HAL
open science

Reconstruction, Détection et Régularisation de Données Discrètes

Sébastien Bougleux

To cite this version:

Sébastien Bougleux. Reconstruction, Détection et Régularisation de Données Discrètes. Interface homme-machine [cs.HC]. Université de Caen, 2007. Français. NNT : . tel-00203445

HAL Id: tel-00203445

https://theses.hal.science/tel-00203445v1

Submitted on 10 Jan 2008

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

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

UNIVERSITÉ de CAEN/BASSE-NORMANDIE

U.F.R Sciences

ÉCOLE DOCTORALE S.I.M.E.M.

THÈSE

présentée par

Mr. Sébastien BOUGLEUX

et soutenue

le 05/12/2007

en vue de l'obtention du

DOCTORAT de l'UNIVERSITÉ de CAEN

Spécialité : Informatique

(Arrêté du 07 août 2006)

Reconstruction, Détection et Régularisation de Données Discrètes

MEMBRES du JURY

Rapporteurs :	**J-M. Chassery**	**Directeur de Recherche CNRS, INPG, GIPSA**
	S. Foufou	**Maître de Conférences, HDR, Univ. de Bourgogne**
Examinateurs :	**E. Grandjean**	**Professeur, Université de Caen/B.-N.**
	M. Revenu	**Professeur, ENSICAEN**
Co-directeurs :	**A. Elmoataz**	**Professeur, Université de Caen/B.-N.**
	M. Melkemi	**Professeur, Université de Haute-Alsace**

Mis en page avec la classe thloria.

Remerciements

Ces années de thèse m'ont permis d'approfondir des problématiques relevant de deux domaines de recherche passionnants : la géométrie algorithmique et régularisation discrète sur graphes. Le travail que j'ai réalisé est dû à la patience, à l'attention et à l'investissement de mes deux encadrants, Abder Elmoataz et Mahmoud Melkemi, qui m'ont sans cesse aidé à faire face dans les moments de doutes. Pour tout, je les remercie, en espérant continuer à échanger nos idées sur des problématiques similaires, ou plus éloignées.

Je remercie tous les membres de mon jury, d'une part Jean-Marc Chassery et Sebti Foufou pour avoir accepté de rapporter mom mémoire de thèse, me permettant ainsi de préciser certains de mes propos, et pour s'être déplacé de bonne heure à ma soutenance. D'autre part, je rermercie Etienne Grandjean et Marinette Revenu pour leurs remarques pertinentes, leur présence, leurs conseils, et surtout leur accueil au sein du laboratoire GREYC pendant toutes les années de ma thèse.

Je remercie tous les permanents, tous les doctorants, et toute l'équipe administrative du GREYC, en particulier de l'équipe Image. Merci à Seb, Jalal, David et Luc pour les différentes discussions studieuses qui m'ont permis d'avancer, et Daniel et Régis qui furent mes guides spirituels en matière d'enseignement. Dans le même contexte, les années d'enseignement passées aux côtés de François Bourdon, de Jerzy Karczmarczuk et d'Hamid Rahbar, ont diversifié mes compétences et mes méthodes de travail.

Une mention spéciale à tous mes amis, rencontrés au cours de mes études. Merci à Arnaud (et Hélène) et à Jean-Hugues (Trouve-Tout) pour avoir partagé avec moi ces années de thèse (et surtout un bureau). Merci à Vinh et à Olivier pour le travail d'équipe réalisé concernant la régularisation discrète. Merci à Ninin et Nicolas pour leur curiosité et leur rigueur, à tous les Masenka, à ceux de toujours (P'ti Seb, David, Zam, Gues, Tita, et les autres) qui m'ont beaucoup aidé à gérer mon indisponibilité.

Je remercie ma famille, mes parents, ma fille, mon frère, et ma femme Valérie.

À mon père ...

Table des matières

Remerciements **i**

Résumé et Contributions **1**

I Reconstruction et Détection de Formes **3**

Introduction **5**

1 Concepts géométriques liés à la reconstruction et à la détection **7**

1.1 Notion de forme d'un ensemble de points 8

1.2 Enveloppe convexe, polytopes et simplexes 9

1.3 Complexes cellulaires et simpliciaux 10

1.4 Triangulation de Delaunay 11

1.5 Diagramme de Voronoï et dualité 11

2 Reconstruction de courbes polygonales simples **15**

2.1 Introduction 16

2.2 Notions liées au problème de la reconstruction 19

2.2.1 Courbes considérées 19

2.2.2 Squelette et axe médian de courbes 19

2.2.3 L'ϵ-échantillonnage 21

2.2.4 Le graphe de Gabriel 23

2.3 Analyse des approches de reconstruction existantes 24

2.3.1 Les β-squelettes 24

2.3.2 Les A-formes et le Crust 25

2.3.3 Le Crust local et l'anti-Crust 27

2.3.4 Le Crust des plus proches voisins 31

2.3.5 Le CRUST conservatif 31
2.3.6 Schéma récapitulatif des graphes de voisinage 33
2.4 Graphes proposés : β-CRUSTS locaux et β-axes médians 35
2.4.1 Notion de voisinage sur les arêtes de Voronoï 35
2.4.2 Définition des graphes 36
2.4.3 Relations avec le graphe de Gabriel 37
2.4.4 Propriété angulaire 38
2.4.5 Algorithme général connaissant β 39
2.5 Analyse des graphes proposés 41
2.5.1 Propriétés hiérarchiques et combinatoires 41
2.5.2 Relations avec le CRUST local et l'anti-CRUST 44
2.6 Algorithme glouton 51
2.6.1 Description de l'algorithme 51
2.6.2 Différences avec les β-CRUSTS locaux 53
2.6.3 Exemples de reconstruction 54
2.7 Expérimentations basées sur l'ϵ-échantillonnage 59
2.7.1 Approximation de l'axe médian 59
2.7.2 Estimation de la distance à l'axe médian 61
2.7.3 Calcul d'ϵ-échantillons 62
2.7.4 Résultats 63
2.8 Conclusion et perspectives 67

3 Extraction de Primitives Géométriques 69
3.1 Introduction 70
3.2 Les α-formes 72
3.3 Concepts anisotropes 73
3.3.1 Anisotropie globale 73
3.3.2 Diagrammes et triangulations anisotropes 74
3.3.3 Les α-formes anisotropes 76
3.3.4 Relations entre les concepts anisotropes et Euclidiens 76
3.3.5 Calcul d'α-formes anisotropes 77
3.4 Extraction anisotrope de structures 79
3.4.1 Cas des droites 81
3.4.2 Cas des plans 82
3.4.3 Suppression des structures non-significatives 83

3.4.4 Détection de structures polygonales 86
3.5 Conclusion et perspectives 89

II Régularisation de Données Discrètes sur Graphes 93

Résumé et Contributions 95

Introduction 97

4 Régularisation de fonctions sur graphes pondérés 101
4.1 Fonctions et opérateurs sur graphes 102
4.1.1 Espaces des fonctions définies sur un graphe 102
4.1.2 Opérateur différence, dérivée directionnelle et adjoint 103
4.1.3 Opérateur gradient 105
4.1.4 Opérateur Laplacien 107
4.2 Opérateurs proposés : p-Laplaciens sur graphes 110
4.2.1 Le p-Laplacien isotrope 110
4.2.2 Le p-Laplacien anisotrope 112
4.2.3 Expression générale et fonctions localement constantes 112
4.3 Modèles variationnels proposés 114
4.3.1 Fonctionnelles de régularisation discrète 114
4.3.2 Expressions non-locales 115
4.3.3 Modèles variationnels p-TV isotrope et p-TV anisotrope . . . 116
4.4 Processus de diffusion comme solutions de la régularisation 120
4.4.1 Diffusion semi-discrète 120
4.4.2 Diffusion discrète 122
4.4.3 Relations entre la diffusion discrète et le filtrage spectral . . . 124
4.4.4 Cas particulier : diffusion discrète linéaire 125
4.4.5 Critères d'arrêt des processus de diffusion 125
4.5 Cas des fonctions vectorielles 126
4.6 Conclusion et perspectives 128

5 Applications 129
5.1 Applications au traitement d'images 130
5.1.1 Des images aux graphes 130

5.1.2 Comportement des régularisations proposées 132
5.1.3 Débruitage et restauration par régularisation isotrope 138
5.1.4 Simplification par pré-segmentation et régularisation 142
5.2 Application au traitement de courbes et de surfaces polygonales . . . 146
5.2.1 Lissage et débruitage 146
5.2.2 Simplification 149
5.3 Régularisation de données discrètes non-organisées 153
5.4 Conclusion et perspectives 154

Conclusion et Perspectives Générales **157**

Bibliographie **163**

Résumé et Contributions

Dans de nombreux domaines scientifiques, tels que l'infographie, l'imagerie numérique, ou l'analyse de données, les données manipulées sont discrètes. Elles peuvent l'être par nature (mesures, bases de données, réseaux complexes, ...) ou par nécessité de discrétisation. Selon les domaines, ces données peuvent être organisées ou non. Lorsque les données sont organisées (images, maillages), il est nécessaire de les traiter et de les simplifier pour faciliter leur analyse. Ces traitements peuvent aller de la restauration à la segmentation. Dans le cas où les données sont non-organisées, il est indispensable de les structurer avant toute étape de traitement. Il arrive aussi que ces données non-organisées aient besoin d'être préalablement traitées pour que leur structuration soit plus efficace. Les données organisées peuvent également être ré-organisées pour que leur traitement soit facilité. Cette thèse, décomposée en deux parties, traite des problématiques de structuration, de traitement et de simplification de données organisées ou non.

La première partie concerne la structuration de données, représentées par des ensembles de points du plan ou de l'espace Euclidien. Dans ce contexte, nous considérons les problèmes de la reconstruction de courbes planaires et de la détection de formes géométriques 3D connues. Ces deux problèmes sont traités par des techniques de géométrie algorithmique, basées sur le diagramme de Voronoï et la triangulation de Delaunay. Dans le cas de la reconstruction de courbes planaires, après une analyse des méthodes existantes, nous proposons une famille hiérarchique de graphes de voisinage : les β-Crusts locaux. Les graphes de cette famille, sous-graphes du graphe de Gabriel, sont générés à partir d'un voisinage, défini pour les arêtes du diagramme de Voronoï. Un cas particulier de cette famille est le Crust local, largement utilisé dans des applications d'échantillonnage et de reconstruction de contours scannés. Ensuite, à partir de l'analyse des propriétés géométriques et combinatoires des β-Crusts locaux, nous proposons également un algorithme glouton qui permet : d'une part de trouver le bon paramètre β pour que la reconstruction soit une collection de courbes polygonales sans intersection, et d'autre part d'améliorer la robustesse des β-Crusts locaux lorsque. Cet algorithme prend en compte les propriétés topologiques des courbes à reconstruire. À défaut de fournir une condition d'échantillonnage pour ce dernier algorithme, nous menons une étude expérimentale pour valider son efficacité par rapport aux approches de reconstruction similaires. Le protocole utilisé permet de rechercher l'ϵ-échantillon le plus éparse d'une courbe donnée, de sorte qu'un algorithme de reconstruction spécifié reconstruise correctement la courbe.
Dans cette même première partie, nous proposons une méthode de détection de formes géométriques connues, à partir d'un ensemble de points 3D non-organisés, même lorsque

les formes à détecter sont plongées dans un milieu perturbé. Pour cela, nous introduisons une famille de diagrammes affines, générés à partir de boules ellipsoïdales dont la forme et l'orientation sont pré-définies. Les triangulations, qui sont duales à ces diagrammes, ont la propriété de connecter des points qui ne sont pas naturellement voisins dans le diagramme de Voronoï. À partir de ces triangulations, nous définissons une extension des α-formes, que nous appelons α-formes anisotropes. Contrairement aux α-formes, l'extension proposée permet de détecter des structures polygonales linéaires dans une direction donnée. Lorsque les boules ellipsoïdales sont suffisamment fines et allongées, les α-formes anisotropes reconstruisent des portions de droites. Lorsque les boules ellipsoïdales sont suffisamment plates, la reconstruction correspond à des portions de plans. Toutes les structures détectées ne sont pas nécessairement significatives, ou ne possèdent pas les propriétés topologiques adéquates. Pour résoudre ce problème, nous proposons une méthode de filtrage qui supprime les plus petites structures, et qui garantit la reconstruction ne comporte que des structures topologiquement équivalentes à un segment de droite ou à un disque. La détection de structures polygonales plus complexes est réalisée en fusionnant les résultats obtenus pour différentes directions. Ces dernières, ainsi que la forme des boules ellipsoïdales, peuvent être préalablement calculées en fonction du contexte applicatif. Cette méthode générale peut venir en complément de la transformation de Hough, qui n'est pas toujours adaptée pour traiter un ensemble de points non-organisés.

La deuxième partie concerne le problème du traitement et de la simplification de données organisées ou non. Ce problème est traité par des méthodes variationnelles discrètes sur graphes de topologie quelconque. Dans ce contexte, nous proposons un cadre général qui unifie les méthodes basées sur des fonctionnelles discrètes de régularisation locale et non-locale. La principale contribution de ce travail est l'introduction d'une large famille de problèmes de régularisation, basée sur les p-Laplaciens isotrope et anisotrope sur graphes pondérés, analogues discrets des p-Laplaciens Euclidiens isotrope et anisotrope. La résolution de ces problèmes mènent à des processus de diffusion discrète, linéaire ou non-linéaire. Nous montrons que ce formalisme permet de retrouver la version discrète de plusieurs méthodes continues de régularisation locale et non-locale, qui interviennent en traitement d'images dans des problèmes de restauration et de débruitage. Contrairement à ces méthodes, les processus de diffusion ne font pas intervenir d'équations aux dérivées partielles, ce qui permet de s'affranchir des problèmes de discrétisation spatiale. D'autre part, plusieurs méthodes de filtrage de données, issues de la théorie spectrale des graphes, sont des cas particuliers de notre approche. Ces méthodes, qui par définition font intervenir des interactions locales ou non-locales, utilisent généralement le Laplacien isotrope discret ($p = 2$). Le comportement de la régularisation pour $p \neq 2$, est illustré avec plusieurs applications de traitement d'images, de courbes et de surfaces polygonales, ainsi que de données discrètes non-organisées. En particulier, lorsque $p \leq 1$, la régularisation se comporte comme un processus de lissage qui préserve les discontinuités principales, et pour $p \to 0$ elle agit comme un processus de regroupement. Ainsi, la régularisation proposée offre une extension naturelle pour traiter des données, ou des fonctions discrètes, définies sur des domaines réguliers ou irréguliers, ou dans des espaces de grande dimension.

Première partie

Reconstruction et Détection de Formes

Introduction

Dans de nombreuses applications, les données sont représentées par des ensembles finis de points non-organisés, dans un espace Euclidien de dimension n. Ici, le terme non-organisé signifie que seules les coordonnées des points sont connues. Afin de traiter ou d'analyser les données, les ensembles de points nécessitent d'être préalablement structurés. Nous considérons dans cette partie les problèmes de la reconstruction et de la détection de formes à partir d'un ensemble de points. Ces problèmes sont par exemple rencontrés en reconnaissance de formes, en analyse de scènes, en navigation virtuelle, ou encore en classification de données. Dans cette partie, décomposée en trois chapitres, nous traitons les problèmes de la reconstruction de courbes planaires et de la détection de formes géométriques 3D connues. Ces deux problèmes sont traités par des techniques de géométrie algorithmique, basées sur le diagramme de Voronoï et la triangulation de Delaunay. Ces concepts généraux sont rappelés dans un premier chapitre.

Le deuxième chapitre est consacré à la reconstruction de courbes planaires. A la suite d'une analyse des méthodes existantes, basées sur le diagramme de Voronoï et la triangulation de Delaunay, nous proposons une famille hiérarchique de graphes de voisinage : les *β-Crusts Locaux*. Les graphes de cette famille, sous-graphes du graphe de Gabriel, sont générés à partir d'un voisinage, défini sur les arêtes du diagramme de Voronoï. Un cas particulier de cette famille est le *Crust Local*, largement utilisé dans des applications d'échantillonnage et de reconstruction de contours scannés. Ensuite, à partir de l'analyse des propriétés géométriques et combinatoires des *β-Crusts Locaux*, nous proposons également un algorithme glouton qui permet : d'une part de trouver le bon paramètre β pour que la reconstruction soit une collection de courbes polygonales sans intersection, et d'autre part d'améliorer la robustesse des *β-Crusts Locaux*. Cet algorithme prend en compte les propriétés topologiques des courbes à reconstruire. A défaut de fournir une condition d'échantillonnage pour ce dernier algorithme, nous menons une étude expérimentale pour valider son efficacité par rapport aux approches similaires. Le protocole utilisé permet de rechercher l'ϵ-échantillon le plus éparse, d'une courbe donnée, tel qu'un algorithme spécifié reconstruise correctement la courbe.

Dans le troisième chapitre, nous proposons une méthode de détection de formes géométriques connues, à partir d'un ensemble de points 3D, même lorsque les formes à détecter sont plongées dans un milieu perturbé. Nous nous restreignons aux cas des structures linéaires et planaires. La détection est basée sur une famille de diagrammes affines, générés à partir de boules ellipsoïdales, dont la forme et l'orientation sont pré-

définies. Les triangulations, duales de ces diagrammes, ont la propriété de connecter des points qui ne sont pas naturellement voisins dans le diagramme de Voronoï. A partir de ces triangulations, nous définissons une extension du concept des α-formes. Contrairement aux α-formes classiques, la famille que nous proposons est capable de détecter des structures fines dans une direction donnée. La détection de structures polygonales plus complexes est réalisée en fusionnant le résultat obtenu pour différentes orientations.

Chapitre 1

Concepts géométriques liés à la reconstruction et à la détection

« Soit donné un système quelconque de points dans l'espace à n dimensions. Je me propose de considérer une sphère se mouvant entre les points de ce système se rétrécissant et se dilatant à volonté et assujettie à la seule condition d'être vide, c'est à dire de ne pas contenir en son intérieur des points de ce système. C'est la méthode de la sphère vide ... », [Delaunay, 1934].

Ce chapitre pose les problèmes que nous considérons dans cette première partie, c'est à dire : la reconstruction de courbes polygonales à partir d'un ensemble de points du plan, et la détection de primitives géométriques à partir d'un ensemble de points de l'espace de dimension trois. Les méthodes, que nous proposons pour traiter ces deux problèmes, sont basées des structures géométriques et combinatoires qui organisent naturellement l'ensemble de points. Ces structures sont le diagramme de Voronoï et son dual, la triangulation de Delaunay. Ce chapitre rappelle également la définition et les propriétés principales de ces deux structures, ainsi que les autres concepts généraux que nous utiliserons par la suite.

1.1 Notion de forme d'un ensemble de points

De façon générale, une forme Γ désigne un sous-ensemble de $\mathbb{R}^n$ dont on peut décrire l'aspect géométrique. Soit $P := \{p_1, \ldots, p_N\}$ un échantillon de la forme Γ, c.-à-d. un ensemble fini de points inclus dans Γ. La **reconstruction** de la forme Γ, à partir de l'échantillon P, consiste à trouver une forme Γ' qui approxime Γ, tout en passant par les points de P (interpolation). Lorsque la forme Γ et la reconstruction sont topologiquement équivalentes, la reconstruction est dite **correcte**. Dans le cas contraire, elle est **incorrecte**. La difficulté de cette opération provient du fait que la forme initiale Γ n'est généralement pas explicitement connue. Ainsi, les caractéristiques géométriques intrinsèques à la forme Γ ne peuvent pas être directement exploitées lors de la reconstruction. En revanche, pour des classes particulières de formes, ces caractéristiques peuvent être déduites de l'ensemble de points, si ce dernier représente un échantillon suffisamment dense de Γ. En particulier, de nombreuses méthodes de reconstruction ont été proposées lorsque l'ensemble P est un échantillon de courbes ou de surfaces de $\mathbb{R}^2$ ou de $\mathbb{R}^3$. Une classe importante de ces méthodes consiste à interpoler l'ensemble P, ou une partie de P, par des éléments finis (segments, triangles, polygones, ...), voir la figure 1.1.

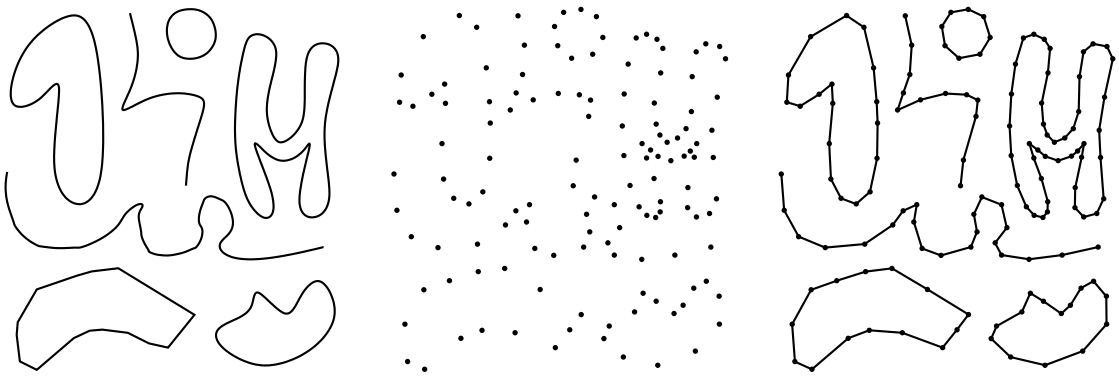


Fig. 1.1: Reconstruction polygonale de courbes à partir d'un échantillon.

Parmi ces méthodes de reconstruction de courbes ou de surfaces, nous considérons celles basées sur la triangulation de Delaunay et le diagramme de Voronoï. Ces méthodes ont montré leur efficacité autant en théorie qu'en pratique dans de nombreuses situations [Dey, 2007]. Ceci est principalement dû aux propriétés géométriques que renferme le diagramme de Voronoï, relativement à celles des courbes ou des surfaces à reconstruire. Bien que ces méthodes soient robustes dans la plupart des configurations, elles rencontrent des difficultés pour reconstruire des formes satisfaisantes lorsque l'ensemble de points est de faible densité, ou lorsque les formes à reconstruire sont plongées dans un milieu perturbé (de nombreux points n'appartiennent pas aux formes à reconstruire). C'est à cette problématique que nous essayons de répondre dans cette partie.

1.2 Enveloppe convexe, polytopes et simplexes

Un ensemble fini $P \in \mathbb{R}^n$ de points est en **position générale** si tout sous-ensemble $P' \subseteq P$, tel que $|P'| = n + 1$, est un ensemble de points indépendants[1]. En dimension 3, c'est équivalent à dire qu'il n'existe pas quatre points coplanaires, ni cinq points cosphériques. Afin d'obtenir des propriétés particulières sur les maillages engendrés par un ensemble de points, et de les utiliser dans des algorithmes basés sur des structures de données efficaces, les ensembles de points sont supposés être en position générale.

Un sous-ensemble S de $\mathbb{R}^n$ est **convexe** si pour toute paire de points p et q de S, le segment $[p, q]$ est entièrement inclus dans S. L'**enveloppe convexe** d'un ensemble fini de points $P \subset \mathbb{R}^n$, notée $conv(P)$, est le plus petit ensemble convexe contenant P. L'enveloppe convexe d'un ensemble fini de points de $\mathbb{R}^n$ est appelée **k-polytope**, où k est sa dimension.

Soit $T \subset \mathbb{R}^n$ un ensemble de $k + 1$ points linéairement indépendants, $0 \leq k < n$. L'enveloppe convexe de T est un polytope particulier appelé **k-simplexe**, et noté $s(T)$. Les k-simplexes de $\mathbb{R}^2$ et $\mathbb{R}^3$ sont illustrés à la figure 1.2. Un 0-simplexe est un point, un 1-simplexe un segment, un 2-simplexe un triangle, et un 3-simplexe un tétraèdre. L'hypothèse sur la position générale d'un ensemble de points $P \subset \mathbb{R}^n$ assure que l'enveloppe convexe des sous-ensembles $T \subseteq P$, tel que $|T| = n + 1$, soit un n-simplexe.

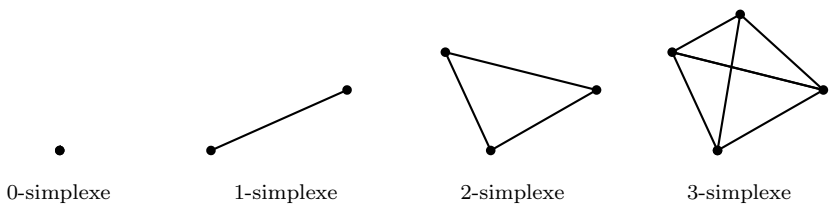


Fig. 1.2: *k-simplexes de $\mathbb{R}^3$.*

Tout hyperplan $\mathcal{H}$ divise l'espace $\mathbb{R}^n$ en deux demi-espaces situés de part et d'autre de $\mathcal{H}$. Soient $\mathcal{H}^+$ et $\mathcal{H}^-$ les deux demi-espaces ouverts limités par $\mathcal{H}$ et, $\overline{\mathcal{H}^+} = \mathcal{H}^+ \cup \mathcal{H}$ et $\overline{\mathcal{H}^-} = \mathcal{H}^- \cup \mathcal{H}$ leurs adhérences respectives. $\mathcal{H}$ est un **hyperplan support** d'un n-polytope s si $\mathcal{H} \cap s \neq \emptyset$ et si $s \subset \overline{\mathcal{H}^+}$ ou $s \subset \overline{\mathcal{H}^-}$. L'intersection $\mathcal{H} \cap s$ est une **face** du polytope. Les faces sont des sous-ensembles convexes de $\mathbb{R}^n$, de dimensions comprises entre 0 et $n-1$. La face de dimension n est le polytope lui-même. Une face de dimension k est appelée une k-face. Le bord d'un k-polytope s, noté ∂s, est constitué de l'ensemble de ses $(k-1)$-faces.

Propriété 1 *Tout polytope a un nombre fini de faces et toute face d'un polytope est un polytope. Soit s un n-polytope.*

[1] $k+1$ points de $\mathbb{R}^n$, $0 \leq k \leq n$, sont indépendants s'ils engendrent, par combinaisons linéaires affines, un espace de dimension k.

1. *L'intersection d'une famille de faces de s est soit une face de s, soit vide.*
2. *Toute $(n-2)$-face de s est l'intersection de deux $(n-1)$-faces de s.*
3. *$\forall (l,k)_{0 \leq l \leq k < n}$, une l-face de s est l'intersection de toutes les k-faces de s qui la contiennent.*

Deux faces d'un polytope sont dites **incidentes** si l'une est incluse dans l'autre et si leurs dimensions diffèrent d'une unité. Deux k-faces sont **adjacentes** (ou voisines) si elles sont incidentes à la même $(k-1)$-face, pour $0 < k \leq n$. Deux sommets (0-faces) sont adjacents s'ils sont incidents à la même arête (1-face).

1.3 Complexes cellulaires et simpliciaux

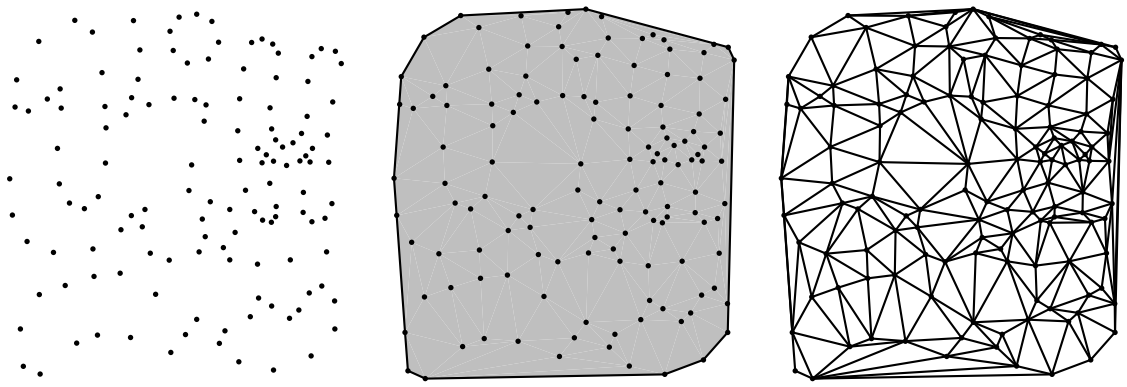
La notion de complexe généralise celle de polytope. Un **complexe cellulaire** $\mathcal{C}$ est un ensemble de polytopes qui vérifient les deux conditions suivantes :

1. $\forall T' \subseteq T,\ s(T) \in \mathcal{C} \Rightarrow s(T') \in \mathcal{C}$,
2. $s(T), s(T') \in \mathcal{C} \Rightarrow s(T) \cap s(T') = \emptyset$, ou $s(T) \cap s(T') = s(T \cap T') = conv(T \cap T')$.

où $s(T) = conv(T)$ et $s(T') = conv(T')$ sont deux polytopes. Si la dimension maximale des polytopes, qui composent le complexe $\mathcal{C}$, est k, alors $\mathcal{C}$ est un k-complexe. Les polytopes qui constituent un complexe sont ses **faces.** Elles vérifient les mêmes propriétés que celles d'un polytope (Propriété 1). Les n-faces sont appelées ***cellules***. Le **domaine de recouvrement** d'un complexe $\mathcal{C}$, noté $|\mathcal{C}| = \bigcup_{s(T) \in \mathcal{C}} s(T)$, est la partie de $\mathbb{R}^n$ couverte par $\mathcal{C}$.

Un **complexe simplicial** est un complexe cellulaire formé uniquement de simplexes. Un complexe simplicial $\mathcal{C}$ est une **triangulation** d'un ensemble fini de points $P \subset \mathbb{R}^n$, si chaque sommet de $\mathcal{C}$ est un point de P et si $|\mathcal{C}| = conv(P)$. C'est une partition de l'enveloppe convexe de P en n-simplexes dont les sommets appartiennent à P. L'enveloppe convexe et une triangulation d'un ensemble de points de $\mathbb{R}^2$ sont illustrés à la figure 1.3. Notons que les $(n-1)$-simplexes qui appartiennent à $\partial conv(P)$ sont incidents à un seul n-simplexe, et que les autres $(n-1)$-simplexes sont incidents à exactement deux n-simplexes. Ces derniers sont dit intérieurs à la triangulation.

Le traitement et l'analyse d'un ensemble fini de points $P \subset \mathbb{R}^n$ utilise souvent des complexes, construits à partir de P. Les polytopes de ces complexes sont générés à partir de critères géométriques. Dans la suite, nous rappelons les définitions et les propriétés des complexes utilisés dans cette partie, tous générés à partir de boules Euclidiennes. Des complexes similaires, définis à partir de boules ellipsoïdales, sont proposés dans le chapitre 3.



P $conv(P)$ une triangulation de P

Fig. 1.3: L'enveloppe convexe et une triangulation d'un ensemble de points.

1.4 Triangulation de Delaunay

Parmi les triangulations possibles d'un ensemble fini de points $P \subset \mathbb{R}^n$, la triangulation de Delaunay [Delaunay, 1934] décrit l'ensemble des sphères qui passent par des points de P et qui ne contiennent aucun point de P.

Soit $s(T)$ un k-simplexe tel que $T \subset P$ et $0 \leq k \leq n$. Soit $b(T)$ une boule ouverte dont le bord $\partial b(T)$ est circonscrit aux points de T. La **boule** $b(T)$ est **vide** si elle ne contient aucun point de P. La **triangulation de Delaunay** de P, notée $Del(P)$, est le n-complexe simplicial tel que chaque k-simplexe soit circonscrit à une boule ouverte vide, quelque soit $0 \leq k \leq n$:

$$s(T) \in Del(P) \Leftrightarrow \exists b(T) \ : \ b(T) \cap P = \emptyset.$$

Un exemple de triangulation de Delaunay d'un ensemble de points de $\mathbb{R}^2$ est illustré à la figure 1.4. Lorsque l'ensemble de points est en position générale, sa triangulation est unique. Dans le cas contraire, il existe différentes méthodes pour traiter les cas dégénérés. Ces méthodes sont généralement dépendantes de celles employées pour calculer la triangulation elle-même. Les sous-complexes de la triangulation de Delaunay que nous décrivons par la suite sont indépendants de ces méthodes. Pour le détail des algorithmes et des structures de données, on peut par exemple se référer à [Mücke, 1993, Overmars *et al.*, 2000, Edelsbrunner, 2001]. En théorie comme en pratique, l'espérance du temps d'exécution du calcul de $Del(P)$ est $O(N \log N + N^{\lceil \frac{n}{2} \rceil})$, où $|P| = N$. L'espérance de la place mémoire requise est $O(N^{\lceil \frac{n}{2} \rceil})$.

1.5 Diagramme de Voronoï et dualité

Le diagramme de Voronoï [2] d'un ensemble de points $P \subset \mathbb{R}^n$ [Voronoï, 1907, Dirichlet, 1850] est une partition de $\mathbb{R}^n$ en n-polytopes, générés à partir des points de P et de

[2] Également connu sous le nom de tesselation de Dirichlet.

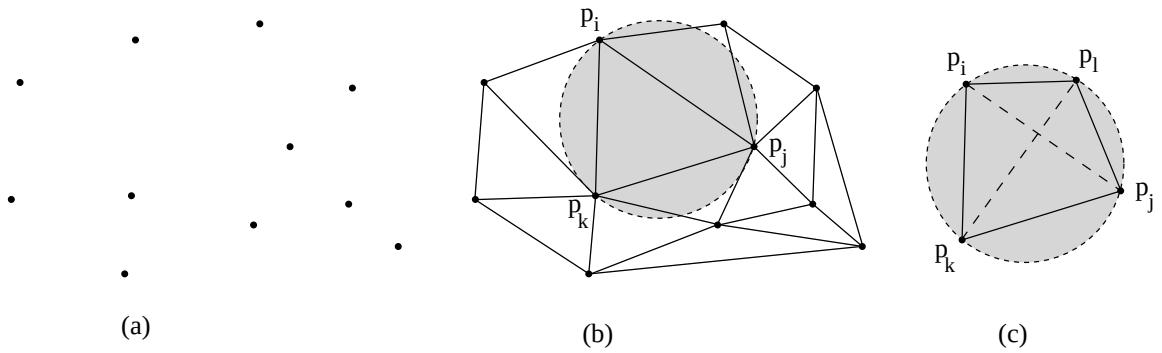


Fig. 1.4: (a) Un ensemble de points P. (b) La triangulation de Delaunay de P. (c) Cas dégénéré de quatre points cocycliques.

la distance Euclidienne. Le **polytope de Voronoï** d'un point $p_i \in P$ (ou **cellule**), noté $Vor(p_i, P)$, est l'ensemble de points $p \in \mathbb{R}^n$, situés plus proche de p_i que des autres points de P, selon la distance Euclidienne :

$$Vor(p_i, P) = \{p \in \mathbb{R}^n \ : \ \|p - p_i\|_2 \leq \|p - p_j\|_2, \forall p_j \in P\},$$

où $\|.\|_2$ est la norme Euclidienne. Alternativement, la cellule de Voronoï du point p_i est l'ensemble des centres $p \in \mathbb{R}^n$ des boules ouvertes $b_p(p_i)$ vides et telles que $p_i \in \partial b_p(p_i)$:

$$Vor(p_i, P) = \{p \in \mathbb{R}^n \ : \ b_p(p_i) \cap P = \emptyset\}.$$

Le **diagramme de Voronoï** de l'ensemble P, noté $Vor(P)$, est le complexe cellulaire formé de l'ensemble des cellule de Voronoï, ainsi que de leurs faces. Parmi les cellules, celles associées aux points de l'enveloppe convexe de P sont non-bornées. C'est également le cas des k-faces de $Vor(P)$ qui sont communes à des cellules non-bornées, pour $1 \leq k < n$. Un exemple de diagramme de Voronoï de points de $\mathbb{R}^2$ est illustré à la figure 1.5. Les cellules bornées sont représentées en gris, et les 1-simplexes (arêtes) non-bornées par des vecteurs.

Il existe une correspondance naturelle entre les k-simplexes de $Del(P)$ et les $(n-k)$-faces de $Vor(P)$, qui permet d'établir la dualité entre les deux complexes (voir la figure 1.5(b)).

Propriété 2 (dualité) *Soit $s(T)$ un k-simplexe de $Del(P)$, avec $0 \leq k \leq n$. Soit $Vor(T, P) = \bigcap_{p \in T} Vor(p, P)$ la face de $Vor(P)$ commune aux cellules des points de T. Alors, nous avons la relation de dualité suivante :*

$$s(T) \in Del(P) \Leftrightarrow Vor(T, P) \text{ est une } (n-k)\text{-face de } Vor(P), \ \forall 0 \leq k \leq n.$$

Le k-simplexe $s(T)$ est dit dual à la $(n-k)$-face de $Vor(T, P)$, et réciproquement : $s(T) = dual(Vor(T, P))$ et $Vor(T, P) = dual(s(T))$.

De plus, notons que les faces non-bornées de $Vor(P)$ sont duales des simplexes de $Del(P)$ qui appartiennent à l'enveloppe convexe de P. La Propriété 2 permet de représenter le diagramme de Voronoï à partir de la triangulation de Delaunay. Pour le détail du

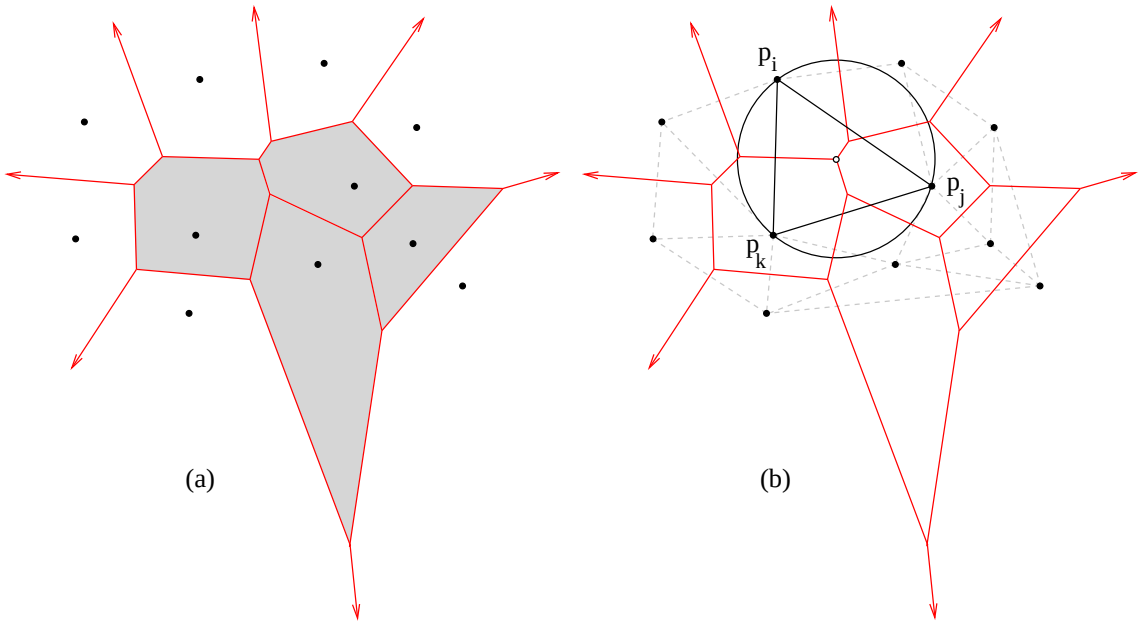


Fig. 1.5: (a) Diagramme de Voronoï d'un ensemble de points, les cellules grisées sont bornées. (b) Relation de dualité entre le diagramme de Voronoï et la triangulation de Delaunay.

diagramme de Voronoï, on peut par exemple consulter [Okabe *et al.*, 2000, Aurenhammer et Klein, 2000].

Chapitre 2

Reconstruction de courbes polygonales simples

« Toutes les formes vont naître du point qui est la forme première, un point sur un plan étant l'image première de toute expression picturale. La naissance des formes se fait tout simplement dès le moment où le point sort de ses limites. Sa tension, dont la nature est concentrique, peut prendre une direction déterminée et alors naît la ligne. Point et ligne sont les deux éléments de base nécessaires à toute composition picturale, ... », Philippe Sers, 1991, préface de Point et Ligne sur le Plan, W. Kandinsky, éditions Gallimard.

Ce chapitre concerne la reconstruction polygonales de courbes sans intersection, à partir d'un ensemble de points du plan Euclidien. Dans ce contexte, nous analysons les méthodes de reconstruction basées sur les graphes de voisinage et la triangulation de Delaunay. En particulier, nous montrons que le Crust local est l'un des graphes les plus efficaces, du point de vue de la complexité de son calcul et de sa capacité de reconstruction. Néanmoins, ce graphe, ainsi que les autres graphes étudiés, rencontrent des difficultés pour reconstruire des courbes à partir d'échantillons de faible densité, ou de densité très variable. Pour cela, nous proposons d'étendre les capacités du Crust local en introduisant une famille hiérarchique de graphes de voisinage, que nous appelons les β-Crusts locaux. Ces graphes, sous-graphes du graphe de Gabriel, sont générés à partir d'un voisinage défini pour les arêtes du diagramme de Voronoï, dont la taille est paramétrée par un réel $\beta > 0$. Ensuite, à partir de l'analyse des propriétés géométriques et combinatoires des β-Crusts locaux, nous proposons un algorithme glouton en $O(N \log N)$, qui permet : d'une part de trouver le bon paramètre β pour que la reconstruction soit une collection de courbes polygonales sans intersection, et d'autre part d'améliorer la robustesse des β-Crusts locaux. Cet algorithme prend en compte les propriétés topologiques des courbes à reconstruire. À défaut de fournir une condition d'échantillonnage pour ce dernier algorithme, nous menons une étude expérimentale pour montrer son efficacité relativement aux approches concurrentes.

2.1 Introduction

Plusieurs méthodes ont été proposées pour résoudre le problème de la reconstruction polygonale de courbes à partir d'un ensemble P de points du plan Euclidien. Parmi elles, les méthodes basées sur les graphes de voisinage, et plus particulièrement sur la triangulation de Delaunay, fournissent une interpolation adéquate autant en pratique qu'en théorie, si l'ensemble P est un échantillon suffisamment dense de courbes. Les graphes de voisinage connectent des points de l'ensemble P si ces points sont voisins relativement à une mesure définie à partir de P. Le voisinage, qui permet de générer les arêtes, peut être interprété comme l'élément structurant des graphes. Les premiers travaux sur la reconstruction de formes, basés sur des graphes de voisinage, sont issus des méthodes de regroupement et de description de nuages de points. Ces méthodes interviennent dans de nombreuses applications de reconnaissance et d'analyse de formes à partir d'ensembles de points (voir par exemple [O'Rourke et Toussaint, 2004] et les références qui y sont associées). Les principaux graphes utilisés par ces méthodes sont la triangulation de Delaunay, le graphe de Gabriel, le graphe des k-plus proches voisins (k-NN), l'arbre de recouvrement minimum (EMST) ou encore le graphe de voisinage relatif (RNG).

Une catégorie importante de ces graphes considère le voisinage des arêtes comme l'union et/ou l'intersection de disques. L'arête appartient alors au graphe si son voisinage, ou son intérieur, ne contient aucun point de l'ensemble P. Dans ce cas, le voisinage est dit vide. Parmi ces graphes, on peut distinguer ceux définis avec un paramètre, permettant ainsi de décrire l'ensemble P à plusieurs niveaux de détails. Les α-formes [Edelsbrunner *et al.*, 1983] sont générées avec des disques vides de rayon α. Les β-squelettes [Kirkpatrick et Radke, 1985] sont formés avec deux disques vides de même rayon. Des extensions de ces deux descripteurs ont été proposées en utilisant les γ-graphes de voisinage [Veltkamp, 1988]. Étant donné que la triangulation de Delaunay est générée avec les disques vides de rayon maximal, elle inclut la plupart des graphes décrits ci-dessus. Tandis que le voisinage de ces graphes est défini en utilisant un paramètre global, celui des A-formes [Melkemi et Melkemi, 1995, Melkemi, 1997], est défini à partir d'un ensemble A de points de contrôle. Les arêtes de cette famille de graphes sont générées par des disques qui ne contiennent aucun point de l'ensemble $P \cup A$. Les A-formes sont également des sous-graphes de la triangulation de Delaunay. Le choix des points de l'ensemble A est lié à l'axe médian des formes à reconstruire. Comme l'axe médian peut être approximé par un sous-ensemble des arêtes du diagramme de Voronoï [Brandt et Algazi, 1992, Brandt, 1994, Attali et Montanvert, 1997], les sommets de Voronoï sont de bons candidats pour l'ensemble A.

Indépendamment, une notion similaire aux A-formes a été proposée pour définir le Crust[3] de l'ensemble P [Amenta *et al.*, 1998], dans le contexte spécifique de la reconstruction de courbes fermées sans intersection. Le Crust est un cas particulier des A-formes, où l'ensemble A correspond à l'ensemble de tous les sommets du diagramme de Voronoï de P. De plus, il a été démontré que le Crust fournit toujours une recons-

[3]Nous choisissons de conserver le terme anglais initial Crust plutôt que le terme français *croûte*, ou *écorce*, pour un soucis de généricité.

truction topologiquement équivalente aux courbes échantillonnées, si l'ensemble P est suffisamment dense et si les courbes sont deux fois différentiables [Amenta *et al.*, 1998]. La garantie de reconstruction exploite le lien entre les courbes et leur axe médian, à travers une condition d'échantillonnage qui permet de répartir non-uniformément les points échantillonnés sur les courbes. Le calcul du CRUST nécessite d'utiliser le diagramme de Voronoï de P, ainsi que la triangulation de Delaunay de l'ensemble $P \cup A$. Afin d'améliorer la complexité de ce calcul, une version locale du CRUST, appelée CRUST local, a été proposée dans [Gold, 1999], et préalablement étudiée sous une autre forme dans [Attali, 1997]. La construction de ce graphe nécessite seulement de calculer la triangulation de Delaunay de P, et fournit des résultats équivalents à ceux obtenus avec le CRUST. De plus, les arêtes du diagramme de Voronoï, dont les arêtes de Delaunay qui leurs sont duales n'appartiennent pas au CRUST local, permettent d'approximer l'axe médian des courbes reconstruites. Ce sous-graphe du diagramme de Voronoï est appelé l'anti-CRUST de P [Gold, 1999].

D'autres méthodes de reconstruction de courbes utilisent des graphes de voisinage. C'est le cas des méthodes utilisant les α-formes [Bernardini et Bajaj, 1997], les β-squelettes [Amenta *et al.*, 1998], l'arbre de recouvrement minimum [Figueiredo et Gomez, 1995], ou encore le graphe de voisinage relatif [Mukhopadhay et Das, 2006]. Dans [Dey et Kumar, 1999], le graphe des plus proches voisins est utilisé pour proposer une alternative au CRUST et au CRUST local, nommée NNCRUST (CRUST des plus proches voisins). Comme toutes les méthodes de reconstruction précédentes ne permettent pas de garantir la reconstruction à partir d'échantillons de courbes ouvertes et/ou qui ne sont pas deux fois différentiables, les idées proposées dans [Dey et Kumar, 1999] ont été étendues en incorporant des paramètres liés à la densité locale et à l'acuité [Dey *et al.*, 2000, Dey et Wenger, 2001]. Le cas des courbes avec intersections a également été proposé dans [Lenz, 2006] en utilisant des idées similaires. Dans [Giesen, 2000, Althaus et Mehlhorn, 2000], la reconstruction est effectuée par le tour du voyageur de commerce (TSP) de l'ensemble de points, qui est également un sous-graphe de la triangulation de Delaunay. Le principal désavantage de cette dernière approche est qu'elle ne permet de prendre en compte qu'une courbe constituée d'une seule composante connexe.

L'objectif principal de ce chapitre est de décrire une famille hiérarchique de graphes dont le voisinage est défini sur les arêtes du diagramme de Voronoï. Notre travail s'inspire de deux propriétés du CRUST local et de l'anti-CRUST que nous démontrons. Premièrement, le CRUST local est un sous-graphe du graphe de Gabriel. Deuxièmement, les arêtes de l'anti-CRUST sont générées en utilisant des disques de Gabriel. Basés sur ces propriétés, nous étendons le voisinage de Gabriel des arêtes de Voronoï au β-voisinage, initialement défini pour les β-squelettes [Kirkpatrick et Radke, 1985]. Lorsque β est fixé, ce voisinage permet de définir deux types de graphes, l'un sous-graphe du diagramme de Voronoï, et l'autre sous-graphe du graphe de Gabriel. Nous appelons respectivement ces graphes le β-axe médian et le β-CRUST local. Nous montrons ensuite que les propriétés d'inclusion du β-voisinage permettent de concevoir un algorithme, efficace et sans paramètre, pour extraire des courbes sans intersection à partir de la triangulation de Delaunay. Nous appelons cet algorithme TOPOCRUST. Les graphes obtenus sont des collections de courbes

polygonales fermées ou ouvertes. Lorsque l'ensemble de points est de faible densité, nous montrons expérimentalement que notre approche améliore les résultats obtenus avec le CRUST local, le NNCRUST, et d'autres méthodes de reconstruction basées sur des approches similaires.

Dans la suite, nous rappelons les concepts associés au problème de la reconstruction polygonale de courbes, et nous analysons les approches existantes qui sont liées à l'approche proposée. Puis, nous définissons et analysons les β-CRUSTS locaux et les β-axes médians. À l'issue de cette analyse, nous détaillons l'algorithme TOPOCRUST.

2.2 Notions liées au problème de la reconstruction

Dans cette section, nous rappelons les concepts liés au problème de la reconstruction et de l'échantillonnage de courbes.

2.2.1 Courbes considérées

Afin d'unifier les notations, nous représentons les courbes planaires sous forme paramétrique. Une courbe Γ de $\mathbb{R}^2$ est alors identifiée par une application continue injective $\gamma : [0,1] \to \mathbb{R}^2$ telle que $\gamma([0,1]) = \Gamma$. De plus, Γ est **fermée** si $\gamma(0) = \gamma(1)$, et **ouverte** sinon. Une courbe ouverte Γ est **simple** si γ est surjective sur $[0,1]$ (ou sur $[0,1[$ si Γ est fermée). Dans le cas contraire, Γ comporte des points multiples.

Rappelons qu'une application est dite de **classe** $\mathcal{C}^k$ lorsqu'elle est k fois dérivable et que sa dérivée $k^{\text{ième}}$ est continue. Une courbe Γ est **régulière** si γ est de classe $\mathcal{C}^1$ et si pour tout $t \in [0,1]$, sa dérivée est non nulle. C'est à dire si en tout point de Γ, les tangentes à droite et à gauche existent et coïncident. Lorsque les tangentes ne coïncident pas en un nombre fini de points, Γ est **semi-régulière**. Dans la suite, nous considérons des courbes simples ouvertes ou fermées, régulières ou semi-régulières.

Deux points $p = \gamma(t_1)$ et $q = \gamma(t_2)$, d'une courbe ouverte simple Γ, définissent un unique **arc de courbe** $\Gamma(pq) = \{\gamma(t) : t \in [t_1, t_2]\}$. Lorsque Γ est fermée, il existe deux arcs de courbe d'extrémités p et q. Dans ce cas, nous considérons celui de longueur minimale. Dans les processus d'échantillonnage puis de reconstruction, ce sont les arcs de courbe que l'on cherche à approximer.

Un **échantillon** P d'une courbe Γ est un sous-ensemble de points inclus dans Γ. Soit Γ une collection de courbes de $\mathbb{R}^2$. Soit $P := \{p_1, \ldots, p_N\}$ un échantillon quelconque d'une courbe Γ de $\mathbb{R}^2$. Soit $\gamma : [0,1] \to \mathbb{R}^2$ la fonction qui associe à la variable $t \in [0,1]$ un point de Γ. Nous supposons qu'il existe au moins trois points de P sur γ ($N = 3$), et que les points de P sont ordonnés tels que $0 \leq \gamma^{-1}(p_1) < \ldots < \gamma^{-1}(p_N) \leq 1$. Ainsi, Γ peut s'écrire comme l'union des segments de courbe ayant pour extrémités les points de P : $\Gamma = \Gamma(p_1p_2) \cup \ldots \cup \Gamma(p_Np_1)$.

2.2.2 Squelette et axe médian de courbes

Les notions d'axe médian et de squelette ont été initialement définies pour représenter et décrire la forme d'objets en analyse d'images et en reconnaissance de formes [Blum, 1967]. Leurs définitions varient selon le contexte. Nous adoptons celles communément utilisées en reconstruction de courbes et de surfaces [Amenta *et al.*, 1998, Dey, 2007].

Définitions. Soit Γ une collection de courbes plongées dans $\mathbb{R}^2$. Le **squelette** de Γ, noté $\mathit{Squel}(\Gamma)$, est l'ensemble des centres des disques maximaux. Un disque fermé B est dit **maximal** si tout disque fermé, qui ne contient aucun point de Γ, et qui contient B est égal à B. Soit $M_0(\Gamma)$ l'ensemble des points de $\mathbb{R}^2 \setminus \Gamma$ qui sont à égale distance

d'au moins deux points de Γ. L'**axe médian** de Γ, noté $M(\Gamma)$, est la fermeture de l'ensemble $M_0(\Gamma)$ [4]. Nous avons alors la relation d'inclusion suivante :

$$M_0(\Gamma) \subseteq \mathit{Squel}(\Gamma) \subseteq M(\Gamma).$$

Comme illustré à la figure 2.1, chaque point de l'axe médian est le centre d'un disque qui intersecte Γ tangentiellement. On peut distinguer les points qui sont situés à égale distance d'exactement deux points de Γ (par exemple le point m). Ils constituent les branches de l'axe médian. Les points qui sont situés à égale distance de plus de deux points de Γ correspondent à l'intersection des branches (par exemple le point v). Ces deux types de points proviennent de l'ensemble $M_0(\Gamma)$. Enfin, les points terminaux de l'axe médian (par exemple les points u et w), sont ceux obtenus par la fermeture de $M_0(\Gamma)$.

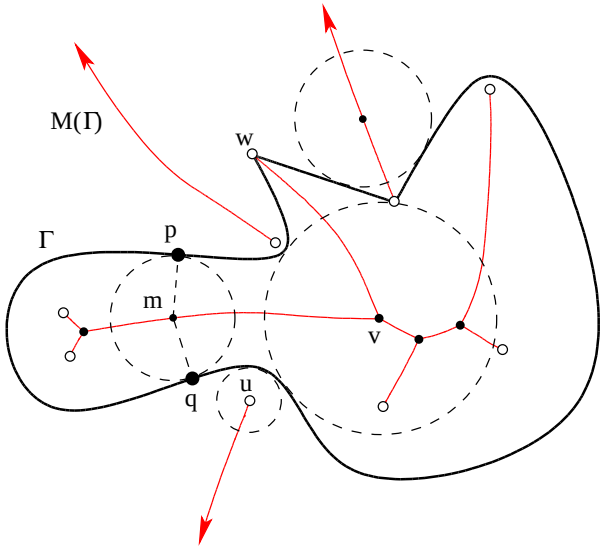


Fig. 2.1: Une courbe de $\mathbb{R}^2$ (traits noirs épais), et son axe médian (traits rouges fins).

Propriétés relatives à la reconstruction. Il existe de nombreuses courbes pour lesquelles le squelette et l'axe médian coïncident. En particulier, c'est le cas des courbes qui sont au moins de classe $\mathcal{C}^2$. Ces dernières courbes ont la particularité de ne jamais intersecter leur axe médian. C'est l'une des propriétés exploitées par les méthodes de reconstruction basées sur le diagramme de Voronoï. En effet, l'ensemble $M_0(\Gamma)$ peut être interprété comme le graphe de Voronoï d'un échantillon infiniment dense de Γ. Ainsi, le squelette et l'axe médian peuvent être approximés par un sous-graphe du graphe de Voronoï d'un échantillon P de Γ [Brandt et Algazi, 1992, Boissonnat et Geiger, 1992, Schmitt et Mattioli, 1993, Attali et Montanvert, 1997]. Par dualité, Γ peut être approximé par un sous-graphe du graphe de Delaunay de P.

En particulier, le squelette de Γ peut être approximé par les faces de $\mathit{Vor}(P)$ (sommets

[4] L'axe médian est parfois défini comme étant l'ensemble M_0.

et arêtes) qui n'intersectent pas Γ [Brandt et Algazi, 1992] :

$$Squel(\Gamma, P) = \bigcup_{F \in Vor(P), F \cap \Gamma = \emptyset} F.$$

Il a été démontré que lorsque Γ représente le bord d'une forme r-régulière[5], et que P est un échantillon infiniment dense de Γ, l'ensemble $Squel(\Gamma, P)$ converge vers le squelette de Γ [Brandt, 1994]. La figure 2.2 illustre cette propriété.

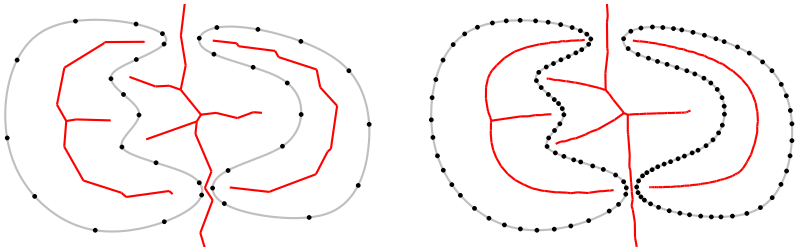


Fig. 2.2: Approximation du squelette d'une collection de courbes (traits rouges).

2.2.3 L'ϵ-échantillonnage

Dans [Amenta *et al.*, 1998] a été proposé d'utiliser l'axe médian pour définir une condition d'échantillonnage non-uniforme. Cette condition est basée sur la fonction $\varrho : \Gamma \to \mathbb{R}$ qui mesure en tout point $p \in \Gamma$ la distance Euclidienne minimale de p à l'axe médian de Γ :

$$\varrho(p) = \min_{m \in M(\Gamma)} \|p - m\|_2. \tag{2.1}$$

La valeur de la fonction ϱ, en un point p de Γ, correspond au rayon du disque maximal centré en p et dont l'intérieur est vide de points de $M(\Gamma)$. Comme illustré à la figure 2.3, la fonction ϱ quantifie localement le niveau de détails de Γ : $\varrho(p)$ est d'autant plus faible que la courbure de Γ en p est importante ou que p est proche d'autres parties de Γ. De plus, on peut remarquer que si p est un point non-régulier (points u et v), alors $\varrho(p) = 0$. Ceci provient du fait que ces points sont à la fois sur Γ et sur son axe médian.

À partir de la fonction ϱ, Amenta *et al* ont proposé d'échantillonner Γ de telle sorte que pour chaque point $p \in \Gamma$, le disque de rayon $\varrho(p)$ inclue un point de l'échantillon. Un ensemble fini de points $P \subset \Gamma$ est un **ϵ-échantillon** de Γ, $\epsilon \in]0, 1]$, si et seulement si pour tout point $q \in \Gamma$, il existe un point $p \in P$ tel que :

$$\|p - q\|_2 \leq \epsilon \varrho(q). \tag{2.2}$$

Cette définition est illustrée à la figure 2.4 pour deux valeurs de ϵ. Les disques gris correspondent à la quantité $\epsilon\varrho(p)$, où les points p appartiennent à $\Gamma \setminus P$. On peut observer

[5] Un sous-ensemble $X \subset \mathbb{R}^2$ est r-régulier si il est morphologiquement ouvert et fermé relativement à un disque de rayon r.

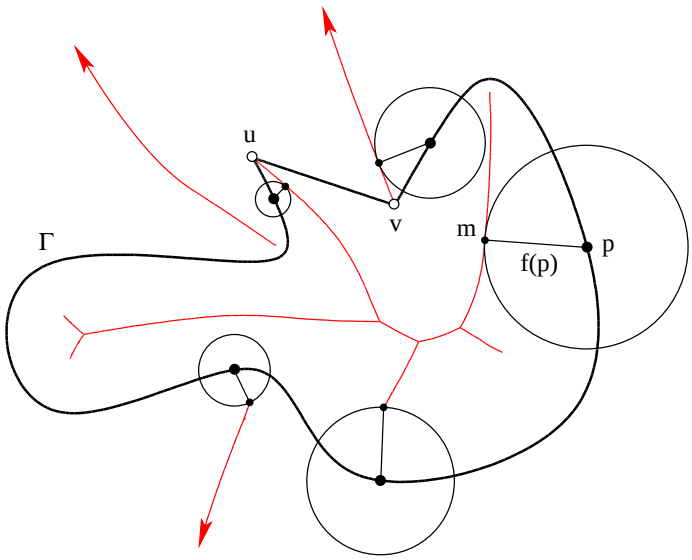


Fig. 2.3: La fonction ϱ, en un point $p \in \Gamma$, mesure la distance minimale de Γ à l'axe médian $M(\Gamma)$. C'est le rayon du disque de rayon maximal centré en p et vide de points de $M(\Gamma)$. Les points u et v, qui appartiennent à la fois à Γ et à $M(\Gamma)$, vérifient $\varrho = 0$.

que l'intersection de ces disques (fermés), avec l'échantillon P, n'est jamais vide.

De nombreuses propriétés découlent de la définition de l'ϵ-échantillonnage. Ces propriétés, qui sont décrites en détail dans [Amenta *et al.*, 1998, Dey, 2007], permettent de garantir la reconstruction correcte de courbes à partir d'ϵ-échantillons, avec différents algorithmes. Pour cela, les courbes nécessitent d'être régulières. Ceci provient du fait qu'à l'approche d'un point non-régulier $p \in \Gamma$, l'échantillon devient infiniment dense ($\varrho(p) = 0$). Dans la suite, nous considérerons donc le cas des courbes régulières.

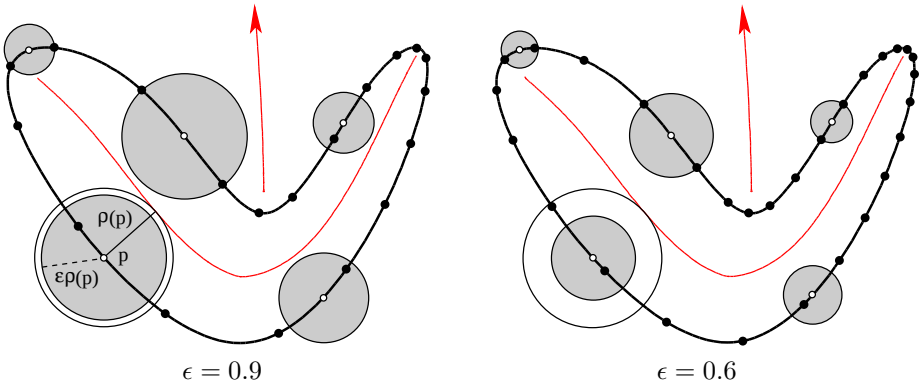


Fig. 2.4: Deux ϵ-échantillons d'une même courbe.

2.2.4 Le graphe de Gabriel

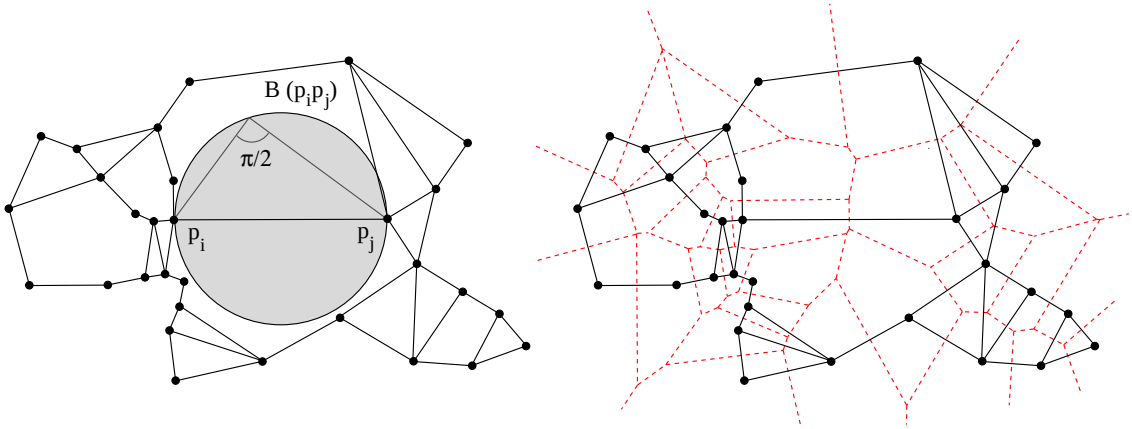
Le **graphe de Gabriel** de l'ensemble de points $P \subset \mathbb{R}^2$ [Gabriel et Sokal, 1969], noté $GG(P)$, est un graphe planaire connecté. Soient p_i et p_j deux point de P. Soit $B(p_ip_j)$ le disque fermé, de diamètre p_ip_j, circonscrit aux points p_i et p_j (voir la figure 2.5(a)). Dans la suite le disque $B(p_ip_j)$ est appelé le **disque de Gabriel** de l'arête p_ip_j. Alors, l'arête p_ip_j appartient à $GG(P)$ si et seulement si son disque de Gabriel, y compris son bord, ne contient aucun point de $P \setminus \{p_i, p_j\}$:

$$p_ip_j \in GG(P) \Leftrightarrow B(p_ip_j) \cap P \setminus \{p_i, p_j\} = \emptyset. \tag{2.3}$$

Le graphe de Gabriel est un sous-graphe de la triangulation de Delaunay de P [Matula et Sokal, 1980]. Soit $p_ip_jp_k$ et $p_ip_jp_l$ les deux triangles de $Del(P)$ incidents à l'arête p_ip_j (un seul lorsque $p_ip_j \in conv(P)$). Alors p_ip_j appartient à $GG(P)$ si et seulement si $B(p_ip_j) \cap \{p_k, p_l\} = \emptyset$.

Par dualité entre la triangulation de Delaunay et le diagramme de Voronoï, $GG(P)$ est également l'ensemble des arêtes de $Del(P)$ qui coupent leurs duaux (voir la figure 2.5(b)) :

$$p_ip_j \in GG(P) \Leftrightarrow p_ip_j \in Del(P) \text{ et } [p_i, p_j] \cap]v_i, v_j[\neq \emptyset,\ v_iv_j = dual(p_ip_j). \tag{2.4}$$



(a) $B(p_ip_j) \cap P \setminus \{p_i, p_j\} = \emptyset$ (b) dualité

Fig. 2.5: Graphe de Gabriel de l'ensemble de points P.

2.3 Analyse des approches de reconstruction existantes

Dans cette section, nous rappelons les définitions des principaux graphes capables de traiter le problème de la reconstruction polygonale de courbes. Ces graphes sont liés à l'approche que nous proposons. Ils sont basés sur la notion de voisinage et sur la triangulation de Delaunay. Afin de montrer les différences entre ces graphes, nous analysons leurs propriétés géométriques et combinatoires.

2.3.1 Les β-squelettes

Les β-squelettes d'un ensemble de points P décrivent une famille hiérarchique de graphes de voisinage, indexés par une valeur réelle positive β [Kirkpatrick et Radke, 1985]. Le voisinage de ces graphes est une extension du disque de Gabriel.

Les **β-disques** de deux points p_i et p_j du plan sont les deux disques fermés, circonscrits à p_i et à p_j, de même rayon :

$$r_\beta(p_ip_j) = \begin{cases} \|p_i - p_j\|_2/2\beta & \text{si } 0 < \beta \leq 1, \\ \beta\|p_i - p_j\|_2/2 & \text{si } 1 \leq \beta < +\infty. \end{cases} \tag{2.5}$$

Comme illustré à la figure 2.6, nous les notons respectivement $B^+_\beta(p_ip_j)$ et $B^-_\beta(p_ip_j)$. Le **β-voisinage** des points p_i et p_j, noté $N_\beta(p_ip_j)$, est l'intersection ou l'union des β-disques de p_i et p_j, selon la valeur de β :

$$N_\beta(p_ip_j) = \begin{cases} B^+_\beta(p_jp_j) \cap B^-_\beta(p_ip_j) & \text{si } 0 < \beta \leq 1, \\ B^+_\beta(p_ip_j) \cup B^-_\beta(p_ip_j) & \text{si } 1 \leq \beta < +\infty. \end{cases} \tag{2.6}$$

Alternativement, le β-voisinage de deux points p_i et p_j peut s'exprimer par l'une des

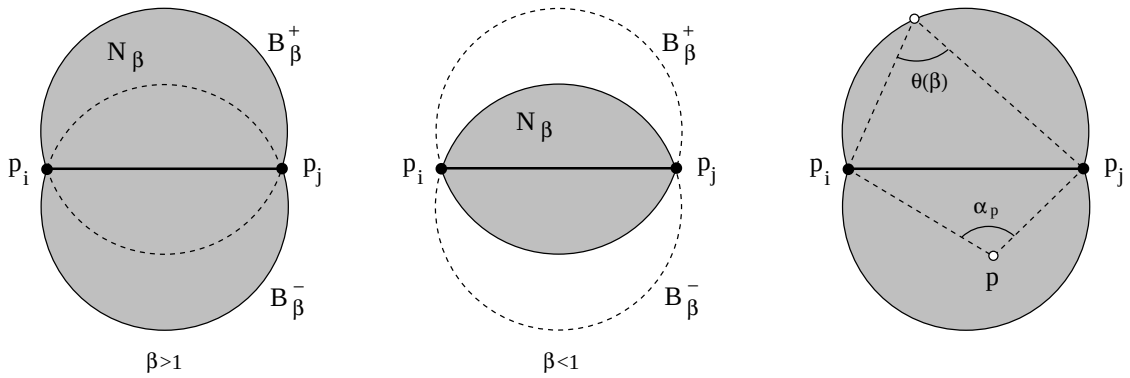


Fig. 2.6: Définition du β-voisinage N_β de deux points p_i et p_j à partir des deux β-disques. Alternativement, N_β est l'ensemble des points p tels que $\alpha_p \geq \theta(\beta)$.

deux expressions suivantes :

$$N_\beta(p_ip_j) = \{p \in \mathbb{R}^2 \ : \ \angle p_ipp_j \geq \theta(\beta)\}, \tag{2.7}$$

$$N_\beta(p_ip_j) = \{p \in \mathbb{R}^2 \ : \ \angle pp_ip_j + \angle p_ip_jp \leq \pi - \theta(\beta)\}, \tag{2.8}$$

où l'angle $\theta \in]0, \pi[$ est définit par :

$$\theta(\beta) = \begin{cases} \pi - \arcsin(\beta) & \text{si } \beta \in]0, 1], \\ \arcsin(1/\beta) & \text{si } \beta \in [1, +\infty[. \end{cases} \tag{2.9}$$

Étant donné $\beta \in]0, +\infty[$, le **β-squelette** de P connecte deux points p_i et p_j de P si leur β-voisinage ne contient aucun point de $P \setminus \{p_i, p_j\}$:

$$p_i p_j \in \beta\text{-squelette}(P) \Leftrightarrow N_\beta(p_i p_j) \cap \{P \setminus \{p_i, p_j\}\} = \emptyset.$$

Pour certaines valeurs de β, les β-squelettes correspondent à d'autres graphes de voisinage. En particulier, lorsque $\beta = 1$, le β-voisinage devient le disque de Gabriel, et le 1-squelette est le graphe de Gabriel. De plus, pour tout $\beta \geq 1$, les β-squelettes sont des sous-graphes de la triangulation de Delaunay.

Il a été montré que les β-squelettes reconstruisent correctement une collection de courbes régulières fermées si P est un ϵ-échantillon avec $\epsilon < 0.297$ [Amenta *et al.*, 1998]. Des notions similaires aux β-squelettes ont ensuite été développées pour garantir la reconstruction de courbes non-régulières, fermées ou ouvertes [Funke et Ramos, 2001].

2.3.2 Les A-formes et le Crust

Les A-formes permettent de décrire la forme d'un ensemble de points P en utilisant un ensemble de **points de contrôle** $A \subset \mathbb{R}^2 \setminus P$, et la propriété de disque vide entre les points de P et les points de A [Melkemi, 1997, Melkemi et Melkemi, 1995].

Étant donné un ensemble de points A, la **A-forme** de P, notée $F_A(P)$, est le graphe formé des arêtes $p_i p_j$ qui connectent des points de P, et pour lesquelles il existe un disque ouvert, circonscrit aux points p_i, p_j et à un point de A, qui ne contient aucun point de $P \cup A$:

$$p_i p_j \in F_A(P) \Leftrightarrow \exists a \in A \ : \ b(p_i p_j a) \cap \{P \cup A\} = \emptyset.$$

D'après la définition de la triangulation de Delaunay, la A-forme de P est à la fois un sous-graphe de $Del(P \cup A)$ et de $Del(P)$. Un exemple de A-forme est illustré à la figure 2.7. On peut y observer qu'elle est capable de reconstruire la forme d'un ensemble de points répartis sur le contour d'objets, aussi bien que la forme d'amas de points. L'ensemble de points A a pour rôle de séparer les différentes parties des contours à reconstruire. Comme les arêtes du diagramme de Voronoï de P constituent l'ensemble séparateur médian de tous les points de P (l'axe médian de P), leurs sommets sont de bons candidats pour l'ensemble A. Dans certaines configurations, A peut être calculé explicitement (voir par exemple [Melkemi, 1997, Melkemi et Djebali, 2000]).

Indépendamment, une notion similaire aux A-formes a été proposée dans le cadre spécifique de la reconstruction de courbes. Elle s'inspire du fait que les sommets du diagramme de Voronoï d'un échantillon dense d'une courbe régulière, approximent l'axe médian de

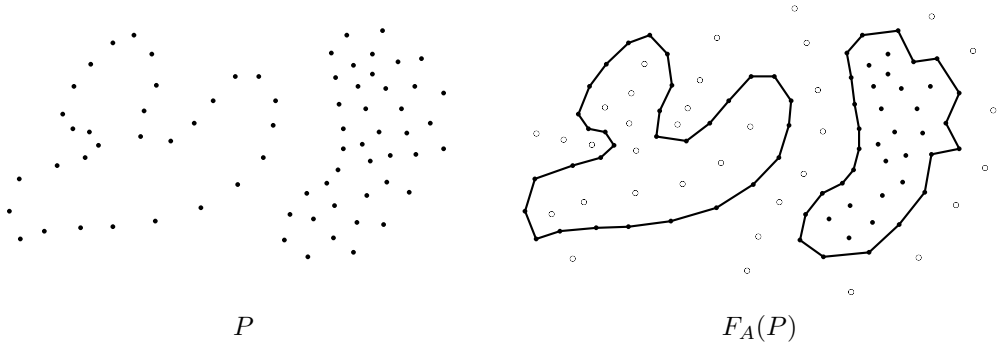


Fig. 2.7: A-forme d'un ensemble de points, où les points de A sont les points cerclés.

cette courbe [Brandt et Algazi, 1992, Brandt, 1994]. Soit V l'ensemble des sommets du diagramme de Voronoï de P. Le **Crust** de P [Amenta *et al.*, 1998] est le sous-graphe de $Del(P)$ formé des arêtes p_ip_j telles que p_i et p_j sont des points de P :

$$p_ip_j \in \text{Crust}(P) \Leftrightarrow p_ip_j \in Del(P \cup V).$$

Alternativement, une arête p_ip_j appartient au Crust de P si et seulement si il existe un disque ouvert, vide de points de $P \cup V$, et circonscrit à p_i et à p_j. Le Crust est également un sous-graphe de $Del(P)$. C'est l'ensemble des arêtes de $Del(P)$ qui subsistent dans $Del(P \cup V)$. Il s'en suit que le Crust est un cas particulier des A-formes lorsque $A = V$. Un exemple de Crust est illustré à la figure 2.8. Remarquons que les arêtes de $Del(P \cup V)$, qui connectent deux points de V, forment également un Crust, qui donne une idée de l'axe médian des courbes reconstruites.

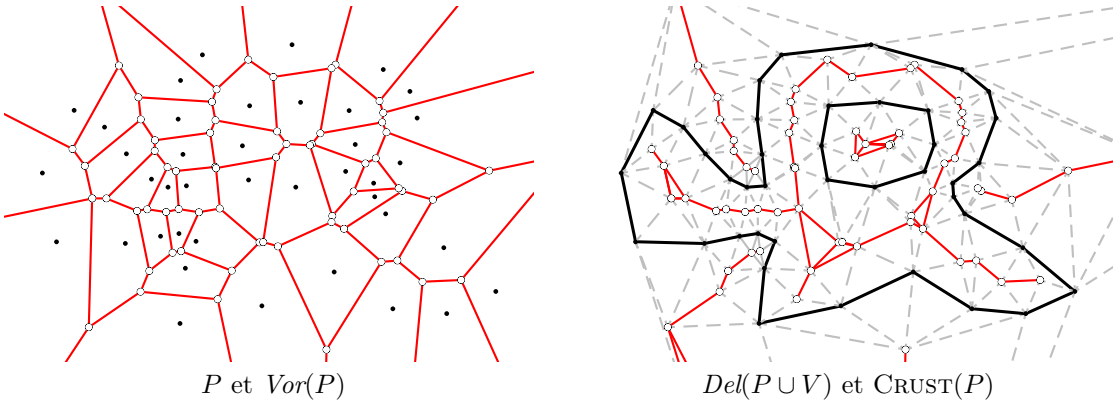


Fig. 2.8: Construction du Crust : calcul de l'ensemble V des sommets de $Vor(P)$ (points cerclés), et extraction des arêtes de $Del(P \cup V)$ qui connectent deux points de P (traits épais).

Le Crust garantit la reconstruction de collection de courbes régulières fermées à partir d'ϵ-échantillons. Il a d'abord été montré que la reconstruction est correcte si $\epsilon < 0.252$ [Amenta *et al.*, 1998]. Cette borne a ensuite été améliorée à 0.42 en montrant que pour $\epsilon < 0.42$, le Crust d'ϵ-échantillons de courbes régulières fermées est équivalent au Crust local [Gold et Snoeyink, 2001], présenté dans la prochaine section.

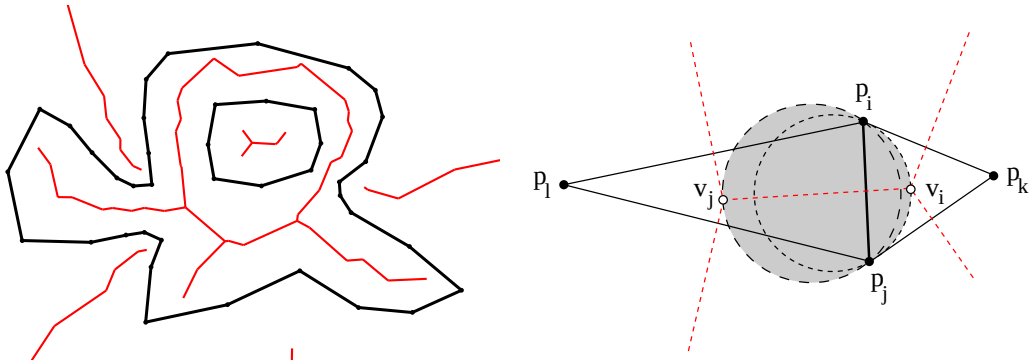
2.3.3 Le CRUST local et l'anti-CRUST

Le CRUST local est une variante du CRUST où les points de contrôle sont définis localement pour chaque arête de la triangulation de Delaunay de P [Gold, 1999]. Formellement, soient p_ip_j une arête de $Del(P)$ et e son arête duale dans $Vor(P)$. L'arête p_ip_j appartient au **CRUST local** de P, noté $LC(P)$, si elle subsiste dans la triangulation $Del(P \cup V)$, où V est l'ensemble des extrémités de l'arête e :

$$p_ip_j \in LC(P) \Leftrightarrow \begin{cases} p_ip_j \in Del(P \cup \{v_i, v_j\}) & \text{si } p_ip_j \notin conv(P), \\ p_ip_j \in Del(P \cup \{v_i\}) & \text{sinon,} \end{cases} \tag{2.10}$$

où v_i et v_j sont les extrémités de l'arête e. Les arêtes duales des arêtes de Delaunay qui n'appartiennent pas au CRUST local de P, forment l'**anti-CRUST** de P, noté $AC(P)$. Ce graphe fournit une approximation de l'axe médian des courbes reconstruites par le CRUST local. Un exemple de CRUST local et d'anti-CRUST est illustré à la figure 2.9(a).

Bien que le CRUST soit un sous-graphe du CRUST local [Gold et Snoeyink, 2001], ces deux graphes sont identiques dans la majorité des configurations. La différence importante entre ces graphes réside dans la façon de les définir, et donc de les calculer. Le CRUST local se calcule plus rapidement que le CRUST. Cela provient du fait qu'il n'est pas nécessaire de calculer $Del(P \cup V)$ pour déterminer si une arête appartient à $LC(P)$. Étant donné que le CRUST local et l'anti-CRUST sont directement liés aux graphes de voisinage que nous proposons dans la section 2.4, nous détaillons cette propriété, ainsi que d'autres propriétés importantes que nous utiliserons par la suite.



(a) $LC(P)$ et $AC(P)$ (b) $p_ip_j \in LC(P)$

FIG. 2.9: (a) exemple de CRUST local (traits noirs épais) et d'anti-CRUST (traits rouges fins), à partir de l'ensemble de points de la figure 2.8. (b) propriété du CRUST local.

Propriétés du CRUST local et de l'anti-CRUST. En utilisant les propriétés de disques vides de la triangulation de Delaunay, on obtient la définition alternative suivante du CRUST local, en partie démontrée dans [Gold, 1999, Gold et Snoeyink, 2001].

Propriété 3 *Soit p_ip_j une arête de Del(P). Alors, nous avons (voir la figure 2.9(b)) :*

$$p_ip_j \in LC(P) \Leftrightarrow \begin{cases} b(p_ip_jv_i) \cap \{v_j\} = \emptyset \text{ et } b(p_ip_jv_j) \cap \{v_i\} = \emptyset & \text{si } p_ip_j \notin conv(P), \\ p_ip_j \in GG(P) & \text{sinon,} \end{cases}$$

où $v_iv_j \in Vor(P)$ est l'arête duale de p_ip_j.

Preuve. Soit $p_ip_j \notin conv(P)$ une arête telle que $p_ip_j \in LC(P)$. D'après l'équation (2.10), $p_ip_j \in Del(P \cup \{v_i, v_j\})$, et donc il existe deux triangles de Delaunay $p_ip_jp_k$ et $p_ip_jp_l$ tels que $p_k, p_l \in P \cup \{v_i, v_j\}$, $b(p_ip_jp_k) \cap \{P \cup \{v_i, v_j\}\} = \emptyset$ et $b(p_ip_jp_l) \cap \{P \cup \{v_i, v_j\}\} = \emptyset$. D'autre part comme v_i et v_j sont les centres de disques de $Del(P)$, on a $\|p_i - v_i\|_2 = \|p_j - v_i\|_2 \leq \|p - v_i\|_2$ pour tout $p \in P$ (et de même pour v_j). Alors, nous avons $p_k = v_i$ et $p_l = v_j$, et par conséquent $b(p_ip_jv_i) \cap \{v_j\} = \emptyset$, et $b(p_ip_jv_j) \cap \{v_i\} = \emptyset$. La réciproque est évidente. Pour les arêtes de $conv(P)$, en suivant le même processus que précédemment, on obtient $p_ip_j \in conv(P)$ et $p_ip_j \in LC(P) \Rightarrow b(p_ip_jv_i) \cap P = \emptyset$. Or, cette dernière expression est vérifiée ssi $v_i \in conv(P)$, c.-à-d. ssi $p_ip_j \in GG(P)$. La réciproque se déduit de la Propriété 4. □

En pratique, pour les arêtes internes à $Del(P)$, c.-à-d. celles n'appartenant pas à $conv(P)$, un seul des deux tests de la Propriété 3 est nécessaire pour déterminer si une arête appartient à $LC(P)$ (tests $b(p_ip_jv_i) \cap \{v_j\} = \emptyset$ et $b(p_ip_jv_j) \cap \{v_i\} = \emptyset$). En effet, d'après la relation de dualité entre les arêtes p_ip_j et v_iv_j, ces deux tests sont équivalents [Gold et Snoeyink, 2001] :

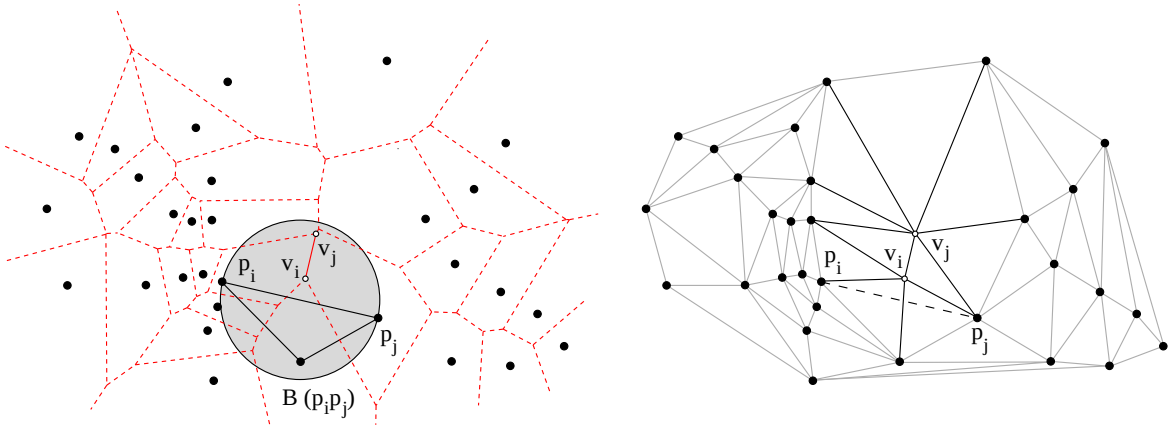
$$p_ip_j \in LC(P),\ p_ip_j \notin conv(P) \ \Leftrightarrow\ b(p_ip_jv_i) \cap \{v_j\} = \emptyset \ \Leftrightarrow\ b(p_ip_jv_j) \cap \{v_i\} = \emptyset.$$

Ce test correspond au test standard $inCircle(p_i, p_j, v_i, v_j)$ [Guibas et Stolfi, 1985], qui détermine si le point v_j est contenu à l'intérieur du cercle circonscrit aux points p_i, p_j et v_i. Numériquement, ce test évalue le signe du déterminant de la matrice 4×4 constituée de quatre lignes de la forme $(1, x_p, y_p, \|p\|_2^2)$, où $p = (x_p, y_p)$ est un des points considérés. C'est avec ce test que Gold et Snoeyink ont proposé de calculer le Crust local à partir du diagramme de Voronoï. Dans cette même section, nous verrons qu'il existe un test plus simple pour construire les arêtes du Crust local à partir du diagramme de Voronoï.

D'autre part, Gold et Snoeyink ont également proposé de construire le Crust local à partir de la triangulation de Delaunay. Soient $p_kp_ip_j$ et $p_jp_ip_l$ les deux triangles de $Del(P)$ incidents à l'arête p_ip_j, et soit v le vecteur tel que $\angle v, (p_j - p_i) = \frac{\pi}{2}$ dans le sens trigonométrique. Alors, ils suffit d'évaluer le test suivant :

$$(p_l - p_i) \cdot (p_l - p_j) * (p_k - p_i) \cdot (p_k - p_j) \geq -(p_l - p_j) \cdot v * (p_k - p_i) \cdot v,$$

qui est vrai si et seulement si l'arête p_ip_j appartient à $LC(P)$.
Une autre propriété importante est que le Crust local de P est un sous-graphe du graphe de Gabriel de P. Nous démontrons ici cette propriété importante du Crust local, indirectement mentionnée dans [Gold et Snoeyink, 2001].



(a) $p_ip_j \notin GG(P)$ (b) $p_ip_j \notin Del(P \cup \{v_i, v_j\})$

Fig. 2.10: Illustration de la preuve de la Propriété 4.

Propriété 4 $LC(P) \subseteq GG(P)$.

Preuve. Soit p_ip_j une arête de $Del(P)$ qui n'appartient pas à $GG(P)$, comme à la figure 2.10(a). Soit v_iv_j son arête duale dans $Vor(P)$. Montrons que p_ip_j n'appartient pas à $LC(P)$. Soit p_k le point de $P \setminus \{p_i, p_j\}$ tel que $p_k \in B(p_ip_j)$. Les sommets v_i et v_j sont alors situés à l'extérieur du triangle de Delaunay $p_ip_jp_k$. Considérons que le sommet v_i soit le centre du cercle de Delaunay circonscrit à $p_ip_jp_k$. L'insertion de v_i à $Del(P)$ crée alors les arêtes p_iv_i, p_jv_i et p_kv_i. De plus, comme illustré à la figure 2.10(b), v_ip_k coupe p_ip_j. Donc $p_ip_j \notin Del(P \cup \{v_i, v_j\})$, et par conséquent $p_ip_j \notin LC(P)$. □

La Propriété 4 montre que les arêtes de $LC(P)$ coupent leurs arêtes duales dans $Vor(P)$, ou par dualité, que les arêtes de $LC(P)$ ne coupent jamais les arêtes de $AC(P)$. Ainsi, les définitions du Crust local et de l'anti-Crust sont consistantes avec celles d'une courbe régulière et de son axe médian. Dans ce contexte, le Crust local garantit la reconstruction de courbes régulières fermées à partir d'ϵ-échantillons où $\epsilon < 0.42$ [Gold et Snoeyink, 2001]. Cette dernière borne (non optimale) est l'une des plus élevée connue.

Il existe une seconde garantie de reconstruction associée au Crust local. Elle a été établie dans le contexte de l'échantillonnage et de la reconstruction de formes r-régulières [Attali, 1997]. Contrairement à l'ϵ-échantillonnage, les courbes sont dans ce cas échantillonnées régulièrement. Néanmoins, le graphe permettant la reconstruction correspond exactement au Crust local, qui y est défini en considérant les angles entre les disques de Delaunay. Plus précisément, si p_ip_j est une arête de $Del(P)$, v_i et v_j les centres des disques de Delaunay passant par p_i et p_j, alors l'angle formé par ces disques est mesuré par (voir la figure 2.11(a)) :

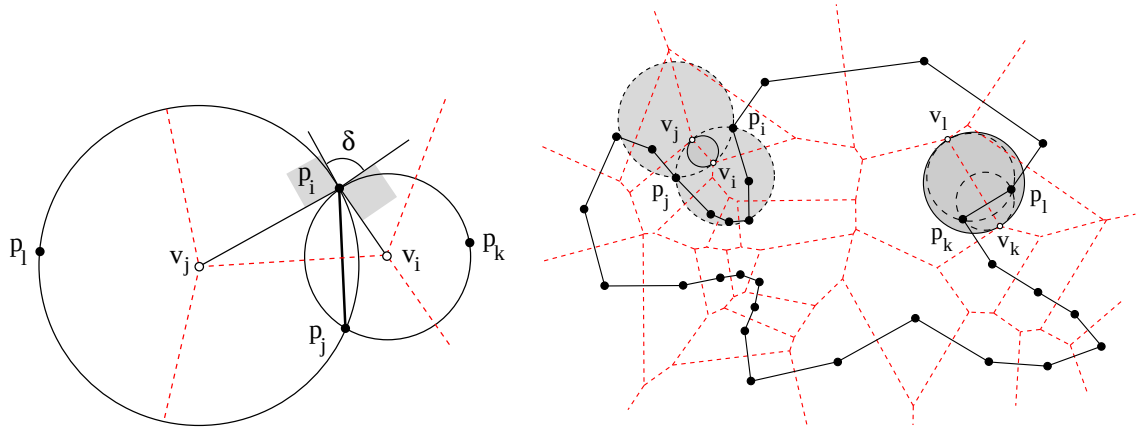
$$\delta(p_ip_j) = \pi - \angle v_ip_iv_j.$$

On peut alors formuler les deux propriétés suivantes, qui montrent le lien entre l'angle δ, la définition du Crust local, et celle de l'anti-Crust.

Propriété 5 *Soient v_iv_j une arête de Vor(P) et p_ip_j son arête duale dans Del(P). Alors :*

$$v_iv_j \in AC(P) \Leftrightarrow \begin{cases} B(v_iv_j) \cap P = \emptyset & \text{si } v_iv_j \text{ bornée,} \\ p_ip_j \notin GG(P) & \text{sinon.} \end{cases}$$

Preuve. Soit v_iv_j une arête bornée. D'après la définition de l'anti-CRUST et la Propriété 3, on a $v_iv_j \in AC(P) \Leftrightarrow v_j \in b(p_ip_jv_i)$ (et $v_i \in b(p_ip_jv_j)$). Comme illustré à la figure 2.11(b), ceci est équivalent à $B(v_iv_j) \subset b(p_ip_jv_i)$ (et respectivement $B(v_iv_j) \subset b(p_ip_jv_j)$). Alors, les points p_i et p_j ne sont pas dans $B(v_iv_j)$. Comme ce sont les deux plus proches points du milieu de v_iv_j, il n'existe pas d'autres points de P dans $B(v_iv_j)$. Le cas où v_iv_j est non-bornée se déduit directement de la définition de $AC(P)$ et de la Propriété 3. □



(a) $\angle v_ip_iv_j = \pi - \delta$ (b) $p_ip_j \notin LC(P)$, $p_kp_l \in LC(P)$

FIG. 2.11: (a) Définition de l'angle δ formé par les triangles de Delaunay. (b) Illustration de la preuve de la Propriété 5.

Propriété 6 *Soit p_ip_j une arête interne à Del(P). Alors, nous avons :*

$$p_ip_j \in LC(P) \Leftrightarrow \delta(p_ip_j) \leq \frac{\pi}{2}$$

Preuve. Soit $v_iv_j \in Vor(P)$ l'arête duale de p_ip_j. Par définition, on a $p_ip_j \in LC(P) \Leftrightarrow v_iv_j \notin AC(P)$. D'après la Propriété 5, ceci est équivalent à $B(v_iv_j) \cap P \neq \emptyset$. Comme p_i et p_j sont les deux points de P les plus proche du centre de $B(v_iv_j)$, p_i et p_j sont inclus dans $B(v_iv_j)$, ce qui est équivalent à $\angle v_ip_iv_j \geq \pi/2$. □.

La Propriété 5 montre que l'anti-CRUST peut être formaliser en utilisant des disques de Gabriel circonscrits aux arêtes du diagramme de Voronoï. De plus, comme le CRUST local est inclus dans le graphe de Gabriel (Propriété 4), il correspond aux arêtes de la triangulation de Delaunay qui sont totalement incluses dans le disque de Gabriel de leurs arêtes duales. Ce sont ces propriétés qui ont inspiré notre méthode de reconstruction. Dans ce cadre, nous verrons d'autres interprétations du CRUST local.

La Propriété 5, permet également de formuler un autre test pour construire les arêtes du CRUST local à partir du diagramme de Voronoï. Ce test détermine si le point p_i est situé à l'intérieur du disque de Gabriel de l'arête v_iv_j (disque de diamètre $[v_i, v_j]$). Numériquement, l'équation à résoudre pour évaluer ce test est la suivante :

$$(x_{p_i} - x_{v_i}) * (x_{v_j} - x_{p_i}) > (y_{p_i} - y_{v_i}) * (y_{p_i} - y_{v_j}).$$

Remarquons que cette dernière fait intervenir moins d'opérations que celle proposée dans [Gold et Snoeyink, 2001] (signe du déterminant d'une matrice 4×4).

Quelque soient la structure (triangulation de Delaunay ou/et diagramme de Voronoï) et le test choisis, l'espérance du calcul du CRUST local (et de l'anti-CRUST) est de $O(N \log N)$, où N est le nombre de points de l'ensemble P.

2.3.4 Le CRUST des plus proches voisins

Rappelons qu'un point $p_j \in P$ est le **plus proche voisin** d'un point $p_i \in P$, noté $NN(p_i)$, si et seulement si il n'existe pas de point $p_k \in P \setminus \{p_i, p_j\}$ tel que $\|p_i - p_k\|_2 < \|p_i - p_j\|_2$. Le **graphe des plus proches voisins** de P, noté $NN(P)$, connecte deux points p_i et p_j de P si et seulement si $p_i = NN(p_j)$ ou si $p_j = NN(p_i)$.

Le **CRUST des plus proches voisins** de P [Dey et Kumar, 1999], noté $NNC(P)$, est le graphe construit à partir des deux étapes suivantes :

(i) Calculer $NN(P)$.

(ii) Pour chaque point $p_i \in P$ qui n'a qu'un seul voisin dans $NN(P)$, connecter p_i au plus proche point de l'ensemble $\{\angle p_j p_i NN(p_i) > \pi/2,\ \forall p_j \in P \setminus \{p_i, NN(p_i)\}\}$, si ce dernier existe.

C'est un sous-graphe de $Del(P)$ qui peut être calculé en $O(N \log N)$. Dans le contexte de l'ϵ-échantillonnage, il a été d'abord montré que $NNC(P)$ fournit la reconstruction correcte d'une collection de courbes régulières fermées si $\epsilon < 1/3$ [Dey et Kumar, 1999]. Cette borne a ensuite été améliorée à 0.48, à travers une extension portant sur l'angle qui intervient dans l'étape (ii) [Lenz, 2006].

2.3.5 Le CRUST conservatif

Le **CRUST conservatif** de P [Dey *et al.*, 2000], noté $CC(P)$, est le sous-graphe du graphe de Gabriel de P construit à partir des trois étapes suivantes :

(i) Calculer le graphe $G = (P, E)$ constitué des arêtes $p_ip_j \in Del(P)$ qui vérifient :

$$p_ip_j \in G \Leftrightarrow p_ip_j \in GG(P) \text{ et } B(p_{ij}, \|p_i - p_j\|_2/\rho) \cap V = \emptyset,$$

où p_{ij} est le milieu du segment $[p_i, p_j]$, V est l'ensemble des sommets de $Vor(P)$, et $\rho > 0$ est un paramètre global.

(ii) Supprimer toutes les arêtes $p_ip_j \in G$ pour lesquelles $B(p_{ij}, \|p_i - p_j\|_2/4\rho) \cap G$ contient un point de P, connecté dans G à au plus un point $p \in P$ tel que p ne soit pas connecté à p_i et à p_j.

(iii) Répéter (ii) tant qu'il n'existe plus d'arêtes à supprimer.

Son espérance d'exécution est de $O(N \log N)$. De plus, il est relié à l'ϵ-échantillonnage par les deux propriétés suivantes [Dey *et al.*, 2000] :

1. Si $\rho < 1/2$ et P est un $(\rho/8)$-échantillon d'une collection Γ de courbes régulières fermées ou ouverte, alors la reconstruction correcte de Γ est incluse dans $CC(P)$.
2. Il existe une constante positive $c \leq 13.35$ telle que pour tout $\rho < 1/8$, P est un $(c\rho)$-échantillon de Γ, et $CC(P)$ est la reconstruction correcte de Γ.

Ces garanties de reconstruction sont les premières concernant des courbes ouvertes et fermées. En pratique, les auteurs ont observés que $\rho \in [1.25, 1.75]$ fournit des résultats satisfaisants. Dans la suite, nous montrons que pour ces valeurs, le Crust conservatif est un sous-graphe du Crust local.

Propriété 7 *Pour tout $\rho < 2$, $CC(P) \subseteq LC(P)$.*

Preuve. Soient $p_ip_j \in GG(P)$, et $v_iv_j \in Vor(P)$ son arête duale. Par définition, p_ip_j appartient au graphe G formé à l'étape (i) si $\|p_i - p_j\|_2/\rho < \|p_{ij} - v_i\|_2$, où v_i est l'extrémité de v_iv_j la plus proche de p_{ij}. En prenant $\rho \in]0, 2[$, ceci implique que $\|p_{ij} - p_i\|_2 < \|p_{ij} - v_i\|_2$. Soit c_i le centre du disque ouvert $b(p_ip_jv_i)$. Alors on a $p_{ij} \in]v_i, v_j[$, et $\|c_i - v_j\|_2 = \|c_i - p_{ij}\|_2 + \|p_{ij} - v_i\|_2$. Comme $\|p_{ij} - v_j\|_2 \geq \|p_{ij} - v_i\|_2$, on a $\|c_i - v_j\|_2^2 \geq \|c_i - p_{ij}\|_2^2 + \|p_{ij} - v_i\|_2^2$. De plus, dans le triangle rectangle $c_ip_{ij}p_i$, on a $\|c_i - p_i\|_2^2 = \|c_i - p_{ij}\|_2^2 + \|p_{ij} - p_i\|_2^2 < \|c_i - p_{ij}\|_2^2 + \|p_{ij} - v_i\|_2^2$. Donc $\|c_i - v_j\|_2^2 > \|c_i - p_i\|_2^2 = \|c_i - v_i\|_2^2$, et par conséquent $b(p_ip_jv_i) \cap \{v_j\} = \emptyset$. D'après la Propriété 3, $p_ip_j \in LC(P)$. Comme $CC(P) \subseteq G$ (étapes (ii) et (iii)), on en déduit la propriété. □

Cette propriété montre que si $\rho < 2$, le Crust conservatif ne peut pas reconstruire correctement des courbes échantillonnées si des arêtes correctes sont manquantes dans le Crust local. En revanche, si $\rho \geq 2$, ces arêtes manquantes peuvent être présentes dans $CC(P)$ (étape (i)), avec éventuellement des arêtes incorrectes. Notre méthode de reconstruction utilise une notion similaire à celle proposée dans l'étape (i), avec un rapport différent entre les arêtes de Delaunay et de Voronoï.

Les idées développées dans le Crust conservatif, et dans le Crust des plus proches voisins, ont ensuite été étendues pour prendre en compte le cas des courbes non-régulières, à travers deux méthodes respectivement nommées Gathan [Dey et Wenger, 2001] et GathanG [Dey et Wenger, 2002]. Bien que nous ne décrivons pas ici ces méthodes, elles sont comparées par la suite à l'approche que nous proposons.

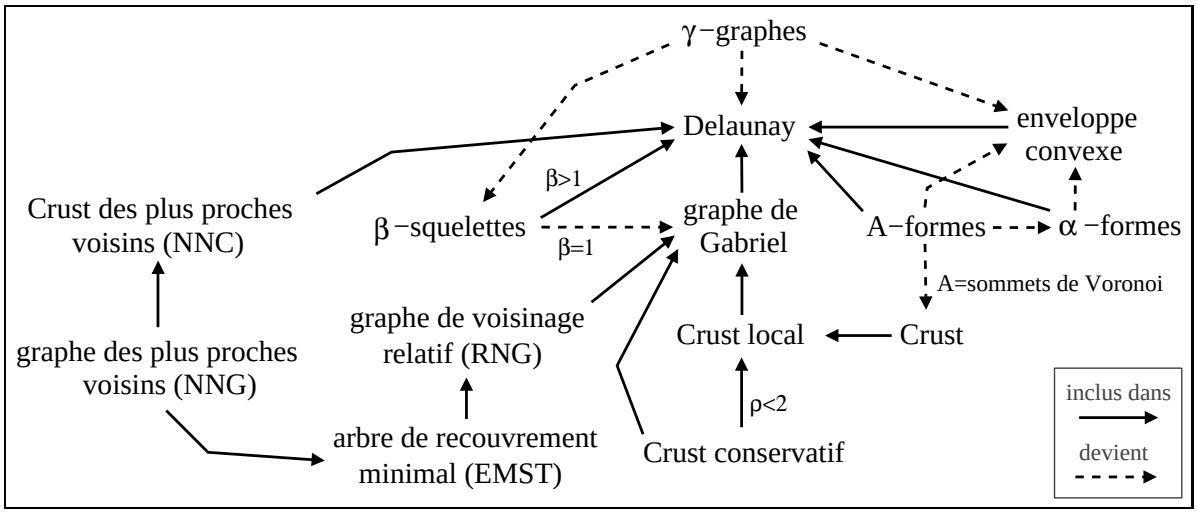


Fig. 2.12: Hiérarchie des principaux graphes de voisinage utilisés en reconstruction de courbes.

2.3.6 Schéma récapitulatif des graphes de voisinage

Le schéma de la figure 2.12 récapitule les graphes de voisinage couramment utilisés en reconstruction polygonale de courbes. Nous pouvons observer qu'ils sont organisés selon une hiérarchie, la majorité d'entre-eux étant des sous-graphes du graphe de Gabriel et/ou de la triangulation de Delaunay. En particulier, nous avons vu que le Crust local se situe au centre de plusieurs graphes efficaces, avec lesquels la garantie de reconstruction correcte peut s'exprimer à partir la condition d'ϵ-échantillonnage. Lorsque cette condition n'est pas satisfaite, la reconstruction peut ne pas être correcte. Par exemple, ceci peut arriver lorsque l'ensemble de points est un ϵ-échantillon avec ϵ supérieur aux bornes annoncées en théorie (voir les figures 2.13 et 2.14).

Pour augmenter les capacités de reconstruction du Crust local, et des autres graphes vus précédemment, nous introduisons une famille de graphes de voisinage qui se situent entre le graphe de Gabriel et le Crust local.

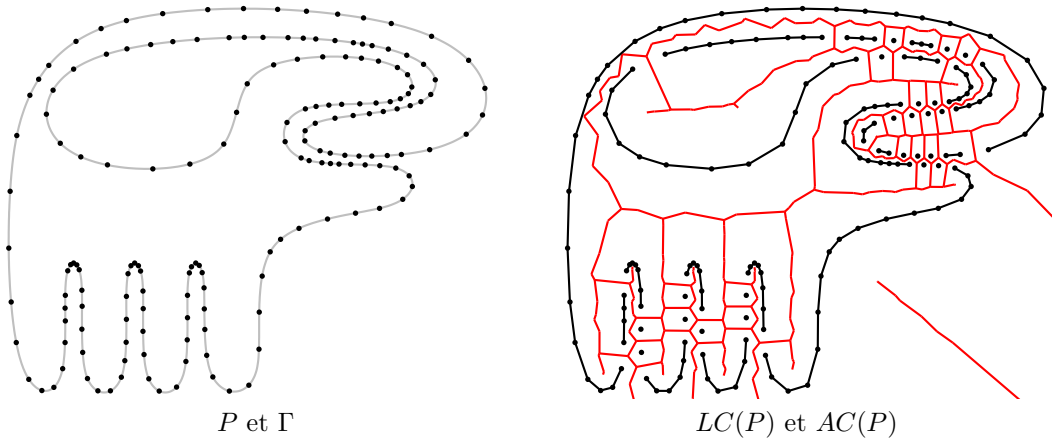


Fig. 2.13: Reconstruction incorrecte d'une courbe à partir d'un échantillon, avec le Crust local.

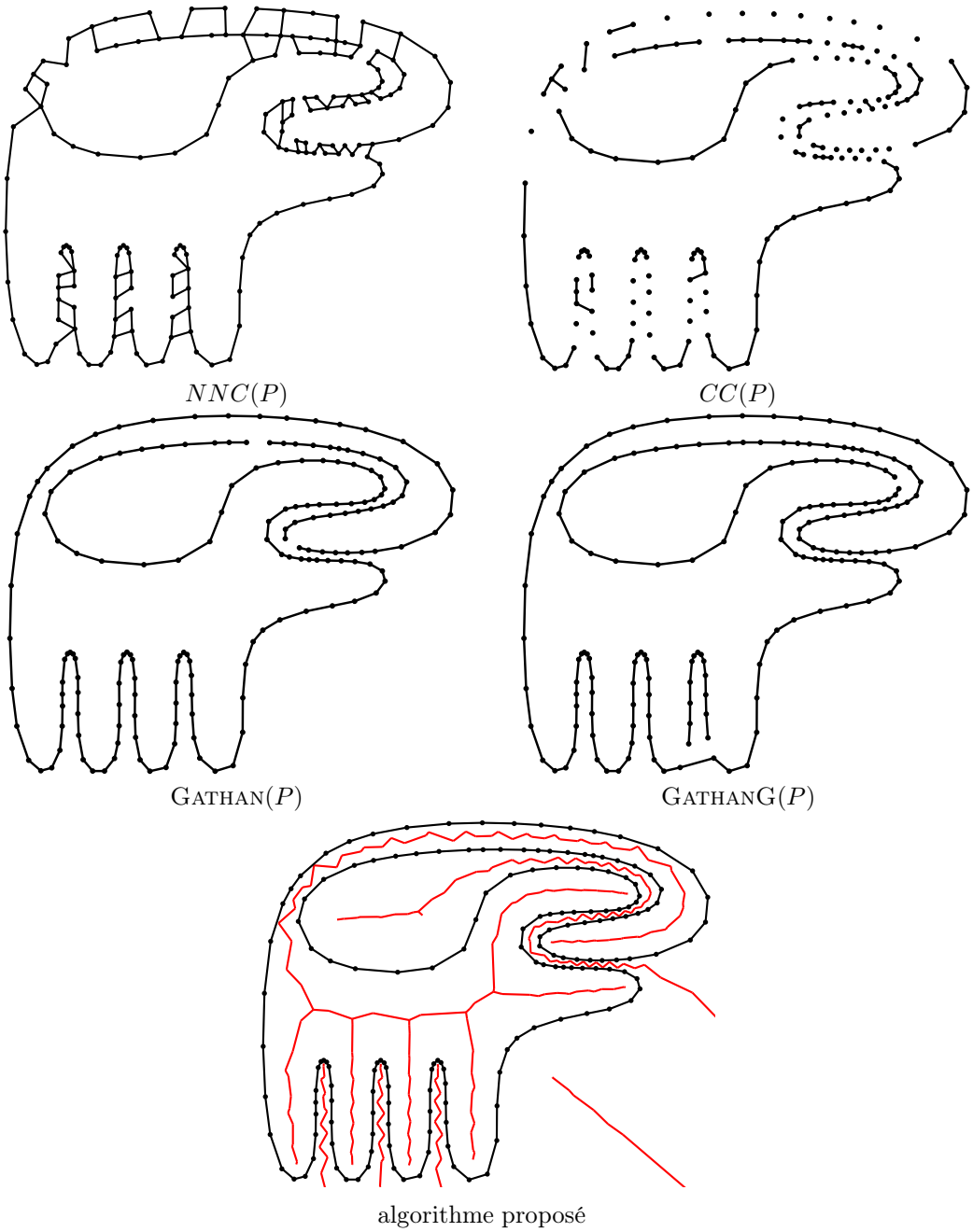


FIG. 2.14: Reconstruction de le courbe de la figure 2.13 avec le CRUST des plus proches voisins, le CRUST conservatif, GATHAN, GATAHNG, et l'algorithme proposé dans ce chapitre (section 2.6).

2.4 Graphes proposés : β-CRUSTS locaux et β-axes médians

Cette section décrit les familles de graphes que nous proposons pour reconstruire des courbes simples et extraire leur axe médian, à partir d'un ensemble de points P. Ces graphes, que nous nommons respectivement les β-CRUSTS locaux et les β-axes médians [Bougleux *et al.*, 2007c], sont générés à partir d'une relation de voisinage entre les arêtes du diagramme de Voronoï de P, et leurs arêtes duales dans la triangulation de Delaunay de P. À partir des propriétés géométriques de ces graphes, nous proposons un algorithme en $O(N \log N)$ pour les calculer.

2.4.1 Notion de voisinage sur les arêtes de Voronoï

Les graphes que nous proposons sont générés à partir du β-voisinage associé aux arêtes du diagramme de Voronoï. Le β-voisinage est initialement défini pour former la famille des β-squelettes (voir la section 2.3.1). Rappelons que le β-voisinage de deux points p et q de $\mathbb{R}^2$ est, selon la valeur de β, l'union ou l'intersection des deux disques de même rayon r_β, circonscrits à p et à q. Dans le cas des arêtes de Voronoï, on ne peut pas directement définir le β-voisinage des arêtes non-bornées (demi-droites). Pour cela, nous distinguons le cas des arêtes bornées (segments) de celui des arêtes non-bornées.

Cas des arêtes bornées. Soit v_iv_j une arête bornée de $Vor(P)$. Dans ce cas, le voisinage considéré est le β-voisinage $N_\beta(v_iv_j)$, privé des extrémités de l'arête v_iv_j, et noté :

$$VN_\beta(v_iv_j) := N_\beta(v_iv_j) \setminus \{v_i, v_j\}.$$

Rappelons que le voisinage N_β, défini à la section 2.3.1, peut s'exprimer par l'une des équations (2.6), (2.7) ou (2.8). Des exemples de β-voisinages sont illustrés à la figure 2.15.

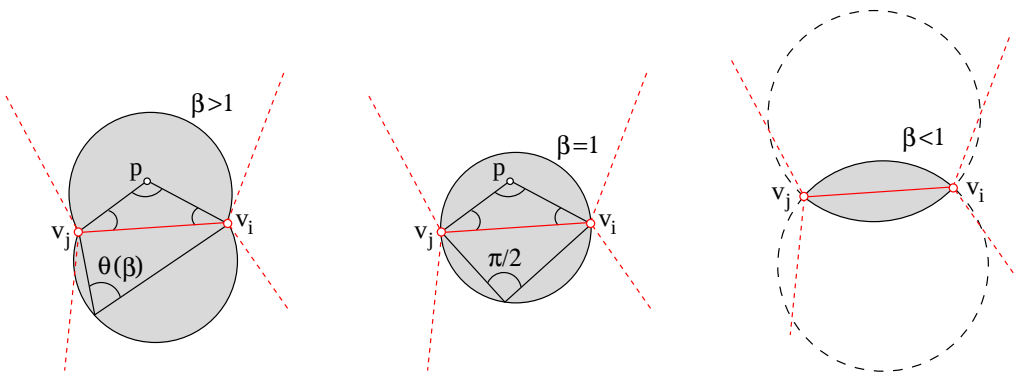


FIG. 2.15: β-voisinage des arêtes bornées de Voronoï.

Cas des arêtes non-bornées. Soit v_iv une arête non-bornée de $Vor(P)$, comme illustrée à la figure 2.16. Pour définir le β-voisinage de cette arête, un point v_j est artificiellement positionné sur la demi-droite $[v_i, v)$, à une distance suffisamment éloignée du

point v_i. Dans la suite, le point v_j est situé à l'infini sur $[v_i, v)$. Ainsi, en considérant l'expression (2.8) du β-voisinage, l'angle $\angle v_i v_j p$ tend vers zéro. Ceci réduit l'expression du β-voisinage à :

$$VN_\beta(v_i v) := \{p \in \mathbb{R}^2 \setminus \{v_i\} \ : \ \angle v v_i p \leq \pi - \theta(\beta)\}. \tag{2.11}$$

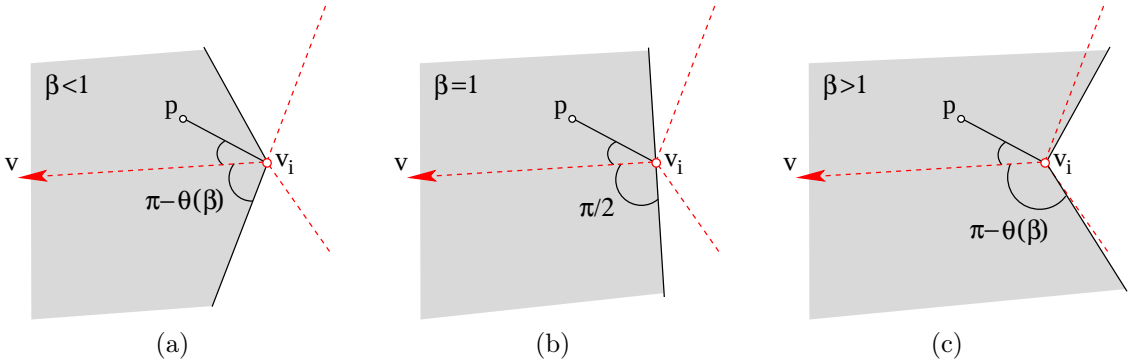


Fig. 2.16: β-voisinage des arêtes de Voronoï non-bornées, pour différentes valeurs de β.

2.4.2 Définition des graphes

À partir du β-voisinage des arêtes du diagramme de Voronoï de l'ensemble de points P, nous définissons maintenant les deux familles proposées de graphes de voisinage. Ces définitions exploitent la relation de dualité entre le diagramme de Voronoï de P et la triangulation de Delaunay de P.

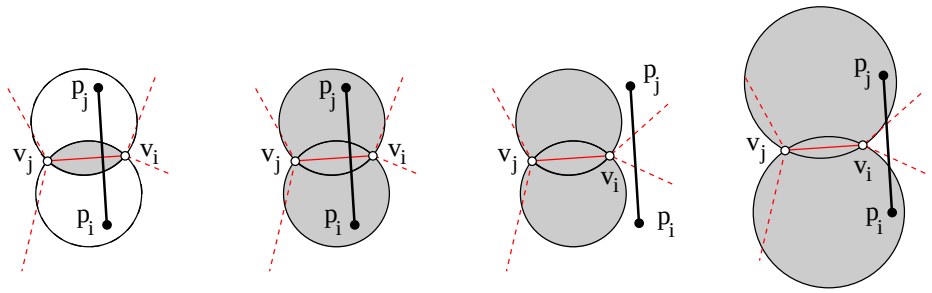
Définition 1 (β-Crust local et β-axe médian) *Étant donné $\beta \in]0, +\infty[$, le β-Crust local de P, noté β-$LC(P)$, est le sous-graphe de $Del(P)$ constitué des arêtes entièrement incluses dans le β-voisinage de leurs arêtes duales dans $Vor(P)$:*

$$\beta\text{-}LC(P) := \{p_i p_j \in Del(P) \ : \ [p_i, p_j] \subset VN_\beta(v_i v_j), \ v_i v_j = dual(p_i p_j)\},$$

où $[p_i, p_j]$ est le segment de droite associé à l'arête $p_i p_j$, et $v_i v_j$ est l'arête duale de $p_i p_j$ dans $Vor(P)$. L'ensemble des arêtes de $Vor(P)$, dont l'arête duale n'appartient pas à β-$LC(P)$, constitue le β-axe médian de P, noté β-$AM(P)$:

$$\beta\text{-}AM(P) := \{v_i v_j \in Vor(P) \ : \ [p_i, p_j] \not\subset VN_\beta(v_i v_j), \ p_i p_j = dual(v_i v_j)\}.$$

En considérant l'ensemble des paires d'arêtes duales $(p_i p_j, v_i v_j)$, avec $p_i p_j \in Del(P)$ et $v_i v_j \in Vor(P)$, la définition précédente consiste à classer ces paires en deux catégories, selon un test local d'inclusion. Ce test local est illustrée à la figure 2.17 pour différentes valeurs de β. On peut observer que certaines arêtes de $Del(P)$ ne peuvent jamais appartenir au β-Crust local. Par conséquent, l'arête duale de ces arêtes appartient toujours au β-axe médian (voir la figure 2.17(c) et la figure 2.17(d)). Ce sont les arêtes de $Del(P)$ qui n'appartiennent pas au graphe de Gabriel de P. Nous décrivons cette propriété dans la section suivante.



(a) $\beta < 1$ $v_iv_j \in \beta\text{-}AM(P)$ (b) $\beta > 1$ $p_ip_j \in \beta\text{-}LC(P)$ (c) $\beta > 1$ $v_iv_j \in \beta\text{-}AM(P)$ (d) $\beta > 1$ $v_iv_j \in \beta\text{-}AM(P)$

FIG. 2.17: Une arête de $Del(P)$ appartient à β-$LC(P)$ si et seulement si elle est entièrement incluse dans le β-voisinage de son arête duale dans $Vor(P)$ (cas (b)). Sinon l'arête de $Vor(P)$ appartient à β-$AM(P)$ (cas (a), (c) et (d)). Une arête de $Del(P)$ qui n'appartient pas à $GG(P)$, n'appartient pas non plus à β-$LC(P)$ (cas (c) et (d)).

2.4.3 Relations avec le graphe de Gabriel

Le β-CRUST local de P est un sous-graphe du graphe de Gabriel de P, quelque soit la valeur prise par β.

Propriété 8 β-$LC(P) \subseteq GG(P)$, $\forall \beta \in]0, +\infty[$.

Preuve. Soit p_ip_j une arête de $Del(P)$ qui n'appartient pas à $GG(P)$. Montrons que p_ip_j n'appartient pas non plus à β-$LC(P)$. Pour cela, nous montrons qu'il existe au moins un point du segment $[p_i, p_j]$ qui n'est pas inclus dans $VN_\beta(v_iv_j)$. Soit p_{ij} le milieu du segment $[p_i, p_j]$. Comme $p_ip_j \notin GG(P)$, on a $]v_i, v_j[\cap\{p_{ij}\} = \emptyset$ (voir l'équation (2.4)). Étant donné que $]v_i, v_j[$ est toujours inclus dans $VN_\beta(v_iv_j)$, alors le point p_{ij} n'appartient pas à $VN_\beta(v_iv_j)$ et $[p_i, p_j] \not\subset VN_\beta(v_iv_j)$.$\square$

Cette relation avec le graphe de Gabriel permet de reformuler le test local qui intervient dans la définition du β-CRUST local et du β-axe médian de P.

Propriété 9 *Soit p_ip_j une arête de $Del(P)$ et v_iv_j son arête duale dans $Vor(P)$. Alors :*

$$p_ip_j \in \beta\text{-}LC(P) \Leftrightarrow p_ip_j \in GG(P) \text{ et } \{p_i, p_j\} \subset N_\beta(v_iv_j).$$

Par dualité, nous avons également :

$$v_iv_j \in \beta\text{-}AM(P) \Leftrightarrow v_iv_j \notin GG(P) \text{ ou } \{p_i, p_j\} \cap N_\beta(v_iv_j) = \emptyset.$$

Preuve. Soit p_ip_j une arête de $GG(P)$ telle que $\{p_i, p_j\} \subset N_\beta(v_iv_j)$ et v_iv_j bornée. Par définition de $GG(P)$, on a $[p_i, p_j]\cap]v_i, v_j[\neq \emptyset$ (voir l'équation (2.4)). Alors, pour tout $p \in [p_i, p_j]$, on a $\angle v_ipv_j \geq \angle v_ip_iv_j$ (et $\angle v_ipv_j \geq \angle v_ip_jv_j$). De plus, d'après l'expression (2.7) du β-voisinage de v_iv_j, on a $\angle v_ip_iv_j \geq \theta(\beta)$. Par conséquent, l'arête p_ip_j est entièrement incluse dans $N_\beta(v_iv_j) \setminus \{v_i, v_j\}$, ce qui est équivalent à $p_ip_j \in \beta$-$LC(P)$. Dans le cas où v_iv_j est non-bornée, il suffit d'utiliser l'expression (2.11) du β-voisinage. $\square$

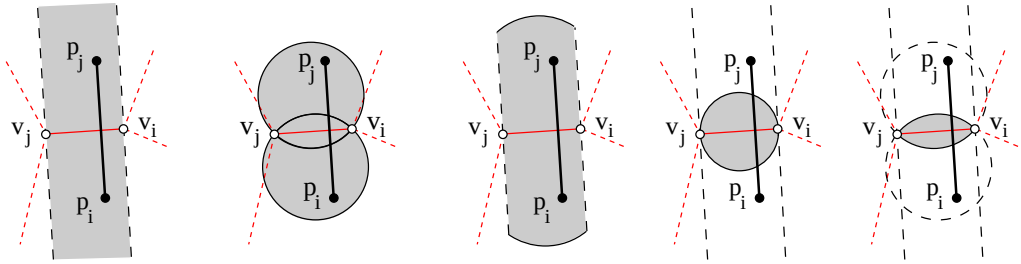
À partir des deux tests qui interviennent dans la Propriété 9, on peut redéfinir la forme du voisinage des arêtes de Voronoï. Considérons la figure 2.18(a). L'arête p_ip_j appartient au graphe de Gabriel si et seulement si elle est incluse dans la partie du plan définie par :

$$H(v_iv_j) := \{p \in \mathbb{R}^2 \ : \ \angle v_iv_jp < \pi/2 \text{ et } \angle v_jv_ip < \pi/2\}.$$

Alors, l'arête p_ip_j appartient à β-$LC(P)$ si et seulement si $\{p_i, p_j\} \subset H(v_iv_j) \cap N_\beta(v_iv_j)$ (voir la figure 2.18(c)). On peut également observer que lorsque $\beta \leq 1$, le β-voisinage se réduit à $N_\beta(v_iv_j)$.

Propriété 10 *Soit p_ip_j une arête de Del(P) et v_iv_j son arête duale dans Vor(P). Alors, pour tout $\beta \leq 1$, $p_ip_j \in \beta$-$LC(P) \Leftrightarrow N_\beta(v_iv_j) \cap \{p_i, p_j\} \neq \emptyset$.*

Preuve. Lorsque $\beta \leq 1$, on a $N_\beta(v_iv_j) \subset H(v_iv_j)$ (voir les figures 2.18(d) et 2.18(e)). □



(a) $H(v_iv_j)$ (b) $N_\beta(v_iv_j)$, $\beta > 1$ (c) $\beta > 1$ (d) $\beta = 1$ (e) $\beta < 1$

Fig. 2.18: Voisinage des arêtes de Voronoï. (a) $p_ip_j \in GG(P) \Leftrightarrow p_ip_j \subset H(v_iv_j)$. (b) β-voisinage. (c,d,e) $H(v_iv_j) \cap N_\beta(v_iv_j)$ pour $\beta > 1$, $\beta = 1$ et $\beta < 1$.

Il existe donc plusieurs façons de tester si une arête de Delaunay appartient à β-$LC(P)$ ou si une arête de Voronoï appartient à β-$AM(P)$. Jusqu'à présent, ce test fait intervenir à la fois la triangulation de Delaunay et le diagramme de Voronoï. Dans la section suivante, nous décrivons une propriété des β-Crusts locaux qui permet d'effectuer ce test uniquement à partir de la triangulation de Delaunay.

2.4.4 Propriété angulaire

Nous montrons que les β-Crusts locaux peuvent se définir alternativement en utilisant les angles des triangles de Delaunay. Soit $\theta \in]0, \pi[$ l'angle définit par :

$$\theta(\beta) := \begin{cases} \pi - \arcsin(\beta) & \text{si } \beta \in]0, 1], \\ \arcsin(1/\beta) & \text{si } \beta \in [1, +\infty[. \end{cases}$$

Propriété 11 *Soit p_ip_j une arête de Del(P). Soient $p_ip_jp_k$ et $p_ip_jp_l$ les deux triangles de Del(P) incidents à p_ip_j (un seul lorsque $p_ip_j \in conv(P)$). Alors, $\forall \beta \in]0, +\infty[$:*

$$p_ip_j \in \beta\text{-}LC(P) \Leftrightarrow \begin{cases} (a) & p_ip_j \in GG(P) & \text{et,} \\ (b1) & \angle p_ip_kp_j \leq \pi - \theta(\beta) & \text{si } p_ip_j \in conv(P), \\ (b2) & \angle p_ip_kp_j + \angle p_jp_lp_i \leq \pi - \theta(\beta) & \text{sinon.} \end{cases}$$

Preuve. Soit p_ip_j une arête de $Del(P)$ telle que $p_ip_j \notin conv(P)$. Soit v_iv_j l'arête duale de p_ip_j. Par dualité, les sommets v_i et v_j sont les centres des cercles circonscrits aux triangles $p_ip_kp_j$ et $p_ip_lp_j$ (voir la figure 2.19). D'après la formule de l'aire de $p_ip_kp_j$, on peut montrer que $\sin(\angle p_ip_kp_j) = \|p_i - p_j\|_2/2\|p_i - v_i\|_2$. D'autre part, comme p_ip_j est perpendiculaire à v_iv_j, on a $\sin(\angle p_iv_iv_j) = \|p_i - p_j\|_2/2\|p_i - v_i\|_2$. Donc $\angle p_iv_iv_j = \angle p_ip_kp_j$. De même, on obtient $\angle v_iv_jp_i = \angle p_jp_lp_i$. Donc $\angle p_iv_iv_j + \angle v_iv_jp_i = \angle p_ip_kp_j + \angle p_jp_lp_i$. Or, d'après l'expression (2.8) du β-voisinage, et la Propriété 9, on en déduit que $p_ip_j \in \beta$-$LC(P) \Leftrightarrow p_ip_j \in GG(P)$ et $\angle p_ip_kp_j + \angle p_jp_lp_i \leq \pi - \theta(\beta)$. Le cas des arêtes de l'enveloppe convexe est obtenu en posant $\angle p_jp_lp_i = 0$ (voir l'équation (2.11)). $\square$

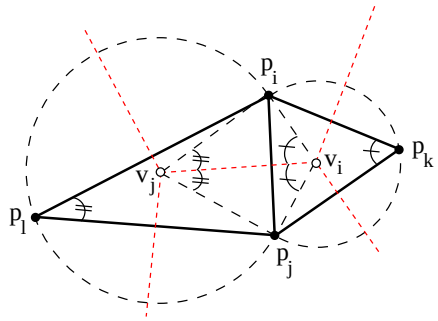


FIG. 2.19: Relations angulaires.

La Propriété 11 permet de concevoir un algorithme qui calcule le β-CRUST local directement à partir de la triangulation de Delaunay, particulièrement efficace lorsque le β-axe médian n'est pas souhaité (les sommets du diagramme de Voronoï n'ont pas besoin d'être calculés), et que la structure de données est la triangulation de Delaunay.

2.4.5 Algorithme général connaissant β

Étant donnés une valeur $\beta \in]0, +\infty[$ et un ensemble de points P, le calcul simultané de β-$LC(P)$ et de β-$AM(P)$ fonctionne en deux étapes, reportées dans l'Algorithme 1. Premièrement, la triangulation de Delaunay de P et le diagramme de Voronoï de P sont calculés. Puis, pour chaque arête de $Del(P)$ celle-ci est ajoutée à β-$LC(P)$ si elle respecte les conditions de la Propriété 11. Dans le cas contraire, son arête duale dans $Vor(P)$ est ajoutée à β-$AM(P)$.

Remarquons que cet algorithme peut être facilement modifié pour ne calculer que β-$LC(P)$ (suppression des lignes 9 et 14). Dans tous les cas, l'espérance du temps d'exécution de l'Algorithme 1 est $O(N \log N)$, où N est le nombre de points de P.

Des exemples de β-CRUSTS locaux et de β-axes médians d'un même ensemble de points sont montrés à la figure 2.20, pour différentes valeurs de $\beta \in]0, +\infty[$. Les points de l'ensemble P sont des échantillons d'une courbe fermée. On peut y observer que ces graphes décrivent deux familles hiérarchiques. En partant de zéro jusqu'à l'infini, les β-CRUSTS locaux extraient progressivement des portions de courbes de plus en plus

Algorithm 1 BetaLC(P, β)

Entrée: un ensemble fini de points $P \subset \mathbb{R}^2$ et une valeur $\beta \in]0, +\infty[$.
Sortie: β-$LC(P)$ et β-$AM(P)$.

1: $D \leftarrow Del(P)$ et $Vor(P)$
2: $LC, AM \leftarrow \emptyset$
3: $\alpha \leftarrow \pi - \theta(\beta)$
4: **pour chaque** arête $p_ip_j \in D$ **faire**
5: **si** $p_ip_j \in conv(P)$ **alors**
6: $p_k \leftarrow p \in P$ tel que le triangle $p_ip_jp \in D$
7: $\alpha_k \leftarrow \angle p_ip_kp_j$
8: **si** $(\alpha_k < \frac{\pi}{2})$ et $(\alpha_k \leq \alpha)$ **alors** $LC \leftarrow LC \cup \{p_ip_j\}$
9: **sinon** $AM \leftarrow AM \cup \{dual(p_ip_j)\}$
10: **sinon**
11: $p_k, p_l \leftarrow \{p \in P$ tel que le triangle $p_ip_jp \in D\}$
12: $\alpha_k \leftarrow \angle p_ip_kp_j$, $\alpha_l \leftarrow \angle p_ip_lp_j$
13: **si** $(\alpha_k < \frac{\pi}{2})$ et $(\alpha_l < \frac{\pi}{2})$ et $(\alpha_k + \alpha_l \leq \alpha)$ **alors** $LC \leftarrow LC \cup \{p_ip_j\}$
14: **sinon** $AM \leftarrow AM \cup \{dual(p_ip_j)\}$
15: **retourner** LC et AM

longues. Lorsque β est grand (tend vers l'infini), on retrouve le graphe de Gabriel (Propriété 8), qui contient généralement des arêtes non nécessaires à la reconstruction correcte de courbes. Simultanément, les β-axes médians coupent progressivement des arêtes du diagramme de Voronoï. Dans la suite, nous mettons en avant ces propriétés hiérarchiques, en particulier le lien entre le Crust local et le β-Crust local.

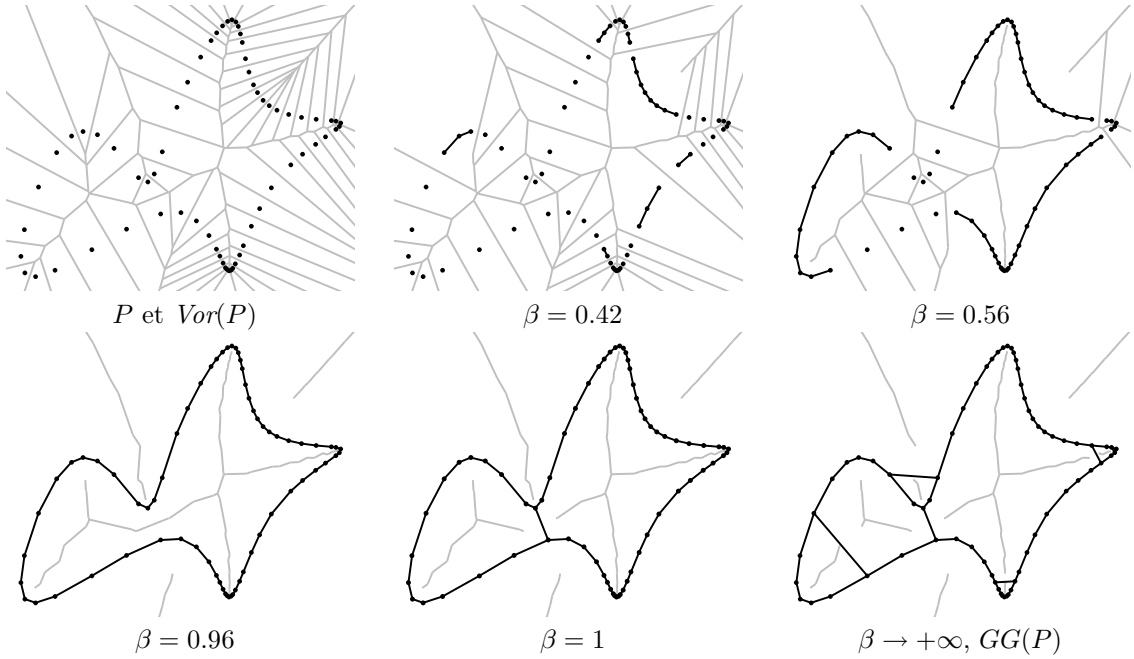


Fig. 2.20: Exemples de β-Crusts locaux (traits noirs) et de β-axes médians (traits gris).

2.5 Analyse des graphes proposés

Dans cette section, nous analysons les familles de β-CRUSTS locaux et de β-axes médians à travers leurs propriétés hiérarchiques. Ces propriétés permettent d'une part de relier les graphes proposés au CRUST local et à l'anti-CRUST [Gold, 1999], et d'autre part de concevoir des algorithmes gloutons de reconstruction de courbes simples (voir la section 2.6).

2.5.1 Propriétés hiérarchiques et combinatoires

Lorsque β varie dans $]0, +\infty[$, les β-CRUSTS locaux et les β-axes médians décrivent deux familles hiérarchiques de graphes. Ceci est dû à la relation d'inclusion qui existe entre les différents β-voisinages.

Propriété 12 *Quelque soit $\beta, \beta' \in]0, +\infty[$, nous avons :*

$$\begin{aligned} \beta < \beta' \quad &\Leftrightarrow \quad \beta\text{-}LC(P) \subseteq \beta'\text{-}LC(P) \subseteq GG(P) \subseteq Del(P), \\ &\Leftrightarrow \quad \beta'\text{-}MA(P) \subseteq \beta\text{-}AM(P) \subset Vor(P). \end{aligned}$$

Preuve. Soient p_ip_j une arête de β-$LC(P)$ et v_iv_j son arête duale. À partir de la Définition 1, nous avons $[p_i, p_j] \subset N_\beta(v_iv_j)$. Soit $\beta' > \beta$. Par définition du β-voisinage, nous avons $N_\beta(v_iv_j) \subset N_{\beta'}(v_iv_j)$, et donc $[p_i, p_j] \subset N_{\beta'}(v_iv_j)$. Ceci démontre la première équivalence. Pour la seconde, selon le même principe, si $[p_i, p_j] \not\subset N_{\beta'}(v_iv_j)$, alors nous avons $[p_i, p_j] \not\subset N_\beta(v_iv_j)$. $\square$

Les deux hiérarchies obtenues, et indexées par la valeurs de β, possèdent un nombre fini de niveaux. Autrement dit, étant donné un ensemble fini de points P, il existe un nombre fini de β-CRUSTS locaux et de β-axes médians de P, borné par le nombre d'arêtes du graphe de Gabriel de P (Propriété 8). En effet, pour chaque arête p_ip_j de $GG(P)$, il existe une valeur minimum $\beta \in]0, +\infty[$ telle que p_ip_j appartient à β-$LC(P)$, et telle que p_ip_j n'appartient pas à β'-$LC(P)$, pour tout $\beta' < \beta$. Par dualité, c'est également la valeur de β à partir de laquelle l'arête duale de p_ip_j est exclue de la famille des β-axes médians. Nous appelons cette valeur locale le poids des arêtes de $Del(P)$ ou de $Vor(P)$.

Définition 2 (poids des arêtes) *Soit (p_ip_j, v_iv_j) une paire d'arêtes duales telles que $p_ip_j \in Del(P)$ et $v_iv_j \in Vor(P)$. Le poids de la paire (p_ip_j, v_iv_j) est la valeur β_{ij} telle que les points p_i et p_j sont situés sur le bord du β_{ij}-voisinage de v_iv_j :*

$$\beta_{ij} := \begin{cases} \beta \ : \ \{p_i, p_j\} \subset \partial N_\beta(v_iv_j) & \text{si } p_ip_j \in GG(P), \\ +\infty & \text{sinon,} \end{cases}$$

où ∂N_β représente le bord du β-voisinage.

La figure 2.21 montre le β_{ij}-voisinage de plusieurs arêtes de Voronoï. D'après la Définition 1, une arête $p_ip_j \in Del(P)$ appartient à β_{ij}-$LC(P)$. De plus, induit de la Propriété 12, ce dernier est formé de toutes les arêtes de $Del(P)$ qui ont un poids inférieur ou égal

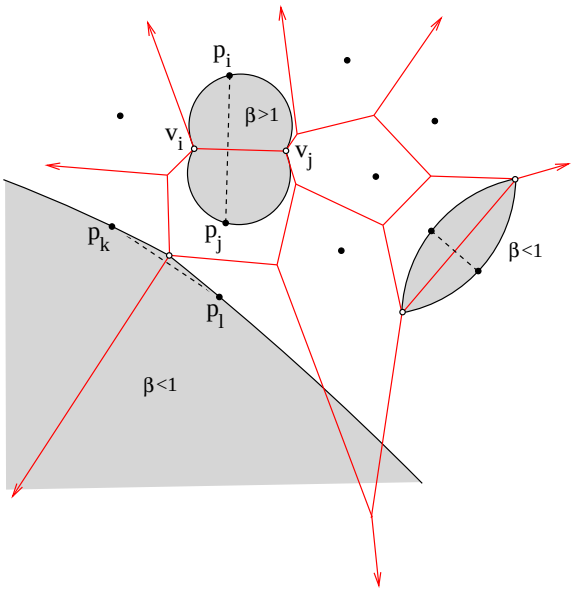


Fig. 2.21: Le voisinage limite des arêtes de Voronoï est défini tel que $\{p_i, p_j\} \subset \partial N_\beta(v_i v_j)$.

à β_{ij}. De façon similaire, le β_{ij}-axe médian de P est formé de toutes les arêtes de $Vor(P)$ dont le poids est strictement supérieur à β_{ij}. On peut alors définir une relation d'ordre sur les arêtes de $Del(P)$:

$$p_i p_j \leq p_k p_l \Leftrightarrow \beta_{ij} \leq \beta_{kl}. \tag{2.12}$$

Ainsi, l'ensemble des β-Crusts locaux et celui des β-axes médians de P peuvent être intégralement représentés par l'ensemble des arêtes de $Del(P)$, triées selon l'ordre croissant des poids. Nous appelons cet ensemble le spectre de P.

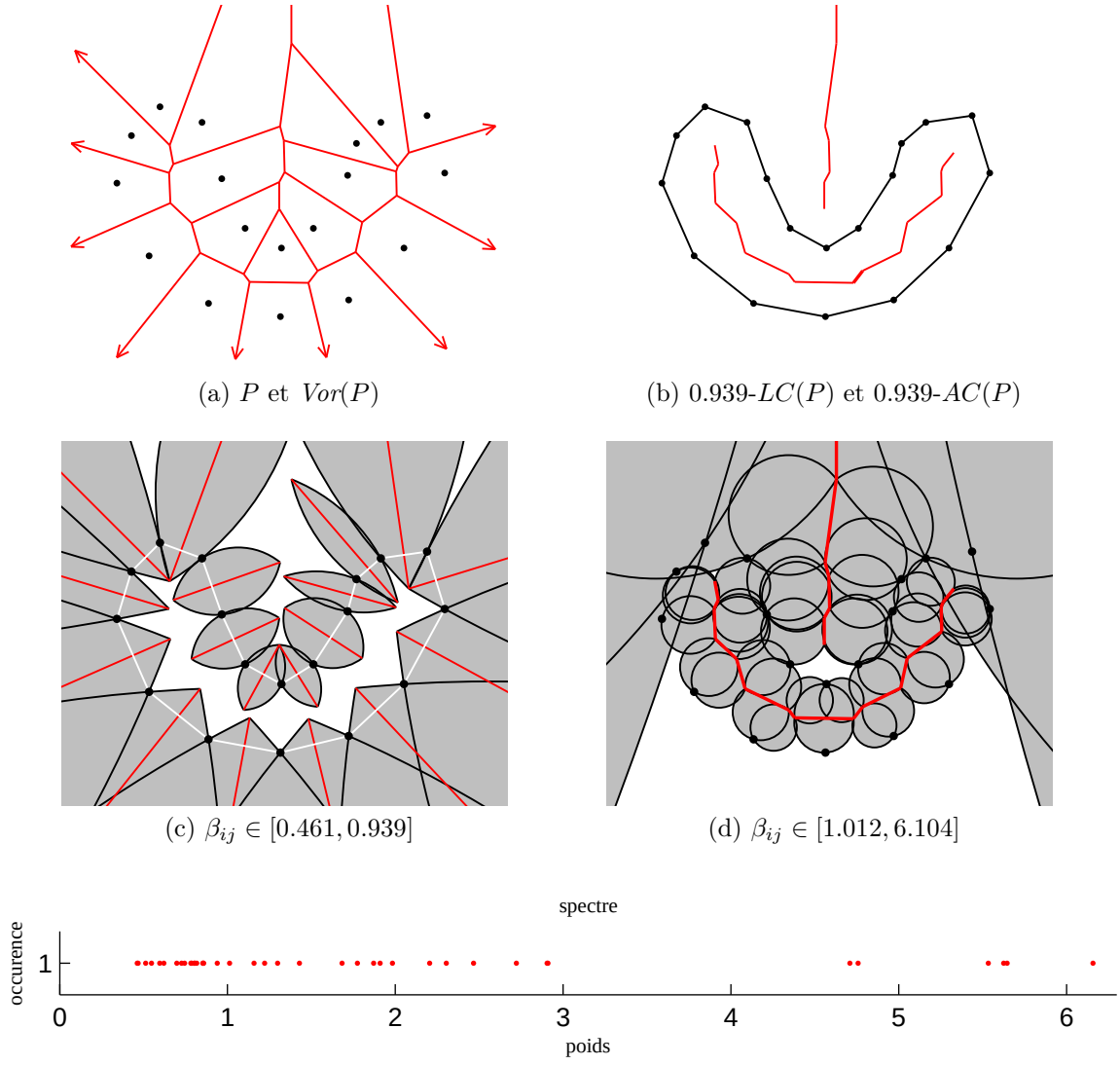
Définition 3 (spectre) *Le spectre de P est l'ensemble des arêtes de $Del(P)$, ordonnées selon la relation (2.12) :*

$$spectre(P) := \{e_1, \ldots, e_m \ : \ e_1 \leq \ldots \leq e_m,\ e_i \in Del(P),\ \forall i = 1, \ldots m\}.$$

Les valeurs minimum et maximum du poids des arêtes de $GG(P)$ sont notées $\beta_{\min}$ et $\beta_{\max}$. Elles correspondent respectivement au poids de l'arête e_1, et à celui de l'arête e_i pour laquelle le poids de toute arête $e_j > e_i$ vaut $+\infty$.

Remarquons que plusieurs arêtes peuvent avoir un poids identique. Dans ce cas, nous ne cherchons pas à les discriminer les unes des autres, et elles apparaissent dans le spectre de P dans un ordre quelconque.

Le spectre d'un ensemble de points est illustré à la figure 2.22. Nous pouvons observer que plus le poids des arêtes de la triangulation de Delaunay est faible, plus ces arêtes sont incluses dans la famille des β-Crusts locaux, et plus elles semblent appartenir à la reconstruction. Réciproquement, plus le poids des arêtes du diagramme de Voronoï est élevé, plus ces arêtes sont incluses dans la famille des β-axes médians, et moins elles semblent faire partie de la reconstruction. Trouver un β-Crust local qui reconstruit des



(a) P et $Vor(P)$

(b) 0.939-$LC(P)$ et 0.939-$AC(P)$

(c) $\beta_{ij} \in [0.461, 0.939]$

(d) $\beta_{ij} \in [1.012, 6.104]$

(f)

Fig. 2.22: Illustration du spectre. (a) Un ensemble de points. (b) Un β-Crust local qui reconstruit une courbe simple, et le β-axe médian associé. (d) Voisinage limite des arêtes de Voronoï dont les duaux appartiennent au β-Crust local. (e) Voisinage limite des arêtes de Voronoï qui appartiennent au β-axe médian. (f) Poids des arêtes (valeurs β_{ij} associées aux voisinages limites).

courbes consiste alors à séparer le spectre selon deux intervalles disjoints des valeurs des poids. Le premier intervalle, c.-à-d. $[\beta_{\min}, \beta]$, correspond aux arêtes de Delaunay qui appartiennent à la reconstruction (à la figure 2.22(f), remarquons la concentration des valeurs des poids pour $\beta_{ij} \leq 0.939$). Le second, c.-à-d. $]\beta, +\infty]$, correspond aux arêtes de Delaunay qui sont excluent de la reconstruction. Par dualité, ce dernier intervalle correspond aux arêtes de Voronoï qui approximent l'axe médian des courbes reconstruites. Une autre façon de visualiser le spectre d'un ensemble de points est illustré à la figure 2.23, où nous avons associé un niveau de gris au poids des arêtes. Plus l'arête est foncée, plus son poids est faible. De même qu'auparavant, les arêtes dont le poids est faible représentent la courbe à extraire.

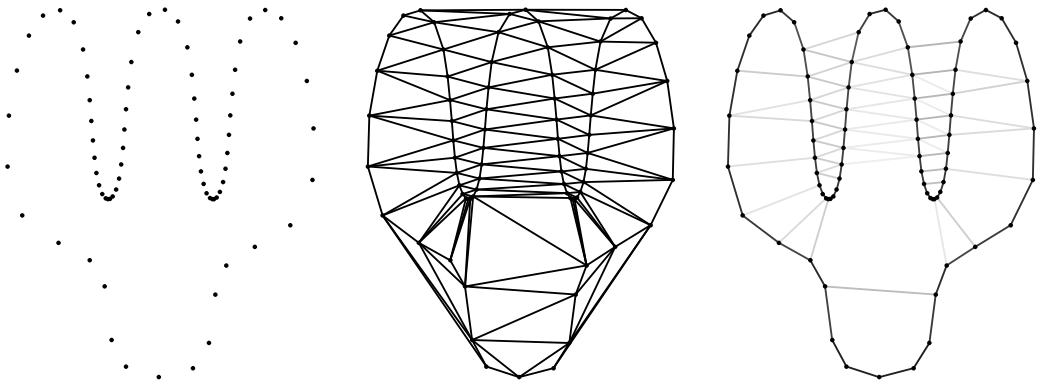


Fig. 2.23: Illustration du spectre d'un ensemble de points. Plus l'arête est foncée et plus son poids est faible. Ici $\beta_{\min} = 0.22$ et $\beta_{\max} = 4.34$.

En plus de permettre de définir un ordre sur les arêtes de Delaunay, les propriétés hiérarchiques (Propriété 12) montrent que le Crust local et l'anti-Crust sont des cas particuliers des graphes proposés.

2.5.2 Relations avec le Crust local et l'anti-Crust

Étant donnée une valeur de β, le β-Crust local est l'ensemble des arêtes de $Del(P)$ qui sont complètement incluses dans le β-voisinage de leurs duaux. Lorsque $\beta = 1$, le β-voisinage d'une arête v_iv_j de $Vor(P)$ est le disque de Gabriel $B(v_iv_j)$. Dans ce cas, la définition du β-axe médian correspond exactement à celle de l'anti-Crust (voir la Propriété 5 à la section 2.3.3). Par dualité, le 1-Crust local est le Crust local. À partir des propriétés hiérarchiques des β-Crusts locaux, nous montrons les différences entre les β-Crusts locaux et le Crust local.

Propriété 13 *Le* Crust *local et l'anti-*Crust *sont reliés à la famille des β-*Crusts *locaux et à celle de β-axes médians par les propriétés suivantes :*

(i) $LC(P) = 1\text{-}LC(P)$, $AC(P) = 1\text{-}AM(P)$.
(ii) $\forall\beta \in]0,1]$, $\beta\text{-}LC(P) \subseteq LC(P)$, $AC(P) \subseteq \beta\text{-}AM(P)$.
(iii) $\forall\beta \in [1,+\infty[$, $LC(P) \subseteq \beta\text{-}LC(P)$, $\beta\text{-}AM(P) \subseteq AC(P)$.

Preuve. La propriété (i) est directement obtenue à partir de la Propriété 5 et de la Définition 1. Les propriétés (ii) et (iii) sont déduites de la propriété (i) et de la Propriété 12. □

La Propriété 13(ii) montre que le Crust local contient tous les β-Crusts locaux pour lesquels $\beta \leq 1$. La Propriété 13(iii) montre que pour $\beta > 1$, les β-Crusts locaux sont au moins constitués des arêtes du Crust local. Dans certaines configurations, ceci permet d'améliorer les résultats fournit par le Crust local.

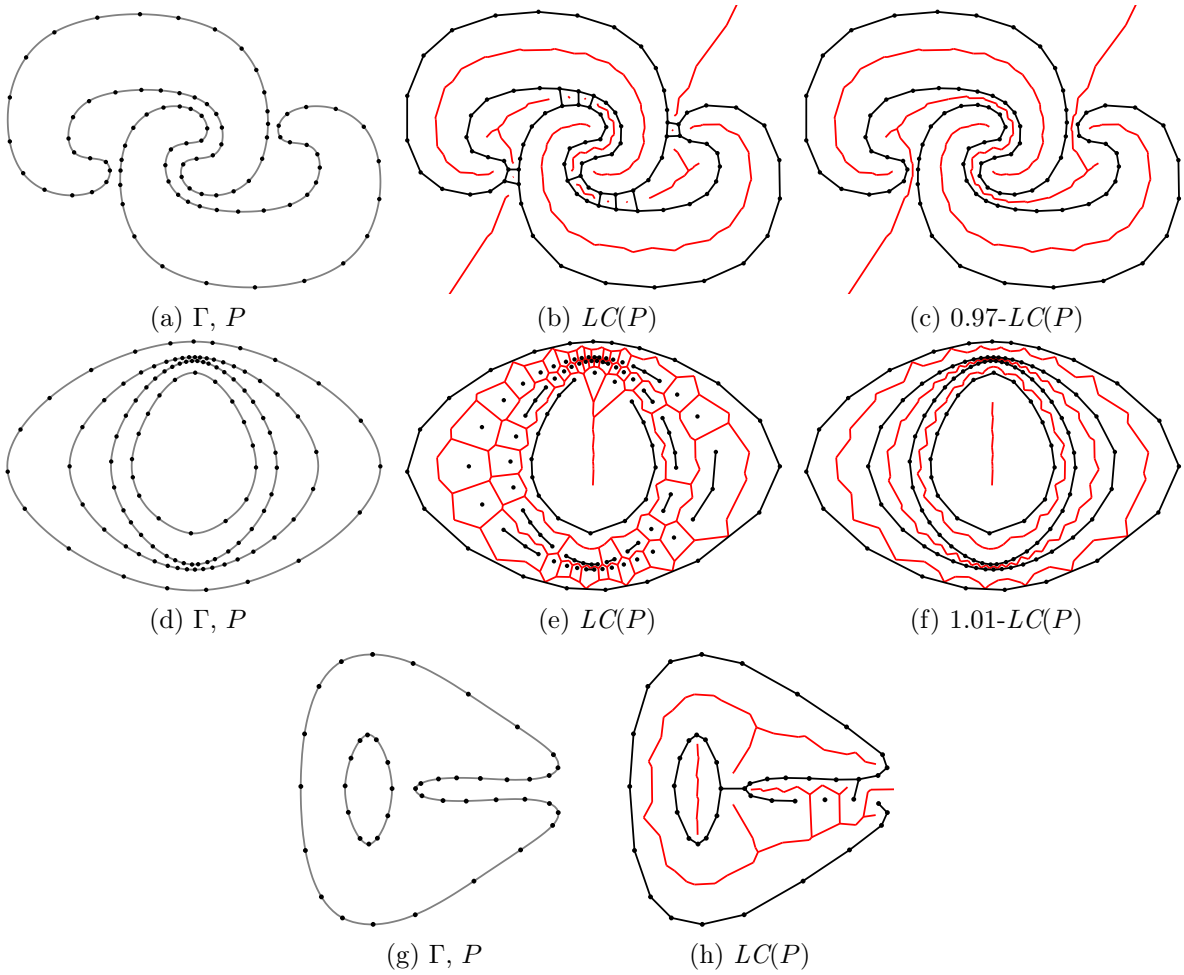


Fig. 2.24: Reconstructions à partir d'échantillons P de courbes Γ illustrant les trois configurations dans lesquelles le Crust local échoue (b,e,h). Dans les deux premières, il peut exister un β-Crust local qui reconstruit correctement les courbes (c,f).

Soit Γ une collection de courbes de $\mathbb{R}^2$ et P un échantillon quelconque de Γ. Comme illustré à la figure 2.24, le Crust local peut ne pas reconstruire correctement Γ à partir de P. Dans ce cas, trois configurations sont rencontrées :

- Toutes les arêtes de la reconstruction correcte de Γ sont présentes dans $LC(P)$ et au moins une arête est incorrecte (figure 2.24(b)).
- Aucune arête incorrecte n'est présente dans $LC(P)$ et il manque au moins une arête correcte (figure 2.24(e)).
- $LC(P)$ inclut à la fois des arêtes correctes et incorrectes (figure 2.24(h)).

Dans les deux premières configurations, il existe un β-Crust local capable de reconstruire correctement Γ si le poids de toutes les arêtes correctes est inférieur au poids des arêtes incorrectes. Dans la troisième configuration, il existe au moins une arête incorrecte ayant un poids inférieur au poids maximum des arêtes correctes. Il ne peut alors pas exis-

ter de β-CRUST local qui reconstruit correctement Γ. Ces problèmes sont discutés à la section 2.6 à travers un algorithme de reconstruction de courbes simples.
Les trois configurations précédentes d'échec peuvent arriver lorsque les conditions d'échantillonnage, associées au CRUST local (voir la section 2.3.3), ne sont pas satisfaites. En particulier, c'est le cas lorsque l'ensemble de points est de faible densité (les points sont très espacés les uns des autres). La figure 2.25 et la figure 2.26 illustrent d'autres exemples de reconstruction avec les β-CRUSTS locaux, qui diffèrent de la reconstruction obtenue avec le CRUST local. On peut également remarquer la différence avec le CRUST des plus proches voisins (NNC), le CRUST conservatif (CC), et GATHANG (voir la section 2.3).

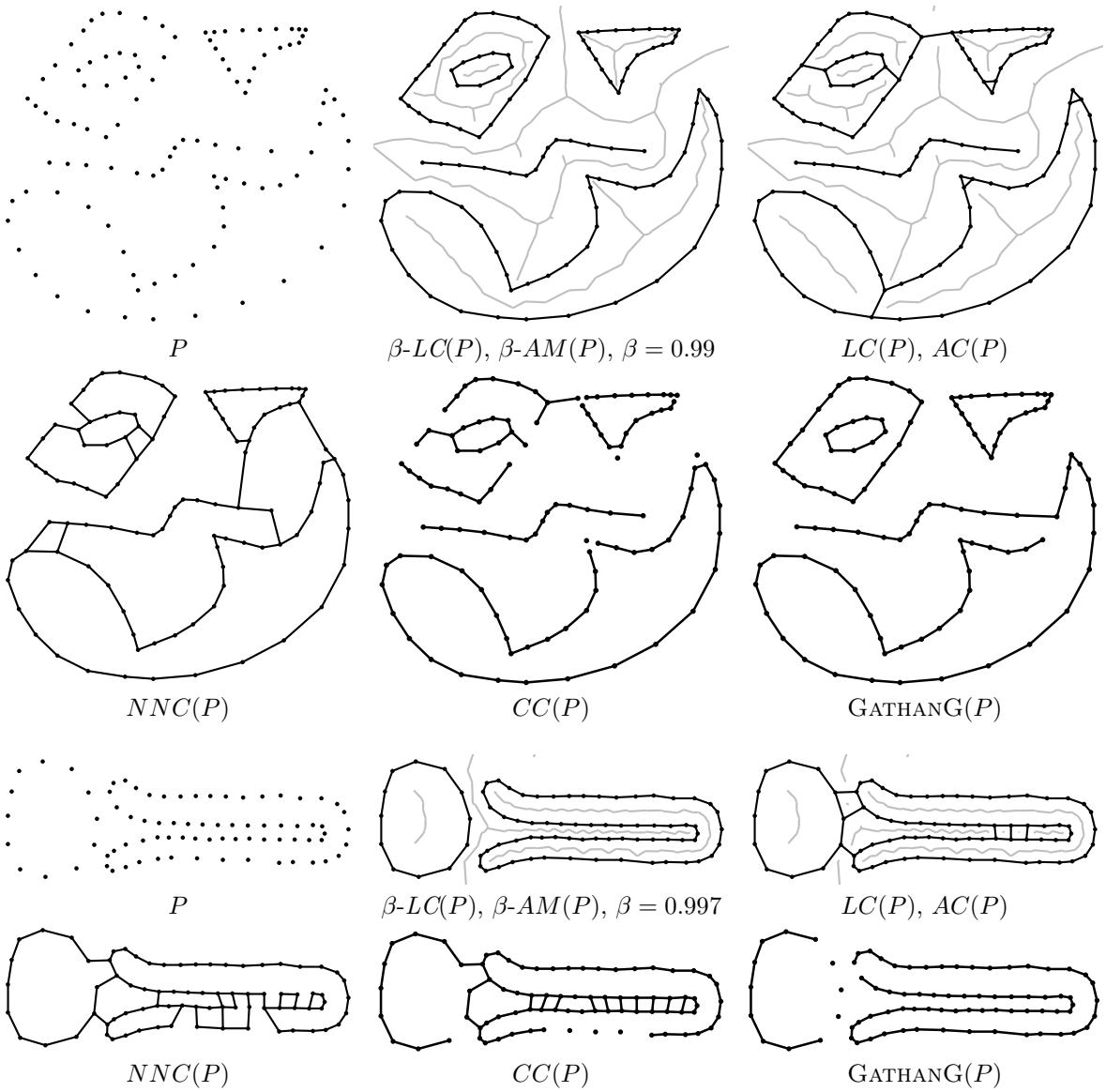


FIG. 2.25: Reconstruction de courbes avec les β-CRUSTS locaux et comparaison avec les CRUSTS classiques.

Les ensembles de points proviennent souvent de la discrétisation de contours bruités.

Dans ce cas, nous avons observé expérimentalement que les β-CRUSTS locaux permettent aussi d'améliorer les résultats obtenus avec le CRUST local. Ceci est illustré à la figure 2.27 pour un échantillon d'une collection de courbes régulières. À partir de l'échantillon non-bruité, toutes méthodes de reconstruction étudiées fournissent une reconstruction correcte. En revanche, en ajoutant un bruit normal, la qualité de la reconstruction se trouve généralement amoindrie. Le même comportement a été observé lorsque l'ensemble de points contient des points perturbateurs, comme illustré à la figure 2.28.

Nous ne décrivons pas de garantie de reconstruction pour les β-CRUSTS locaux. Néanmoins, pour $\beta \leq 1$, les β-CRUSTS locaux possèdent les mêmes garanties de reconstruction que le CRUST local. En particulier, il serait intéressant d'établir une relation entre la valeur de β, et celle de ϵ intervenant dans l'ϵ-échantillonnage. De plus, lorsque $\beta > 1$, les meilleurs résultats avec les β-CRUSTS locaux ont été obtenus pour β proche de 1. Intuitivement, pour reconstruire des courbes simples, la valeur maximum de β semble être proche de 1. Au-delà, des arêtes indésirables apparaissent dans la reconstruction alors que certaines arrêtes correctes peuvent manquer. Dans la section suivante, nous décrivons un algorithme, qui d'une part est capable de trouver automatiquement la valeur de β qui reconstruit des collections de courbes simples, et qui d'autre part améliore les résultats obtenus dans cette section.

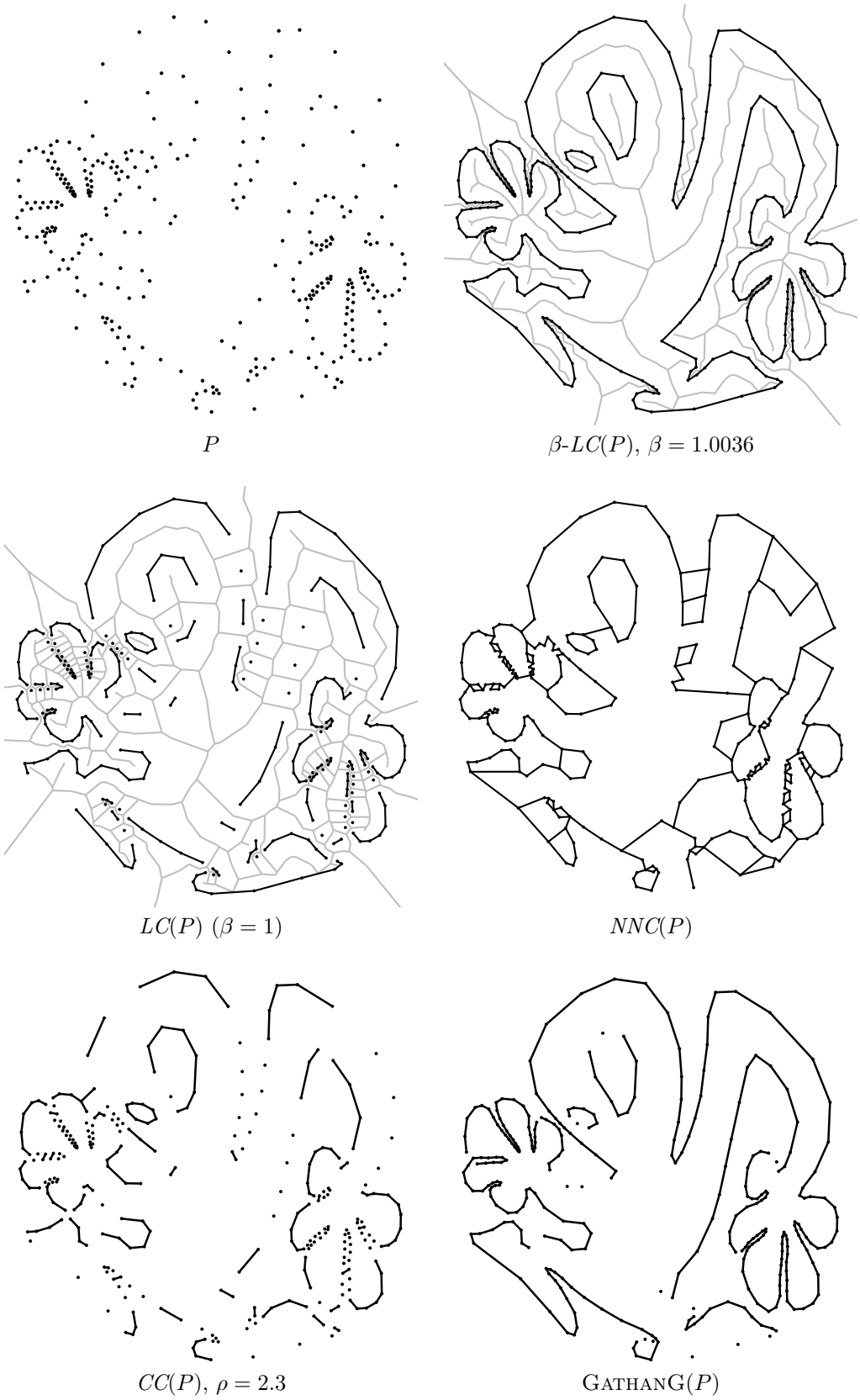


Fig. 2.26: Reconstruction avec les β-Crusts locaux et comparaison avec les Crusts classiques.

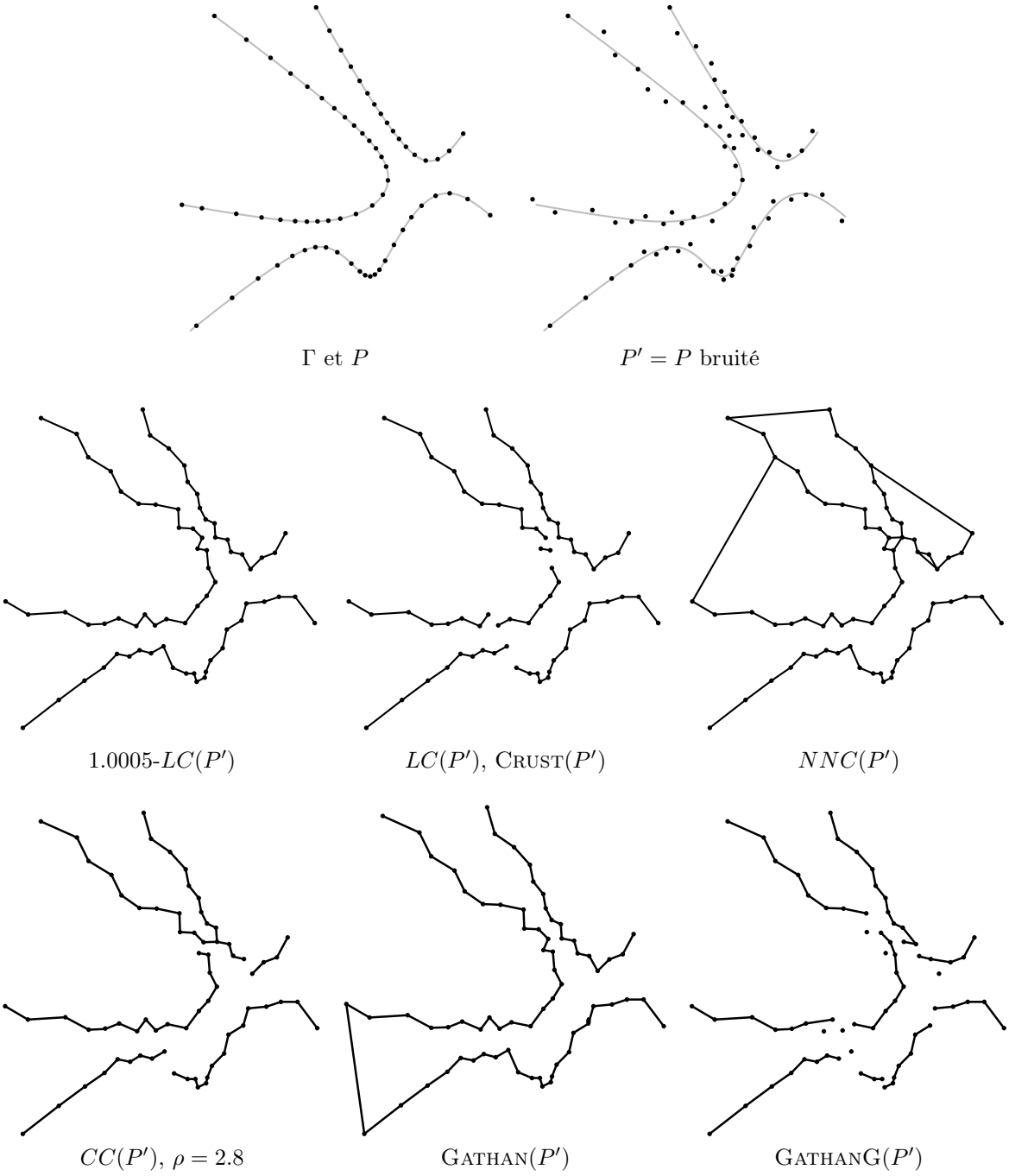


Fig. 2.27: Effet du bruit normal sur la reconstruction.

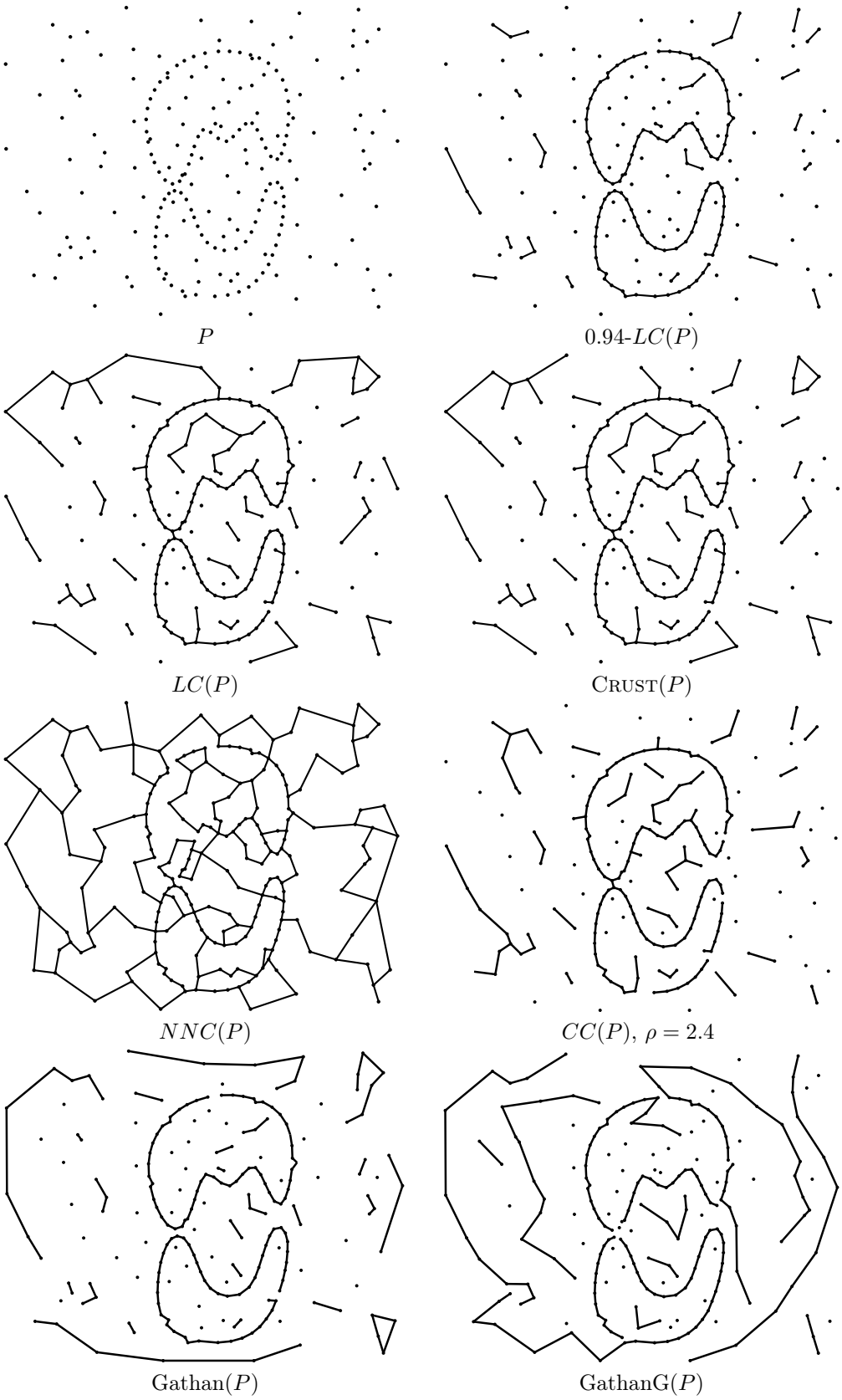


FIG. 2.28: Effet de points de bruit sur la reconstruction.

2.6 Algorithme glouton

Cette section présente un algorithme glouton pour extraire des collections de courbes simples à partir d'un ensemble de points. Pour cela, les propriétés hiérarchiques des β-CRUSTS locaux sont combinées aux propriétés topologiques des courbes à extraire. Cette approche peut être vue comme une extension des β-CRUSTS locaux. En particulier, lorsqu'il existe un β-CRUST local qui reconstruit correctement les courbes, l'algorithme permet de calculer automatiquement ce dernier.

2.6.1 Description de l'algorithme

Soit $P \subset \mathbb{R}^2$ un ensemble de points. La famille des β-CRUSTS locaux de P revient à ordonner les arêtes de la triangulation de Delaunay de P selon une relation géométrique entre les arêtes de Delaunay et leurs arêtes duales. Cette relation consiste à attribuer un poids à chaque arête de $Del(P)$, qui incorpore plusieurs caractéristiques : la longueur de l'arête de $Del(P)$ et celle de son arête duale, ainsi que les positions mutuelles de ces arêtes. Lorsque l'ensemble P est un échantillon de courbes simples, nous avons observé que plus la valeur de ce poids est faible et plus l'arête est candidate à la reconstruction correcte des courbes. Pour reconstruire les courbes, il est alors naturel de parcourir l'ensemble ordonné des arêtes de $Del(P)$ et d'insérer progressivement ces arêtes à la reconstruction si elles n'introduisent pas une courbe à points multiples. Cette condition d'insertion garantit que toutes les courbes polygonales résultant du parcours sont simples.

La méthode ci-dessus, que nous nommons TOPOCRUST(P), nécessite d'ordonner les arêtes de $Del(P)$ selon la hiérarchie des β-CRUSTS locaux de P. Cette première étape consiste à calculer le spectre de P (voir la section 2.5.1).

Calcul du spectre. Le spectre de P est représenté par la liste :

$$L := (e_1, \ldots, e_m \ : \ w(e_1) \leq \ldots \leq w(e_m), \ e_q \in Del(P)),$$

où le poids des arêtes $e_q = p_i p_j$ (Définition 2) est calculé à partir de la Propriété 11 :

$$w(p_i p_j) := \begin{cases} \angle p_i p_k p_j & \text{si } \angle p_i p_k p_j < \pi/2 \text{ et } p_i p_j \in conv(P) \\ \angle p_i p_k p_j + \angle p_j p_l p_i & \text{si } \angle p_i p_k p_j < \pi/2 \text{ et } \angle p_j p_l p_i < \pi/2 \text{ et } p_i p_j \notin conv(P) \\ +\infty & \text{sinon.} \end{cases}$$

Les sommets p_k et p_l sont les autres sommets des triangles incidents à l'arête $p_i p_j$. Remarquons que la valeur β_{ij} de la Définition 2 peut être obtenue avec l'équation :

$$\beta_{ij}(w(p_i p_j)) = \begin{cases} 1/\sin(\pi - w(p_i p_j)) & \text{si } w(p_i p_j) > \pi/2 \\ \sin(\pi - w(p_i p_j)) & \text{sinon.} \end{cases} \tag{2.13}$$

Le calcul du spectre est détaillé dans l'Algorithme 2. Il consiste à parcourir l'ensemble des arêtes de $Del(P)$ dans un ordre quelconque. Pour chaque arête parcourue, son poids est calculé, et l'arête est ajoutée à la liste L. Une fois l'ensemble des arêtes parcourues,

la liste L est triée dans l'ordre croissant des poids. Le parcours des arêtes de $Del(P)$, ainsi que le tri, ont une espérance d'exécution de $O(N \log N)$, où N est le nombre de points de P.

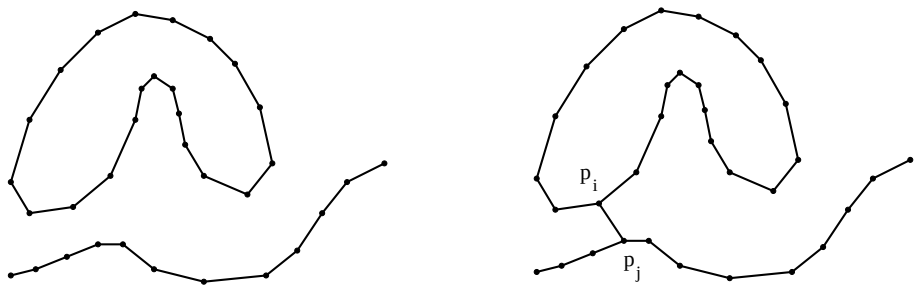
Algorithm 2 SPECTRE(P, D)

Entrée: la triangulation de Delaunay D de l'ensemble fini de points $P \subset \mathbb{R}^2$.
Sortie: la liste $L = \{e_1, \ldots, e_m\}$ ordonnée telle que $w(e_1) \leq \ldots \leq w(e_m)$.

1: $L \leftarrow \emptyset$
2: **pour chaque** arête $p_ip_j \in D$ **faire**
3: **si** $p_ip_j \in conv(P)$ **alors**
4: $p_k \leftarrow p \in P$ tel que le triangle $p_ip_jp \in D$
5: **si** $\angle p_ip_kp_j < \pi/2$ **alors**
6: $w(p_ip_j) \leftarrow \angle p_ip_kp_j$
7: **sinon** $w(p_ip_j) \leftarrow +\infty$
8: **sinon**
9: $p_k, p_l \leftarrow \{p \in P$ tel que le triangle $p_ip_jp \in D\}$
10: **si** $\angle p_ip_kp_j < \pi/2$ et $\angle p_jp_lp_i < \pi/2$ **alors**
11: $w(p_ip_j) \leftarrow \angle p_ip_kp_j + \angle p_ip_lp_i$
12: **sinon** $w(p_ip_j) \leftarrow +\infty$
13: ajouter l'arête p_ip_j à la fin de L
14: trier $L = \{e_1, \ldots, e_m\}$ telle que $w(e_1) \leq \ldots \leq w(e_m)$
15: **retourner** L

Détail de l'Algorithme TOPOCRUST(P). Une fois le spectre calculé, la méthode de reconstruction consiste à générer progressivement un graphe $G := (P, E)$ en parcourant le spectre de P. L'ensemble E de ses arêtes est initialement vide. Rappelons que la ***valence*** d'un sommet p_i de G, notée $\delta(p_i, G)$, est le nombre d'arêtes de E incidentes à ce sommet. Elle mesure la topologie du graphe en un sommet. Initialement, comme $E = \emptyset$, on a $\delta(p_i, G) = 0$ pour tout $p_i \in P$. Les arêtes de $Del(P)$ sont insérées à G en respectant la condition suivante (voir la figure 2.29) :

(Condition topologique) Le graphe G représente une collection de courbes simples, ouvertes ou fermées, si et seulement si $\delta(p_i, G) \leq 2$ pour tout $p_i \in P$.



(a) condition topologique satisfaite (b) condition topologique non-satisfaite

FIG. 2.29: (a) Tous les sommets sont de valence au plus 2. (b) Les sommets p_i et p_j sont de valence 3.

Ainsi, au cours du parcours du spectre, une arête est insérée à G si et seulement si elle appartient au graphe de Gabriel de P (son poids est inférieur à $+\infty$), et la valence de chacun de ses sommets est strictement inférieure à deux. Si c'est le cas, la valence de ses sommets est incrémentée de un. Lorsque toutes les arêtes du spectre ont été parcourues, le graphe G représente alors un ou plusieurs polygones simples, fermés (tous les sommets sont de valence deux) ou ouvertes (deux sommets sont de valence un).

Simultanément à l'insertion des arêtes de $Del(P)$ à G, il est également possible d'extraire les arêtes de $Vor(P)$ qui approximent l'axe médian des courbes reconstruites. Ce sont les arêtes duales des arêtes de $Del(P)$ qui n'appartiennent pas au graphe de Gabriel de P (celles dont le poids vaut $+\infty$), ou qui ne respectent pas la condition topologique d'insertion. L'ensemble de ces arêtes forme un graphe G'. Par construction, les arêtes du graphe G et celles du graphe G' ne s'intersectent jamais. La méthode complète est détaillée dans l'Algorithme 3. Son espérance d'exécution est de $O(N \log N)$.

Algorithm 3 TopoCrust(P)

Entrée: un ensemble fini de points $P \subset \mathbb{R}^2$.

Sortie: le graphe $G := (V, E)$ correspondant aux courbes, et le graphe $G' := (V', E')$ correspondant à l'approximation de l'axe médian des courbes.

1: $V \leftarrow P$
2: $E, V', E' \leftarrow \emptyset$
3: $D \leftarrow Del(P)$
4: $L \leftarrow$ Spectre(P, D)
5: $pq \leftarrow$ 1er élément de L
6: **tant que** $pq \neq$ NULL **faire**
7: **si** $(w(pq) < +\infty)$ et $(\delta(p, G) < 2)$ et $(\delta(q, G) < 2)$ **alors**
8: $E \leftarrow E \cup \{pq\}$
9: $\delta(p, G) \leftarrow \delta(p, G) + 1$
10: $\delta(q, G) \leftarrow \delta(q, G) + 1$
11: **sinon** $E' \leftarrow E' \cup \{dual(pq)\}$
12: $pq \leftarrow$ suivant(pq, L)
13: **retourner** G et G'

2.6.2 Différences avec les β-Crusts locaux

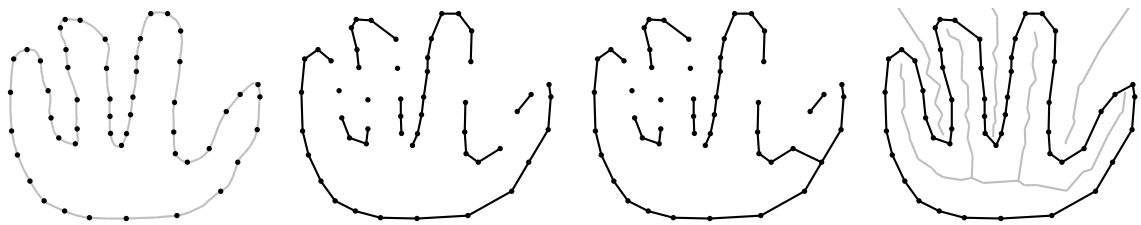
Dans de nombreux cas, les graphes issus de l'Algorithme 3 permettent d'extraire exactement un β-Crust local et un β-axe médian. Soient Γ une collection de courbes simples et P un échantillon quelconque de ces courbes. Lorsque toutes les arêtes de la reconstruction correcte de Γ ont un poids inférieur à celui des arêtes incorrectes, le graphe G calculé par l'Algorithme 3 est un β-Crust local de P. Les valeurs de β qui permettent de reconstruire correctement Γ sont supérieures au poids de la dernière arête insérée à G, et strictement inférieures au poids de la première arête qui ne garantit pas la condition

topologique. Soient β_{c1} et β_{c2} ces deux valeurs. Pour tout $\beta' \geq \beta_{c2}$, les β'-CRUSTS locaux comportent des points multiples (sommets de valence 3). L'Algorithme 3 peut donc être vu comme une méthode pour calculer automatiquement la valeur maximum de β qui permet de reconstruire une collection de courbes simples (fermées ou ouvertes). Pour cela, il suffit d'arrêté le parcours du spectre lorsque la condition topologique n'est pas satisfaite (ligne 11). Les duaux des arêtes non parcourues constituent alors le β-axe médian de P.

Au contraire, lorsqu'il existe au moins une arête incorrecte ayant un poids inférieur au poids maximum des arêtes correctes, il n'existe pas de β-CRUST local qui peut reconstruire correctement Γ (voir la figure 2.30). Avec l'Algorithme 3, si l'un des sommets de cette arête est déjà connecté à deux autres sommets, l'arête n'est pas ajoutée à la reconstruction (son arête duale est ajoutée à l'axe médian). On en déduit alors la relation suivante :

$$\beta\text{-}LC(P) \subseteq \textsc{TopoCrust}(P), \quad \forall \beta \in]0, \beta_{c2}[. \tag{2.14}$$

Cette différence permet d'améliorer les résultats obtenus avec les β-CRUSTS locaux. Ceci est illustré dans la prochaine section.



Γ, P β_{c1}-$LC(P)$, $\beta_{c1} = 0.98$ β_{c2}-$LC(P)$, $\beta_{c2} = 0.99$ TOPOCRUST(P)

FIG. 2.30: Différence entre les β-CRUSTS locaux et l'Algorithme 3 (TOPOCRUST(P)).

2.6.3 Exemples de reconstruction

L'algorithme de reconstruction TOPOCRUST est particulièrement efficace pour reconstruire des échantillons de courbes régulières fermées. Soient Γ une collection de courbes régulières fermées et P un échantillon de Γ. Comme illustré à la figure 2.31(a), si P est un échantillon suffisamment dense de Γ, le poids des arêtes de la reconstruction correcte de Γ est très faible relativement à celui des arêtes incorrectes. Dans ce cas, TOPOCRUST(P) fournit généralement la reconstruction correcte. C'est également le cas des méthodes de reconstruction analysées à la section 2.3. En particulier, à partir de la relation (2.14), on peut observer que TOPOCRUST reconstruit correctement Γ si P respecte les conditions d'échantillonnage associées au CRUST local (voir la section 2.3.3).

Comme illustré à la figure 2.31(b), lorsque l'échantillon P est plus éparse, le poids des arêtes de la reconstruction correcte se distingue difficilement du poids des arêtes incorrectes. Ceci arrive par exemple si P ne satisfait pas les conditions d'échantillonnage associées au CRUST local. Dans ce cas, nous avons observé que TOPOCRUST fournit

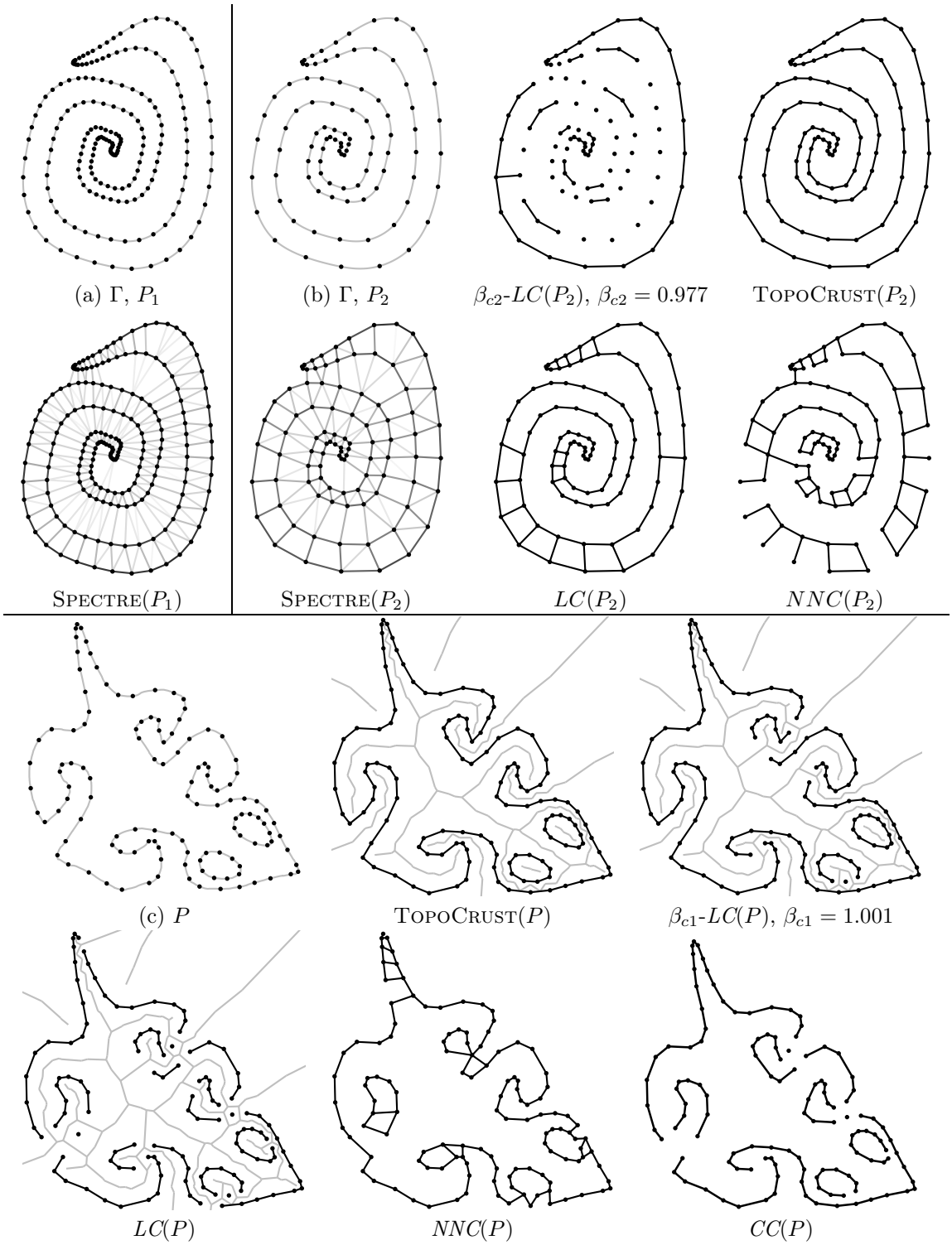


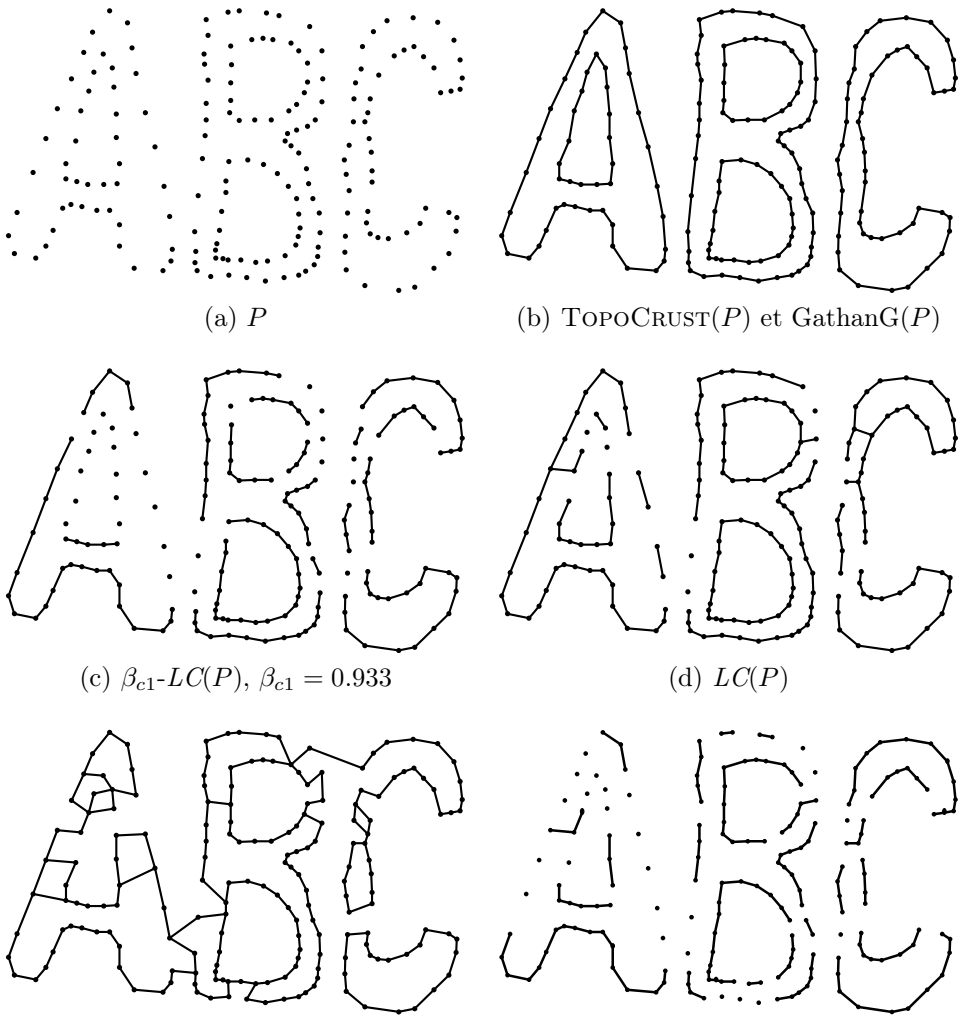
Fig. 2.31: Reconstruction de courbes régulières avec TopoCrust. Comparaison avec le meilleur β-Crust local et les Crusts classiques.

généralement de meilleurs résultats que le β-Crust local ou les méthodes de reconstruction analysées à la section 2.3. Un autre exemple de reconstruction correcte avec

TopoCrust est illustré à la figure 2.31(c). L'échantillon P est ici quelconque. Une comparaison plus concise des méthodes de reconstruction est effectuée à la section 2.7 à travers l'ϵ-échantillonnage.

L'algorithme TopoCrust permet plus généralement d'extraire des courbes polygonales simples de la triangulation de Delaunay de P, quelque soit la nature et la densité de P. Un exemple est illustré à la figure 2.32. On peut observer que TopoCrust fournit le même résultat que l'algorithme GathanG. En pratique, nous avons constaté que ces deux algorithmes donnent souvent des résultats similaires, lorsque les courbes à reconstruire ne comportent pas d'angles aigus. Dans le cas contraire, illustré à la figure 2.33, l'algorithme TopoCrust ne prend généralement pas en compte les sommets qui correspondent à un coin. Notons que les algorithmes Gathan et GathanG, conçus pour reconstruire des courbes avec des angles aigus, ne fournissent pas nécessairement des résultats satisfaisants.

Dans la section suivante, nous expérimentons l'efficacité de l'algorithme TopoCrust sur plusieurs ϵ-échantillons de courbes régulières.



(a) P (b) TopoCrust(P) et GathanG(P)

(c) β_{c1}-$LC(P)$, $\beta_{c1} = 0.933$ (d) $LC(P)$

(e) $NNC(P)$ (f) $CC(P)$

Fig. 2.32: Reconstruction avec TopoCrust. Comparaison avec le meilleur β-Crust local (c), et les Crusts classiques (b,d,e,f).

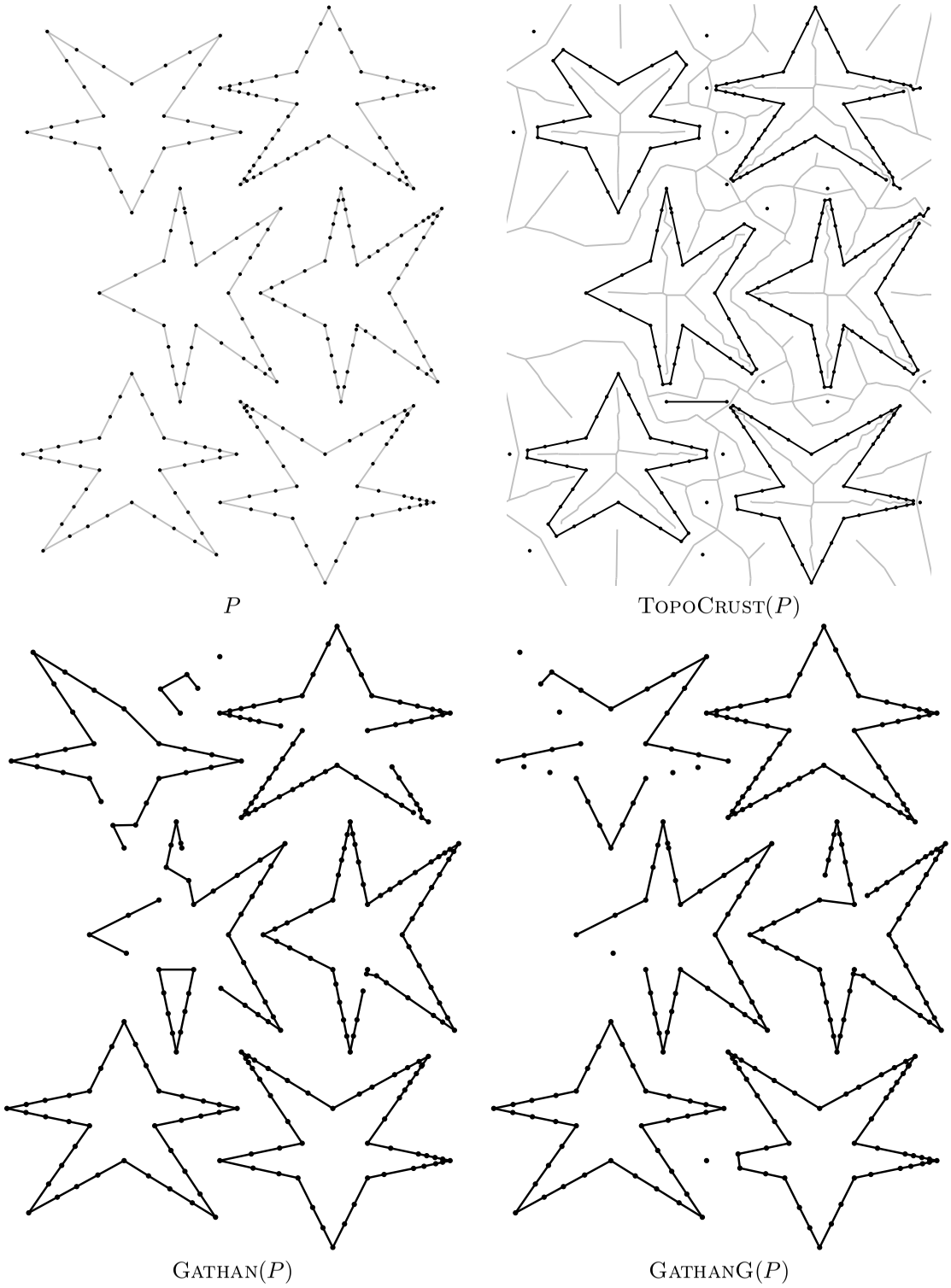


Fig. 2.33: Reconstruction de courbes simples non-régulières avec TopoCrust. Comparaison avec Gathan et GathanG.

2.7 Expérimentations basées sur l'ϵ-échantillonnage

À défaut d'avoir défini une condition d'échantillonnage qui permet de garantir la reconstruction correcte de courbes, nous avons étudié expérimentalement le comportement des β-CRUSTS LOCAUX et de l'algorithme TOPOCRUST sur des ϵ-échantillons. Nous comparons nos résultats aux algorithmes similaires, analysés à la section 2.3. Pour cela, nous disposons d'une base de courbes régulières, chacune accompagnée d'une approximation de son axe médian. Cette approximation permet de calculer le distance des points de la courbe à l'axe médian, nécessaire pour construire les ϵ-échantillons. Ensuite, pour une courbe donnée et un algorithme donné, nous recherchons expérimentalement la valeur maximum de $\epsilon \in]0, 1]$, telle que la reconstruction soit correcte. Les sections suivantes décrivent les étapes d'approximation de l'axe médian et de la construction d'ϵ-échantillons. Les résultats obtenus pour les différents algorithmes sont ensuite analysés.

2.7.1 Approximation de l'axe médian

Par définition, le calcul d'un ϵ-échantillon repose sur la distance de la courbe à son axe médian. Ceci nécessite de connaître l'expression de l'axe médian. En pratique, l'expression exacte de l'axe médian n'est connue que pour des cas simples. Sans être exhaustif, c'est le cas par exemple des segments de droites, des paraboles, des polygones, des arcs de cercles ou des unions finies de boules. Pour des courbes plus générales, l'axe médian est approximé à partir de l'axe médian d'une version simplifiée des courbes (voir [Attali *et al.*, 2007] pour un récent aperçu des méthodes). Deux approches sont usuellement considérées :

1. *Discrète* : Les courbes sont représentées par des images numériques binaires (images de pixels). L'axe médian nécessite alors d'être redéfini en terme de géométrie discrète. De ce fait, il ne partage pas les mêmes propriétés que l'axe médian continu que nous considérons dans cette section.
2. *Continue* : Les courbes sont approximées par des courbes polygonales (continues par morceaux), et l'axe médian est généralement approximé par un sous-graphe des arêtes du diagramme de Voronoï.

Ce sont les méthodes continues que nous proposons d'utiliser ici. Elles sont directement liées aux méthodes de reconstruction de courbes, considérées dans nos expérimentations. Rappelons que dans le cas de courbes régulières Γ, l'axe médian de Γ peut être approximé par un sous-ensemble des arêtes de Voronoï d'un échantillon dense de Γ [Brandt et Algazi, 1992], c.-à-d. le sous-ensemble des arêtes de Voronoï qui ne coupent pas Γ (voir la section 2.2.2).

Soit P_{dense} un échantillon dense de Γ. L'axe médian de Γ est alors approximé par l'ensemble :

$$M(\Gamma,\, P_{\text{dense}}) = \{v_i v_j \in \mathit{Vor}(P_{\text{dense}}) \,:\, [v_i, v_j] \cap \Gamma = \emptyset\}. \tag{2.15}$$

Les courbes Γ sont interpolées par l'ensemble des arêtes, de la triangulation de Delaunay de P_{dense}, dont les arêtes duales dans $Vor(P_{\text{dense}})$ coupent Γ, c.-à-d. à la triangulation de Delaunay de P_{dense} restreinte à Γ [Edelsbrunner et Shah, 1994]. Ce dernier graphe joue un rôle important dans les processus d'échantillonnage et d'approximation de courbes et de surfaces [Edelsbrunner, 2001, Boissonnat et Oudot, 2005, Dey, 2007].

L'échantillon dense peut être obtenu de différentes façons, selon la nature des courbes à approximer (équations paramétriques, équations algébriques). On peut considérer par exemple des méthodes d'interpolation polynomiale (polynômes de Legendre, splines, ...), et faire tendre le paramètre de discrétisation vers zéro. Dans la suite nous ne traitons que le cas des courbes sous forme paramétrique : $\gamma(t) = (x(t), y(t))$, avec $t \in [0, 1]$. Alors P_{dense} est généré en deux étapes :

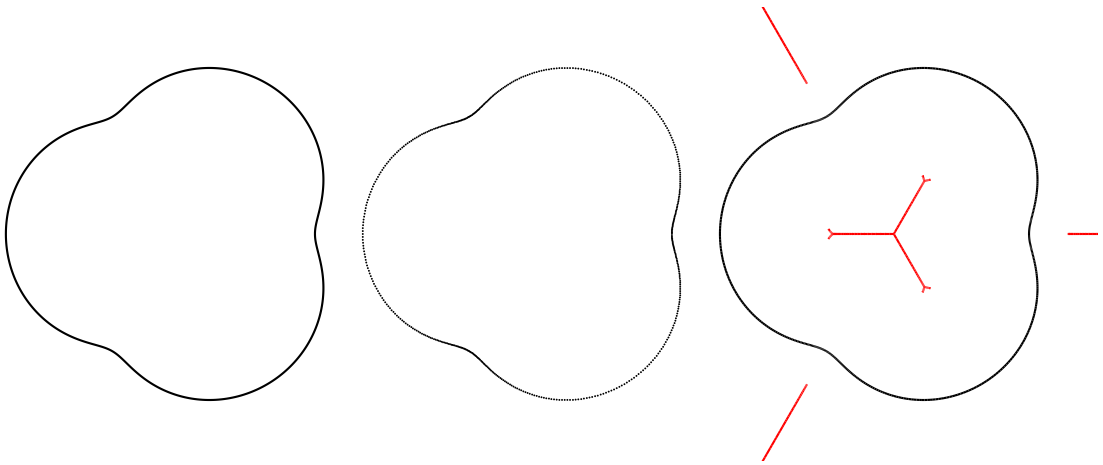
a. L'intervalle $[0, 1]$ est divisé en k sous-intervalles disjoints, insérés dans une liste L.
b. Tant que la liste L n'est pas vide, l'intervalle $[t_i, t_j]$ est retiré de L. Si la distance Euclidienne entre les points associés à t_i et à t_j vérifie

$$\|\gamma(t_i) - \gamma(t_j)\|_2 < r,$$

alors $[t_i, t_j]$ est divisé en deux sous-intervalles, qui sont ajoutés à L.

Un exemple de calcul d'un échantillon dense, et d'approximation de l'axe médian associé, sont illustrés à la figure 2.34. La courbe considérée est l'épicycloïde raccourcie :

$$\gamma(t) = (4\cos 2\pi t - 0.5\cos 8\pi t, 4\sin 2\pi t - 0.5\sin 8\pi t)$$



(a) Γ (b) P_{dense} (c) $M(\Gamma, P_{\text{dense}})$

Fig. 2.34: Approximation de l'axe médian d'une épicycloïde raccourcie.

Remarque. *Nous avons vu que lorsque $r \to 0$, l'approximation $M(\Gamma, P_{dense})$ converge vers l'axe médian de Γ (voir la section 2.2.2) [Brandt, 1994]. Néanmoins, du aux erreurs*

d'approximation numérique sur la position des échantillons, le paramètre r ne peut pas être trop faible. En effet, choisir r très proche de zéro peut introduire des arêtes indésirables à l'approximation de l'axe médian. D'autre part, l'erreur d'approximation de l'axe médian peut aussi être importante aux points terminaux de ses ramifications. Dans le contexte de notre expérimentation, la validité de $M(\Gamma, P_{dense})$ est contrôlée visuellement.

2.7.2 Estimation de la distance à l'axe médian

À partir de l'approximation de l'axe médian, la distance minimale à l'axe médian de Γ est estimée, en tout point $p \in \Gamma$, par la distance minimale de p au sous-ensemble de $\mathbb{R}^2$ couvert par l'union des arêtes de $M(\Gamma,\, P_{\text{dense}})$:

$$\varrho(p, AM(\Gamma)) \approx \varrho(p, M(\Gamma,\, P_{\text{dense}})) = \min_{[v_i,v_j]\in M} (\min_{v\in[v_i,v_j]} \|v-p\|_2). \tag{2.16}$$

Soit P un échantillon quelconque de Γ. Plutôt que de calculer la distance ϱ, nous estimons la quantité suivante en chaque point de Γ :

$$\epsilon(p, P, AM) \approx \epsilon(p, P, M) = \frac{\|p - NN(p, P)\|_2}{\varrho(p, M)}, \tag{2.17}$$

L'algorithme 4 détaille le calcul de cette estimation. Il parcourt l'ensemble des arêtes de $M(\Gamma,\, P_{\text{dense}})$, et pour chaque arête il calcule la distance minimale de p à un segment.

Algorithm 4 $\epsilon(p,\, p_i,\, M)$

Entrée: un point p d'une collection de courbes Γ, son plus proche voisin p_i dans l'échantillon, et la liste M des arêtes de l'approximation de l'axe médian.
Sortie: la densité locale de P en p, relativement à M.

1: $d_{\min} \leftarrow +\infty$
2: **si** $p \in P$ **alors**
3: **retourner** 0
4: **pour chaque** arête $v_iv_j \in M$ **faire**
5: **si** $\angle pv_iv_j \geq \frac{\pi}{2}$ **alors**
6: **si** $(d \leftarrow \|p - v_i\|_2) < d_{\min}$ **alors**
7: $d_{\min} \leftarrow d$
8: **sinon si** $\angle pv_jv_i \geq \frac{\pi}{2}$ **alors**
9: **si** $(d \leftarrow \|p - v_j\|_2) < d_{\min}$ **alors**
10: $d_{\min} \leftarrow d$
11: **sinon**
12: $q \leftarrow$ projeté orthogonal de p sur $[v_i, v_j]$
13: **si** $(d \leftarrow \|p - q\|_2) < d_{\min}$ **alors**
14: $d_{\min} \leftarrow d$
15: **retourner** $\|p - p_i\|_2/d_{\min}$

Cet algorithme a une espérance d'exécution en $O(m)$, où m est le nombre d'arêtes

de $M(\Gamma,\ P_{\text{dense}})$. Cette complexité peut être considérablement améliorée en réduisant le nombre d'arêtes de $M(\Gamma,\ P_{\text{dense}})$ parcourues (ligne 4). Par exemple, ceci peut être réalisé en utilisant la triangulation de Delaunay de P_{dense} (et son diagramme de Voronoï).

Un exemple d'estimation est illustré à la figure 2.35 pour un 0.6-échantillon de l'épicycloïde raccourcie. On peut observer que le graphe de la fonction ϵ, introduite par l'équation (2.17), peut être vu comme une signature de la courbe. Cet aspect n'est pas développé ici, et fera l'objet de futures investigations.

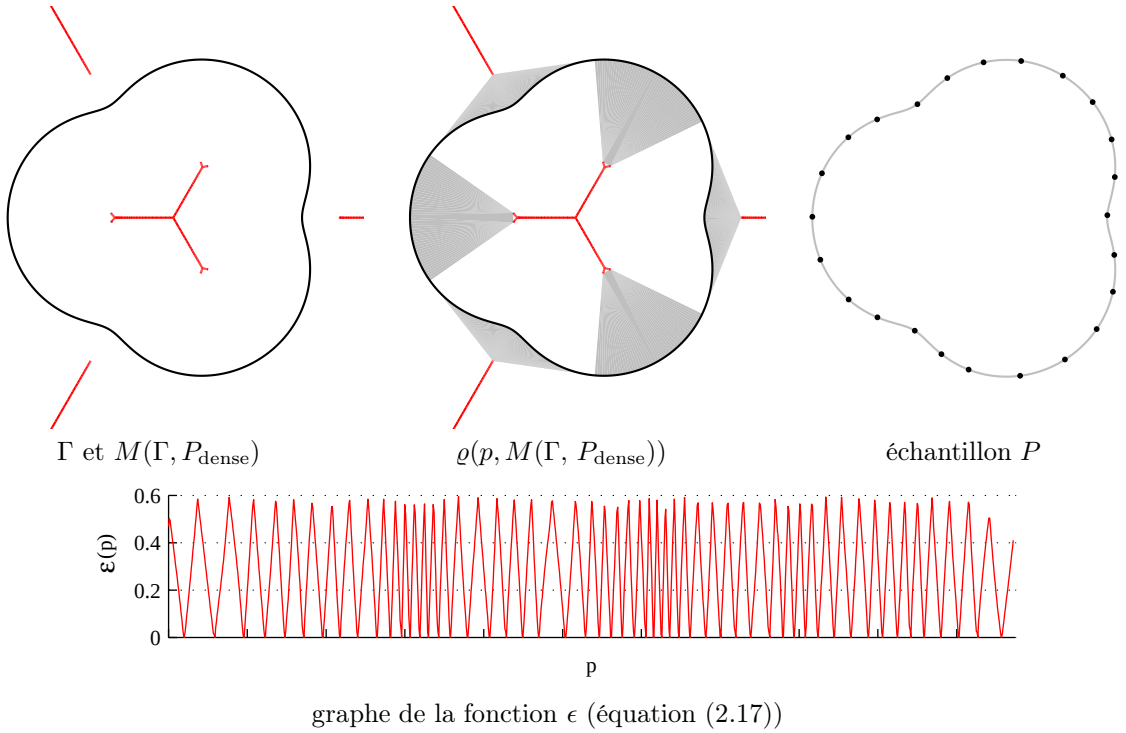


Fig. 2.35: Approximation de la distance à l'axe médian et estimation de la fonction ϵ aux points de Γ.

2.7.3 Calcul d'ϵ-échantillons

Soient $\Gamma : \gamma(t) = (x(t), y(t))$ une courbe régulière fermée, et $M(\Gamma, P_{\text{dense}})$ l'approximation de son axe médian. Soit $0 < \epsilon_0 < 1$ une valeur donnée. Pour calculer un ϵ-échantillon de Γ, nous utilisons une méthode classique, qui a déjà prouvé son efficacité pour échantillonner des contours discrets (voir par exemple [Jaillet *et al.*, 2000]). Le cas des courbes régulières ouvertes, et celui d'une collection de courbes, peuvent être traités en modifiant légèrement la méthode.

Soit $P = \{p_0\}$ un échantillon initial de Γ, constitué d'un point $p_0 \in \Gamma$ ($\gamma(0)$ par exemple). Le processus d'échantillonnage consiste alors à ajouter itérativement des points à l'ensemble P, en parcourant la courbe Γ à partir de p_0, avec un pas de discrétisation donné.

Soit $p_i \in P$ le point courant (initialement le p_0). Une itération commence par trouver le point x de Γ au-delà duquel la condition d'échantillonnage :

$$\varrho(x, M) \leq \epsilon_0 \|x - p_i\|_2$$

n'est plus satisfaite (Algorithme 4). Le point suivant est ensuite obtenu en recherchant le point $p_{i+1} \in \Gamma$ au-delà duquel la condition :

$$\varrho(x, M) \leq \epsilon_0 \|x - p_{i+1}\|_2$$

n'est plus satisfaite, ou au-delà duquel x n'est plus le plus proche voisin de p_{i+1}. Finalement p_{i+1} devient p_i, et le processus est itéré jusqu'à ce que le plus proche voisin de x, satisfaisant la condition d'échantillonnage, soit p_0.

2.7.4 Résultats

Pour effectuer notre étude expérimentale, nous avons construit une base de données de 100 courbes régulières fermées. Pour chacune de ces courbes, l'axe médian est approximé avec la méthode de la section 2.7.1. Ensuite, pour chacune de ces courbes, nous construisons un ensemble d'ϵ-échantillons avec la méthode de la section 2.7.3. Pour cela, l'intervalle $]0.3, 1]$ est divisé en sous-intervalles $]\epsilon_i, \epsilon_j]$, tels que $\epsilon_j - \epsilon_i = 0.1$. Pour chaque sous-intervalle, la valeur de ϵ imposée à la méthode d'échantillonnage est ϵ_j. Les bornes 0.3 et 1 s'expliquent par le fait que tous les graphes considérés garantissent une reconstruction correcte, à partir d'un ϵ-échantillon, si $\epsilon < 0.3$. D'autre part, si $\epsilon > 1$, il a été démontré que la répartition des points introduit une ambiguïté quand à la solution [Amenta *et al.*, 1998]. Ces graphes sont ceux analysés à la section 2.3, c.-à-d. le Crust local [Gold, 1999], le Crust des plus proches voisins (NNC) [Dey et Kumar, 1999], le Crust conservatif (CC) [Dey *et al.*, 2000], et Gathan [Dey et Wenger, 2001].

Pour trouver expérimentalement une borne ϵ pour chaque algorithme de reconstruction A, et pour chaque courbe Γ de la base, nous adoptons le protocole suivant :

a. Pour chaque valeur ϵ_j, allant de 0.4 à 1 (avec un pas de 0.1), l'algorithme A teste si l'ϵ_j-échantillon P permet de reconstruire correctement Γ. Si c'est le cas, on peut passer à l'étape suivante. Si ce n'est pas le cas, la borne pour A est inférieure à ϵ_j.
b. Cette étape permet de vérifier si la reconstruction obtenue est robuste à l'ajout de points de $\Gamma \setminus P$. Pour cela, la courbe est parcourue avec un pas de discrétisation donné. Pour chaque point parcouru $p \in \Gamma \setminus P$, nous vérifions que la reconstruction de la courbe avec $P \cup \{p\}$ est toujours correcte. Si ce n'est pas le cas, la borne pour A est inférieure à ϵ_j. À l'issue du parcours, la reconstruction est supposée stable pour ϵ_j, la valeur de ϵ_j est incrémentée de 0.1 (changement de sous-intervalle), et l'étape a. est de nouveau effectuée.

La figure 2.36 illustre un extrait de la base de courbes que nous avons testées. Le résultat de l'expérimentation, sur cet extrait de la base, est reporté à la Table 2.36. On

peut alors faire trois observations principales. Premièrement, tous les algorithmes reconstruisent correctement les courbes à partir d'ϵ-échantillons bien plus éparses que ceux prévus par la théorie (voir la section 2.3). Ceci ne veut pas nécessairement dire que les bornes théoriques ne sont pas optimales. En effet, le protocole expérimental ne garantit pas qu'avec la borne trouvée, il n'existe pas de configuration dans laquelle la reconstruction échoue. D'autre part, les différences entre les bornes trouvées expérimentalement, pour l'ensemble des algorithmes, sont bien plus proches les unes des autres, que l'annonce la théorie. Une valeur de $\epsilon = 0.6$ semble permettre la reconstruction quelque soit l'algorithme. Dernièrement, lorsqu'il y a des différences, les β-Crusts locaux, ainsi que l'algorithme TopoCrust, fournissent des résultats au moins aussi satisfaisant que les autres approches testées. Entre autres, on peut remarquer que TopoCrust, qui donne toujours la borne la plus élevée, à un comportement similaire à Gathan. Néanmoins, dans certains cas, nous avons trouvé que Gathan est instable (voir étape b.), alors que TopoCrust ne l'est pas. Ceci est illustré à la figure 2.37 sur la courbe (j) de la figure 2.36.

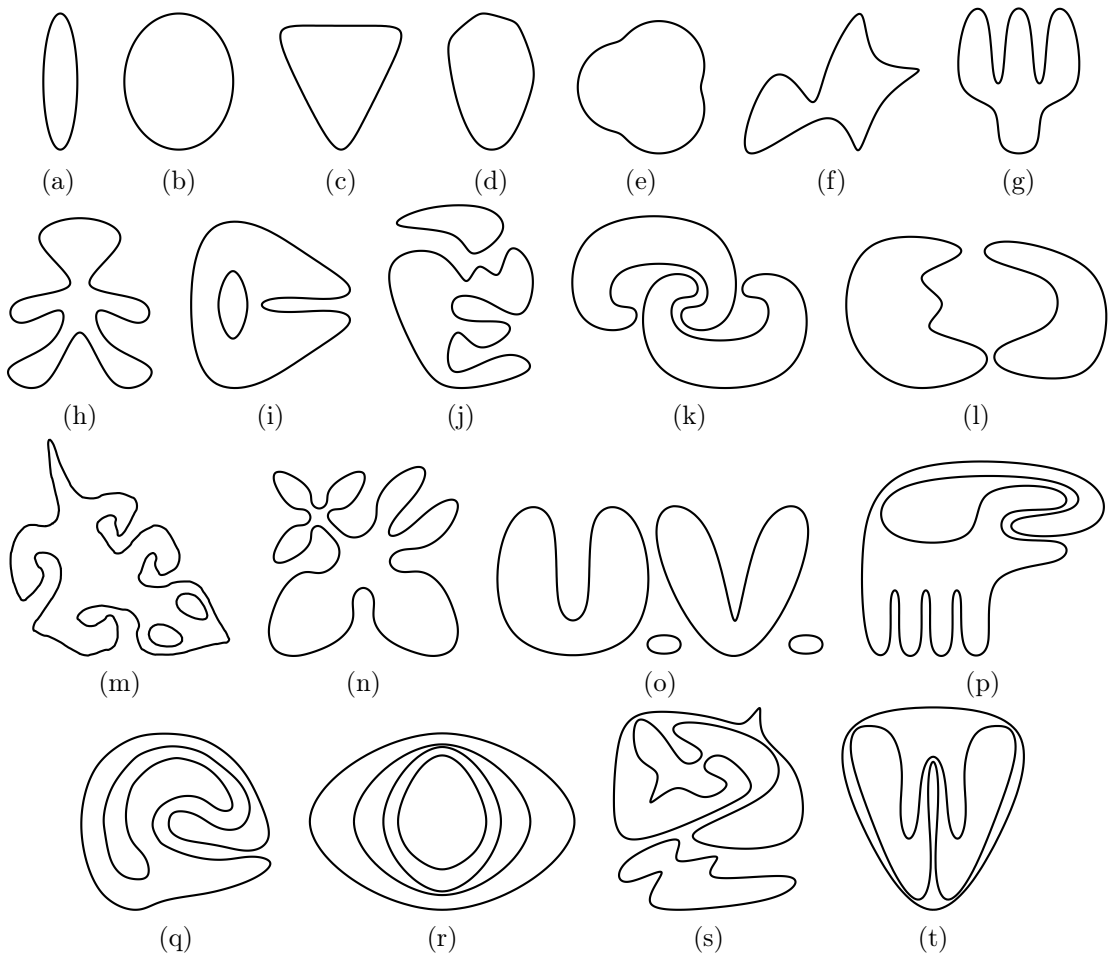
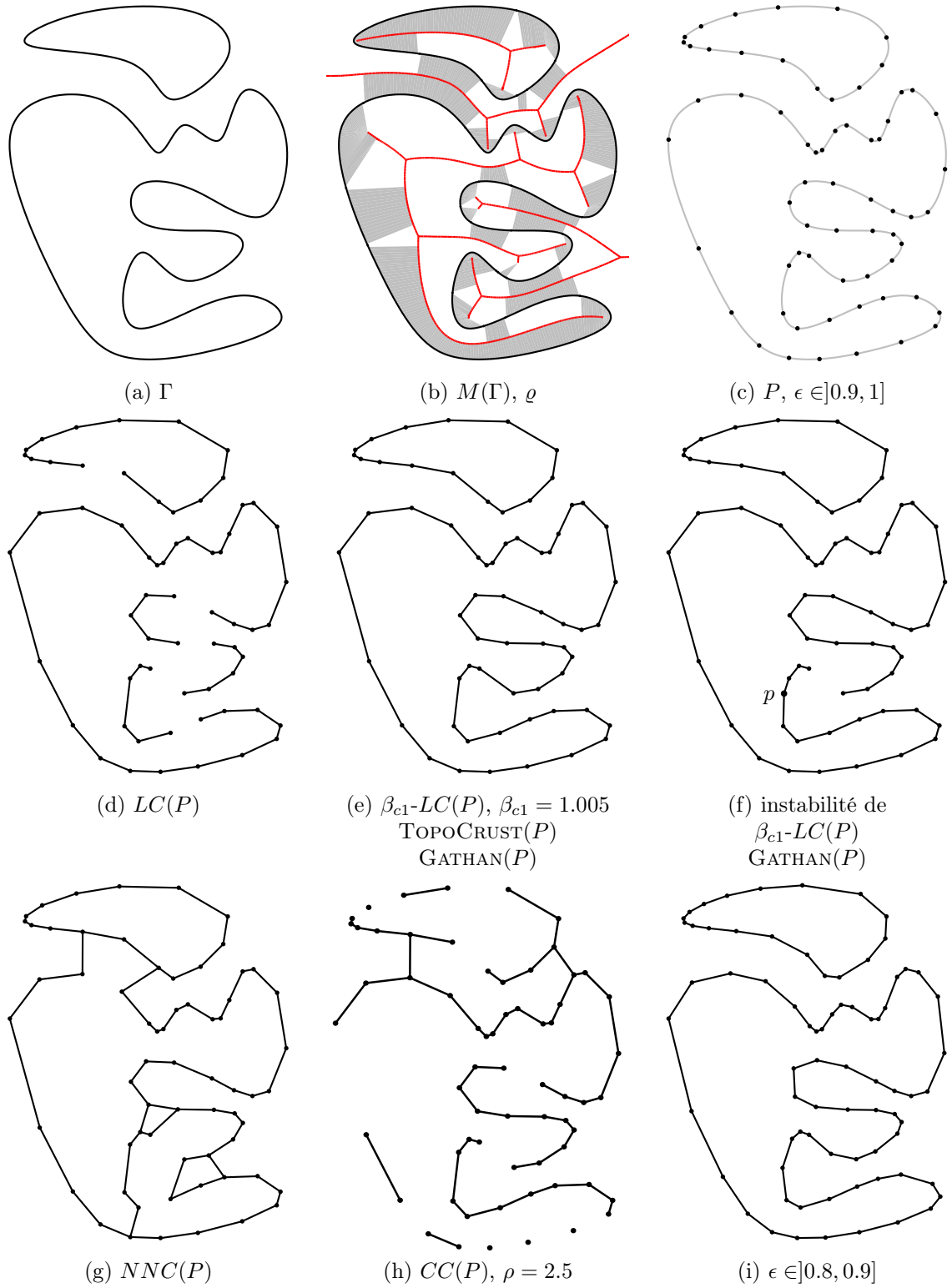


Fig. 2.36: Extrait de la base de données de courbes régulières fermées servant à l'expérimentation.

Algorithme	*LC*	β_{c1}-*LC*	TopoCrust	*NNC*	*CC*	Gathan
(a)	$]0.9, 1]$	$]0.9, 1]$	$]0.9, 1]$	$]0.9, 1]$	$]0.9, 1]$	$]0.9, 1]$
(b)	$]0.9, 1]$	$]0.9, 1]$	$]0.9, 1]$	$]0.9, 1]$	$]0.9, 1]$	$]0.9, 1]$
(c)	$]0.9, 1]$	$]0.9, 1]$	$]0.9, 1]$	$]0.9, 1]$	$]0.9, 1]$	$]0.9, 1]$
(d)	$]0.9, 1]$	$]0.9, 1]$	$]0.9, 1]$	$]0.9, 1]$	$]0.9, 1]$	$]0.9, 1]$
(e)	$]0.9, 1]$	$]0.9, 1]$	$]0.9, 1]$	$]0.9, 1]$	$]0.9, 1]$	$]0.9, 1]$
(f)	$]0.9, 1]$	$]0.9, 1]$	$]0.9, 1]$	$]0.8, 0.9]$	$]0.9, 1]$	$]0.9, 1]$
(g)	$]0.7, 0.8]$	$]0.7, 0.8]$	$]0.9, 1]$	$]0.7, 0.8]$	$]0.8, 0.9]$	$]0.9, 1]$
(h)	$]0.7, 0.8]$	$]0.7, 0.8]$	$]0.8, 0.9]$	$]0.8, 0.9]$	$]0.7, 0.8]$	$]0.8, 0.9]$
(i)	$]0.7, 0.8]$	$]0.7, 0.8]$	$]0.9, 1]$	$]0.7, 0.8]$	$]0.7, 0.8]$	$]0.9, 1]$
(j)	$]0.8, 0.9]$	$]0.8, 0.9]$	$]0.9, 1]$	$]0.7, 0.8]$	$]0.8, 0.9]$	$]0.8, 0.9]$
(k)	$]0.7, 0.8]$	$]0.7, 0.8]$	$]0.9, 1]$	$]0.7, 0.8]$	$]0.7, 0.8]$	$]0.7, 0.8]$
(l)	$]0.7, 0.8]$	$]0.8, 0.9]$	$]0.9, 1]$	$]0.7, 0.8]$	$]0.8, 0.9]$	$]0.8, 0.9]$
(m)	$]0.7, 0.8]$	$]0.8, 0.9]$	$]0.9, 1]$	$]0.7, 0.8]$	$]0.7, 0.8]$	$]0.8, 0.9]$
(n)	$]0.7, 0.8]$	$]0.7, 0.8]$	$]0.8, 0.9]$	$]0.6, 0.7]$	$]0.7, 0.8]$	$]0.6, 0.7]$
(o)	$]0.8, 0.9]$	$]0.8, 0.9]$	$]0.9, 1]$	$]0.8, 0.9]$	$]0.8, 0.9]$	$]0.8, 0.9]$
(p)	$]0.8, 0.9]$	$]0.8, 0.9]$	$]0.9, 1]$	$]0.8, 0.9]$	$]0.8, 0.9]$	$]0.8, 0.9]$
(q)	$]0.7, 0.8]$	$]0.8, 0.9]$	$]0.9, 1]$	$]0.8, 0.9]$	$]0.8, 0.9]$	$]0.8, 0.9]$
(r)	$]0.7, 0.8]$	$]0.8, 0.9]$	$]0.9, 1]$	$]0.8, 0.9]$	$]0.8, 0.9]$	$]0.8, 0.9]$
(s)	$]0.7, 0.8]$	$]0.8, 0.9]$	$]0.9, 1]$	$]0.7, 0.8]$	$]0.7, 0.8]$	$]0.8, 0.9]$
(t)	$]0.8, 0.9]$	$]0.8, 0.9]$	$]0.9, 1]$	$]0.8, 0.9]$	$]0.8, 0.9]$	$]0.8, 0.9]$

Tab. 2.1: Résultats de l'expérimentation sur des ϵ-échantillons des courbes régulières fermées de la figure 2.36.



(a) Γ (b) $M(\Gamma)$, ϱ (c) P, $\epsilon \in]0.9, 1]$

(d) $LC(P)$ (e) β_{c1}-$LC(P)$, $\beta_{c1} = 1.005$ TopoCrust(P) Gathan(P) (f) instabilité de β_{c1}-$LC(P)$ Gathan(P)

(g) $NNC(P)$ (h) $CC(P)$, $\rho = 2.5$ (i) $\epsilon \in]0.8, 0.9]$

Fig. 2.37: Expérimentation sur l'ϵ-échantillonage et la reconstruction. (a) un ϵ-échantillon qui permet de reconstruire correctement avec un β-Crust local, TopoCrust et Gathan (e). Néanmoins, le β-Crust local et Gathan sont instables (f). L'ajout d'un point coupe une arête de la reconstruction correcte. Les autres graphes (d,g,h) ne permettent pas la reconstruction correcte. Ils le permettent avec un ϵ-échantillon plus dense (i).

2.8 Conclusion et perspectives

Dans ce chapitre, nous avons proposé la famille hiérarchique des β-CRUSTS locaux pour reconstruire des courbes à partir d'un ensemble de points du plan. Les graphes de cette famille, inclus dans le graphe de Gabriel, sont générés à partir de relations géométriques entre les arêtes du diagramme de Voronoï et de la triangulation de Delaunay. En nous basant sur les propriétés hiérarchiques de ces graphes, et sur les propriétés topologiques des courbes à reconstruire, nous avons également proposé un algorithme (TOPOCRUST) pour extraire des courbes simples. À travers plusieurs exemples, nous avons montré que notre approche étend les capacités de reconstruction de plusieurs méthodes classiques, également basées sur le diagramme de Voronoï et la triangulation de Delaunay. Le principal avantage est de pouvoir reconstruire à partir d'échantillons de plus faible densité. Simultanément à la reconstruction, l'axe médian des courbes reconstruites est également extrait par l'algorithme, comme un sous-ensemble des arêtes du diagramme de Voronoï. À défaut de proposer une garantie de reconstruction avec l'algorithme TOPOCRUST, nous avons comparé son efficacité, avec des méthodes de reconstructions similaires, dans le contexte de l'ϵ-échantillonnage.

Une des perspectives des travaux présentés dans ce chapitre serait d'utiliser le principe de l'algorithme TOPOCRUST avec les propriétés hiérarchiques d'autres graphes que les β-CRUSTS locaux. Une autre perspective serait d'étendre les différents algorithmes ainsi obtenus, pour des ensembles de points définis en trois dimensions.

Chapitre 3

Extraction de Primitives Géométriques

« . . . local, global and contextual factors have a part to play in human visual perception of dot patterns. » [Urquhart, 1982]

Dans cette même première partie, nous proposons une méthode de détection de formes géométriques connues, à partir d'un ensemble de points 3D, même lorsque les formes à détecter sont plongées dans un milieu perturbé. Nous nous restreignons aux cas des structures linéaires et planaires. La détection est basée sur une famille de diagrammes affines, générés à partir de boules ellipsoïdales, dont la forme et l'orientation sont prédéfinies. Les triangulations, duales de ces diagrammes, ont la propriété de connecter des points qui ne sont pas naturellement voisins dans le diagramme de Voronoï. À partir de ces triangulations, nous définissons une extension du concept des α-formes. Contrairement aux α-formes classiques, la famille que nous proposons est capable de détecter des structures fines dans une direction donnée. La détection de structures polygonales plus complexes est réalisée en fusionnant le résultat obtenu pour différentes orientations.

3.1 Introduction

Dans ce chapitre, nous étudions le problème de la détection de structures géométriques connues, à partir d'un ensemble de points de $\mathbb{R}^3$. Nous considérons ici le cas des droites, des plans, et plus généralement des structures ayant une forme polygonale. Les méthodes existantes, basées sur les graphes de voisinage et le diagramme de Voronoï sont spécialement conçues pour des ensembles de points qui satisfont des conditions d'échantillonnage spécifiques. Lorsque les formes à détecter sont des courbes ou des surfaces, ces conditions peuvent être établies explicitement pour que la reconstruction soit correcte, voir par exemple [Bernardini et Bajaj, 1997, Boissonnat et Oudot, 2005, Dey, 2007] et les références qui y sont associées. Dans la plupart des situations, ces méthodes résolvent avec efficacité le problème de la reconstruction. La détection de structures géométriques connues, à partir des formes reconstruites, peut alors être effectuée de différentes façons. La détection intervient généralement dans des processus de simplification, de segmentation, ou de reconnaissance de formes [Rossignac, 2004, O'Rourke et Toussaint, 2004]. Néanmoins, lorsque l'ensemble de points est de faible densité, ou lorsque les formes à détecter sont plongées dans un milieu perturbé, les méthodes de reconstruction sont généralement insuffisantes, ce qui rend difficile l'extraction ou la reconnaissance des structures, même lorsque celles-ci sont simples. Ceci s'explique par le fait que les méthodes de reconstruction sont conceptuellement basées sur une analyse de la densité, sans intégrer l'aspect morphologique des structures à détecter.

Pour traiter ce problème, nous proposons une analyse anisotrope de l'ensemble de points, qui prend en compte la forme géométrique des structures. L'analyse anisotrope d'ensemble de points, ou plus généralement de données discrètes, a été utilisée avec succès pour détecter des structures linéaires bi-dimensionnelles [Melkemi et Djebali, 2001], pour reconstruire des surfaces par interpolations locales [Teichmann et Capps, 1998], ou encore pour générer des maillages [Labelle et Shewchuk, 2003]. Dans le contexte de la détection, nous définissons une famille d'α-formes anisotropes, qui étend le concept des α-formes [Edelsbrunner *et al.*, 1983, Mücke, 1993, Edelsbrunner et Mücke, 1994]. Ce dernier concept est une généralisation de l'enveloppe convexe d'un ensemble de points, qui permet de décrire la forme d'un ensemble de points à plusieurs niveaux de détail. Les α-formes sont générées à partir de boules de rayon α, et peuvent être efficacement calculées à partir de la triangulation de Delaunay et du diagramme de Voronoï de l'ensemble de points.

Notre extension des α-formes remplace les boules de rayon α par des boules ellipsoïdales, paramétrées par leur élongation et leur orientation. Ces paramètres contrôlent l'aspect morphologique des structures à détecter. La taille globale des boules ellipsoïdales est contrôlée par le paramètre α. Lorsque l'orientation et l'élongation sont fixées, et que la valeur de α varie de $+\infty$ à 0, les α-formes anisotropes respectives détectent premièrement des structures globales. Puis, des structures plus fines et localisées apparaissent dans la direction donnée par les boules ellipsoïdales. Si les boules sont suffisamment fines ou aplaties, les structures détectées approximent des portions de droites ou de plans. Les structures significatives sont alors celles qui sont formées d'un nombre suffi-

sant d'éléments (arêtes ou triangles). Plus généralement, lorsque la densité des structures à détecter est connue, des structures polygonales plus complexes peuvent être extraites de l'ensemble de points. Pour cela, plusieurs α-formes anisotropes sont calculées pour différentes orientations pré-définies. Puis, pour chaque forme calculée, les structures sont extraites en supprimant celles qui sont non-significatives. Enfin, les structures ainsi obtenues sont fusionnées.

Comme les α-formes, les α-formes anisotropes sont liées à des triangulations de l'ensemble de points, et à leurs diagrammes duaux. Nous appelons ces derniers les triangulations et les diagrammes anisotropes. Ils sont également définis à partir de boules ellipsoïdales, et ont la particularité de pouvoir connecter des points qui ne sont pas naturellement voisins dans la triangulation de Delaunay. De plus, l'ensemble des concepts anisotropes est lié à la triangulation de Delaunay par une transformation affine de l'ensemble de points. Ceci permet de les calculer directement à partir de la triangulation de Delaunay.

Dans la suite de ce chapitre, nous rappelons dans un premier temps le concept des α-formes. Puis nous présentons les concepts anisotropes et leurs propriétés, relatives à la détection de structures dans une direction donnée. Enfin, nous décrivons la méthode qui nous permet de détecter des structures ayant une forme polygonale plus générale.

3.2 Les α-formes

Soit $P := \{p_1, \ldots, p_N\} \subset \mathbb{R}^3$ un ensemble fini de points. Intuitivement, l'α-forme de P est constituée de l'ensemble des arêtes et des triangles, ayant pour sommets des points de P, et pour lesquels il existe une boule circonscrite dont l'intérieur ne contient aucun point de P [Edelsbrunner *et al.*, 1983, Mücke, 1993, Edelsbrunner et Mücke, 1994].

Soit $s(T_k)$ un k-simplexe tel que $T_k \subset P$, $k = 0, 1, 2$. Soit $b(\alpha, T_k)$ la boule ouverte de rayon $\alpha \in]0, +\infty[$ telle que $T_k \subset \partial b(\alpha, T_k)$. Pour k et α fixés, soit $F_{k,\alpha}(P)$ l'ensemble des k-simplexes défini par :

$$F_{k,\alpha}(P) := \{s(T_k),\, T_k \subset P \ : \ \exists b(\alpha, T_k),\, b(\alpha, T_k) \cap P = \emptyset\}.$$

L'α**-forme** de P, notée $F_\alpha(P)$ est alors constituée de l'union des ensemble $F_{k,\alpha}(P)$ pour $k = 0$, 1 et 2 (sommets, arêtes et triangles).

L'α-forme de P est un sous-complexe de la triangulation de Delaunay de P. Pour $k = 1$ et $k = 2$, un k-simplexe $s(T_k)$ de $Del(P)$ est un simplexe de $F_{k,\alpha}(P)$ si et seulement si $\alpha_{\min}(s(T_k)) \leq \alpha \leq \alpha_{\max}(s(T_k))$, où $\alpha_{\min}$ et $\alpha_{\max}$ sont deux réels positifs qui peuvent être calculés à partir du diagramme de Voronoï de P. La valeur de $\alpha_{\min}$ correspond à la distance minimum d'un sommet de $s(T_k)$ à la face duale de $s(T_k)$ dans $Vor(P)$. De façon similaire, la valeur de $\alpha_{\max}$ correspond à la distance maximum.

En faisant varier la valeur du paramètre α de 0 à $+\infty$, les α-formes décrivent la forme de l'ensemble de points P à différents niveaux de détails. Lorsque $\alpha \to +\infty$, l'α-forme correspond à l'enveloppe convexe de P. En faisant décroître la valeur de α, les α-formes extraient des structures de plus en plus détaillées et fines. Elles ont été utilisées efficacement dans de nombreuses applications. De plus, lorsque les points de P sont des échantillons de variétés, la reconstruction est généralement satisfaisante si l'échantillon est suffisamment dense et régulier [Bernardini et Bajaj, 1997]. L'avantage des α-formes est qu'elles permettent d'extraire des courbes avec l'ensemble $F_{1,\alpha}$ ou des surfaces avec l'ensemble $F_{2,\alpha}$. Néanmoins, pour le problème considéré dans ce chapitre, les α-formes sont insuffisantes. Ceci est illustré à la figure 3.1. L'ensemble de points P est un échantillon de droites (figure 3.1(a)), plongé dans un milieu perturbé (figure 3.1(b)). On peut observer que les α-formes ne peuvent pas extraire correctement les droites initiales. Ceci est dû à des points perturbateurs, qui se situent systématiquement dans les boules de rayon α circonscrites à deux échantillons successifs.

Dans la suite, nous montrons qu'avec une transformation affine de l'ensemble de points P, il est possible d'utiliser le concept des α-formes pour détecter des échantillons réguliers de droites et de plans.

Remarque. *Les autres méthodes de reconstruction, issue de la géométrie algorithmique, ne permettent pas non plus de détecter des structures géométriques données dans un ensemble de points $\mathbb{R}^3$ [Dey, 2007]. En effet, elles supposent que les points soient organisés*

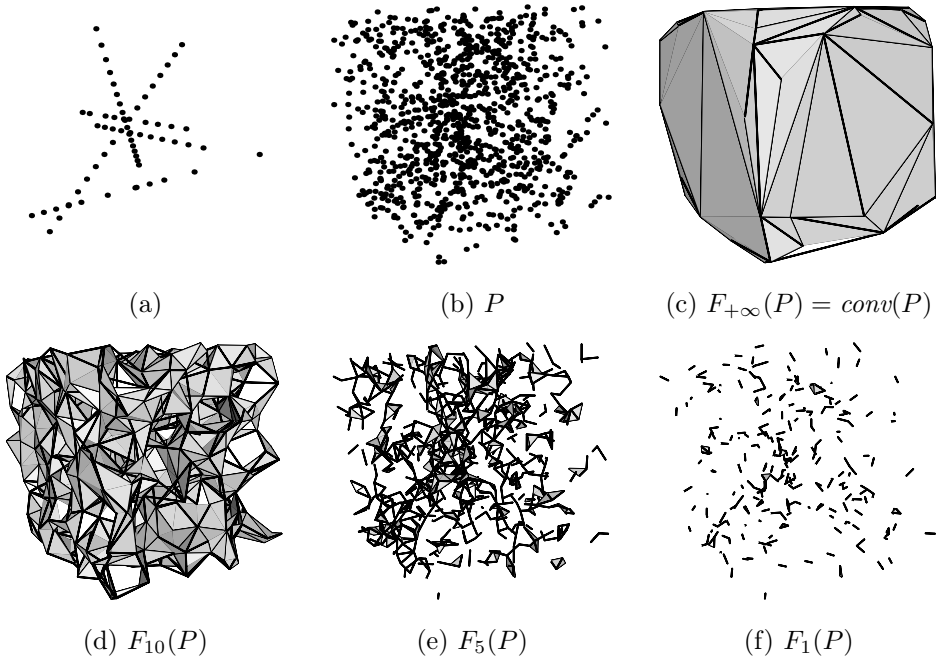


Fig. 3.1: Spectre des α-formes. (a) Un échantillon de droites. (b) L'échantillon est plongé dans un milieu perturbé (tirage aléatoire de 5000 points). (c,d,e,f) Différentes α-formes de P. Aucune d'entre-elles n'est capable de reconstruire les droites échantillonnées en (a).

près de la surface des objets à reconstruire. Le type de perturbation habituellement pris en compte est une indétermination sur la position exacte de la surface, généralement un bruit normal à la surface.

3.3 Concepts anisotropes

Dans cette section nous définissons les structures géométriques et combinatoires à partir desquelles les méthodes de détection sont construites. Ces structures sont les extensions du diagramme de Voronoï, de la triangulation de Delaunay et des α-formes, obtenues en remplaçant les boules par des boules ellipsoïdales.

3.3.1 Anisotropie globale

L'**anisotropie globale** est formalisée par la matrice 3×3 à coefficients réels :

$$Q = \begin{pmatrix} e_1 r_{11} & e_1 r_{12} & e_1 r_{13} \\ e_2 r_{21} & e_2 r_{22} & e_2 r_{23} \\ e_3 r_{31} & e_3 r_{32} & e_3 r_{33} \end{pmatrix}, \tag{3.1}$$

où $R = (r_{ij})_{1 \leq i,j \leq 3}$ est une matrice de rotation, et $E = \{e_1, e_2, e_3\}$ est un ensemble de paramètres d'échelle. Basée sur Q, la **distance anisotrope** entre deux points de $\mathbb{R}^3$,

notée $\|.\|_Q$, est définie par :

$$\|p-q\|_Q = \sqrt{(Q(p-q))^T Q(p-q)}. \tag{3.2}$$

Ceci revient à effectuer un changement d'échelle non-uniforme, le long des axes d'un système orthogonal en rotation, avant de mesurer la distance Euclidienne entre p et q.

Soit $\tau : \mathbb{R}^3 \to \mathbb{R}^3$ la transformation (affine) associée à la matrice Q : $\tau(p) = Qp$, pour tout $p \in \mathbb{R}^3$. On peut observer que la distance anisotrope est liée à la distance Euclidienne par l'équation suivante :

$$\|p-q\|_Q = \|\tau(p) - \tau(q)\|_2. \tag{3.3}$$

Les boules, associées à cette distance, sont des **boules ellipsoïdales** dont les axes sont donnés par R^{-1} (direction), et par E (élongation). Dans la suite, nous notons $b_Q(p, \alpha)$ la boule ellipsoïdale ouverte centrée en p, et de rayon $\alpha > 0$. Les semi-axes de $b_Q(p, \alpha)$ ont pour longueur $\alpha\lambda_i$, avec $\lambda_i = 1/e_i$, $i = 1, 2, 3$.

L'effet de la transformation τ est illustré à la figure 3.2, sur un échantillon de droites. L'axe majeur de la boule ellipsoïdale de référence ($\alpha = 1$) est orienté dans la direction d'une droite à détecter. Dans cette direction, les points transformés sont alors plus proches, comparativement à leur répartition dans l'ensemble initial. Réciproquement, cela revient à éloigner les points dans toutes les autres directions.

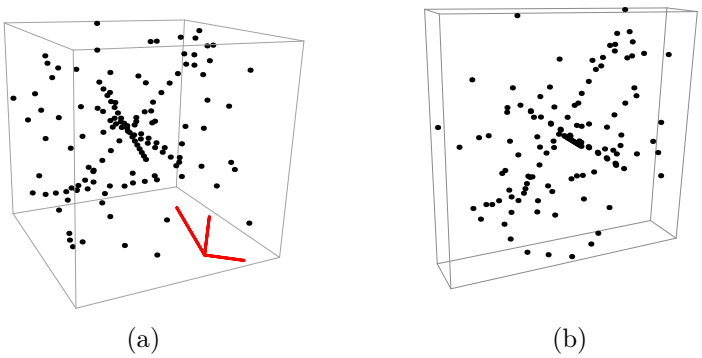


Fig. 3.2: (a) Un ensemble de points de $\mathbb{R}^3$ contenant des droites, et semi-axes de la boule ellipsoïdale de référence (l'axe majeur est dans la direction d'une droite). (b) L'ensemble de points transformés par τ.

3.3.2 Diagrammes et triangulations anisotropes

Soit $P \subset \mathbb{R}^3$ un ensemble de points. Par analogie avec le diagramme de Voronoï et la triangulation de Delaunay, nous définissons maintenant les diagrammes et les triangulations qui sont basés sur la distance anisotrope.

Soit p_i un point de P. Pour une anisotropie donnée Q, la **cellule anisotrope** de p_i,

notée $Vor_Q(p_i, P)$, est l'ensemble des points de $\mathbb{R}^3$ situés plus proche de p_i que des autres points de P, selon la distance anisotrope :

$$Vor_Q(p_i, P) = \{p \in \mathbb{R}^3 \ : \ \|p - p_i\|_Q \leq \|p - p_j\|_Q, \forall p_j \in P\}.$$

Notons que l'ensemble de points équidistants de deux points de $\mathbb{R}^3$, d'après la distance anisotrope, est un plan. Ceci montre que l'union des cellules anisotropes des points de P est un diagramme affine [Aurenhammer et Imai, 1988]. Nous appelons ce diagramme, le **diagramme anisotrope** de P, et nous le notons $Vor_Q(P)$:

$$Vor_Q(P) = \{\, Vor(p_i, P), p_i \in P\}.$$

D'un point de vue morphologique, le diagramme $V_Q(P)$ est généré à partir de boules ellipsoïdales qui ont toutes la même orientation et les mêmes paramètres d'élongation.

Le dual géométrique et combinatoire du diagramme anisotrope de P, noté $Del_Q(P)$, est une triangulation de P que nous appelons **triangulation anisotrope**. C'est l'ensemble des k-simplexes $s(T_k)$, $0 \leq k \leq 3$ et $T_k \subset P$, duaux de l'intersection des $k+1$ cellules anisotropes $V_Q(p_i, P)$, pour tout $p_i \in T_k$. C'est-à-dire l'ensemble des simplexes pour lesquels il existe une boule ellipsoïdale circonscrite qui ne contient aucun point de P :

$$Del_Q(P) = \{s(T_k), 0 \leq k \leq 3, T_k \subset P \ : \ \exists b_Q(T_k), b_Q(T_k) \cap P = \emptyset\},$$

où $b_Q(T_k)$ est une boule ellipsoïdale ouverte telle que ∂b_Q contient les points de T_k. Le diagramme anisotrope et sa triangulation duale possèdent toutes les propriétés des diagrammes affines et de leurs duaux, qui n'est pas nécessaire de détailler pour la compréhension de la méthode proposée dans ce chapitre.

En faisant varier l'orientation et l'élongation des boules ellipsoïdales, les diagrammes et les triangulations anisotropes décrivent deux familles duales qui étendent le diagramme de Voronoï et la triangulation de Delaunay. En effet, lorsque $E = \{1, 1, 1\}$, les boules

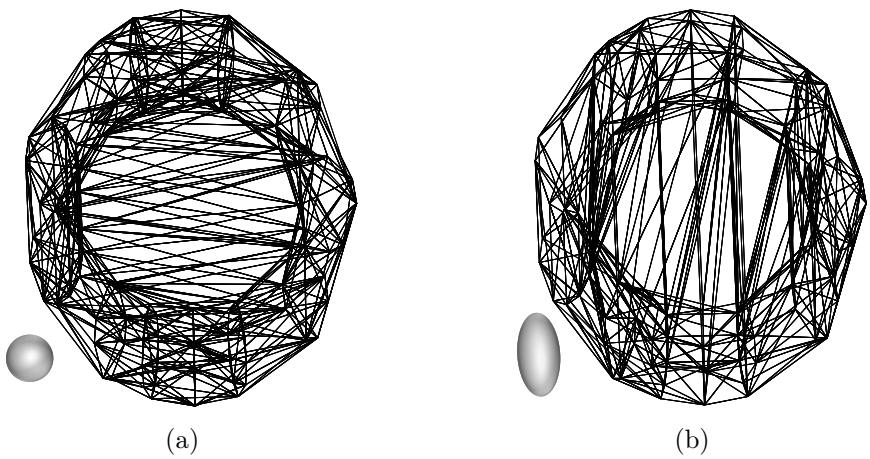


Fig. 3.3: (a) Triangulation de Delaunay. (b) Une triangulation anisotrope.

ellipsoïdales deviennent des boules classiques, et $Vor_Q(P) = Vor(P)$ (respectivement $Del_Q(P) = Del(P)$). Un exemple de triangulation anisotrope est présentée à la figure 3.3 avec son élément structurant de référence, et comparée à la triangulation de Delaunay. On peut y observer que $Del_Q(P)$ connecte des points de P qui ne sont pas connectés dans $Del(P)$. C'est l'un des avantages de ces concepts.

3.3.3 Les α-formes anisotropes

Étant donné une valeur $\alpha \in]0, +\infty[$, soit $F_{k,\alpha,Q}$ l'ensemble des k-simplexes $s(T_k)$, $T_k \subset P$, pour lesquels il existe une boule ellipsoïdale circonscrite, de rayon α, qui ne contient aucun point de P :

$$F_{k,\alpha,Q}(P) = \{s(T_k), T_k \subset P \ : \ \exists b_Q(\alpha, T_k), b_Q(\alpha, T_k) \cap P = \emptyset\}.$$

L'α**-forme anisotrope** de P, notée $F_{\alpha,Q}(P)$, est l'union des ensembles $F_{0,\alpha,Q}(P)$, $F_{1,\alpha,Q}(P)$ et $F_{2,\alpha,Q}(P)$.

En faisant varier l'anisotropie Q, les différentes α-formes anisotropes de P décrivent une famille de graphes qui étendent les α-formes Euclidiennes. Lorsque $E = \{1, 1, 1\}$, nous avons $F_{\alpha,Q}(P) = F_\alpha(P)$. De plus, quelque soit α et Q, les α-formes anisotropes sont des sous-graphes de la triangulation anisotrope.

Propriété 14 *Si $s(T_k)$ est un k-simplexe de $F_{\alpha,Q}(P)$, alors $s(T_k)$ est un k-simplexe de $Del_Q(P)$, pour tout Q, $\alpha \in]0, +\infty[$ et $0 \leq k \leq 2$.*

Preuve. Soit $b_Q(\alpha, T_k)$ une boule ellipsoïdale de rayon α telle que $b_Q(\alpha, T_k) \cap P = \emptyset$. Par définition, $s(T_k)$ est un simplexe de $F_{\alpha,Q}(P)$. Soit $c \in \mathbb{R}^3$ le centre de $b_Q(\alpha, T_k)$. Nous avons alors $\|p_i - c\|_Q \leq \|p_j - c\|_Q$ pour tout $p_i \in T_k$ et tout $p_j \in P$. D'après la définition des cellules anisotropes, ceci est équivalent à $c \in \cap_{p_i \in T_k} Vor_Q(p_i, P)$. Alors $\cap_{p_i \in T_k} Vor_Q(p_i, P) \neq \emptyset$, et par conséquent $s(T_k)$ est un simplexe de $Del_Q(P)$. $\square$

3.3.4 Relations entre les concepts anisotropes et Euclidiens

Les concepts anisotropes sont liés aux concepts Euclidiens par la transformation τ. Rappelons que τ est la transformation associée à la matrice Q. Soit $\tau(P)$ l'ensemble de points de P transformés par τ. De même, soit $s(\tau(T_k))$ le k-simplexe dont les sommets ont été transformés par τ. Nous avons alors les propriétés suivantes.

Propriété 15 *Soit Q une anisotropie donnée. (i) Pour tout $p_i \in P$, v est un point de $Vor(\tau(p_i), \tau(P))$ si et seulement si $\tau^{-1}(v)$ est un point de $Vor_Q(p_i, P)$. (ii) Pour tout $\alpha \in]0, +\infty[$, $s(T_k)$ est un simplexe de $F_{\alpha,Q}(P)$ si et seulement si $s(\tau(T_k))$ est un simplexe de $F_\alpha(\tau(P))$.*

Preuve. (i) soit v un point de $Vor(\tau(p_i), \tau(P))$. Par définition, on a $\|\tau(p_i) - v\|_2 \leq \|\tau(p_j) - v\|_2$ pour tout $p_j \in P$. Observons que $\tau^{-1}(\tau) = \tau$. Alors, à partir de l'équation (3.3), la dernière inéquation est équivalente à $\|p_i - \tau^{-1}(v)\|_Q \leq \|p_j - \tau^{-1}(v)\|_Q$,

pour tout $p_j \in P$. Ceci veut dire que $\tau^{-1}(v)$ est un point de $Vor_Q(p_i, P)$. (ii) soit $s(T_k)$ un k-simplexe de $F_{\alpha,Q}(P)$. Il existe donc une boule ellipsoïdale $b_Q(\alpha, T_k)$ telle que $b_Q \cap P = \emptyset$. Soit c le centre de $b_Q(\alpha, T_k)$. Par définition, on a $\|p_i - c\|_Q \leq \|p_j - c\|_Q$ pour tout $p_i \in T_k$ et tout $p_j \in P$. À partir de l'équation (3.3), ceci est équivalent à $\|\tau(p_i) - \tau(c)\|_2 \leq \|\tau(p_j) - \tau(c)\|_2$, avec $\|\tau(p_i) - \tau(c)\|_2 = \alpha$, pour tout $p_i \in T_k$ et tout $p_j \in P$. Il existe donc une boule ouverte $b(\alpha, \tau(T_k))$, centrée en $\tau(c)$, telle que $b(\alpha, \tau(T_k)) \cap \tau(P) = \emptyset$. Par conséquent, le simplexe $s(\tau(T_k))$ appartient à $F_\alpha(\tau(P))$. □

Étant donné que les α-formes Euclidiennes peuvent être calculées à partir de $Del(P)$, la propriété précédente montre que les α-formes anisotropes de P peuvent également être calculées à partir de $Del(P)$. Dans la suite, nous décrivons l'algorithme qui effectue ce calcul.

3.3.5 Calcul d'α-formes anisotropes

Les α-formes anisotropes sont utilisées pour détecter des structures géométriques élémentaires dans P (droites et plans). Ces structures sont matérialisées soit par des ensembles d'arêtes (cas des droites), soit par des ensembles de triangles (cas des plans). Pour cela, nous décrivons un algorithme qui calcule uniquement l'ensemble $F_{k,\alpha,Q}$, pour $k = 1$ ou 2 (Algorithme 5). Il est basé sur la Propriété 15, et il se décompose en trois étapes principales :

1. calculer l'ensemble P' des points de P transformés par τ,
2. calculer $Del(P')$,
3. extraire $F_{k,\alpha,Q}(P)$ de $Del(P')$.

Nous détaillons la troisième étape (lignes 4 à 17), qui est similaire au calcul d'α-formes Euclidiennes [Mücke, 1993]. Elle revient à calculer $F_\alpha(P')$, tout en construisant les k-simplexes avec les points de P (ligne 16).

Rappelons qu'un k-simplexe $s(T)$ appartient à $F_\alpha(P')$ si et seulement si $\alpha_{\min} \leq \alpha \leq \alpha_{\max}$, où $\alpha_{\min}$ et $\alpha_{\max}$ sont respectivement les distances minimale et maximale des sommets de s au dual de s dans $Vor(P')$ (ligne 14). Par dualité, ces bornes locales peuvent être obtenues directement à partir de $Del(P')$ [Mücke, 1993]. Soit b_T la boule ouverte, de rayon minimal r_T, telle que $T \subset \partial b_T$. Un k-simplexe $s(T)$ est dit **attaché** si $b_T \cap P' \neq \emptyset$ (ligne 7). Soit $\overline{s(T)}$ l'ensemble des $(k+1)$-simplexes incidents à $s(T)$:

$$\overline{s(T)} = \{s(T') \in Del(P') \ : \ s(T) \subset s(T'), |T'| = |T| + 1\}.$$

Alors, $\alpha_{\min}$ est égal à :

$$\mu_T = \min(\{r_{T'} \ : \ s(T') \in \overline{s(T)} \ \textit{non-attaché}\} \cup \{\mu_{T'} \ : \ s(T') \in \overline{s(T)} \ \textit{attaché}\}),$$

si $s(T)$ est attaché, et à r_T sinon. De même, $\alpha_{\max}$ est égal au rayon maximal des boules circonscrites aux 3-simplexes incidents à $s(T)$, si $s(T) \notin conv(P')$. Dans le cas contraire

Algorithm 5 Forme(k, α, Q, P)

Entrée: un ensemble de points $P \subset \mathbb{R}^3$, $k = 1$ ou 2, le rayon α, et l'anisotropie Q.
Sortie: $F_{k,\alpha,Q}(P)$.

1: initialiser $F_{k,\alpha,Q}$ à l'ensemble vide
2: $P' \leftarrow \tau(P)$ (à chaque point $p \in P'$ est associé $\tau^{-1}(p)$)
3: calculer $Del(P')$
4: **pour** $l = 3, \ldots, k$ **faire**
5: **pour chaque** l-simplexe $s(T) \in Del(P')$ **faire**
6: calculer et sauvegarder le rayon r_T de la sphère minimale circonscrite à s
7: **si** $\exists p \in T'$ tel que $s(T') \in \overline{s(T)}$ et $b_T \cap \{p\} \neq \emptyset$ **alors**
8: marquer s comme *attaché*
9: **pour chaque** k-simplexe $s(T) \in Del(P')$ **faire**
10: **si** $s(T) \in conv(P')$ **alors** $\alpha_{\max} \leftarrow +\infty$
11: **sinon** $\alpha_{\max} \leftarrow \max(\{r_{T'} \ : \ |T'| = 4, \ s(T) \subset s(T')\})$
12: **si** $s(T)$ est *attaché* **alors** $\alpha_{\min} \leftarrow \mu_T$
13: **sinon** $\alpha_{\min} \leftarrow r_T$
14: **si** $\alpha_{\min} \leq \alpha \leq \alpha_{\max}$ **alors**
15: $T' \leftarrow$ récupérer les points de P associés à T_k
16: insérer le k-simplexe $s(T')$ à $F_{k,\alpha,Q}$
17: **retourner** $F_{k,\alpha,Q}$

$\alpha_{\max} = +\infty$. Pour calculer $\alpha_{\min}$ et $\alpha_{\max}$ (lignes 10 à 12), le rayon r_T de chaque l-simplexe de $Del(P')$, pour $l = 3, \ldots, k$, est préalablement sauvegardé (ligne 6).

Le résultat de l'Algorithme 5, pour $k = 2$, est illustré à la figure 3.4 sur différents ensembles de points. On peut observer que les α-formes anisotropes peuvent reconstruire des formes à partir des points, que ces points soient inclus ou non dans un milieu perturbé (présence de bruit). Le choix des paramètres α et Q dépend du contexte applicatif. Dans la suite, nous analysons le comportement général des α-formes anisotropes, et montrons qu'elles sont capable de détecter des structures linéaires et planaires.

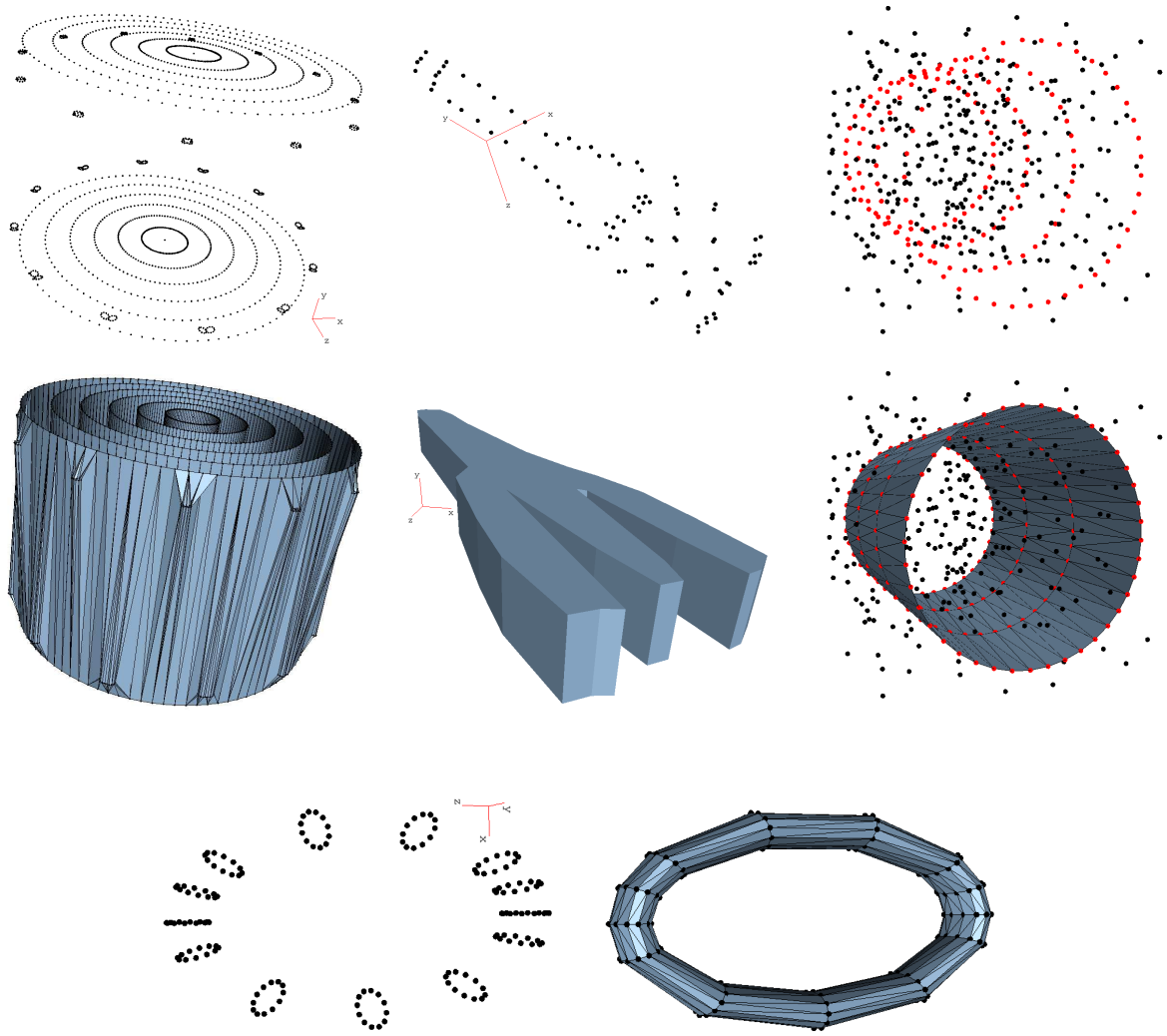


Fig. 3.4: Exemples d'α-formes anisotropes.

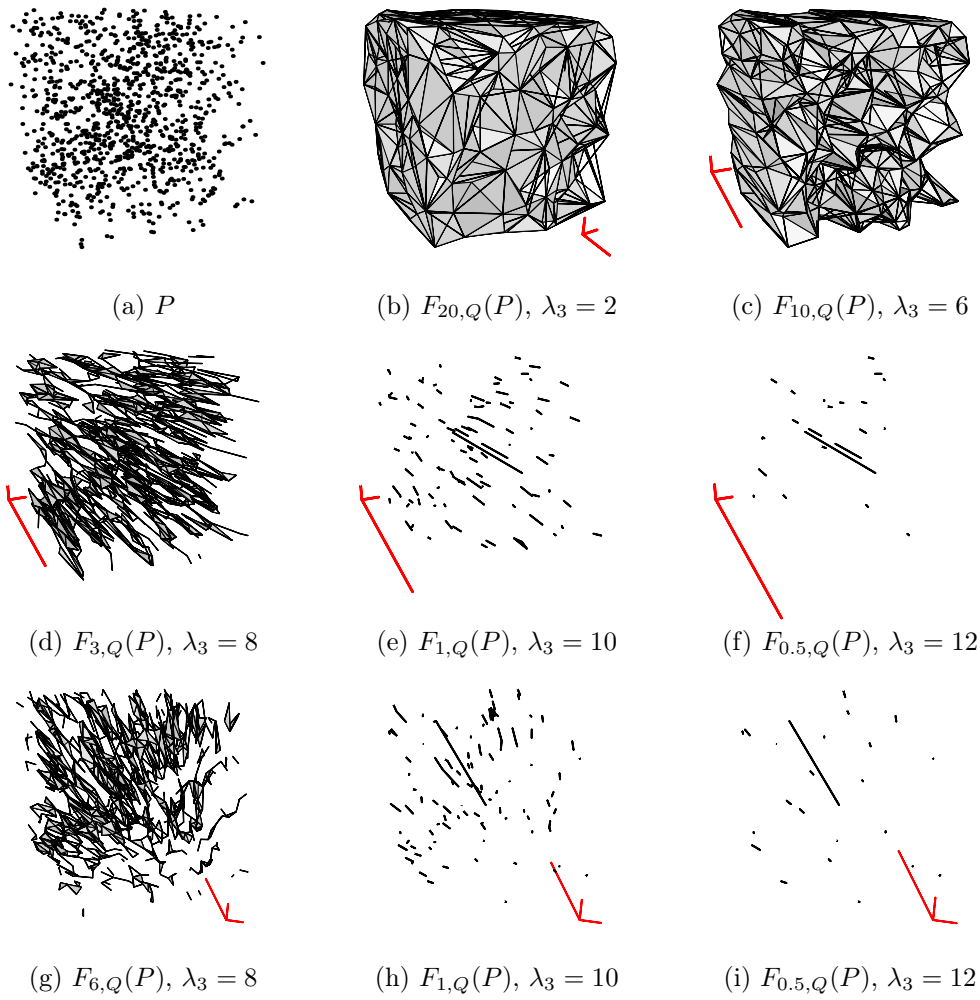
3.4 Extraction anisotrope de structures

Les α-formes anisotropes détectent des structures géométriques, comme des droites et des plans. Les paramètres α et Q (E et R) contrôlent la forme et l'orientation des boules ellipsoïdales. Si l'orientation R est fixée, et si α décroît de $+\infty$ et que les coefficients λ_i de l'élongation E croissent à partir de 1, la forme de la boule ellipsoïdale varie d'une boule (de rayon α) à des formes plus fines et allongées. Simultanément, le comportement de l'α-forme anisotrope est le suivant :

(i) En partant de l'enveloppe convexe de P, l'α-forme anisotrope (proche de l'α-forme Euclidienne) détecte premièrement des structures globales.

(ii) Puis, des structures plus fines et localisées apparaissent progressivement dans la direction R et selon l'élongation E.

Ces propriétés sont illustrées aux figures 3.5 et 3.6. L'exemple de la figure 3.5 montre des α-formes anisotropes d'un même ensemble de points, pour deux orientations différentes et plusieurs élongations. L'ensemble de points contient des échantillons de droites, qui sont détectées pour des valeurs appropriées données aux paramètres α et Q (figures 3.5(f) et 3.5(i)). On peut observer que des structures non-significatives sont aussi détectées. Nous verrons à la Section 3.4.3 que ces dernières peuvent facilement être supprimées. Le même comportement est illustré à la figure 3.6 sur un ensemble de points contenant des échantillons de plans. Il est comparé à celui des α-formes Euclidiennes (figures 3.6(f), 3.6(g) et 3.6(h)), qui ne peuvent pas détecter les structures souhaitées. Ceci est dû à l'interaction entre les échantillons d'une structure et leurs plus proches voisins qui ne sont pas des échantillons de cette structure (points de bruit ou échantillons d'autres structures).

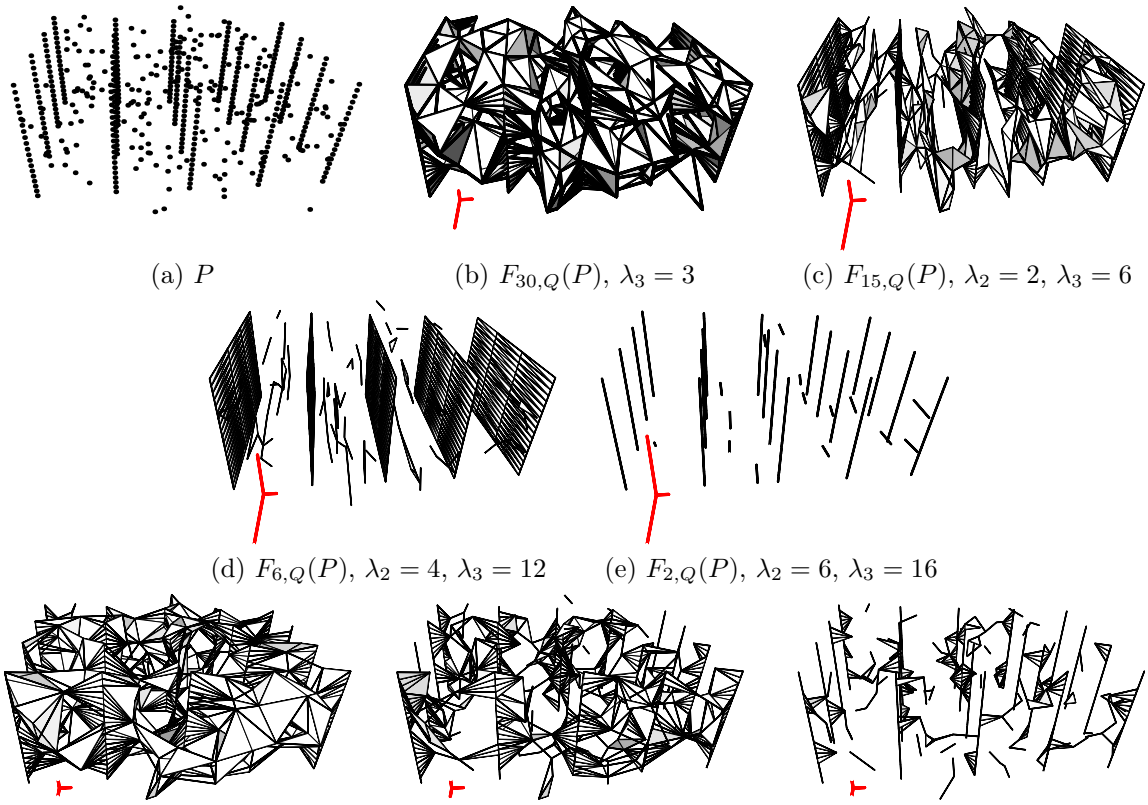


(a) P (b) $F_{20,Q}(P)$, $\lambda_3 = 2$ (c) $F_{10,Q}(P)$, $\lambda_3 = 6$

(d) $F_{3,Q}(P)$, $\lambda_3 = 8$ (e) $F_{1,Q}(P)$, $\lambda_3 = 10$ (f) $F_{0.5,Q}(P)$, $\lambda_3 = 12$

(g) $F_{6,Q}(P)$, $\lambda_3 = 8$ (h) $F_{1,Q}(P)$, $\lambda_3 = 10$ (i) $F_{0.5,Q}(P)$, $\lambda_3 = 12$

Fig. 3.5: α-formes anisotropes d'échantillons de droites plongés dans un milieu perturbé par du bruit. De (b) à (f), l'orientation est fixée, et un des trois paramètres λ_i augmente tandis que la valeur de α décroît. Les exemples (g), (h) et (i) montrent les mêmes résultats pour une autre orientation.



(a) P (b) $F_{30,Q}(P)$, $\lambda_3 = 3$ (c) $F_{15,Q}(P)$, $\lambda_2 = 2$, $\lambda_3 = 6$

(d) $F_{6,Q}(P)$, $\lambda_2 = 4$, $\lambda_3 = 12$ (e) $F_{2,Q}(P)$, $\lambda_2 = 6$, $\lambda_3 = 16$

(f) $F_{40}(P)$ (g) $F_{30}(P)$ (h) $F_{20}(P)$

Fig. 3.6: α-formes anisotropes d'échantillons de plans plongés dans un milieu perturbé par du bruit. De (b) à (e), deux des paramètres λ_i augmentent tandis que l'orientation est fixée, et la valeur de α décroît. De (f) à (h), les α-formes Euclidiennes ne peuvent pas détecter les structures, du fait de l'influence des points de bruit.

Dans la suite, nous analysons le lien entre les paramètres α et Q, et la détection de structures linéaires et planaires. Puis, nous décrivons une méthode pour supprimer les structures non-significatives qui peuvent être détectées en même temps que les structures géométriques correctes. Enfin, nous montrons comment extraire des structures plus complexes à partir des mêmes concepts.

3.4.1 Cas des droites

Soit $L = \{p_i, p_{i+1}, \ldots, p_j\} \subseteq P$ un ensemble de points ordonnés et régulièrement espacés sur une droite D. Notons d la distance entre deux points consécutifs de L. De plus, supposons qu'il n'existe aucun point de $P \setminus L$ sur la droite D.

Propriété 16 *Il existe $\alpha > 0$ et Q tels que $p_k p_{k+1} \in F_{1,\alpha,Q}(P)$, $\forall k = i, \ldots, j-1$.*

Preuve. Soit $R(u_1, u_2, u_3)$ une base orthonormale telle que le vecteur directeur de L soit colinéaire à u_1. Soit p_{i+l} et p_{i+l+1} deux échantillons successifs de L. Soit c le milieu du segment $[p_{i+l}, p_{i+l+1}]$. Considérons la boule ellipsoïdale $b_Q(\alpha)$, centrée en c, dont

la longueur des demi-axes est $\alpha\lambda_1$ dans la direction u_1, et α dans les autres directions ($\lambda_2 = \lambda_3 = 1$). Soit p le point de $P \setminus \{p_{i+l}, p_{i+l+1}\}$ ayant la plus petite distance anisotrope avec c. Alors $[p_{i+l}, p_{i+l+1}]$ est isolé de tout autre point de $P \setminus L$ si il correspond à une arête de $F_{1,\alpha,Q}(P)$ qui satisfait les propriétés suivantes : (i) $\alpha < \|c - p\|_Q$, et (ii) $\alpha\lambda_1 = d/2$. La condition (i) garantie que $q \notin b_Q(\alpha)$ et $q \notin \partial b_Q(\alpha)$, pour tout $q \in P \setminus L$. La condition (ii) garantie que $\{p_{i+l}, p_{i+l+1}\} \subset \partial b_Q(\alpha)$. Par conséquent, un choix adéquat pour les paramètres α et E, qui vérifient les conditions (i) et (ii), est $\lambda_1 = d/2\alpha$ et $\alpha = \min(d/2, \|c - p\|_Q - \epsilon)$. La constante $\epsilon \in]0, \|c - p\|_Q[$, permet de satisfaire (i). Ainsi, la boule $b_Q(\alpha)$ est déterminée telle que p_{i+l}, p_{i+l+1} est une arête de $F_{1,\alpha,Q}(P)$. $\square$

La propriété précédente montre que les α-formes anisotropes détectent nécessairement des droites dans P si les points sont régulièrement espacés sur ces droites. Un résultat similaire peut être formulé pour le cas des structures planaires.

3.4.2 Cas des plans

Soit $P' \subseteq P$ l'ensemble des échantillons d'un plan L. Soient p, p' et p'' trois points de P'. Comme illustré à la figure 3.7, supposons que le plan soit échantillonné régulièrement tel que $\|p - p'\|_2 = d$ dans la direction donnée par la droite l_1, $\|p - p''\|_2 = h$ dans la direction donnée par la droite $l2$, avec $d \leq h$. Considérons le triangle $s(T)$ de sommets $T = \{p, p', p''\}$. Alors, $s(T)$ possède une unique ellipse circonscrite, d'aire minimum, centrée au centre de gravité c de $s(T)$. C'est l'ellipse de Steiner de $s(T)$ [Kimberling, 1998]. De plus, supposons qu'il n'existe pas de points de $P \setminus T$ sur ou à l'intérieur de l'ellipse de Steiner de $s(T)$. Sous ces conditions, nous avons le résultat suivant.

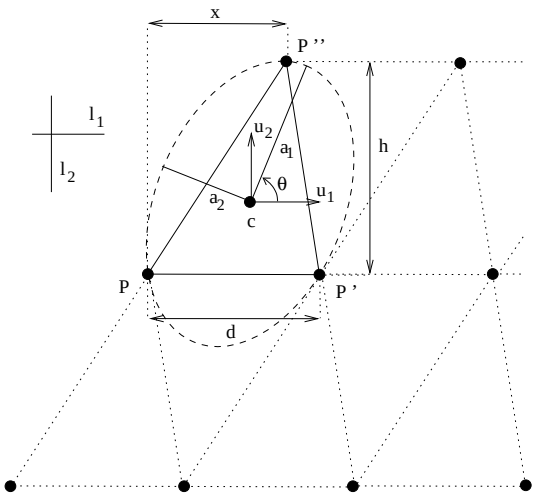


Fig. 3.7: Échantillons d'un plan et ellipse de Steiner circonscrite au triangle $pp'p''$. Son centre est le centre de gravité du triangle, et ses demi-axes sont a_2 et a_3.

Propriété 17 *Il existe $\alpha > 0$ et Q tels que $s(T) \in F_{2,\alpha,Q}(P)$.*

Preuve. Soit $R = (u_1, u_2, u_3)$ la base orthonormale telle que u_1 soit le vecteur pp', et u_2 le vecteur unitaire dans le plan passant par $s(T)$ avec $\angle u_1, u_2 = \pi/2$. Considérons

l'ellipse de Steiner de $s(T)$. Soient a_1 la longueur de son axe majeur et a_2 celle de son axe mineur. Dans la base $(u_1, u2)$ de dimension deux, a_1 fait un angle θ avec u_1 (voir la figure 3.7). Les paramètres a_1, a_2 et θ peuvent s'exprimer par :

$$\begin{aligned} a_1 &= \frac{1}{3}\sqrt{A+2\sqrt{B}} \\ a_2 &= \frac{1}{3}\sqrt{A-2\sqrt{B}} \\ \theta &= tan^{-1}\left(\frac{1}{2hd}\left((a^2-b^2)^2+d^2\left(2d^2-a^2-b^2+2\sqrt{B}\right)\right)\right) \end{aligned}$$

où $A = a^2+b^2+d^2$, $B = a^4+b^4+d^4-a^2b^2-b^2d^2-a^2d^2$, $a^2 = (d-x)^2+h^2$ et $b^2 = x^2+h^2$. Soit $b_Q(\alpha)$ la boule ellipsoïdale centrée en c et d'orientation $R_{u_3}(\theta)R$, où R_{u_3} est la rotation d'angle θ autour de u_3. Soit p le point de $P \setminus T$ qui est situé le plus proche de c, selon la distance anisotrope. Alors, $s(T)$ est isolé de tout point de $P \setminus T$ si c'est un 2-simplexe de $F_{\alpha,Q}(P)$ qui satisfait les propriétés suivantes : (i) $\alpha < \|c-p\|_Q$, et (ii) $\alpha\lambda_1 = a_1$ et $\alpha\lambda_2 = a_2$. La condition (i) garantie que $q \notin b_Q(\alpha)$ et $q \notin \partial b_Q(\alpha)$, pour tout $q \in P \setminus L$. La condition (ii) garantie que $T \subset \partial b_Q(\alpha)$. Un choix adéquat de α et Q, qui vérifie les conditions (i) e (ii), est $\lambda_1 = a_1/\alpha$, $\lambda_2 = a_2/\alpha$ et $\alpha = \min(\|c-p\|_Q - \epsilon, a_1)$. Par conséquent, la boule $b_Q(\alpha)$ est déterminée telle que $s(T)$ est un triangle de $F_{2,\alpha,Q}(P)$. □

Comme dans le cas des droites, la propriété précédente montre que les α-formes anisotropes détectent nécessairement des plans passant par des points de P régulièrement espacés.

3.4.3 Suppression des structures non-significatives

Pour un choix adéquat de α et de Q, l'α-forme anisotrope détecte des portions de droites ou de plans. Elle détecte également plusieurs structures non-significatives qui sont caractérisées par le faible le nombre de points connectés (voir les figures 3.5(f), 3.5(i), et 3.5(h)). Pour supprimer ces structures non-significatives, $F_{k,\alpha,Q}(P)$ est filtré en analysant ses composants connectés.

Formellement, un k-simplexe est **h-connecté** si il est incident à h $(k+1)$-simplexes. Pour une valeur k fixée (1 ou 2), un ensemble de k-simplexes $\{s_1, \ldots, s_l\}$ est un **chemin** de longueur l dans $F_{k,\alpha,Q}(P)$ si et seulement si les quatre propriétés suivantes sont satisfaites :

(i) s_z est un k-simplexe de $F_{k,\alpha,Q}(P)$ tel que $s_z \cap s_{z+1} \neq \emptyset$, pour tout $z = 1, \ldots, l-1$.

(ii) Pour tout $(k-1)$-simplexe $s \subset s_z$, s est 1-connecté ou 2-connecté.

(iii) Il n'existe pas de k-simplexe $s \in F_{k,\alpha,Q}(P)$ tel que $\{s_1, \ldots, s_l\} \cup \{s\}$ vérifie les propriétés (i) et (ii).

(iv) Le nombre d'éléments de $\{s_1, \ldots, s_l\}$ est l.

La propriété (i) garantit que le chemin représente une structure connectée. Le propriété (ii) assure que le chemin ne possède pas de ramification (plus de deux arêtes incidentes à un sommet, ou plus de deux triangles incidents à une arête). Les propriétés (iii) et (iv) garantissent que la longueur l est maximale.

Un chemin est **non-significatif** dans $F_{k,\alpha,Q}(P)$ si et seulement si sa longueur est inférieure à un seuil nb_s. Pour supprimer les chemins non-significatifs, les chemins de longueur $l = 1, \ldots, nb_s$ sont itérativement supprimés, ainsi que les sommets devenus 0-connectés. La méthode fonctionne en deux étapes principales, détaillées dans l'Algorithme 6, et illustrées à la figure 3.8.

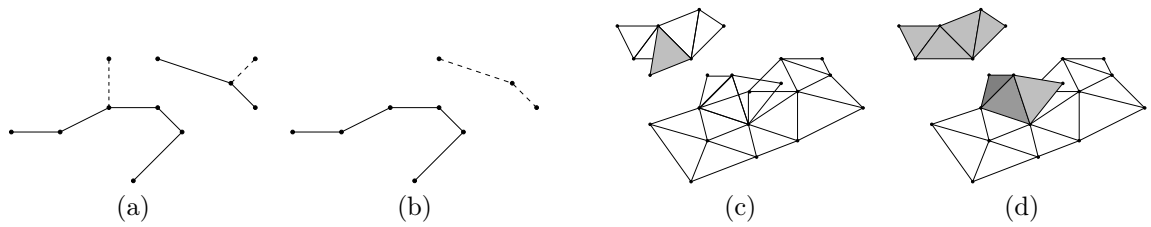


Fig. 3.8: $F_{1,\alpha,Q}(P)$ avec (a) $nb_c = 1$ et (b) $nb_c = 2$ où les arêtes en pointillés sont supprimés, et pour $F_{2,\alpha,Q}(P)$ avec (c) $nb_c = 1$ et (d) $nb_c = 3, 4$ où le triangles gris sont supprimés.

La première étape (lignes 1 à 9), consiste à décomposer $F_{k,\alpha,Q}(P)$ en chemins qui respectent les conditions (i), (ii) et (iii), et à supprimer les chemins de longueur 1 qui ne sont pas connectés à d'autres chemins de $F_{k,\alpha,Q}(P)$ (chemins isolés). Un chemin est calculé en recherchant un k-simplexe $s(T_k)$ non parcouru (non marqué), qui inclue un $(k-1)$-simplexe 1-connecté (ligne 7). Puis (voir Algorithme 7), les k-simplexes non marqués, adjacents à $s(T_k)$, sont ajoutés au chemin si ils partagent avec $s(T_k)$ un $(k-1)$-simplexe 2-connecté. Ce processus est itéré avec les k-simplexes ajoutés, jusqu'à ce qu'ils soient tous parcourus. Afin de modifier dynamiquement les chemins dans l'étape suivante, un chemin est représenté par un triplet (l, C, B), où l est sa longueur, C ses k-simplexes, et B l'ensemble de ses $(k-1)$-simplexes h-connectés, $h > 2$ (c.-à-d. la frontière commune avec d'autres chemins). L'ensemble des chemins est sauvegardé dans une liste L.

Dans la seconde étape, la liste L des chemins est premièrement triée dans l'ordre croissant de leurs longueurs. Puis, les chemins de longueurs $l = 1, \ldots, nb_s$ sont supprimés. Lors de la suppression d'un chemin, si sa frontière B est non vide, chaque $(k-1)$-simplexe de B est considéré. Si ce dernier connecte exclusivement deux autres chemins différents, ces derniers sont fusionnés en un nouveau chemin, et supprimés de la liste L. Le nouveau chemin est ensuite inséré à L, à la position donnée par sa longueur, nécessairement supérieure à l.

Comme illustré à la figure 3.8, le filtre obtenu permet de supprimer deux types de structures : les artefacts à la surface des structures de longueur importante, et les structures connexes de faible longueur. Un autre exemple de suppression de chemins non-significatifs

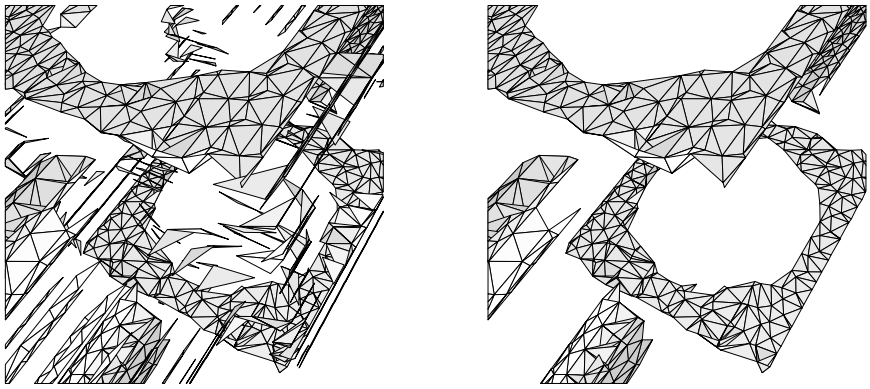


Fig. 3.9: Suppression des structures non-significatives.

Algorithm 6 SuppressionChemins(k, F, nb_s)

Entrée: un k-complexe simplicial F et le seuil nb_s.
Sortie: F filtré.

1: $label \leftarrow 1$
2: $L \leftarrow \emptyset$
3: **pour chaque** k-simplexe $s(T_k) \subset F$ **faire**
4: **si** $s(T_k)$ n'est pas marqué **alors**
5: **si** $s(T_k)$ n'est adjacent à aucun k-simplexe de F **alors**
6: supprimer $s(T_k)$ et T_k de F
7: **sinon si** $s(T_k)$ est adjacent à au plus k k-simplexes de F **alors**
8: ajouter Chemin($k, s(T_k), F, label$) à L
9: $label \leftarrow label+1$
10: trier L dans l'ordre croissant des longueurs de chemins
11: $(l, C, B) \leftarrow$ premier élément de L
12: **tant que** $((l, C, B) \neq \emptyset)$ et $(l \leq nb_s)$ **faire**
13: supprimer de F les k-simplexes de C, et leurs sommets devenus 0-connectés
14: **pour chaque** $(k-1)$-simplexe s de B **faire**
15: **si** s est 2-connecté dans F **alors**
16: $s_1, s_2 \leftarrow$ les deux k-simplexes incidents à s
17: **si** le *label* de s_1 est différent du *label* de s_2 **alors**
18: $(l_1, C_1, B_1) \leftarrow$ élément de L contenant s_1
19: $(l_2, C_2, B_2) \leftarrow$ élément de L contenant s_2
20: mettre le *label* des k-simplexes de C_1 au *label* de C_2
21: insérer $(l_1 + l_2, C_1 \cup C_2, \{B_1 \cup B_2\} \setminus \{s\})$ dans L
22: supprimer (l_1, C_1, B_1) et (l_2, C_2, B_2) de L
23: **sinon**
24: $(l_1, C_1, B_1) \leftarrow$ élément de L contenant s_1 (et s_2)
25: supprimer s de B_1
26: $(l, C, B) \leftarrow$ élément suivant dans L
27: **retourner** F

Algorithm 7 Chemin$(k, s(T_k), F, label)$

Entrée: un k-simplexe $s(T_k) \subset F$, et le *label* du chemin.

Sortie: le chemin C de longueur l qui inclue $s(T_k)$, et les $(k-1)$-simplexes h-connectés dans F, $h > 2$, inclus dans des k-simplexes de C.

```
l ← 0
B, C ← ∅
ajouter s(T_k) à Q
tant que Q ≠ ∅ faire
   extraire le premier k-simplexe s de Q
   si s n'est pas marqué avec label alors
      marquer s avec label et l'ajouter à C
      l ← l + 1
      pour chaque (k − 1)-simplexe s' ⊂ s faire
         si (s' est non marqué) et (s' est 2-connecté dans F) alors
            marquer s'
            ajouter à Q le k-simplexe adjacent à s' et incident à s
         sinon si s' est h-connecté dans F, h > 2 alors
            ajouter s' à B
retourner (l, C, B)
```

est illustré à la figure 3.9 sur un ensemble de 2-simplexes. Dans la suite, nous montrons comment combiner ce filtre et les α-formes anisotropes pour détecter des structures géométrique plus complexes que des droites et des plans.

3.4.4 Détection de structures polygonales

Pour k, α et E fixés, la famille des α-formes anisotropes, obtenues pour différentes orientations R, est capable de détecter des structures ayant des formes polygonales. Pour extraire ces structures, nous proposons une méthode générale qui fonctionne selon les trois étapes principales suivantes :

(i) calculer un ensemble fini d'orientations,

(ii) calculer et filtrer $F_{k,\alpha,Q}(P)$ pour chaque orientation,

(iii) fusionner les différents $F_{k,\alpha,Q}(P)$ et filtrer le résultat.

Étape (i). Nous représentons la matrice R par une composition de matrices de rotation, paramétrées par les angles d'Euler :

$$R = R(\theta, \psi, \phi) = R_z(\theta) R_x(\psi) R_y(\phi),$$

où $R_a(r)$ est la rotation d'angle r autour de l'axe a. Les coefficients de R sont alors donnés par :

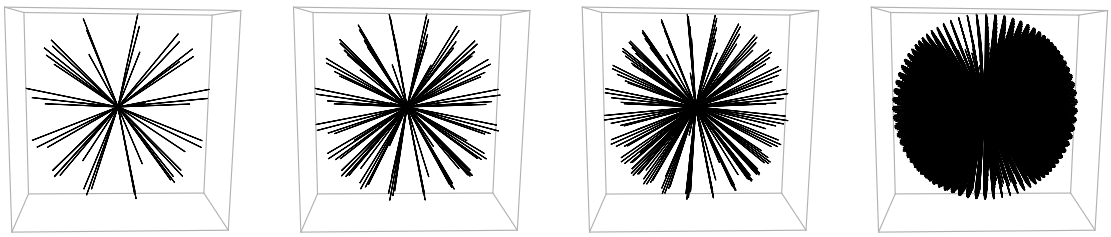
$$\begin{aligned}
r_{11} &= \cos\phi\cos\theta - \sin\phi\sin\theta\sin\psi \\
r_{12} &= -\sin\theta\cos\psi \\
r_{13} &= \cos\theta\sin\phi + \cos\phi\sin\theta\sin\psi \\
r_{21} &= \cos\phi\sin\theta + \cos\theta\sin\phi\sin\psi \\
r_{22} &= \cos\theta\cos\psi \\
r_{23} &= \sin\phi\sin\theta - \cos\phi\cos\theta\sin\psi \\
r_{31} &= -\sin\phi\cos\psi \\
r_{32} &= \sin\psi \\
r_{33} &= \cos\phi\cos\psi
\end{aligned}$$

où $\theta, \phi, \psi \in]-\pi/2, \pi/2]$.

Dans le but de couvrir uniformément l'espace $\mathbb{R}^3$, nous échantillons la surface d'un cube par une grille uniforme. Soit C le cube d'extrémités $(1,1,1)^T$ et $(-1,-1,0)^T$. Sa surface, privée du plan XY, noté $\partial C \setminus XY$, est échantillonnée uniformément le long des axes du repère Euclidien, avec une distance m_1 entre deux échantillons successifs. Ceci donne un ensemble fini d'orientations

$$R_{m_1} = \{R(0,\, \psi_i,\, \phi_i),\, i = 1, \ldots, l\},$$

où chaque orientation correspond à une droite OM, M étant un échantillon de $\partial C \setminus XY$, et O l'origine du repère Euclidien. Cette méthode, illustrée à la figure 3.10, est suffisante pour extraire des structures linéaires.



$m1 = 0.6$ $m1 = 0.4$ $m1 = 0.3$ $m1 = 0.1$

Fig. 3.10: Calcul d'ensembles d'orientations.

Dans le cas des structures planaires, l'ensemble d'orientations est calculé à partir de R_{m_1}. Ensuite, pour chaque orientation $R(0,\, \psi_i,\, \phi_i)$, l'angle $\theta \in]-\pi/2, \pi/2]$ est divisé en m_2 angles $\theta_j = \theta_{j-1} + \pi/m_2$, avec $j = 1, \ldots, m_2$ et $\theta_0 = -\pi/2$. Ceci fournit un ensemble :

$$R_{m_1,m_2} = \{R(\theta_j, \psi_i, \phi_i),\, i = 1, \ldots, l,\, j = 1, \ldots, m2\}.$$

Cette étape peut être calculée, indépendamment du reste de la méthode d'extraction. Dans nos expérimentations, nous utilisons des ensembles d'orientations pré-calculées, avec $m_1 = 0.2$ et $m_2 = \pi/12$.

Étapes (ii) et (iii) : extraction. Soit A un ensemble d'orientations. Le paramètre d'élongation est donné par $E = \{1, 1, 1/\lambda_3\}$ si $k = 1$, ou par $E = \{1, 1/\lambda_2, 1/\lambda_3\}$ si $k = 2$. Soit $F = \emptyset$ un ensemble de k-simplexes, $k = 1$ ou 2. Alors, pour chaque orientation $R(\theta, \psi, \phi)$ de A, la méthode consiste à :

1. calculer la matrice Q (voir Section 3.3.1),
2. calculer $F_{k,\alpha,Q}(P)$ (Algorithme 5),
3. filtrer $F_{k,\alpha,Q}(P)$ avec le seuil nb_c (Algorithme 6), et insérer ses k-simplexes dans F.

Ces étapes sont illustrées aux figures 3.11 et 3.12 dans le cas de $k = 2$, et à la figure 3.13 dans le cas de $k = 1$. L'ensemble de k-simplexe F qui en résulte peut comporter des structures non-significatives, issues des fusions de l'étape 3 (voir la figure 3.12(g)). Pour les supprimer, F est à son tour filtré avec l'Algorithme 6 (voir la figure 3.11(h)).

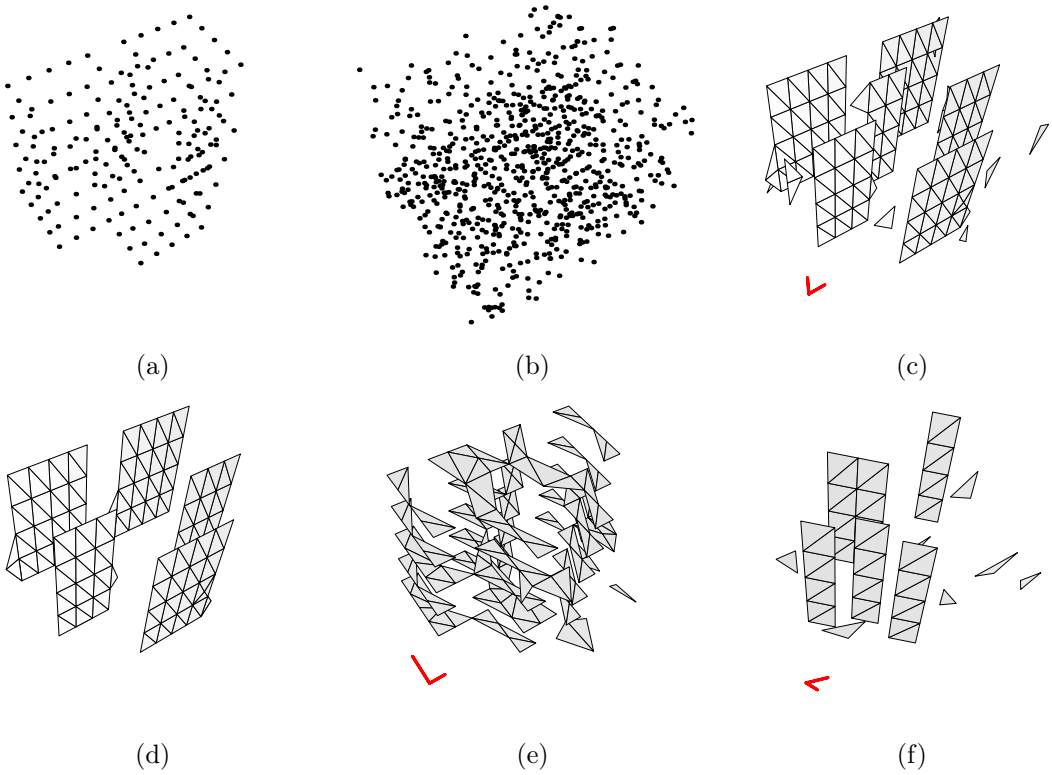
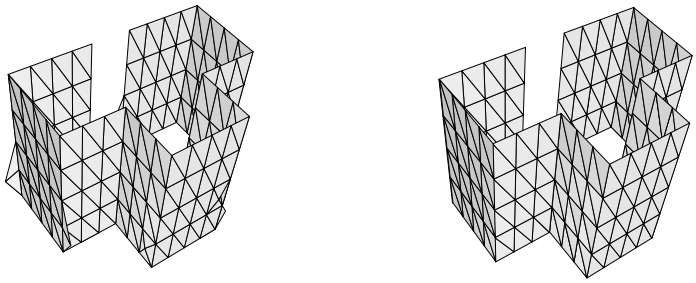


Fig. 3.11: Détection de structures polygonales avec $\alpha = 1$, $\lambda_2 = 4$, $\lambda_3 = 8$ et $nb_c = 20$. (a) Échantillons S d'une structure polygonale (190 points). (b) S est plongé dans un milieu perturbé aléatoirement (800 points), ce qui donne l'ensemble P. (c) Une α-forme anisotrope de P. (d) L'α-forme anisotrope, obtenue en (c), est filtrée. (e) et (f) montrent des exemples d'α-formes anisotropes composées uniquement de structures non-significatives.



(g) (h)

Fig. 3.12: Détection de structures polygonales avec $\alpha = 1$, $\lambda_2 = 4$, $\lambda_3 = 8$ et $nb_c = 20$ (suite de la figure 3.11). (g) Résultat de la fusion des α-formes anisotropes significatives. (h) Résultat de la détection, obtenu après filtrage de (g).

3.5 Conclusion et perspectives

Dans ce chapitre, nous avons proposé une méthode géométrique pour détecter des échantillons de droites et de plans dans un ensemble de points réels 3D, même si cet ensemble de points est plongé dans un milieu perturbé. Cette méthode est basée sur une famille de diagrammes affines et de leurs duaux, générés à partir de l'ensemble de points et de boules ellipsoïdales, dont l'élongation et l'orientation sont pré-définies. La détection est réalisée à travers une extension des α-formes. L'ensemble des concepts introduit se calcule directement à partir de la triangulation de Delaunay, en effectuant une transformation affine de l'ensemble de points. L'avantage d'une telle approche est de pouvoir détecter des structures plus ou moins fines dans des directions données, mais également de connecter des points qui ne sont pas naturellement voisins dans la triangulation de Delaunay. Lors de la détection, plusieurs structures non-significatives peuvent également être détectées. Nous proposons un mécanisme général pour les supprimer.

La principale perspective de ce travail est de concevoir des méthodes similaires pour détecter des structures géométriques plus complexes, comme par exemple des formes quadratiques. Dans ce sens, nous avons déjà proposé une méthode de détection de paraboles et de cercles à partir d'un ensemble de points 2D [Bougleux *et al.*, 2006]. L'idée est de transformer la détection de ces primitives géométriques en la détection de primitives plus simples. Par exemple, la détection de paraboles revient à détecter des droites, et la détection de cercles revient à détecter des amas de points. Pour cela, l'ensemble de points est transformé en un autre ensemble de points, en utilisant les propriétés différentielles des primitives à détecter, estimées à partir du diagramme de Voronoï. Ce concept prometteur nécessite encore de nombreuses investigations théoriques.

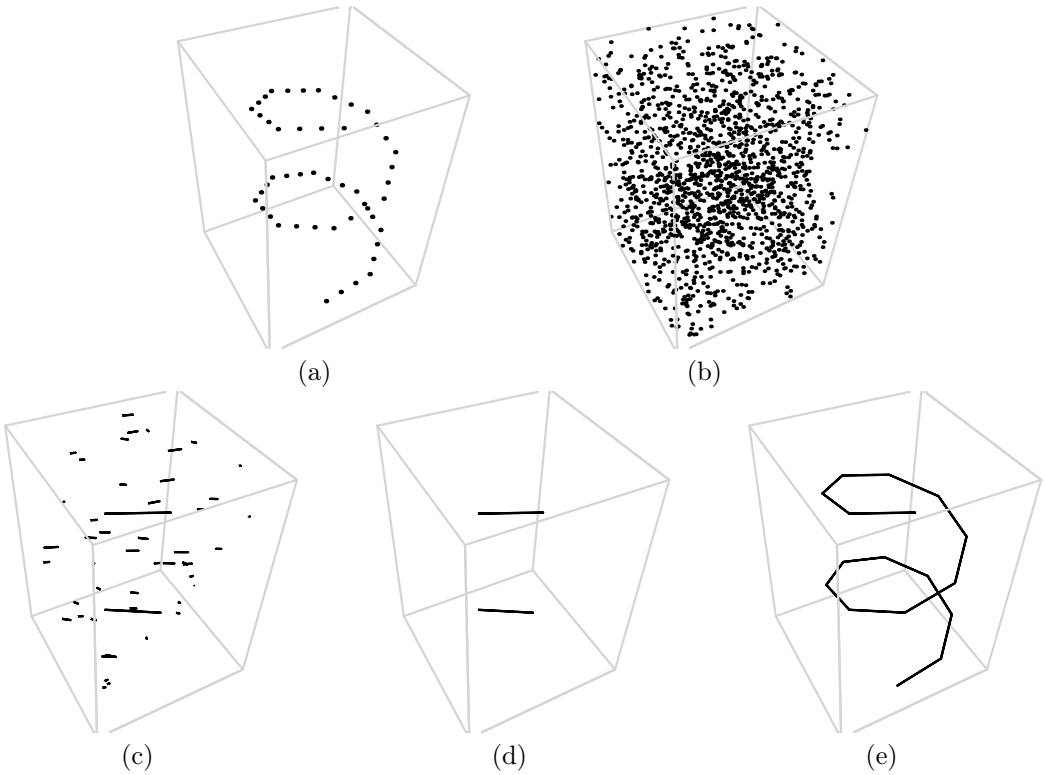


Fig. 3.13: Détection de structures polygonales ($k = 1$) avec $\alpha = 1$, $\lambda_3 = 10$ et $nb_c = 4$. (a) Un ensemble de 50 points. (b) L'ensemble est plongé dans un milieu bruité aléatoirement (1500 points). (c) Exemple d'une α-forme anisotrope significative. (d) Les structures non-significatives de (c) ont été supprimées. (e) Le résultat de la détection.

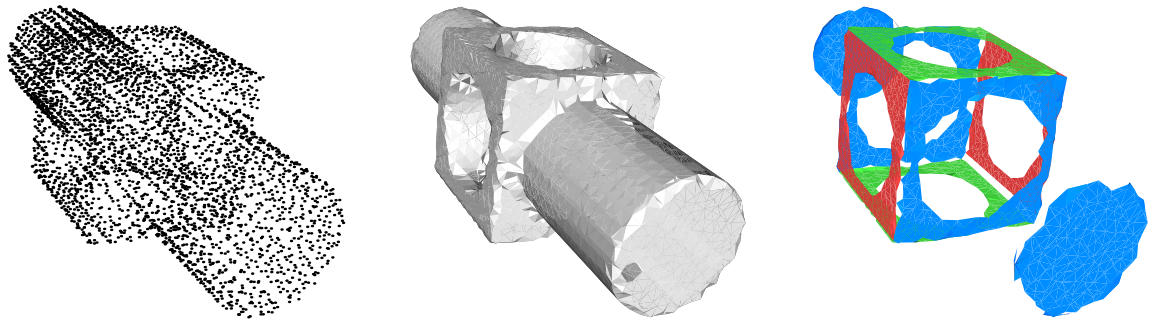
Publications associées à la Partie I

[Bougleux *et al.*, 2007d]. S. Bougleux, M. Melkemi et A. Elmoataz.
Structure Detection From A 3D Set Of Points With Anisotropic Alpha–Shapes. *International Journal of Image and Graphics*, Vol. 7(4), pages 689–708, World Scientific, 2007.

[Bougleux *et al.*, 2007c] S. Bougleux, M. Melkemi et A. Elmoataz.
Local Beta-Crusts for Simple Curves Reconstruction. *4th International Symposium on Voronoi Diagrams in Science and Engineering*, édité par C. M. Gold, pages 48–57, IEEE Computer Society, 2007.

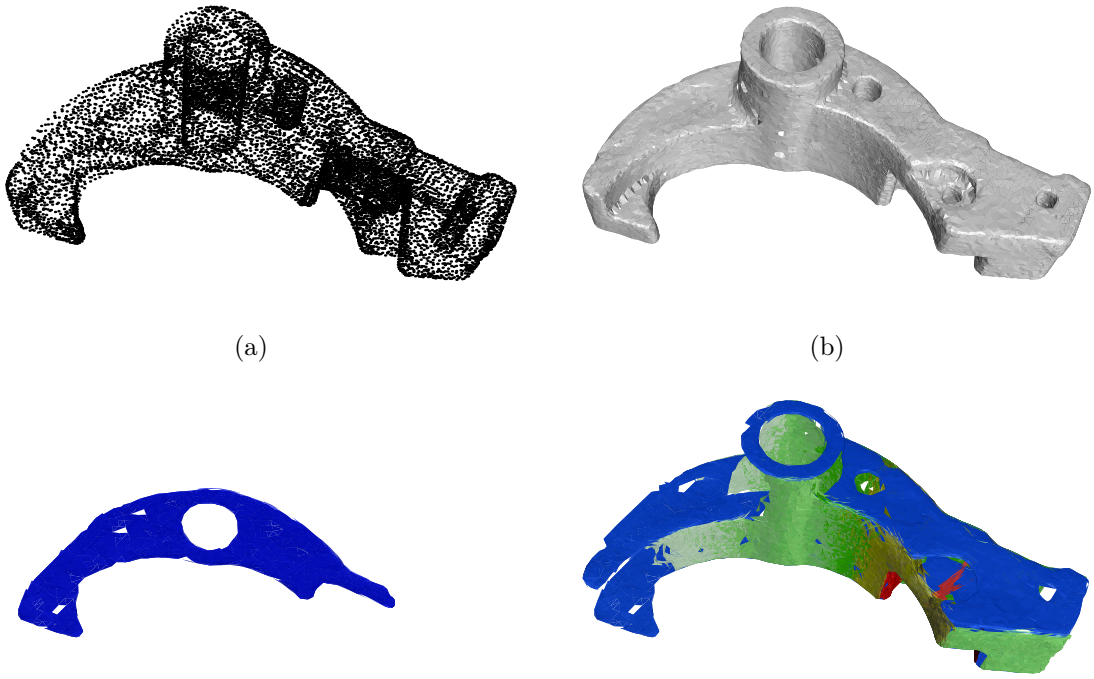
[Bougleux *et al.*, 2006] S. Bougleux, M. Melkemi et A. Elmoataz.
Geometrical Algorithms to Detect Patterns from a Set of Points. *3rd International Symposium on Voronoi Diagrams in Science and Engineering*, édité par M. L. Gavrilova, pages 94–101, IEEE Computer Society, 2006.

[Bougleux *et al.*, 2005] S. Bougleux, M. Melkemi et A. Elmoataz.
Structure Detection From A 3D Set Of Points With Anisotropic Alpha-Shapes. *2nd International Conference on Image Analysis and Recognition*. LNCS Vol. 3656, édité par M. S. Kamel et A. Campilho, Springer, pages 651–658, 2005.



(a) (b) (c)

Fig. 3.14: Détection de structures planaires. (a) L'ensemble de points P ($|P| = 4102$). (b) Une α-forme Euclidienne de P. (c) Structures planaires détectées par notre méthode avec $nb_c = 400$ et $\lambda_2 = \lambda_3 = 20$.



(a) (b)

(c) (d)

Fig. 3.15: Détection de structures planaires avec $\lambda_2 = \lambda_3 = 10$. (a) L'ensemble P ($|P| = 12593$). (b) Une α-forme Euclidienne de P. (c) La structure planaire détectée ayant la longueur maximum (2350 composants connectés). (d) Toutes les structures planaires détectées avec $nb_c = 600$.

Deuxième partie

Régularisation de Données Discrètes sur Graphes

Résumé et Contributions

Cette partie concerne le problème du traitement et de la simplification de données organisées ou non. Ce problème est traité par des méthodes variationnelles discrètes sur graphes de topologie quelconque. Dans ce contexte, nous proposons un cadre général qui unifie les méthodes basées sur des fonctionnelles discrètes de régularisation locale et non-locale.

La principale contribution de ce travail est l'introduction d'une large famille de problèmes de régularisation, basée sur les p-Laplaciens isotrope et anisotrope sur graphes pondérés, analogues discrets des p-Laplaciens Euclidiens isotrope et anisotrope. La résolution de ces problèmes mènent à des processus de diffusion discrète, linéaire ou non-linéaire. Nous montrons que ce formalisme permet de retrouver la version discrète de plusieurs méthodes continues de régularisation locale et non-locale, qui interviennent en traitement d'images dans des problèmes de restauration et de débruitage. Contrairement à ces méthodes, les processus de diffusion ne font pas intervenir d'équations aux dérivées partielles, ce qui permet de s'affranchir des problèmes de discrétisation spatiale. D'autre part, plusieurs méthodes de filtrage de données, issues de la théorie spectrale des graphes, sont des cas particuliers de notre approche. Ces méthodes, qui par définition font intervenir des interactions locales ou non-locales, utilisent généralement le Laplacien isotrope discret ($p = 2$).

Le comportement de la régularisation pour $p \neq 2$, est illustré avec plusieurs applications de traitement d'images, de courbes et de surfaces polygonales, ainsi que de données discrètes non-organisées. En particulier, lorsque $p \leq 1$, la régularisation se comporte comme un processus de lissage qui préserve les discontinuités principales, et pour $p \to 0$ elle agit comme un processus de regroupement. Ainsi, la régularisation proposée offre une extension naturelle pour traiter des données, ou des fonctions discrètes, définies sur des domaines réguliers ou irréguliers, ou dans des espaces de grande dimension.

Introduction

La régularisation d'images, de maillages ou plus généralement de données discrètes, par des méthodes variationnelles, a été appliquée avec succès pour résoudre différents problèmes intervenant dans des domaines tels que la vision par ordinateur, l'informatique graphique, ou encore l'analyse de données. Dans la majorité des domaines, les données manipulées, provenant d'observations ou de mesures, ont subi un certain nombre de dégradations. La nature des dégradations dépend de l'environnement et des méthodes d'acquisition (bruit, quantification, discrétisation, ...). La régularisation de ces données est donc une étape importante qui facilite leur traitement et leur analyse. Dans ce cadre, l'objectif principal de la régularisation est de fournir une approximation des données réelles à partir des données observées, souvent en introduisant de l'information *a priori*. Plus généralement, son utilisation intervient pour fournir une solution à des problèmes inverses et mal-posés (ou mal-conditionnés), au sens de Hadamard [Tikhonov et Arsenin, 1977].

En particulier, dans le contexte de la vision par ordinateur, et du traitement d'images, un grand nombre de problèmes sont par nature inverses et mal-posés [Poggio *et al.*, 1984, Alvarez *et al.*, 1993, Bertero et Boccacci, 1998]. C'est le cas de la restauration, de la segmentation, de la stéréo-vision ou encore de la reconstruction. La régularisation offre donc un cadre mathématique général pour rendre ces problèmes bien posés. Une méthodologie classique consiste à représenter les images discrètes par des fonctions continues définies sur un domaine continu Ω. L'image observée $f^0 : \Omega \to \mathbb{R}^m$ est alors modélisée par la somme de l'image recherchée $f : \Omega \to \mathbb{R}^m$, convoluée par un opérateur K (généralement un opérateur de flou), et d'une image résiduelle $r : \Omega \to \mathbb{R}^m$:

$$f^0 = K * f + r.$$

Ensuite, l'image f est reconstruite en minimisant une fonctionnelle de la forme :

$$\mathcal{E}(f, f^0, \lambda) := \mathcal{E}_{\text{régu}}(f) + \lambda \mathcal{E}_{\text{approx}}(f, f^0),$$

où $\mathcal{E}_{\text{régu}}$ est un terme de régularisation, ou énergie interne, et $\mathcal{E}_{\text{approx}}$ est un terme d'approximation, ou énergie externe. Le paramètre de fidélité $\lambda \geq 0$, appelé multiplicateur de Lagrange, contrôle la quantité d'approximation mesurée. La minimisation de l'énergie $\mathcal{E}$ revient alors à trouver la fonction f, suffisamment régulière sur Ω, tout en étant suffisamment proche de la fonction f^0 observée. La solution de ce problème est généralement obtenue en considérant l'équation d'Euler-Lagrange associée à l'énergie $\mathcal{E}$. Puis, les équa-

tions différentielles résultantes, comme l'équation de diffusion ou l'équation d'équilibre, sont discrétisées pour s'adapter au domaine discret initial [Chan et Shen, 2005, Aubert et Kornprobst, 2006].

Pour illustrer cette technique de régularisation d'images, nous considérons une importante famille de modèles variationnels basés sur la p-forme de Dirichlet :

$$\min_{f:\Omega\to\mathbb{R}^m} \left\{ \mathcal{E}(f, f^0, \lambda, p) := \frac{1}{p}\int_{x\in\Omega} \| \bigtriangledown f\|_2^p dx + \frac{\lambda}{2}\|f^0 - f\|_2^2 \right\}, \tag{4}$$

où $\bigtriangledown f$ est le gradient de la fonction f, et $\| \bigtriangledown f\|_2$ est sa variation locale basée sur la norme $\mathcal{L}_2$. Le premier terme est le terme de régularisation, et le deuxième terme est le terme d'approximation. L'équation d'Euler-Lagrange associée au problème de minimisation (4) s'écrit :

$$- div\left(\| \bigtriangledown f\|_2^{p-2} \bigtriangledown f\right) - \lambda(f^0 - f) = 0,$$

où div est l'opérateur divergence. Une façon usuelle de trouver la solution à cette équation est de considérer l'équation de diffusion qui correspond à la descente infinitésimale :

$$\begin{cases} f^{(0)} = f^0, \\ \frac{\partial}{\partial t} f^{(t)} = div\left(\| \bigtriangledown f\|_2^{p-2} \bigtriangledown f\right) + \lambda(f^0 - f). \end{cases}$$

Cette dernière équation traduit une famille de flots géométriques, dont l'utilisation dépasse largement le domaine du traitement d'images. Lorsque $p = 2$, le problème de minimisation (4) correspond à la régularisation de Tikhonov [Tikhonov et Arsenin, 1977], et l'équation de diffusion fait intervenir l'opérateur de Laplace ou l'opérateur de Laplace-Beltrami selon la nature du domaine Ω. Un autre cas intéressant de ce modèle est donné pour $p = 1$, où la régularisation est basée sur la variation totale (notée TV) [Rudin *et al.*, 1992], et la diffusion fait intervenir l'opérateur courbure. Ce dernier modèle, ainsi que ses variantes, sont largement utilisés en traitement d'images.

Dans le contexte du traitement et de l'analyse de courbes et de surfaces polygonales, les flots géométriques, issus de la minimisation de fonctionnelles basées sur la régularisation, constituent également une méthodologie commune pour fournir une solution à différents problèmes. Sans être exhaustif, c'est le cas du calcul de surfaces minimales [Pinkall et Polthier, 1993], de l'évolution et du débruitage de surfaces [Taubin, 1995, Taubin, 2000, Desbrun *et al.*, 1999, Ohtake *et al.*, 2001, Bajaj et Xu, 2003, Hildebrandt et Polthier, 2004], ou plus généralement de la modélisation de surfaces (voir par exemple [Xu *et al.*, 2006]). La plupart de ces flots sont basés sur la discrétisation de l'opérateur de Laplace-Beltrami sur des surfaces triangulaires (voir par exemple [Xu, 2004]), ou sur des diffusions préservant les structures, principalement inspirées par la diffusion anisotrope en traitement d'images.

Bien que les modèles variationnels continus, basés sur la régularisation, possèdent de nombreux avantages, ils souffrent d'un certain nombre de problèmes liés à la nature discrète des données et à leur dimension. Dans un grand nombre de cas, le domaine de

définition des données, ainsi que les données, sont supposés continus. Ceci nécessite de discrétiser les opérateurs différentiels qui interviennent dans les EDP de diffusion, introduisant une ambiguïté sur le choix de cette discrétisation. De plus, il est difficile de discrétiser ces opérateurs lorsque le domaine discret initial est de topologie irrégulière, ou lorsque les données ou les fonctions sont définies dans des espaces de grande dimension. D'autre part, ces opérateurs sont locaux. Ils ne tiennent pas compte des caractéristiques non-locales intrinsèques aux données, comme par exemple les détails fins ou les structures répétitives.

Depuis l'introduction du filtre NLMEANS (filtre à moyennes non-locales) [Buades *et al.*, 2005], proposé dans le contexte de la restauration d'images, plusieurs travaux ont montrés l'intérêt d'utiliser des interactions non-locales basées sur des patches de similarité [Kervrann *et al.*, 2007, Yoshizawa *et al.*, 2006]. Les interactions non-locales permettent de capturer les structures complexes présentes dans les données. Une interprétation variationnelle du filtre NLMEANS, ainsi que d'autres filtres de voisinage, a été développée par [Kinderman *et al.*, 2005] comme une fonctionnelle non-convexe. Récemment, une fonctionnelle quadratique non-locale a été proposée par [Gilboa et Osher, 2007a, Gilboa et Osher, 2007b] pour la régularisation d'images et la segmentation semi-supervisée. Ces modèles non-locaux se veulent plus flexibles et plus efficaces que les modèles locaux. De plus, ils peuvent être considérés comme les analogues non-locaux des modèles de régularisation basée sur la variation totale. Cependant, ils sont également développés dans un contexte continu, et la discrétisation des opérateurs différentiels reste un problème non-négligeable, particulièrement lorsque les données sont définies dans des espaces de grande dimension ou sur des domaines irréguliers. Notons que d'autres relations entre les filtres de voisinage et la diffusion non-linéaire ont également été montrées dans le contexte du traitement d'images [Barash, 2002, Mrázek *et al.*, 2006].

Pour apporter une solution à ces problèmes, nous considérons la régularisation directement dans un cadre discret. Il s'agit de considérer les images, les maillages, les données discrètes, ou toute fonction définie sur ces données, comme des fonctions définies sur un graphe de topologie arbitraire. Ensuite, nous redéfinissons des opérateurs de géométrie différentielle sur graphes, puis nous proposons l'analogue discret des méthodes variationnelles continues. Pour cela, on considère tout ensemble fini de données $\mathcal{X} := \{x_1, \ldots, x_N\}$, où $x_i \in \mathbb{R}^m$, modélisé par un graphe pondéré $G := (V, E, w)$ et une fonction $f : V \to \mathbb{R}^m$ qui associe une donnée de $\mathcal{X}$ à chaque sommet de l'ensemble V. Les données sont connectées par les arêtes de l'ensemble $E \subset V \times V$ avec un certain degré de similarité décrit par la fonction de poids $w : E \to \mathbb{R}^+$. Le graphe G représente alors le domaine sur lequel les données de $\mathcal{X}$ sont traitées. Soit $f^0 : V \to \mathbb{R}^m$ une fonction initiale. La régularisation de f^0 sur G peut se formaliser par le problème de minimisation suivant, analogue au problème (4) :

$$\min_{f:V\to\mathbb{R}^m} \left\{ E_w(f, f^0, \lambda, p) := \frac{1}{p} \sum_{u\in V} \| \nabla_w f(u)\|^p + \frac{\lambda}{2} \|f^0 - f\|_2^2 \right\}, \tag{5}$$

où $\nabla_w f(u)$ représente le gradient pondéré de la fonction f au sommet u du graphe. La solution du problème (5), mène à une famille de traitements itératifs linéaires et non-linéaires, sans avoir à résoudre d'EDP. En effet, elle s'exprime directement par des équations algébriques. Les équations de diffusion discrètes qui en découlent peuvent alors être exprimées par des systèmes d'équations différentielles ordinaires. Cette famille de traitements est paramétrée par le degré de régularité p, le multiplicateur de Lagrange λ, la structure du graphe, et la fonction de poids w. Selon la topologie du graphe, et le choix de w, la régularisation peut être locale, semi-locale ou non-locale. Nous montrons que de nombreuses méthodes continues de régularisation locale ou non-locale, avec une discrétisation donnée, peuvent s'interpréter comme un cas particulier du modèle discret proposé, comme le filtre bilatéral [Tomasi et Manduchi, 1998] ou le filtre NLMEANS [Buades *et al.*, 2005]. De plus, une telle formulation permet d'éviter les problèmes de discrétisation spatiale qui existent avec les modèles continus, même lorsque les données sont définies dans des espaces discrets irréguliers ou de grande dimension.

L'idée d'utiliser la régularisation ou la diffusion locale sur graphes non-pondérés a déjà été utilisée pour restaurer des données définies sur des grilles régulières [Osher et Shen, 2000, Chan *et al.*, 2001]. Les modèles variationnels discrets considérés dans ces travaux sont des cas particuliers de l'énergie E_w^p, pour $p = 1, 2$ et $w = 1$. D'autres méthodes existantes peuvent aussi être considérées comme des régularisations discrètes sur graphes non-pondérés [Chambolle, 2005, Darbon et Sigelle, 2004]. Elles mènent à des champs aléatoires de Markov, où des variables binaires sont impliquées dans la minimisation. Dans le contexte de l'analyse de données, un modèle variationnel sur graphes pondérés a également été utilisé pour traiter le problème de la classification semi-supervisée [Zhou et Schölkopf, 2004, Zhou et Schölkopf, 2005]. Le modèle que nous présentons ici est plus général. En particulier, pour $p = 2$, il permet d'établir le lien entre la régularisation discrète sur graphes pondérés, et le filtrage de Markov [Coifman *et al.*, 2005, Coifman *et al.*, 2006] par le biais de la théorie spectrale des graphes [Chung, 1997].

Dans la suite de cette partie, le chapitre 4 décrit le principe général de la régularisation de fonctions sur graphes pondérés. Le comportement de la régularisation proposée est ensuite analysé au chapitre 5 dans le contexte du traitement d'images, de maillages et plus généralement de données discrètes non-organisées. En particulier, nous montrons que la méthode s'applique efficacement à des problématiques de lissage, de débruitage et de simplification.

Chapitre 4

Régularisation de fonctions sur graphes pondérés

Ce chapitre présente les modèles variationnels sur graphes que nous proposons pour traiter des fonctions définies sur des domaines discrets. Ces modèles sont basés sur des familles de fonctionnelles de régularisation isotrope et anisotrope. Leur minimisation mène à des processus de diffusion semi-discrète et discrète, linéaire et non-linéaire. Les fonctionnelles de régularisation, ainsi que les processus de diffusion qui en découlent, sont définis à partir d'opérateurs discrets qui intègrent la topologie des graphes, la géométrie des fonctions et celle de leur domaine de définition.

4.1 Fonctions et opérateurs sur graphes

Cette section définit et décrit les principales propriétés des opérateurs élémentaires qui interviennent dans les méthodes de régularisation que nous proposons. Ces opérateurs peuvent être considérés comme les analogues sur graphes pondérés, des opérateurs différentiels définis dans l'espace Euclidien. Par analogie, nous montrons comment obtenir ces opérateurs à partir de la définition de l'opérateur différence. Cette démarche est similaire à celle développée en théorie spectrale des graphes [Biggs, 1974, Cvetković *et al.*, 1980, Chung, 1997], en analyse fonctionnelle sur graphes [Requardt, 1997, Bensoussan et Menaldi, 2005, Friedman et Tillich, 2004], et en classification semi-supervisée de données discrètes [Zhou et Schölkopf, 2004, Zhou et Schölkopf, 2005].

4.1.1 Espaces des fonctions définies sur un graphe

Nous considérons des graphes combinatoires pondérés, non-orientés, sans boucle et sans arête multiple. Soit $G := (V, E, w)$ un tel graphe, constitué d'un ensemble fini V de N sommets, et d'un ensemble fini $E \subset V \times V$ d'arêtes pondérées. Le poids d'une arête $(u, v) \in E$, noté $w_{uv} \in \mathbb{R}^+$, décrit la dissimilarité entre ses sommets u et v. Il est généralement obtenu à partir d'une mesure de similarité $w : V \times V \to \mathbb{R}^+$ qui respecte les trois propriétés suivantes :

(i) $\forall uv \in E$, $w(u, v) = w_{uv} \geq 0$ (non-négative),
(ii) $\forall uv \notin E$, $w(u, v) = 0$,
(iii) $\forall u, v \in V$, $w(u, v) = w(v, u)$ (symétrique).

Par convention, plus la valeur du poids est élevée et plus les sommets sont similaires. Au contraire, lorsque le poids vaut zéro, ou tend vers zéro, les sommets sont dissimilaires. Le **degré** d'un sommet $u \in V$ est la somme des poids des arêtes incidentes à ce sommet :

$$\delta_w(u) := \sum_{v \sim u} w_{uv},$$

où $v \sim u$ désigne les sommets $v \in V$ connectés à u par une arête de E. Il peut être vu comme une mesure de dissimilarité en un sommet du graphe.

Pour simplifier les notations et les calculs, nous considérons des fonctions (numériques) réelles $f : V \to \mathbb{R}$ qui attribuent à chaque sommet $u \in V$ une valeur réelle $f(u)$. Ces fonctions forment un espace discret de dimension N. Elles peuvent s'interpréter comme l'intensité d'un signal discret défini sur les sommets du graphe, et être représentées par des vecteurs de $\mathbb{R}^N$: $f = (f(v_1), \ldots, f(v_N))^T$, $v_i \in V$. Le cas des fonctions vectorielles $f : V \to \mathbb{R}^m$ est traité à la section 4.5. Par analogie avec les espaces fonctionnels continus, l'intégrale d'une fonction $f : V \to \mathbb{R}$ est usuellement définie par :

$$\int_V f := \sum_V f.$$

Soit $\mathcal{H}(V)$ l'espace de Hilbert de toutes les fonctions $f : V \to \mathbb{R}$. Il est muni du produit interne :

$$\langle f, h\rangle_{\mathcal{H}(V)} := \sum_{u\in V} f(u)h(u), \quad f, h : V \to \mathbb{R}, \tag{4.1}$$

et de la norme $\mathcal{L}_2$ induite $\|f\|_2 := \langle f, f\rangle_{\mathcal{H}(V)}^{1/2}$.

Il existe également des fonctions définies sur les arêtes du graphe. Soit $\mathcal{H}(E)$ l'espace de Hilbert des fonctions (numériques) à valeurs réelles $F : E \to \mathbb{R}$, définies sur les arêtes du graphe G. Il est muni du produit interne :

$$\langle F, H\rangle_{\mathcal{H}(E)} := \sum_{u\in V}\sum_{v\sim u} F(u, v)H(u, v), \quad F, H : E \to \mathbb{R}, \tag{4.2}$$

où $F(u, v)$ et $H(u, v)$ sont les valeurs des fonctions F et H en l'arête $uv \in E$. Remarquons que les fonctions F et H n'ont pas besoin d'être symétriques, et que le produit interne peut se réécrire[6] comme :

$$\langle F, H\rangle_{\mathcal{H}(E)} = \sum_{(u,v)\in E} F(u, v)H(u, v), \quad F, H : E \to \mathbb{R}. \tag{4.3}$$

La norme $\mathcal{L}_2$ d'une fonction $F \in \mathcal{H}(E)$, induite de la définition du produit interne, est définie par $\|F\|_2 := \langle F, F\rangle_{\mathcal{H}(E)}^{1/2}$.

Remarque. *D'autres expressions du produit interne dans $\mathcal{H}(V)$ et $\mathcal{H}(E)$ peuvent également être considérées, comme par exemple $\langle f, h\rangle_{\mathcal{H}(V)} := \sum_{u\in V} f(u)h(u)\mu(u)$, où μ est une mesure sur G.*

4.1.2 Opérateur différence, dérivée directionnelle et adjoint

Tous les opérateurs élémentaires considérés par la suite, sont définis à partir de l'opérateur différence ou de la dérivée directionnelle, analogues sur graphes au taux d'accroissement d'une fonction continue.

Soit $G := (V, E, w)$ un graphe pondéré. L'opérateur **différence pondérée** d'une fonction $f : V \to \mathbb{R}$, noté $d_w : \mathcal{H}(V) \to \mathcal{H}(E)$, est défini en une arête $uv \in E$ par l'expression :

$$(d_w f)(u, v) := \frac{f(v) - f(u)}{\rho_w(u, v)},$$

où $\rho_w : E \to \mathbb{R}^+$ est une mesure de distance qui fait intervenir la fonction de poids w. Le choix de ρ_w, tout comme celui de la fonction de poids w, dépend du contexte applicatif. Dans la suite, nous utilisons $\rho_w = 1/\sqrt{w}$ comme mesure de similarité, principalement pour respecter les propriétés de la dérivée d'une fonction continue, et des opérateurs différentiels qui en découlent. L'opérateur différence considéré dans toute la partie II,

[6] Pour chaque arête uv de E, l'arête vu appartient aussi à E.

s'écrit alors :

$$(d_w f)(u,v) := \sqrt{w_{uv}}(f(v) - f(u)). \tag{4.4}$$

La **dérivée partielle** (ou **directionnelle**) d'une fonction $f \in \mathcal{H}(V)$, par rapport à l'arête $e = (u,v) \in E$, notée $\partial/\partial e : \mathcal{H}(V) \to \mathbb{R}$, est définie en un sommet $u \in V$ par l'expression :

$$\left.\frac{\partial f}{\partial e}\right|_u := \partial_v f(u) := (d_w f)(u,v). \tag{4.5}$$

Remarquons que l'opérateur différence, et par conséquent la dérivée directionnelle, sont linéaires. De plus, comme la fonction w est symétrique, la dérivée directionnelle est consistante avec la définition de la dérivée d'une fonction continue. En particulier, on peut observer les trois propriétés suivantes :

(i) si $f(u) = f(v)$ alors $\partial_u f(v) = 0$ ($\Rightarrow$ la dérivée d'une fonction constante est nulle),

(ii) $\partial_u f(v) = -\partial_v f(u)$ (anti-symétrie),

(iii) $\partial_u f(u) = 0$.

Remarque. *En fonction du choix de la fonction w ou de la fonction ρ_w, on retrouve différentes expressions de l'opérateur différence. Par exemple, sa définition standard sur une grille régulière carrée s'écrit $(df)(i,j) := f(j) - f(i)$. De plus, on peut remarquer que lorsque la fonction ρ_w n'est pas symétrique, la propriété (ii) n'est pas respectée. C'est le cas par exemple sur des graphes orientés avec $\rho_w = 1/\sqrt{w}$, où la valeur de w_{uv} n'est pas nécessairement égale à celle de w_{vu}. Il existe également d'autres expressions de l'opérateur différence, comme celle considérée dans le contexte de la classification semi-supervisée [Zhou et Schölkopf, 2004, Zhou et Schölkopf, 2005] :*

$$(d_w f)(u,v) := \sqrt{w_{uv}}\left(\frac{f(v)}{\sqrt{\delta_w(v)}} - \frac{f(u)}{\sqrt{\delta_w(u)}}\right).$$

Comme le degré de u et le degré de v ne sont généralement pas identiques, cette dernière définition ne respecte pas la propriété (i).

L'opérateur **adjoint** de l'opérateur différence (4.4), noté $d_w^* : \mathcal{H}(E) \to \mathcal{H}(V)$, est défini par la relation suivante :

$$\langle H, d_w f\rangle_{\mathcal{H}(E)} := \langle d_w^* H, f\rangle_{\mathcal{H}(V)}, \quad \forall f \in \mathcal{H}(V), H \in \mathcal{H}(E). \tag{4.6}$$

À partir des expressions des produits internes dans $\mathcal{H}(E)$ et $\mathcal{H}(V)$, et de celle de l'opérateur différence, on peut exprimer l'opérateur adjoint en un sommet du graphe.

Proposition 1 *L'opérateur adjoint d^*, d'une fonction $H \in \mathcal{H}(E)$, s'exprime localement*

en un sommet $u \in V$ par :

$$(d_w^* H)(u) = \sum_{v \sim u} \sqrt{w_{uv}}(H(v,u) - H(u,v)). \tag{4.7}$$

Preuve. D'après l'expression du produit interne dans l'espace $\mathcal{H}(E)$ et de celle de l'opérateur différence, la partie gauche de la définition de l'opérateur adjoint (équation (4.6)) s'écrit :

$$\begin{aligned}
\langle H, d_w f\rangle_{\mathcal{H}(E)} &\overset{(4.3),(4.4)}{=} \sum_{(u,v)\in E} \sqrt{w_{uv}} H(u,v)\left(f(v) - f(u)\right), \\
&= \sum_{(u,v)\in E} \sqrt{w_{uv}} H(u,v) f(v) - \sum_{(u,v)\in E} \sqrt{w_{uv}} H(u,v) f(u) \\
&\overset{(4.2)}{=} \sum_{u\in V}\sum_{v\sim u} \sqrt{w_{uv}} H(v,u) f(u) - \sum_{u\in V}\sum_{v\sim u} \sqrt{w_{uv}} H(u,v) f(u) \\
&= \sum_{u\in V} f(u) \sum_{v\sim u} \sqrt{w_{uv}}\left(H(v,u) - H(u,v)\right) \\
&\overset{(4.6)}{=} \langle d_w^* H, f\rangle_{\mathcal{H}(V)} \overset{(4.1)}{=} \sum_{u\in V} (d_w^* H)(u) f(u).
\end{aligned}$$

Alors, en se plaçant en un sommet $u \in V$, on obtient l'équation (4.7). □

L'adjoint est un opérateur linéaire qui mesure le flux d'une fonction de $\mathcal{H}(E)$ en chaque sommet du graphe. Par analogie avec les opérateurs différentiels continus, l'opérateur **divergence** d'une fonction $F \in \mathcal{H}(V)$ est défini par :

$$div_w F := -d_w^* F.$$

À partir de cette définition, il est alors facile de vérifier la propriété de flux nul.

Propriété 18 *Toute fonction $F \in \mathcal{H}(E)$ à un flux nul : $\sum_{u\in V}(div_w F)(u) = 0$.*

Preuve. D'après la Proposition 1, $\sum_{u\in V}(d_w^* H)(u)$ est la somme de termes $\sqrt{w_{uv}}(H(v,u) - H(u,v)) + \sqrt{w_{vu}}(H(u,v) - H(v,u))$, pour tout $u, v \in V$. Comme la fonction w est symétrique, on en déduit la propriété. □

4.1.3 Opérateur gradient

L'opérateur **gradient pondéré** ∇_w d'une fonction $f \in \mathcal{H}(V)$ en un sommet $u \in V$, est le vecteur défini par :

$$\nabla_w f(u) := (\partial_v f(u),\, v \sim u)^T = (\partial_{v_1} f(u), \ldots, \partial_{v_k} f(u))^T, \qquad v_i \sim u. \tag{4.8}$$

Cette définition ne dépend pas de la structure du graphe, et donc le gradient à la même expression générale quelque soit la nature du graphe (régulier, irrégulier, géométrique, combinatoire, ...). L'information relative à la nature du graphe, ou à la fonction f, peut

être incorporée dans la fonction de poids w.

Remarque. *Le gradient sur graphe est l'analogue discret pondéré du gradient d'une fonction différentiable dans l'espace Euclidien $\mathbb{R}^n$, qui s'exprime en un point $x \in \mathbb{R}^n$ par :*

$$\bigtriangledown f(x) := \left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n}\right)^T.$$

Ici, les dérivées partielles sont selon le système de coordonnées, qui est global à tous les points de $\mathbb{R}^n$. Dans le cas des graphes, en chaque sommet, il existe autant de dérivées partielles que d'arêtes incidentes à ce sommet (au plus $N-1$).

Plusieurs normes du gradient peuvent être définies pour mesurer la **régularité** d'une fonction $f \in \mathcal{H}(V)$ en chaque sommet du graphe. Nous considérons l'expression de la norme $\mathcal{L}_p$ pour $p \in]0, +\infty[$:

$$\| \bigtriangledown_w f(u)\|_p := \left(\sum_{v\sim u} |\partial_v f(u)|^p\right)^{\frac{1}{p}} \overset{(4.5)}{=} \left(\sum_{v\sim u} w_{uv}^{\frac{p}{2}} |f(v) - f(u)|^p\right)^{\frac{1}{p}}. \tag{4.9}$$

En particulier, pour $p = 2$ on a la norme $\mathcal{L}_2$, qui se réécrit :

$$\| \bigtriangledown_w f(u)\|_2 = \sqrt{\sum_{v\sim u} (\partial_v f(u))^2} = \sqrt{\sum_{v\sim u} w_{uv}(f(v) - f(u))^2}, \tag{4.10}$$

et pour $p = 1$, la norme $\mathcal{L}_1$ (norme de Manhattan) :

$$\| \bigtriangledown_w f(u)\|_1 = \sum_{v\sim u} |\partial_v f(u)| = \sum_{v\sim u} \sqrt{w_{uv}} |f(v) - f(u)|. \tag{4.11}$$

Pour $p \geq 1$, on peut facilement vérifier que l'expression (4.9) est une semi-norme :

(i) $\| \bigtriangledown_w (af)(u)\|_p = |a| \, \| \bigtriangledown_w f(u)\|_p$, $\forall a \in \mathbb{R}$, $\forall f \in \mathcal{H}(V)$,

(ii) $\| \bigtriangledown_w (f + h)(u)\|_p \leq \| \bigtriangledown_w f(u)\|_p + \| \bigtriangledown_w h(u)\|_p$, $\forall f, h \in \mathcal{H}(V)$,

(iii) $\| \bigtriangledown_w f(u)\|_p = 0 \Leftrightarrow (f(v) = f(u))$ ou $(w_{uv} = 0)$, $\forall v \sim u$.

En revanche, pour $p < 1$ l'expression (4.9) n'est pas une norme, l'inégalité triangulaire n'est pas satisfaite (propriété (ii)).

Les normes ci-dessus mesurent l'amplitude du gradient en un sommet du graphe, ou la **variation locale** de la fonction f en un de ses sommets. Remarquons que cette variation dépend d'un voisinage local à chaque sommet. Le voisinage peut décrire différentes sortes d'interactions entre les sommets du graphe. Néanmoins, comme $w_{uv} = 0$ si le sommet u n'est pas connecté à v par une arête de E, la variation locale peut se réécrire

en considérant des **interactions non-locales** :

$$\| \bigtriangledown_w^{NL} f(u)\|_p := \left(\sum_{v \in V} |\partial_v f(u)|^p \right)^{\frac{1}{p}} = \| \bigtriangledown_w f(u)\|_p. \tag{4.12}$$

Cette formulation revient à considérer l'ensemble V comme un ensemble de données discrètes, et à former le graphe complet où $E = V \times V \setminus \{vv \ : \ v \in V\}$.

Remarque. *La variation locale, basée sur la norme $\mathcal{L}_p$ et exprimée sous forme non-locale, est l'analogue continu de la norme du gradient non-local d'une fonction $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}$:*

$$\| \bigtriangledown_w f(x)\|_p := \left(\int_\Omega w(x,y)^{\frac{p}{2}} |f(y) - f(x)|^p dy \right)^{\frac{1}{p}}.$$

D'autres normes usuelles peuvent également être formalisées à partir de la dérivée directionnelle, comme par exemple la norme infinie $\| \bigtriangledown_w f(u)\|_\infty := \max_{v \sim u} |\partial_v f(u)|$.

4.1.4 Opérateur Laplacien

Le Laplacien est un opérateur du second ordre, qui est utilisé dans de nombreux traitements, en particulier dans des processus de diffusion. Il est directement lié à la théorie spectrale des graphes, qui consiste à étudier les valeurs et les vecteurs propres de la matrice Laplacienne [Chung, 1997]. Selon la nature du graphe et des données à traiter, le Laplacien prend différentes expressions. Principalement, nous pouvons distinguer le Laplacien combinatoire des Laplaciens normalisés. Dans cette thèse, nous considérons le Laplacien combinatoire, dont l'expression matricielle est rappelée dans ce qui suit.

Soit $G := (V, E, w)$ un graphe pondéré. Le poids des arêtes peut se représenter par une matrice d'adjacence W définie par :

$$W(u,v) := \begin{cases} w_{uv} & \text{si } v \sim u, \\ 0 & \text{sinon.} \end{cases} \tag{4.13}$$

D'après la définition de la fonction w, la matrice W est symétrique et semi-définie positive. Soit D la matrice diagonale qui représente le degré des sommets du graphe :

$$D(u,v) := \begin{cases} \delta_w(v) & \text{si } u = v, \\ 0 & \text{sinon.} \end{cases} \tag{4.14}$$

Soit $L := D - W$ la **matrice Laplacienne** dont les éléments sont donnés par :

$$L(u,v) := \begin{cases} \delta_w(v) & \text{si } u = v, \\ -w_{uv} & \text{si } v \sim u, \\ 0 & \text{sinon.} \end{cases} \tag{4.15}$$

Soit $f : V \to \mathbb{R}$ une fonction de $\mathcal{H}(V)$, et $\mathbf{f}$ sa représentation vectorielle. En particulier, nous avons l'expression locale suivante du Laplacien de f en un sommet $u \in V$:

$$L\mathbf{f}(u) = \sum_{v \in V} w(u,v)(\mathbf{f}(u) - \mathbf{f}(v)). \tag{4.16}$$

Une fonctionnelle de régularisation, couramment utilisée et associée à la matrice Laplacienne, est la suivante :

$$R(\mathbf{f}) := \mathbf{f}^T L\mathbf{f} = \mathbf{f}^T (D - W)\mathbf{f} = \sum_{u \in V} \sum_{v \in V} w(u,v)(\mathbf{f}(u) - \mathbf{f}(v))^2. \tag{4.17}$$

La minimisation de cette dernière revient à calculer les vecteurs propres de la matrice $D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$. Cette fonctionnelle et ses variantes [Chung, 1997], les processus de diffusion qui y sont associés, ainsi que l'étude des propriétés des Laplaciens matriciels (valeurs et vecteurs propres), sont à la base de nombreuses méthodes d'analyse de données discrètes, comme par exemple la réduction de dimensions, le regroupement, la segmentation d'images, ou encore la classification semi-supervisée, voir par exemple [Belkin et Niyogi, 2001, Lafon et Lee, 2006, Shi et Malik, 2000, Smola et Kondor, 2003, Belkin *et al.*, 2004, Zhou et Schölkopf, 2004] et les références qui y sont associées.

Nous ne détaillons pas ici cette approche matricielle, nous nous intéressons à une approche fonctionnelle. Pour cela, nous montrons maintenant comment obtenir l'expression locale du Laplacien à partir de l'opérateur différence et de son adjoint.

L'opérateur **Laplacien combinatoire** $\Delta_w : \mathcal{H}(V) \to \mathcal{H}(V)$, d'une fonction $f \in \mathcal{H}(V)$, est défini par :

$$\Delta_w f := \tfrac{1}{2} d_w^*(d_w f). \tag{4.18}$$

En remplaçant les expressions de d_w et de d_w^* dans cette dernière équation, nous obtenons l'expression locale (4.16) du Laplacien.

Proposition 2 *L'opérateur Laplacien Δ_w, d'une fonction $f \in \mathcal{H}(V)$, s'exprime en sommet $u \in V$ par :*

$$\Delta_w f(u) = \sum_{v \sim u} w_{uv} \left(f(u) - f(v) \right), \tag{4.19}$$

ou de manière équivalente par :

$$\Delta_w f(u) = \sum_{v \sim u} \partial_v^2 f(u) = -\sum_{v \sim u} \sqrt{w_{uv}} \partial_v f(u).$$

Preuve. À partir des équations (4.18), (4.4) et (4.7), nous obtenons :

$$\Delta_w f(u) = \tfrac{1}{2} \sum_{v \sim u} \sqrt{w_{uv}}((d_w f)(v,u) - (d_w f)(u,v)) = \sum_{v \sim u} \sqrt{w_{uv}} (d_w f)(v,u). \ \square$$

Comme l'opérateur différence et son adjoint sont linéaires, le Laplacien est un opérateur

linéaire. Il est également auto-adjoint :

$$\begin{aligned}\langle \Delta_w f, h\rangle_{\mathcal{H}(V)} &\overset{(4.22)}{=} \langle d_w^*(d_w f), h\rangle_{\mathcal{H}(V)} \overset{(4.6)}{=} \langle d_w f, d_w h\rangle_{\mathcal{H}(E)} \\ &\overset{(4.6)}{=} \langle f, d_w^*(d_w h)\rangle_{\mathcal{H}(V)} \overset{(4.22)}{=} \langle f, \Delta_w h\rangle_{\mathcal{H}(V)},\end{aligned} \tag{4.20}$$

et semi-défini positif :

$$\langle \Delta_w f, f\rangle_{\mathcal{H}(V)} \overset{(4.20)}{=} \langle d_w f, d_w f\rangle_{\mathcal{H}(E)} \overset{(4.2)}{=} \sum_{u\in V}\sum_{v\sim u}(d_w(f)(u,v))^2 \overset{(4.10)}{=} \sum_{u\in V} \| \bigtriangledown_w f(u)\|_2^2 \geq 0. \tag{4.21}$$

Remarque. *Dans le cas d'une fonction à valeurs réelles $f : \mathbb{R}^m \to \mathbb{R}$, définie dans l'espace Euclidien $\mathbb{R}^m$, le Laplacien est un opérateur de second ordre défini comme la divergence du gradient, $\Delta := div(\bigtriangledown) = \bigtriangledown^2$, et en un point $x \in \mathbb{R}^m$ par :*

$$\Delta f(x) = \sum_{i=1}^{m} \frac{\partial^2 f}{\partial x_i^2}.$$

L'équation (4.21) est alors l'analogue pondéré discret de la relation entre le Laplacien et le gradient continus :

$$\int_\Omega f\Delta f = \int_\Omega \| \bigtriangledown f\|_2^2.$$

Dans la section suivante, nous généralisons la définition du Laplacien combinatoire à celle de deux familles de p-Laplaciens.

4.2 Opérateurs proposés : p-Laplaciens sur graphes

Dans cette section, nous proposons deux familles de p-Laplaciens isotropes et anisotropes discrets, qui généralisent la définition de l'opérateur Laplacien combinatoire. Ces p-Laplaciens sont les analogues, sur graphes pondérés $G := (V, E, w)$, des p-Laplaciens isotropes et anisotropes définis pour des fonctions continues de l'espace Euclidien. Les familles proposées interviennent directement dans la solution des problèmes de régularisation qui sont traités à la section 4.3.

4.2.1 Le p-Laplacien isotrope

L'opérateur p-Laplacien (isotrope) décrit une famille d'opérateurs du second ordre. Cette famille inclue l'opérateur Laplacien combinatoire pour $p = 2$ et l'opérateur courbure pour $p = 1$. Nous décrivons dans un premier temps le cas où $p \in]0, +\infty[$ est quelconque.

L'opérateur **p-Laplacien isotrope** (pondéré) d'une fonction $f \in \mathcal{H}(V)$, noté $\Delta_w^p : \mathcal{H}(V) \to \mathcal{H}(V)$, est défini par :

$$\Delta_w^p f := \tfrac{1}{2} d_w^* \left(\| \nabla_w f \|_2^{p-2} d_w f \right), \quad p \in]0, +\infty[. \tag{4.22}$$

C'est un opérateur non-linéaire, excepté dans le cas où $p = 2$ pour lequel nous retrouvons la définition de l'opérateur Laplacien combinatoire de la section 4.1.4. En remplaçant les expressions de l'opérateur différence et de son adjoint dans l'équation (4.22), on peut calculer l'expression de l'opérateur p-Laplacien en un sommet du graphe.

Proposition 3 *L'opérateur p-Laplacien isotrope Δ_w^p, d'une fonction $f \in \mathcal{H}(V)$, s'exprime en sommet $u \in V$ par :*

$$\Delta_w^p f(u) = \tfrac{1}{2} \sum_{v \sim u} w_{uv} \left(\| \nabla_w f(v) \|_2^{p-2} + \| \nabla_w f(u) \|_2^{p-2} \right) (f(u) - f(v)), \tag{4.23}$$

ou de manière équivalente par :

$$\Delta_w^p f(u) = \tfrac{1}{2} \sum_{\substack{v \sim u \\ e=(u,v)}} \frac{\partial}{\partial e} \left(\| \nabla_w f \|_2^{p-2} \frac{\partial f}{\partial e} \right) \Bigg|_u, \tag{4.24}$$

$$\Delta_w^p f(u) = -\tfrac{1}{2} \sum_{v \sim u} \sqrt{w_{uv}} (\| \nabla_w f(v) \|_2^{p-2} + \| \nabla_w f(u) \|_2^{p-2}) \partial_v f(u). \tag{4.25}$$

Preuve. À partir des équations (4.22), (4.4) et (4.7), nous obtenons :

$$\begin{aligned} \Delta_w^p f(u) &= \tfrac{1}{2} \sum_{v \sim u} \sqrt{w_{uv}} (\| \nabla_w f(v) \|_2^{p-2} (d_w f)(v, u) - \| \nabla_w f(u) \|_2^{p-2} (d_w f)(u, v)) \\ &= \tfrac{1}{2} \sum_{v \sim u} \sqrt{w_{uv}} (\| \nabla_w f(v) \|_2^{p-2} (d_w f)(v, u) + \| \nabla_w f(u) \|_2^{p-2} (d_w f)(v, u)) = (4.23). \end{aligned}$$

Montrons maintenant que l'équation (4.24) est équivalente à l'équation (4.23) :

$$\begin{aligned}(4.24) \quad &\overset{(4.5)}{=} \frac{1}{2} \sum_{\substack{v \sim u \\ e=(u,v)}} \sqrt{w_{uv}} \left(\| \nabla_w f(v) \|_2^{p-2} \frac{\partial f}{\partial e}\bigg|_v - \| \nabla_w f(u) \|_2^{p-2} \frac{\partial f}{\partial e}\bigg|_u \right) \\ &\overset{(4.5)}{=} \frac{1}{2} \sum_{\substack{v \sim u \\ e=(u,v)}} \sqrt{w_{uv}} \left(\sqrt{w_{uv}} \| \nabla_w f(v) \|_2^{p-2} (f(u) - f(v)) \right. \\ &\qquad \left. - \sqrt{w_{uv}} \| \nabla_w f(u) \|_2^{p-2} (f(v) - f(u)) \right) = (4.23).\end{aligned}$$

Enfin, l'équation (4.25) est une réécriture directe de l'équation (4.23). □

L'équation (4.24) et l'équation (4.25) montrent le lien entre les dérivées discrètes premières et secondes d'une fonction de $\mathcal{H}(V)$.

Nous avons vu que pour $p = 2$, le p-Laplacien donné par l'équation (4.22) est le Laplacien combinatoire. Lorsque $p = 1$, il correspond à l'**opérateur courbure** (pondérée) :

$$\kappa_w f := \tfrac{1}{2} d_w^* \left(\frac{d_w f}{\| \nabla_w f \|_2} \right) = \Delta_w^1 f.$$

À partir de la Proposition 3, l'opérateur courbure s'exprime en un sommet du graphe par :

$$\kappa_w f(u) \overset{(4.23)}{=} \frac{1}{2} \sum_{v \sim u} w_{uv} \left(\frac{1}{\| \nabla_w f(v) \|_2} + \frac{1}{\| \nabla_w f(u) \|_2} \right) (f(u) - f(v)) \tag{4.26}$$

$$\overset{(4.24)}{=} \frac{1}{2} \sum_{\substack{v \sim u \\ e=(u,v)}} \frac{\partial}{\partial e} \left(\frac{1}{\| \nabla_w f \|_2} \frac{\partial f}{\partial e} \right) \bigg|_u . \tag{4.27}$$

Un cas particulier de cet opérateur est obtenu sur des graphes non-pondérés ($w_{uv} = 1$ pour toute arête $(u, v) \in E$). Il a été proposé dans [Osher et Shen, 2000, Chan *et al.*, 2001], dans le contexte de la restauration d'images par minimisation de la variation totale. L'opérateur courbure (4.26), et plus généralement le p-Laplacien (4.23), peuvent être considérés comme des extensions directes de dernier, par la fonction de poids w et par le degré de régularité p.

Remarque. *L'opérateur courbure sur graphe est la version discrète pondérée de l'opérateur courbure d'une fonction continue $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}$ définie sur un domaine Ω par l'expression :*

$$\kappa f = -div \left(\frac{\nabla f}{\| \nabla f \|_2} \right).$$

Le p-Laplacien peut alors être vu comme une généralisation du Laplacien et de la courbure :

$$\Delta_p f = -div\left(\frac{\bigtriangledown f}{\| \bigtriangledown f\|_2^{2-p}}\right).$$

4.2.2 Le p-Laplacien anisotrope

Cette section définit une autre généralisation du Laplacien combinatoire, qui est basée sur les différences absolues locales. Nous définissons l'opérateur **p-Laplacien anisotrope** (pondéré) d'une fonction $f \in \mathcal{H}(V)$, noté $\widetilde{\Delta_w^p} : \mathcal{H}(V) \to \mathcal{H}(V)$, par l'expression :

$$\widetilde{\Delta_w^p} f := \tfrac{1}{2} d_w^*(|d_w f|^{p-2} d_w f), \quad \forall p \in]0, +\infty[, \tag{4.28}$$

avec $|(d_w f)(u, v)| = |f(v) - f(u)|$ pour toute arête $(u, v) \in E$.

Proposition 4 *L'opérateur p-Laplacien anisotrope $\widetilde{\Delta_w^p}$, d'une fonction $f \in \mathcal{H}(V)$, s'exprime en sommet $u \in V$ par :*

$$\begin{aligned} \widetilde{\Delta_w^p} f(u) &= \sum_{v \sim u} w_{uv}^{\frac{p}{2}} |f(u) - f(v)|^{p-2}(f(u) - f(v)) \\ &= \sum_{v \sim u} w_{uv}^{\frac{p}{2}} |f(u) - f(v)|^{p-1} sign(f(u) - f(v)), \end{aligned} \tag{4.29}$$

où $sign : \mathbb{R} \to \{-1, 1\}$ est la fonction signe.

Preuve. À partir de l'équation (4.28), on a :

$$\begin{aligned} \widetilde{\Delta_w^p} f(u) &\overset{(4.7)}{=} \tfrac{1}{2} \sum_{v \sim u} \sqrt{w_{uv}} \left(|(d_w f)(v, u)|^{p-2}(d_w f)(v, u) - |(d_w f)(u, v)|^{p-2}(d_w f)(u, v)\right) \\ &= \sum_{v \sim u} \sqrt{w_{uv}} \left(|(d_w f)(v, u)|^{p-2}(d_w f)(v, u)\right) \\ &\overset{(4.4)}{=} \sum_{v \sim u} w_{uv} \left(|\sqrt{w_{uv}}(f(u) - f(v))|^{p-2}(f(u) - f(v))\right) = (4.29). \ \square \end{aligned}$$

4.2.3 Expression générale et fonctions localement constantes

Les deux p-Laplaciens proposés dans les sections précédentes ont la même forme générale suivante :

$$L_w^p f(u) := \sum_{v \sim u} \gamma_w^f(u, v)(f(u) - f(v)), \tag{4.30}$$

où $\gamma_w^f : V \times V \to \mathbb{R}$ est une fonction qui dépend de la fonction de poids w, et de la fonction $f \in \mathcal{H}(V)$. Le tableau 4.1 récapitule les différentes expressions de γ_w^f que nous avons rencontrées.

Fonction $\gamma_w^f(u,v)$	*Opérateur* L_w^p
w_{uv}	Laplacien combinatoire [Chung, 1997]
$\left(\| \bigtriangledown f(v)\|_2^{-1} + \| \bigtriangledown f(u)\|_2^{-1}\right)$	courbure [Osher et Shen, 2000]
$\frac{1}{2}w_{uv}\left(\| \bigtriangledown_w f(v)\|_2^{p-2} + \| \bigtriangledown_w f(u)\|_2^{p-2}\right)$	p-Laplacien isotrope (cette thèse)
$w_{uv}^{\frac{p}{2}}\lvert f(u) - f(v)\rvert^{p-2}$	p-Laplacien anisotrope (cette thèse)

Tab. 4.1: Opérateurs p-Laplaciens associés à l'équation (4.30).

Remarquons que si la fonction $f \in \mathcal{H}(V)$ est localement constante, et que $p < 2$, les p-Laplaciens nécessitent d'être régularisés pour éviter la division par zéro :

(i) p-Laplacien isotrope : si $f(u) = f(v)$ pour tout sommet v voisin du sommet u, la variation locale peut être nulle. Pour cela, elle est régularisée par :

$$\| \bigtriangledown_w f(u)\|_{2,\epsilon} := \sqrt{\| \bigtriangledown_w f(u)\|_2^2 + \epsilon^2}, \quad \epsilon > 0. \tag{4.31}$$

(ii) p-Laplacien anisotrope : si $(d_w f)(u,v) = 0$ pour une arête $(u,v) \in E$, la différence absolue n'est pas définie. Dans ce cas, elle est régularisée par :

$$|f(u) - f(v)|_\epsilon := |f(u) - f(v)| + \epsilon, \quad \epsilon > 0. \tag{4.32}$$

Remarque. *Il existe d'autres expressions de l'opérateur p-Laplacien sur graphes (pondérés). Par exemple, dans [Zhou et Schölkopf, 2005] le p-Laplacien isotrope est défini comme une extension du Laplacien normalisé [Chung, 1997], qui est l'analogue sur graphes pondérés de l'opérateur de Laplace-Beltrami sur des variétés Riemannienne.*

Dans la suite, nous décrivons et analysons des problèmes variationnels discrets basés sur la variation totale (voir section 4.1.3). La résolution de ces problèmes fait intervenir les opérateurs de premier et de second ordre définis dans les sections précédentes.

4.3 Modèles variationnels proposés

Cette section présente les modèles variationnels que nous proposons pour régulariser des fonctions définies sur les sommets de graphes pondérés, ainsi que les équations de régularisation qui en découlent. Soient $G := (V, E, w)$ un graphe pondéré, et $f^0 : V \to \mathbb{R}$ une fonction donnée de $\mathcal{H}(V)$. La régularisation de la fonction f^0 consiste à trouver une fonction $f \in \mathcal{H}(V)$ qui soit proche de f^0, et suffisamment régulière sur G. Nous formalisons ce problème d'optimisation comme la minimisation de la somme de deux fonctionnelles de la forme :

$$\min_{f \in \mathcal{H}(V)} \left\{ \sum_{u \in V} \| \nabla_w f(u) \|_p^q + \tfrac{\lambda}{2} \| f - f^0 \|_2^2 \right\}. \tag{4.33}$$

La première fonctionnelle mesure la régularité de la fonction f sur le graphe G. Elle étend la fonctionnelle discrète basée sur la variation totale avec $p = 2$ et $q = 1$ (TV isotrope) [Osher et Shen, 2000]. La seconde fonctionnelle mesure l'écart entre la fonction f et la fonction initiale f^0. La compétition entre les deux termes est contrôlée par le paramètre de fidélité $\lambda \geq 0$, ou multiplicateur de Lagrange. Dans la suite, nous détaillons deux familles de fonctionnelles pour mesurer la régularité. Ces familles sont respectivement obtenues avec $p = 2$ et $q = p$.

4.3.1 Fonctionnelles de régularisation discrète

La première famille de fonctionnelles est basée sur la norme $\mathcal{L}_2$ du gradient local (équation (4.10)). Elle peut être vue comme une extension des fonctionnelles discrètes proposées par [Osher et Shen, 2000, Chan *et al.*, 2001], avec une définition plus générale de la régularité. Formellement, la **p-forme de Dirichlet** d'une fonction $f : V \to \mathbb{R}$, ou **fonctionnelle p-TV isotrope**, est définie par l'expression :

$$R_w^{TVi}(f, p) := \tfrac{1}{2p} \sum_{u \in V} \| \nabla_w f(u) \|_2^p, \quad p \in]0, +\infty[\tag{4.34}$$

$$\overset{(4.10)}{=} \tfrac{1}{2p} \sum_{u \in V} \left(\sum_{v \sim u} w_{uv} (f(v) - f(u))^2 \right)^{\frac{p}{2}}. \tag{4.35}$$

La seconde famille proposée est basée sur la norme $\mathcal{L}_p$ du gradient local (équation (4.9)). La **fonctionnelle p-TV anisotrope** d'une fonction $f \in \mathcal{H}(V)$ est définie par :

$$R_w^{TVa}(f, p) := \tfrac{1}{2p} \sum_{u \in V} \| \nabla_w f(u) \|_p^p, \quad p \in]0, +\infty[\tag{4.36}$$

$$\overset{(4.9)}{=} \tfrac{1}{2p} \sum_{u \in V} \sum_{v \sim u} w_{uv}^{\frac{p}{2}} |f(v) - f(u)|^p. \tag{4.37}$$

Remarque. *Les fonctionnelles p-TV isotropes et anisotropes sont les extensions discrètes pondérées des fonctionnelles basées sur la variation totale isotrope et anisotrope d'une*

fonction continue $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}$, *et définies respectivement par :*

$$J^{TVi}(f,p) := \tfrac{1}{p}\int_\Omega \| \bigtriangledown f\|_2^p dx,$$

et

$$J^{TVa}(f,p) := \tfrac{1}{2p}\int_\Omega \| \bigtriangledown f\|_p^p dx.$$

Pour $p = 2$, *les fonctionnelles isotropes et anisotropes sont équivalentes, aussi bien dans le cadre discret que dans le cadre continu.*

4.3.2 Expressions non-locales

D'après la définition de la fonction de poids w, les variations locales peuvent s'exprimer sous forme non-locale (voir l'équation (4.12) à la section 4.1.3). Les expressions non-locales des fonctionnelles p-TV isotrope et anisotrope sont alors respectivement définies par :

$$\begin{aligned} R_w^{NLTVi}(f,p) &:= \tfrac{1}{2p}\sum_{u\in V} \| \bigtriangledown_w^{NL} f(u)\|_2^p \\ &\overset{(4.12)}{=} \tfrac{1}{2p}\sum_{u\in V}\left(\sum_{v\in V} w(u,v)(f(v)-f(u))^2\right)^{\frac{p}{2}} = R_w^{TVi}(f,p), \end{aligned} \tag{4.38}$$

et par :

$$\begin{aligned} R_w^{NLTVa}(f,p) &:= \tfrac{1}{2p}\sum_{u\in V} \| \bigtriangledown_w^{NL} f(u)\|_p^p \\ &\overset{(4.12)}{=} \tfrac{1}{2p}\sum_{u\in V}\sum_{v\in V} w(u,v)^{\frac{p}{2}}|f(v)-f(u)|^p = R_w^{TVa}(f,p). \end{aligned} \tag{4.39}$$

Remarque. *Les fonctionnelles non-locales* $R_w^{NLTVi}(f,p)$ *et* $R_w^{NLTVa}(f,p)$ *sont des extensions discrètes des fonctionnelles non-locales proposées récemment dans le contexte du traitement d'images par [Kinderman* et al.*, 2005, Gilboa et Osher, 2007a, Gilboa et Osher, 2007b]. La fonctionnelle isotrope non-locale définie pour des fonctions* $f : \Omega \subset \mathbb{R}^n \to \mathbb{R}$, *et analogue à* $R_w^{NLTVi}(f,p)$, *s'écrit :*

$$J_w^{NLTVi}(f,p) := \tfrac{1}{p}\int_\Omega \left(\int_\Omega w(x,y)(f(y)-f(x))^2 dy\right)^{\frac{p}{2}} dx.$$

En particulier, pour $p = 2$ *elle correspond à :*

$$J_w^{NLTVi}(f,2) = \tfrac{1}{2}\int_{\Omega\times\Omega} w(x,y)(f(y)-f(x))^2 dydx,$$

qui est une interprétation variationnelle continue d'une famille de filtres de voisinage [Kinderman et al.*, 2005, Gilboa et Osher, 2007a]. Pour* $p = 1$, J_w^{NLTVi} *est la variation totale*

non-locale [Gilboa et Osher, 2007b] :

$$J_w^{NLTVi}(f,1) = \int_\Omega \left(\int_\Omega w(x,y)(f(y)-f(x))^2 dy \right)^{\frac{1}{2}}.$$

De même, l'analogue continu de la fonctionnelle anisotrope non-locale $R_w^{NLTVa}(f,p)$ s'écrit :

$$J_w^{NLTVa}(f,p) := \tfrac{1}{2p} \int_{\Omega\times\Omega} w(x,y)^{\frac{p}{2}} |f(y)-f(x)|^p dydx, \qquad f : \Omega \subset \mathbb{R}^m \to \mathbb{R}.$$

Pour $p = 1$, elle correspond à la fonctionnelle anisotrope non-locale basée sur les différences [Gilboa et Osher, 2007b] :

$$J_w^{NLTVa}(f,1) = \tfrac{1}{2} \int_{\Omega\times\Omega} \sqrt{w(x,y)} |f(y)-f(x)| dydx.$$

Comme dans le cas local, les fonctionnelles non-locales isotrope et anisotrope coïncident pour $p = 2$, que ce soit dans le cadre discret ou dans le cadre continu.

Les fonctionnelles non-locales discrètes sont conceptuelles. Elles prennent tout leurs sens dans le contexte de la régularisation de données discrètes, ou de fonctions définies sur ces données. C'est la structure du graphe, le choix de la fonction de poids, ainsi que la fonction traitée qui rend la régularisation non-locale (voir le chapitre 5).

4.3.3 Modèles variationnels p-TV isotrope et p-TV anisotrope

Modèle isotrope. Basé sur la fonctionnelle p-TV isotrope, $R_w^{TVi}(f,p)$, le problème de minimisation (4.33) se réécrit :

$$\min_{f\in\mathcal{H}(V)} \left\{ E_w^{TVi}(f,f^0,\lambda,p) := \tfrac{1}{2p} \sum_{u\in V} \| \nabla_w f(u)\|_2^p + \tfrac{\lambda}{2} \sum_{u\in V} (f(u)-f^0(u))^2 \right\}, \tag{4.40}$$

pour $p \in]0,+\infty[$. Ce problème est la version discrète sur graphes pondérés de la régularisation de Tikhonov [Tikhonov et Arsenin, 1977] pour $p = 2$, et de la régularisation basée sur le modèle ROF [Rudin *et al.*, 1992] pour $p = 1$. Pour $p \geq 1$, $E_w^{TVi}(f,f^0,\lambda,p)$ est une fonctionnelle convexe. Comme $\lim_{f\to\infty} E_w^{TVi}(f,f^0,\lambda,p) = \infty$, par des arguments standards en analyse convexe, il existe un minimum global à la minimisation (4.40). Ainsi, tout minimum local est un minimum global. Soit $\mathcal{C}(V)$ l'ensemble des fonctions constantes $f : V \to \mathbb{R}$, telles que $f(u) = c$ pour tout $u \in V$, avec $c \in \mathbb{R}$. Pour les fonctions de $\mathcal{H}(V) \setminus \mathcal{C}(V)$, la fonctionnelle $R_w^{TVi}(f,p)$ est strictement convexe, et la minimisation à une solution unique. C'est également le cas des fonctions de $\mathcal{C}(V)$. En effet, $R_w^{TVi}(f,p)$ est invariante par translation, c.-à-d. $R_w^{TVi}(f+c,p) = R_w^{TVi}(f,p)$ pour toute constante $c \in \mathbb{R}$. Ainsi il existe un unique minimum global à la minimisation (4.40), pour $p \geq 1$. Lorsque $p < 1$, $R_w^{TVi}(f,p)$ est non-convexe, et la minimisation globale peut ne pas exister. Néanmoins, cela ne veut pas dire que les processus de diffusion associés ne sont

pas intéressants (voir la section 4.4.1 et le chapitre 5).

En supposant qu'il existe une solution unique pour toute valeur de $p \in]0, +\infty[$, elle peut être obtenue en calculant le passage par zéro de la dérivée partielle de $E_w^{TVi}(f, f^0, \lambda, p)$ par rapport à f en un sommet $u \in V$:

$$\frac{\partial E_w^{TVi}(f, f^0, \lambda, p)}{\partial f(u)} = \frac{\partial R_w^{TVi}(f, p)}{\partial f(u)} + \lambda(f(u) - f^0(u)) = 0, \quad \forall u \in V. \tag{4.41}$$

La dérivée de la fonctionnelle p-TV isotrope correspond exactement au p-Laplacien discret défini à la section 4.2.1.

Propriété 19 $\frac{\partial R_w^{TVi}(f,p)}{\partial f(u)} = \Delta_w^p f(u), \quad \forall u \in V, \ \forall f \in \mathcal{H}(V).$

Preuve. La dérivée partielle de $R_w^{p\text{-}TV}(f)$, par rapport à f, s'écrit :

$$\frac{\partial R_w^{TVi}(f,p)}{\partial f} = \frac{\partial}{\partial f}\left(\frac{1}{2p}\sum_{u\in V} \| \nabla_w f(u)\|_2^p\right) \overset{(4.10)}{=} \frac{1}{2p}\frac{\partial}{\partial f}\left(\sum_{u\in V}\left(\sum_{v\sim u} w_{uv}\left(f(v) - f(u)\right)^2\right)^{\frac{p}{2}}\right).$$

En se plaçant en un sommet $u_1 \in V$, la dérivée ne fait alors qu'intervenir les arêtes incidentes à ce sommet :

$$\begin{aligned}
2\frac{\partial R_w^{TVi}(f,p)}{\partial f(u_1)} = &- \sum_{v\sim u_1} w_{u_1v}\left(f(v) - f(u_1)\right)\left(\sum_{v\sim u_1} w_{u_1v}\left(f(v) - f(u_1)\right)^2\right)^{\frac{p-2}{2}} \\
&+ w_{u_1v_1}\left(f(u_1) - f(v_1)\right)\left(\sum_{v\sim v_1} w_{v_1v}\left(f(v) - f(v_1)\right)^2\right)^{\frac{p-2}{2}} \\
&+ \ldots + w_{u_1v_k}\left(f(u_1) - f(v_k)\right)\left(\sum_{v\sim v_k} w_{v_kv}\left(f(v) - f(v_k)\right)^2\right)^{\frac{p-2}{2}} \\
\overset{(4.10)}{=} &\sum_{v\sim u_1} w_{u_1v}\left(f(u_1) - f(v)\right) \| \nabla_w f(u_1)\|_2^{p-2} \\
&+ \sum_{v\sim u_1} w_{u_1v}\left(f(u_1) - f(v)\right) \| \nabla_w f(v)\|_2^{p-2} = 2\Delta_w^p f(u_1). \ \square
\end{aligned}$$

À partir de la Propriété 19, la solution du problème de minimisation (4.40) est la solution du système d'équations suivant :

$$\Delta_w^p f(u) + \lambda(f(u) - f^0(u)) = 0, \quad \forall u \in V. \tag{4.42}$$

Cette dernière équation peut être vue comme l'analogue discret de l'équation d'Euler-Lagrange associée à la fonctionnelle isotrope E_w^{TVi}. Contrairement au cas continu, elle ne fait pas intervenir d'EDP, et elle est indépendante de la structure du graphe. En substi-

tuant l'expression (4.23) du p-Laplacien dans le système (4.42), on obtient directement le système d'équations suivant :

$$\left(\lambda + \sum_{v\sim u} \gamma_w^f(u,v)\right) f(u) - \sum_{v\sim u} \gamma_w^f(u,v) f(v) = \lambda f^0(u), \quad \forall u \in V, \tag{4.43}$$

où

$$\gamma_w^f(u,v) := \tfrac{1}{2} w_{uv} \left(\| \nabla_w f(v) \|_2^{p-2} + \| \nabla_w f(u) \|_2^{p-2} \right). \tag{4.44}$$

Comme γ_w^f dépend de la variation locale de la fonction f, le système obtenu est non-linéaire, excepté dans le cas où $p = 2$. Dans ce dernier cas, le système fait intervenir le Laplacien combinatoire :

$$\begin{aligned} \Delta_w f(u) + \lambda(f(u) - f^0(u)) &= 0, \quad \forall u \in V, \\ \Leftrightarrow (\lambda + \delta_w(u)) f(u) - \sum_{v\sim u} w_{uv} f(v) &= \lambda f^0(u), \quad \forall u \in V. \end{aligned}$$

Il peut être résolu par plusieurs méthodes numériques qui convergent efficacement vers la solution du problème de minimisation. Dans le cadre continu, une méthode classique pour résoudre la minimisation du modèle ROF (cas isotrope avec $p = 1$) est l'algorithme de projection de Chambolle [Chambolle, 2004], basé sur la méthode des points fixes. Cet algorithme efficace peut être aisément adapté au cas discret sur graphes. Nous proposons d'utiliser des processus de diffusion semi-discrète ou discrète (voir la section 4.4).

Modèle anisotrope. Similairement à la régularisation isotrope, nous décrivons maintenant la régularisation anisotrope basée sur la fonctionnelle R_w^{TVa}. Le problème de minimisation (4.33) se réécrit alors :

$$\min_{f\in\mathcal{H}(V)} \left\{ E_w^{p\text{-}TVa}(f, f^0, \lambda) := \tfrac{1}{2p} \sum_{u\in V} \| \nabla_w f(u) \|_p^p + \tfrac{\lambda}{2} \sum_{u\in V} (f(u) - f^0(u))^2 \right\}, \tag{4.45}$$

avec $p \in]0, +\infty[$. La solution du problème de minimisation (4.45) peut être obtenue en résolvant le système d'équations :

$$\frac{\partial E_w^{TVa}(f, f^0, \lambda, p)}{\partial f(u)} = \frac{\partial R_w^{TVa}(f,p)}{\partial f(u)} + \lambda(f(u) - f^0(u)) = 0, \quad \forall u \in V. \tag{4.46}$$

Comme dans le cas du modèle isotrope, la dérivée de la fonctionnelle $R_w^{TVa}(f,p)$, par rapport à f en chaque sommet du graphe, correspond exactement à l'expression du p-Laplacien anisotrope défini à la section 4.2.2.

Propriété 20 *Pour tout $u \in V$, et toute fonction $f \in \mathcal{H}(V)$, nous avons :*

$$\frac{\partial R_w^{TVa}(f,p)}{\partial f(u)} = \widetilde{\Delta_w^p} f(u). \tag{4.47}$$

Preuve. La dérivée partielle de $R_w^{TVa}(f,p)$, par rapport à f, s'écrit :

$$\frac{\partial R_w^{TVa}(f,p)}{\partial f} \overset{(4.34)}{=} \frac{\partial}{\partial f}\left(\frac{1}{2p}\sum_{u\in V}\| \nabla_w f(u)\|_p^p\right) \overset{(4.10)}{=} \frac{\partial}{\partial f}\left(\frac{1}{2p}\sum_{u\in V}\sum_{v\sim u} w_{uv}^{\frac{p}{2}}|f(v)-f(u)|^p\right).$$

En se plaçant en un sommet $u_1 \in V$, la dérivée ne fait alors qu'intervenir les arêtes incidentes à ce sommet :

$$\begin{aligned}
2\frac{\partial R_w^{TVa}(f,p)}{\partial f(u_1)} &= -\sum_{v\sim u_1} w_{u_1v}^{\frac{p}{2}}(f(v)-f(u_1))|f(v)-f(u_1)|^{p-2} \\
&\quad + w_{u_1v_1}^{\frac{p}{2}}(f(u_1)-f(v_1))|f(u_1)-f(v_1)|^{p-2} \\
&\quad + \ldots + w_{u_1v_k}^{\frac{p}{2}}(f(u_1)-f(v_k))|f(u_1)-f(v_k)|^{p-2} \\
&= 2\sum_{v\sim u_1} w_{u_1v}^{\frac{p}{2}}(f(u_1)-f(v))|f(u_1-f(v)|^{p-2} = 2\widetilde{\Delta_w^p} f(u_1). \ \square
\end{aligned}$$

À partir de la Propriété 20 la solution devient :

$$\widetilde{\Delta_w^p} f(u) + \lambda(f(u)-f^0(u)) = 0, \quad \forall u \in V. \tag{4.48}$$

Cette dernière équation peut être considérée comme l'équation d'Euler-Lagrange associée au problème de minimisation (4.45). Elle est basée sur le p-Laplacien anisotrope, et pour $p = 2$ nous retrouvons le cas linéaire basé sur le Laplacien combinatoire. En remplaçant l'expression du p-Laplacien anisotrope dans l'équation (4.48), nous obtenons le système d'équations :

$$\left(\lambda + \sum_{v\sim u}\gamma_w^f(u,v)\right) f(u) - \sum_{v\sim u}\gamma_w^f(u,v) f(v) = \lambda f^0(u), \quad \forall u \in V, \tag{4.49}$$

où

$$\gamma_w^f(u,v) = w_{uv}^{\frac{p}{2}}|f(u)-f(v)|^{p-2}.$$

Un cas particulier intéressant est donné pour $p = 1$. Dans ce cas, le système peut se réécrire en utilisant la fonction signe (deuxième expression de l'opérateur p-Laplacien anisotrope) :

$$\lambda f(u) + \sum_{v\sim u}\sqrt{w_{uv}}\, sign(f(u)-f(v)) = \lambda f^0(u), \quad \forall u \in V. \tag{4.50}$$

Habituellement, la solution de la minimisation des fonctionnelles anisotropes s'effectue en utilisant des algorithmes basés sur des coupures de graphes [Boykov *et al.*, 2001, Darbon et Sigelle, 2004]. Dans la section suivante, nous proposons d'utiliser deux méthodes de diffusion pour résoudre les systèmes d'équations issus de la minimisation des fonctionnelles de régularisation isotrope et anisotrope.

4.4 Processus de diffusion comme solutions de la régularisation

Les problèmes de minimisation, basés sur la régularisation isotrope et anisotrope sur graphes, reviennent à résoudre des systèmes linéaires et non-linéaires d'équations. Comme dans le cadre continu, les solutions de ces systèmes peuvent se formuler comme des processus de diffusion. Ces mêmes processus peuvent également être utilisés indépendamment de la régularisation. L'avantage du cadre discret est qu'ils ne font pas intervenir d'EDP. Dans cette section, nous proposons d'utiliser deux méthodes classiques pour approximer la solution des systèmes d'équations. D'une part, la descente de marche infinitésimale mène à des processus de diffusion semi-discrète. D'autre part, la méthode linéarisée de Gauss-Jacobi mène à des processus de diffusion discrète.

Dans la suite, nous unifions la régularisation isotrope et la régularisation anisotrope en utilisant les notations adoptées pour décrire les familles de p-Laplaciens (voir la section 4.2.3). Les différentes équations qui interviennent dans les régularisations considérées sont rappelées au tableau 4.2.

équations	*modèle p-TV isotrope*	*modèle p-TV anisotrope*
énergie	$\frac{1}{p}\sum_{u\in V} \parallel \bigtriangledown_w f(u)\parallel_2^p + \frac{\lambda}{2}\|f-f^0\|_2^2$	$\frac{1}{2p}\sum_{u\in V} \parallel \bigtriangledown_w f(u)\parallel_p^p + \frac{\lambda}{2}\|f-f^0\|_2^2$
système	$\Delta_w^p f(u) + \lambda(f(u)-f^0(u)=0,\ \forall u\in V$	$\widetilde{\Delta_w^p} f(u) + \lambda(f(u)-f^0(u)=0,\ \forall u\in V$
p-Laplacien L_w^p	$\Delta_w^p f(u) = \sum_{v\sim u}\gamma_w^f(u,v)(f(u)-f(v))$	$\widetilde{\Delta_w^p} f(u) = \sum_{v\sim u}\gamma_w^f(u,v)(f(u)-f(v))$
$\gamma_w^f(u,v)$	$\frac{1}{2}w_{uv}\left(\parallel \bigtriangledown_w f(v)\parallel_2^{p-2} + \parallel \bigtriangledown_w f(u)\parallel_2^{p-2}\right)$	$w_{uv}^{\frac{p}{2}}\|f(u)-f(v)\|^{p-2}$

Tab. 4.2: Équations intervenant dans la régularisation d'une fonction $f^0 : V \to \mathbb{R}$, définie sur les sommets d'un graphe pondéré $G := (V, E, w)$.

4.4.1 Diffusion semi-discrète

Les systèmes d'équation (4.42) et (4.48) (voir la troisième ligne du tableau 4.2) peuvent s'exprimer avec le système suivant :

$$L_w^p f(u) + \lambda(f(u) - f^0(u)) = 0, \quad \forall u \in V, \quad f, f^0 : V \to \mathbb{R}, \tag{4.51}$$

où $L_w^p = \Delta_w^p$ dans le cas de la régularisation isotrope, et $L_w^p = \widetilde{\Delta_w^p}$ dans le cas de la régularisation anisotrope. Pour approximer sa solution, nous pouvons considérer la méthode de descente de marche infinitésimale :

$$\begin{cases} f^{(0)} = f^0, \\ \dfrac{df^{(t)}(u)}{dt} = -\,L_w^p f(u) + \lambda\left(f^0(u) - f(u)\right), \quad \forall u \in V, \end{cases} \tag{4.52}$$

où $f^{(t)}$ est la paramétrisation de la fonction f par un temps artificiel t. C'est un système d'équations différentielles ordinaires (ODE), qui décrit des processus de diffusion. Contrairement à la diffusion continue, qui fait intervenir des EDP, aucune discrétisation spatiale est nécessaire sur les graphes. La solution du système d'ODE peut être établie en utilisant différentes méthodes. Nous proposons d'utiliser les méthodes itératives, comme celle d'Euler ou de Runge-Kutta, qui sont considérées dans de nombreux problèmes de diffusion pour leur qualité d'approximation et leur simplicité.

L'Algorithme 8 décrit la méthode itérative de Runge-Kutta d'ordre 1 pour résoudre le problème de diffusion (4.52). Une itération de la méthode est donnée par l'expression suivante :

$$f^{(t+1)}(u) = f^{(t)}(u) + \tau \left(-L_w^p f^{(t)}(u) + \lambda \left(f^0(u) - f^{(t)}(u)\right)\right), \quad \forall u \in V, \tag{4.53}$$

où $\tau > 0$ est la taille de la marche infinitésimale. Dû à l'introduction de ce dernier paramètre, le processus de diffusion est semi-discret. La diffusion de la fonction f^0 à l'instant t est la fonction $f^{(t)}$, initialisée en $t = 0$ à f^0. Le nombre d'itérations dépend généralement du contexte, qui est discuté à la section 4.4.5. Néanmoins, pour satisfaire le système (4.51), le processus nécessite d'être itéré jusqu'à ce qu'il converge, c.-à-d. $f^{(t+1)} = f^{(t)}$ (s'il converge). Remarquons que les lignes 5, 6 et 7 de l'Algorithme 8 calculent le p-Laplacien de la fonction $f^{(t)}$. Selon l'expression de la fonction γ_w^f (isotrope ou anisotrope), et le choix du paramètre p de régularité, le calcul de γ_w^f peut s'effectuer en plusieurs étapes. En particulier, la diffusion isotrope non-linéaire ($p \neq 2$) fait intervenir la variation locale de la fonction f en chaque sommet du graphe, et à chaque itération. Ces variations locales, qui apparaissent dans l'expression de γ_w^f, sont alors calculées et sauvegardées avant chaque étape (4.53) (entre la ligne 3 et la ligne 4).

Algorithm 8 DiffusionSemiDiscrete($G, f^0, \lambda, p, \tau, t$)

Entrée: un graphe pondéré $G = (V, E, w)$, une fonction $f^0 : V \to \mathbb{R}$, le niveau λ de fidélité, le degré p de régularité, la taille τ de la marche infinitésimale, et le nombre d'itérations t.

Sortie: la fonction régularisée $f : V \to \mathbb{R}$.

1: $f, g : V \to \mathbb{R}$
2: $f \leftarrow f^0$
3: **tant que** $t > 0$ (ou une condition d'arrêt non-satisfaite) **faire**
4: **pour chaque** sommet $u \in V$ **faire**
5: $s \leftarrow 0$
6: **pour chaque** arête $(u, v) \in E$ incidente à u **faire**
7: $s \leftarrow s + \gamma_w^f(u, v) * (f(v) - f(u))$
8: $g(u) \leftarrow f(u) + \tau * (s + \lambda * (f^0(u) - f(u)))$
9: $f \leftarrow g$
10: $t \leftarrow t - 1$
11: **retourner** f

Bien que cet algorithme fournisse une bonne régularisation au cours des itérations, si la valeur de τ est suffisamment faible, le choix de cette dernière peut s'avérer délicate. Il est donc préférable de l'utiliser en tant que processus de diffusion, indépendamment de la résolution des problèmes de minimisation basés sur la régularisation, en fixant le nombre d'itérations. Dans la suite, nous décrivons une autre méthode, à notre sens mieux adaptée à fournir une approximation de la solution des problèmes de minimisation.

4.4.2 Diffusion discrète

Une autre façon de résoudre les problèmes de régularisation est de considérer les systèmes d'équations (4.43) et (4.49), donnés par l'expression générale suivante :

$$\left(\lambda + \sum_{v \sim u} \gamma_w^f(u,v)\right) f(u) - \sum_{v \sim u} \gamma_w^f(u,v) f(v) = \lambda f^0(u), \ \ \forall u \in V, \tag{4.54}$$

et d'utiliser la méthode itérative linéarisée de Gauss-Jacobi, dont une itération s'exprime par :

$$f^{(t+1)}(u) = \frac{\lambda f^0(u) + \sum_{v \sim u} \gamma_w^{f^{(t)}}(u,v) f^{(t)}(v)}{\lambda + \sum_{v \sim u} \gamma_w^{f^{(t)}}(u,v)}, \qquad \forall u \in V. \tag{4.55}$$

Itérer cette dernière équation à partir de zéro (avec $f^{(0)} = f^0$), jusqu'à la convergence (si elle existe), permet d'approximer la solution du système (4.54). Contrairement à la diffusion semi-discrète, cette approche ne fait pas intervenir de paramètre d'approximation. Elle décrit des processus de diffusion discrète. La méthode complète est résumée par l'Algorithme 9. Comme dans le cas de la diffusion semi-discrète isotrope non-linéaire, le calcul des coefficients γ_w^f peut être effectué avant d'appliquer l'équation (4.55).

Algorithm 9 DiffusionDiscrete(G, f^0, λ, p, t)

Entrée: un graphe pondéré $G = (V, E, w)$, une fonction $f^0 : V \to \mathbb{R}$, le niveau λ de fidélité, le degré p de régularité, et le nombre d'itérations t.

Sortie: la fonction régularisée $f : V \to \mathbb{R}$.

```
f, g : V → ℝ
f ← f^0
tant que t > 0 (ou une condition d'arrêt non-satisfaite) faire
  pour chaque sommet u ∈ V faire
    s, d ← 0
    pour chaque arête (u, v) ∈ E incidente à u faire
      s ← s + γ_w^f(u, v) * f(v)
      d ← d + γ_w^f(u, v)
    g(u) ← (λ f^0(u) + s)/(λ + d)
  f ← g
  t ← t − 1
retourner f
```

Analyse de la méthode. À partir de l'itération (4.55), nous pouvons observer que la nouvelle valeur $f^{(t+1)}(u)$ dépend de deux quantités : la valeur initiale $f^0(u)$ (si $\lambda \neq 0$), et une moyenne pondérée des valeurs de $f^{(t)}$ dans le voisinage du sommet u. La diffusion discrète se comporte alors comme un filtre moyenneur de voisinage, qui s'adapte aux valeurs filtrées au cours des itérations. Pour voir cela, nous considérons la fonction φ_w^f, définie par l'expression :

$$\varphi_w^f(u,v) = \begin{cases} \dfrac{\gamma_w^f(u,v)}{\lambda + \sum_{k\sim u} \gamma_w^f(u,k)} & \text{si } u \neq v, \\ \dfrac{\lambda}{\lambda + \sum_{k\sim u} \gamma_w^f(u,k)} & \text{si } u = v, \end{cases} \tag{4.56}$$

qui satisfait les propriétés suivantes (dû à celles de γ_w^f) :

$$\begin{cases} \varphi_w^f(u,u) + \displaystyle\sum_{v\sim u} \varphi_w^f(u,v) = 1, \\ \varphi_w^f(u,u) \geq 0, & \forall u \in V. \\ \displaystyle\sum_{v\sim u} \varphi_w^f(u,v) \geq 0, \end{cases} \tag{4.57}$$

Avec ces notations, le processus de diffusion discrète se réécrit sous la forme suivante :

$$\begin{cases} f^{(0)} = f^0 \\ f^{(t+1)}(u) = \varphi_w^{f^{(t)}}(u,u) f^0(u) + \displaystyle\sum_{v\sim u} \varphi_w^{f^{(t)}}(u,v) f^{(t)}(v), \quad \forall u \in V. \end{cases} \tag{4.58}$$

Ceci permet de montrer que le processus de diffusion discrète vérifie le principe du minimum – maximum.

Propriété 21 *Le processus de diffusion (4.58) (et l'Algorithme 9) satisfait :*

$$\min_{v\in V} f^0(v) \leq f^{(t)}(u) \leq \max_{v\in V} f^0(v), \qquad \forall t > 0,\ \forall u \in V.$$

Preuve. En considérant (4.58), à partir des propriétés (4.57), nous pouvons déduire que $f^{(t+1)}(u)$ est une combinaison convexe de $f^0(u)$ et de $f^{(t)}(v)$ pour tout $t > 0$. Donc, nous avons :

$$\min\{f^0(u), \min_{v\sim u} f^{(t)}(v)\} \leq f^{(t+1)}(u) \leq \max\{f^0(u), \max_{v\sim u} f^{(t)}(v)\}, \quad \forall u \in V.$$

La propriété (i) est alors obtenue par récursivité sur la valeur de t. □

À partir de ces propriétés, nous montrons dans la section suivante que selon le type de régularité (isotrope ou anisotrope), le degré de régularité p, ainsi que le niveau λ d'approximation, la diffusion discrète est au filtrage de fonctions dans le domaine spectral. De plus, en fonction de la nature de la fonction à traiter et de la structure du graphe

(topologie et pondération), nous retrouvons l'expression exacte, ou la version discrétisée, de plusieurs processus de diffusion et de filtres de voisinage (voir le chapitre 5).

4.4.3 Relations entre la diffusion discrète et le filtrage spectral

Le processus de diffusion discrète (4.58), décrit dans la section précédente, est issu de la minimisation des modèles variationnels isotrope et anisotrope. Lorsque $\lambda = 0$, il correspond à un processus de diffusion pure, qui peut se réécrire sous forme non-locale (dû à la définition de la fonction de poids w) :

$$\begin{cases} f^{(0)} = f^0, \\ f^{(t+1)}(u) = \sum_{v \in V} \varphi_w^{f^{(t)}}(u,v) f^{(t)}(v), \quad \forall u \in V. \end{cases} \tag{4.59}$$

Comme $\varphi_w^{f^{(t)}}(u,v) \geq 0$ et $\sum_{v \in V} \varphi_w^f(u,v) = 1$, la fonction $\varphi_w^{f^{(t)}}$ peut s'interpréter comme la probabilité de passer du sommet u au sommet v en une seule itération. Pour cela, nous considérons la matrice $K \in \mathbb{R}^{N \times N}$ définie par

$$K(f)(u,v) := \begin{cases} \varphi_w^f(u,v) & \text{si } u \sim v, \\ 0 & \text{sinon.} \end{cases}$$

En considérant les formes vectorielles $\mathbf{f}$ et $\mathbf{f}^0$ des fonctions $f, f^0 : V \to \mathbb{R}$, le processus de diffusion discrète (4.59) se réécrit de la façon suivante :

$$\begin{cases} \mathbf{f}^{(0)} = \mathbf{f}^0 \\ \mathbf{f}^{(t+1)} = K(\mathbf{f}^{(t)}) \mathbf{f}^{(t)}. \end{cases} \tag{4.60}$$

En particulier, comme K est une matrice non-négative satisfaisant $\sum_{v \in V} K(u,v) = 1$ pour tout $u \in V$ et $K(u,v) = K(v,u)$, K est une matrice stochastique. Ceci implique que le processus de diffusion (4.60) est lié à la théorie des chaînes de Markov, voir par exemple [Seneta, 1981]. Cette interprétation est exactement la diffusion sur graphes basée sur des marches aléatoires de Markov (pour $p = 2$), utilisée pour étudier la géométrie d'ensembles de données discrètes, ainsi que de fonctions définies sur ces données [Chung, 1997, Coifman *et al.*, 2006, Szlam *et al.*, 2006]. Un élément $K(u,v)$, décrit la probabilité de transition de u à v en une itération. La matrice K code les similarités locales entre les sommets du graphe. À chaque itération, elle peut se décomposer comme suit :

$$K\phi_i = a_i \phi_i, \tag{4.61}$$

où $1 \geq a_1 \geq \ldots \geq a_N \geq 0$ sont les valeurs propres de K, et $\{\phi_i\}$ ses vecteurs propres. Les vecteurs propres associés aux k premières valeurs propres contiennent l'information principale. Ainsi, une façon équivalente d'exprimer K dans l'équation (4.64), est de décomposer chaque valeur de $\mathbf{f}$ sur les k premiers vecteurs propres de K. De plus, les vecteurs propres de la matrice K peuvent être considérés comme une extension des fonctions de la transformée de Fourier, avec les a_i^{-1} représentant les fréquences des ϕ_i. À

chaque itération du processus de diffusion, cela définit une base de la fonction $f \in \mathcal{H}(V)$, qui se décompose sur les k premiers vecteurs propres de K par :

$$f^{(t)} \approx \sum_{i=1}^{i=k} \langle f^{(t)}, \phi_i \rangle \phi_i. \tag{4.62}$$

Cela peut s'interpréter comme un processus de filtrage dans le domaine spectral, qui a des propriétés intéressantes lorsque $p = 2$.

4.4.4 Cas particulier : diffusion discrète linéaire

Lorsque $p = 2$, les modèles variationnels isotrope et anisotrope coïncident, et correspondent à la régularisation de Tikhonov sur graphes pondérés. Dans ce cas, le processus de diffusion discrète (4.58), fait intervenir le Laplacien combinatoire et se réécrit :

$$\begin{cases} f^{(0)} = & f^0 \\ f^{(t+1)}(u) = & \dfrac{1}{\lambda + \sum_{v \sim u} w_{uv}} \left(\lambda f^0(u) + \displaystyle\sum_{v \sim u} w_{uv} f^{(t)}(v) \right), \quad \forall u \in V. \end{cases} \tag{4.63}$$

Comme le Laplacien est un opérateur linéaire, la diffusion est également linéaire. En considérant le cas particulier de $\lambda = 0$, et la forme matricielle de ce processus, c.-à-d. (4.60), il se réécrit :

$$\begin{cases} \mathbf{f}^{(0)} = & \mathbf{f}^0 \\ \mathbf{f}^{(t+1)} = & K\mathbf{f}^{(t)} = K^2\mathbf{f}^{(t-1)} = \ldots = K^t\mathbf{f}^0. \end{cases} \tag{4.64}$$

Alors, la décomposition spectrale (4.61), de la matrice K, s'exprime au bout de t itérations comme :

$$K^t \phi_i = \lambda_i^t \phi_i,$$

L'avantage est de calculer la diffusion avec la matrice K à la puissance t en une seule itération, ce qui accélère considérablement les traitements.

4.4.5 Critères d'arrêt des processus de diffusion

Dans les sections précédentes, nous avons proposé plusieurs processus itératifs de diffusion pour approximer la solution des problèmes de minimisation (4.40) et (4.45). La solution de ces méthodes est théoriquement atteinte lorsque la fonction $f^{(t+1)}$ est égale à $f^{(t)}$. Une façon simple de déterminer la convergence est de tester à chaque itération comment la fonction varie par rapport à l'itération précédente :

$$\|f^{(t+1)} - f^{(t)}\|_2 < \epsilon \|f^{(t+1)}\|_2, \qquad \epsilon \to 0.$$

En fonction de l'application, les processus de diffusion peuvent aussi être utilisés indépendemment des problèmes de minimisation. C'est particulièrement le cas pour $\lambda = 0$

(pas de terme d'approximation). Le nombre d'itérations est alors passé explicitement aux algorithmes. Ce sont ces deux critères d'arrêt qui sont utilisés dans les applications décrites au chapitre 5.

4.5 Cas des fonctions vectorielles

Jusqu'à présent, nous avons considéré le cas de fonctions scalaires définies sur les sommets d'un graphe. Lorsque la fonction à traiter est vectorielle, c.-à-d. une fonction $f^0 : V \to \mathbb{R}^m$, avec $f^0(u) = (f_1^0(u), \ldots, f_m^0(u))$ et $f_i^0 : V \to \mathbb{R}$, la régularisation est effectuée sur chaque composante f_i^0. Il y a alors m processus de diffusion à calculer pour régulariser une fonction vectorielle, un par composante, quelque soit la diffusion choisie (semi-discrète ou discrète). La diffusion semi-discrète devient :

$$\begin{cases} f^{(0)} = f^0, \\ f_i^{(t)}(u) = f_i^{(t)}(u) + \tau \left(-L_w^P f_i^{(t)}(u) + \lambda(f_i^0(u) - f_i^{(t)}(u)) \right), \quad \forall i = 1, \ldots, m, \ \forall u \in V. \end{cases}$$

Et la diffusion discrète devient :

$$\begin{cases} f^{(0)} = f^0 \\ f_i^{(t+1)}(u) = \dfrac{\lambda f_i^0(u) + \sum_{v \sim u} \gamma_w^{f^{(t)}}(u,v) f_i^{(t)}(v)}{\lambda + \sum_{v \sim u} \gamma_w^{f^{(t)}}(u,v)}, \quad \forall i = 1, \ldots, m, \ \forall u \in V. \end{cases}$$

Avec ces notations, la fonction γ_w^f, qui intervient dans l'expression des p-Laplaciens, dépend seulement de la i^{ème} composante de la fonction f. Pour surmonter cette limitation, il est usuel de considérer la dépendance entre les composantes. Ceci peut s'effectuer de plusieurs façons, en fonction du type de régularité considéré (isotrope ou anisotrope), et en fonction du degré de régularité p.

Cas de la régularisation linéaire ($p = 2$). Dans ce cas, nous avons vu que la régularisation mène à des processus de diffusion qui font intervenir le Laplacien combinatoire. Ce dernier à exactement la même expression pour les m composantes de la fonction traitée :

$$\Delta_w f_i(u) = \sum_{v \sim u} w_{uv}(f_i(u) - f_i(v)), \quad \forall i = 1, \ldots, m, \ \forall u \in V.$$

La dépendance entre les composantes peut être prise en compte par la fonction de poids. Plusieurs exemples sont discutés au chapitre 5. Lorsque cette dernière est constante (cas de graphes non-pondérés), les m processus de diffusion sont totalement indépendants.

Cas de la régularisation isotrope ($p \neq 2$). Dans ce cas, le p-Laplacien isotrope (4.23) fait intervenir la variation locale (4.10). Effectuer la diffusion composante par composante ne permet pas de prendre en compte la dépendance entre les m composantes de la fonction. De plus, si les poids w_{uv} n'incorporent pas d'information entre les composantes, la régularisation est totalement indépendante sur chacune des m composantes.

Afin de corréler les composantes entre elles, nous remplaçons la variation locale sur la $i^{\text{ème}}$ composante par la variation locale sur les m composantes :

$$\| \bigtriangledown_w f(u)\|_2 := \left(\sum_{i=1}^{i=m} \| \bigtriangledown_w f_i(u)\|_2^2 \right)^{\frac{1}{2}}, \quad f : V \to \mathbb{R}^m. \tag{4.65}$$

L'expression du p-Laplacien isotrope, pour la $i^{\text{ème}}$ composante de la fonction f, devient alors :

$$\Delta_w^p f_i(u) := \tfrac{1}{2} \sum_{v \sim u} w_{uv} \left(\| \bigtriangledown_w f(u)\|_2^{p-2} + \| \bigtriangledown_w f(v)\|_2^{p-2} \right) (f_i(u) - f_i(v)), \quad \forall i = 1, \ldots, m.$$

Cas de la régularisation anisotrope ($p \neq 2$). La même démarche peut être effectuée pour l'expression de la diffusion anisotrope. Celle-ci fait intervenir le p-Laplacien anisotrope (4.29), que nous redéfinissons en chaque sommet $u \in V$ par :

$$\widetilde{\Delta_w^p} f_i(u) := \sum_{v \sim u} w_{uv}^{\frac{p}{2}} \| f(v) - f(u)\|_1^{p-2} (f_i(u) - f_i(v)), \qquad \forall i = 1, \ldots, m. \tag{4.66}$$

Quelque soit la régularisation considérée (isotrope ou anisotrope) pour $p \neq 2$, et si le graphe est non-pondéré, les expressions des p-Laplaciens ci-dessus montrent que les composantes sont toujours corrélées, contrairement au cas où $p = 2$. Dans ce dernier cas, la pondération joue un rôle important pour prendre en compte la dépendance entre les composantes.

4.6 Conclusion et perspectives

Dans ce chapitre, nous avons présenté des modèles variationnels discrets pour régulariser des fonctions définies sur des graphes pondérés de topologie arbitraire. Nous avons proposé deux types de fonctionnelles de régularisation, l'une basée sur la norme $\mathcal{L}_2$ du gradient discret, et l'autre sur la norme $\mathcal{L}_p$ de ce même gradient. Nous avons montré le lien avec les fonctionnelles de régularisation locale et non-locale continue, principalement utilisées en traitement d'images. Contrairement à ces méthodes de régularisation continues, basée sur des EDP de diffusion, les algorithmes de résolution que nous avons proposés ne nécessitent aucune discrétisation spatiale. Un lien a également été établi avec le filtrage de fonctions discrètes dans le domaine spectral.

Nous avons considéré ici un type particulier de régularisation. Le formalisme introduit permet également d'exprimer sous forme discrète de nombreuses fonctionnelles de régularisation utilisées en traitement d'image ou de maillages. Une grande partie de ces fonctionnelles sont basées sur des dérivées d'ordre un ou deux, et sur des normes continues. Comme nous l'avons fait, il suffit de remplacer les dérivées et les normes par leurs analogues discrets. Par la suite, il conviendra d'appliquer les concepts sur graphes pour résoudre des problématiques comme le flot optique, la stéréo vision, la segmentation, ou la partition de graphes.

Chapitre 5

Applications

Nous avons proposé des familles de modèles variationnels discrets pour régulariser des fonctions définies sur les sommets de graphes pondérés. Ceci permet de régulariser des ensembles de données discrètes, ou des fonctions définies sur ces données, en considérant différents graphes de voisinage. En fonction du choix de la structure du graphe et du degré de régularité p, la régularisation permet d'effectuer des traitements de nature différente. En particulier, lorsque $p \leq 2$ la régularisation lisse les fonctions à traiter tout en préservant leurs discontinuités sur le graphe. Dans ce chapitre, nous illustrons le comportement des familles de régularisation sur des images discrètes, des courbes et des surfaces polygonales, ainsi que sur des données discrètes non-organisées. Notre objectif n'est pas de trouver les meilleurs paramètres, mais de montrer l'action du degré de régularité et de la structure du graphe sur la régularisation. Nous l'illustrons sur des applications de lissage, de débruitage et de simplification.

5.1 Applications au traitement d'images

Cette section présente le comportement des processus de régularisation isotrope et anisotrope pour des images discrètes, représentées par des graphes de voisinage. Selon le choix du voisinage, et de la fonction de poids, nous retrouvons la version discrète de plusieurs méthodes de régularisation locales et non-locales basées sur des EDP de diffusion. L'intérêt de passer de l'image au graphe est de pouvoir représenter explicitement des interactions locales et non-locales, et de pouvoir travailler sur des représentations plus abstraites, comme par exemple des images pré-segmentées.

5.1.1 Des images aux graphes

Une image discrète peut être vue comme une application $I : \mathcal{P} \subset \mathbb{Z}^2 \to \mathcal{Q} \subset \mathbb{R}^m$ qui associe à chaque cellule (i, j) d'une grille régulière $\mathcal{P}$ une intensité de l'espace de quantification $\mathcal{Q}$. Pour régulariser l'image I, celle-ci est transformée en un graphe $G := (V, E, w)$ et une fonction $f^0 : V \to \mathcal{Q}$. Il y deux façons de construire ce graphe. La première est la représentation naturelle des images discrètes, où V est l'ensemble des cellules $\mathcal{P}$. Les interactions entre les sommets, c.-à-d. les arêtes pondérées de l'ensemble E, sont générées à partir de relations de voisinage. Dans ce cas, l'image est modélisée par la fonction :

(i) $\xi : \mathcal{P} \to V = \mathcal{P}, \quad (i, j) \mapsto (i, j).$

La deuxième représentation consiste à plonger l'image dans un espace discret de dimension supérieure, où V est l'ensemble des points de cet espace. L'image est alors considérée comme une fonction sur les sommets de ce graphe. Par exemple, on peut considérer les deux représentations suivantes :

(ii) $\xi : \mathcal{P} \to V \subset \mathcal{P} \times \mathcal{Q}, \quad (i, j) \mapsto (i, j, I(i, j)).$
(iii) $\xi : \mathcal{P} \to V \subset \mathcal{P} \times \mathcal{Q} \times \mathcal{F}, \quad (i, j) \mapsto ((i, j), I(i, j), F_I(i, j)),$

où $F_I \in \mathcal{F} \subset \mathbb{R}^n$ exprime des caractéristiques intrinsèques à l'image I. La première représentation (cas (i)) revient à traiter l'image dans son espace géométrique naturel, tandis que la seconde (cas (ii) et (iii)) consiste à modifier son domaine de définition. Cette dernière représentation est plus riche, dans le sens où la construction du graphe permet de préserver directement certaines structures de l'image. Le poids des arêtes permet alors de prendre en compte la géométrie du domaine de définition par une expression de la forme :

$$w_{uv} := \gamma(\rho(u, v)), \quad u, v \in V,$$

où $\rho : V \times \mathbb{R}^+$ est une mesure de distance, et $\gamma : \mathbb{R}^+ \to \mathbb{R}^+$ est une fonction de dissimilarité. Plusieurs exemples de ces fonctions sont données par la suite. Quelque soit la représentation de l'image, et le choix de la fonction de poids w, les interactions entre les sommets du graphe peuvent être modélisées par des voisinages. Ces voisinages sont générés à partir d'une distance, comme par exemple ρ. Dans la suite, nous considérons des voisinages géométriques sur la grille régulière $\mathcal{P}$.

Construction des graphes de voisinage. Soit $G_r := (V, E, w)$ le graphe pondéré où le voisinage d'un sommet $u \in V$ est l'ensemble des cellules de $\mathcal{P}$ situées à une distance $r \in \mathbb{Z}$ de u selon une métrique de $\mathbb{Z}^2$:

$$\mathcal{N}_r(u) = \{v \in V \setminus \{u\} \ : \ \rho_{\mathbb{Z}^2}(u, v) \leq r\}. \tag{5.1}$$

Nous utilisons la distance de Chebyshev définie par :

$$\rho_{\mathbb{Z}^2}(u = (i, j),\, v = (i', j')) = \max\{|i - i'|,\, |j - j'|\}. \tag{5.2}$$

Comme illustré aux figures 5.1(a) et (b), le voisinage d'un sommet, construit à partir de cette distance, correspond à fenêtre carrée de $\mathbb{Z}^2$ centrée en ce sommet, et de coté $(2r+1)$.

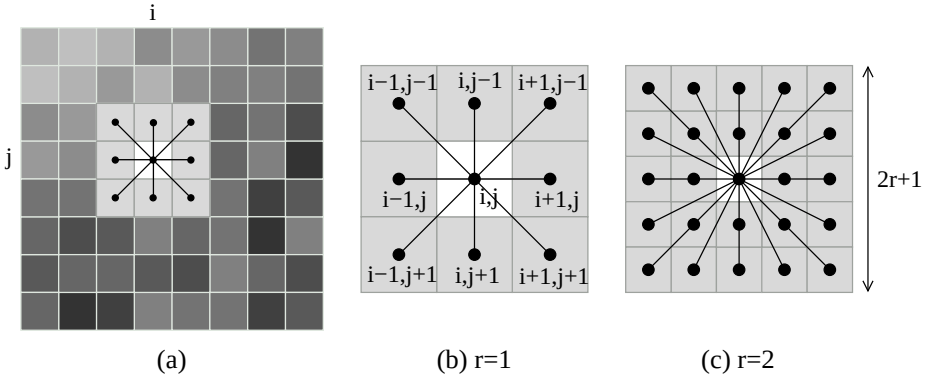


Fig. 5.1: Voisinage $\mathcal{N}_r$ d'un sommet (i, j) du graphe selon la distance (5.2). (a) Une grille régulière cartésienne carrée. (b) $\mathcal{N}_1$ correspond à la 8-adjacence. (c) $\mathcal{N}_2$.

Remarquons que le voisinage ainsi défini est symétrique. Une arête uv appartient alors à E si $v \in \mathcal{N}(u)$ (et $u \in \mathcal{N}(v)$). Le graphe complet formé à partir du voisinage $\mathcal{N}_\infty(u) = V \setminus \{u\}$ est noté G_∞. De plus, $\mathcal{N}_1$ est le voisinage classique en 8-adjacence. Le voisinage en 4-adjacence est noté $\mathcal{N}_0$.

Les voisinages $\mathcal{N}_0$ et $\mathcal{N}_1$ sont **locaux** (géométriquement dans $\mathbb{Z}^2$), ils décrivent la topologie naturelle des images. Le voisinage $\mathcal{N}_r$ est **semi-local** si $r > 1$, et **non-local** lorsqu'il est constitué de tous les sommets du graphe (voisinage $\mathcal{N}_\infty$).

5.1.2 Comportement des régularisations proposées

Selon la taille du voisinage et le choix de la fonction de poids w, on peut alors formuler différentes classes de régularisations locales, semi-locales et non-locales. Pour calculer la régularisation, nous utilisons la diffusion discrète, qui à l'avantage de ne pas faire intervenir de paramètre d'approximation. L'image $I^0 : \mathcal{P} \subset \mathbb{Z}^2 \to \mathbb{R}^m$ sur le graphe G_r est alors régularisée avec le processus de diffusion discrète, qui dans le contexte se réécrit :

$$\begin{cases} I^{(0)} = I^0, \\ I^{(t+1)}(u) = \dfrac{\lambda I^0(u) + \sum_{v \in \mathcal{N}_r(u)} \gamma_w^{I^{(t)}} I^{(t)}(v)}{\lambda + \sum_{v \in \mathcal{N}_r(u)} \gamma_w^{I^{(t)}}}, \end{cases} \tag{5.3}$$

avec :

$$\gamma_w^{I^{(t)}}(u,v) = \tfrac{1}{2} w_{uv} (\| \bigtriangledown_u I^{(t)}(u)\|_2^{p-2} + \| \bigtriangledown_v I^{(t)}(v)\|_2^{p-2}),$$

dans le cas de la régularisation isotrope, ou :

$$\gamma_w^{I^{(t)}}(u,v) = w_{uv}^{\frac{p}{2}} \|I^{(t)}(u) - I^{(t)}(v)\|_1^{p-2},$$

dans le cas de la régularisation anisotrope. Dans la suite, nous considérons plus particulièrement la régularisation isotrope.

Régularisations locales. En considérant le graphe G_0 ou G_1, et la fonction de poids :

$$w_1(u,v) = \frac{1}{\|u-v\|_2^2}, \quad \forall (u,v) \in E, \tag{5.4}$$

les modèles variationnels $E_{w_1}^{TVi}$ et $E_{w_1}^{TVa}$ sont les versions discrètes des modèles variationnels isotrope et anisotrope continus :

$$\mathcal{E}^{TVi}(f^0, f, \lambda, p) = \frac{1}{p} \int_\Omega \| \bigtriangledown f\|_2^p dx + \frac{\lambda}{2} \|f - f^0\|^2, \quad f^0, f : \Omega \subset \mathbb{R}^2 \to \mathbb{R},$$

$$\mathcal{E}^{TVa}(f^0, f, \lambda, p) = \frac{1}{2p} \int_\Omega \| \bigtriangledown f\|_p^p dx + \frac{\lambda}{2} \|f - f^0\|^2, \quad f^0, f : \Omega \subset \mathbb{R}^2 \to \mathbb{R}.$$

Ces deux modèles sont des extensions du modèle ROF [Rudin *et al.*, 1992]. Leur minimisation est généralement obtenue en discrétisant des EDP de diffusion sur le graphe G_0 (voir par exemple [Aubert et Kornprobst, 2006]). Les dérivées ou différences utilisées dans ces discrétisations sont similaires à celles de l'expression de la dérivée directionnelle (4.5) pour w_1 :

$$\partial_v I^0(u) = \frac{I^0(v) - I^0(u)}{\|u-v\|_2}.$$

Nous illustrons le comportement de la régularisation isotrope pour différentes taille de voisinage et valeurs de p. On peut observer à la figure 5.2, que pour les mêmes paramètres de régularisation (p et λ), la topologie du graphe influe sur les résultats (voir les trois

premières images de la première ligne). Pour un graphe de même topologie, le lissage linéaire ($p = 2$) ne préserve pas les contours. On peut observer que lorsque $p \leq 1$, la régularisation isotrope permet de lisser l'image tout en préservant ses discontinuités. En particulier, lorsque $p \to 0$, la régularisation décompose l'image en une fonction presque constante par morceaux. La régularisation se comporte alors comme un processus de regroupement.

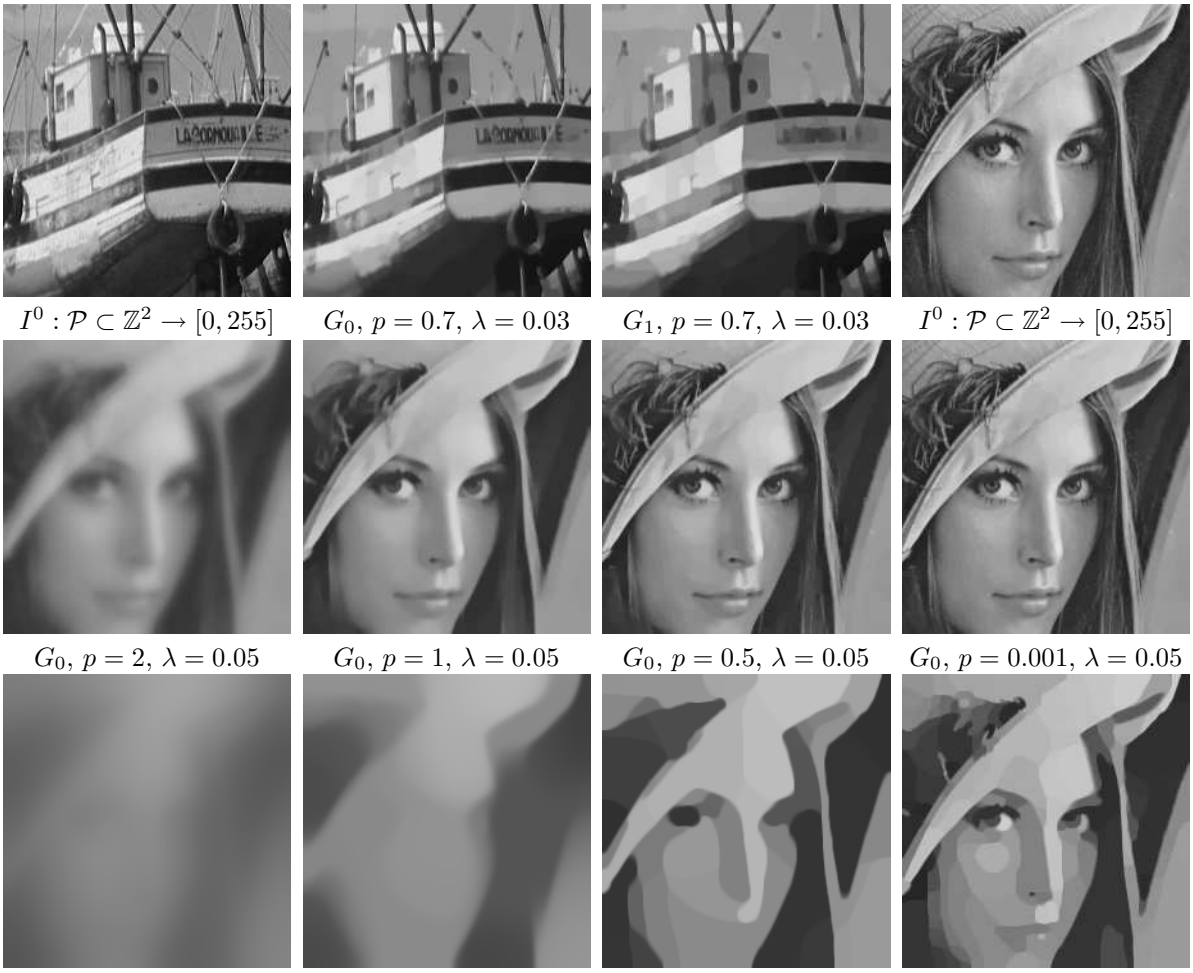
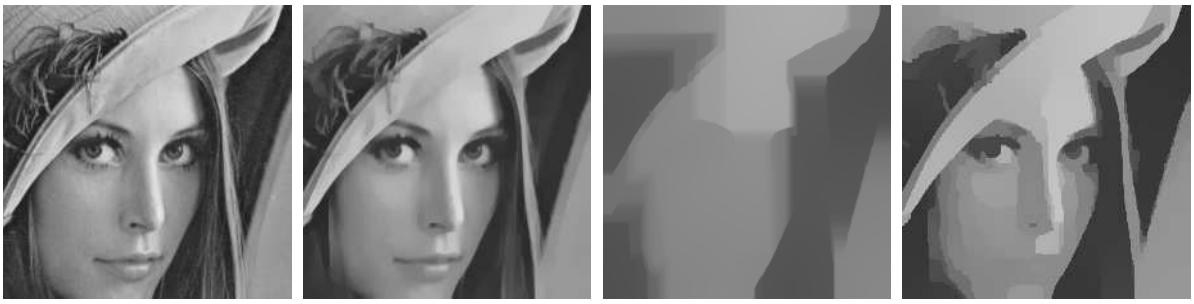


Fig. 5.2: Régularisation isotrope locale de I^0 par diffusion discrète sur G_0 avec w_1 (cas non-pondéré).

Le comportement de la régularisation anisotrope est illustré à la figure 5.3 pour $p = 1$ et $p = 0.5$. On peut observer qu'à la même échelle λ et pour le même degré de régularité, la régularisation anisotrope conserve mieux les discontinuités principales de l'image que la régularisation isotrope (voir le cas $p = 1$ à la figure 5.2).

On peut également considérer une fonction de poids qui tient compte de la géométrie de



$I^0 : \mathcal{P} \subset \mathbb{Z}^2 \to [0, 255]$ $p = 1$, $\lambda = 0.05$ $p = 1$, $\lambda = 0.001$ $p = 0.5$, $\lambda = 0.001$

Fig. 5.3: Régularisation anisotrope locale de I^0 par diffusion semi-discrète ($\tau = 0.03$) sur G_0 (cas non-pondéré).

l'image I^0 sur la grille, comme par exemple :

$$w_2(u, v) = \frac{1}{\|u - v\|_2^2} \exp\left(-\frac{\|I^0(v) - I^0(u)\|_2^2}{\sigma^2}\right), \quad \forall (u, v) \in E, \tag{5.5}$$

où $\sigma > 0$ est la variance du noyau Gaussien. Cela permet de favoriser la régularisation des zones de l'image I^0 où les valeurs sont similaires. Ceci est illustré sur le graphe $G^0 = (\mathcal{P}, E, w_2)$ à la figure 5.4(b) pour la régularisation isotrope, et à la figure 5.4(a) pour la régularisation anisotrope. Afin de comparer avec les régularisations

(a) Régularisation anisotrope locale sur G_0

$I^0 : \mathcal{P} \subset \mathbb{Z}^2 \to [0, 255]$ w_2, $p = 1$, $\lambda = 0.001$

(b) Régularisation isotrope locale sur G_0

$p = 2$, $\lambda = 0.001$ $p = 1$, $\lambda = 0.001$ $p = 0.5$, $\lambda = 0.001$ $p = 0.001$, $\lambda = 0.001$

Fig. 5.4: Régularisations isotropes et anisotropes locales sur G_0 (4-adjacence) avec w_2 (noyau Gaussien avec $\sigma = 15$).

sur $G^0 = (\mathcal{P}, E, w_1)$, les mêmes valeurs des paramètres p et λ sont utilisées. Le choix de la variance σ dépend de l'application. Néanmoins, plus la valeur de σ est élevée, et plus les régularisations sont proches de celles obtenues en considérant la fonction de poids w_1. La valeur de σ peut également être calculée localement en chaque arête du graphe. Bien que l'introduction d'une fonction de poids, qui dépend des valeurs de l'image, permet de préserver certaines structures, on peut également apprécier le comportement de la régularisation lorsque $p < 1$.

Pour une image $I^0 : \mathcal{P} \subset \mathbb{Z}^2 \to \mathbb{R}^m$ avec $m > 1$ (fonction à valeurs vectorielles), une autre façon de prendre en compte la géométrie de l'image au cours de la régularisation est de considérer la fonction de poids :

$$w_3(u,v) = \frac{1}{\|u-v\|_2^2(\|I^0(v) - I^0(u)\|_2^2 + \epsilon)}, \qquad \forall (u,v) \in E,\ \epsilon \to 0. \tag{5.6}$$

Les processus de régularisation isotrope ou anisotrope ainsi obtenus utilisent des interactions locales pour traiter les images. Selon le choix du degré de régularité p et de la fonction de poids, ils conservent les structures géométriques principales présentes dans les images. En revanche, les structures géométriques plus fines et les textures sont difficilement préservées. Afin de traiter correctement ces informations, leur degré de redondance nécessite d'être exploité au cours de la régularisation. La redondance d'information est modélisée en considérant des graphes où le voisinage des sommets est suffisamment important.

Régularisations semi-locales et non-locales. Soit G_r un graphe de voisinage tel que $r > 1$. La régularisation sur G_r prend alors en compte des interactions semi-locales ou non-locales (sur G_∞). Pour calculer le poids des arêtes du graphe, on peut par exemple supposer que les sommets proches géométriquement ont plus de probabilité d'être similaires :

$$w_4(u,v) = \exp\left(-\frac{\|u-v\|_2^2}{\sigma_{\mathcal{P}}^2}\right) \exp\left(-\frac{\|I^0(u) - I^0(v)\|_2^2}{\sigma_{I^0}^2}\right). \tag{5.7}$$

Lorsque le graphe $G_\infty := (\mathcal{P}, E, w_4)$ est considéré, une itération du processus de diffusion isotrope discrète (5.3), avec $\lambda = 0$ et $p = 2$, est équivalent à l'expression du filtre bilatéral [Tomasi et Manduchi, 1998, Smith et Brady, 1997].

Une autre façon de rechercher les redondances contenues dans l'image I^0 est de supposer que les sommets éloignés peuvent avoir autant d'importance que les plus proches. Ceci peut être effectué en considérant la fonction de poids w_4 et en négligeant le terme géométrique ($\sigma_{\mathcal{P}} \to +\infty$). Mais dans ce cas, la similarité de deux sommets ne dépend que de l'information locale à chaque sommet, c.-à-d. la valeur de l'image I^0. Une meilleure manière de mesurer la similarité de deux sommets est de comparer des caractéristiques intrinsèques à l'image I^0 en ces sommets. Plusieurs caractéristiques peuvent être considérées. Nous choisissons ici d'utiliser des patches centrés en chaque sommet. Cette idée a été proposée pour synthétiser des images [Efros et Leung, 1999] et plus récemment pour les débruiter et les restaurer [Buades *et al.*, 2005, Kinderman *et al.*, 2005, Gilboa

et Osher, 2006]. Le poids d'une arête du graphe G_r est alors calculé par la fonction :

$$w_5^{r'}(u,v) = \exp\left(-\frac{\rho_a(F_{r'}(I^0,u), F_{r'}(I^0,v))^2}{h^2}\right), \tag{5.8}$$

où la caractéristique locale $F_{r'}(I^0,u)$ est l'ensemble des valeurs de la fonction I^0 dans le r'-voisinage de u, avec $0 \leq r' < r$ fixé, ainsi que la valeur de I^0 en u (voir figure 5.5) :

$$F_{r'}(I^0,u) = \{I^0(v) \ : \ v \in \mathcal{N}_{r'}(u) \cup \{u\}\}, \tag{5.9}$$

La métrique ρ_a mesure la distance entre les valeurs de l'image I^0 sur les patches associées aux deux sommets u et v :

$$\rho_a^2(F_{r'}(I^0,u), F_{r'}(I^0,v)) = \sum_{i=-r'}^{r'} \sum_{j=-r'}^{r'} g_a((i,j)) \|I^0(u+(i,j)) - I^0(v+(i,j))\|_2^2,$$

où g_a est un noyau gaussien de variance a. Pour des raisons de complexité en temps, G_a peut être remplacé par la distance de Chebyshev (équation (5.2)).

Fig. 5.5: Voisinage $\mathcal{N}_r$ d'un sommet du graphe G_r, et interactions entre les arêtes avec la fonction de poids w_5 ($\mathcal{W}_{r'}(p) = \mathcal{N}_{r'}(p) \cup \{p\}$).

Pour une itération, $p = 2$ et $\lambda = 0$, le processus de diffusion isotrope discrète (5.3), sur le graphe $G_\infty := (\mathcal{P}, E, w_5^{r'})$, correspond au filtre à moyennes non-locales (NLMEANS) [Buades *et al.*, 2005].

Ainsi, pour $p = 2$, la régularisation isotrope avec les fonctions de poids w_4 et w_5 se comportent respectivement comme un filtre bilatéral itératif et NLMEANS itératif. La différence avec l'utilisation itérative classique de ces filtres réside dans la mise à jour des poids au cours des itérations. Dans la régularisation proposée, les poids dépendent de l'image initiale I^0 et non de $I^{(t)}$. Itérer le filtre bilatéral ou NLMEANS conduit à une image presque constante par morceaux. C'est également le cas avec la régularisa-

tion proposée. De plus, l'utilisation de $\lambda > 0$ permet de se rapprocher du contenu de l'image initiale. D'autre part, lorsque $p < 1$, la régularisation (ou la diffusion) permet d'améliorer la préservation des discontinuités de l'image. Ceci est illustré à la figure 5.6 pour une image $I^0 : \mathcal{P} \to \mathbb{R}$ sur le graphe $G_7 := (\mathcal{P}, E, w_5^5)$. Afin d'apprécier le rôle du terme de régularisation, le terme d'approximation est négligé ($\lambda = 0$). D'autre part, afin de comparer les résultats avec la régularisation anisotrope locale, un nombre fixé d'itérations est imposé (30 itérations). Ceci revient à effectuer une diffusion isotrope selon le p-Laplacien non-local. Nous pouvons alors observer la différence entre la diffusion local et non-local. Tout en augmentant les performances de lissage, il préserve mieux les structures fines et les textures. Nous pouvons également remarquer que lorsque $p < 1$ nous avons une meilleure préservation des contours. De plus, lorsque $p \to 0$, l'utilisation d'un voisinage non-local permet d'accélérer le processus de regroupement constaté pour la régularisation locale.

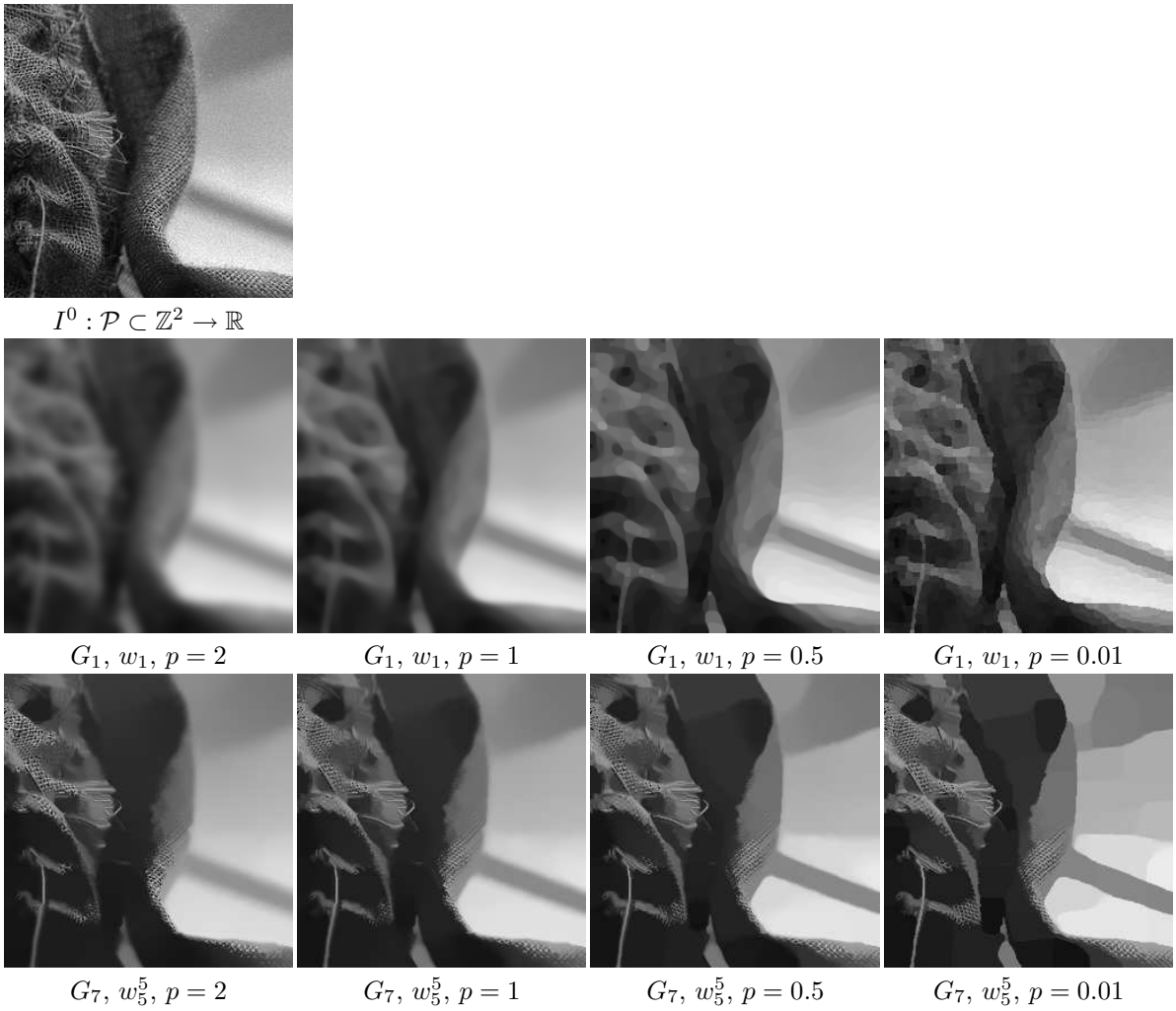


Fig. 5.6: Diffusion isotrope non-locale (discrète) sur le graphe $G_7 := (\mathcal{P}, E, w_5^5)$ pour différentes valeurs du paramètre de régularité (dernière ligne). Comparaison avec la diffusion isotrope locale sur $G_1 := (\mathcal{P}, E, w_1)$ (deuxième ligne).

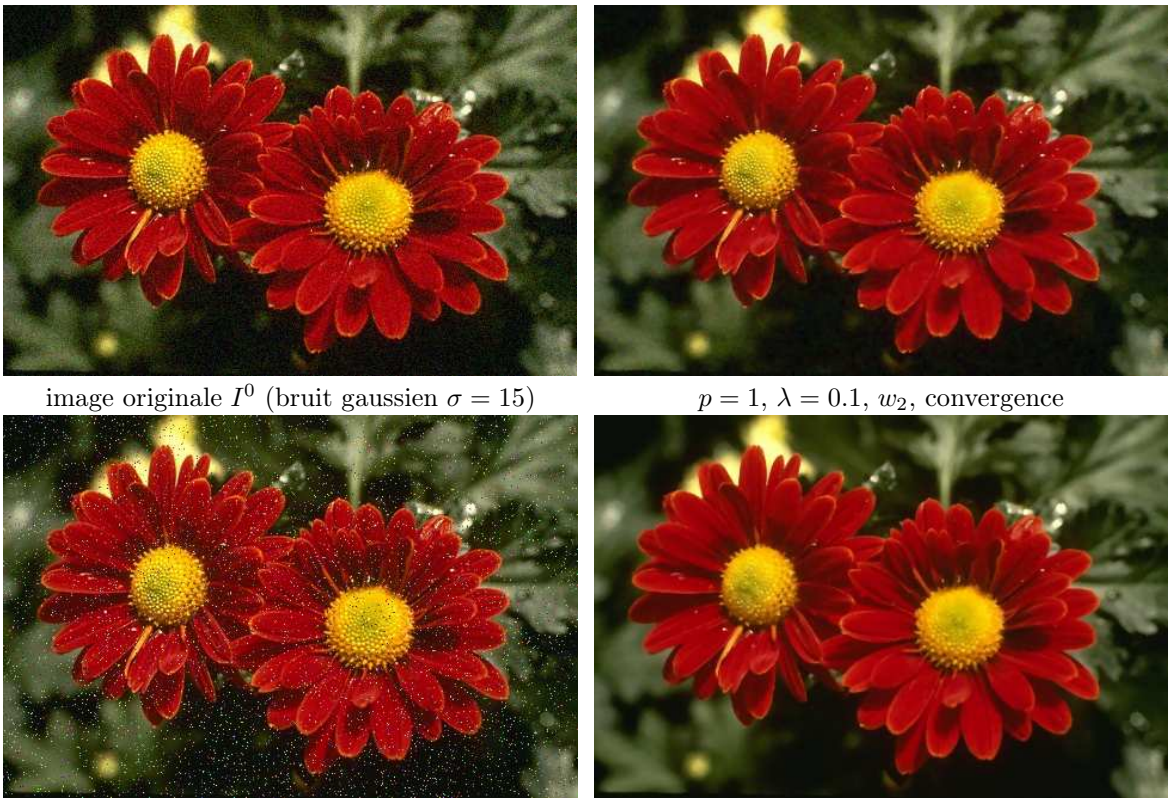


image originale I^0 (bruit gaussien $\sigma = 15$) — $p = 1$, $\lambda = 0.1$, w_2, convergence

image originale I^0 (15% de bruit impulsionel) — $p = 1$, $\lambda = 0.001$, w_3, $n_{\text{it}} = 6$

Fig. 5.7: Débruitage d'une image couleur (RVB) par régularisation isotrope locale sur le graphe G_1.

5.1.3 Débruitage et restauration par régularisation isotrope

Les applications principales de la régularisation isotrope sont le débruitage, la restauration et plus généralement l'amélioration d'images dégradées. Soit $I^0 : \mathcal{P} \subset \mathbb{Z}^2 \to \mathbb{R}^m$ une image dégradée (bruitée) telle que $I^0 = I_o + \mu$, avec $I_o, \mu : \mathcal{P} \subset \mathbb{Z}^2 \to \mathbb{R}^m$. L'image I_o est la version sans dégradation de I^0, et μ est la dégradation. En utilisant la régularisation, le débruitage de I^0 consiste à estimer l'image I_o par une version régularisée de I^0 :

$$I_o \approx I = \min_{I:\mathcal{P}\to\mathbb{R}^m} \{R_w^{TVi}(f,p) + \tfrac{\lambda}{2}\|I - I^0\|_2^2\}, \quad p \leq 2.$$

La dégradation (le bruit) est alors estimée par $\mu \approx I^0 - I$. Le degré de régularité p, la structure du graphe (la taille du voisinage) et la fonction de poids sont choisis en fonction de la nature de la dégradation. De plus, si des contraintes sont imposées sur le temps d'exécution de la régularisation, les versions locales, ou semi-locales (avec des voisinages peu importants) répondent mieux à ces contraintes.

Exemples de débruitage. La régularisation discrète permet de débruiter des images corrompues par différents types de bruits. La fonction de poids doit être adaptée à la nature du bruit. En particulier, pour un *bruit additif Gaussien*, les fonctions de poids basées sur des noyaux Gaussiens sont préférables, comme les fonctions de poids w_2, w_4 et w_5 (voir Section 5.1.2). Pour un *bruit impulsionnel*, nous avons observé que des fonctions

de poids de type w_3 fournissent de meilleurs résultats [Lézoray *et al.*, 2006, Lézoray *et al.*, 2007b].

À la figure 5.7 on peut observer cette régularisation dans le cas d'une image couleur $I^0 : \mathcal{P} \to [0, 255]^3$ bruitée de deux façons. La régularisation considérée est locale avec $p = 1$ (préservation des contours), sur le graphe G_1. Dans le cas du bruit Gaussien l'algorithme de diffusion est itéré jusqu'à la convergence ($\lambda \neq 0$). Pour une image bruitée par un bruit impulsionnel, la nature de ce bruit impose que le terme d'approximation soit négligé ($\lambda \to 0$). De ce fait, on ne peut pas itérer l'algorithme de diffusion jusqu'à la convergence, et le nombre d'itérations doit être choisi explicitement, donnant ainsi une diffusion pure.

image originale — image bruitée I^0 (bruit Gaussien $\sigma = 20$)

G_1, $p = 1$ et w_1 et $n_{\text{it}} = 8$ — G_4, $p = 2$, w_5^2 et $n_{\text{it}} = 2$.

Fig. 5.8: Diffusion isotrope non-locale et comparaison avec la diffusion isotrope locale pour $\lambda = 0.1$.

Le débruitage illustré à la figure 5.7 est basé sur la diffusion discrète locale selon la courbure ($p = 1$). Les contours les plus importants sont ainsi préservés. En pratique, nous avons observé que des valeurs de $p \in [0.7, 1]$ fournissent des résultats satisfaisants. La régularisation locale est intéressante lorsque le bruit est peu important, et lorsque

les images ne comportent pas de détails fins à préserver, comme par exemple des structures fines ou des textures. Par contre lorsque le bruit est important, ou qu'il y a des structures répétitives, la régularisation non-locale permet de lisser l'image tout en préservant la majorité de ces structures. Comme nous l'avons vu dans la section précédente, nous considérons la diffusion discrète non-locale avec un nombre fixé d'itérations (1 à 6 itérations suffisent si $\lambda \to 0$). La figure 5.8 illustre la différence entre le lissage local et non-local sur une image comportant des textures. Un autre exemple de débruitage non-local est illustré avec $p = 1$ à la figure 5.9 sur une image comportant des détails fins. Nous avons constaté que pour des tailles importantes de voisinage, la régularisation avec $p = 2$ est généralement suffisante pour débruiter correctement les images.

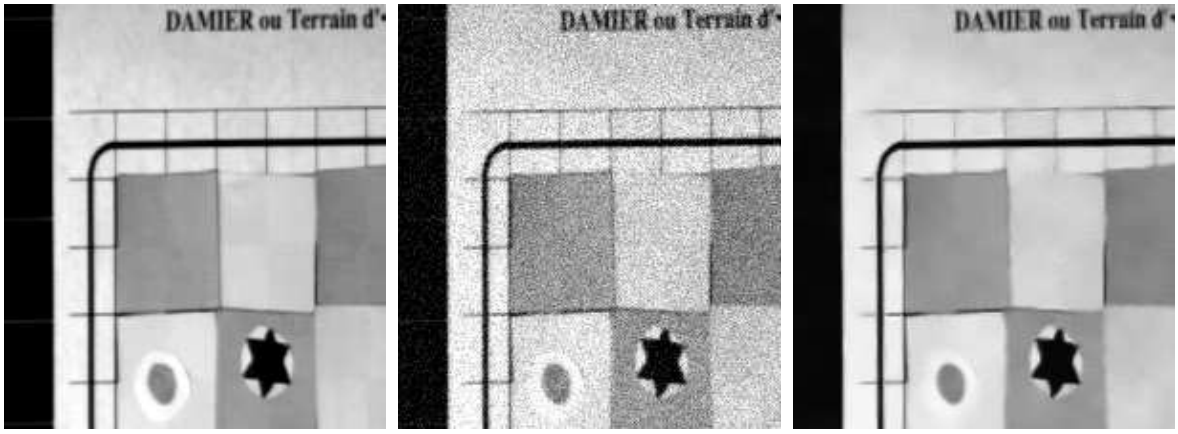


image originale | I^0 (bruit Gaussien $\sigma = 20$) | G_5, w_5^3, $p = 1$, $\lambda = 0.1$

Fig. 5.9: Débruitage d'une image bruitée par diffusion isotrope non-locale.

Un autre exemple de débruitage par diffusion non-locale, avec $p = 0.7$, est illustré à la figure 5.10 sur une image RVB bruitée (bruit Gaussien). Deux niveaux de bruit sont considérés ($\sigma = 20$ et $\sigma = 40$). De nouveau, nous pouvons observer que la régularisation permet de diminuer le bruit tout en préservant les structures fines et répétitives principales. Néanmoins, on peut aussi constater que lorsque le bruit est important (voir le cas $\sigma = 40$), les structures et textures fines se trouvent lissées. Ceci provient du fait que le bruit se confond avec ces structures.

D'une manière générale, un bon compromis pour débruiter des images est de considérer la régularisation isotrope, avec $p = 1$, sur des graphes pondérés par w_4 ou w_5 et dont le voisinage est moyennement important ($r = 3$ à $r = 11$). Les autres paramètres, comme le nombre d'itérations et la valeur de λ peuvent alors être calculés automatiquement. Nous ne décrivons pas ces calculs, qui sont détaillés dans [Lézoray *et al.*, 2007a].

image originale RVB

zoom

image bruitée I^0 (bruit Gaussien $\sigma = 20$)

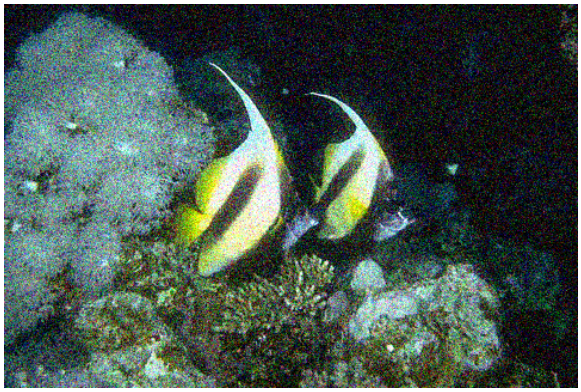
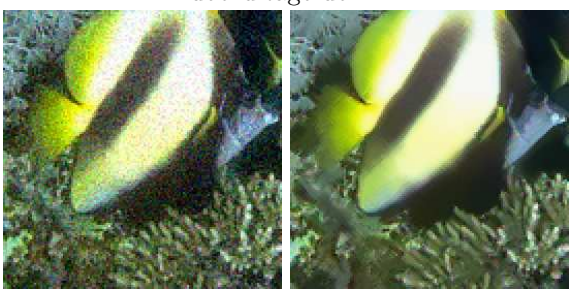


image bruitée I^1 (bruit Gaussien $\sigma = 40$)

débruitage de I^0

débruitage de I^1



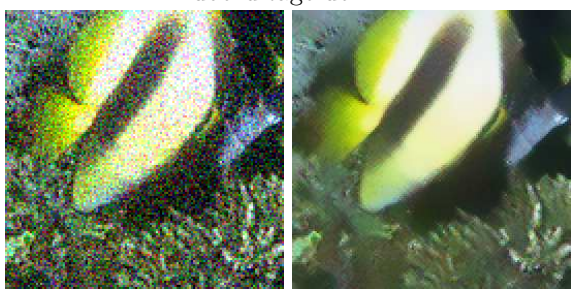


Fig. 5.10: Débruitage par diffusion isotrope non-locale sur G_5, avec $p = 0.7$, $\lambda = 0.001$, w_5^3, et $n_{\text{it}} = 2$.

5.1.4 Simplification par pré-segmentation et régularisation

Nous avons montré comment régulariser des images définies sur des grilles régulières en les modélisant par des graphes de voisinage dont la topologie est également régulière. De la même manière, nous montrons comment se comporte le processus de régularisation sur des graphes de topologie arbitraire, avec des images pré-segmentées.

Soit $I : \mathcal{P} \subset \mathbb{Z}^2 \to \mathbb{R}^m$ une image représentée par le graphe géométrique $G_r := (\mathcal{P}, E, w)$, et une fonction $f^0 : \mathcal{P} \to \mathbb{R}^m$. Lorsque $p \leq 1$, nous avons vu que la régularisation de f^0 simplifie l'information contenue dans l'image I tout en préservant ses caractéristiques principales (locales ou non-locales). Afin d'analyser ou de décrire l'image, d'autres représentations doivent être considérées. Le graphe G_r n'est pas adapté à la représentation visuelle des objets contenus dans l'image. Il est plus naturel et efficace de travailler sur des partitions de l'image, obtenues par des processus de pré-segmentation ou de segmentation. Rappelons qu'une partition de l'image I est un ensemble de régions disjointes $R_1, \ldots, R_q$ telles que $R_1 \cup \ldots \cup R_q = I$. À chaque région est usuellement associée la valeur moyenne ou médiane de I dans cette région. Dans cette section, nous proposons une méthode rapide de simplification par calcul de partitions fines et par régularisation de ces partitions. Une partition fine de l'image I est une pré-segmentation proche de l'image originale. La structure de graphe, associée à une partition (fine), est le graphe d'adjacence de régions (RAG). À chaque sommet du RAG est associé une région R_i de la partition, et deux sommets du RAG sont connectés par une arête si les régions qui y sont associées ont au moins deux cellules voisines dans l'image I (en 4 ou 8-adjacence de pixels).

Dans la suite, nous présentons la méthode qui nous permet de calculer la partition fine, puis nous montrons que la régularisation discrète peut être utilisée sur le RAG pour simplifier cette partition.

Pré-segmentation par partition d'énergie sur graphes. Il existe de nombreuses méthodes pour calculer une partition fine d'une image $I : \Omega \subset \mathbb{Z}^2 \to \mathbb{R}^m$. Une classe importante de ces méthode consiste à plonger un ensemble S de germes ponctuels dans I, et à calculer un diagramme de Voronoï généralisé de S en utilisant une distance qui tient compte de la valeur des pixels. Un tel diagramme peut être obtenu de plusieurs façons. De manière à rester dans le cadre général de données définies sur les sommets d'un graphe $G := (V, E, w)$, nous proposons ici une version analogue à celle proposée par [Arbeláez et Cohen, 2004] dans le contexte du domaine de l'image. Pour cela, nous définissons le diagramme de Voronoï généralisé d'un sous-ensemble de sommets de V [Bougleux et Elmoataz, 2005].

Rappelons qu'un **chemin** entre deux sommets u et v de V, noté $c(u, v)$, est une séquence $u = v_1, \ldots, v_k = v$ telle que chaque paire $v_i v_{i+1}$ soit une arête de E, pour $i = 1, \ldots, k-1$. Soit $C_G(u, v)$ l'ensemble des chemins qui connectent les sommets u et v dans G. Soit $f \in \mathcal{H}(V)$ une fonction définie sur les sommets du graphe. Soit $|c(u, v)|$ la

longueur du chemin $c(u,v)$ définie par :

$$|c(u,v)| = \sum_{i=1}^{i=k-1} d(f)(v_i v_{i+1}) = \sum_{i=1}^{i=k-1} \sqrt{w(v_i v_{i+1})}(f(v_{i+1}) - f(v_i)), \tag{5.10}$$

où $d(f)$ est l'opérateur différence définit à la Section 4.1.2.

Le graphe G peut alors être muni d'une métrique $\rho_G : V \times V \to \mathbb{R}^+$, qui mesure le chemin de plus faible longueur entre deux sommets de G :

$$\rho_G(u,v) = \min_{c(u,v) \in C_G(u,v)} |c(u,v)|. \tag{5.11}$$

La métrique ρ_G est une géodésique sur G. Soit $S = \{s_1, \ldots, s_l\}$ un sous-ensemble de sommets de V, appelés **sommets sources** ou **germes**. L'énergie minimale induite de la métrique ρ_G, relativement à l'ensemble S, en un sommet source $u \in V$ est définie par :

$$\rho_S(u) = \inf_{s_i \in S} \rho_G(s_i, u). \tag{5.12}$$

Par analogie avec le diagramme de Voronoï, la **zone d'influence** d'un sommet source $s_i \in S$, notée $Z_G(s_i, S)$, est définie comme le sous-graphe de G constitué de l'ensemble des sommets de V situés plus près de s_i que des autres sommets sources de S (relativement à la métrique ρ_G), et de l'ensemble des arêtes de G qui connectent les sommets de :

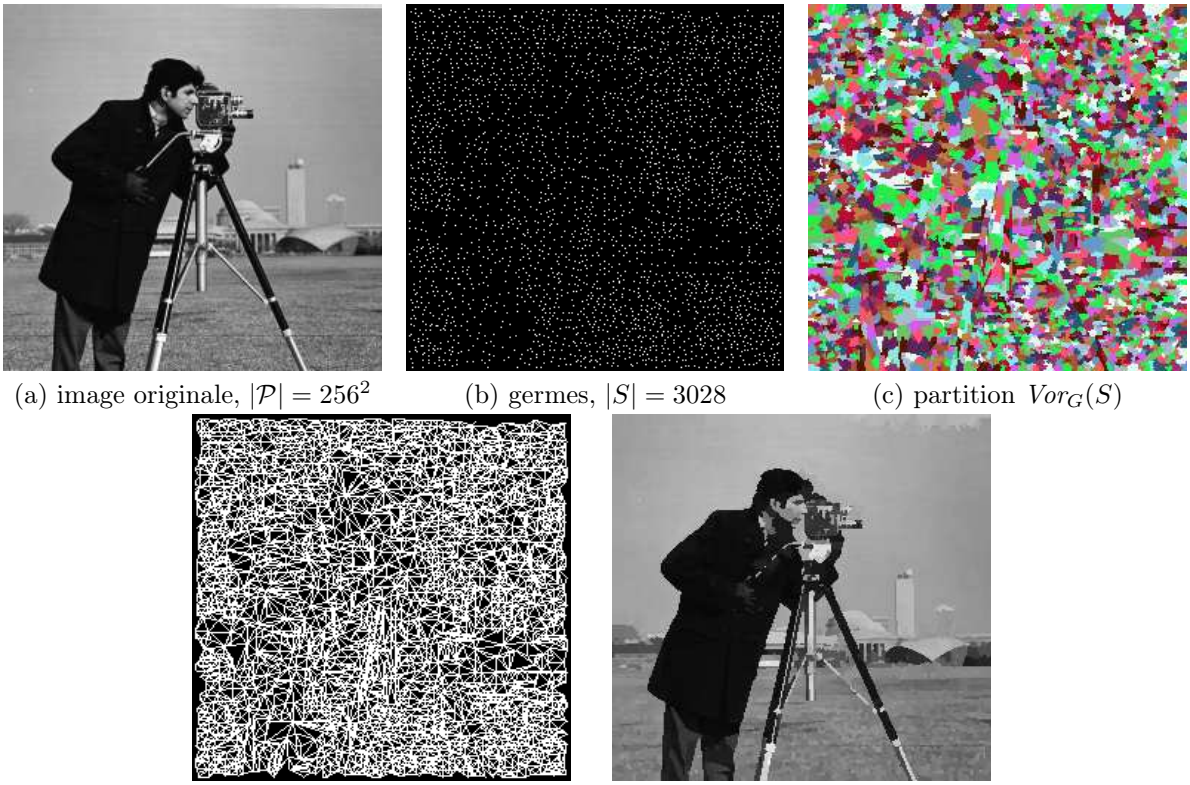
$$Z_G(s_i, S) = \{u \in V \ : \ \rho_G(s_i, u) \leq \rho_G(s_j, u), \forall s_j \in S\}. \tag{5.13}$$

La **partition d'énergie** de G, relativement à la métrique ρ_G et à l'ensemble des sources S, est l'ensemble des zones d'influences des sommets de S :

$$Vor_G(S) = \{Z_G(s_i, S), \forall s_i \in S\}. \tag{5.14}$$

La partition d'énergie peut être vue comme un diagramme de Voronoï généralisé généré à partir de S et de la métrique ρ_G. Deux sommets sources s_i et s_j de S sont **voisins** dans la partition $Vor_G(S)$ si et seulement si il existe une arête de G qui connecte un sommet de Z_i et un sommet de Z_j. Le dual combinatoire de la partition $Vor_G(S)$ est le RAG formé en connectant les sommets sources voisins.

Les germes sont choisis parmi l'ensemble des extrema de l'image initiale I. La figure 5.11 présente les étapes du calcul de la partition d'énergie d'une image $I : \mathcal{P} \to \mathbb{R}$. Cette partition est formée d'un nombre de régions qui est significativement moins important que le nombre de pixels de l'image originale. Le taux de réduction de l'information dépend du nombre de germes utilisés pour construire la partition. Dans nos expérimentations, nous avons observé qu'il était au minimum de 80%. On peut observer que le contenu de la partition des valeurs moyennes (figure 5.11(e)) et celui de l'image originale (figure5.11) sont proches visuellement, même si ce taux de réduction est important. L'ensemble des germes (figure 5.11(b)), munis de la valeur moyenne de leur zone d'influence, représente



(a) image originale, $|\mathcal{P}| = 256^2$ (b) germes, $|S| = 3028$ (c) partition $Vor_G(S)$

(d) RAG associé à (c) (e) partition des valeurs moyennes

Fig. 5.11: Simplification par partition (réduction de 95% de l'information).

l'image originale compressée. Une approximation de l'image originale peut être ensuite calculée en accomplissant une interpolation des valeurs moyennes associées aux germes, sur le domaine original $\mathcal{P}$. Indépendamment, les partition d'énergie peut être simplifiée en régularisant et fusionnant le RAG qui y est associé.

Régularisation de RAG. Soit $I : \mathcal{P} \subset \mathbb{Z}^2 \to \mathbb{R}^m$ une image, représentée par un graphe $G^* = (\mathcal{P}, E^*)$ en 8-adjacence (graphe G_1). Soit Z une partition fine de I obtenue avec la méthode de partition d'énergie précédente, ou avec une autre méthode de pré-segmentation. Soit $G = (S, E, w)$ le RAG associé à la partition Z. Nous proposons de simplifier la partition en régularisant, sur le RAG, la fonction $f^0 : S \to \mathbb{R}^m$ définie par :

$$f^0(s) = \frac{\sum_{p \in Z(s)} I(p)}{\sum_{p \in Z(s)} 1}, \qquad \forall s \in S, \tag{5.15}$$

où $Z(s)$ est la région de la partition associée au sommet s. Rappelons qu'une itération du processus de diffusion isotrope discrète est donnée, en chaque sommet $s \in S$, par :

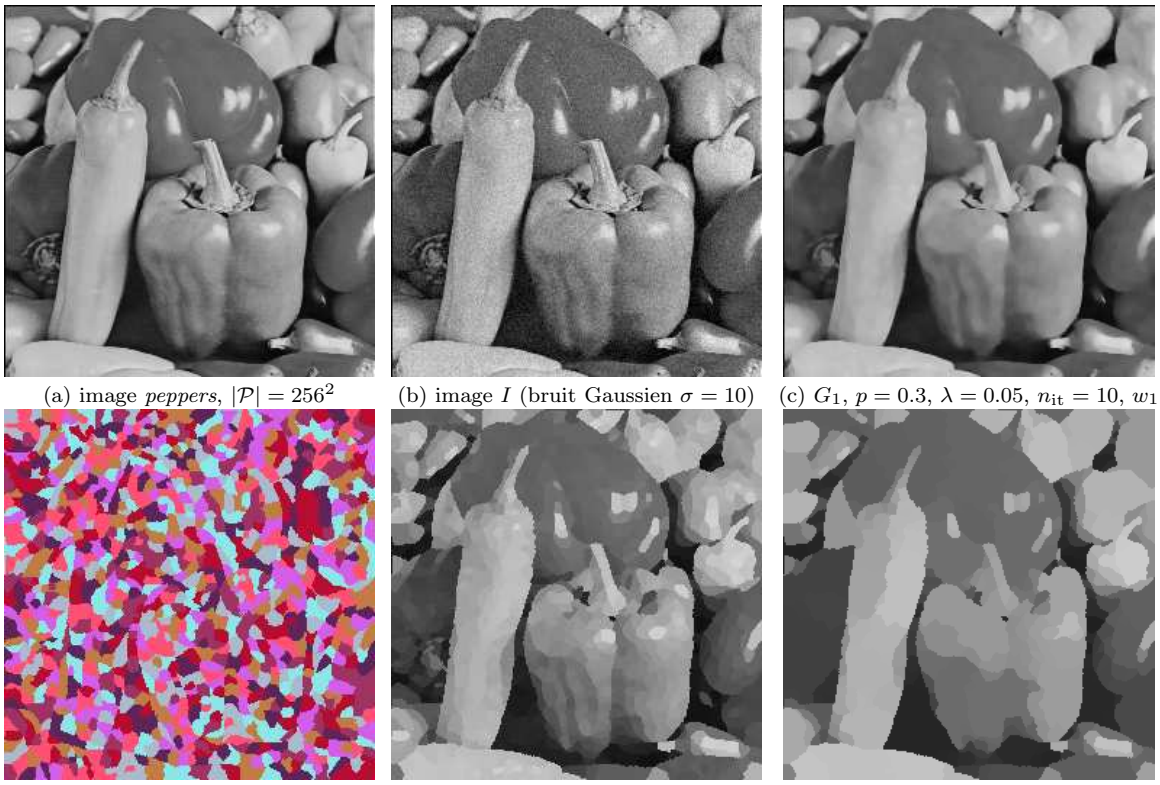
$$f^{(t+1)}(s) = \frac{\lambda f^0(s) + \sum_{v \sim s} w_{sv} (\| \bigtriangledown_s f^{(t)} \|_2^{p-2} + \| \bigtriangledown_v f^{(t)} \|_2^{p-2}) \, f^{(t)}(v)}{\lambda + \sum_{v \sim s} w_{sv} (\| \bigtriangledown_s f^{(t)} \|_2^{p-2} + \| \bigtriangledown_v f^{(t)} \|_2^{p-2})}. \tag{5.16}$$

Le choix de la fonction de poids peut dépendre de la façon dont la partition a été obtenue. Pour des partitions d'énergie, la ligne de partage des eaux, ou toute autre méthode basée sur la diffusion à partir de germes, le poids d'une arête est la distance géodésique entre ses sommets, sur l'image originale I : $w_{uv} = 1/\rho_{G^*}(u, v)$. Dans ce cas, le poids d'une arête est d'autant plus élevé que les régions, associées à ses sommets, appartiennent à deux objets différents dans l'image I. Le filtrage tient compte de la géométrie de l'image I sur le graphe G^*, relativement à la distance ρ_{G^*}.

Pour choisir la fonction de poids, une méthode plus générale consiste à traiter uniquement l'information attachée aux sommets du RAG. Ceci permet de régulariser les partitions qui ne sont pas issues d'une diffusion, c.-à-d. les RAG dont les sommets ne sont pas plongés dans $\mathcal{P}$. Comme les partitions correspondent à des versions déjà simplifiées de l'image originale (constante par morceaux), il est souvent suffisant de considérer le cas non-pondéré, avec $p \leq 1$ (agissant lui-même comme une pondération, mais adaptative). Néanmoins, dans nos expérimentations nous avons constatés que la fonction de poids :

$$w_{u,v} = \frac{\sigma}{\|f^0(u) - f^0(v)\|_2 + \epsilon}, \qquad \forall (u, v) \in E, \tag{5.17}$$

donne de bons résultats. La figure 5.12 illustre la régularisation de la partition d'une image $I : \mathcal{P} \to \mathbb{R}$ sur le RAG qui y est associé (figure 5.12(f)).



(a) image *peppers*, $|\mathcal{P}| = 256^2$ (b) image I (bruit Gaussien $\sigma = 10$) (c) G_1, $p = 0.3$, $\lambda = 0.05$, $n_{\text{it}} = 10$, w_1

(d) partition coloriée (e) partition des valeurs moyennes (f) $p = 0.3$, $\lambda = 0.05$, $n_{\text{it}} = 30$, w^4

Fig. 5.12: Régularisation isotrope d'une partition sur un RAG avec $p = 0.3$.

5.2 Application au traitement de courbes et de surfaces polygonales

Les courbes et les surfaces polygonales ont par nature une structure de graphe. Ceci permet de les traiter avec la régularisation discrète proposée. Soit $M \subset \mathbb{R}^m$ une courbe ou une surface polygonale constituée d'un ensemble de sommets $V \subset \mathbb{R}^m$, d'un ensemble d'arêtes E, et d'un ensemble de faces F (dans le cas d'une surface). Globalement, les mêmes traitements que ceux décrits dans le contexte du traitement d'images peuvent être effectués, sur la position des sommets de M, aussi bien que sur toute fonction définie sur les sommets, les arêtes ou les faces de M. Cette section montre des applications de la régularisation isotrope discrète au lissage, au débruitage et à la simplification d'une fonction $f^0 : V \to \mathbb{R}^n$, en utilisant le processus de diffusion discrète :

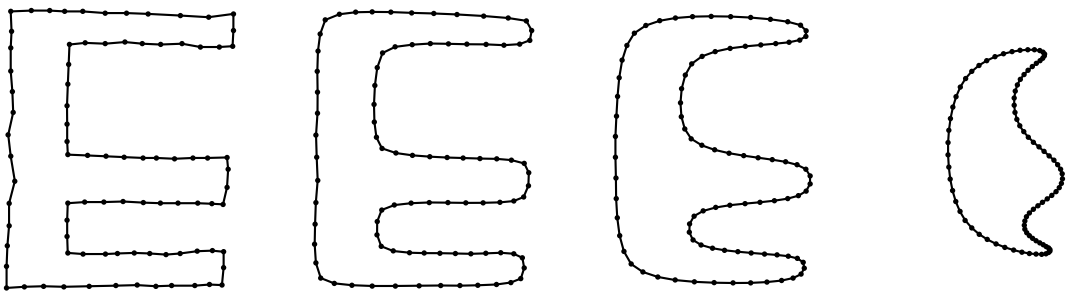
$$\begin{cases} f^{(0)} = f^0, \\ f^{(t+1)}(u) = \dfrac{\lambda f^0(u) + \sum_{v \sim u} w_{uv}(\| \nabla_w f^{(t)}(u)\|_2^{p-2} + \| \nabla_w f^{(t)}(v)\|_2^{p-2})\, f^{(t)}(v)}{\lambda + \sum_{v \sim u} w_{uv}(\| \nabla_w f^{(t)}(u)\|_2^{p-2} + \| \nabla_w f^{(t)}(v)\|_2^{p-2})}, \end{cases} \tag{5.18}$$

où $f : V \to \mathbb{R}^n$ est une fonction définie sur les sommets de la courbe ou de la surface polygonale.

5.2.1 Lissage et débruitage

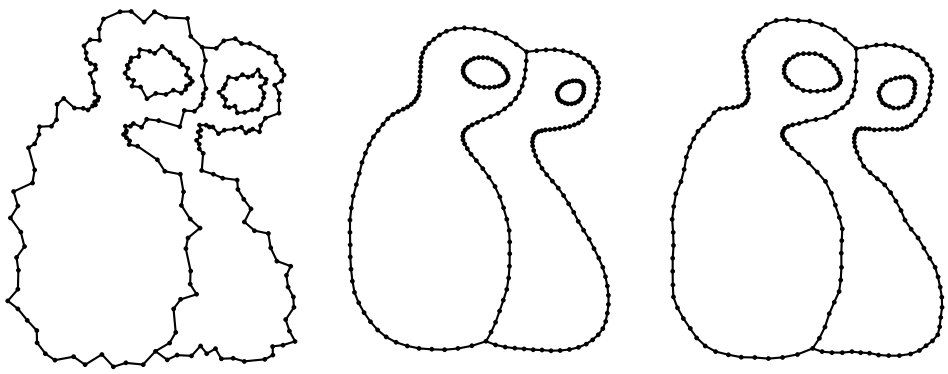
La courbe ou la surface polygonale M peut être régularisée en diffusant la position de ses sommets sur le graphe $G := (V, E, w)$. Ceci revient à la faire évoluer dans $\mathbb{R}^m$. Pour cela, le processus de diffusion discrète (5.18) est utilisé avec la fonction position $f^0 : V \subset \mathbb{R}^m \to \mathbb{R}^m$ telle que $f^0(u) = u$ pour tout $u \in V$. Le résultat est la nouvelle position $f : V \to \mathbb{R}^m$ des sommets.

Selon le choix de la fonction de poids w et du paramètre de régularité, la régularisation décrit plusieurs flots géométriques qui peuvent être utilisés pour faire évoluer, lisser ou débruiter M. Les poids peuvent être choisis de nombreuses façons, selon le traitement à effectué et la nature de M.



(a) $f^0 : V \subset \mathbb{R}^2 \to \mathbb{R}^2$ (b) $\lambda = 1$ (c) $\lambda = 0.2$ (d) $\lambda = 0.02$

Fig. 5.13: Lissage d'une courbe polygonale par régularisation isotrope locale, pour $p = 2$, $w = 1$, et différentes valeurs de λ, jusqu'à convergence.



$f^0 : V \to \mathbb{R}^2$ $\lambda = 0.05$ $\lambda = 0.15$

Fig. 5.14: Débruitage d'une courbe polygonale par régularisation avec $p = 2$ et $w = 1$.

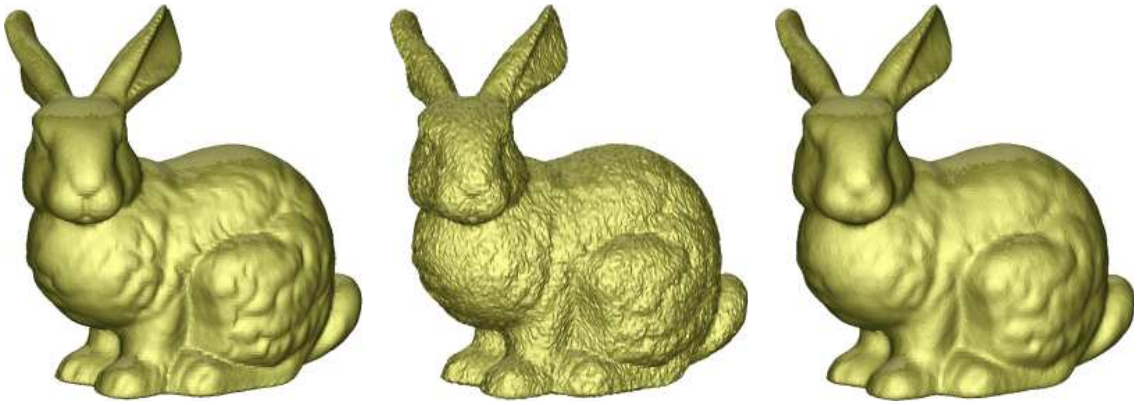
La figure 5.13 montre l'évolution d'une courbe polygonale fermée pour $p = 2$ et $w = 1$ (par diffusion discrète selon l'opérateur Laplacien non-pondéré). On peut apprécier l'effet de la régularisation selon la valeur du paramètre d'approximation $\lambda \neq 0$ (le processus de diffusion est itéré jusqu'à la convergence).

Pour le débruitage le terme d'approximation est très important à prendre en compte. Pour bien préserver la géométrie des courbes tout en lissant, il faut un bon compromis entre le terme d'approximation et le terme de régularisation. Les méthodes de débruitage classiques de courbes ou de surfaces nécessitent de choisir un nombre d'itérations, ainsi que de recalculer le poids des arêtes à chaque itération [Taubin, 1995, Taubin, 2000]. Dans notre méthode, le seul paramètre est relatif au compromis entre l'approximation et la régularisation. Le débruitage d'une courbe polygonale est illustré à la figure 5.14, avec deux exemples de ce paramètre. De même que pour les courbes polygonales, le débruitage de surfaces est illustré à la figure 5.15. Le poids considéré ici est le noyau Gaussien :

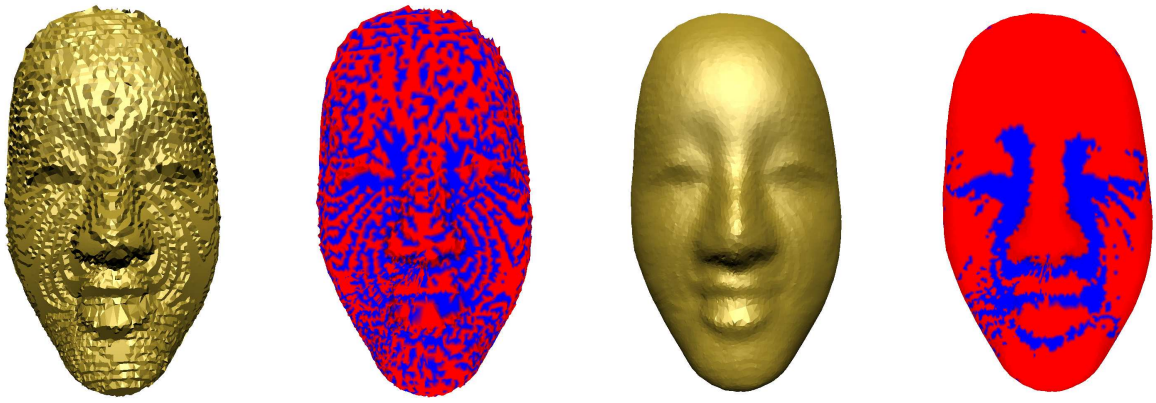
$$w(u, v) = \exp\left(-\frac{\|f^0(u) - f^0(v)\|_2^2}{\sigma^2}\right).$$

La régularisation de surfaces trouve son utilité lorsqu'on a à estimer des caractéristiques géométriques qui dépendent des dérivées d'ordre un ou deux, qui sont par nature sensibles au bruit. La deuxième ligne de la figure 5.15 illustre le calcul de la courbure moyenne sur une surface polygonale bruitée et sur une version régularisée.

Une autre caractéristique usuelle pour traiter les courbes ou les surfaces est le squelette (voir Partie I). Le squelette est très sensible au bruit. La figure 5.16 illustre l'effet de la régularisation sur la simplification du squelette d'une courbe polygonale bruitée. La régularisation est effectuée sur les sommets de la courbe, et le squelette est ici calculé à partir du diagramme de Voronoï (Euclidien) des sommets de la courbe.



surface originale $f^0 : V \subset \mathbb{R}^3 \to \mathbb{R}^3$ (bruit normal) $p = 2$, $\lambda = 0.5$



$f^0 : V \subset \mathbb{R}^3 \to \mathbb{R}^3$ courbure moyenne $p = 2$, $\lambda = 0.5$ courbure moyenne

Fig. 5.15: Débruitage de surfaces triangulaires par régularisation de la position des sommets avec $p = 2$.

$f^0 : V \subset \mathbb{R}^2 \to \mathbb{R}^2$ squelette $p = 2$, $\lambda = 0.2$

Fig. 5.16: Simplification du squelette d'une courbe polygonale. De gauche à droite : la courbe polygonale initiale, son squelette, et une version simplifiée du squelette par régularisation de f^0 avec $p = 2$ et $w = 1$.

5.2.2 Simplification

Dans le cadre du traitement d'images, lorsque $p < 1$, nous avons vu que la régularisation a pour effet de lisser l'image tout en préservant les contours. De plus, lorsque $p \to 0$, l'image tend à devenir constante par morceaux. La régularisation agit alors comme une méthode de regroupement. On retrouve le même comportement dans le cadre des courbes et des surfaces polygonales, que nous proposons comme processus de simplification.

Ceci est illustré à la figure 5.17 sur une courbe polygonale dont une partie est faiblement bruitée. On peut observer qu'au cours des itérations, les sommets se rapprochent les uns des autres pour former des groupements géométriques. Remarquons de plus que les groupements semblent préserver les discontinuités, ainsi que les sommets de forte courbure. Les graphes G' et G'' (deuxième ligne) fournissent une version simplifiée de la courbe initiale G. La simplification peut être complétée en détectant les groupements et en remplaçant chacun d'entre-eux par le sommet du graphe initial qui s'est le moins déplacé.

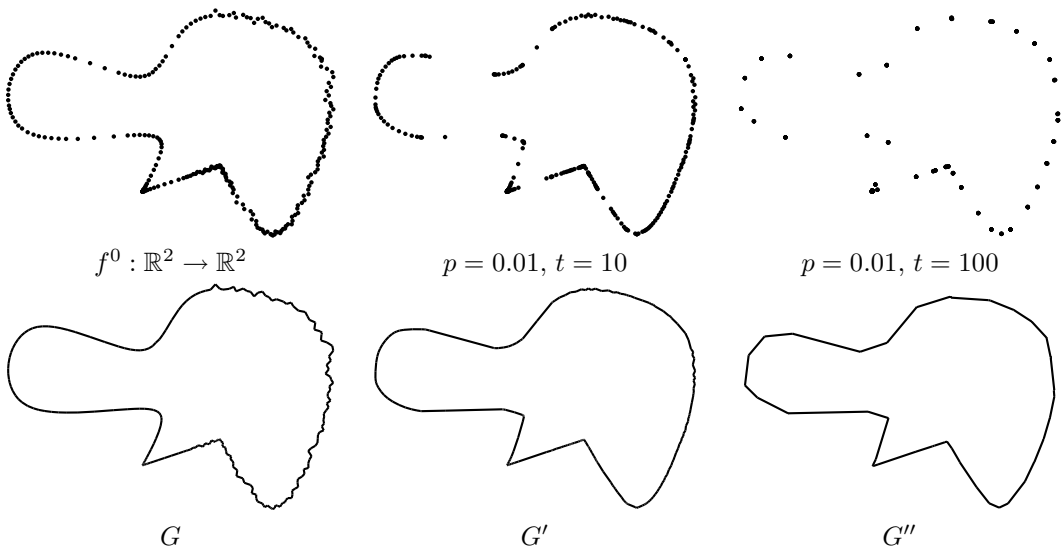


Fig. 5.17: Simplification d'une courbe polygonale par régularisation de la position des ses sommets et $p < 1$. Première ligne : les sommets des polygones représentés à la deuxième ligne.

La régularisation a le même comportement lorsque la fonction à traiter est la position des sommets d'une surface polygonale. La figure 5.18 montre la simplification d'une surface triangulaire avec $p = 0.001$. Le résultat est comparé avec celui obtenu avec $p = 1$. On peut observer que dans ce dernier cas, la surface est lissée tout en préservant son aspect principal. En revanche, pour $p = 0.001$, les sommets se regroupent à des positions spécifiques. Comme dans le cas des courbes, c'est la fonction qui est ici visualisée. Le processus de simplification peut être complété en remplaçant chaque groupement par le sommet de la surface initiale qui s'est le moins déplacé au cours de la régularisation. L'effet de simplification de la fonction position est illustré sur d'autres courbes et surfaces, aux figures 5.19 et 5.20, pour différentes valeurs de p et de λ. De nouveau, nous pouvons

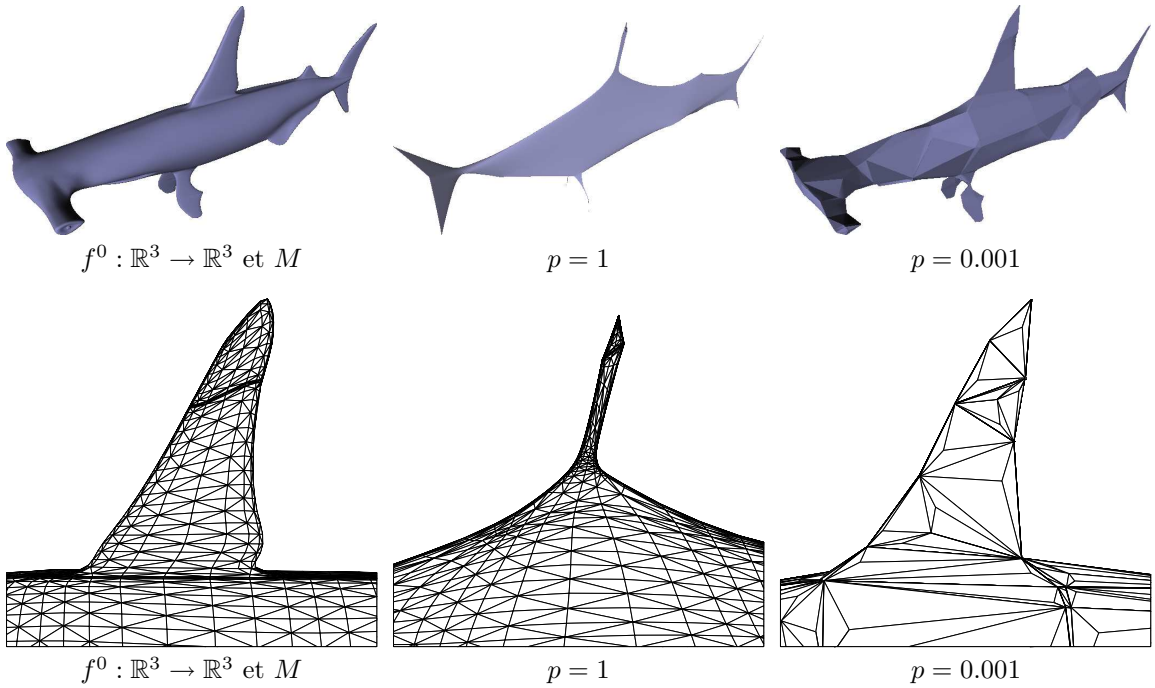


Fig. 5.18: Simplification d'une courbe polygonale par régularisation de la position des ses sommets et $p < 1$. Première ligne : les sommets des polygones représentés à la deuxième ligne.

observer l'effet de regroupement constaté précédemment, lorsque p tend vers zéro.

La régularisation isotrope discrète offre donc un nouveau moyen efficace de simplification de courbes et de surfaces polygonales. L'une de nos perspectives est d'employer le même principe pour décomposer ces données en structures géométriques élémentaires. Dans la suite, nous montrons que la régularisation s'applique également au lissage et à la simplification de données discrètes non-organisées.

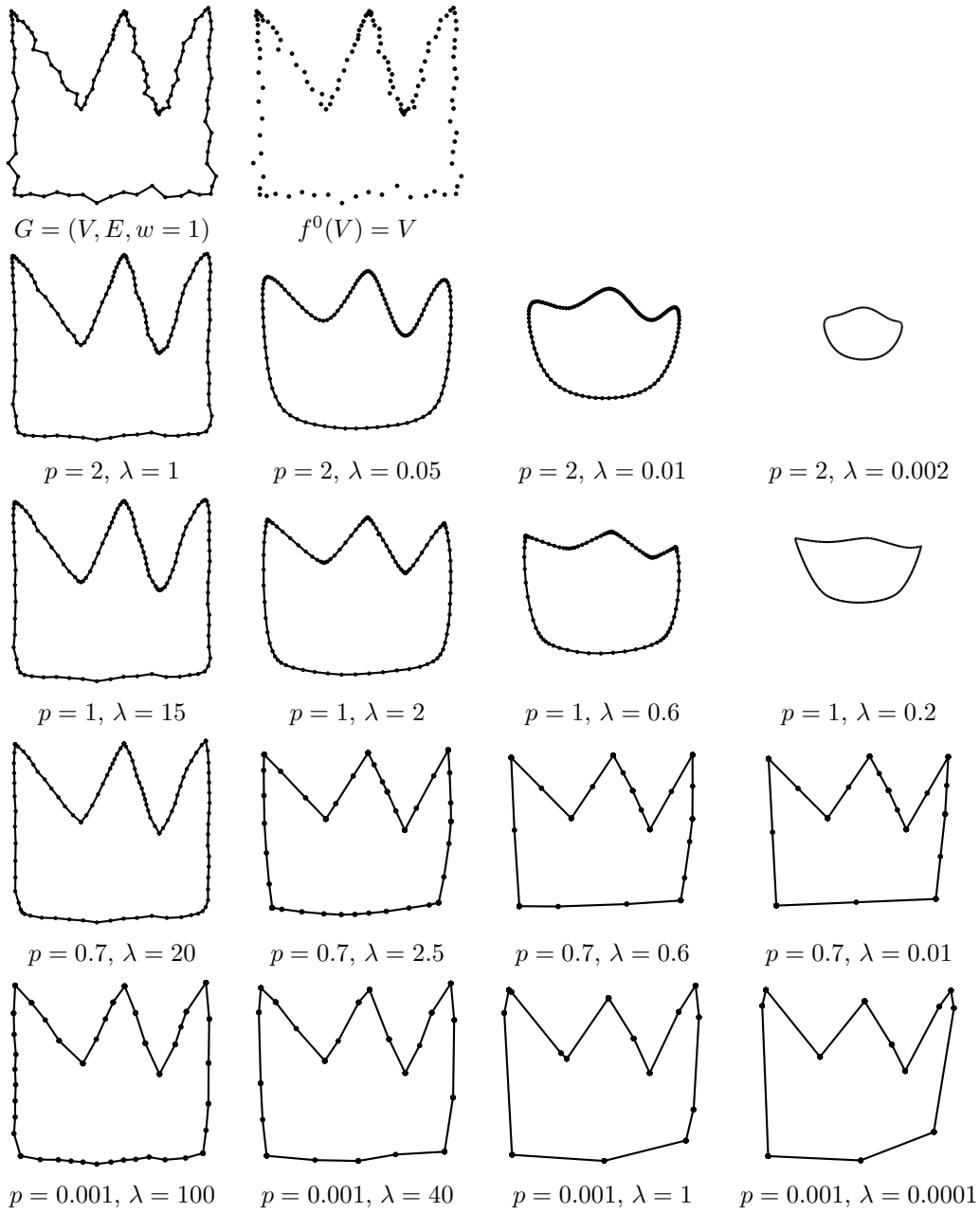


Fig. 5.19: Comportement de la régularisation isotrope d'une courbe polygonale G (première ligne), en utilisant le processus de diffusion discrète (jusqu'à la convergence). Chaque ligne correspond à une valeur de p. Pour chaque p, les différentes régularisations sont obtenues avec des valeurs décroissantes du paramètre λ.

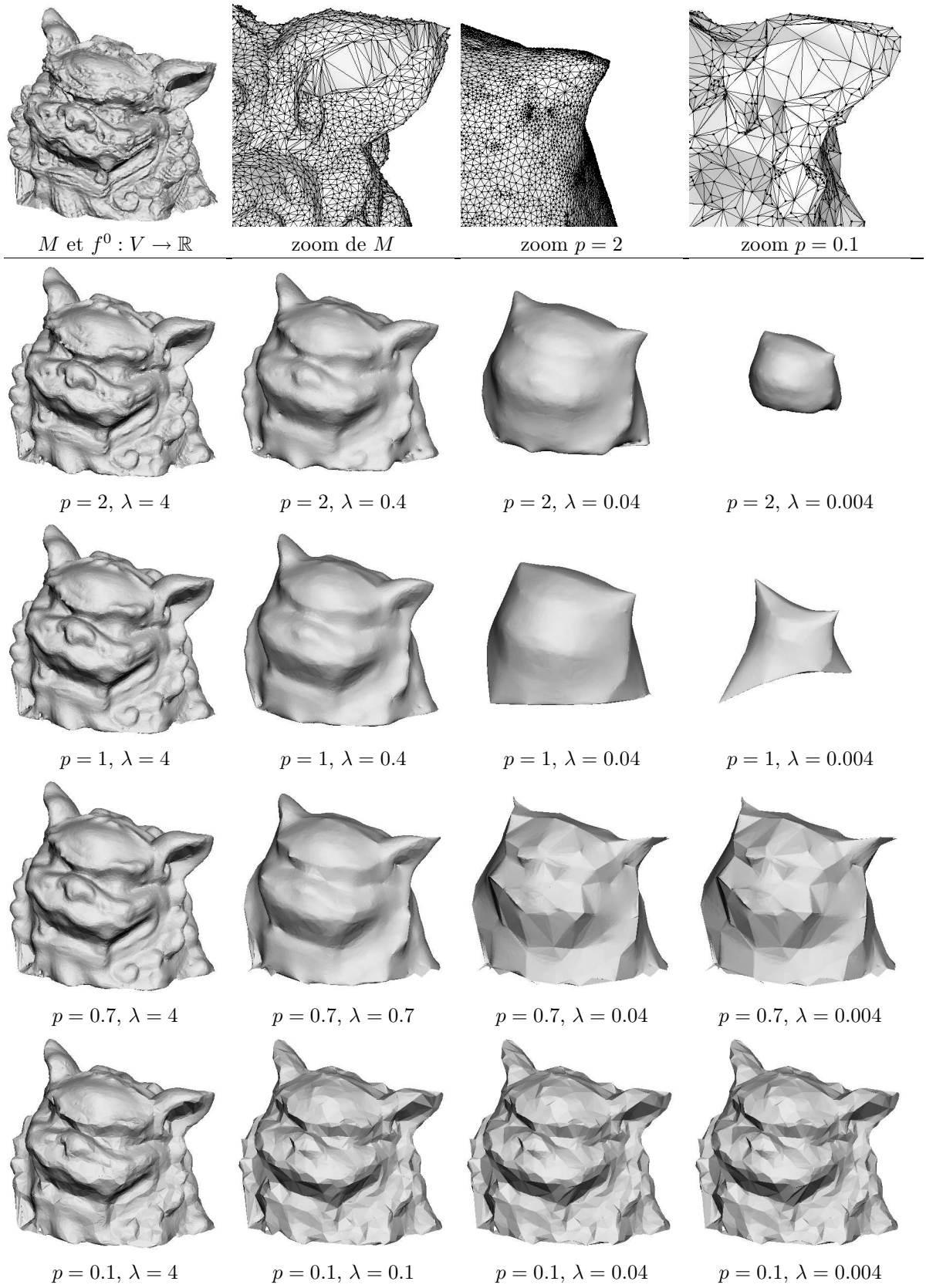


Fig. 5.20: Comportement de la régularisation isotrope d'une surface polygonale M (première ligne), en utilisant le processus de diffusion discrète (jusqu'à la convergence). Chaque ligne correspond à une valeur de p. Pour chaque p, les différentes régularisations sont obtenues avec des valeurs décroissantes du paramètre λ. Les agrandissement à la première ligne correspondent à $\lambda = 0.04$.

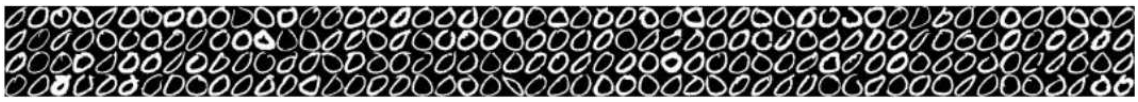
5.3 Régularisation de données discrètes non-organisées

Dans cette section, nous appliquons la régularisation isotrope discrète au traitement de données non-organisées définies dans des espaces de grande dimension. Soit $X := \{x_1, \ldots, x_N\}$ un ensemble fini de données $x_i \in \mathbb{R}^m$. Pour régulariser X, nous considérons ici le graphe complet $G_\infty := (V, E, w)$ où chaque sommet de V correspond à une donnée $x_i \in X$, et est connecté à tous les autres sommets de V. Le graphe est pondéré avec le noyau Gaussien :

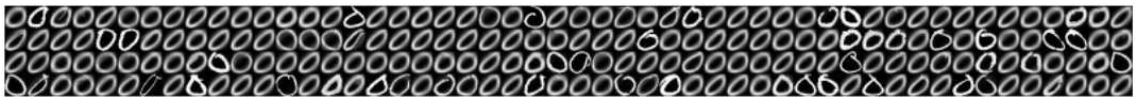
$$w_{uv} = \exp\left(-\frac{\|f^0(u) - f^0(v)\|_2^2}{\sigma^2}\right),$$

où $f^0 : V \to \mathbb{R}^m$ est la fonction à régulariser. Pour calculer la régularisation, qui est non-locale par la structure du graphe, le processus de diffusion isotrope discrète est utilisé.

La figure 5.21 illustre la régularisation d'une base d'images de chiffres manuscrits : base *United States Postal Service* (USPS) du chiffre 0. Chaque occurrence du chiffre 0 est une image de 16×16 pixels, représentée par un vecteur de 256 dimensions. L'ensemble des images est structuré par le graphe complet $G_\infty := (V, E, w)$ tel que chaque image de la base corresponde à un sommet de V. La fonction à régulariser est alors la fonction $f^0 : V \to \mathbb{R}^{256}$. On peut observer l'effet de la régularisation. Plus λ est faible, plus les images régularisées convergent vers une même image moyenne (figure 5.21(c)). La diffusion est ici itérée jusqu'à la convergence.



(a) base USPS de l'image manuscrite du chiffre 0

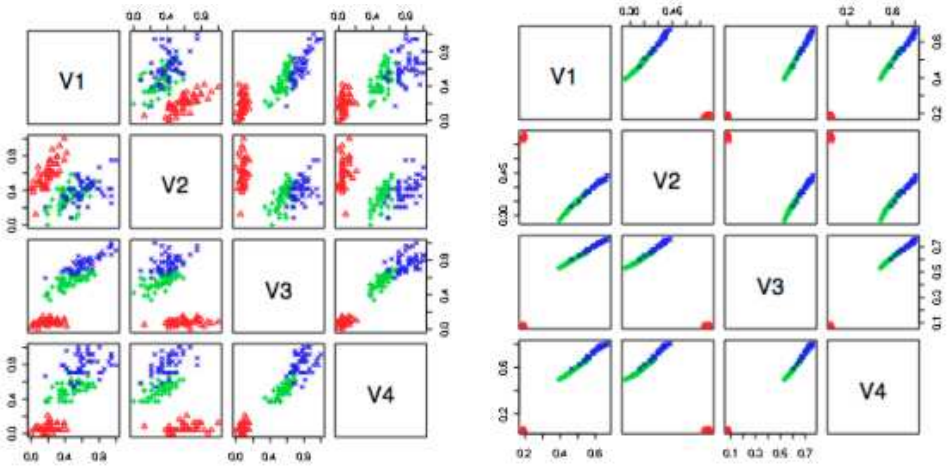


(b) Régularisation non-locale avec $\lambda = 0.01$

(c) Régularisation non-locale avec $\lambda = 0$

Fig. 5.21: Régularisation isotrope non-locale d'une base de données d'images avec $p = 2$ (base USPS).

Un autre exemple de régularisation non-locale de données est illustré à la figure 5.22. L'ensemble de données est la base *Iris* [Hettich *et al.*, 1998], qui contient 3 classes de 50 éléments. Chaque élément est un vecteur de $\mathbb{R}^4$. La figure 5.22(a) présente la projection cette base dans le plan. Les données sont naturellement bruitées (données réelles). Elles sont représentées par la fonction $f^0 : V \to \mathbb{R}^4$ et régularisées sur le graphe complet $G_\infty := (V, E, w)$ où chaque sommet de V est une donnée. La figure 5.22(b) présente le résultat de la régularisation de f^0 pour $p = 2$. On peut observer qu'elle agit comme un processus de regroupement. Ceci à l'avantage de faciliter la classification ou plus généralement l'analyse ultérieure des données.



(a) $f^0 : V \to \mathbb{R}^4$ (b) $p = 2$, $\lambda = 0.01$, $n_{it} = 10$

Fig. 5.22: Régularisation isotrope non-locale de la base de données *Iris*.

5.4 Conclusion et perspectives

Dans ce chapitre nous avons montré le comportement de la famille de régularisations sur graphes pondérés pour traiter des images, des courbes et des surfaces polygonales, ainsi que des données discrètes non-organisées. En particulier, nous avons noté l'importance du paramètre de régularité p qui intervient dans le modèle isotrope. Pour $p \leq 1$, la régularisation isotrope lisse les données tout en préservant leurs discontinuités. De plus, lorsque $p \to 0$, elle agit comme une méthode de regroupement. Le choix de la structure du graphe et de la fonction de poids est aussi déterminante lorsque l'on veut préserver des structures répétitives. Nous avons appliqué la régularisation à des problématiques de lissage, de débruitage et de simplification. Les mêmes méthodes peuvent être employées dans des processus d'interpolation, de synthèse, de segmentation ou de classification. Dans le contexte du traitement de maillages, une application envisagée est la segmentation par régularisation multi-échelles, où une échelle correspond à une valeur du paramètre d'approximation λ. D'autre part, nous avons montré le comportement de la régularisation anisotrope que dans le contexte du traitement d'images. Il serait intéressant de l'étudier dans le cas général de données discrètes non-organisées pour réaliser des processus de classification semi-supervisée.

Publications associées à la Partie II

[Ta *et al.*, 2007] V. T. Ta, S. Bougleux, A. Elmoataz et O. Lézoray.
Nonlocal anisotropic discrete regularization for image, data filtering and clustering. *Rapport technique, Université de Caen Basse-Normandie, hal–00187165*, soumis à Computer Vision and Image Understanding, Special Issue on Discrete Optimization in Computer Vision, Novembre 2007.

[Bougleux *et al.*, 2007b] S. Bougleux, A. Elmoataz et M. Melkemi.
Local and Nonlocal Discrete Regularization on Weighted Graphs for Image and Mesh Processing. *Article invité (édition spéciale de la Conférence SSVM'07). International Journal of Computer Vision*, soumis, 2007.

[Elmoataz *et al.*, 2007] A. Elmoataz, O. Lézoray et S. Bougleux.
Nonlocal discrete regularization on weighted graphs : a framework for image and manifold processing. *Rapport technique, Université de Caen Basse-Normandie, hal–00163573*, soumis à IEEE Transactions on Image Processing, juillet 2007.

[Lézoray *et al.*, 2007a] O. Lézoray, A. Elmoataz et S. Bougleux.
Graph regularization for color image processing. *Special issue on color image processing, Computer Vision and Image Understanding*, Vol. 107(1–2), édité par R. Lukac, K. N. Plataniotis et A. N. Venetsanopoulos, pages 38–55, Elsevier, 2007.

[Lézoray *et al.*, 2007b] O. Lézoray, S. Bougleux et A. Elmoataz.
Nonparametric discrete regularization on graphs for color image filtering. *4th International Conference on Image Analysis and Recognition.* LNCS Vol. 4633, édité par M. S. Kamel et A. Campilho, pages 46–57, Springer, 2007.

[Bougleux *et al.*, 2007a] S. Bougleux, A. Elmoataz et M. Melkemi.
Prix du meilleur article étudiant.
Discrete Regularization on Weighted Graphs for Image and Mesh Filtering. *1st International Conference on Scale Space and Variational Methods in Computer Vision.* LNCS Vol. 4485, édité par F. Sgallari, A. Murli et N. Paragios, pages 128–139, Springer, 2007.

[Lézoray *et al.*, 2006] O. Lézoray, A. Elmoataz et S. Bougleux.
Régularisation sur graphe pour le traitement d'images couleur. *10ème Conférence Compression et Représentation des Signaux Audiovisuels*, pages 236-241, 2006.

[Bougleux et Elmoataz, 2005] S. Bougleux et A. Elmoataz.
Image Smoothing and Segmentation by Graph Regularization. *1st International Symposium on Visual Computing.* LNCS Vol. 3804, édité par G. Bebis, R. Boyle, D. Koracin et B. Parvin, pages 745-752, Springer, 2005.

Conclusion et Perspectives Générales

Dans cette thèse, nous avons apporté plusieurs contributions pour structurer et traiter des données discrètes, organisées ou non. Dans le contexte de la structuration, nous avons considéré le problème de la reconstruction de formes géométriques à partir d'un ensemble de points, avec des techniques issues de la géométrie algorithmique. Dans un premier temps, nous avons traité le problème de la reconstruction de formes globales à partir d'échantillons de courbes 2D. Puis, nous avons étudié celui de la reconstruction de primitives géométriques données, à partir d'un ensemble de points 3D. Dans le contexte du traitement de données, nous avons considéré le problème de la régularisation discrète, pour effectuer des processus de filtrage et de simplification. Ce dernier problème est traité par des méthodes variationnelles sur graphes pondérés.

Pour reconstruire des formes globales à partir d'un ensemble de points 2D, nous avons proposé une famille hiérarchique de graphes de voisinage, inclus dans le graphe de Gabriel. Les graphes de cette famille, que nous avons appelés les β-Crusts locaux, sont générés à partir d'une relation géométrique entre les arêtes du diagramme de Voronoï et celles de la triangulation de Delaunay. En nous basant sur les propriétés hiérarchiques de ces graphes, et sur les propriétés topologiques des formes à reconstruire, nous avons également proposé un algorithme glouton pour extraire des courbes simples (TopoCrust). Simultanément à la reconstruction, l'axe médian des courbes reconstruites est également extrait par l'algorithme, comme un sous-ensemble des arêtes du diagramme de Voronoï. À défaut de proposer une garantie de reconstruction avec les β-Crusts locaux, et l'algorithme TopoCrust, nous avons comparé leur efficacité avec les méthodes classiques de reconstruction de courbes simples. La comparaison est réalisée au sein d'une étude expérimentale sur le lien entre l'ϵ-échantillonnage et la reconstruction. Le principal avantage de la méthode proposée est de supporter des échantillons de faible densité, ou de densité très variable, que les méthodes existantes traitent difficilement.

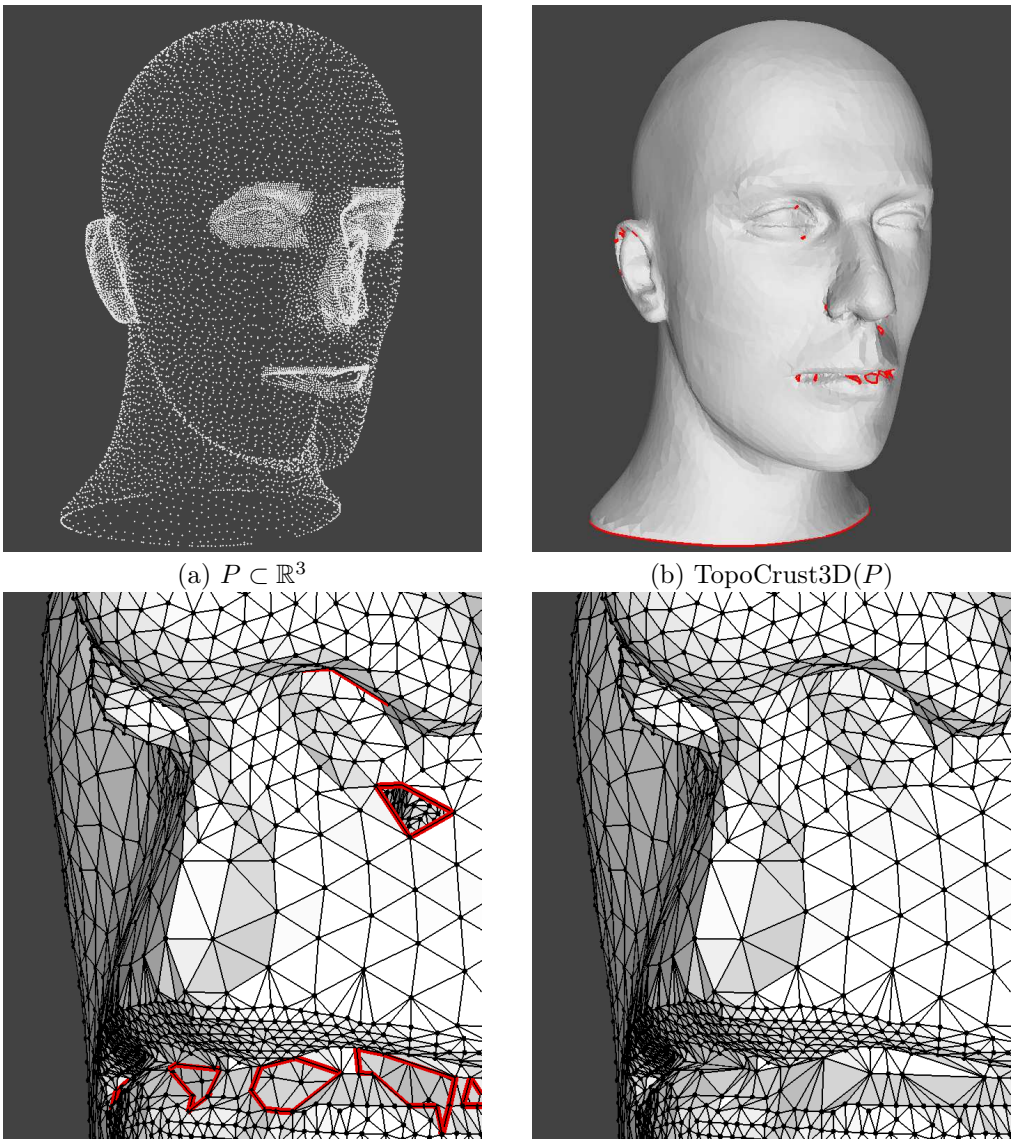
Une alternative à la reconstruction de formes globales est la détection de formes spécifiques par des méthodes de reconstruction. Dans ce contexte, nous avons montré que les α-formes sont capables de détecter des structures linéaires dans des ensembles de points non-structurés, comme par exemple des droites et des plans. Pour cela, nous avons étendu la propriété de boule vide, qui intervient dans la définition des α-formes, par la propriété de boule ellipsoïdale vide. Ceci nous a permis de définir une famille d'α-formes anisotropes, paramétrées par l'orientation et la forme des boules ellipsoïdales. Par analogie avec les α-formes, chaque α-forme anisotrope est liée à un diagramme affine et à sa triangulation duale, qui se déduisent du diagramme de Voronoï de l'ensemble de points (respectivement de la triangulation de Delaunay) par une transformation affine. Les α-formes anisotropes peuvent ainsi se calculer directement à partir de la triangu-

lation de Delaunay de l'ensemble de points. Lorsque la forme des boules ellipsoïdales est suffisamment fine, les α-formes anisotropes détectent des portions de droites ou de plans, dans une direction donnée. Plusieurs structures non-significatives peuvent également être reconstruites par les α-formes anisotropes. Pour les supprimer, nous avons proposé un mécanisme général basé sur une analyse topologique et combinatoire des structures détectées. Enfin, la détection de structures polygonales plus complexes est effectuée en fusionnant les résultats obtenus pour différentes orientations des boules ellipsoïdales. Les principaux avantages de l'approche proposée sont : la connection de points qui ne sont pas nécessairement voisins dans la triangulation de Delaunay, et la détection d'échantillons de structures linéaires dans un milieu bruité. Cette méthode peut venir en complément des approches classiques de détection de formes, basées sur la transformation de Hough et de ses variantes, qui ne sont pas toujours adaptées à des données non-structurées.

Dans le contexte du traitement de données discrètes, nous avons proposé des modèles variationnels pour régulariser des fonctions définies sur des graphes pondérés de topologie arbitraire. Ces modèles font intervenir une fonctionnelle de régularisation et une fonctionnelle d'approximation. Nous avons proposé deux types de fonctionnelle de régularisation, l'une est basée sur la norme $\mathcal{L}_2$ du gradient discret, et l'autre sur la norme $\mathcal{L}_p$ de ce même gradient. Nous avons montré le lien avec les fonctionnelles de régularisation locale et non-locale continues, principalement utilisées en traitement d'images. Contrairement à ces méthodes de régularisation continue, qui mènent à des EDP de diffusion, la minimisation des fonctionnelles discrètes proposées mène à des familles de processus de diffusion, basés sur le p-Laplacien discret isotrope ou anisotrope. Ces processus sont paramétrés par le degré de régularité p, la quantité d'approximation, et la structure du graphe (topologie et géométrie). Ceci permet de traiter tout ensemble de données discrètes, ou toute fonction définie sur ces données. En particulier, nous avons étudié le comportement de la régularisation isotrope sur des images, et sur des courbes et des surfaces polygonales. Nous avons pu constater que pour $p \leq 1$, la régularisation lisse la fonction à traiter, tout en préservant ses discontinuités principales. De plus, lorsque p tend vers zéro, elle agit comme un processus de regroupement ou de facettisation (dans le cas des images). La méthodologie proposée offre donc un cadre général pour filtrer et simplifier des fonctions discrètes. Le même formalisme peut être utilisé pour concevoir des procédures d'interpolation, de synthèse ou de segmentation.

Perspectives. Le travail proposé dans cette thèse offre plusieurs perspectives :

– Concernant la reconstruction de courbes à partir d'un ensemble de points, nous avons constaté expérimentalement que le paramètre β des β-Crusts locaux, est lié au paramètre ϵ qui intervient dans la condition d'ϵ-échantillonnage. L'une des perspectives est d'établir le lien théorique entre ces deux paramètres. D'autre part, la définition des β-Crusts locaux, et l'algorithme TopoCrust, s'étendent naturellement pour reconstruire des surfaces à partir d'ensembles de points de l'espace Euclidien de dimension trois. Cependant, comme illustré à la figure 23 et contrairement à la reconstruction de courbes, la surface reconstruite comporte généralement des trous, même lorsque l'ensemble de points est un échantillon dense d'une surface régulière. Pour palier à ce problème, plusieurs techniques peuvent être envisagées pour combler les trous, comme par exemple



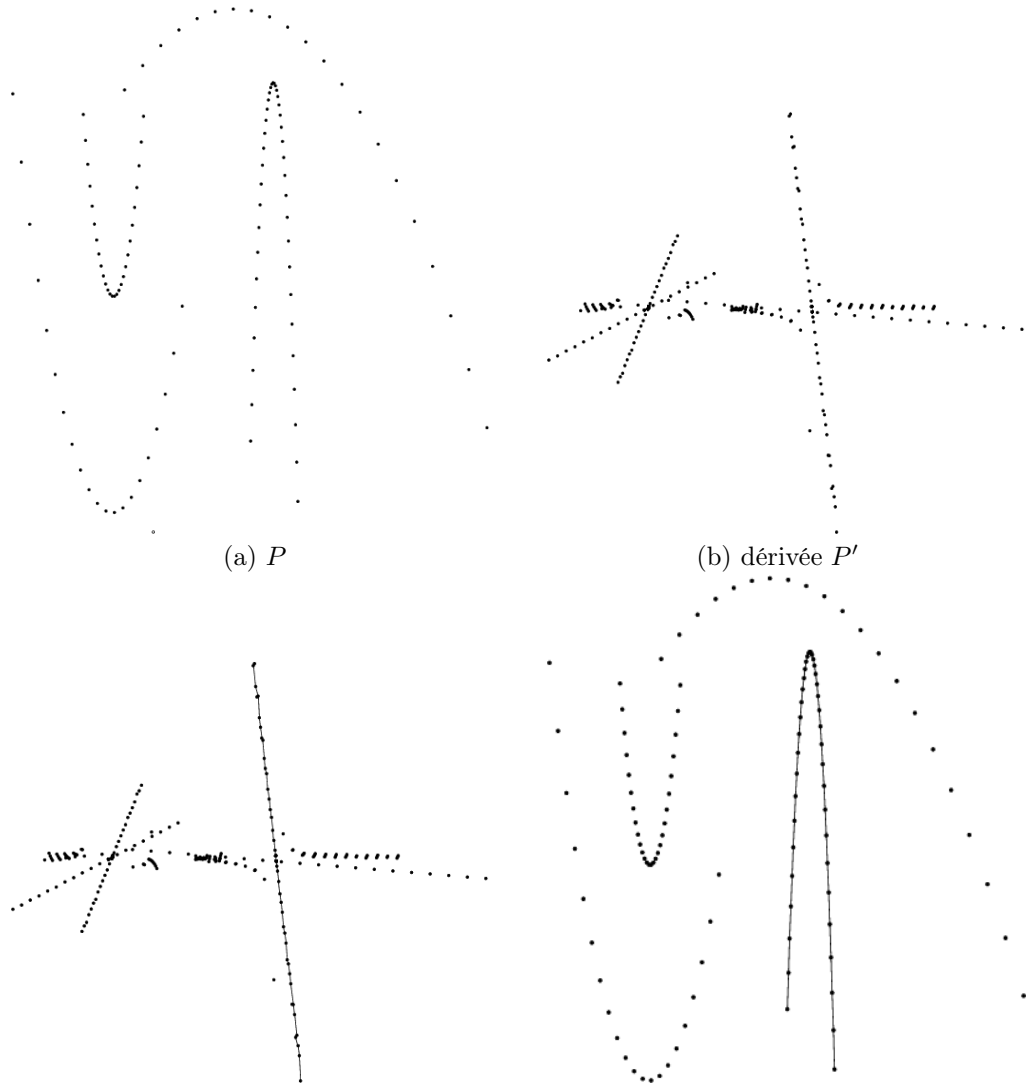
(a) $P \subset \mathbb{R}^3$ (b) TopoCrust3D(P)

(c) agrandissement de (b), présence de trous (d) remplissage des trous par continuité

Fig. 23: Reconstruction de surfaces avec l'extension en 3D de l'algorithme TopoCrust.

le prolongement de la surface par continuité (figure 23(d)). Dans ce contexte, il serait intéressant d'utiliser des techniques d'interpolation offertes par la régularisation discrète. Plus généralement, nous avons vu que d'autres méthodes de reconstruction, basées sur des graphes de voisinage, possèdent des propriétés hiérarchiques. C'est le cas par exemple des α-formes ou des β-squelettes. Il serait intéressant de construire pour chacun de ces graphes un algorithme similaire à TopoCrust, et de comparer les résultats obtenus à ceux fournis par ce dernier. Enfin, une autre perspective concernant la reconstruction serait d'étudier le comportement des méthodes à partir d'ensembles de points bruités. Dans ce contexte, la régularisation discrète peut être utilisée comme post-traitement, ou comme pré-traitement, pour améliorer la reconstruction.

– Dans le contexte de la détection de formes connues à partir d'un ensemble de



(a) P (b) dérivée P'

(c) détection d'une droite dans P' (d) parabole associée à (c) dans P.

Fig. 24: Détection de paraboles dans un ensemble de points 2D.

points, l'une des perspectives est de tester la méthode proposée dans des applications réelles, où la nature des surfaces à détecter est *a priori* connue. D'autre part, la méthode peut constituer une alternative à la reconstruction de formes générales, particulièrement lorsque l'ensemble de points est plongé dans un milieu bruité. Une voie très intéressante à poursuivre est de commencer par détecter des structures géométriques élémentaires, puis d'isoler les points de ces structures pour leur appliquer par la suite des méthodes classiques de reconstruction. Toujours dans le contexte de la détection, la méthode présentée concerne la détection de structures linéaires. D'une manière analogue, on peut envisager la détection de structures plus complexes, telles que des cercles, des paraboles, et plus généralement des formes quadratiques. Dans ce sens, nous avons déjà proposé une méthode générale pour détecter des paraboles et des cercles à partir d'un ensemble de points de plan, en introduisant la notion de dérivée d'un ensemble de points [Bougleux *et al.*, 2006]. L'idée est de transformer la détection de ces primitives géométriques en la détection de primitives plus simples. Par exemple, la détection de cercles revient à dé-

tecter des amas de points qui correspondent à leurs centres, et la détection de paraboles revient à détecter des droites qui correspondent à leurs dérivées. Pour cela, l'ensemble de points est transformé en un autre ensemble de points, en utilisant les propriétés différentielles des primitives à détecter, estimées à partir du diagramme de Voronoï. Ce concept prometteur, illustré à la figure 24, nécessite encore de nombreuses investigations théoriques.

– Dans le contexte du filtrage et de la simplification de données discrètes, le formalisme que nous avons introduit à permis d'unifier les méthodes de régularisation locale et non-locale des images, et de les étendre naturellement pour traiter des données définies dans des espaces de topologie quelconque et de grande dimension. Nous avons récemment montré que ce formalisme discret peut être utilisé pour traiter des problèmes de segmentation et de classification semi-supervisées de données [Ta *et al.*, 2007] (voir la figure 25 pour un exemple de segmentation semi-supervisée d'images, et un exemple de classification semi-supervisée de données non-organisées). Il serait intéressant de transcrire ce formalisme à d'autres problèmes inverses, tels que la reconstruction, le flot optique ou encore la classification.

– Une des perspectives générales serait d'étudier les problèmes de la segmentation et de la compression de données discrètes, particulièrement de maillages, en combinant les techniques de régularisation discrète aux techniques utilisées en géométrie algorithmique.

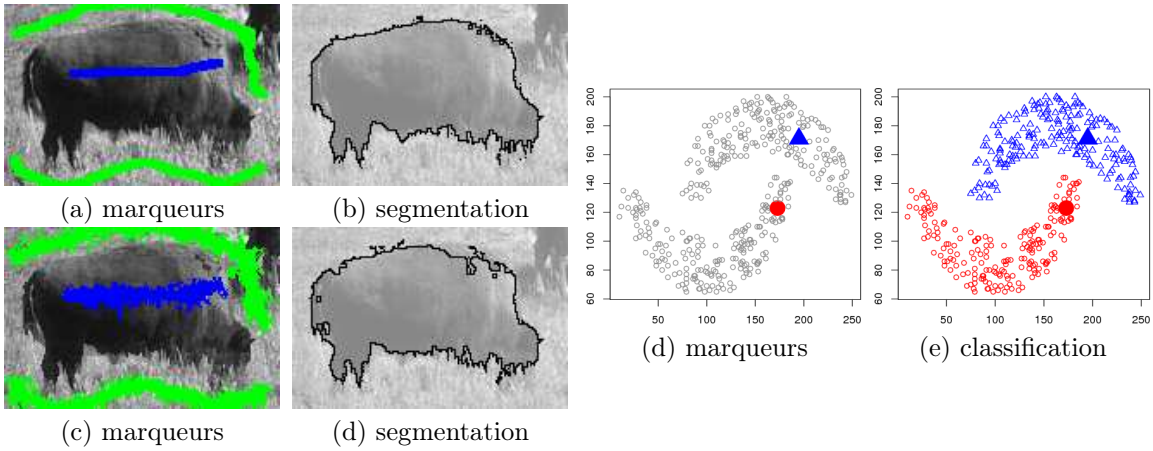


Fig. 25: Segmentation/classification semi-supervisée par régularisation non-locale discrète anisotrope. (a,b) Pour une image de pixels. (c,d) Pour une image pré-segmentée (RAG). (d,e) Pour un ensemble de données non-organisées (problème des deux lunes).

Bibliographie

[Althaus et Mehlhorn, 2000] Althaus, E. et Mehlhorn, K. (2000). TSP-based curve reconstruction in polynomial time. *Dans Proc. of the 11th Symposium on Discrete Algorithms (SODA)*, pages 686–695. ACM-SIAM.

[Alvarez *et al.*, 1993] Alvarez, L., Guichard, F., Lions, P.-L. et Morel, J.-M. (1993). Axioms and fundamental equations of image processing. *Archive for Rational Mechanics and Analysis*, 123(3):199–257.

[Amenta *et al.*, 1998] Amenta, N., Bern, M. W. et Eppstein, D. (1998). The crust and the β-skeleton : combinatorial curve reconstruction. *Graphical Models and Image Processing*, 60(2):125–135.

[Arbeláez et Cohen, 2004] Arbeláez, P. A. et Cohen, L. D. (2004). Energy partitions and image segmentation. *Journal of Mathematical Imaging and Vision*, 20(1-2):43–57.

[Attali, 1997] Attali, D. (1997). r-regular shape reconstruction from unorganized points. *Dans Proc. of the 13th annual Symp. on Computational Geometry*, pages 248–253. ACM Press.

[Attali *et al.*, 2007] Attali, D., Boissonnat, J.-D. et Edelsbrunner, H. (2007). Stability and computation of the medial axis–a state-of-the-art report. *Dans* T. Möller, B. Hamann, B. R., éditeur : *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration*. Springer-Verlag.

[Attali et Montanvert, 1997] Attali, D. et Montanvert, A. (1997). Computing and simplifying 2D and 3D continuous skeletons. *Computer Vision and Image Understanding*, 67(3):261–273.

[Aubert et Kornprobst, 2006] Aubert, G. et Kornprobst, P. (2006). *Mathematical Problems in Image Processing, Partial Differential Equations and the Calculus of Variations*. Numéro 147 de Applied Mathematical Sciences. Springer, 2nde édition.

[Aurenhammer et Imai, 1988] Aurenhammer, F. et Imai, H. (1988). Geometric relations among voronoi diagrams. *Geometriae Dedicata*, 27:65–75.

[Aurenhammer et Klein, 2000] Aurenhammer, F. et Klein, R. (2000). Voronoi diagrams. *Dans* Sack, J. et Urrutia, G., éditeurs : *Handbook of Computational Geometry*, chapitre V, pages 201–290.

[Bajaj et Xu, 2003] Bajaj, C. et Xu, G. (2003). Anisotropic diffusion on surfaces and functions on surfaces. *ACM Transactions on Graphics*, 22(1):4–32.

[Barash, 2002] Barash, D. (2002). A fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(6):844–847.

[Belkin *et al.*, 2004] Belkin, M., Matveeva, I. et Niyogi, P. (2004). Regularization and semi-supervised learning on large graphs. *Dans COLT*, pages 624–638.

[Belkin et Niyogi, 2001] Belkin, M. et Niyogi, P. (2001). Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in Neural Information Processing Systems (NIPS)*, 14.

[Bensoussan et Menaldi, 2005] Bensoussan, A. et Menaldi, J.-L. (2005). Difference equations on weighted graphs. *Journal of Convex Analysis*, 12(1):13–44.

[Bernardini et Bajaj, 1997] Bernardini, F. et Bajaj, C. L. (1997). Sampling and reconstructing manifolds using alpha-shapes. *Dans Proc. of the 9th Canadian Conf. on Computational Geometry*, pages 193–198.

[Bertero et Boccacci, 1998] Bertero, M. et Boccacci, P. (1998). *Introduction to inverse problems in imaging.* Institute of Physics Publishing.

[Biggs, 1974] Biggs, N. (1974). *Algebraic Graph Theory.* Numéro 67 de Cambridge Tracts in Mathematics. Cambridge University Press.

[Blum, 1967] Blum, H. (1967). A transformation for extracting new descriptors of shape. *Dans* Wathen-Dunn, W., éditeur : *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press.

[Boissonnat et Geiger, 1992] Boissonnat, J.-D. et Geiger, B. (1992). Three-dimensional reconstruction of complex shapes based on the Delaunay triangulation. Rapport technique 1697, INRIA.

[Boissonnat et Oudot, 2005] Boissonnat, J.-D. et Oudot, S. (2005). Provably good sampling and meshing of surfaces. *Graphical Models*, 67(5).

[Bougleux et Elmoataz, 2005] Bougleux, S. et Elmoataz, A. (2005). Image smoothing and segmentation by graph regularization. *Dans* Bebis, G., Boyle, R. D., Koracin, D. et Parvin, B., éditeurs : *Proc. of the 1st International Symposium on Visual Computing (ISVC)*, volume 3656 de *LNCS*, pages 745–752. Springer.

[Bougleux *et al.*, 2007a] Bougleux, S., Elmoataz, A. et Melkemi, M. (2007a). Discrete regularization on weighted graphs for image and mesh filtering. *Dans* Sgallari, F., Murli, A. et Paragios, N., éditeurs : *Proc. of the 1st Int. Conf. on Scale Space and Variational Methods in Computer Vision (SSVM)*, volume 4485 de *LNCS*, pages 128–139. Springer.

[Bougleux *et al.*, 2007b] Bougleux, S., Elmoataz, A. et Melkemi, M. (2007b). Local and nonlocal discrete regularization on weighted graphs for image and mesh processing. Soumis à International Journal of Computer Vision, Special issue on SSVM'07 Conference.

[Bougleux *et al.*, 2005] Bougleux, S., Melkemi, M. et Elmoataz, A. (2005). Structure detection from a 3D set of points with anisotropic alpha shapes. *Dans* Kamel, M. S. et Campilho, A., éditeurs : *2nd International Conference on Image Analysis and Recognition*, volume 3656 de *LNCS*, pages 651–658. Springer.

[Bougleux *et al.*, 2006] Bougleux, S., Melkemi, M. et Elmoataz, A. (2006). Geometrical algorithms to detect patterns from a set of points. *Dans* Gavrilova, M. L., éditeur : *3rd International Symposium on Voronoi Diagrams in Science and Engineering (ISVD'06)*, pages 94–101. IEEE Computer Society.

[Bougleux *et al.*, 2007c] Bougleux, S., Melkemi, M. et Elmoataz, A. (2007c). Local beta-crusts for simple curves reconstruction. *Dans* Gold, C. M., éditeur : *4th International Symposium on Voronoi Diagrams in Science and Engineering (ISVD'07)*, pages 48–57. IEEE Computer Society.

[Bougleux *et al.*, 2007d] Bougleux, S., Melkemi, M. et Elmoataz, A. (2007d). Structure detection from a 3D set of points with anisotropic alpha shapes. *International Journal of Image and Graphics*, 7(4):689–708.

[Boykov *et al.*, 2001] Boykov, Y., Veksler, O. et Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239.

[Brandt, 1994] Brandt, J. W. (1994). Convergence and continuity criteria for discrete approximations of the continuous planar skeletons. *CVGIP : Image Understanding*, 59(1):116–124.

[Brandt et Algazi, 1992] Brandt, J. W. et Algazi, V. R. (1992). Continuous skeleton computation by Voronoi diagram. *CVGIP : Image Understanding*, 55(3):329–337.

[Buades *et al.*, 2005] Buades, A., Coll, B. et Morel, J.-M. (2005). A review of image denoising algorithms, with a new one. *Multiscale Modeling and Simulation*, 4(2):490–530.

[Chambolle, 2004] Chambolle, A. (2004). An algorithm for total variation minimization and applications. *Journal of Mathematical Image and Vision*, 20:89–97.

[Chambolle, 2005] Chambolle, A. (2005). Total variation minimization and a class of binary MRF models. *Dans* Rangarajan, A., Vemuri, B. C. et Yuille, A. L., éditeurs : *Proc. of the 5th International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition (MMCVPR)*, volume 3757 de *LNCS*, pages 136–152. Springer.

[Chan *et al.*, 2001] Chan, T., Osher, S. et Shen, J. (2001). The digital TV filter and nonlinear denoising. *IEEE Transactions on Image Processing*, 10(2):231–241.

[Chan et Shen, 2005] Chan, T. F. et Shen, J. (2005). *Image Processing and Analysis - variational, PDE, wavelets, and stochastic methods.* SIAM.

[Chung, 1997] Chung, F. R. K. (1997). Spectral graph theory. *CBMS Regional Conference Series in Mathematics*, 92:1–212.

[Coifman *et al.*, 2005] Coifman, R., Lafon, S., Lee, A., Maggioni, M., Nadler, B., Warner, F. et Zucker, S. (2005). Geometric diffusions as a tool for harmonic analysis and structure definition of data. *Proc. of the National Academy of Sciences*, 102(21).

[Coifman *et al.*, 2006] Coifman, R., Lafon, S., Maggioni, M., Keller, Y., Szlam, A., Warner, F. et Zucker, S. (2006). Geometries of sensor outputs, inference, and information processing. *Dans Proc. of the SPIE : Intelligent Integrated Microsystems*, volume 6232.

[Cvetković *et al.*, 1980] Cvetković, D. M., Doob, M. et Sachs, H. (1980). *Spectra of Graphs, Theory and Application.* Pure and Applied Mathematics. Academic Press.

[Darbon et Sigelle, 2004] Darbon, J. et Sigelle, M. (2004). Exact optimization of discrete constrained total variation minimization problems. *Dans* Klette, R. et Zunic, J. D., éditeurs : *Proc. of the 10th Int. Workshop on Combinatorial Image Analysis*, volume 3322 de *LNCS*, pages 548–557.

[Delaunay, 1934] Delaunay, B. (1934). Sur la sphère vide. *Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk*, 7:793–800.

[Desbrun *et al.*, 1999] Desbrun, M., Meyer, M., Schröder, P. et Barr, A. H. (1999). Implicit fairing of irregular meshes using diffusion and curvature flow. *Dans SIGGRAPH'99 Conference Proceedings*, pages 317–324. ACM.

[Dey, 2007] Dey, T. K. (2007). *Curve and Surface Reconstruction.* Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press.

[Dey et Kumar, 1999] Dey, T. K. et Kumar, P. (1999). A simple provable algorithm for curve reconstruction. *Dans Proc. of the 10th Symposium on Discrete Algorithms (SODA)*, pages 893–894. ACM-SIAM.

[Dey *et al.*, 2000] Dey, T. K., Mehlhorn, K. et Ramos, E. A. (2000). Curve reconstruction : Connecting dots with good reason. *Computational Geometry*, 15(4):229–244.

[Dey et Wenger, 2001] Dey, T. K. et Wenger, R. (2001). Reconstructing curves with sharp corners. *Computational Geometry*, 19(2-3):89–99.

[Dey et Wenger, 2002] Dey, T. K. et Wenger, R. (2002). Fast reconstruction of curves with sharp corners. *Int. Journal of Computational Geometry and Applications*, 12(5):353–400.

[Dirichlet, 1850] Dirichlet, G. L. (1850). über die reduktion der positiven quadratischen formen mit drei unbestimmten ganzen zahlen. *Journal für die Reine und Angewandte Mathematik*, 40:209–227.

[Edelsbrunner, 2001] Edelsbrunner, H. (2001). *Geometry and Topology for Mesh Generation.* Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press.

[Edelsbrunner *et al.*, 1983] Edelsbrunner, H., Kirkpatrick, D. G. et Seidel, R. (1983). On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, IT-29:65–75.

[Edelsbrunner et Mücke, 1994] Edelsbrunner, H. et Mücke, E. P. (1994). Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13(1):43–72.

[Edelsbrunner et Shah, 1994] Edelsbrunner, H. et Shah, N. R. (1994). Triangulating topological spaces. *Dans SCG'94 : Proc. of the 10th annual Symposium on Computational Geometry*, pages 285–292. ACM Press.

[Efros et Leung, 1999] Efros, A. et Leung, T. (1999). Texture synthesis by non-parametric sampling. *Dans Proc. of the International Conference on Computer Vision (ICCV)*, pages 1033–1038. IEEE Computer Society.

[Elmoataz *et al.*, 2007] Elmoataz, A., Lézoray, O. et Bougleux, S. (2007). Nonlocal discrete regularization on weighted graphs : a framework for image and manifold processing. Rapport technique hal-00163573, Université de Caen Basse-Normandie. Soumis à IEEE Transactions on Image Processing.

[Figueiredo et Gomez, 1995] Figueiredo, L. H. D. et Gomez, J. D. M. (1995). Computational morphology of curves. *Visual Computing*, 11:105–112.

[Friedman et Tillich, 2004] Friedman, J. et Tillich, J.-P. (2004). Wave equations for graphs and the edge-based laplacian. *Pacific Journal of Mathematics*, 216(2):229–266.

[Funke et Ramos, 2001] Funke, S. et Ramos, E. A. (2001). Reconstructing a collection of curves with corners and endpoints. *Dans Proceedings of the 12th Symposium on Discrete algorithms (SODA)*, pages 344–353. ACM-SIAM.

[Gabriel et Sokal, 1969] Gabriel, K. R. et Sokal, R. R. (1969). A new statistical approach to geographic variation analysis. *Systematic Zoology*, 18(3):259–278.

[Giesen, 2000] Giesen, J. (2000). Curve reconstruction , the travelling salesman problem and menger's theorem. *Discrete and Computational Geometry*, 24:577–603.

[Gilboa et Osher, 2006] Gilboa, G. et Osher, S. (2006). Nonlocal linear image regularization and supervised segmentation. CAM Report 06-47, UCLA, Los Angeles, USA.

[Gilboa et Osher, 2007a] Gilboa, G. et Osher, S. (2007a). Nonlocal linear image regularization and supervised segmentation. *SIAM Multiscale Modeling and Simulation*, 6(2):595–630.

[Gilboa et Osher, 2007b] Gilboa, G. et Osher, S. (2007b). Nonlocal operators with applications to image processing. CAM Report 07-23, UCLA, Los Angeles.

[Gold, 1999] Gold, C. (1999). Crust and anti-crust : a one-step boundary and skeleton extraction algorithm. *Dans Proc. of the 15th annual Symposium on Computational Geometry (SCG'99)*, pages 189–196. ACM Press.

[Gold et Snoeyink, 2001] Gold, C. M. et Snoeyink, J. (2001). A one-step crust and skeleton extraction algorithm. *Algorithmica*, 30(2):144–163.

[Guibas et Stolfi, 1985] Guibas, L. et Stolfi, J. (1985). Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Tansactions on Graphics*, 4:74–123.

[Hettich *et al.*, 1998] Hettich, S., Blake, C. et Merz, C. (1998). UCI repository of machine learning databases. Rapport technique, Dept. of Information and Computer Sciences, University of California, Irvine, http ://www.ics.uci.edu/~mlearn/MLRepository.html.

[Hildebrandt et Polthier, 2004] Hildebrandt, K. et Polthier, K. (2004). Anisotropic filtering of non-linear surface features. *Eurographics 2004 : Computer Graphics Forum*, 23(3):391–400.

[Jaillet *et al.*, 2000] Jaillet, F., Ghamri-Doudane, M. Y., Melkemi, M. et Baskurt, A. (2000). Adaptive contour sampling and coding using skeleton and curvature. *Dans Proc. of the Int. Conf. on Image Processing (ICIP)*, volume 2, pages 899–902. IEEE Computer Society.

[Kervrann *et al.*, 2007] Kervrann, C., Boulanger, J. et Coupé, P. (2007). Bayesian non-local means filter, image redundancy and adaptive dictionaries for noise removal. *Dans* Sgallari, F., Murli, A. et Paragios, N., éditeurs : *Proc. of the 1st Int. Conf. on Scale Space and Variational Methods in Computer Vision (SSVM)*, volume 4485 de *LNCS*, pages 520–532. Springer.

[Kimberling, 1998] Kimberling, C. (1998). Triangle centers and central triangles. *Congressus Numerantium*, 129:1–295.

[Kinderman *et al.*, 2005] Kinderman, S., Osher, S. et Jones, S. (2005). Deblurring and denoising of images by nonlocal functionals. *Multiscale Modeling and Simulation*, 4(4):1091–1115.

[Kirkpatrick et Radke, 1985] Kirkpatrick, D. G. et Radke, J. D. (1985). A framework for computational morphology. *Dans* Toussaint, G. T., éditeur : *Computational Geometry*, pages 217–248. Elsevier.

[Labelle et Shewchuk, 2003] Labelle, F. et Shewchuk, J. (2003). Anisotropic Voronoi diagrams and guaranteed-quality anisotropic mesh generation. *Dans Proc. of the 9th Annual Symp. on Computational Geometry (SCG'03)*, pages 191–200. ACM Press.

[Lafon et Lee, 2006] Lafon, S. et Lee, A. B. (2006). Diffusion maps and coarse-graining : A unified framework for dimensionality reduction, graph partitioning, and data set parameterization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1393–1403.

[Lenz, 2006] Lenz, T. (2006). How to sample and reconstruct curves with unusual features. *Dans Proc. of the 22nd European Workshop on Computational Geometry (EWCG)*, pages 29–32.

[Lézoray *et al.*, 2007a] Lézoray, O., Bougleux, S. et Elmoataz, A. (2007a). Parameterless discrete regularization on graphs for color image filtering. *Dans* Kamel, M. S. et Campilho, A. C., éditeurs : *Proc. of the 4th International Conference on Image Analysis and Recognition (ICIAR)*, volume 4633 de *LNCS*, pages 46–57. Springer.

[Lézoray *et al.*, 2006] Lézoray, O., Elmoataz, A. et Bougleux, S. (2006). Régularisation sur graphe pour le traitement d'images couleur. *Dans 10ème Conférence Compression et Représentation des Signaux Audiovisuels (CORESA)*, pages 236–241, Caen, France.

[Lézoray *et al.*, 2007b] Lézoray, O., Elmoataz, A. et Bougleux, S. (2007b). Graph regularization for color image processing. *Computer Vision and Image Understanding, Special issue on color image processing*, 107(1-2):38–55.

[Matula et Sokal, 1980] Matula, D. W. et Sokal, R. R. (1980). Properties of Gabriel graphs relevant to geographic variation research and the clustering of points in the plane. *Geographical Analysis*, 12(3):205–222.

[Melkemi, 1997] Melkemi, M. (1997). A-shapes of a finite point set. *Dans Proc. of the 13th annual Symp. on Computational Geometry (SCG)*, pages 367–369. ACM Press.

[Melkemi et Djebali, 2000] Melkemi, M. et Djebali, M. (2000). Computing the shape of a planar points set. *Pattern Recognition*, 33(9):1423–1436.

[Melkemi et Djebali, 2001] Melkemi, M. et Djebali, M. (2001). Elliptic diagrams : application to patterns detection from a finite set of points. *Pattern Recognition Letters*, 22(8):835–844.

[Melkemi et Melkemi, 1995] Melkemi, M. et Melkemi, L. (1995). An algorithm for detecting dot patterns. *Dans* Earnshaw, R. A. et Vince, J. A., éditeurs : *Computer graphics : developments in virtual environments*, pages 161–169. Academic Press Ltd.

[Mrázek *et al.*, 2006] Mrázek, P., Weickert, J. et Bruhn, A. (2006). On robust estimation and smoothing with spatial and tonal kernels. *Dans Geometric Properties from Incomplete Data*, volume 31 de *Computational Imaging and Vision*, pages 335–352. Springer.

[Mücke, 1993] Mücke, E. P. (1993). *Shapes and Implementations in three-dimensional geometry.* Thèse de doctorat, Dept. of Computer Science, University of Illinois, Urbana-Champaign.

[Mukhopadhay et Das, 2006] Mukhopadhay, A. et Das, A. (2006). An RNG-based heuristic for curve reconstruction. *Dans* Gavrilova, M. L., éditeur : *Proc. of the 3rd International Symposium on Voronoi Diagrams in Science and Engineering (ISVD)*, pages 246–251. IEEE Computer Society.

[Ohtake *et al.*, 2001] Ohtake, Y., Belyaev, A. et Bogaeski, I. (2001). Mesh regularization and adaptive smoothing. *Computer-Aided Design*, 33:789–800.

[Okabe *et al.*, 2000] Okabe, A., Boots, B., Sugihara, K. et Chiu, S. N. (2000). *Spatial Tessellations - Concepts and Applications of Voronoi Diagrams.* John Wiley, 2nd édition.

[O'Rourke et Toussaint, 2004] O'Rourke, J. et Toussaint, G. T. (2004). Pattern recognition. *Dans* Goodman, J. E. et O'Rourke, J., éditeurs : *Handbook of Discrete and Computational Geometry*, chapitre 51, pages 1135–1162. Chapman & Hall, second édition.

[Osher et Shen, 2000] Osher, S. et Shen, J. (2000). Digitized PDE method for data restoration. *Dans* Anastassiou, E. G. A., éditeur : *In Analytical-Computational methods in Applied Mathematics*, pages 751–771. Chapman & Hall/CRC.

[Overmars *et al.*, 2000] Overmars, M., De Berg, M., Van Kreveld, M. et Schwarzkopf, O. (2000). *Computational Geometry - Algorithms and Applications.* Springer-Verlag.

[Pinkall et Polthier, 1993] Pinkall, U. et Polthier, K. (1993). Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, 2(1):15–36.

[Poggio *et al.*, 1984] Poggio, T. A., Torre, V. et Koch, C. (1984). Ill-posed problems and regularization analysis in early vision. *Dans Image Understanding Workshop*, pages 257–263.

[Requardt, 1997] Requardt, M. (1997). A new approach to functional analysis on graphs, the connes-spectral triple and its distance function.

[Rossignac, 2004] Rossignac, J. (2004). *Handbook of Discrete and Computational Geometry (2nd edition)*, chapitre Surface simplification and 3D geometry compression, pages 1209–1240. CRC Press.

[Rudin *et al.*, 1992] Rudin, L. I., Osher, S. et Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4):259–268.

[Schmitt et Mattioli, 1993] Schmitt, M. et Mattioli, J. (1993). *Morphologie Mathématique*. Masson, Paris.

[Seneta, 1981] Seneta, E. (1981). *Non-negative matrices and Markov chains*. Springer.

[Shi et Malik, 2000] Shi, J. et Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905.

[Smith et Brady, 1997] Smith, S. et Brady, J. (1997). SUSAN - a new approach to low level image processing. *International Journal of Computer Vision*, 23:45–78.

[Smola et Kondor, 2003] Smola, A. J. et Kondor, R. I. (2003). Kernels and regularization on graphs. *Dans* Schölkopf, B. et Warmuth, M. K., éditeurs : *Proc of the 16th Annual Conf. on Computational Learning Theory (COLT)*, volume 2777 de *LNCS*, pages 144–158. Springer.

[Szlam *et al.*, 2006] Szlam, A. D., Maggioni, M. et Coifman, R. R. (2006). A general framework for adaptive regularization based on diffusion processes on graphs. Rapport technique YALE/DCS/TR1365, YALE.

[Ta *et al.*, 2007] Ta, V. T., Bougleux, S., Elmoataz, A. et Lézoray, O. (2007). Nonlocal anisotropic discrete regularization for image, data filtering and clustering. Rapport technique hal-00187165, Université de Caen Basse-Normandie. Soumis à Computer Vision and Image Understanding, Special Issue on Discrete Optimization in Computer Vision.

[Taubin, 1995] Taubin, G. (1995). A signal processing approach to fair surface design. *Dans* Press, A., éditeur : *SIGGRAPH'95 : Proc. of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, pages 351–358.

[Taubin, 2000] Taubin, G. (2000). Geometric signal processing on polygonal meshes. *Dans Eurographics, State of the Art Report*.

[Teichmann et Capps, 1998] Teichmann, M. et Capps, M. (1998). Surface reconstruction with anisotropic density-scaled alpha shapes. *Dans Proc. of the Conf. on Visualization (VIS'98)*, pages 67–72. IEEE Computer Society.

[Tikhonov et Arsenin, 1977] Tikhonov, A. N. et Arsenin, V. Y. (1977). *Solutions of ill-posed problems*. Winston & Sons.

[Tomasi et Manduchi, 1998] Tomasi, C. et Manduchi, R. (1998). Bilateral filtering for gray and color images. *Dans Proc. of the IEEE International Conference on Computer Vision (ICCV)*, pages 836–846.

[Urquhart, 1982] Urquhart, R. (1982). Graph theoretical clustering based on limited neighborhood sets. *Pattern Recognition*, 15(3):173–187.

[Veltkamp, 1988] Veltkamp, R. C. (1988). The γ-neighborhood graph for computational morphology. *Dans Proc. of Computing Science in the Netherlands*, pages 451–462.

[Voronoï, 1907] Voronoï, G. (1907). Nouvelles applications des paramètres continus à la théorie des formes quadratiques. *Journal für die Reine und Angewandte Mathematik*, 133:97–178.

[Xu, 2004] Xu, G. (2004). Discrete laplace-beltrami operators and their convergence. *Computer Aided Geometric Design*, 21:767–784.

[Xu *et al.*, 2006] Xu, G., Pan, Q. et Bajaj, C. (2006). Discrete surface modeling using partial differential equations. *Computer Aided Geometric Design*, 23:125–145.

[Yoshizawa *et al.*, 2006] Yoshizawa, S., Belyaev, A., et Seidel, H.-P. (2006). Smoothing by example : mesh denoising by averaging with similarity-based weights. *Dans Proc. of the International Conference on Shape Modeling and Applications (SM'06)*, pages 38–44.

[Zhou et Schölkopf, 2004] Zhou, D. et Schölkopf, B. (2004). A regularization framework for learning from graph data. *Dans ICML Workshop on Statistical Relational Learning and Its Connections to Other Fields*, pages 132–137.

[Zhou et Schölkopf, 2005] Zhou, D. et Schölkopf, B. (2005). Regularization on discrete spaces. *Dans* Kropatsch, W. G., Sablatnig, R. et Hanbury, A., éditeurs : *Proc. of the 27th DAGM Symposium*, volume 3663 de *LNCS*, pages 361–368. Springer.

Résumé. Cette thèse traite des problématiques de structuration et de traitement de données discrètes organisées ou non. Elle se décompose en deux parties. La première partie concerne la structuration de données représentées par des ensembles de points du plan ou de l'espace Euclidien. Dans ce contexte, nous considérons les problèmes de la reconstruction polygonale de courbes planaires et de la détection de formes géométriques 3D connues. Ces deux problèmes sont traités par des techniques de géométrie algorithmique et combinatoire, basées sur le diagramme de Voronoï et la triangulation de Delaunay. Dans le cadre de la reconstruction de courbes planaires, nous proposons une famille hiérarchique de sous-graphes du graphe de Gabriel, que nous appelons les β-Crusts locaux. Nous étudions les propriétés de cette famille, qui nous permettent de concevoir un algorithme de reconstruction des courbes simples. Ensuite, nous proposons une méthode de détection de formes géométriques connues à partir d'un ensemble de points 3D, plongés dans un milieu bruité ou non. Cette méthode est basée sur une extension des α-formes, générées à partir de boules ellipsoïdales. Dans une deuxième partie, nous traitons le problème de la régularisation de données par des méthodes variationnelles discrètes, sur graphes pondérés de topologie quelconque. Pour cela, nous proposons une large famille de fonctionnelles discrètes, basées sur les normes $\mathcal{L}_2$ et $\mathcal{L}_p$ du gradient. Ceci conduit à des processus de diffusion linéaire ou non-linéaire sur graphes. Ce formalisme est appliqué à des problèmes de restauration, de lissage, et de simplification d'images, de maillages et de données discrètes non-organisées.

Title. Reconstruction, Detection and Regularization of Discrete Data

Abstract. This thesis deals with the problems of structuring and processing discrete data, organized or not. It is organized into two parts. The first part deals with the structuration of data represented by sets of points of the Euclidean plane or space. In this context, we consider the problems of the polygonal reconstruction of planar curves and the detection geometric 3D shapes. These two problems are handled by computational and combinatorial geometry techniques, based on the Voronoi diagram and the Delaunay triangulation. In the context of planar curve reconstruction, we propose a hierarchical family of subgraphs of the Gabriel graph, which we call the local β-crusts. We study the properties of this family, which allow us to design a simple curves reconstruction algorithm. Then, we propose a method for detecting known shapes from a set of 3D points, embedded or not into a noisy background. This method is based on an extension of the α-shapes, generated from ellipsoidal balls. In a second part, we treat the problem of data regularization by discrete variational methods on weighted graphs of arbitrary topology. For this, we propose a broad family of discrete functionals, based on the $\mathcal{L}_2$ and $\mathcal{L}_p$ norms of the graph gradient. This leads to several linear and nonlinear diffusion processes on graphs. This formalism is applied to smooth, to restore, and to simplify discrete images, meshes and non-organized discrete data.

Mots-clés. Géométrie algorithmique, Infographie, Interpolation, Traitement d'images – Techniques numériques, Lissage, Équations à opérateurs non-linéaires, Processus de diffusion ; Reconstruction de formes, diagramme de Voronoï, Méthodes variationnelles sur graphes.

Discipline. Informatique – Traitement d'images.

GREYC–CNRS UMR 6072, Équipe Image
ENSICAEN, 6 Bd du Maréchal Juin, 14050 Caen Cedex France.