- L. Brun et J. Pruvot. Hierarchical matching using combinatorial pyramid framework. Dans *ICISP 2008*, volume 5099, pages 346–355, Cherbourg, 2008.
- H. Bunke et U. Buhler. Applications of approximate string matching to 2D shape recognition. *Pattern Recognition*, 26(12):1797–1812, 1993.
- H. Bunke et K. Riesen. A family of novel graph kernels for structural pattern recognition. Dans *CIARP 2007*, pages 20–31, 2007.
- M. Burger et O. Scherzer. Regularization methods for blind deconvolution and blind source separation problems. *Mathematics of control signals and systems*, 14:358–383, 2001.
- S. R. Cajal. Reglas y consejos sobre investigación biológica. Imprenta de Fontanet, Madrid, 1899.
- E. Candès, L. Demanet, D. Donoho, et L. Ying. Fast discrete curvelet transforms. *SIAM Multiscale Model. Simul.*, 5:861–899, 2005.
- E. J. Candès et D. L. Donoho. Curvelets – a surprisingly effective nonadaptive representation for objects with edges. Dans A. Cohen, C. Rabut, et L. Schumaker, éditeurs, *Curve and Surface Fitting: Saint-Malo 1999*, Nashville, TN, 1999. Vanderbilt University Press.
- W. Carrington. Image restoration in 3D microscopy with limited data. Dans *SPIE Proceedings*, volume 1205, pages 72–83, 1990.
- W. A. Carrington, R. M. Lynch, E. D. W. Moore, G. Isenberg, K. E. Fogarty, et F. S. Fay. Superresolution three-dimensional images of fluorescence in cells with minimal light exposure. *Science*, 268:1483–1487, 1995.
- L. Cavalier et J.-Y. Koo. Poisson intensity estimation for tomographic data using a wavelet shrinkage approach. *IEEE Transactions on Information Theory*, 48:2794–2802, 2002.
- A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20(1-2):89–97, 2004.
- O. Chapelle, V. Vapnik, O. Bousquet, et S. Mukherjee. Choosing Multiple Parameters for Support Vector Machines. *Machine Learning*, 46(1-3):131–159, 2002.
- C. Chaux, L. Blanc-Féraud, et J. Zerubia. Wavelet-based restoration methods: application to 3D confocal microscopy images. Dans *SPIE Wavelets XII*, volume 6701, page 67010E, 2007a.
- C. Chaux, P. L. Combettes, J.-C. Pesquet, et V. R. Wajs. A variational formulation for frame-based inverse problems. *Inv. Prob.*, 23:1495–1518, 2007b.

- C. Chaux, J.-C. Pesquet, et N. Pustelnik. Nested iterative algorithms for convex constrained image recovery problems. *SIAM Journal on Imaging Sciences*, 2(2):730–762, 2009.
- J. Cheng, X. Zhou, E. Miller, R. M. Witt, J. Zhu, B. L. Sabatini, et S. T. Wong. A novel computational approach for automatic dendrite spines detection in two-photon laser scan microscopy. *Journal of Neuroscience Methods*, 165(1):122–134, 2007.
- F. R. K. Chung. *Spectral Graph Theory*. CBMS Regional Conference Series in Mathematics. American Mathematical Society, 1997.
- P. G. Ciarlet. *Introduction à l'analyse numérique matricielle et à l'optimisation*. Masson-Paris, 1985.
- A. Cohen, R. DeVore, P. Petrushev, et H. Xu. Nonlinear approximation and the space $BV(\mathbb{R}^2)$. *American Journal of Mathematics*, 121:587–628, 1999.
- A. Cohen, M. Hoffman, et M. Reiss. Adaptive wavelet-galerkin methods for inverse problems. *SIAM J. Numerical Analysis*, 42:1479–1501, 2004.
- P. Combettes, D. Dung, et B. C. Vu. Fenchel-rockafellar dualization of signal recovery problems. presentation at the workshop on Approximation and Optimization in Image Restoration and Reconstruction, June 8-12 2009.
- P. L. Combettes. Solving monotone inclusions via compositions of nonexpansive averaged operators. *Optimization*, 53(5-6):475–504, December 2004.
- P. L. Combettes et S. A. Hirstoaga. Equilibrium programming in Hilbert spaces. *Journal of Nonlinear and Convex Analysis*, 6:117–136, 2005.
- P. L. Combettes et J.-. Pesquet. A Douglas-Rachford splitting approach to nonsmooth convex variational signal recovery. *IEEE Journal of Selected Topics in Signal Processing*, 1(4): 564–574, 2007.
- P. L. Combettes et J.-C. Pesquet. A proximal decomposition method for solving convex variational inverse problems. *Inverse Problems*, 24(6), 2008.
- P. L. Combettes et V. R. Wajs. Signal recovery by proximal forward-backward splitting. *SIAM Multiscale Model. Simul.*, 4(4):1168–1200, 2005.
- J.-A. Conchello et Q. Yu. Parametric blind deconvolution of fluorescence microscopy images: Preliminary results. *SPIE*, 2655:164–174, 1996.
- C. Cortes, P. Hallner, et M. Mohri. Rational Kernels: Theory and Algorithms. *Journal of Machine Learning Research*, 5:1035–1062, 2004.

- M. Couprie, D. Coeurjolly, et R. Zrou. Discrete bisector function and Euclidean skeleton in 2D and 3D. *Image Vision Comput.*, 25(10):1543–1556, 2007.
- M. Cuturi, J.-P. Vert, O. Birkenes, et T. Matsui. A kernel for time series based on global alignments. *CoRR*, abs/cs/0610033, 2006.
- I. Daubechies, M. Defrise, et C. D. Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraints. *Comm. Pure Appl. Math.*, 112:1413–1541, 2004.
- L. Demanet et L. Ying. Wave atoms and sparsity of oscillatory patterns. *Applied and Computational Harmonic Analysis*, 23(3):368–387, 2007.
- F. Desobry, M. Davy, et C. Doncarli. An online kernel change detection algorithm. *IEEE Transaction on Signal Processing*, 53(8):2961–2974, August 2005.
- N. Dey, L. Blanc-Féraud, J. Zerubia, C. Zimmer, J.-C. Olivo-Marin, et Z. Kam. A deconvolution method for confocal microscopy with total variation regularization. Dans *Proceedings of the 2004 IEEE International Symposium on Biomedical Imaging: From Nano to Macro, Arlington, VA, USA, 15-18 April 2004*, pages 1223–1226. IEEE, 2004.
- R. Diestel. *Graph Theory*. Springer-Verlag, 2005.
- I. DiMatteo, C. Genovese, et R. Kass. Bayesian curve fitting with free-knot splines. *Biometrika*, 88:1055–1071, 2001.
- D. L. Donoho. Nonlinear solution of inverse problems by wavelet-vaguelette decomposition. *Applied and Computational Harmonic Analysis*, 2:101–126, 1995.
- C. Dossal. A necessary and sufficient condition for exact recovery by ℓ_1 minimization, 2007. URL <http://hal.archives-ouvertes.fr/hal-00164738/en/>. Preprint, hal-00164738.
- J. Douglas et H. H. Rachford. On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American Mathematical Society*, 82:421–439, 1956.
- J. Eckstein. *Splitting methods for monotone operators with application to parallel optimization*. Thèse de doctorat, MIT, 1989.
- J. Eckstein et D. P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Programming*, 55:293–318, 1992.
- J. Eckstein et B. Svaiter. General projective splitting methods for sums of maximal monotone operators. Rapport Technique RRR 23-2007, RUTCOR, Rutgers University, august 2007.

- J. Eckstein et B. F. Svaiter. A family of projective splitting methods for the sum of two maximal monotone operators. *Mathematical Programming, Series B*, 111:173–199, 2008.
- B. Efron. The estimation of prediction error: covariance penalties and cross-validation. *Journal of the American Statistical Association*, 99:619–642, 2004.
- B. Efron. How biased is the apparent error rate of a prediction rule. *Journal of the American Statistical Association*, 81:461–470, 1981.
- I. Ekeland et R. Teman. *Analyse convexe et problème variationnels*. Dunod, 1974.
- J. Fadili et G. Peyré. Total Variation Projection with First Order Schemes. 2009. URL <http://hal.archives-ouvertes.fr/hal-00380491/en/>.
- M. Fadili et E. Bullmore. Penalized partially linear models using sparse representations with an application to fMRI time series. *IEEE Transactions on Signal Processing*, 53(9):3436–3448, 2005.
- M. Fadili, J.-L. Starck, et F. Murtagh. Inpainting and zooming using sparse representations. *The Computer Journal*, 2006.
- M. J. Fadili et J.-L. Starck. Sparse representation-based image deconvolution by iterative thresholding. Dans *ADA IV*, France, 2006. Elsevier.
- M. J. Fadili et J.-L. Starck. Monotone operator splitting for fast solutions in sparse recovery. *SIAM*, 2009. to be submitted.
- D. Field. Wavelets, vision and the statistics of natural scenes. *Philosophical Transactions of the Royal Society of London A*, 357:2527–2542, 1999.
- M. Figueiredo et R. Nowak. An em algorithm for wavelet-based image restoration. *IEEE Transactions on Image Processing*, 12:906–916, 2003.
- M. Fisz. The limiting distribution of a function of two independent random variables and its statistical application. *Colloquium Mathematicum*, 3:138–146, 1955.
- G. Fudenberg et L. Paninski. Bayesian Image Recovery for Dendritic Structures Under Low Signal-to-Noise Conditions. *IEEE Transactions on Image Processing*, 18(3):471–482, March 2009.
- D. Gabay. Applications of the method of multipliers to variational inequalities. Dans M. Fortin et R. Glowinski, éditeurs, *Augmented Lagrangian Methods: Applications to the Solution of Boundary-Value Problems*. North-Holland, Amsterdam, 1983.

- P. García-López, V. García-Marín, et M. Freire. The discovery of dendritic spines by cajal in 1888 and its relevance in the present neuroscience. *Progress in Neurobiology*, 83(2):110 – 130, 2007.
- T. Gärtner. Exponential and geometric kernels for graphs. Dans *NIPS*02 workshop on unreal data*, volume Principles of modeling nonvectorial data, 2002.
- T. Gärtner, P. Flach, et S. Wrobel. On graph kernels: Hardness results and efficient alternatives. Dans B. Schölkopf et M. K. Warmuth, éditeurs, *Proc. Annual Conf. Computational Learning Theory*, pages 129–143. Springer, 2003.
- Y. Gdalyahu et D. Weinshall. Flexible syntactic matching of curves and its application to automatic hierarchical classification of silhouettes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(12):1312–1328, 1999.
- S. Gibson et F. Lanni. Experimental test of an analytical model of aberration in an oil-immersion objective lens used in three dimensional light microscopy. *Journal of Optical Society of America A*, 8(10):1601–1613, October 1991.
- W.-B. Goh. Strategies for shape matching using skeletons. *Computer Vision and Image Understanding*, 110:326–345, 2008.
- S. Gold et A. Rangarajan. A graduated assignment algorithm for graph matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(4):377–388, 1996.
- G. H. Golub, M. Heath, et G. Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979.
- L. Grady. Random Walks for Image Segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(11):1768–1783, 2006.
- M. Gu. *Advanced Optical Imaging Theory*. Berlin, 2000.
- J. Guo, R. Niedermeier, et J. Uhlmann. Two fixed-parameter algorithms for vertex covering by paths on trees. *Information Processing Letters*, 106:81–86, 2008.
- B. Haasdonk. Feature space interpretation of svms with indefinite kernels. *IEEE Trans. on PAMI*, 27(4):482–492, April 2005.
- B. Haasdonk et E. Pełkalska. Classification with Kernel Mahalanobis Distance Classifiers. Dans *German Classification Society Annual Conference*, 2008.
- F. Harary. *Graph theory*. Addison-Wesley, 1969.

- D. Haussler. Convolution kernels on discrete structures. Rapport Technique UCSC-CRL-99-10, Department of Computer Science, University of California at Santa Cruz, 1999.
- A. Hillery et R. Chin. Iterative Wiener filters for image restoration. *IEEE Transactions on Signal Processing*, 39(8):1892–1899, 1991.
- T. Hofmann, B. Schölkopf, et A. J. Smola. Kernel methods in machine learning. *The Annals of Statistics*, 36(3):1171–1220, 2008.
- T. Horváth. Cyclic pattern kernels revisited. Dans *PAKDD*, pages 791–801. Springer, 2005.
- T. Horváth, T. Gärtner, et S. Wrobel. Cyclic pattern kernels for predictive graph mining. Dans W. Kim, R. Kohavi, J. Gehrke, et W. DuMouchel, éditeurs, *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, 2004*, pages 158–167. ACM, 2004.
- A. Hyvärinen et P. Hoyer. A two-layer sparse coding model learns simple and complex cell receptive fields and topography from natural images. *Vision Research*, 41:2413–2433, 2001.
- G. Jammal et A. Bijaoui. Dequant: a flexible multiresolution restoration framework. *Signal Processing*, 84:1049–1069, 2004.
- F. Janoos, K. Mosaliganti, X. Xu, R. Machiraju, K. Huang, et S. T. Wong. Robust 3D reconstruction and identification of dendritic spines from optical microscopy imaging. *Medical Image Analysis*, 13(1):167–179, 2009.
- M. Jansen, M. Malfait, et A. Bultheel. Generalized cross validation for wavelet thresholding. *Signal Processing*, 56(1):33–44, 1997.
- P. Jansson. *Deconvolution of Images and Spectra*. Academic Press, New York, 1997.
- P. A. Jansson. Method for Determining the Response Function of a High-Resolution Infrared Spectrometer. *Journal of the Optical Society of America*, 60(2):184–190, 1970.
- H. Kashima et T. Koyanagi. Kernels for semi-structured data. Dans C. Sammut et A. G. Hoffmann, éditeurs, *Machine Learning, Proceedings of the Nineteenth International Conference (ICML 2002), University of New South Wales, Sydney, Australia, July 8-12, 2002*, pages 291–298. Morgan Kaufmann, 2002.
- H. Kashima, K. Tsuda, et A. Inokuchi. Marginalized kernels between labeled graphs. Dans T. Fawcett et N. Mishra, éditeurs, *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA*, pages 321–328. AAAI Press, 2003.

- S. Kawata et Y. Ichioka. Iterative image restoration for linearly degraded images. i. basis and ii. reblurring procedure. *Journal of the Optical Society of America*, 70:762–772, 1980.
- S. Kaygin et M. M. Bulut. Shape recognition using attributes string matching with polygon vertices as the primitives. *Pattern Recognition Letters*, 23:287–294, 2002.
- S. S. Keerthi, V. Sindhwani, et O. Chapelle. An Efficient Method for Gradient-Based Adaptation of Hyperparameters in SVM Models. Dans B. Schölkopf, J. C. Platt, et T. Hoffman, éditeurs, *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, pages 673–680, 2006.
- G. Kempen. *Image Restoration in Fluorescence Microscopy*. Thèse de doctorat, Technische Universiteit Delft - Holland, January 1999.
- G. Kempen et L. van Vliet. The influence of the regularization parameter and the first estimate on the performance of Tikhonov regularized non-linear image restoration algorithm. *Journal of Microscopy*, 198:63–75, April 2000.
- W.-Y. Kim et Y.-S. Kim. A New region-Based Shape Descriptor. Dans *Mpeg Meeting TP 15-01*, Pise, Italie, Décembre 1999.
- P. N. Klein, T. B. Sebastian, et B. B. Kimia. Shape matching using edit-distance: an implementation. Dans *SODA*, pages 781–790, 2001.
- I. Y. Y. Koh, W. B. Lindquist, K. Zito, E. A. Nimchinsky, et K. Svoboda. An image analysis algorithm for dendritic spines. *Neural Computation*, 14(6):1283–1310, 2002.
- R. I. Kondor et J. D. Lafferty. Diffusion kernels on graphs and other discrete input spaces. Dans C. Sammut et A. G. Hoffmann, éditeurs, *Machine Learning, Proceedings of the Nineteenth International Conference (ICML 2002), University of New South Wales, Sydney, Australia, July 8-12, 2002*, pages 315–322. Morgan Kaufmann, 2002.
- I. Kovács, A. Féhér, et B. Julesz. Medial-point description of shape: a representation for action coding and its psychophysical correlates. *Vision Research*, 38:2323–2333, 1998.
- K. Kreutz-Delgado, J. Murray, B. Rao, K. Engan, T. Lee, et T. Sejnowski. Dictionary learning algorithms for sparse representation. *Neural Computation*, 15(2):349–396, 2003.
- R. Lagendijk et J. Biemond. *Iterative identification and restoration of images*. Kluwer Academic Publishers, 1991.

- L. Landweber. An iteration formula for Fredholm integral equations of the first kind. *American Journal of Mathematics*, 73:615–624, 1951.
- K. Lange. *Numerical Analysis for Statisticians*. Springer-Verlag, New York, 1999.
- C. Lemaréchal et J.-B. Hiriart-Urruty. *Convex Analysis and Minimization Algorithms I*. Springer, 2nd edition, 1996.
- M. S. Lewicki et T. J. Sejnowski. Learning overcomplete representations. *Neural Computation*, 12(2):337–365, 2000.
- F. F. Leymarie et B. B. Kimia. The shock scaffold for representing 3D shape. Dans *In Proc. of 4th International Workshop on Visual Form (IWVF4)*, pages 216–229. Springer-Verlag, 2001.
- G. Lin, Z. Cai, et D. Lin. Vertex covering by paths on trees with its applications in machine translation. *Information Processing Letters*, 97:73–81, 2006.
- P.-L. Lions. Une méthode itérative de resolution d’une inéquation variationnelle. *Israel Journal of Mathematics*, 31:204–208, 1978.
- P.-L. Lions et B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM J. Numer. Anal.*, 16:964–979, 1979.
- M. Locatelli, I. M. Bomze, et M. Pelillo. The combinatorics of pivoting for the maximum weight clique. *Operations Research Letters*, 32(6):523–529, 2004.
- L. Lucy. An iterative technique for the rectification of observed distributions. *Astronomy Journal*, 79:745–765, 1974.
- P. Mahé et J.-P. Vert. Graph kernels based on tree patterns for molecules. *Machine Learning*, 75:3–35, 2009.
- P. Mahé, N. Ueda, T. Akutsu, J.-L. Perret, et J.-P. Vert. Extensions of marginalized graph kernels. Dans C. E. Brodley, editeur, *Machine Learning, Proceedings of the Twenty-first International Conference (ICML 2004), Banff, Alberta, Canada, July 4-8, 2004*, volume 69 of *ACM International Conference Proceeding Series*. ACM, 2004.
- S. G. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 2nd edition, 1998.
- B. Martinet. Détermination approchée d’un point d’une application pseudo-contractante. *Comptes Rendus de l’Académie des Sciences de Paris*, 274:163–165, 1972.

- A. Massaro et M. Pelillo. Matching graphs by pivoting. *Pattern Recognition Letters*, 24(8): 1099–1106, 2003.
- J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philos. Trans. Roy. Soc. London*, 209, 1909.
- J. H. Metzen, T. Kröger, A. Schenk, S. Zidowitz, H.-O. Peitgen, et X. Jiang. Matching of tree structures for registration of medical images. Dans *GbRPR*, pages 13–24, 2007.
- F. Meyer. Topographic distance and watershed lines. *Signal Processing*, 38(1):113–125, 1994.
- M. Minsky. Microscopy apparatus. U.S. Patent #3013467, 1957.
- M. Minsky. Memoir on inventing the confocal scanning microscope. *Scanning*, 10:128–138, 1988.
- R. Molina, J. Mateos, et A. K. Katsaggelos. Blind deconvolution using a variational approach to parameter, image, and blur estimation. *IEEE Trans. on image processing*, 15:3715–3727, 2006.
- J.-J. Moreau. Fonctions convexes duales et points proximaux dans un espace hilbertien. *CRAS Sér. A Math.*, 255:2897–2899, 1962.
- J.-J. Moreau. Propriétés des applications “prox”. *Comptes Rendus de l’Académie des Sciences Série A Mathématiques*, 256:1069–1071, 1963.
- J.-J. Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société Mathématique de France*, 93:273–299, 1965.
- R. Mukundan et K. R. Ramakrishnan. *Moment Functions in Image Analysis : Theory and Applications*. World Scientific, 1998.
- F. Murtagh, J.-L. Starck, et A. Bijaoui. Image Restoration with Noise Suppression Using a Multiresolution Support. *Astronomy and Astrophysics*, 112:179, 1994.
- Y. Nesterov. Gradient methods for minimizing composite objective function. Rapport Technique CORE DP-2007/76, Center for Operations Research and Econometrics (CORE), Catholic University of Louvain (UCL), september 2007.
- M. Neuhaus et H. Bunke. Edit-distance based kernel for structural pattern classification. *Pattern Recognition*, 39:1852–1863, 2006.
- M. Neuhaus et H. Bunke. A quadratic programming approach to the graph edit distance problem. Dans *GbRPR*, pages 92–102, 2007a.

- M. Neuhaus et H. Bunke. *Bridging the Gap between Graph Edit Distance and Kernel Machines*, volume 68 of *Machine Perception and Artificial Intelligence*. World Scientific, September 2007b.
- X. S. Ni et X. Huo. Another look at huber’s estimator: a new minimax estimator in regression with stochastically bounded noise. *Journal of Statistical Planning and Inference*, 2009. accepted.
- M. Novotni et R. Klein. Shape retrieval using 3D Zernike descriptor. *Computer-Aided Design*, 36(11):1047–1062, 2004.
- R. Nowak et E. Kolaczyk. A statistical multiscale framework for poisson inverse problems. *IEEE Transactions on Information Theory*, 46:2794–2802, 2000.
- B. Olshausen et D. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research.*, 37:3311–3325, 1996a.
- B. A. Olshausen et D. J. Field. Emergence of simple-cell receptive-field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996b.
- B. Pankajakshan, Praveenand Zhang, L. Blanc-Féraud, Z. Kam, J.-C. Olivo-Marin, et J. Zerubia. On blind deconvolution for thin layered confocal imaging. *Applied Optics*, 48(21), 2009.
- G. Passty. Ergodic convergence to a zero of the sum of monotone operators in hilbert space. *Journal of Mathematical Analysis and Applications*, 72:383–390, 1979.
- M. Pavan et M. Pelillo. Dominant sets and pairwise clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):167–172, 2007.
- J. Pawley. *Handbook of Confocal Microscopy*. Plenum Press, 2005.
- M. Pelillo. Replicator equations, maximal cliques and graph isomorphism. Rapport Technique CVC TR-98-001/CS TR-1149, Departement of Computer Science, Yale University, 1998.
- M. Pelillo, K. Siddiqi, et S. Zucker. Matching hierarchical structures using association graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1105–1120, november 1999.
- A. Peters et I. Kaiserman-Abramof. The small pyramidal neuron of the rat cerebral cortex. The synapses upon dendritic spines. *Z. Zellforsch*, 100:487–506, 1969.
- G. Peyré, M. J. Fadili, et J.-L. Starck. Learning adapted dictionaries for geometry and texture separation. Dans *Wavelet XII*, San Diego, 2007.

- C. Pudney. Distance-Ordered Homotopic Thinning: A Skeletonization Algorithm for 3D Digital Images. *Computer Vision and Image Understanding*, 73(3):404–413, 1998.
- J. Ramon et T. Gärtner. Expressivity versus Efficiency of Graph Kernels. Dans T. Washio et L. D. Raedt, éditeurs, *First International Workshop on Minig Graphs, Trees and Sequences*, pages 65–74, 2003.
- S. W. Reyner. An analysis of a good algorithm for the subtree problem. *SIAM J. Comput.*, 6(4):730–732, 1977.
- T. Ridler et S. Calvard. Picture thresholding using an interactive selection method. *IEEE Transaction on Syst. Man. Cybern.*, SMC-8:630–632, 1978.
- K. Riesen, V. Kilchherr, et H. Bunke. Reducing the dimensionality of vector space embeddings of graphs. Dans P. Perner, éditeur, *Machine Learning and Data Mining in Pattern Recognition, 5th International Conference, MLDM 2007, Leipzig, Germany, July 18-20, 2007, Proceedings*, volume 4571 of *Lecture Notes in Computer Science*, pages 563–573. Springer, 2007a.
- K. Riesen, M. Neuhaus, et H. Bunke. Bipartite graph matching for computing the edit distance of graphs. Dans *GbrPR*, pages 1–12, 2007b.
- K. Rink et K. D. Tönnies. A modification of the level set speed function to bridge gaps in data. Dans K. Franke, K.-R. Müller, B. Nickolay, et R. Schäfer, éditeurs, *Pattern Recognition, 28th DAGM Symposium, Berlin, Germany, September 12-14, 2006, Proceedings*, volume 4174 of *Lecture Notes in Computer Science*, pages 152–161. Springer, 2006.
- K. Rink et K. D. Tönnies. A level set bridging force for the segmentation of dendritic spines. Dans W. G. Kropatsch, M. Kampel, et A. Hanbury, éditeurs, *CAIP*, volume 4673 of *Lecture Notes in Computer Science*, pages 571–578. Springer, 2007.
- R. Rockafellar. *Convex analysis*. Princeton University Press, 1970.
- R. Rockafellar. *The Theory of Subgradients and its Applications to Problems of Optimization: Convex and Nonconvex Functions*. Helderman-Verlag, Berlin, 1981.
- A. Rodriguez, D. B. Ehlenberger, D. L. Dickstein, P. R. Hof, et S. L. Wearne. Automated three-dimensional detection and shape classification of dendritic spines from fluorescence microscopy images. *PLoS ONE*, 3(4):e1997, April 2008.
- B. Roy. *Algèbre moderne et théorie des graphes*. Dunod, 1969.

- C. D. Ruberto. Recognition of shapes by attributed skeletal graphs. *Pattern Recognition*, 37(1):21–31, 2004.
- D. A. Rusakov et M. G. Stewart. Quantification of dendritic spine populations using image analysis and a tilting disector. *Journal of Neuroscience Methods*, 60(1-2):11–21, 1995.
- P. Sarder et A. Nehorai. Deconvolution Method for 3-D Fluorescence Microscopy Images. *IEEE Sig. Pro. Mag.*, 23:32–45, 2006.
- S. Schmitt, J. F. Evers, C. Duch, M. Scholz, et K. Obermayer. New methods for the computer-assisted 3-d reconstruction of neurons from confocal image stacks. *NeuroImage*, 23(4):1283–1298, 2004.
- B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, et R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.
- G. Schwarz. Estimation of the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.
- T. Sebastian, P. Klein, et B. Kimia. Recognition of shapes by editing their shock graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):550–571, 2004.
- J. Shen et D. Shen. Orthogonal legendre moments and their calculation. Dans *ICPR'1996*, volume 2, 1996.
- X. Shen et J. Ye. Adaptive model selection. *Journal of the American Statistical Association*, 97:210–221, 2002.
- A. Shokoufandeh et S. J. Dickinson. Graph-theoretical methods in computer vision. Dans *Theoretical Aspects of Computer Science*, pages 148–174, 2000.
- A. Shokoufandeh, S. J. Dickinson, K. Siddiqi, et S. W. Zucker. Indexing using a spectral encoding of topological structure. Dans *CVPR*, pages 2491–2497, 1999.
- K. Siddiqi et B. B. Kimia. A shock grammar for recognition. Dans *1996 Conference on Computer Vision and Pattern Recognition (CVPR '96), June 18-20, 1996 San Francisco, CA, USA*, pages 507–513. IEEE Computer Society, 1996.
- K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, et S. W. Zucker. Shock Graphs and Shape Matching. *International Journal on Computer Vision*, 35(1):13–32, 1999.
- K. Siddiqi, B. B. Kimia, A. Tannenbaum, et S. W. Zucker. On the psychophysics of the shape triangle. *Vision Research*, 41(9):1153–1178, 2001.

- K. Siddiqi, S. Bouix, A. Tannenbaum, et S. W. Zucker. Hamilton-jacobi skeletons. *International Journal of Computer Vision*, 48(3):215–231, 2002.
- E. Simoncelli et B. Olshausen. Natural image statistics and neural representation. *Annual Review of Neuroscience*, 24:1193–1216, 2001.
- Soulier. *Du Serment d’Hippocrate à l’éthique médicale*. Thèse de doctorat, Faculté de médecine de Marseille, 1985.
- J.-L. Starck et F. Murtagh. *Astronomical Image and Data Analysis*. Springer, 2006.
- J.-L. Starck et F. Murtagh. Image restoration with noise suppression using the wavelet transform. *Astronomy and Astrophysics*, 288:343–348, 1994.
- J.-L. Starck, A. Bijaoui, et F. Murtagh. Multiresolution support applied to image filtering and deconvolution. *CVGIP: Graphical Models and Image Processing*, 57:420–431, 1995.
- J.-L. Starck, F. Murtagh, et A. Bijaoui. *Image Processing and Data Analysis*. Cambridge University Press, 1998.
- J.-L. Starck, M. Nguyen, et F. Murtagh. Wavelets and curvelets for image deconvolution: a combined approach. *Signal Processing*, 83:2279–2283, 2003.
- H. Stark. *Image Recovery: Theory and Application*. Elsevier Science & Technology Books, 1987.
- C. Stein. Estimation of the mean of a multivariate normal distribution. *Annals of Statistics*, 9:1135–1151, 1981.
- F. Suard, A. Rakotomamonjy, et A. Benrshair. Mining shock graphs with kernels. Rapport technique, LITIS, 2006. <http://hal.archives-ouvertes.fr/hal-00121988/en/>.
- F. Suard, A. Rakotomamonjy, et A. Benrshair. Kernel on bag of paths for measuring similarity of shapes. Dans *ESANN 2007, 15th European Symposium on Artificial Neural Networks, Bruges, Belgium, April 25-27, 2007, Proceedings*, pages 355–360, 2007.
- D. M. Tax et R. P. Duin. Support vector data description. *Machine Learning*, 54:45–66, 2004.
- G. Teschke. Multi-frame representations in linear inverse problems with mixed multi-constraints. *Applied and Computational Harmonic Analysis*, 22(1):43–60, 2007.
- A. Torsello et E. R. Hancock. Curvature Correction of the Hamilton-Jacobi Skeleton. Dans *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003), 16-22 June 2003, Madison, WI, USA*, pages 828–834, 2003.

- A. Torsello et E. R. Hancock. A skeletal measure of 2D shape similarity. *Computer Vision and Image Understanding*, 95(1):1–29, 2004.
- P. Tseng. A modified forward-backward splitting method for maximal monotone mappings. *SIAM J. Control & Optim.*, 38:431–446, 2000.
- P. Tseng. Applications of a splitting algorithm to decomposition in convex programming and variational inequalities. *SIAM Journal on Control and Optimization*, 29(1):119–138, 1991.
- W. T. Tutte. *Graph theory*. Cambridge university press, 1984.
- H. van der Voort et K. Strasters. Restoration of confocal images for quantitative image analysis. *Journal of Microscopy*, 178:165–181, 1995.
- S. Vishwanathan, K. M. Borgwardt, I. R. Kondor, et N. N. Schraudolph. Graph kernels. *Journal of Machine Learning Research*, 9:1–37, 2008.
- C. Vonesch et M. Unser. Fast wavelet-regularized image deconvolution. Dans *Proceedings of the 2007 IEEE International Symposium on Biomedical Imaging: From Nano to Macro, Washington, DC, USA, April 12-16, 2007*, pages 608–611, 2007.
- C. Vonesch et M. Unser. A fast thresholded Landweber algorithm for general wavelet bases: Application to 3D deconvolution microscopy. Dans *Proceedings of the 2008 IEEE International Symposium on Biomedical Imaging: From Nano to Macro, Paris, France, May 14-17, 2008*, pages 1351–1354, 2008.
- R. Watzel, K. Braun, A. Hess, H. Scheich, et W. Zuschratter. Detection of dendritic spines in 3-dimensional images. Dans G. Sagerer, S. Posch, et F. Kummert, éditeurs, *Mustererkennung 1995, 17. DAGM-Symposium, Bielefeld, 13.-15. September 1995, Proceedings*, Informatik Aktuell, pages 160–167. Springer, 1995.
- R. Watzel, K. Braun, A. Hess, W. Zuschratter, et H. Scheich. Restoration of Dendrites and Spines with the Objective of Topologically Correct Segmentation. Dans *Proceedings of the 1996 International Conference on Pattern Recognition (ICPR '96)*, pages 472–476. IEEE, 1996.
- C. M. Weaver, P. R. Hof, S. L. Wearne, et W. B. Lindquist. Automated algorithms for multiscale morphometry of neuronal dendrites. *Neural Computation*, 16(7):1353–1383, 2004.
- P. Weiss. *Algorithmes rapides d'optimisation convexe. Applications à la restauration d'images et à la détection de changements*. Thèse de doctorat, Université de Sophia-Antipolis, 2009.

- R. Willett et R. Nowak. Fast multiresolution photon-limited image reconstruction. Dans *IEEE ISBI*, 2004.
- R. C. Wilson et P. Zhu. A study of graph spectra for comparing graphs and trees. *Pattern Recognition*, 41:2833–2841, 2008.
- J. Ye. On measuring and correcting the effects of data mining and model selection. *Journal of the American Statistical Association*, 93:120–131, 1998.
- B. Zhang, J. Zerubia, et J.-C. Olivo-Marin. Gaussian approximations of fluorescence microscope point-spread function models. *Applied Optics*, 46(10):1819–1829, 2007a.
- Y. Zhang, X. Zhou, R. M. Witt, B. L. Sabatini, D. Adjero, et S. T. Wong. Dendritic spine detection using curvilinear structure detector and lda classifier. *NeuroImage*, 36(2):346–360, 2007b.
- W. Zhou, H. Li, et X. Zhou. 3D Dendrite Reconstruction and Spine Identification. Dans D. N. Metaxas, L. Axel, G. Fichtinger, et G. Székely, éditeurs, *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2008, 11th International Conference, New York, NY, USA, September 6-10, 2008, Proceedings, Part II*, volume 5242 of *Lecture Notes in Computer Science*, pages 18–26. Springer, 2008.
- H. Zou, T. Hastie, et R. Tibshirani. On the "degrees of freedom" of the lasso. *Annals of Statistics*, 35:2173–2192, 2007.

Index

- arbre
 - association, 119
 - chocs, 114
 - motif, 124
- atome, 68
- chemin, 108
 - élémentaire, 108
 - cycle, 109
 - eulerien, 109
 - hamiltonien, 109
 - pertinence, 139
 - simple, 108
- clique, 119
 - maximale, 119
- décomposition
 - Moreau, 35
- descripteur
 - global, 112
 - local, 112
- dictionnaire, 68
- ensemble
 - convexe, 30
 - fonction indicatrice, 31
 - enveloppe affine, 30
 - intérieur relatif, 30
- enveloppe de Moreau, 34
- fonction
 - épigraphique, 30
 - appui, 31
 - coercive, 30
 - conjuguée, 31
 - convexe, 30
 - convolution infimale, 32
 - dérivée, 32
 - domaine, 30
 - fortement convexe, 32
 - indicatrice, 31
 - inf-convolution, 32
 - régularisation de Moreau-Yosida, 35
 - semi-continue, 31
 - sous-différentielle, 32
 - sous-gradient, 32
 - support, 31
- graphe, 108
 - arbre, 109
 - connexe, 109
 - labélisé, 109
 - matrice
 - adjacence, 116
 - laplacien, 116
 - multigraphe, 108
 - planaire, 109
 - face, 109
 - pseudographe, 108
 - simple, 108

- sous-graphe, 108
- morphologique
 - composante, 70
- noyau
 - briques de base, 122
 - chemin, 122
 - marches aléatoires, 123
- opérateur
 - analyse, 43
 - Lipschitz continuité, 33
 - maximal monotone, 33
 - monotone, 33
 - non-expansif, 33
 - réflexion, 34
 - résolvante, 33
 - synthèse, 43
 - trame, 43
- opérateur proximal, 34
 - propriétés, 35
- opération
 - analyse, 68
 - synthèse, 68
- squelette, 110
 - feu de forêt, 110
 - graphe de chocs, 113
 - graphe direct, 112
 - propriétés, 110
- transformée
 - Legendre-Fenchel, 31

Résumé : Dans le cadre de l'étude du cerveau, les biologistes et les médecins s'intéressent à de petites excroissances sur les dendrites appelées *épines dendritiques*. Ces épines jouent un rôle important au niveau des transmissions synaptiques. Elles sont observées par fluorescence avec un microscope confocal. Les images acquises sont difficiles à exploiter car elles sont 1) dégradées par le flou dû aux limites de la diffraction et du bruit, 2) les épines présentent une grande diversité morphologique et 3) une dendrite peut posséder une densité importante d'épines. Cette thèse se décompose donc en trois parties : la première partie présente un cadre théorique d'optimisation aboutissant à des algorithmes de déconvolution sous bruit de Poisson. La deuxième partie concerne la comparaison de formes par leur morphologie avec des noyaux sur graphes. La troisième partie propose une plateforme d'exploitation d'images d'épines pour obtenir automatiquement des informations morphologiques sur ces épines.

Title : Restoration and shape recognition inside images with Poisson noise: application to the analysis of neurons acquired by fluorescence microscopy

Abstract: While studying the brain, biologists and doctors focus on some small protrusions on the dendrites which go by the name of *dendritic spines*. These spines play an important role in the synaptic transmission process. They are observed using a fluorescent confocal microscope. Observed images are hard to exploit because 1) they are degraded by diffraction-related blur and noise, 2) they present a large morphological diversity and 3) one dendrite may show an important density of spines. This thesis is then divided into three parts : the first part presents a theoretical optimization framework that is then applied to deconvolution under Poisson noise. The second part considers shape comparison using morphological information by graph kernels. The third part proposes a framework that deploys the image analysis tools developed in the previous chapters to get automatically morphological information about the observed spines.

Mots-Clés : Traitement d'images - Techniques numériques – Optimisation convexe – Analyse numérique – Noyaux (analyse fonctionnelle) – Reconnaissance des formes (informatique) – Traitement du signal – Microscopie de fluorescence – Imagerie en biologie

Discipline : Traitement du Signal et des Images

GREYC UMR CNRS 6072, équipe Image
ENSICAEN, 6 Boulevard du Maréchal Juin, 14050 Caen Cedex France